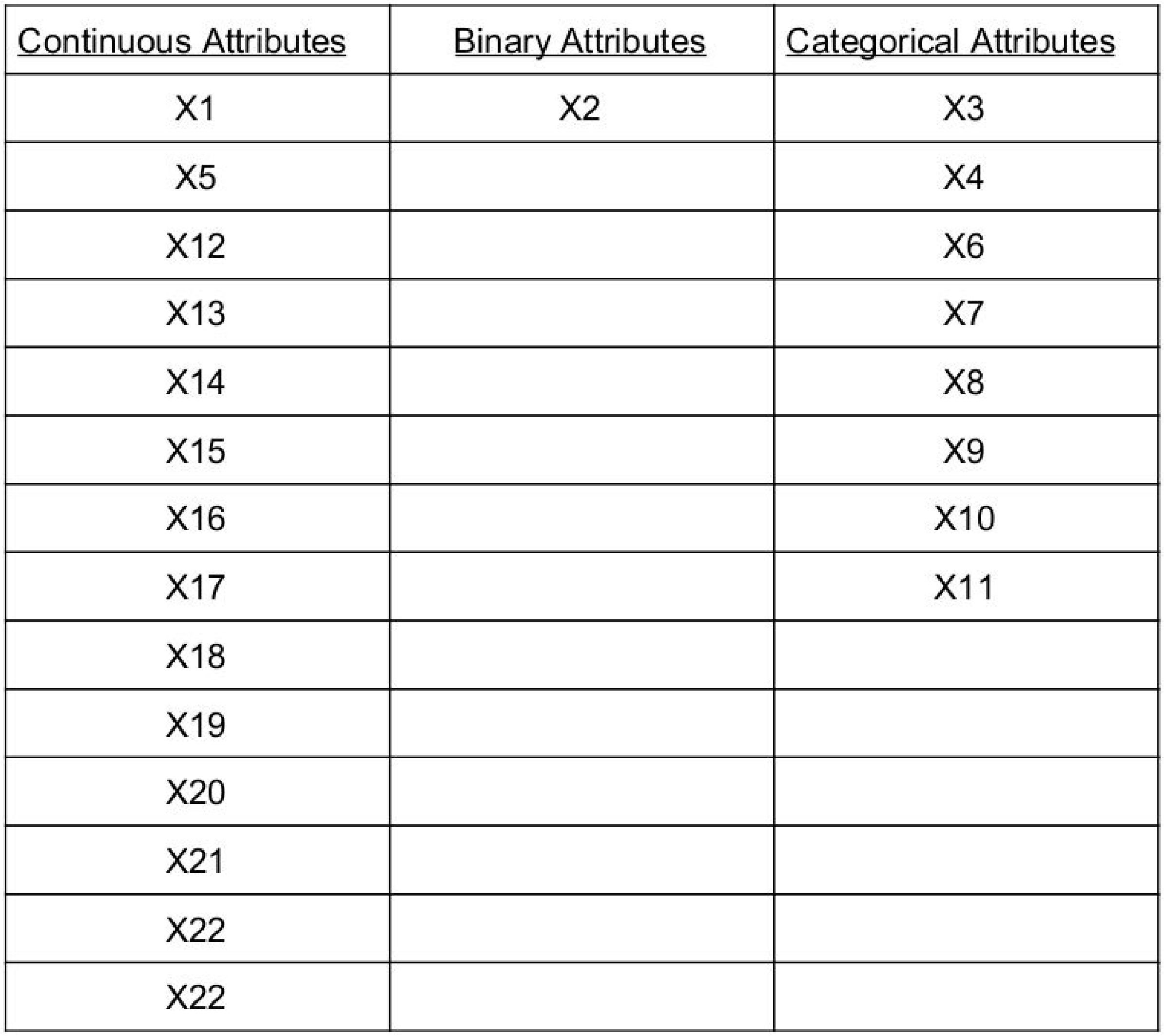
COL774 - MACHINE LEARNING

ASSIGNMENT 3 REPORT

Submitted By:
Mehak
2018MCS2143

1. Decision Trees



Continuous Attributes	Binary Attributes	Categorical Attributes
X1	X2	X3
X5		X4
X12		X6
X13		X7
X14		X8
X15		X9
X16		X10
X17		X11
X18		
X19		
X20		
X21		
X22		
X22		

Table 1: Different types of attributes in the data set

For continuous attributes, I converted them to binary based on whether the value is greater than the median threshold or not.
For binary attr, I did boolean(two-way) split.
For categorical, I did the multi-way split.

a. Accuracies against number of nodes in the tree as tree grows

BFS growth

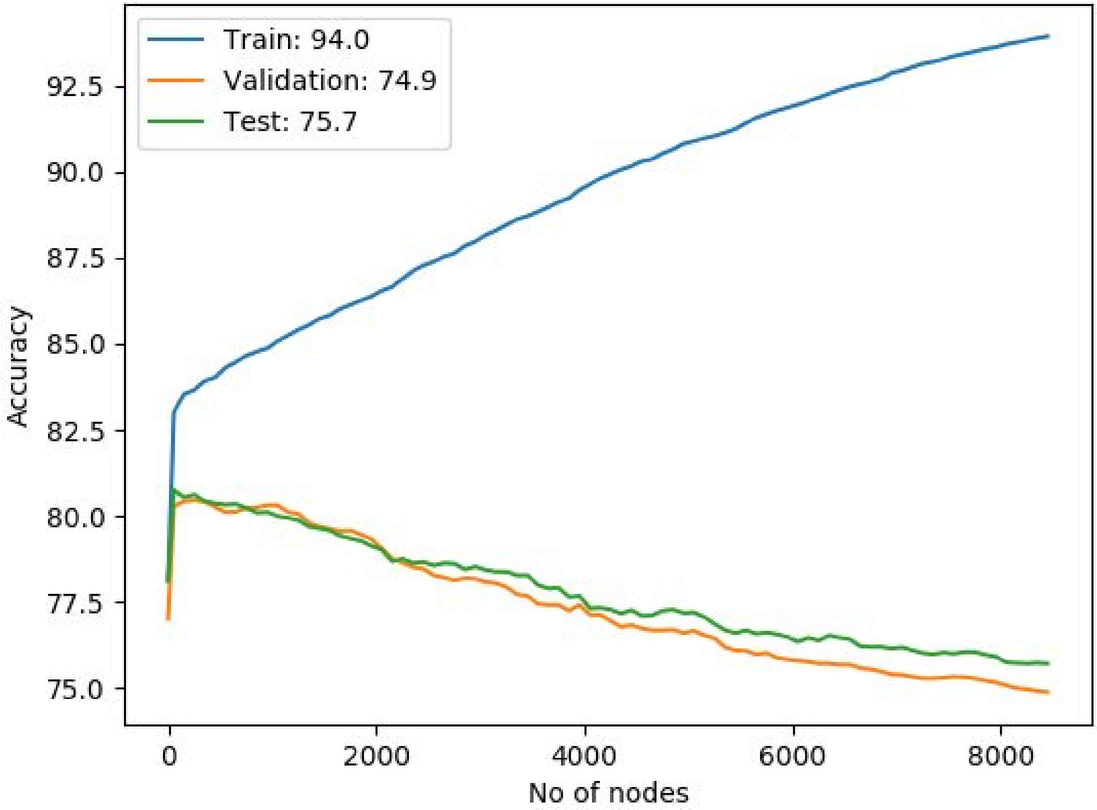
Tree height 22

Number of nodes 8555

Accuracy (training set) 93.97222222222223

Accuracy (validation set) 74.83333333333333

Accuracy (testing set) 75.68333333333334

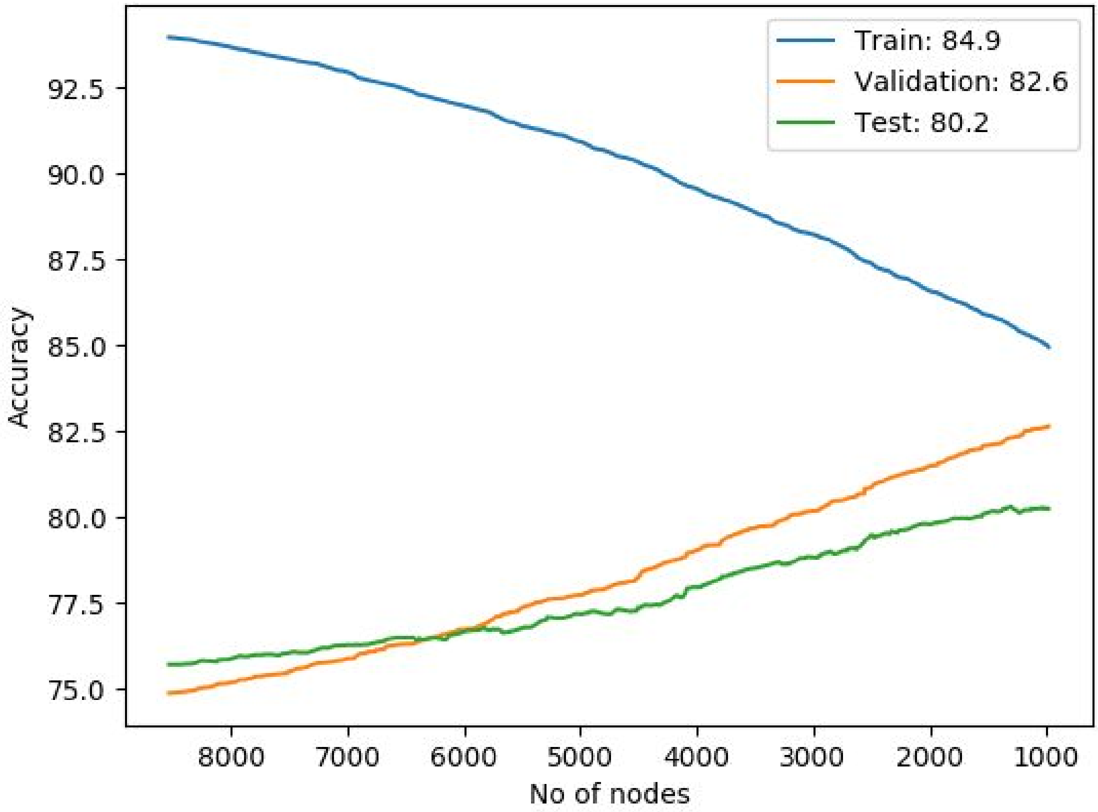


Observations:

Decision Tree with a single node predicts the majority class giving the accuracy of ~78% . As number of nodes increases, Training accuracy increases while Testing and Validation accuracies decreases i.e. overfitting happens.

b. Post pruning based on validation set

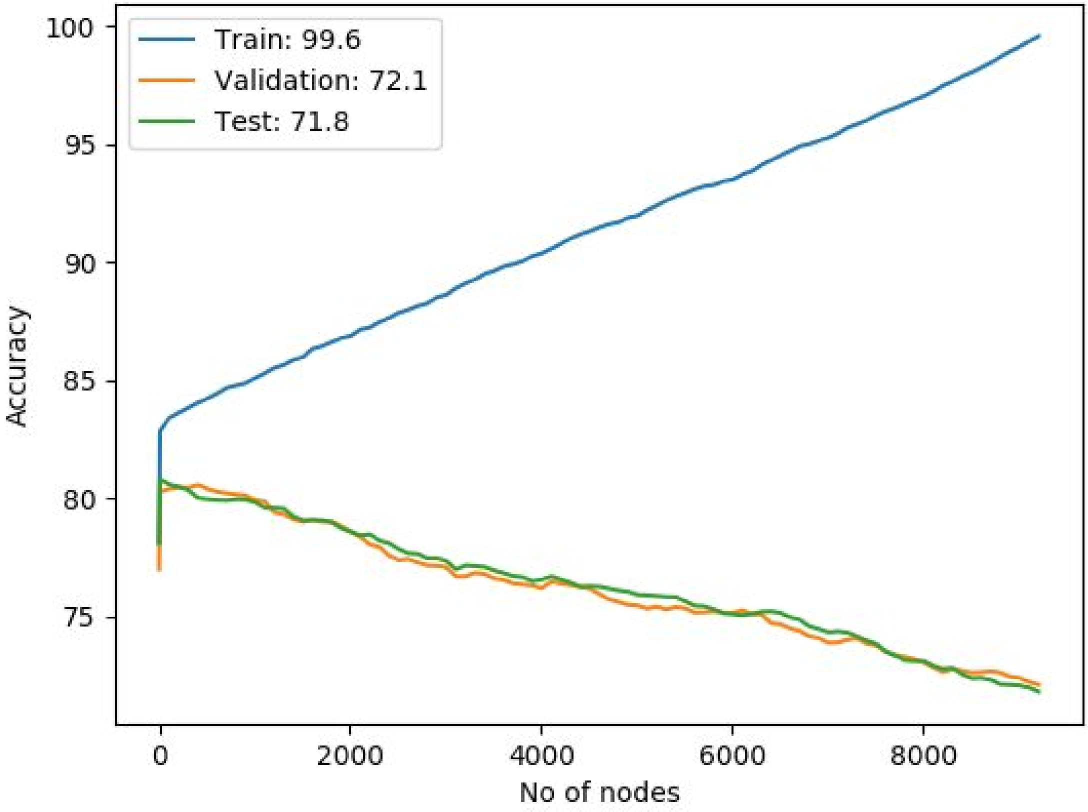
BFS growth
Tree height 15
Number of nodes 981
Accuracy (training set) 84.93888888888888
Accuracy (validation set) 82.63333333333334
Accuracy (testing set) 80.23333333333333



Observations:
Pruning decreases the height of tree to 15 and number of nodes from 8555 to 981 improving validation accuracy from 74% to 82% and testing accuracy from 75% to 80%. And hence helps in generalizing well while reducing overfitting.

c. Using medians dynamically (without pruning)

Tree height 19
Number of nodes 9309
Accuracy (training set) 99.79444444444444
Accuracy (validation set) 72.01666666666667
Accuracy (testing set) 71.75



Numerical Attributes split multiple times in a branch:
1 [120000.0, 50000.0, 70000.0, 80000.0, 95000.0, 110000.0]
5 [36.0, 41.5, 48.0, 50.0, 51.0, 54.0]
12 [107948.0, 34711.5, 59307.0, 46385.0]
13 [44413.0, 49256.0, 59156.5]

```
14 [19633.0, 28010.0, 25103.0, 27376.0]
15 [40385.0, 48097.0, 48635.0]
16 [39369.0, 45794.0, 41296.0]
17 [15500.5, 18929.0, 17132.0]
18 [3101.5, 2000.0, 1530.0]
19 [1500.0, 1602.0, 1803.0]
20 [1500.0, 1287.0, 1058.0]
21 [2882.5, 3360.0, 5000.0]
22 [1162.0, 2000.0, 1458.0]
23: [2453.0, 4031.0]
```

Observations:
The training accuracy boosts to 99.6 while testing and validation set accuracy is ~72% which shows how bad it overfits the data. Also as same attributes are split multiple times based on median, number of nodes of tree is increased.

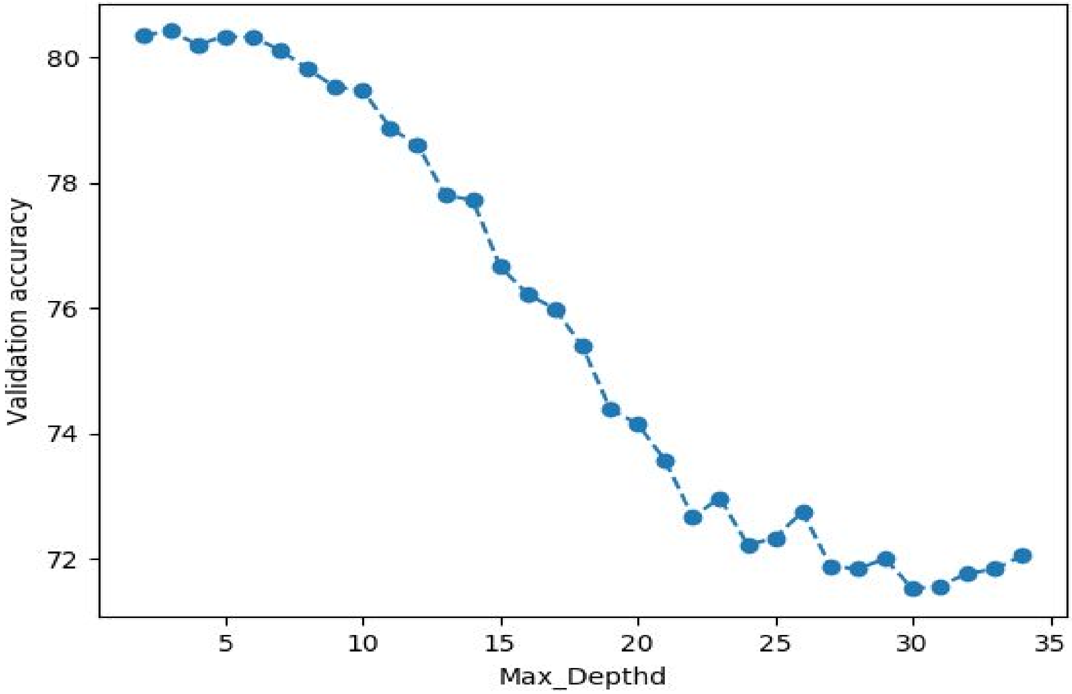
d. Using Sklearn library

Scikit-learn implementation :
(i) min_sample_leaf : A split at any depth will only be considered if it
leaves at least min_sample_leaf samples in both left and right branches.
node.
(ii) min_sample_split : Min samples required to split an internal
(iii) max_depth : Max height of the tree.

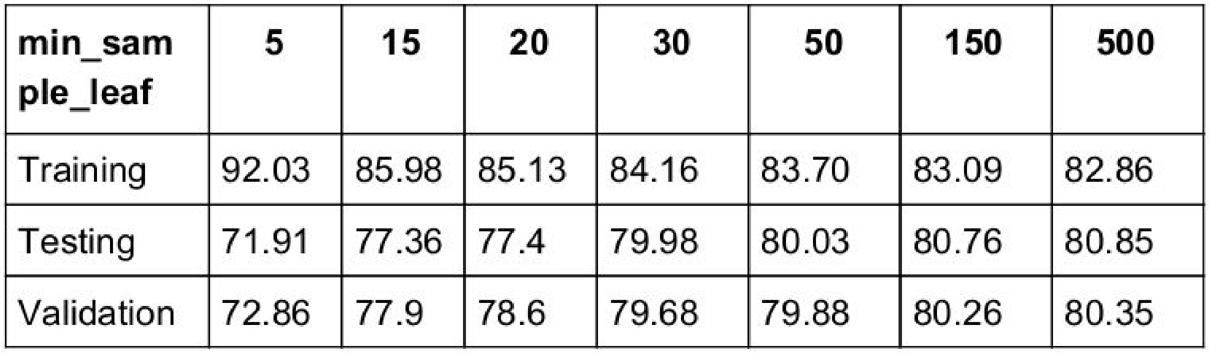
With default parameters
(max_depth=None,min_samples_split=2,min_samples_leaf=1)

```
Accuracy on training set  99.96111111111111
Accuracy on validation set  72.39999999999999
Accuracy on testing set  72.85000000000001
Height of tree  53
No of nodes  5127
```

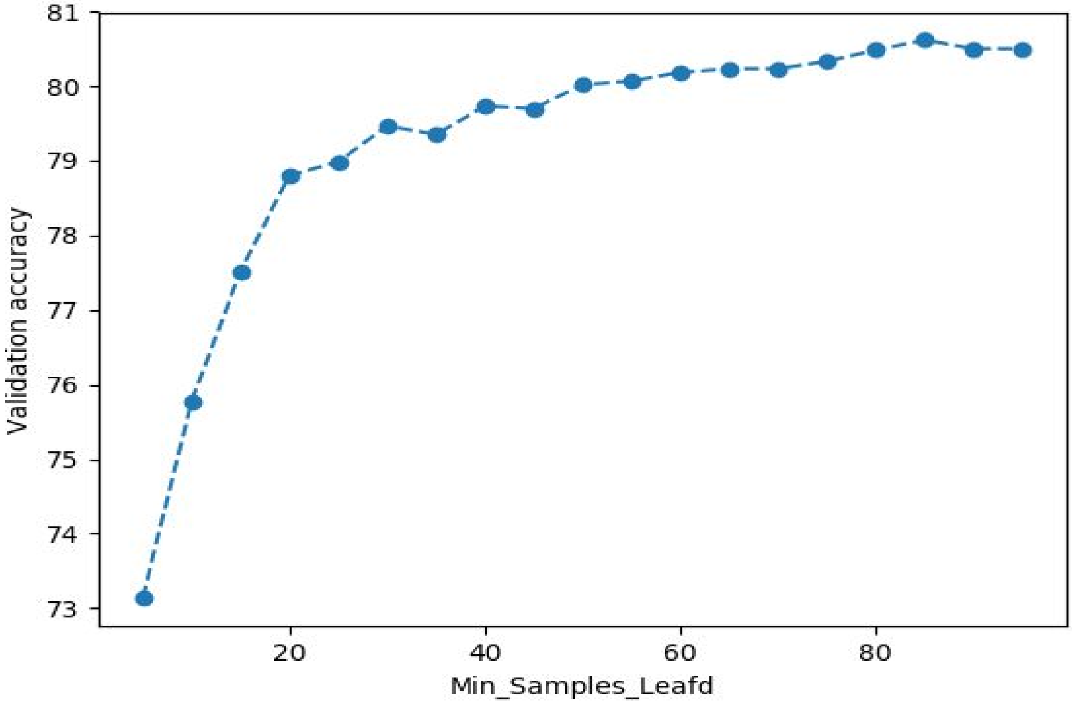
max_depth	**2**	**5**	**7**	**10**	**15**	**20**	**50**
Training	82.86	83.45	83.95	85.73	91.10	95.53	100
Testing	80.85	80.86	80.6	79.83	75.73	72.8	70.88
Validation	80.35	80.41	80.3	79.61	75.11	72.91	71.41



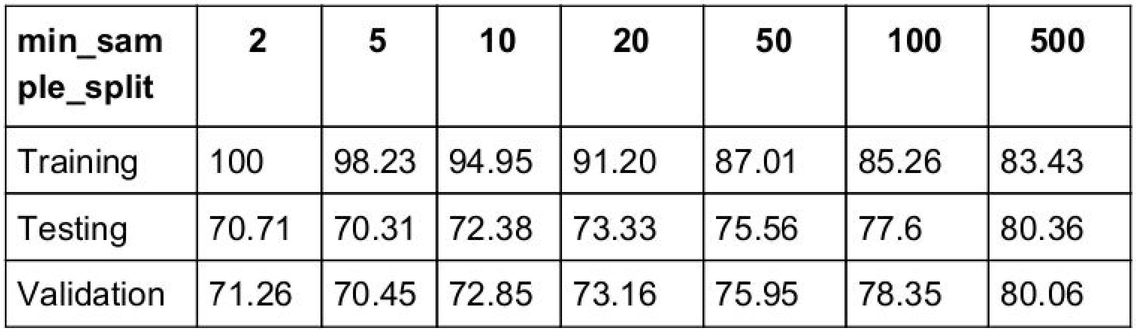
Validation accuracy drops with increase in max_depth.



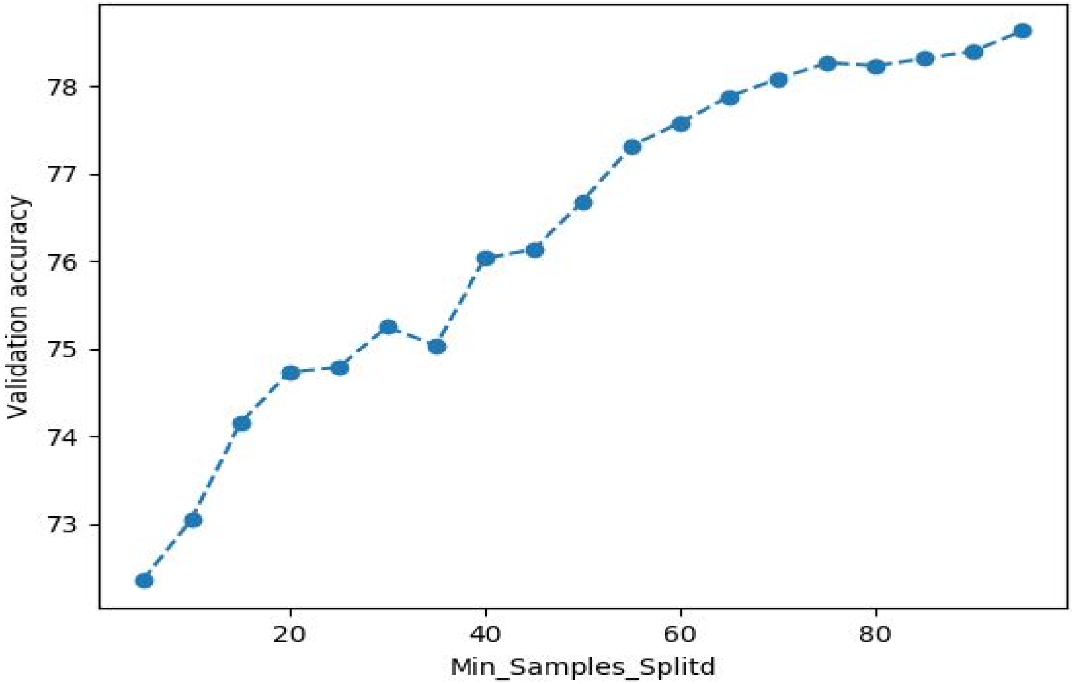
min_sample_leaf	**5**	**15**	**20**	**30**	**50**	**150**	**500**
Training	92.03	85.98	85.13	84.16	83.70	83.09	82.86
Testing	71.91	77.36	77.4	79.98	80.03	80.76	80.85
Validation	72.86	77.9	78.6	79.68	79.88	80.26	80.35



Validation accuracy increases with increase in min_samples_leaf.



min_sample_split	2	5	10	20	50	100	500
Training	100	98.23	94.95	91.20	87.01	85.26	83.43
Testing	70.71	70.31	72.38	73.33	75.56	77.6	80.36
Validation	71.26	70.45	72.85	73.16	75.95	78.35	80.06



Validation accuracy increases with increase in min_samples_split.

After running grid parameter search, parameters with best validation accuracy are:

Parameters : {'min_samples_split': 95, 'max_depth': 5, 'min_samples_leaf': 70}

Train set accuracy: 83.16666666666667

Validation set accuracy: 80.48333333333333

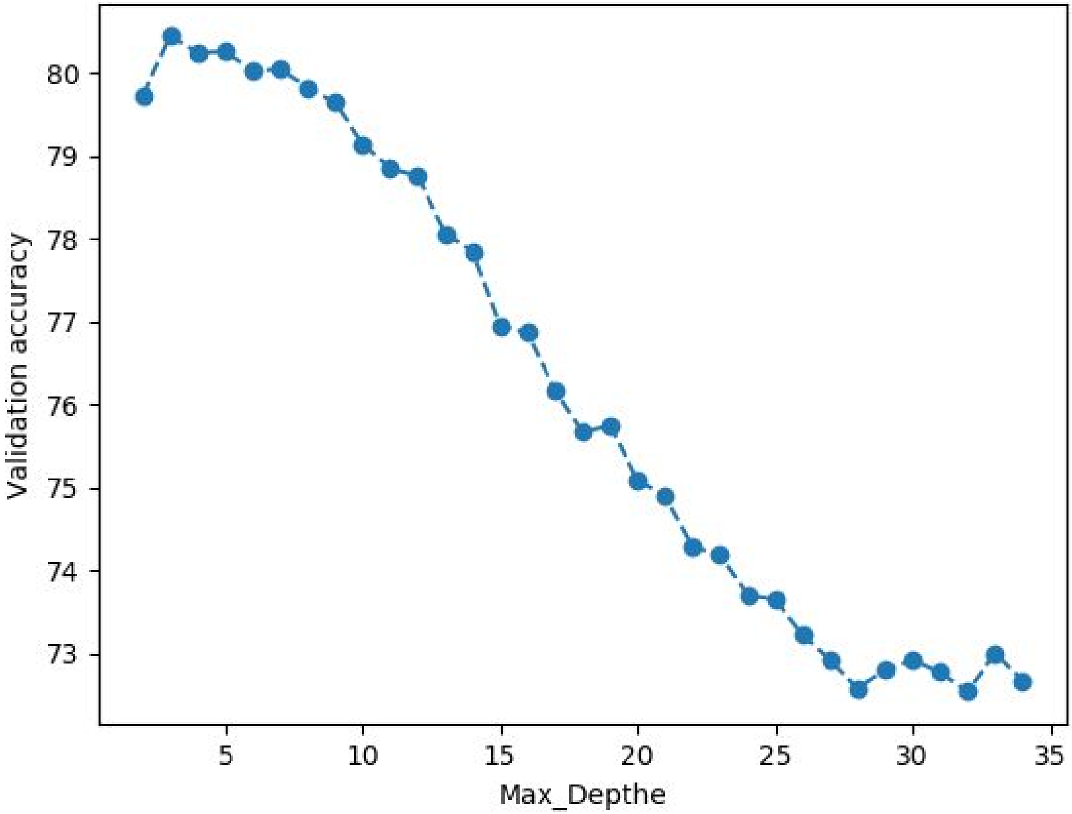
Test set accuracy: 80.78333333333333

Execution time 3050.06785297

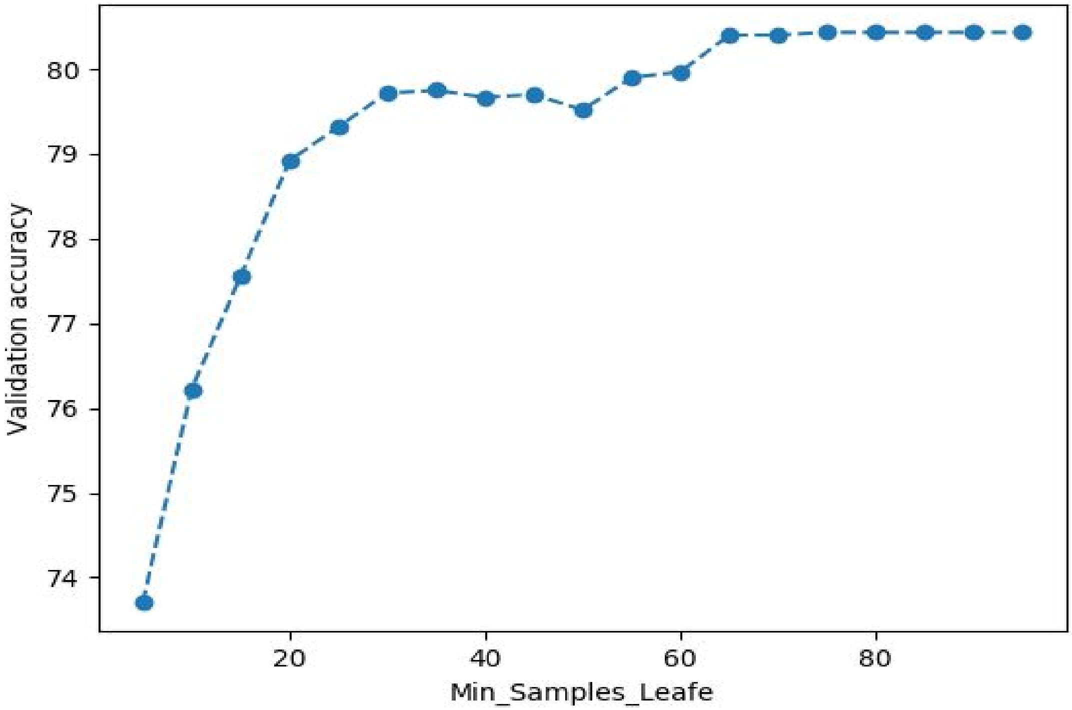
Observations:
Training accuracy decreases than part c and is almost same as of part b. However, testing and validation accuracy is increased than part c and is almost same as that in part b. Therefore, the results it produces are close to the results produced by post pruning.

e. Using one hot encoding

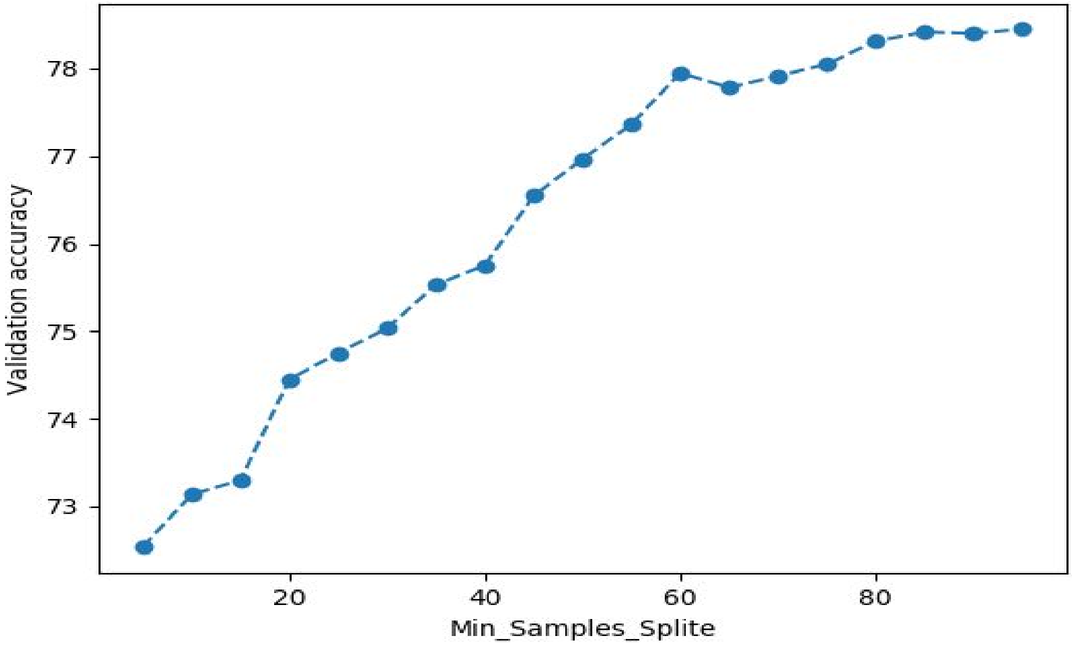
Accuracy on training set 99.96111111111111
Accuracy on validation set 72.38333333333333
Accuracy on testing set 72.45
Height of tree 40
No of nodes 5197



Validation accuracy decreases with increase in max_depth.



Validation accuracy increases with increase in min_samples_leaf



Validation accuracy increases with increase in min_samples_leaf.

Using grid parameter search:

Parameters : {'min_samples_split': 95, 'max_depth': 7, 'min_samples_leaf': 35}
Train set accuracy: 83.43888888888888
Validation set accuracy: 80.48333333333333
Test set accuracy: 80.61666666666667
Execution time 2115.11174202

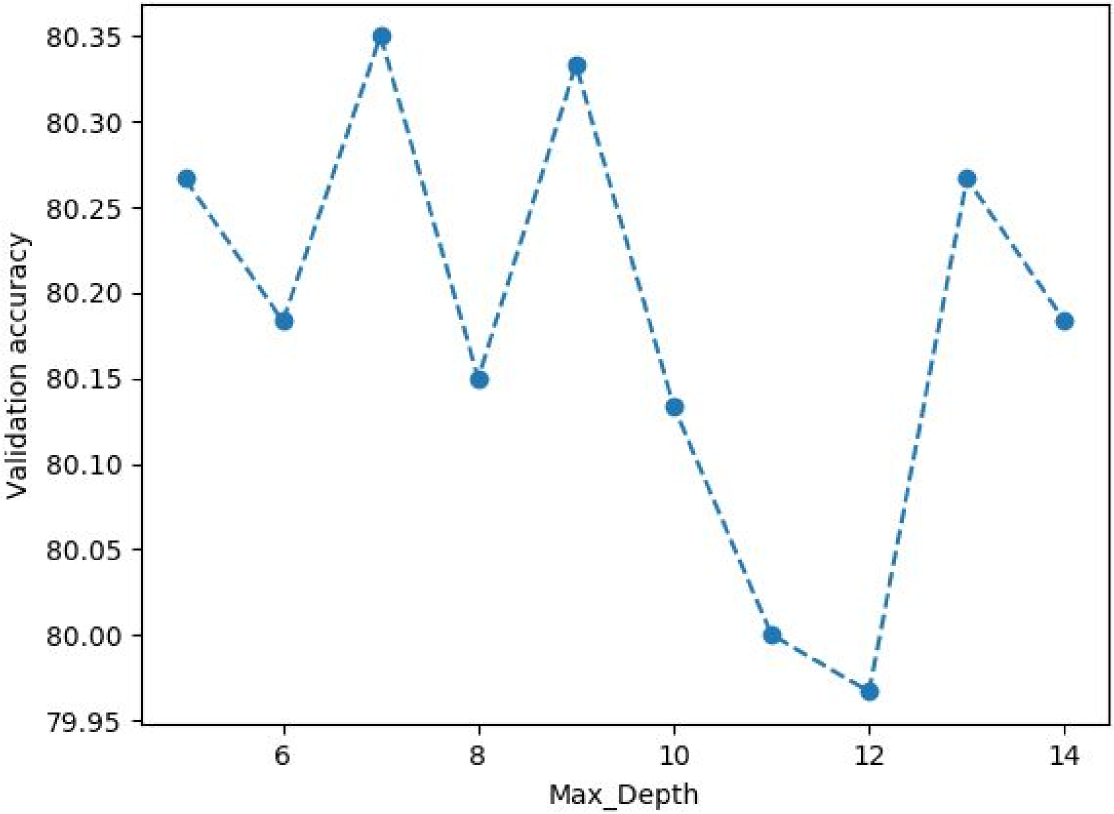
With previous settings i.e. max_depth = 5, min_samples_split = 95, min_samples_leaf = 70

Accuracy on training set 83.01111111111112
Accuracy on validation set 80.15
Accuracy on testing set 80.93333333333334
Height of tree 5
No of nodes 41

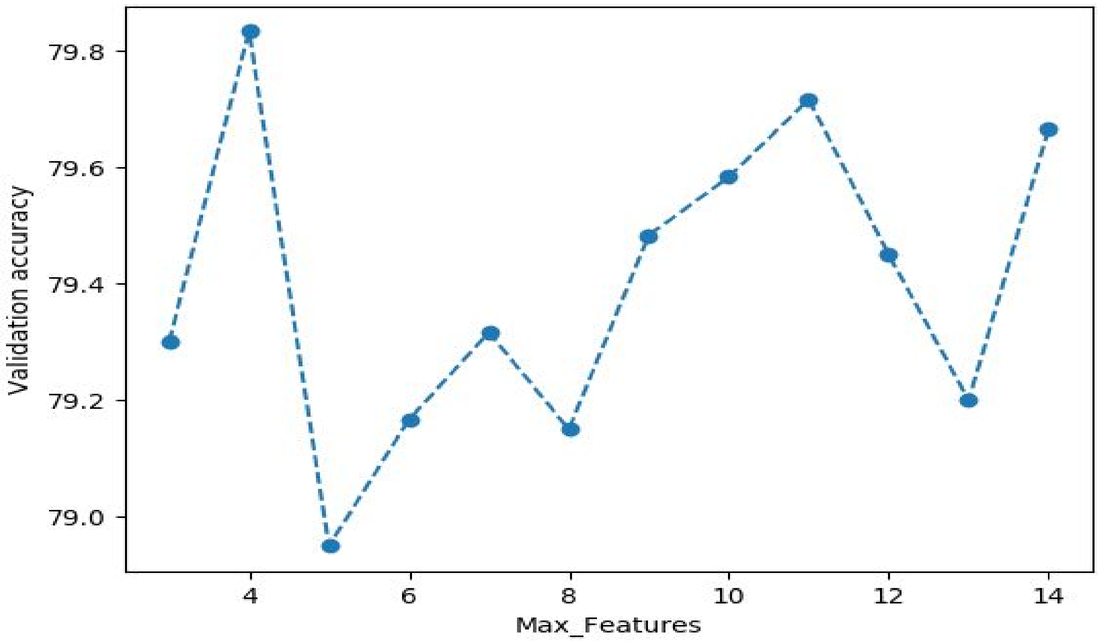
Observations:
Training accuracy decreases than part c and is almost same as of part b & d. However, testing and validation accuracy is increased than part c and is almost same as that in part b & d. Therefore, one hot encoding is not showing improvement over the accuracies.

f. Random Forest using sklearn

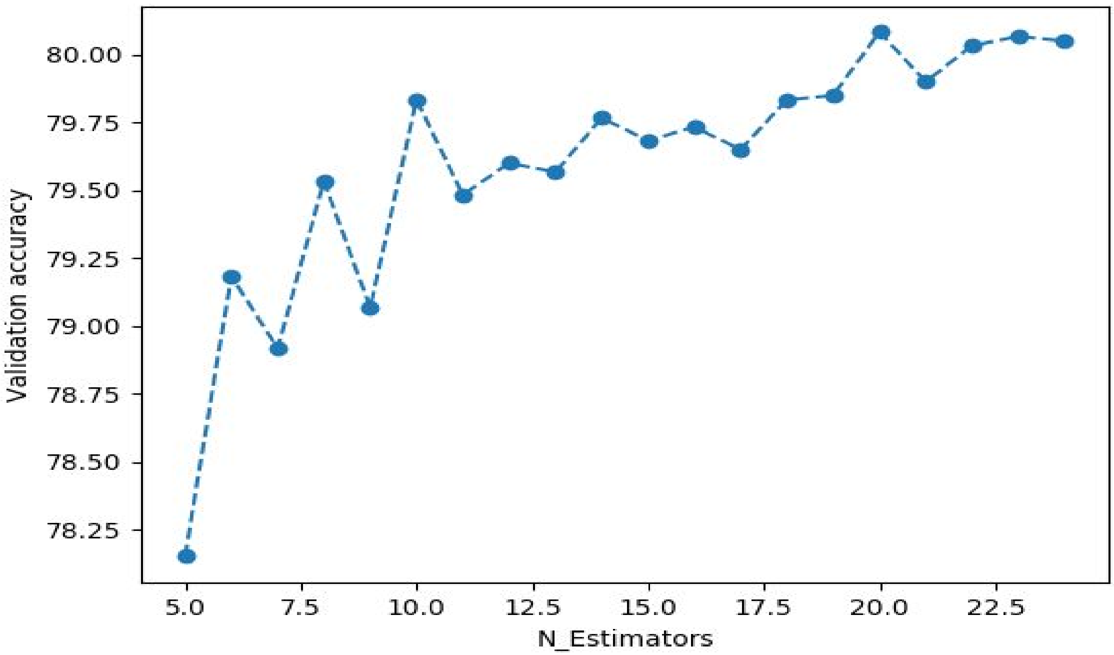
Using default parameters: 'max_features': auto, 'n_estimators': 10, 'bootstrap': True, 'max_depth': None
Accuracy on training set 98.26111111111111
Accuracy on validation set 79.83333333333333
Accuracy on testing set 79.63333333333334



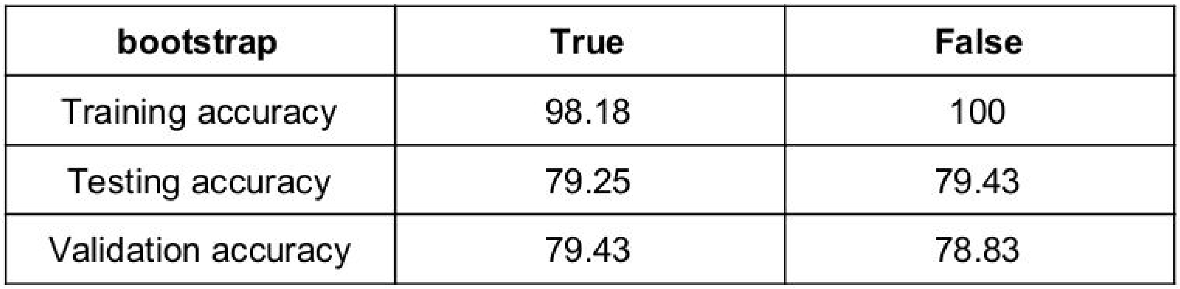
Validation accuracy decreases at some values and increases at some i.e. no fixed pattern is there.



On max_features also, validation accuracy increases at some values and decreases at some i.e. no fixed pattern is followed.



Validation accuracy increases with n_estimators i.e. number of trees in forest.



bootstrap	True	False
Training accuracy	98.18	100
Testing accuracy	79.25	79.43
Validation accuracy	79.43	78.83

Best parameters using grid search:
Parameters : {'max_features': 7, 'n_estimators': 21, 'bootstrap': False, 'max_depth': 11}
Train set accuracy: 87.59444444444443
Validation set accuracy: 80.78333333333333
Test set accuracy: 80.85
Execution time 8202.64859104

Observations :

Training accuracy decreases than part c and is almost same as of part b, d & e. However, testing and validation accuracy is increased than part c and is almost same as that in part b, d & e. Therefore, random forest generalizes quite well as done by post-pruning.

2. Neural Networks

a. The link for the one-hot encoding of train and test data is as follows:

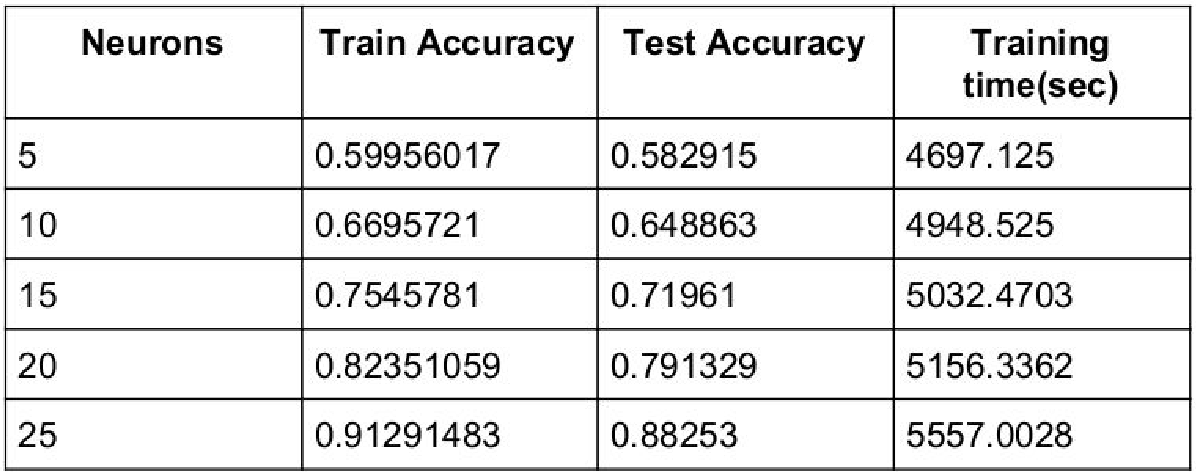
For one_hot_train.csv
https://drive.google.com/open?id=13zdqEg1qHc4VyA-DjyS67KoHkSmpQo3A

For one_hot_test.csv
https://drive.google.com/open?id=1-CL8AayM93nS4hTmB2nOu27vQffktJfV

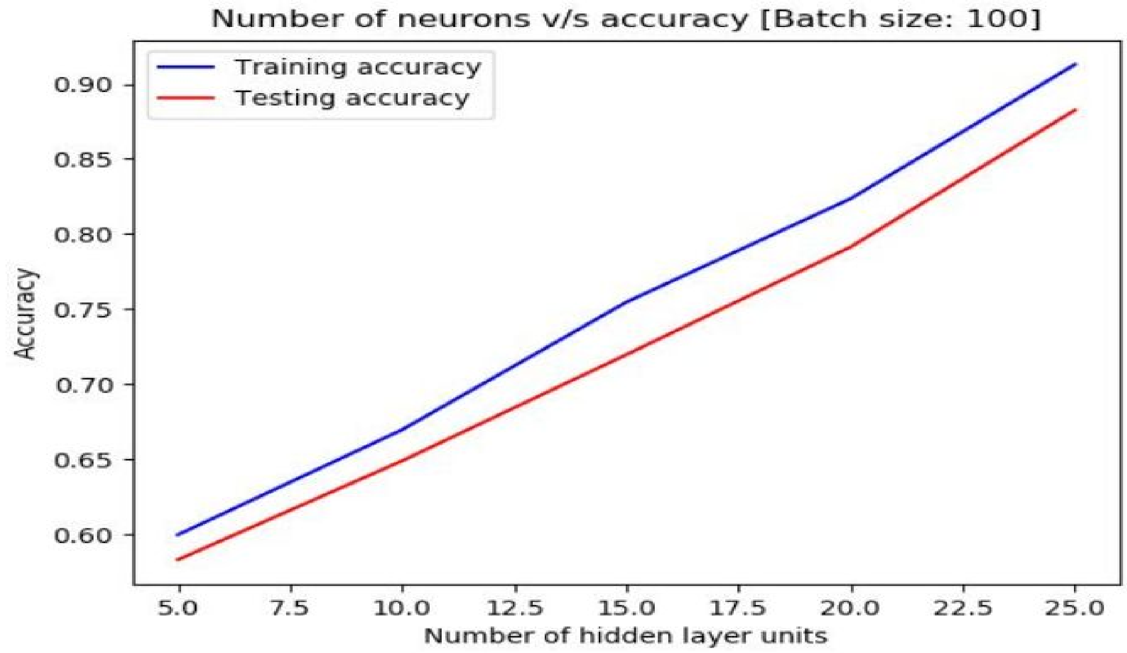
b. Neural Network implemented

c. Single hidden layer.
The neural network was tested with a single hidden layer and by varying number of units in that layer.
Number of neurons: [5, 10, 15, 20, 25]
Stopping criteria :

- Max epochs: 2500
- abs(Loss_{t+1} - Loss_{t}) <= 10 ** -9

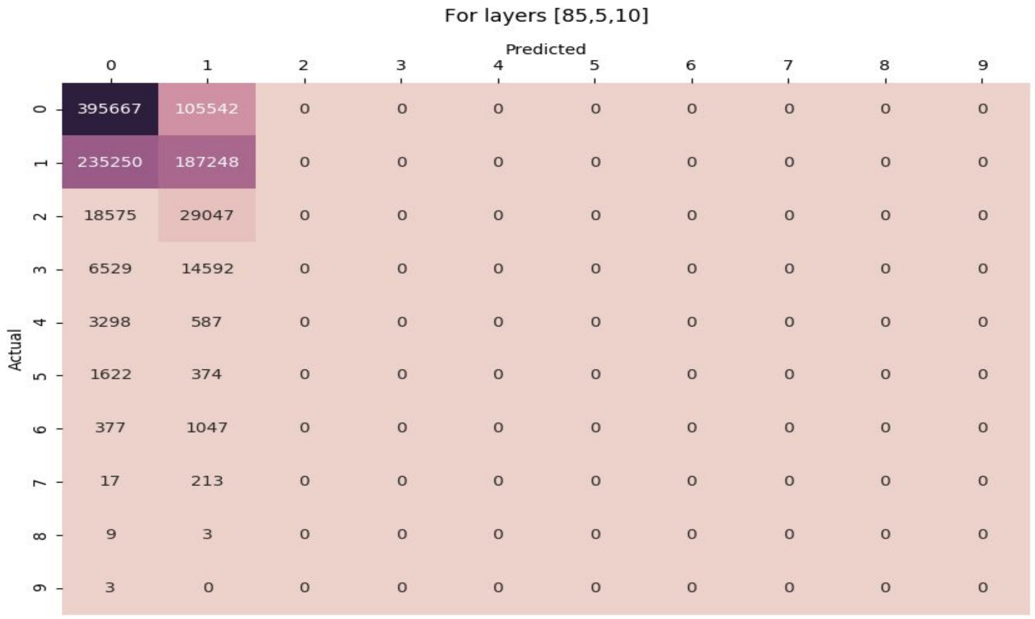


Neurons	Train Accuracy	Test Accuracy	Training time(sec)
5	0.59956017	0.582915	4697.125
10	0.6695721	0.648863	4948.525
15	0.7545781	0.71961	5032.4703
20	0.82351059	0.791329	5156.3362
25	0.91291483	0.88253	5557.0028



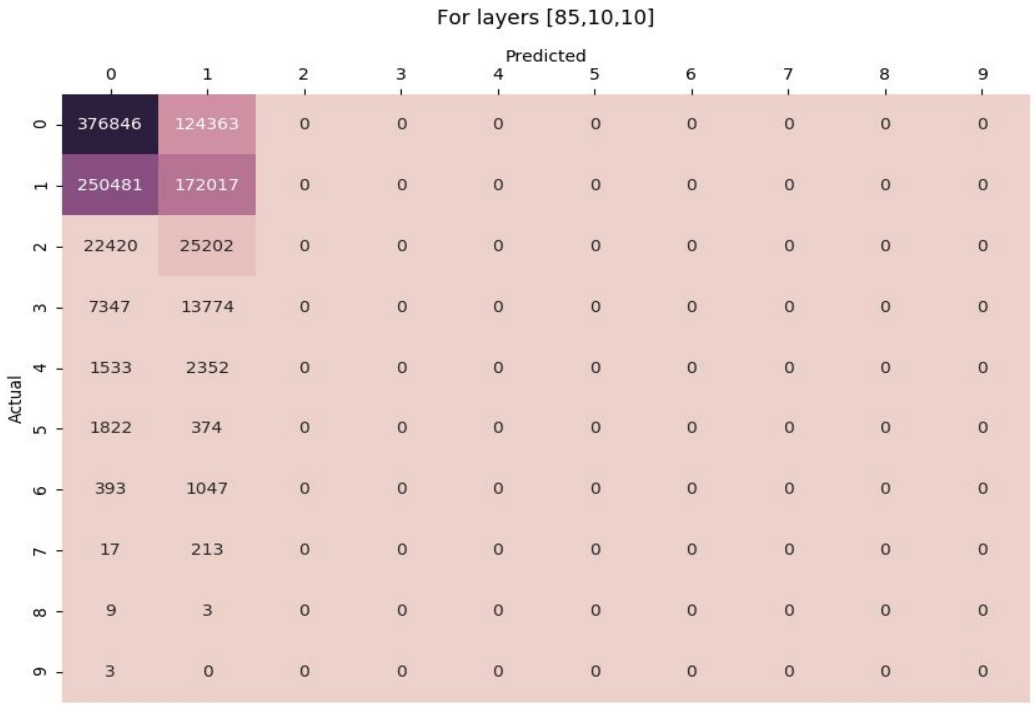
By increasing the number of units in the hidden layer accuracy has gone up. This may be because of the fact that with more neurons we get more parameters and our model learns better. But if we increase it by large number, the model may overfit.

For 5 neurons in single hidden layer


For layers [85,5,10]

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	395667	105542	0	0	0	0	0	0	0	0
1	235250	187248	0	0	0	0	0	0	0	0
2	18575	29047	0	0	0	0	0	0	0	0
3	6529	14592	0	0	0	0	0	0	0	0
4	3298	587	0	0	0	0	0	0	0	0
5	1622	374	0	0	0	0	0	0	0	0
6	377	1047	0	0	0	0	0	0	0	0
7	17	213	0	0	0	0	0	0	0	0
8	9	3	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0

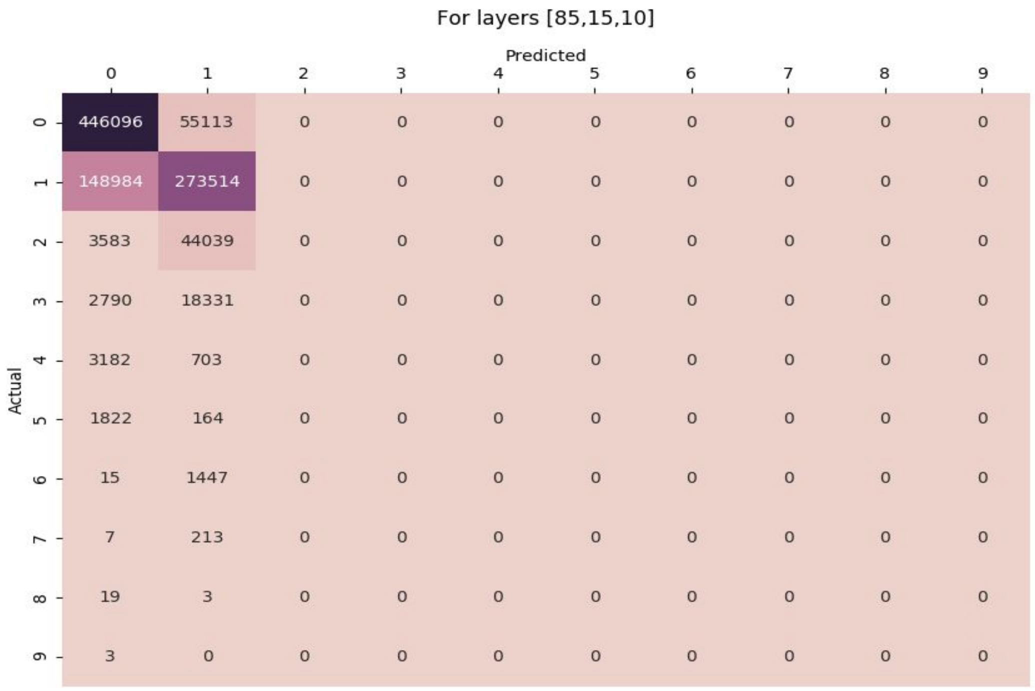
For 10 neurons in hidden layer



For layers [85,10,10]

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	376846	124363	0	0	0	0	0	0	0	0
1	250481	172017	0	0	0	0	0	0	0	0
2	22420	25202	0	0	0	0	0	0	0	0
3	7347	13774	0	0	0	0	0	0	0	0
4	1533	2352	0	0	0	0	0	0	0	0
5	1822	374	0	0	0	0	0	0	0	0
6	393	1047	0	0	0	0	0	0	0	0
7	17	213	0	0	0	0	0	0	0	0
8	9	3	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0

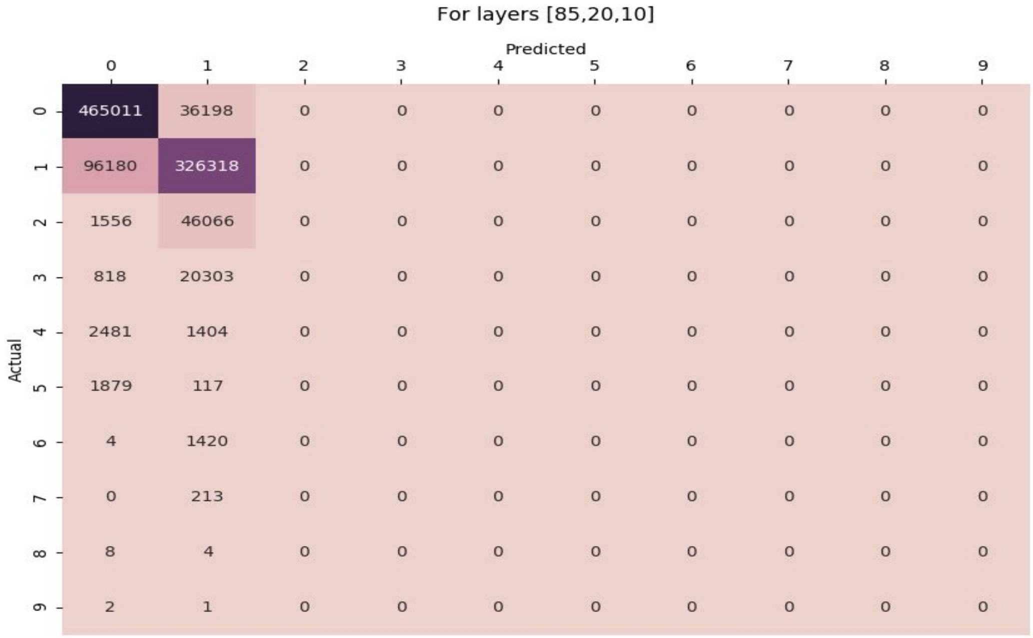
For 15 neurons in hidden layer



For layers [85,15,10]

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	446096	55113	0	0	0	0	0	0	0	0
1	148984	273514	0	0	0	0	0	0	0	0
2	3583	44039	0	0	0	0	0	0	0	0
3	2790	18331	0	0	0	0	0	0	0	0
4	3182	703	0	0	0	0	0	0	0	0
5	1822	164	0	0	0	0	0	0	0	0
6	15	1447	0	0	0	0	0	0	0	0
7	7	213	0	0	0	0	0	0	0	0
8	19	3	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0

For 20 neurons in hidden layer

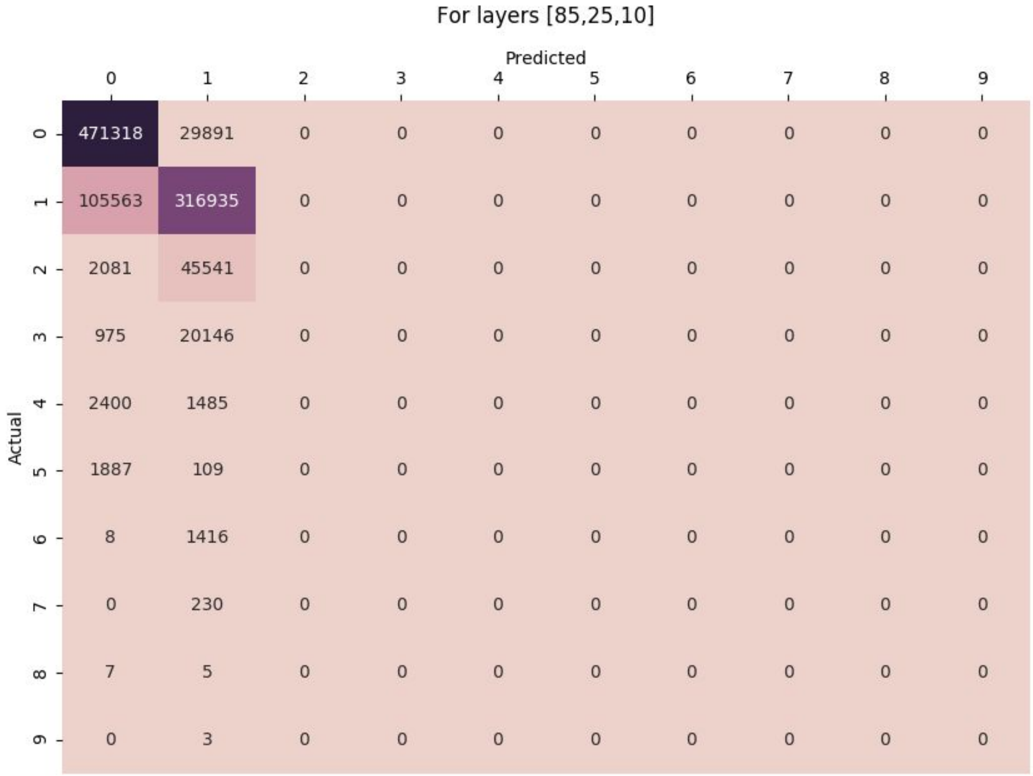


For layers [85,20,10]

Predicted

Actual	0	1	2	3	4	5	6	7	8	9
0	465011	36198	0	0	0	0	0	0	0	0
1	96180	326318	0	0	0	0	0	0	0	0
2	1556	46066	0	0	0	0	0	0	0	0
3	818	20303	0	0	0	0	0	0	0	0
4	2481	1404	0	0	0	0	0	0	0	0
5	1879	117	0	0	0	0	0	0	0	0
6	4	1420	0	0	0	0	0	0	0	0
7	0	213	0	0	0	0	0	0	0	0
8	8	4	0	0	0	0	0	0	0	0
9	2	1	0	0	0	0	0	0	0	0

For 25 neurons in hidden layer



For layers [85,25,10]

Predicted

Actual	0	1	2	3	4	5	6	7	8	9
0	471318	29891	0	0	0	0	0	0	0	0
1	105563	316935	0	0	0	0	0	0	0	0
2	2081	45541	0	0	0	0	0	0	0	0
3	975	20146	0	0	0	0	0	0	0	0
4	2400	1485	0	0	0	0	0	0	0	0
5	1887	109	0	0	0	0	0	0	0	0
6	8	1416	0	0	0	0	0	0	0	0
7	0	230	0	0	0	0	0	0	0	0
8	7	5	0	0	0	0	0	0	0	0
9	0	3	0	0	0	0	0	0	0	0

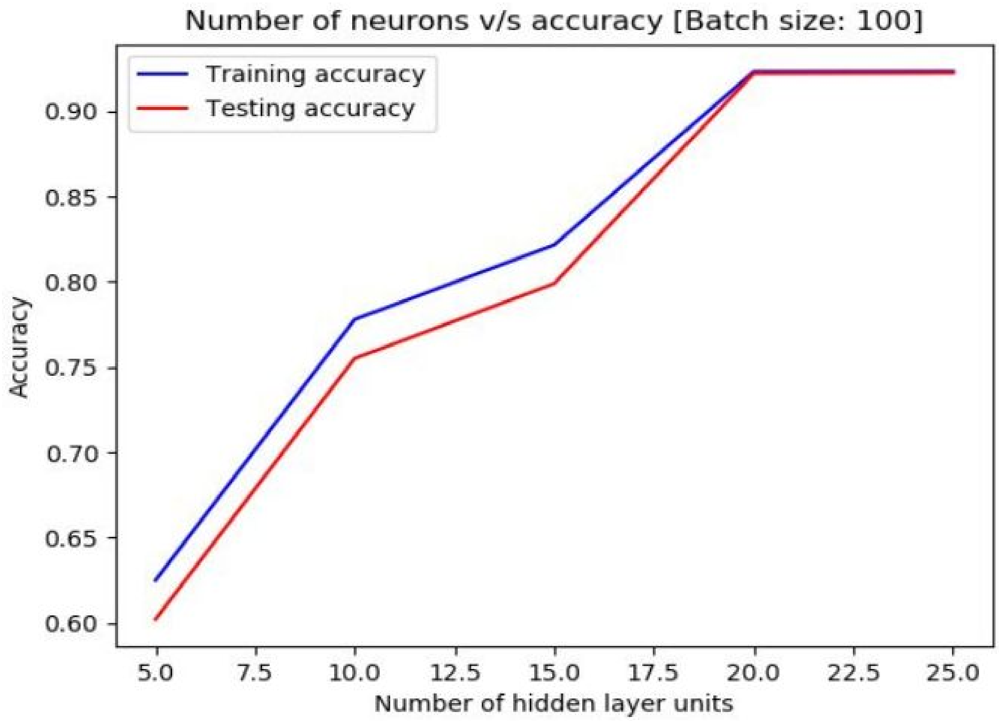
d. 2 hidden layers and same neurons in both of them

Stopping criteria : 2000 epochs

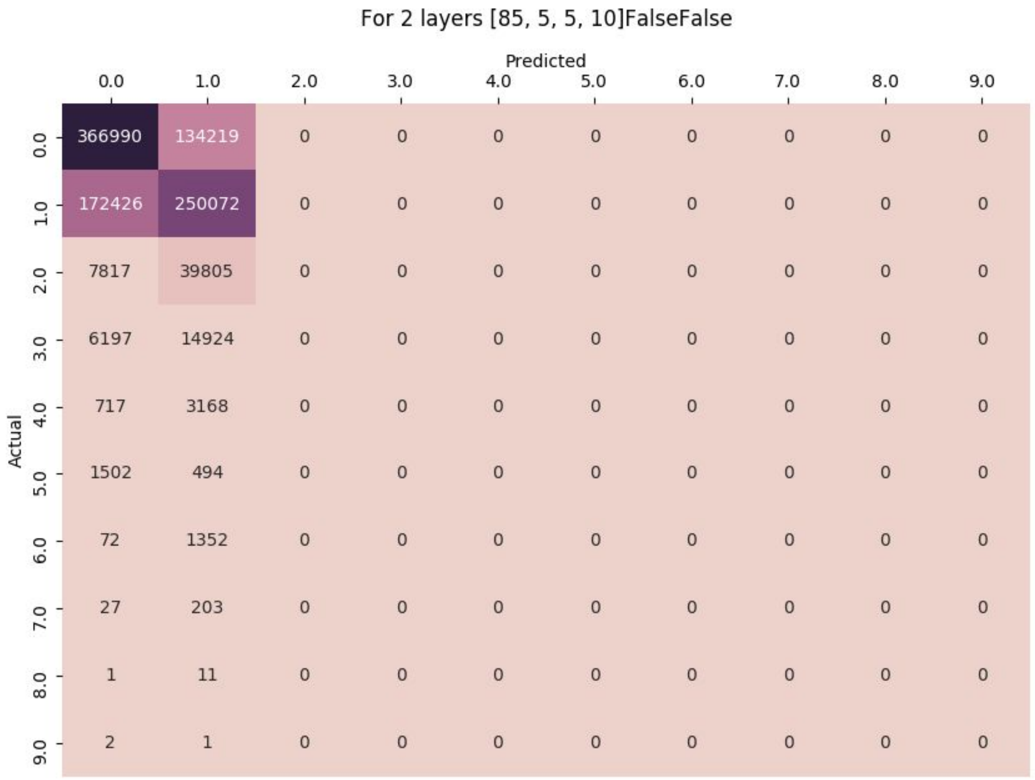
Error threshold: 10 ** -9

The accuracy improved with addition of one more hidden layer. However, there wasn’t much improvement after 20 neurons units.

Neurons	Train Accuracy	Test Accuracy	Training time(sec)
5	0.62514994	0.6023	4997.125
10	0.777968812	0.755127	5991.868798
15	0.8215920	0.79895	6031.3261
20	0.92323070	0.9221	6186.97881
25	0.92331075	0.922361	8264.47905



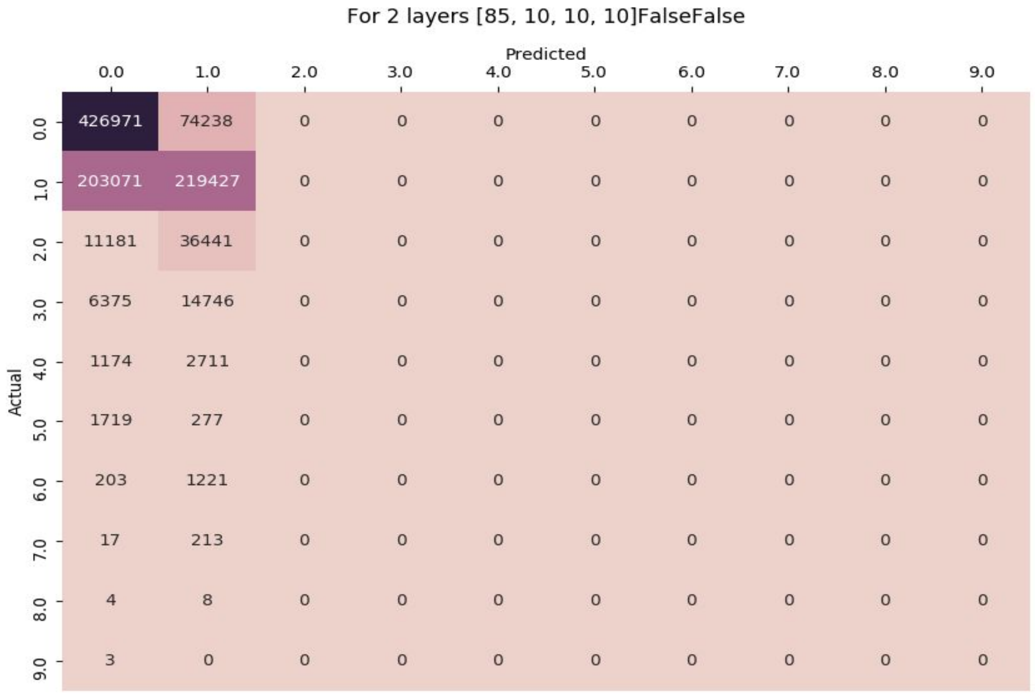
For 5 5 neurons in two hidden layers



For 2 layers [85, 5, 5, 10]FalseFalse

	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0
0.0	366990	134219	0	0	0	0	0	0	0	0
1.0	172426	250072	0	0	0	0	0	0	0	0
2.0	7817	39805	0	0	0	0	0	0	0	0
3.0	6197	14924	0	0	0	0	0	0	0	0
4.0	717	3168	0	0	0	0	0	0	0	0
5.0	1502	494	0	0	0	0	0	0	0	0
6.0	72	1352	0	0	0	0	0	0	0	0
7.0	27	203	0	0	0	0	0	0	0	0
8.0	1	11	0	0	0	0	0	0	0	0
9.0	2	1	0	0	0	0	0	0	0	0

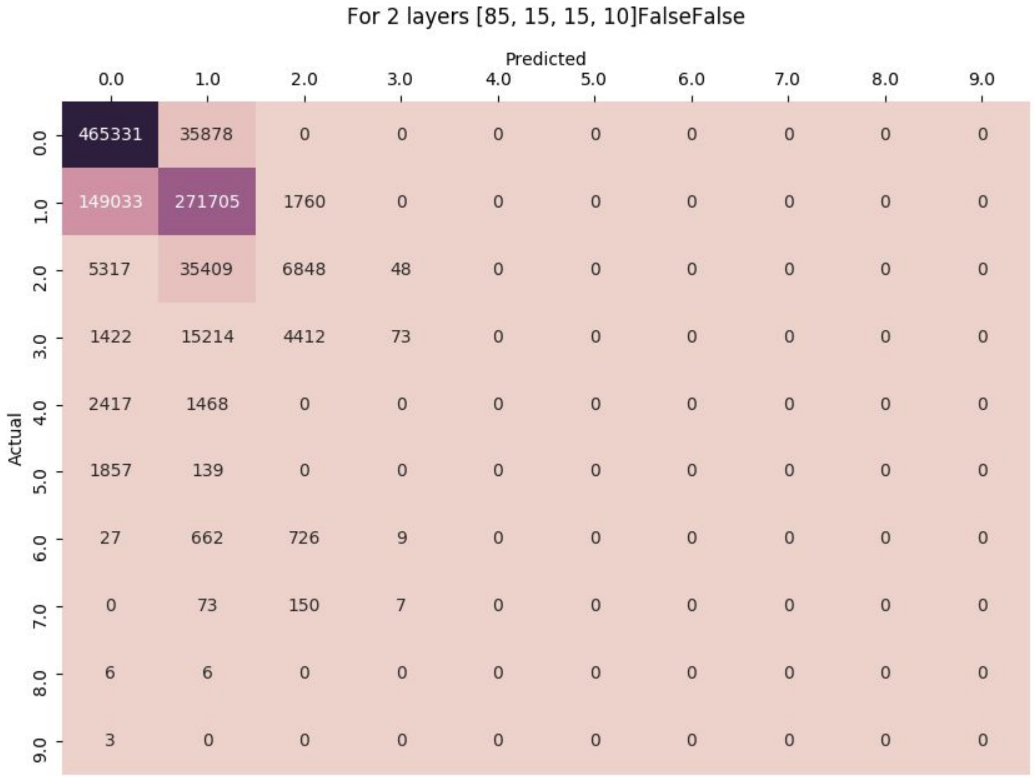
For 10 10 neurons in two hidden layers



For 2 layers [85, 10, 10, 10]FalseFalse

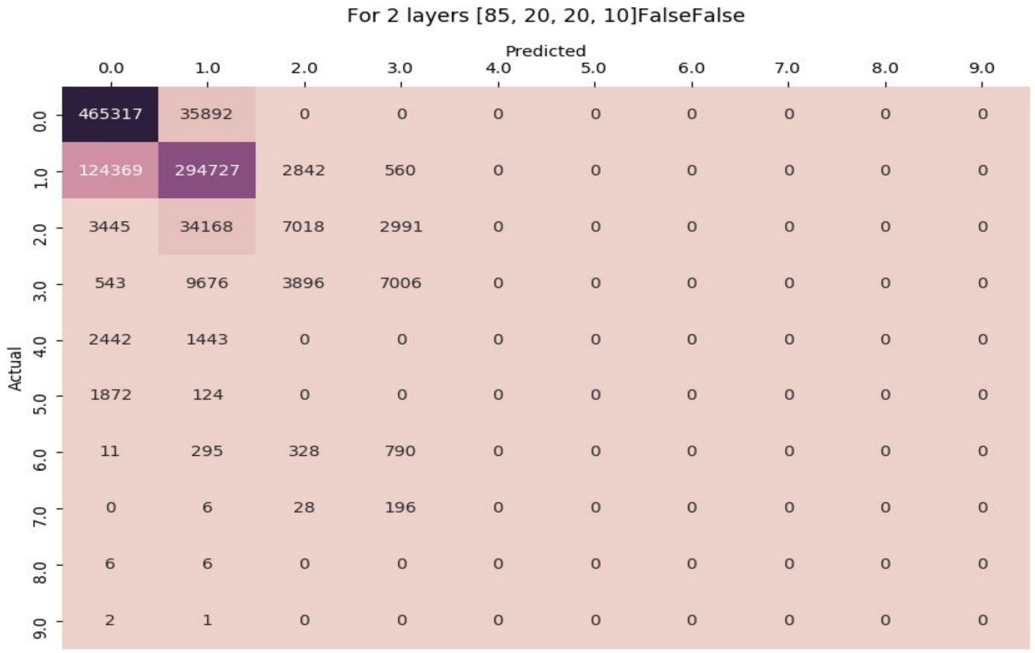
	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0
0.0	426971	74238	0	0	0	0	0	0	0	0
1.0	203071	219427	0	0	0	0	0	0	0	0
2.0	11181	36441	0	0	0	0	0	0	0	0
3.0	6375	14746	0	0	0	0	0	0	0	0
4.0	1174	2711	0	0	0	0	0	0	0	0
5.0	1719	277	0	0	0	0	0	0	0	0
6.0	203	1221	0	0	0	0	0	0	0	0
7.0	17	213	0	0	0	0	0	0	0	0
8.0	4	8	0	0	0	0	0	0	0	0
9.0	3	0	0	0	0	0	0	0	0	0

For 15 15 neurons in two hidden layers



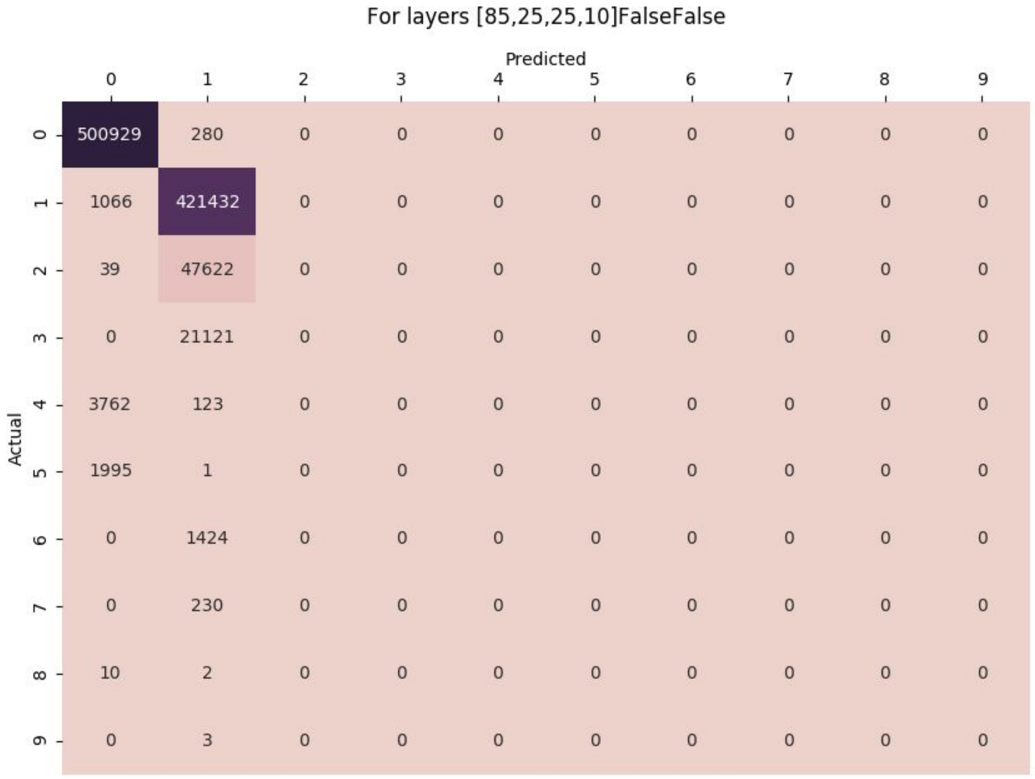
	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0
0.0	465331	35878	0	0	0	0	0	0	0	0
1.0	149033	271705	1760	0	0	0	0	0	0	0
2.0	5317	35409	6848	48	0	0	0	0	0	0
3.0	1422	15214	4412	73	0	0	0	0	0	0
4.0	2417	1468	0	0	0	0	0	0	0	0
5.0	1857	139	0	0	0	0	0	0	0	0
6.0	27	662	726	9	0	0	0	0	0	0
7.0	0	73	150	7	0	0	0	0	0	0
8.0	6	6	0	0	0	0	0	0	0	0
9.0	3	0	0	0	0	0	0	0	0	0

For 20 20 neurons in two hidden layers



	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0
0.0	465317	35892	0	0	0	0	0	0	0	0
1.0	124369	294727	2842	560	0	0	0	0	0	0
2.0	3445	34168	7018	2991	0	0	0	0	0	0
3.0	543	9676	3896	7006	0	0	0	0	0	0
4.0	2442	1443	0	0	0	0	0	0	0	0
5.0	1872	124	0	0	0	0	0	0	0	0
6.0	11	295	328	790	0	0	0	0	0	0
7.0	0	6	28	196	0	0	0	0	0	0
8.0	6	6	0	0	0	0	0	0	0	0
9.0	2	1	0	0	0	0	0	0	0	0

For 25 25 neurons in two hidden layers


For layers [85,25,25,10]FalseFalse

	0	1	2	3	4	5	6	7	8	9
0	500929	280	0	0	0	0	0	0	0	0
1	1066	421432	0	0	0	0	0	0	0	0
2	39	47622	0	0	0	0	0	0	0	0
3	0	21121	0	0	0	0	0	0	0	0
4	3762	123	0	0	0	0	0	0	0	0
5	1995	1	0	0	0	0	0	0	0	0
6	0	1424	0	0	0	0	0	0	0	0
7	0	230	0	0	0	0	0	0	0	0
8	10	2	0	0	0	0	0	0	0	0
9	0	3	0	0	0	0	0	0	0	0

e. Adaptive learning rate

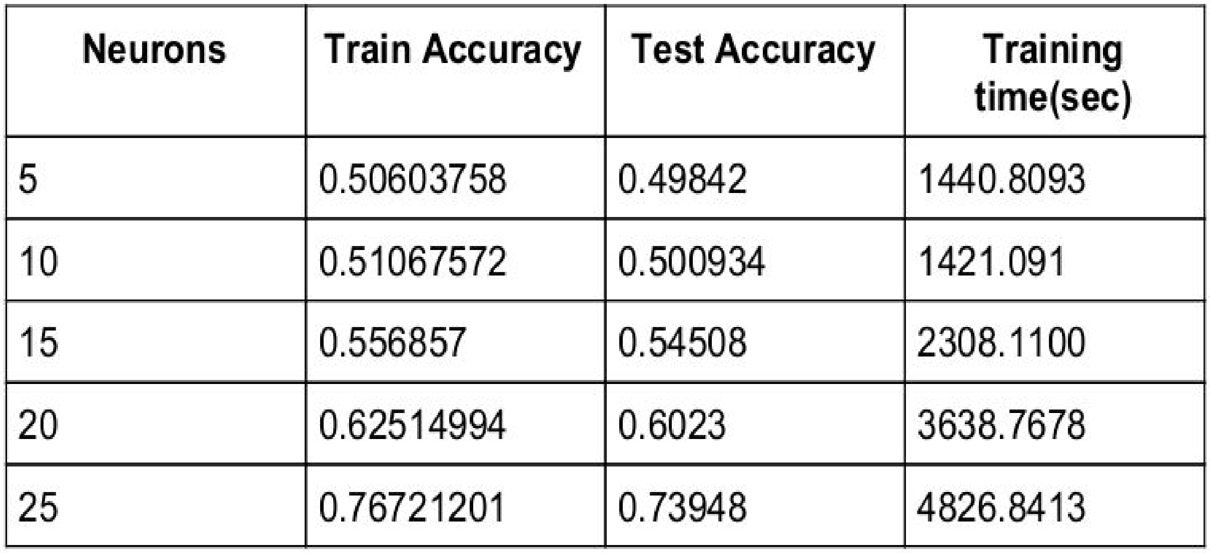
There wasn’t any improvement in the accuracy when adaptive learning with tol = 10 ** -4 was used. Some accuracies remained same as earlier while some became even worse.

Stopping condition:

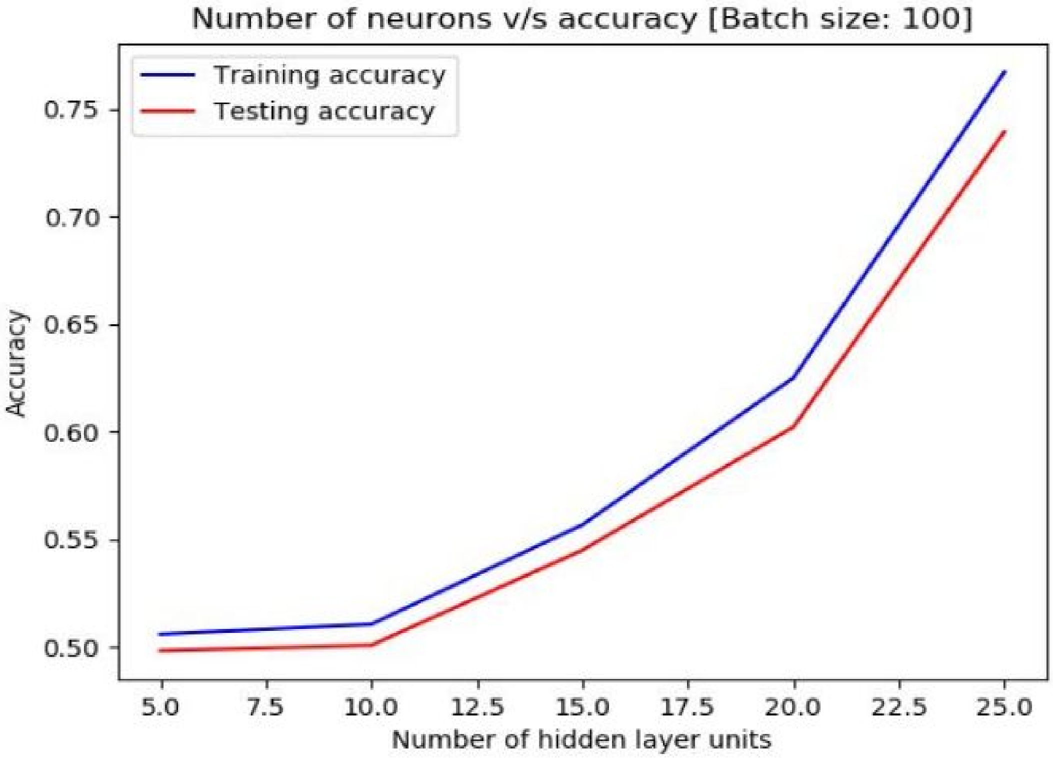
Error threshold: 10 ** -9

Epochs: 1000

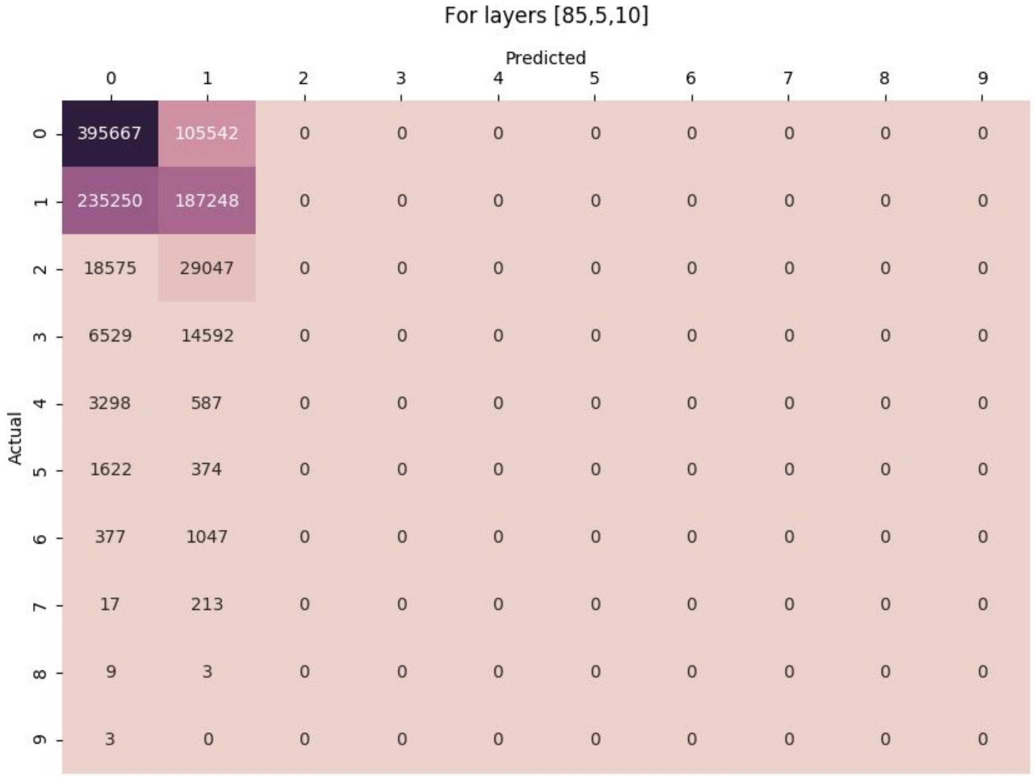
i. Single hidden layers



Neurons	Train Accuracy	Test Accuracy	Training time(sec)
5	0.50603758	0.49842	1440.8093
10	0.51067572	0.500934	1421.091
15	0.556857	0.54508	2308.1100
20	0.62514994	0.6023	3638.7678
25	0.76721201	0.73948	4826.8413

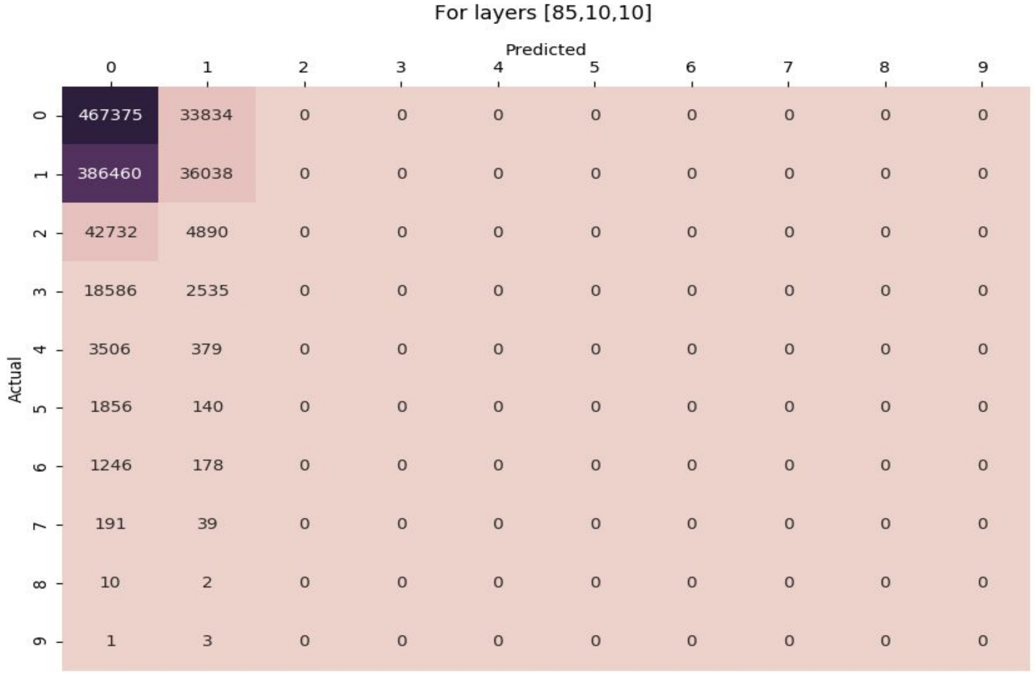


For 5 neurons in single layer



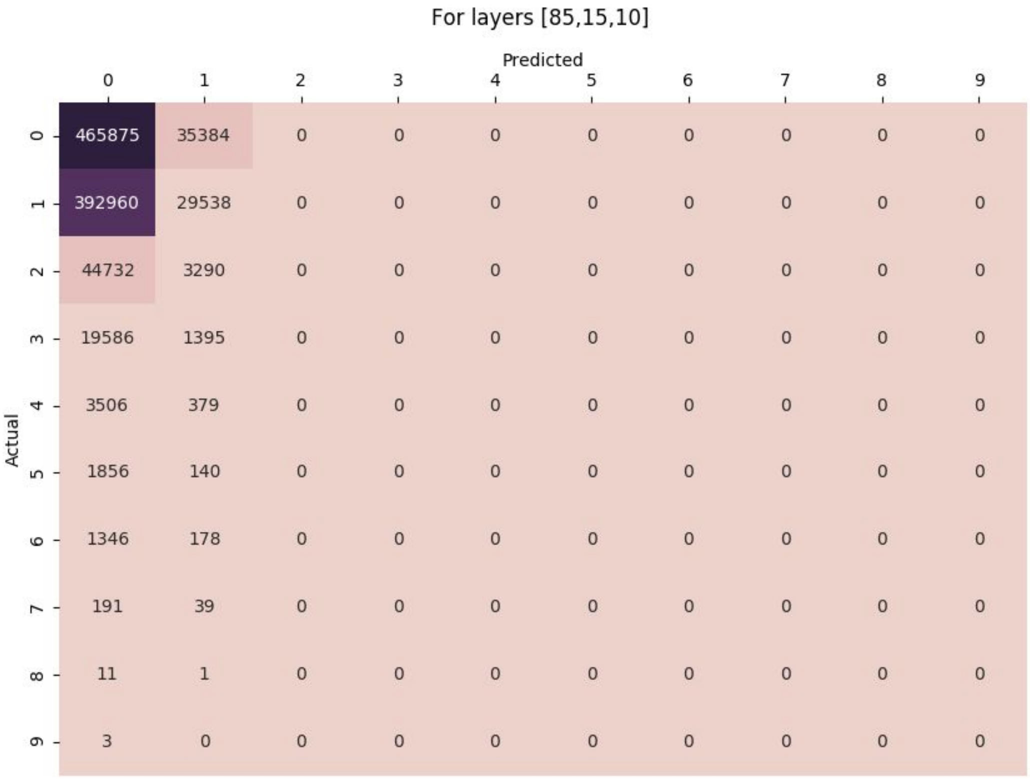
	0	1	2	3	4	5	6	7	8	9
0	395667	105542	0	0	0	0	0	0	0	0
1	235250	187248	0	0	0	0	0	0	0	0
2	18575	29047	0	0	0	0	0	0	0	0
3	6529	14592	0	0	0	0	0	0	0	0
4	3298	587	0	0	0	0	0	0	0	0
5	1622	374	0	0	0	0	0	0	0	0
6	377	1047	0	0	0	0	0	0	0	0
7	17	213	0	0	0	0	0	0	0	0
8	9	3	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0

For 10 neurons in single layer

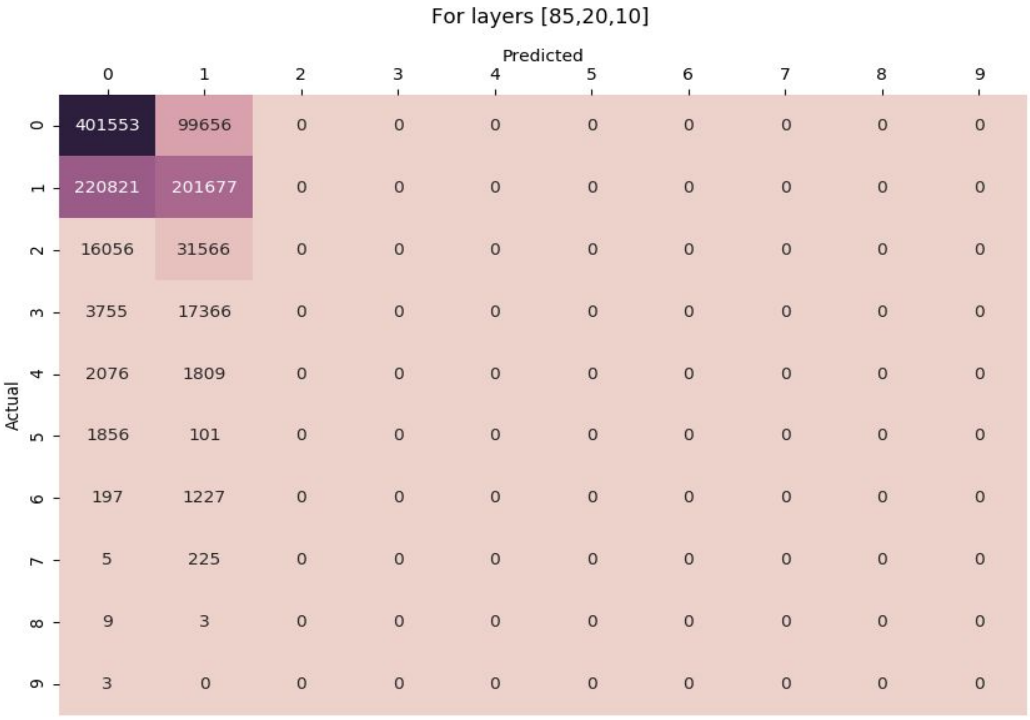


	0	1	2	3	4	5	6	7	8	9
0	467375	33834	0	0	0	0	0	0	0	0
1	386460	36038	0	0	0	0	0	0	0	0
2	42732	4890	0	0	0	0	0	0	0	0
3	18586	2535	0	0	0	0	0	0	0	0
4	3506	379	0	0	0	0	0	0	0	0
5	1856	140	0	0	0	0	0	0	0	0
6	1246	178	0	0	0	0	0	0	0	0
7	191	39	0	0	0	0	0	0	0	0
8	10	2	0	0	0	0	0	0	0	0
9	1	3	0	0	0	0	0	0	0	0

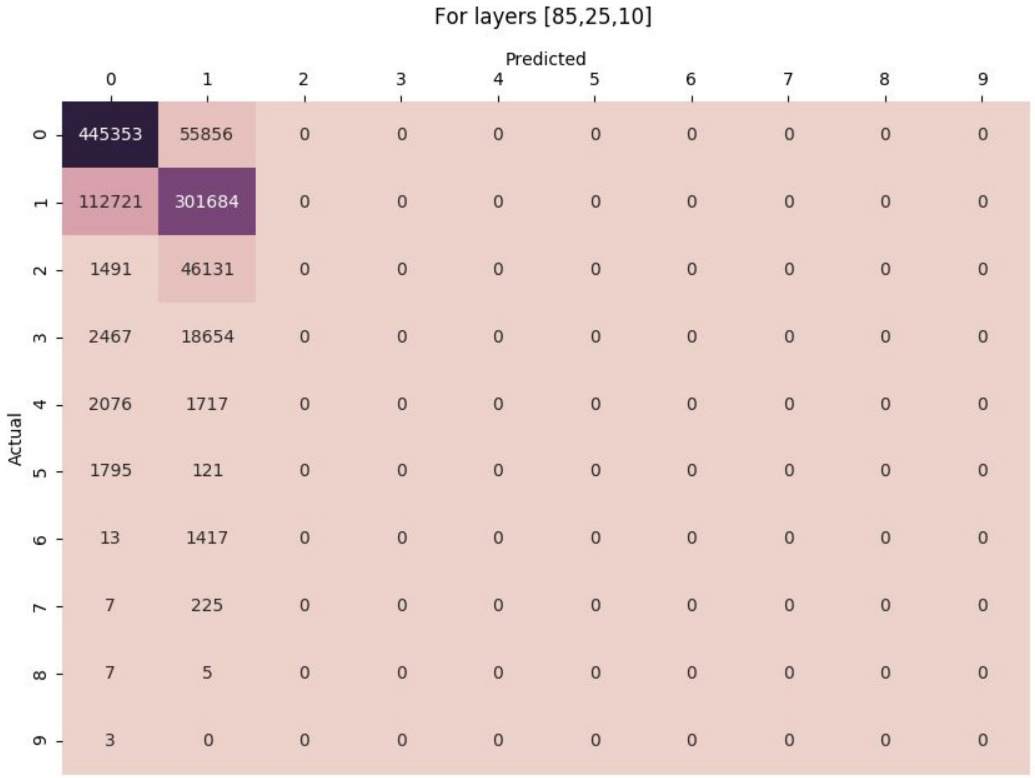
For 15 neurons in single layer



For 20 neurons in single layer



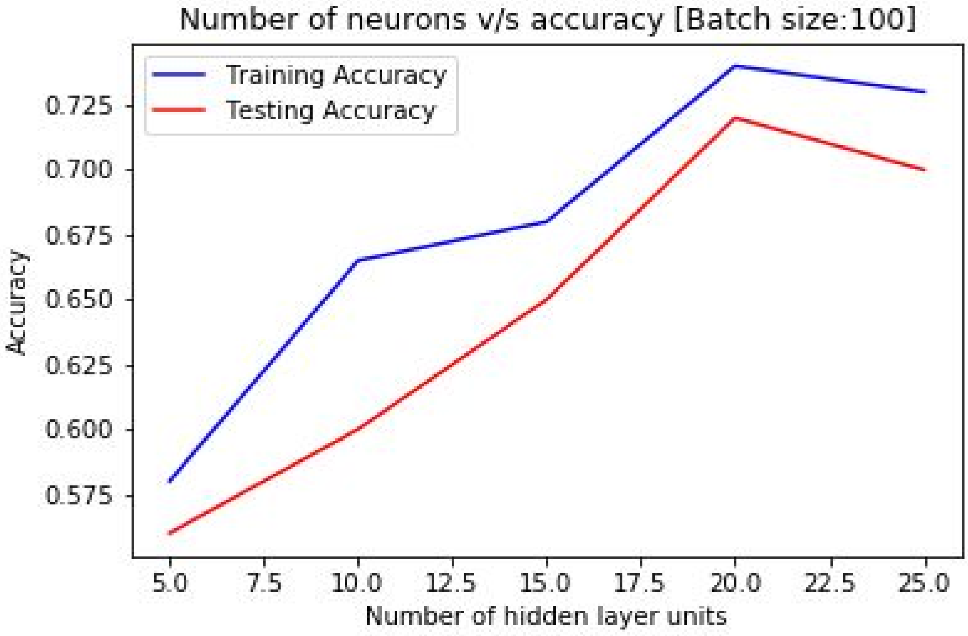
For 25 neurons in single layer



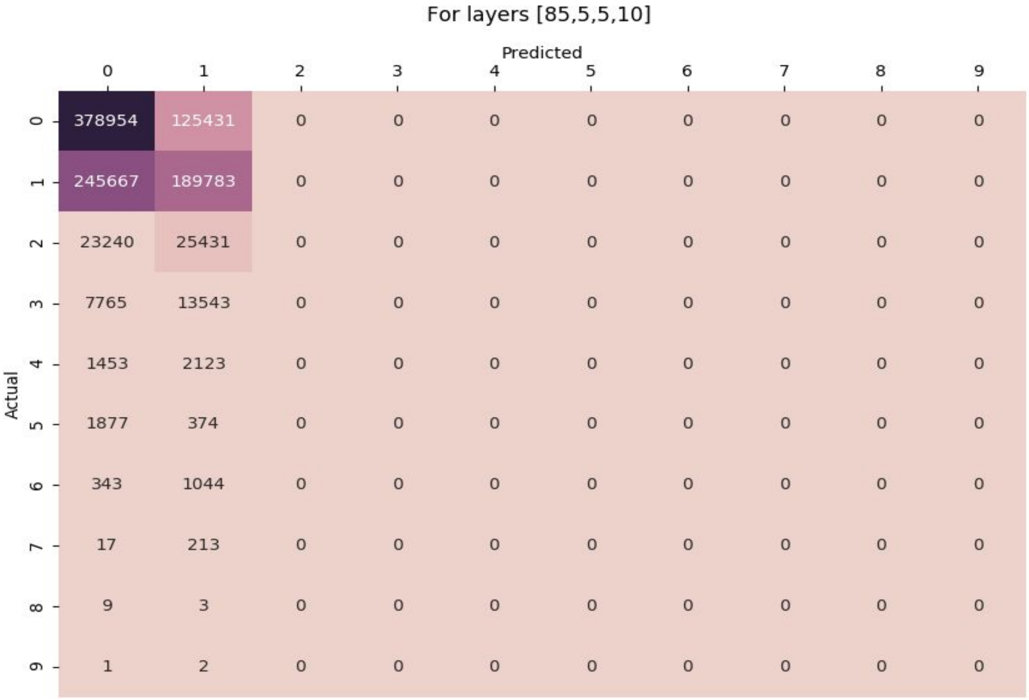
For layers [85,25,10]

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	445353	55856	0	0	0	0	0	0	0	0
1	112721	301684	0	0	0	0	0	0	0	0
2	1491	46131	0	0	0	0	0	0	0	0
3	2467	18654	0	0	0	0	0	0	0	0
4	2076	1717	0	0	0	0	0	0	0	0
5	1795	121	0	0	0	0	0	0	0	0
6	13	1417	0	0	0	0	0	0	0	0
7	7	225	0	0	0	0	0	0	0	0
8	7	5	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0

ii. Two hidden layers



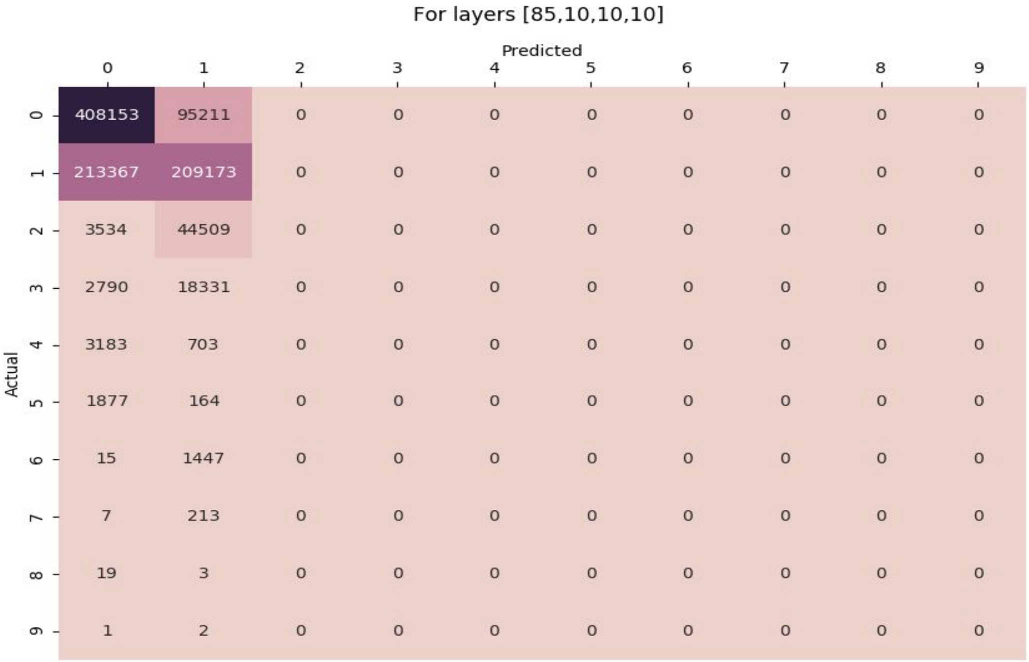
For 5 5 neurons in two hidden layers



For layers [85,5,5,10]

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	378954	125431	0	0	0	0	0	0	0	0
1	245667	189783	0	0	0	0	0	0	0	0
2	23240	25431	0	0	0	0	0	0	0	0
3	7765	13543	0	0	0	0	0	0	0	0
4	1453	2123	0	0	0	0	0	0	0	0
5	1877	374	0	0	0	0	0	0	0	0
6	343	1044	0	0	0	0	0	0	0	0
7	17	213	0	0	0	0	0	0	0	0
8	9	3	0	0	0	0	0	0	0	0
9	1	2	0	0	0	0	0	0	0	0

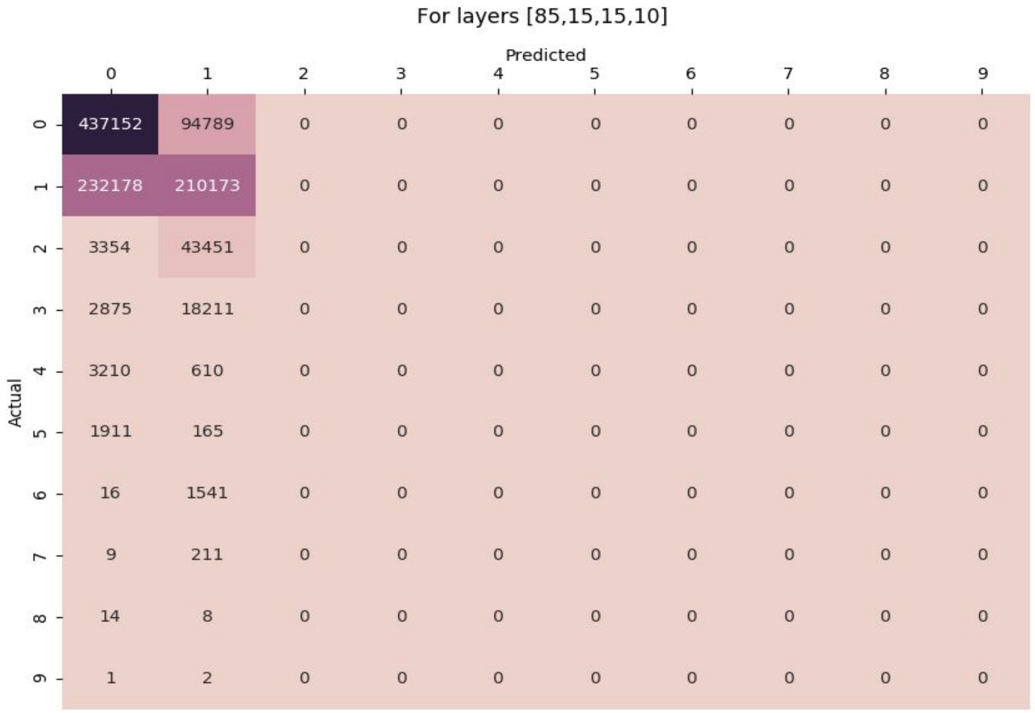
For 10 10 neurons in two hidden layer



For layers [85,10,10,10]

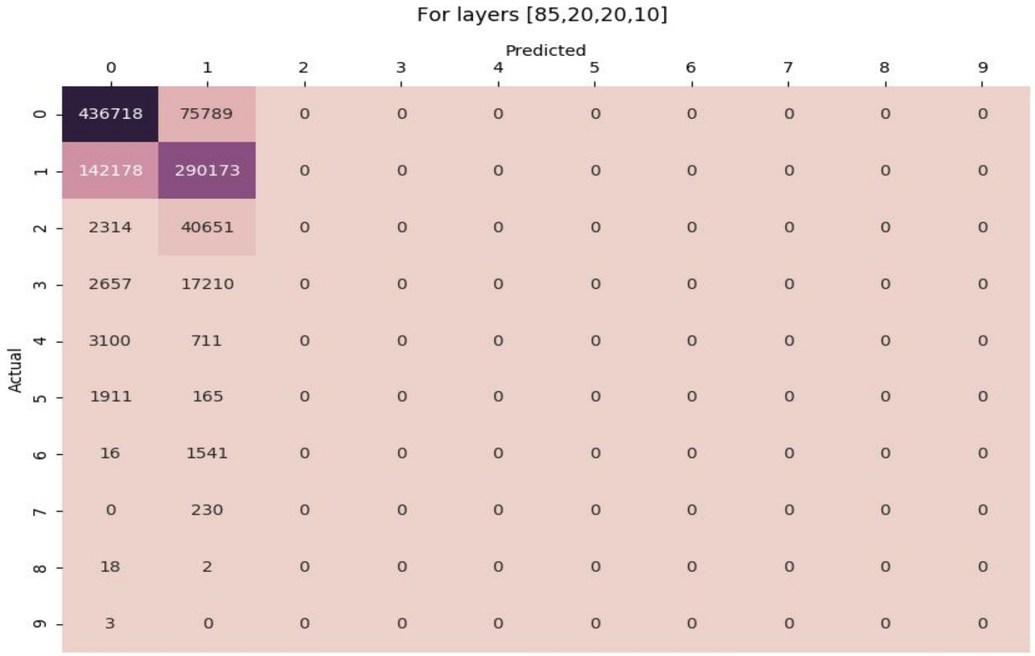
Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	408153	95211	0	0	0	0	0	0	0	0
1	213367	209173	0	0	0	0	0	0	0	0
2	3534	44509	0	0	0	0	0	0	0	0
3	2790	18331	0	0	0	0	0	0	0	0
4	3183	703	0	0	0	0	0	0	0	0
5	1877	164	0	0	0	0	0	0	0	0
6	15	1447	0	0	0	0	0	0	0	0
7	7	213	0	0	0	0	0	0	0	0
8	19	3	0	0	0	0	0	0	0	0
9	1	2	0	0	0	0	0	0	0	0

For 15 15 neurons in two hidden layers


For layers [85,15,15,10]

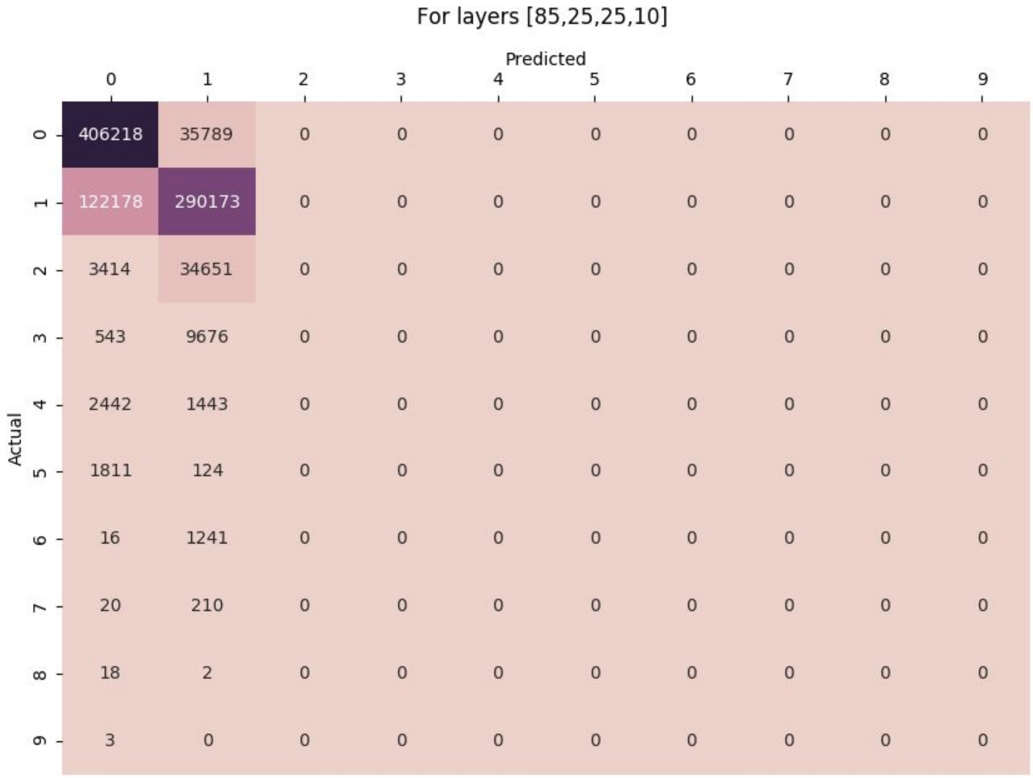
Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	437152	94789	0	0	0	0	0	0	0	0
1	232178	210173	0	0	0	0	0	0	0	0
2	3354	43451	0	0	0	0	0	0	0	0
3	2875	18211	0	0	0	0	0	0	0	0
4	3210	610	0	0	0	0	0	0	0	0
5	1911	165	0	0	0	0	0	0	0	0
6	16	1541	0	0	0	0	0	0	0	0
7	9	211	0	0	0	0	0	0	0	0
8	14	8	0	0	0	0	0	0	0	0
9	1	2	0	0	0	0	0	0	0	0

For 20 20 neurons in two hidden layers


For layers [85,20,20,10]

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	436718	75789	0	0	0	0	0	0	0	0
1	142178	290173	0	0	0	0	0	0	0	0
2	2314	40651	0	0	0	0	0	0	0	0
3	2657	17210	0	0	0	0	0	0	0	0
4	3100	711	0	0	0	0	0	0	0	0
5	1911	165	0	0	0	0	0	0	0	0
6	16	1541	0	0	0	0	0	0	0	0
7	0	230	0	0	0	0	0	0	0	0
8	18	2	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0

For 25 25 neurons in two hidden layers



For layers [85,25,25,10]

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	406218	35789	0	0	0	0	0	0	0	0
1	122178	290173	0	0	0	0	0	0	0	0
2	3414	34651	0	0	0	0	0	0	0	0
3	543	9676	0	0	0	0	0	0	0	0
4	2442	1443	0	0	0	0	0	0	0	0
5	1811	124	0	0	0	0	0	0	0	0
6	16	1241	0	0	0	0	0	0	0	0
7	20	210	0	0	0	0	0	0	0	0
8	18	2	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0

e. Relu activation in hidden layers

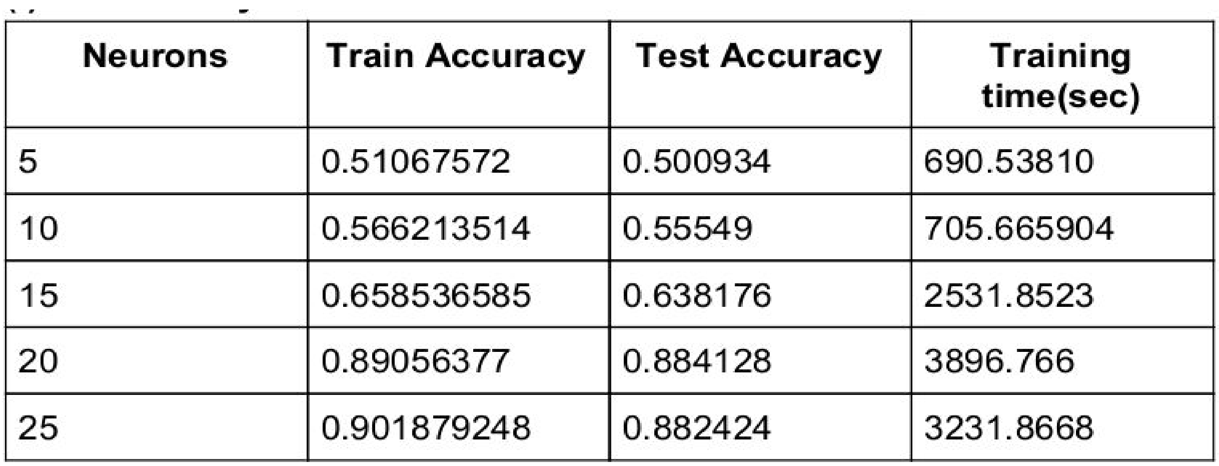
Relu improved the accuracy to some extent but the key point is it was quite faster than sigmoid activation. The training time was comparatively low.

Stopping condition:

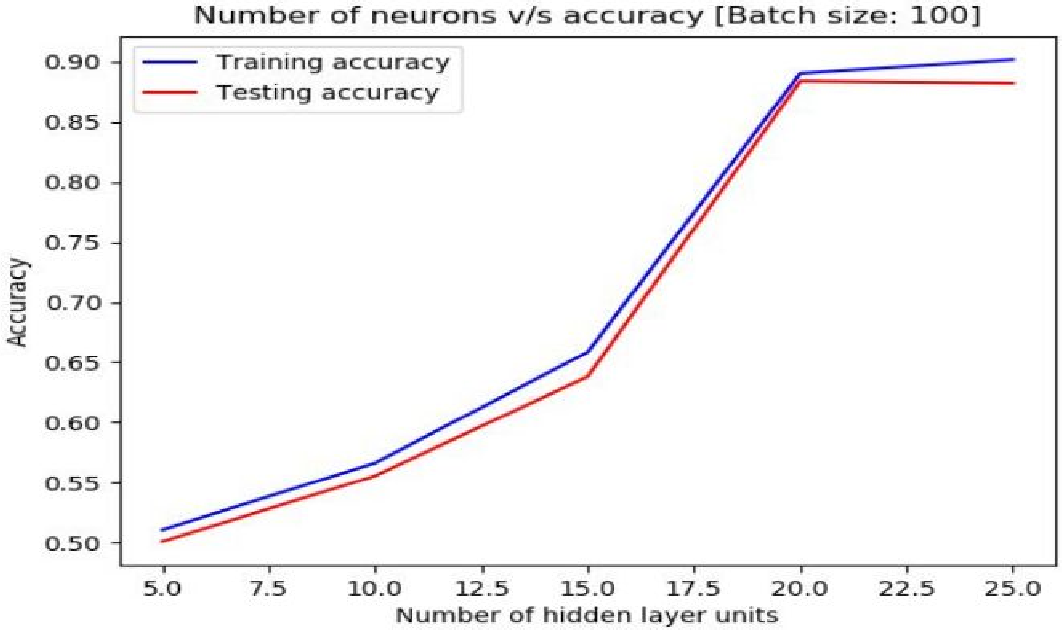
Error threshold: 10 ** -12

Epochs : 1000

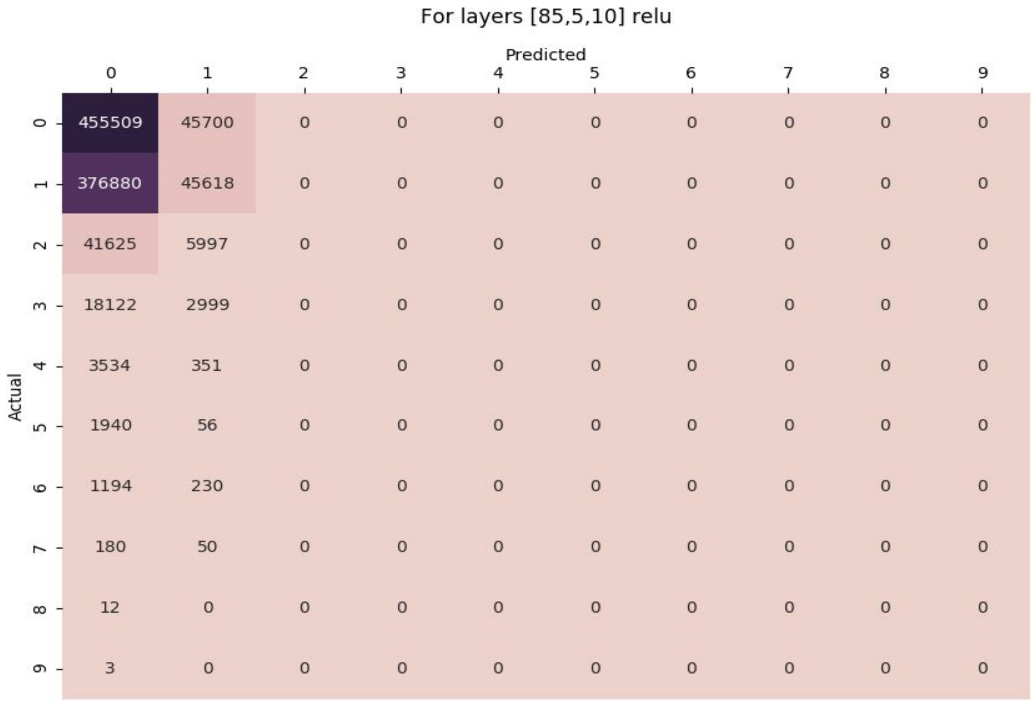
(i) For 1 hidden layer



Neurons	Train Accuracy	Test Accuracy	Training time(sec)
5	0.51067572	0.500934	690.53810
10	0.566213514	0.55549	705.665904
15	0.658536585	0.638176	2531.8523
20	0.89056377	0.884128	3896.766
25	0.901879248	0.882424	3231.8668



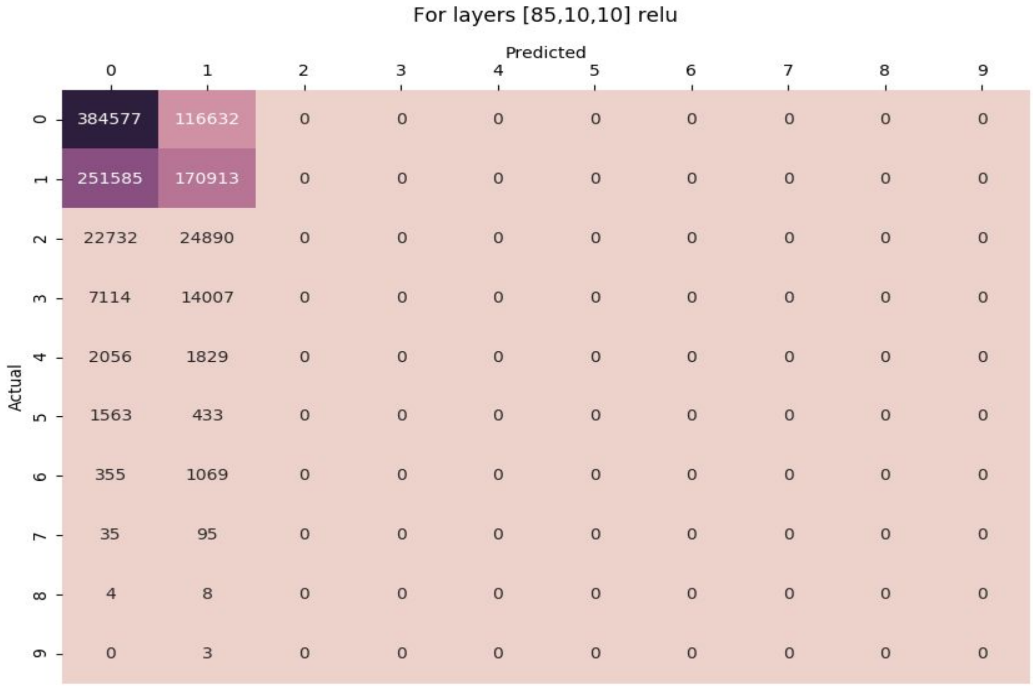
For 5 neurons in single layer



For layers [85,5,10] relu

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	455509	45700	0	0	0	0	0	0	0	0
1	376880	45618	0	0	0	0	0	0	0	0
2	41625	5997	0	0	0	0	0	0	0	0
3	18122	2999	0	0	0	0	0	0	0	0
4	3534	351	0	0	0	0	0	0	0	0
5	1940	56	0	0	0	0	0	0	0	0
6	1194	230	0	0	0	0	0	0	0	0
7	180	50	0	0	0	0	0	0	0	0
8	12	0	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0

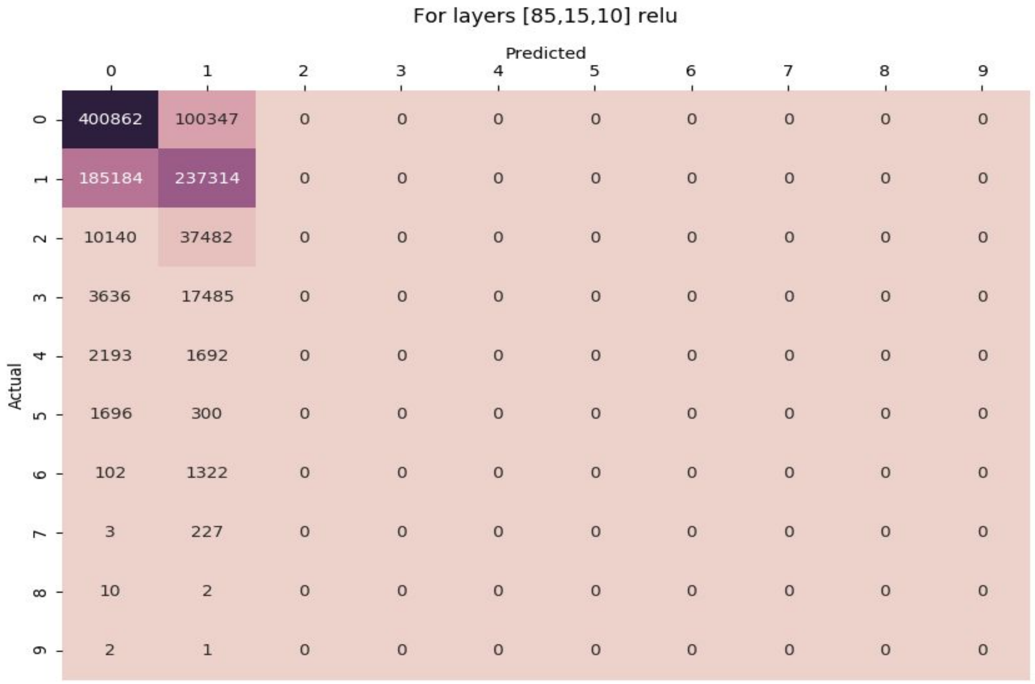
For 10 neurons in single layer



For layers [85,10,10] relu

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	384577	116632	0	0	0	0	0	0	0	0
1	251585	170913	0	0	0	0	0	0	0	0
2	22732	24890	0	0	0	0	0	0	0	0
3	7114	14007	0	0	0	0	0	0	0	0
4	2056	1829	0	0	0	0	0	0	0	0
5	1563	433	0	0	0	0	0	0	0	0
6	355	1069	0	0	0	0	0	0	0	0
7	35	95	0	0	0	0	0	0	0	0
8	4	8	0	0	0	0	0	0	0	0
9	0	3	0	0	0	0	0	0	0	0

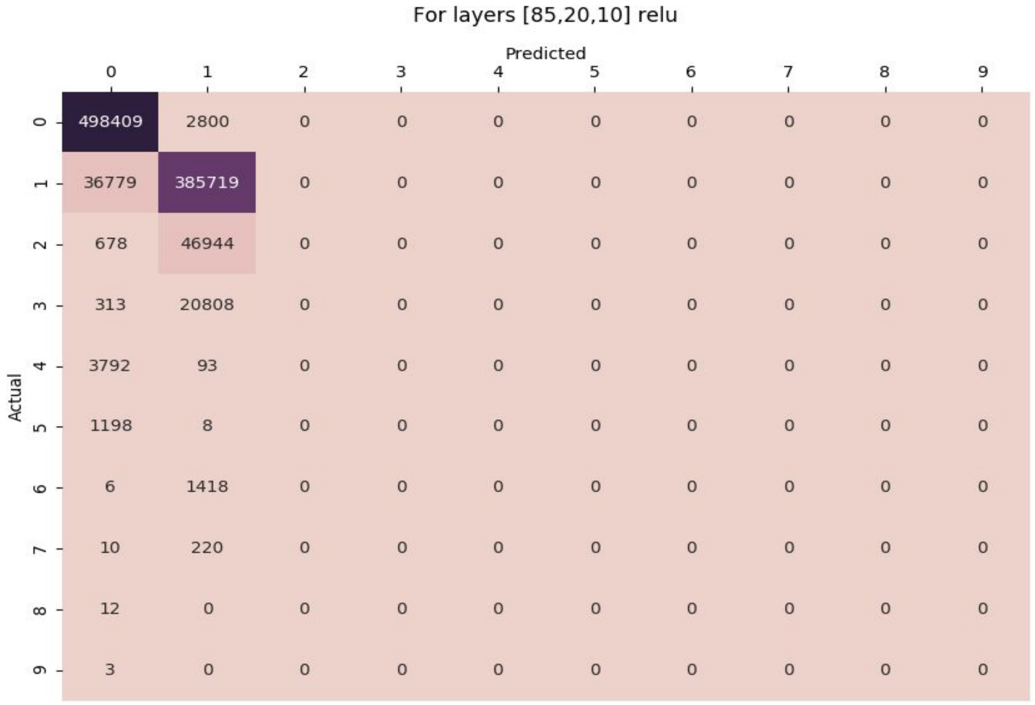
For 15 neurons in single layer



For layers [85,15,10] relu

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	400862	100347	0	0	0	0	0	0	0	0
1	185184	237314	0	0	0	0	0	0	0	0
2	10140	37482	0	0	0	0	0	0	0	0
3	3636	17485	0	0	0	0	0	0	0	0
4	2193	1692	0	0	0	0	0	0	0	0
5	1696	300	0	0	0	0	0	0	0	0
6	102	1322	0	0	0	0	0	0	0	0
7	3	227	0	0	0	0	0	0	0	0
8	10	2	0	0	0	0	0	0	0	0
9	2	1	0	0	0	0	0	0	0	0

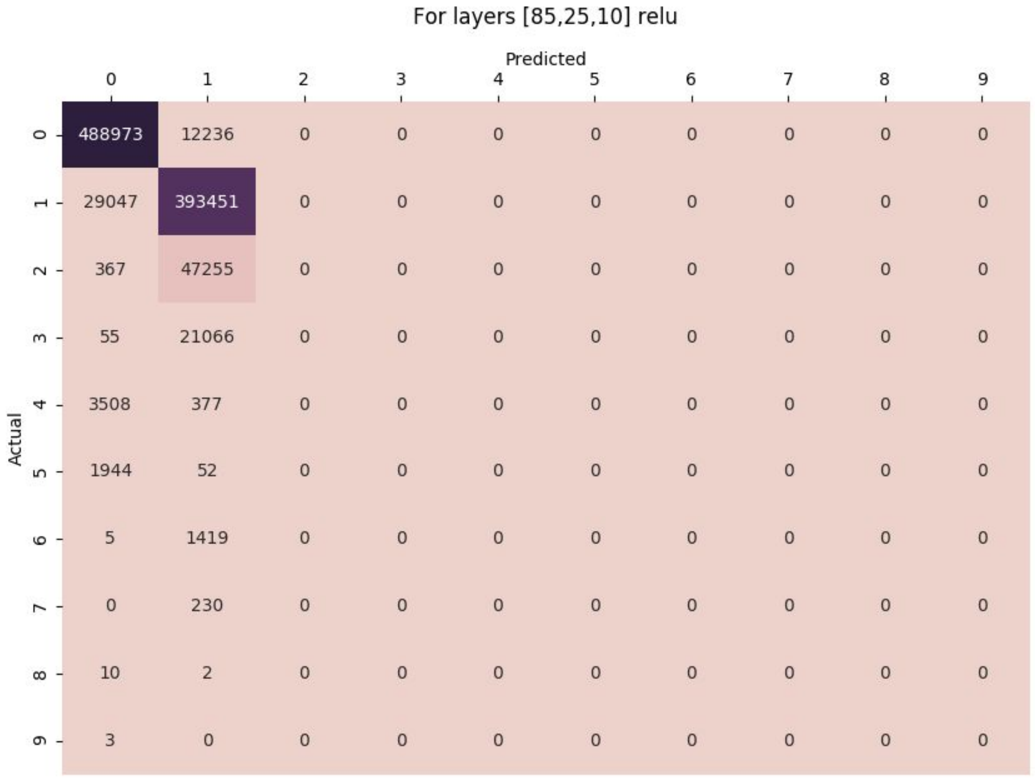
For 20 neurons in single layer



For layers [85,20,10] relu

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	498409	2800	0	0	0	0	0	0	0	0
1	36779	385719	0	0	0	0	0	0	0	0
2	678	46944	0	0	0	0	0	0	0	0
3	313	20808	0	0	0	0	0	0	0	0
4	3792	93	0	0	0	0	0	0	0	0
5	1198	8	0	0	0	0	0	0	0	0
6	6	1418	0	0	0	0	0	0	0	0
7	10	220	0	0	0	0	0	0	0	0
8	12	0	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0

For 25 neurons in single layer



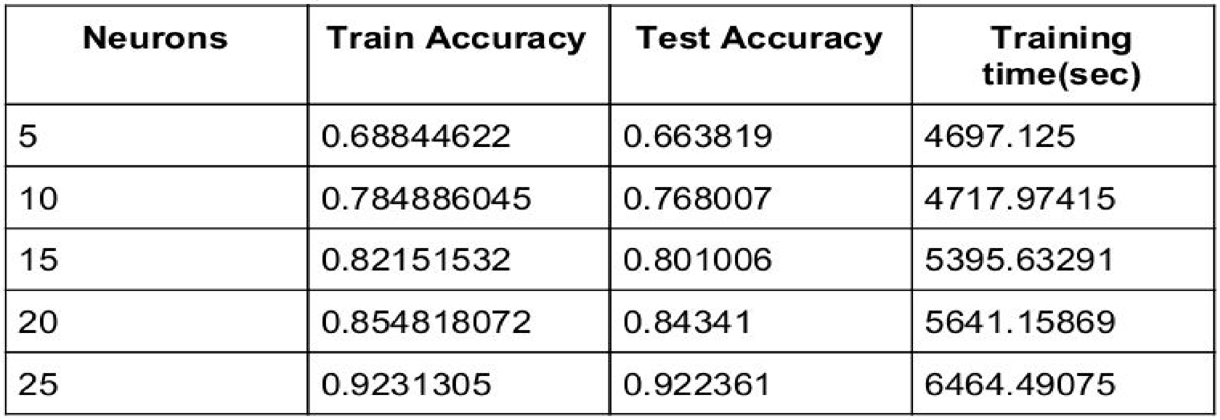
For layers [85,25,10] relu

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	488973	12236	0	0	0	0	0	0	0	0
1	29047	393451	0	0	0	0	0	0	0	0
2	367	47255	0	0	0	0	0	0	0	0
3	55	21066	0	0	0	0	0	0	0	0
4	3508	377	0	0	0	0	0	0	0	0
5	1944	52	0	0	0	0	0	0	0	0
6	5	1419	0	0	0	0	0	0	0	0
7	0	230	0	0	0	0	0	0	0	0
8	10	2	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0

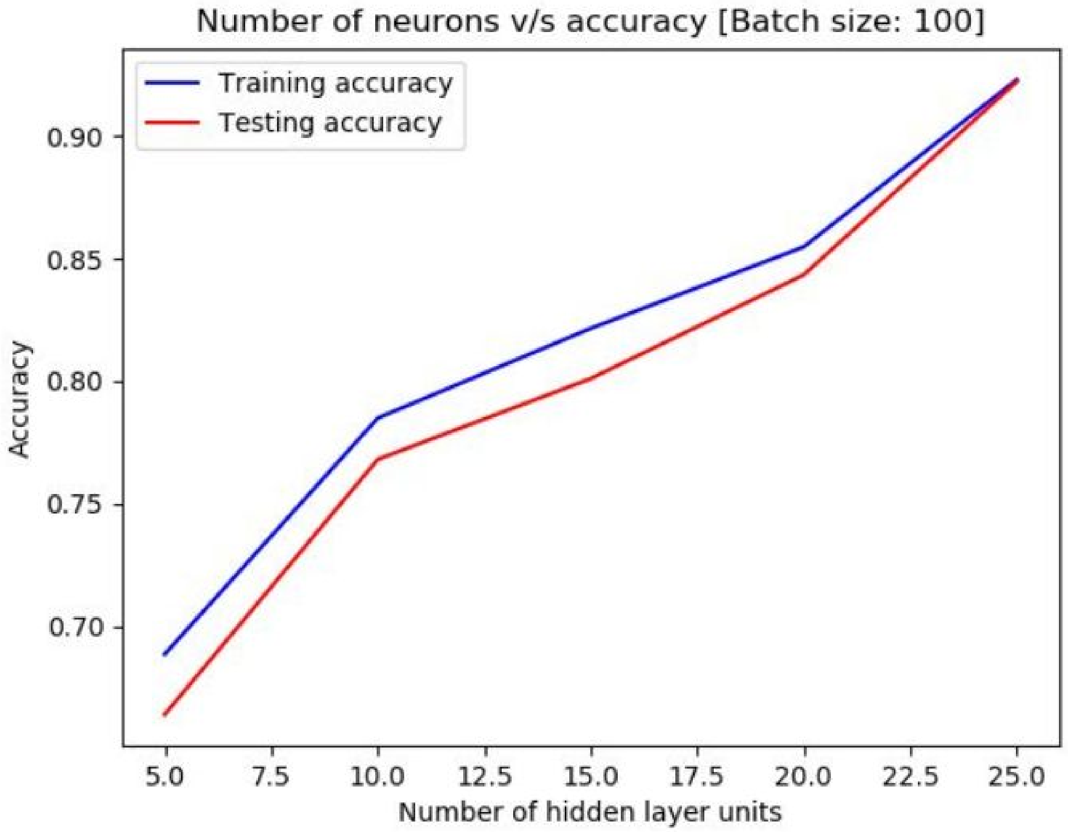
(ii) 2 hidden layers

Error threshold : 10 ** -12

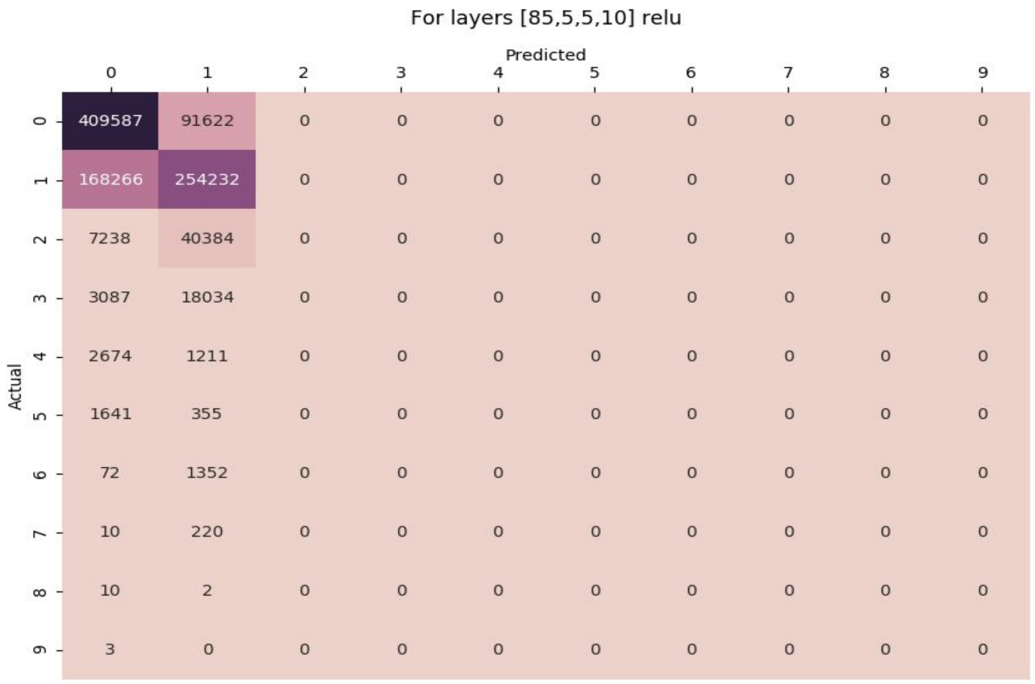
Epochs: 1000



Neurons	Train Accuracy	Test Accuracy	Training time(sec)
5	0.68844622	0.663819	4697.125
10	0.784886045	0.768007	4717.97415
15	0.82151532	0.801006	5395.63291
20	0.854818072	0.84341	5641.15869
25	0.9231305	0.922361	6464.49075

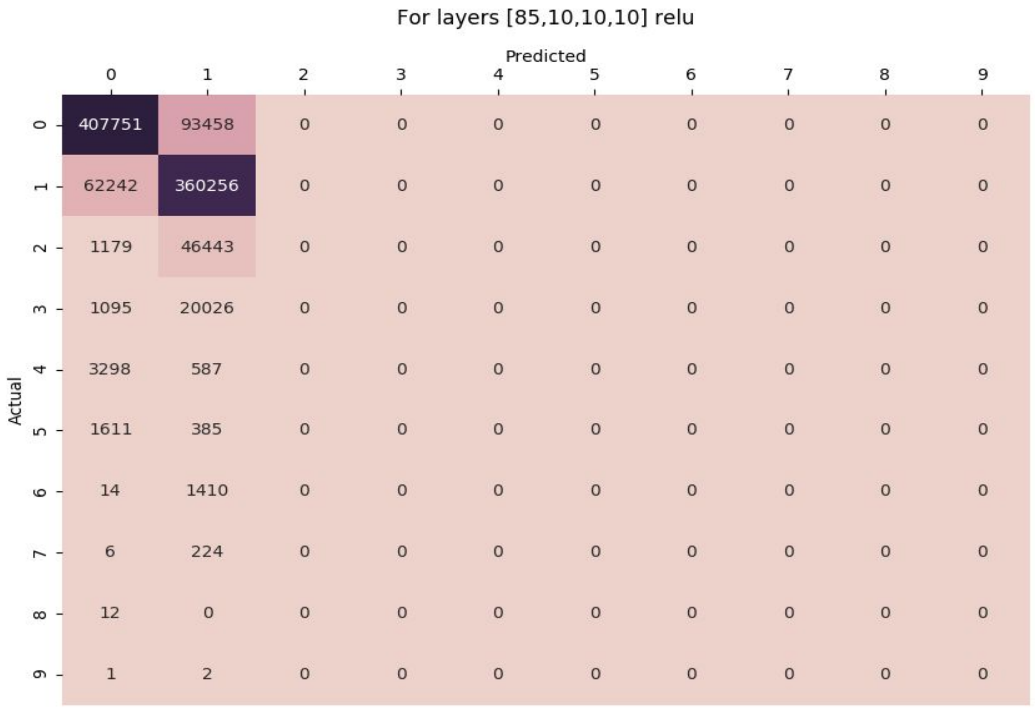


For 5 5 neurons in two layers


For layers [85,5,5,10] relu

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	409587	91622	0	0	0	0	0	0	0	0
1	168266	254232	0	0	0	0	0	0	0	0
2	7238	40384	0	0	0	0	0	0	0	0
3	3087	18034	0	0	0	0	0	0	0	0
4	2674	1211	0	0	0	0	0	0	0	0
5	1641	355	0	0	0	0	0	0	0	0
6	72	1352	0	0	0	0	0	0	0	0
7	10	220	0	0	0	0	0	0	0	0
8	10	2	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0

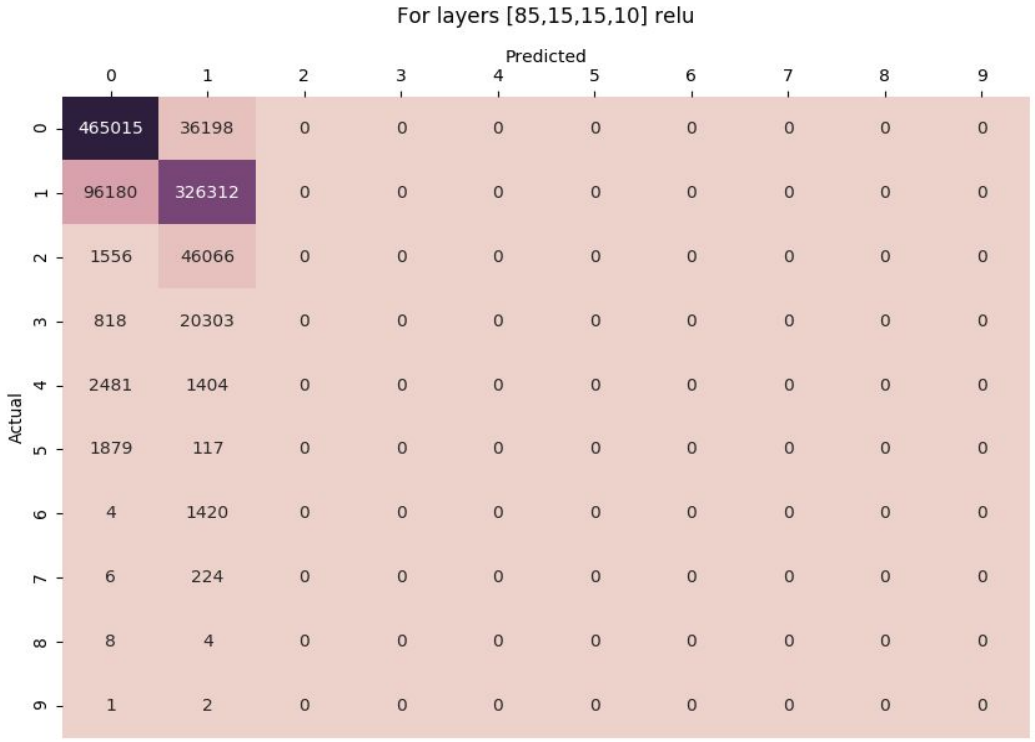
For 10 10 neurons in two hidden layers



For layers [85,10,10,10] relu

	0	1	2	3	4	5	6	7	8	9
0	407751	93458	0	0	0	0	0	0	0	0
1	62242	360256	0	0	0	0	0	0	0	0
2	1179	46443	0	0	0	0	0	0	0	0
3	1095	20026	0	0	0	0	0	0	0	0
4	3298	587	0	0	0	0	0	0	0	0
5	1611	385	0	0	0	0	0	0	0	0
6	14	1410	0	0	0	0	0	0	0	0
7	6	224	0	0	0	0	0	0	0	0
8	12	0	0	0	0	0	0	0	0	0
9	1	2	0	0	0	0	0	0	0	0

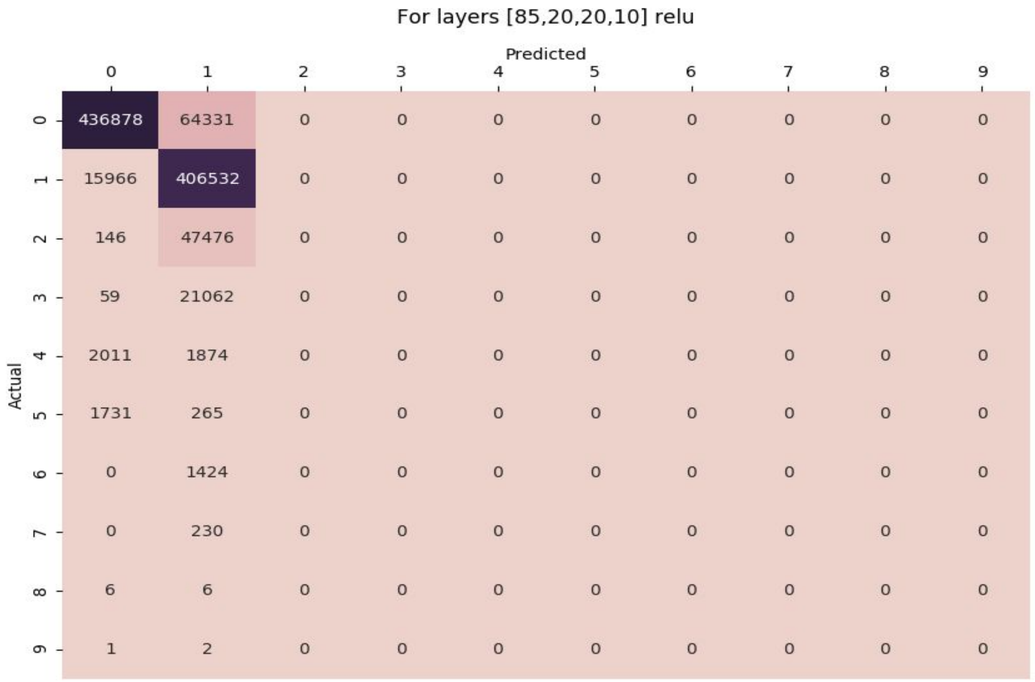
For 15 15 neurons in two hidden layers



For layers [85,15,15,10] relu

	0	1	2	3	4	5	6	7	8	9
0	465015	36198	0	0	0	0	0	0	0	0
1	96180	326312	0	0	0	0	0	0	0	0
2	1556	46066	0	0	0	0	0	0	0	0
3	818	20303	0	0	0	0	0	0	0	0
4	2481	1404	0	0	0	0	0	0	0	0
5	1879	117	0	0	0	0	0	0	0	0
6	4	1420	0	0	0	0	0	0	0	0
7	6	224	0	0	0	0	0	0	0	0
8	8	4	0	0	0	0	0	0	0	0
9	1	2	0	0	0	0	0	0	0	0

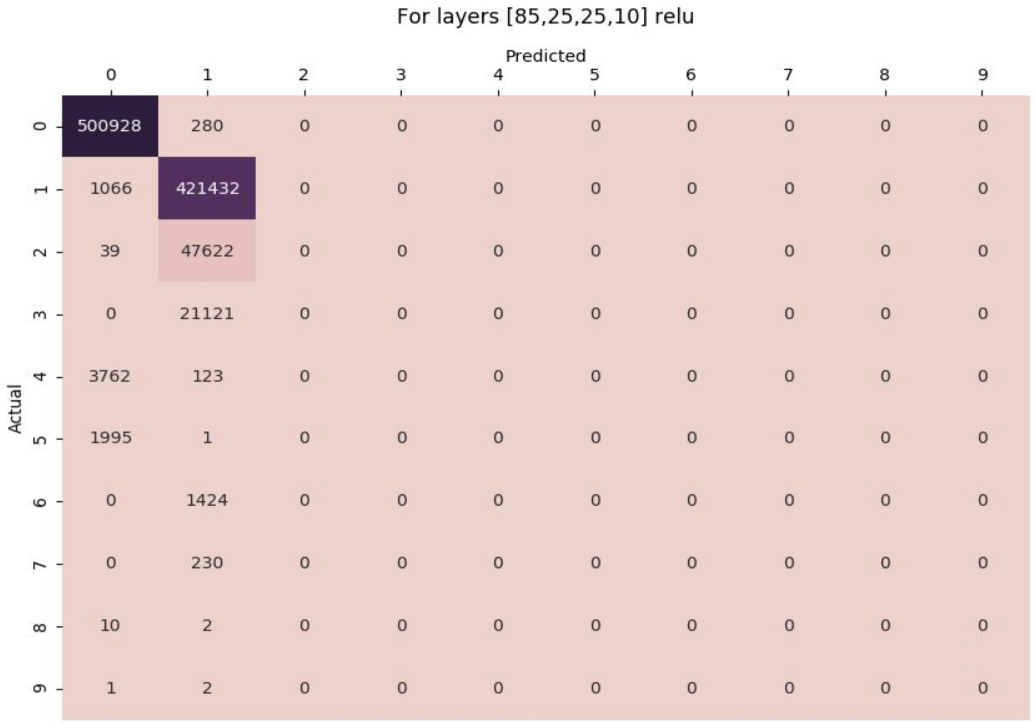
For 20 20 neurons in two hidden layer



For layers [85,20,20,10] relu

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	436878	64331	0	0	0	0	0	0	0	0
1	15966	406532	0	0	0	0	0	0	0	0
2	146	47476	0	0	0	0	0	0	0	0
3	59	21062	0	0	0	0	0	0	0	0
4	2011	1874	0	0	0	0	0	0	0	0
5	1731	265	0	0	0	0	0	0	0	0
6	0	1424	0	0	0	0	0	0	0	0
7	0	230	0	0	0	0	0	0	0	0
8	6	6	0	0	0	0	0	0	0	0
9	1	2	0	0	0	0	0	0	0	0

For 25 25 neurons in two hidden layer



For layers [85,25,25,10] relu

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	500928	280	0	0	0	0	0	0	0	0
1	1066	421432	0	0	0	0	0	0	0	0
2	39	47622	0	0	0	0	0	0	0	0
3	0	21121	0	0	0	0	0	0	0	0
4	3762	123	0	0	0	0	0	0	0	0
5	1995	1	0	0	0	0	0	0	0	0
6	0	1424	0	0	0	0	0	0	0	0
7	0	230	0	0	0	0	0	0	0	0
8	10	2	0	0	0	0	0	0	0	0
9	1	2	0	0	0	0	0	0	0	0