COL774 - Machine Learning: Assignment - 3

Param Khakhar - 2018CS10362

10 December 2020

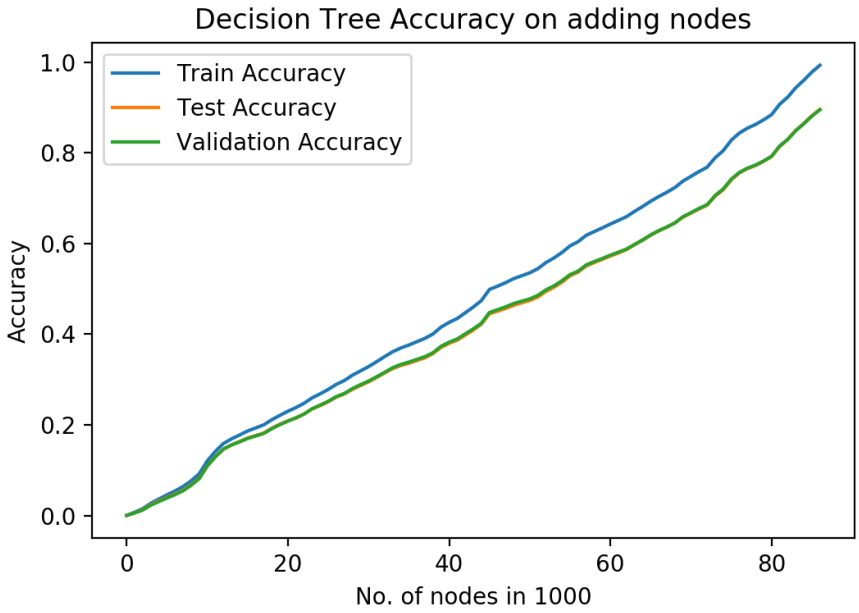
Introduction

The following is a writeup for the Assignment-3, Machine Learning (COL774). There are sections corresponding to each of the four questions and within each section, there are sub-sections for each sub-part.

Task -1 : Decision Trees and Random Forests

Task - 1.a: Implementing Decision Trees

The following graph was obtained for the accuracy vs no. of nodes added in the decision tree.

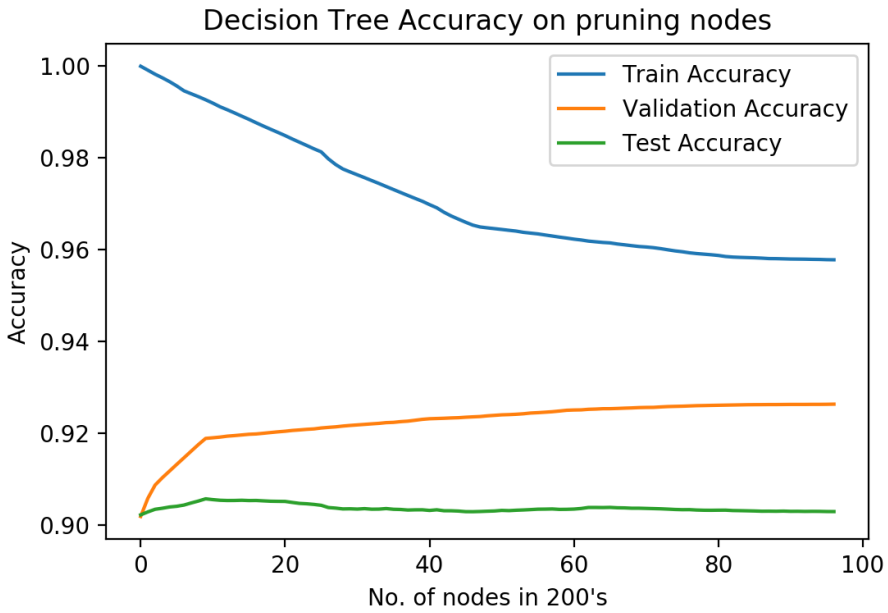


The graph begins starts from 0, and then as and when the nodes are added, the accuracy does increase. The accuracy for validation and test sets almost overlap each other. The linear increase in accuracy with nodes is consistent theoretically. The total number of nodes are 86258, and the final accuracies on the **train**, **validation** and **test** set are **0.999**, **0.902**, **0.902** respectively. Since, the accuracy on the validation set isn't saturating or decreasing after certain number of nodes, the model doesn't seem to overfit the data.

Task - 1.b: Pruning Decision Trees

Each instance of the validation data is passed down the tree and a count is maintained for all the nodes in the path which is incremented if the label of the instance doesn't match the majority class of the data at the node (to be assigned while growing the tree). Then, a second pass is made over the entire tree wherein, a priority queue for the node is formed where the nodes are ordered according to the difference of their error - the error of their children. Here, error refers to the number of instances misclassified if the node was a leaf node. The node having the minimum value of its error - error of their children is selected and pruned if this value is negative. The following trend

is observed for the accuracy vs number of nodes to pruned.



Around **20,000** nodes were pruned and the there's an increase in the validation accuracy along with the decrease in the accuracy on the training set. The accuracy on the test set slightly increases. After pruning, the accuracy on the **train**, **validation** and **test** set is **0.958**, **0.926** and **0.903** respectively. The removal of this many nodes without affecting the validation and test accuracies signifies the overfitting during growing decision trees.

Task - 1.c: Random Forests

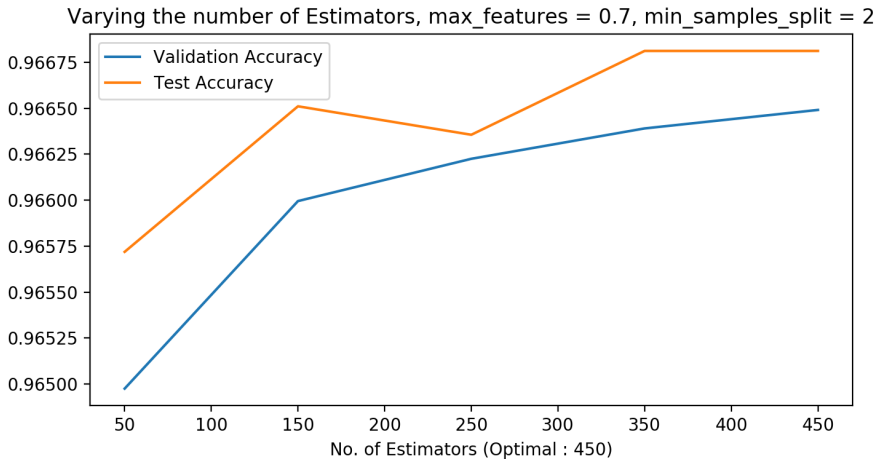
The best set of hyper-parameters obtained while performing the grid search with out-of-bag score is: **n_estimators: 450**, **max_features: 0.7**, and **min_samples_split: 2**. The following scores were obtained on the train, validation and test sets. The criterion used for sklearn is 'entropy'.
Train Accuracy: **1.0**
Out of Bag Score, for training samples: **0.9663**
Validation Accuracy: **0.9662**
Test Accuracy: **0.9671**

A total of 125 fits were made corresponding to all the hyperparameter combinations. The accuracy scores obtained is higher as compared to using

only decision tree. Higher accuracies also indicate less overfitting and more generalization of Random Forests as compared to Decision Trees.

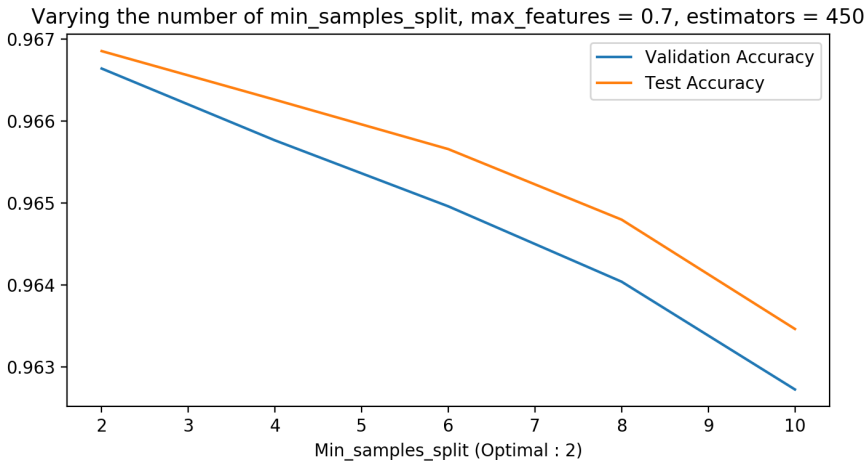
Task - 1.d: Parameter Sensitivity Analysis

- **Number of Estimators:**



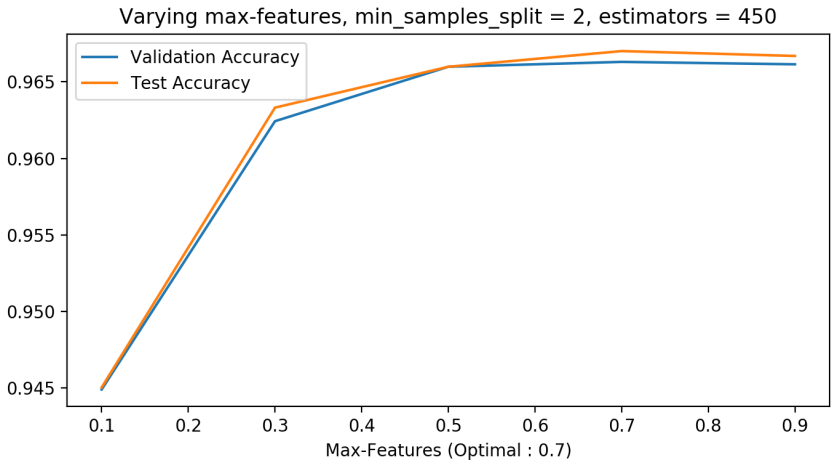
The optimal value obtained is 450, but there aren't any significant changes to the accuracy values on the validation and test set for different number of estimators. However, using **higher** number of estimators would result in **increased training time**.

- **Min_samples_split:**



Min_samples_split is more sensitive to accuracy as compared to the number of estimators. For the data, provided the highest accuracy values are obtained for min_samples_split = 2. The optimal value of this hyperparameter would be a characteristic of the dataset.

- **Max_features:**



Max_features is the hyperparameter which is the most sensitive to the accuracy values. The optimal value obtained is 0.7, again which would be characteristic of the dataset.

Task -2 :Neural Networks

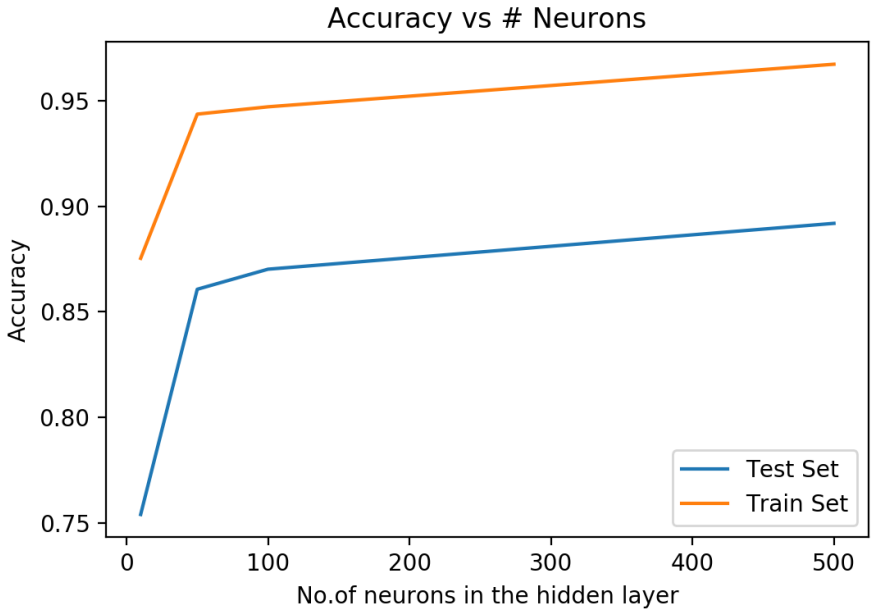
Task - 2.a: Single Hidden Layer

The **Stopping Criteria** used is to compute the maximum absolute difference between the average error for all the batches, for last 10 epochs. The threshold is set at 5e-05. The **weights** are initialized by sampling from a normal distribution with mean 0 and variance 1.

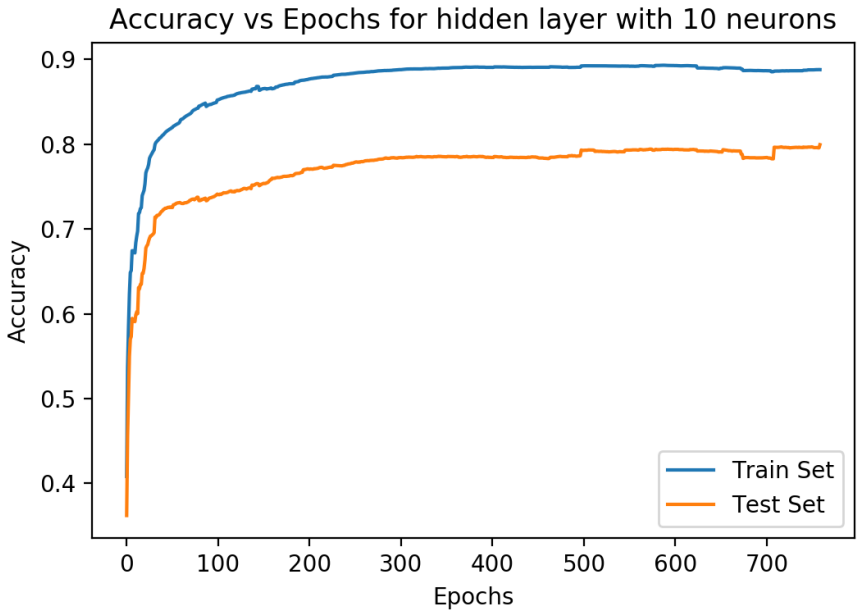
- **Accuracy:** The following table represents the final accuracies obtained for the training and the test set.

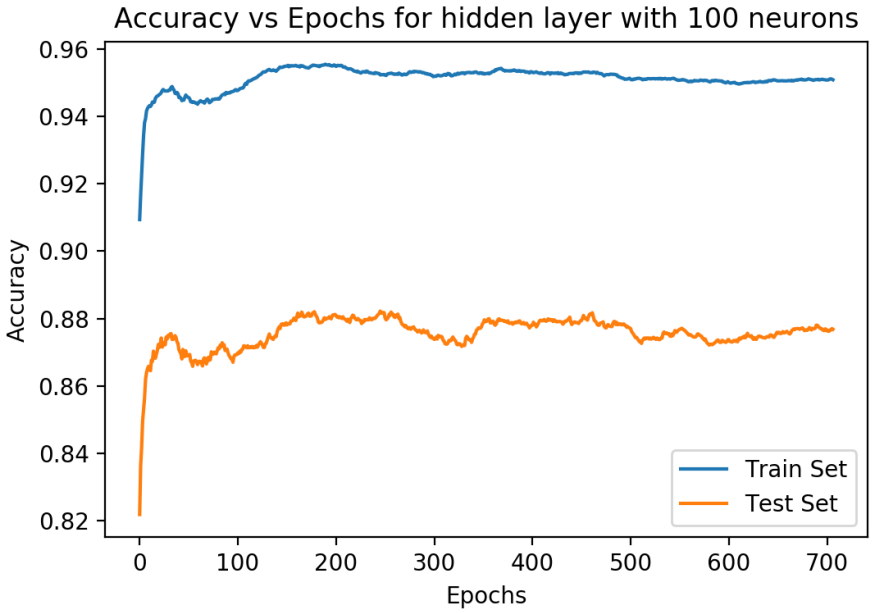
Neurons in Hidden Layer	Accuracy on Train Set	Accuracy on Test Set
1	0.1	0.1
10	0.875	0.754
50	0.944	0.861
100	0.947	0.870
500	0.967	0.892

The graph for the above table, without including the single neuron case.



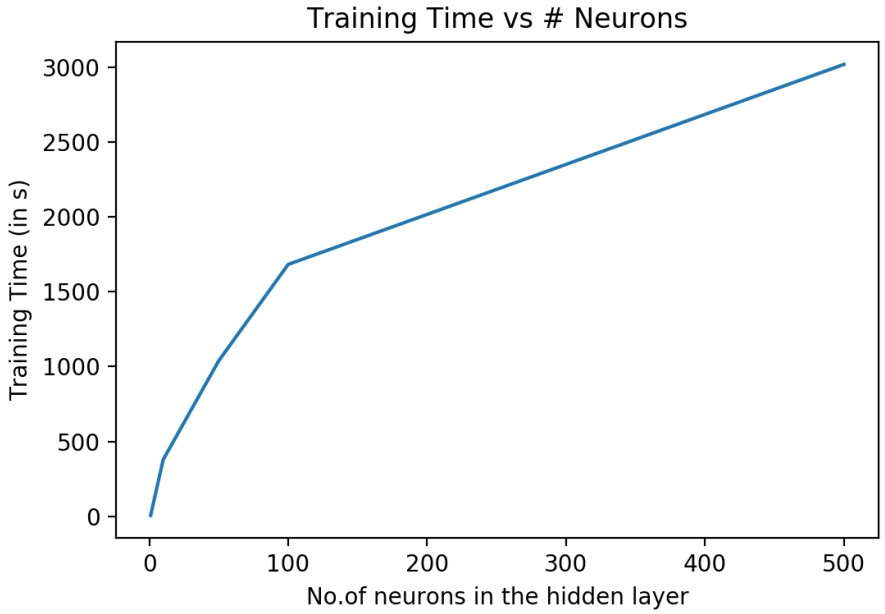
The plot of **accuracy vs epochs** for different number of neurons in the hidden layer during training is as follows:


Accuracy vs Epochs for hidden layer with 10 neurons
Accuracy
Epochs
Train Set
Test Set
0.9
0.8
0.7
0.6
0.5
0.4
0
100
200
300
400
500
600
700


Accuracy vs Epochs for hidden layer with 100 neurons
Accuracy
Epochs
Train Set
Test Set
0.96
0.94
0.92
0.90
0.88
0.86
0.84
0.82
0
100
200
300
400
500
600
700

- **Training Time:** The following table represents the training time for different number of neurons in the hidden layer.

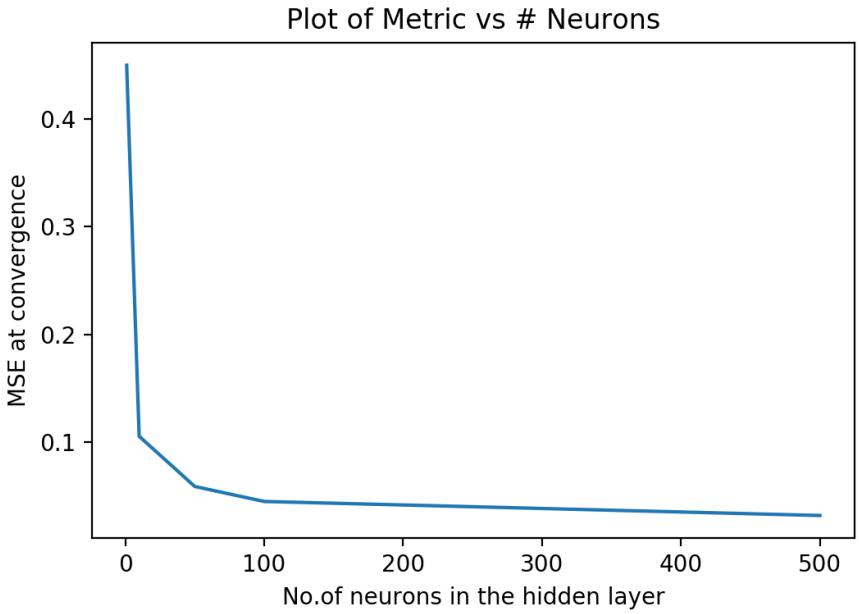
Neurons in Hidden Layer	Training Time in s
1	6.18s
10	378.01s
50	1039.16s
100	1683.32s
500	3020.15s



On increasing the number of neurons in the hidden layer, the matrix multiplication operations for forward pass and backward pass are now carried out on larger matrices, thereby resulting in higher training times.

- **Mean Squared Error at Convergence:** The following table represents the MSE at convergence for different number of neurons in the hidden layer.

Neurons in Hidden Layer	MSE at convergence
1	0.45
10	0.105
50	0.059
100	0.045
500	0.032



On increasing the number of neurons in the hidden layer, the model is better able to learn the patterns in the data, therefore resulting in lower MSE, and consequently higher accuracies.

Task - 2.b: Adaptive Learning Rate

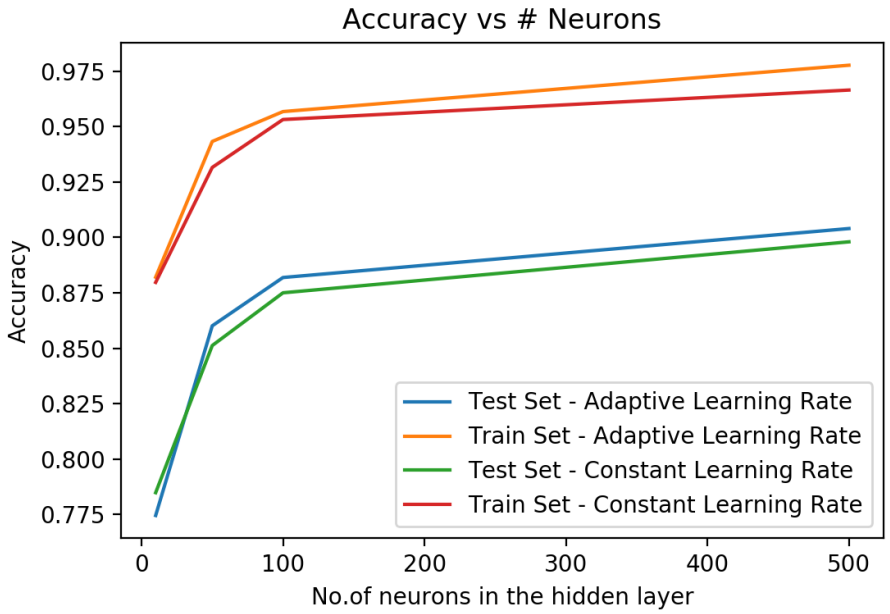
On experimentation, it was found that the **previous stopping criteria works well, but the threshold is now kept at 1e-04**, even for the

case of adaptive learning rate, therefore it is unchanged. The weights are initialized similarly to the constant learning case as well.

- **Accuracy:** The following table represents the final accuracies obtained for the training and the test set for adaptive learning rate.

Neurons in Hidden Layer	Accuracy on Train Set	Accuracy on Test Set
1	0.1	0.1
10	0.882	0.775
50	0.943	0.860
100	0.957	0.882
500	0.978	0.904

The graph for the accuracy measurements on the training and the test set is as follows:

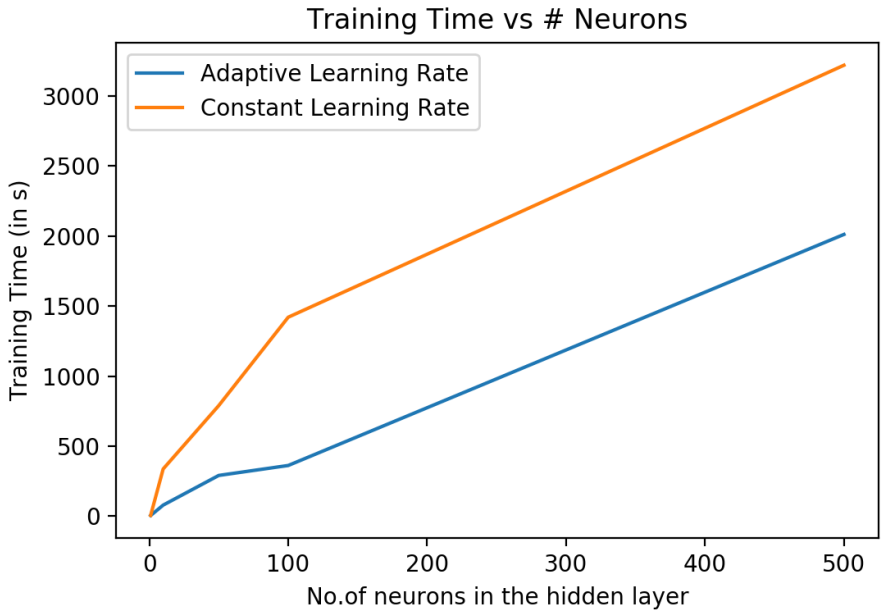


The accuracy for adaptive learning rate is slightly better as initially the model doesn't get stuck to local optima, and gradually the learning rate decreases thereby reducing the fluctuations near the global optima.

- **Training Time:** The following table represents the training time for

different number of neurons in the hidden layer while using the adaptive learning rate.

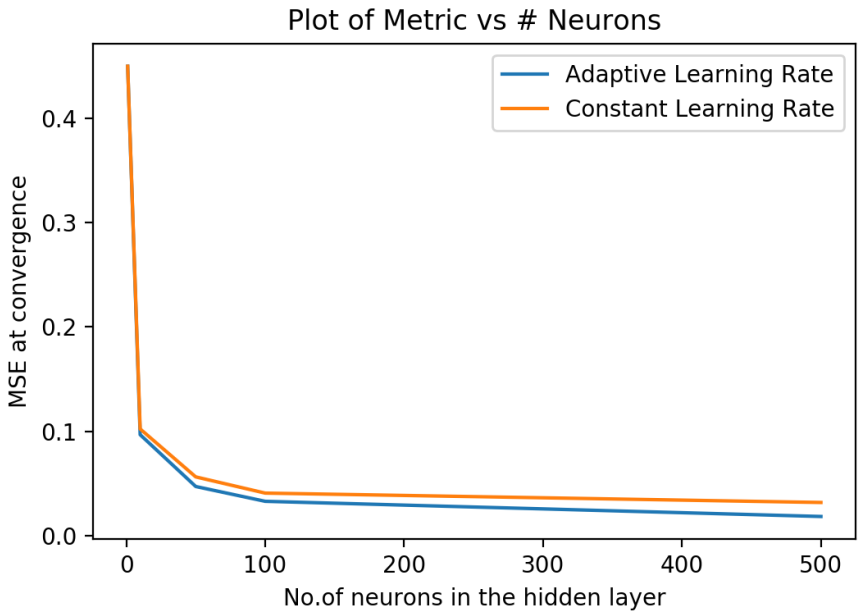
Neurons	Training Time (Constant LR)	Training Time (Adaptive LR)
1	6.18s	5.28s
10	378.01s	79.29s
50	1039.16s	291.09s
100	1683.32s	362.05s
500	3020.15s	2010.77s



Using adaptive learning rate resulted in improved training times for all the hidden layer configurations.

- **Mean Squared Error at Convergence:** The following table represents the MSE at convergence for different number of neurons in the hidden layer for adaptive learning rate.

Neurons in Hidden Layer	MSE at convergence
1	1.25
10	0.11
50	0.052
100	0.050
500	0.058



The MSE values are more or less similar as compared to the constant learning rate training.

Task - 2.c: ReLU Activation

It was observed that using the learning rate of 0.001, for ReLU didn't converge, therefore the learning rate for constant case was changed to 0.00001. The following table indicates the differences between the *ReLU* activation and the *sigmoid* activations for 2 layered architecture with 100 neurons in each hidden layer. The weights are now initialized by sampling from a normal distribution with 0 mean and 0.01 variance. The stopping criteria is the same however, the threshold has been changed to 1e-04 with only last 5

epochs considered in place of earlier 10. The seed value is set at 1e-03.

Metric	Sigmoid	ReLU
Accuracy on Train Set	0.933	0.991
Accuracy on Test Set	0.869	0.93.9
Training Time	333.61	90.14s
MSE at convergence	0.046	0.0002

The ReLU activation function is better in all the metrics as compared to the sigmoid activation function. Following are the results obtained on using ReLU for single hidden layer.

The accuracy results for this architecture [**100,100**] is **better** than the results with the single layer architectures [**1,10,50,100,500**] as carried out in 2b, as well. The training time is significantly due to the computationally simple max function as compared to the exponentiation. The simple nature of the function also accelerates the training process.

Task - 2.d: MLPClassifier

The following table summarizes the comparison between the *MLPClassifier* from *scikit-learn* and *my own implementation*.

Metric	MLPClassifier	my implementation
Accuracy on Train Set	1.0	0.991
Accuracy on Test Set	0.915	0.936
Training Time	173.6s	58.59s

The differences in the test accuracy would mainly be due to the different initializations of the weights. From the results it can be concluded that MSE is a better loss function as compared to Cross Entropy for this dataset.