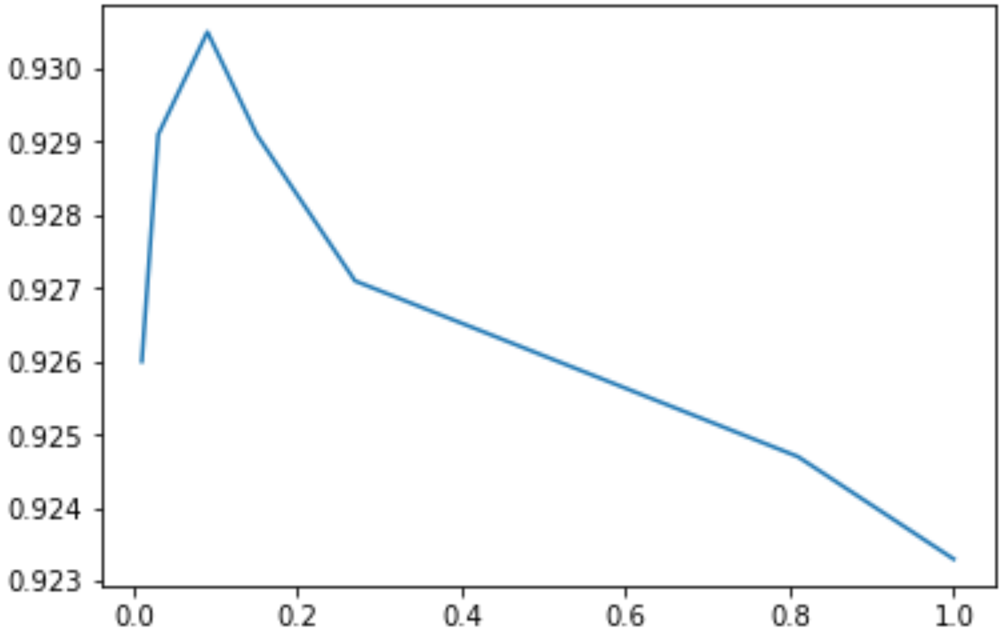
Project 3 - Classification.

The classification task will be that of recognizing a 28×28 grayscale handwritten digit image and identify it as a digit among 0, 1, 2, ... , 9. It consists of implementation of ensemble of four classifiers where results of the individual classifiers are combined to make a final decision. The four classifiers trained are as follows:

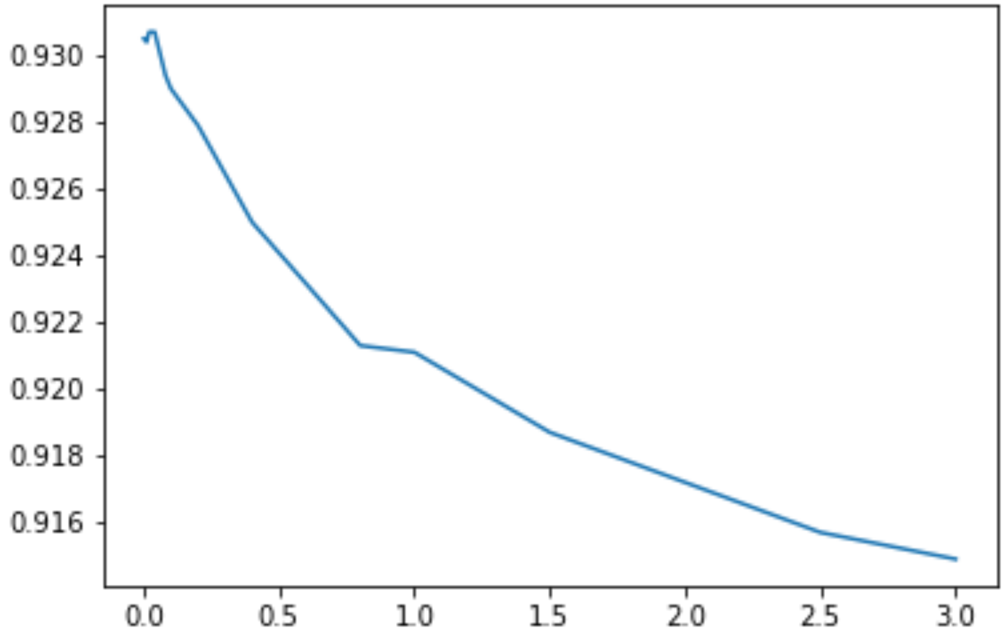
1. Softmax Linear Regression.

This is a multi class linear regression model, where we the design matrix is the 28x28 pixel matrix. We also add a bias term to each sample of this design matrix. After computing the 'logits', we apply a softmax function on them which gives us a probability distribution over each class, and we select the class that has the highest probability. Also, we use batch-gradient descent instead of stochastic one, to train the weights of our model.
We tune this model over a range of hyper parameters such as learning rate, regularization parameter, number of epochs on the Cross Validation Data Set.
We have the following graphs to depict them:

1. Graph for Learning Rate vs. Accuracy(Cross Validation):



2. Graph for Regularization Parameter vs. Accuracy:



Therefore, our most optimal choice of hyper parameters is as follows:

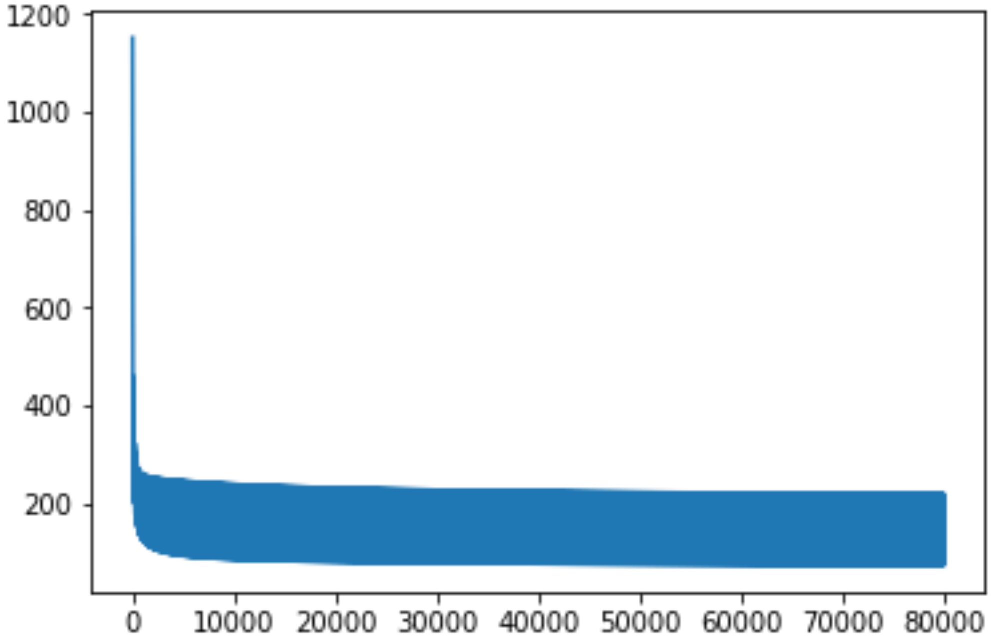
1. Learning Rate = 0.09
2. Regularization Parameter = 0.03
3. Number of epochs = 8000
4. Batch size = 128

After training this model on the MNIST training data, we have the following metrics for the best possible choice of hyperpatameters:

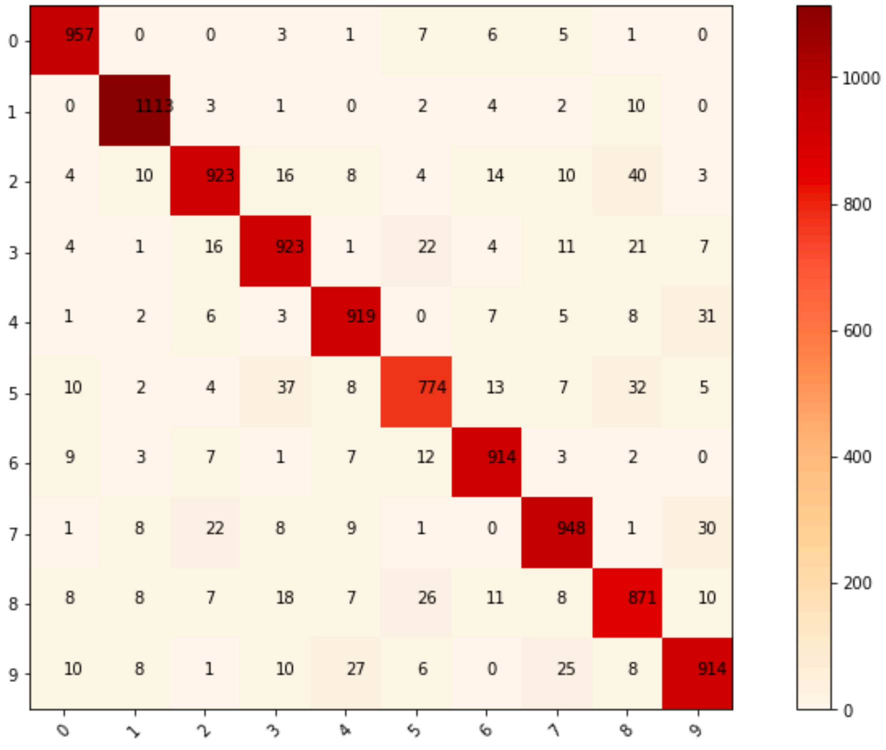
MNIST:

```
Accuracy on training data   : 0.9352
Accuracy on validation data : 0.9299
Accuracy on test data       : 0.9256
USPS:
Accuracy : 0.3135656782839142
```

We also got the plot of 'loss function' while training as given under:



Confusion Matrix for this model on test data:



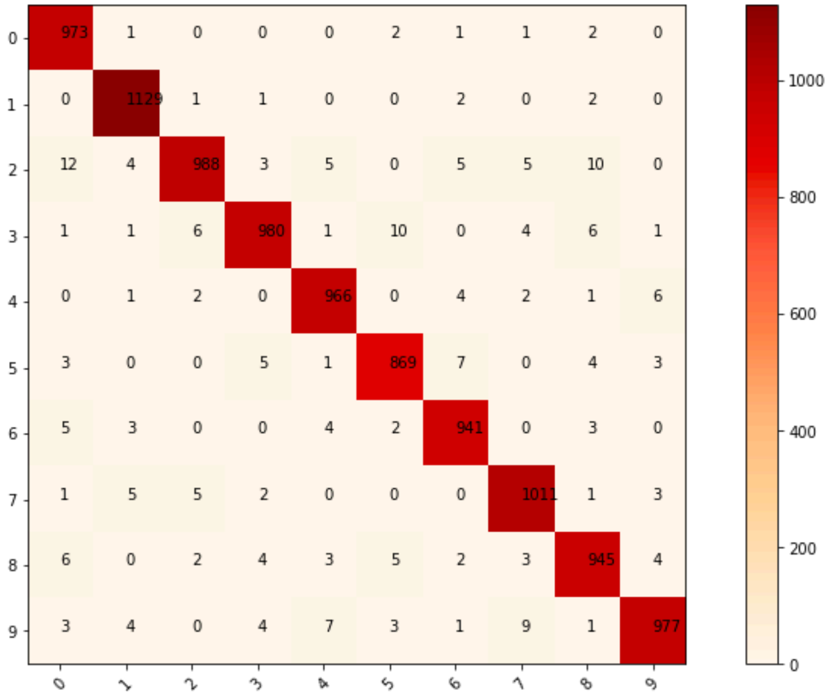
	0	1	2	3	4	5	6	7	8	9
0	957	0	0	3	1	7	6	5	1	0
1	0	1113	3	1	0	2	4	2	10	0
2	4	10	923	16	8	4	14	10	40	3
3	4	1	16	923	1	22	4	11	21	7
4	1	2	6	3	919	0	7	5	8	31
5	10	2	4	37	8	774	13	7	32	5
6	9	3	7	1	7	12	914	3	2	0
7	1	8	22	8	9	1	0	948	1	30
8	8	8	7	18	7	26	11	8	871	10
9	10	8	1	10	27	6	0	25	8	914

2 . Deep Neural Network:

After trying a various configurations of hyper parameters, our neural network model gave the best performance for the following set of hyper parameters:
Hidden Layers: 2 hidden layers of 800 nodes each.
Number of epochs: 3000
Learning rate: 0.5
Optimization method: GradientDescentOptimizer
Activation Method: RELU

Training accuracy : 0.99384
Test accuracy MNIST : 0.9779
Loss: 0.018650109
Test accuracy USPS: 0.5077753887694385

Confusion Matrix:



	0	1	2	3	4	5	6	7	8	9
0	973	1	0	0	0	2	1	1	2	0
1	0	1129	1	1	0	0	2	0	2	0
2	12	4	988	3	5	0	5	5	10	0
3	1	1	6	980	1	10	0	4	6	1
4	0	1	2	0	966	0	4	2	1	6
5	3	0	0	5	1	869	7	0	4	3
6	5	3	0	0	4	2	941	0	3	0
7	1	5	5	2	0	0	0	1011	1	3
8	6	0	2	4	3	5	2	3	945	4
9	3	4	0	4	7	3	1	9	1	977

3. Support Vector Machine:

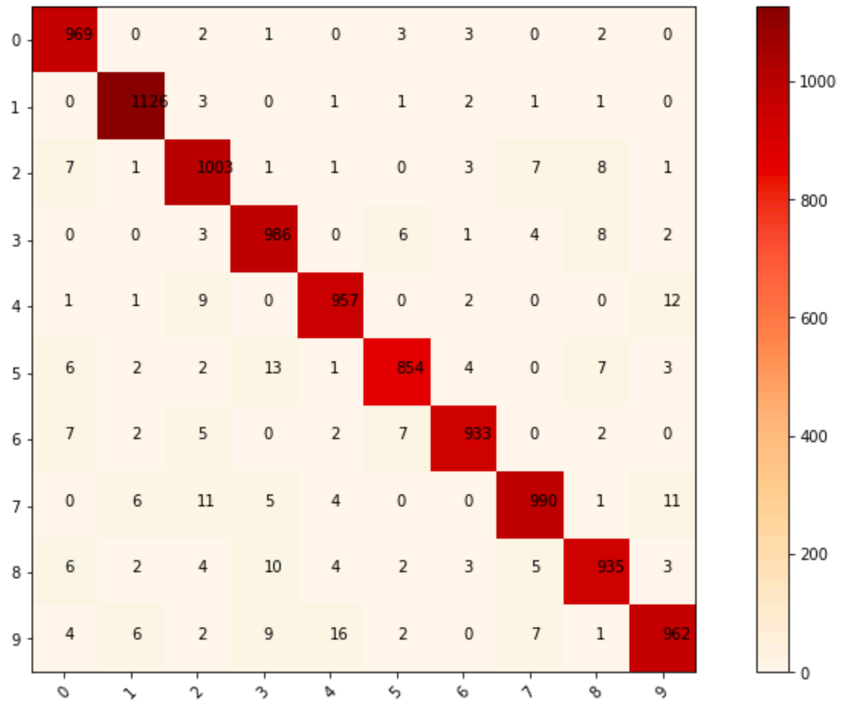
For the Support Vector we used the sklearn’s grid search over different values of hyper parameters: kernel, C (smoothness), Gamma(spread).
We ran the grid-search over the following ranges of parameters:
‘kernel': ('linear', 'rbf'),
‘C': [1, 10, 100],
'gamma': [0.001, 0.0001]

Note: we only used 10000 training samples of MNIST to run the grid search.

The grid-search returned the best model with following configuration and metrics:
‘kernel': 'rbf',
‘C': 100,
'gamma': 0.001

Accuracy on MNIST test data: 0.9715
Accuracy on USPS data: 0.3530176508825441

Confusion Matrix:

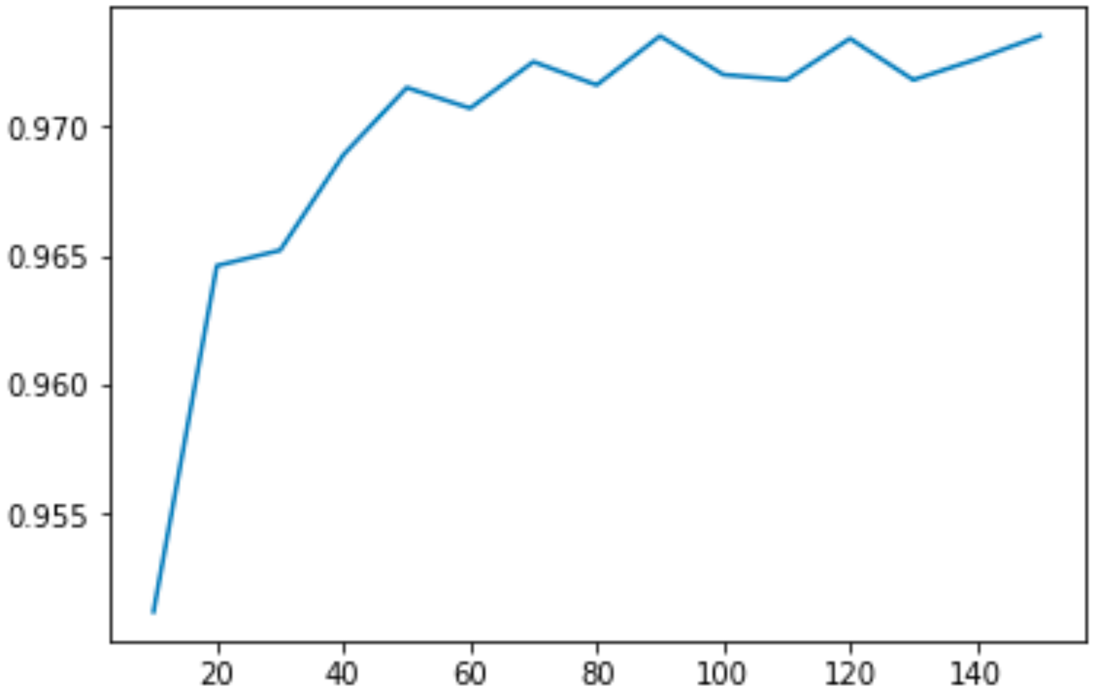


	0	1	2	3	4	5	6	7	8	9
0	969	0	2	1	0	3	3	0	2	0
1	0	1126	3	0	1	1	2	1	1	0
2	7	1	1003	1	1	0	3	7	8	1
3	0	0	3	986	0	6	1	4	8	2
4	1	1	9	0	957	0	2	0	0	12
5	6	2	2	13	1	854	4	0	7	3
6	7	2	5	0	2	7	933	0	2	0
7	0	6	11	5	4	0	0	990	1	11
8	6	2	4	10	4	2	3	5	935	3
9	4	6	2	9	16	2	0	7	1	962

4. Random Forest Classifier:

This classifier was tuned for only one hyper parameter, that is n_estimators, which is the number of decision trees in the random forest.

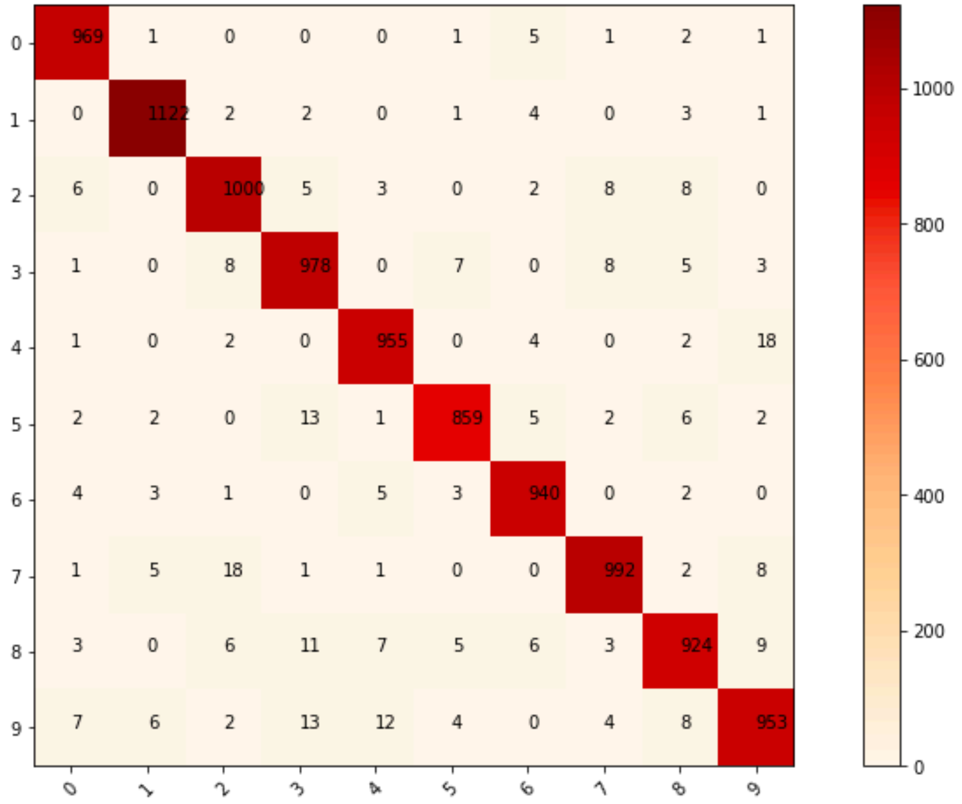
This is the plot of accuracies vs. n_estimators for validation data:



Thus, we see that the accuracy is highest when n_estimators = 150. After training and testing the model with this configuration, we had the following metrics:

```
Accuracy on MNIST:0.9692
Accuracy on USPS:0.39156957847892393
```

```
Confusion Matrix for Random Forest Classifier:
```



	0	1	2	3	4	5	6	7	8	9
0	969	1	0	0	0	1	5	1	2	1
1	0	1122	2	2	0	1	4	0	3	1
2	6	0	1000	5	3	0	2	8	8	0
3	1	0	8	978	0	7	0	8	5	3
4	1	0	2	0	955	0	4	0	2	18
5	2	2	0	13	1	859	5	2	6	2
6	4	3	1	0	5	3	940	0	2	0
7	1	5	18	1	1	0	0	992	2	8
8	3	0	6	11	7	5	6	3	924	9
9	7	6	2	13	12	4	0	4	8	953

Ensembling:

At last, we use an ensemble of the of the individual classifiers by implementing a voting functionality. For a particular sample, we choose the class that gets selected by most number of individual classifiers. In case of a tie, the class selected by classifiers whose accuracies sum up to a greater value is elected.
After ensembling, the accuracy of our system further increases to 0.9786

Conclusion:

1. The deep neural network model performs the best among all 4 individual classifiers.
2. The ensembling of individual classifiers increases the overall performance of the system as a data point misclassified by one model can be classified correctly by other models, since the probability of all the classes misclassifying the same data sample is low.
3. From the confusion matrices of the individual classifiers, we can observe that the model with higher accuracy also has higher precision and recall.
4. Though, the models and their ensemble perform very well on MNIST data set, they don't perform well on USPS data set. This is because a model trained on one data set cannot suffice for every other data set of the same domain, as the patterns observed in one data set may be different than the other. This follows the "no free lunch" theorem, which states that a model can perform as expected only on the kind of data on which it was trained. If the training and testing data have different sources or curations, the performance will drop.