

Stats 202A

Final Project

Devanshi Patel (UID: 504945601)

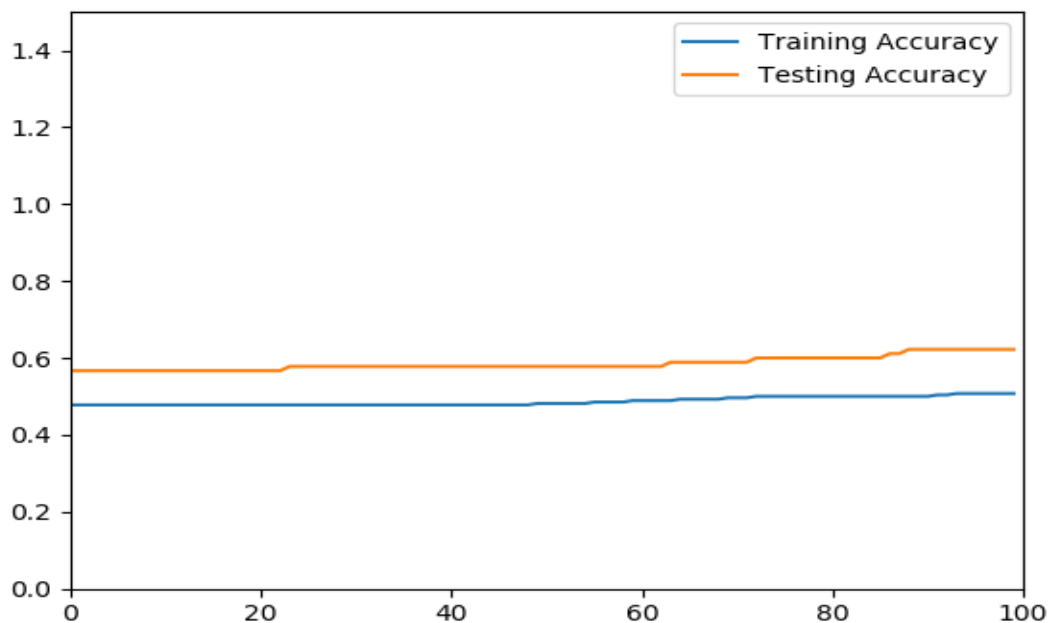
Neural Networks

The following plots are obtained when we plot Training accuracy and Testing accuracy for digits data with different values of learning rate and number of iterations.

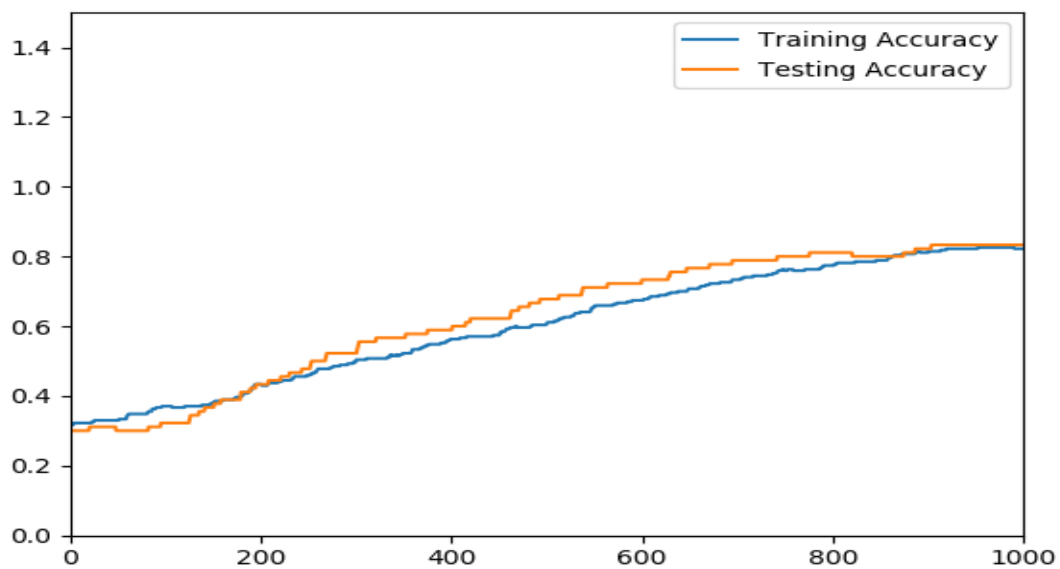
It can be seen from the plots that as we increase the number of iterations, testing accuracy tries to reach closer to training accuracy. From the below plot, it can be seen that for a low value of learning rate and lesser iterations, there is a big enough difference in the accuracies. But as we adjust the learning rate, the convergence happens faster.

In the last plot, it can be seen that the difference between training and testing accuracy is negligible.

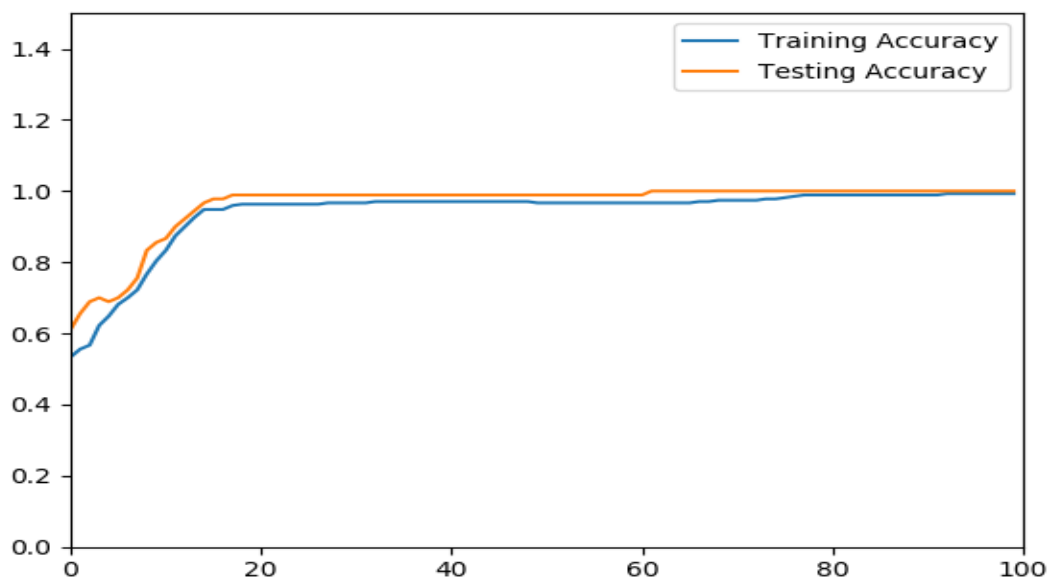
Learning rate: 0.001, Number of iterations: 100



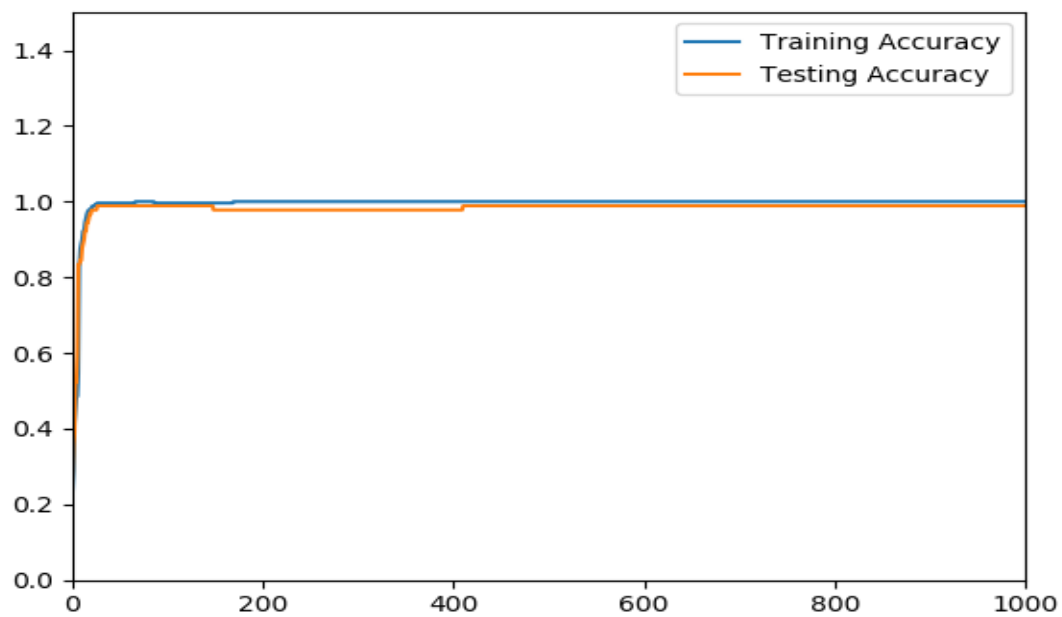
Learning rate: 0.001, Number of iterations: 1000



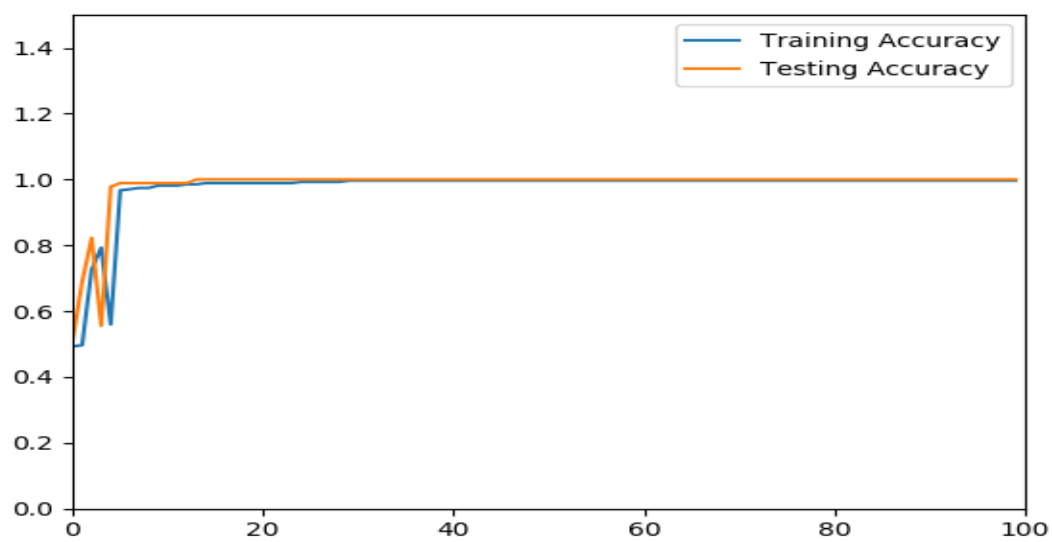
Learning rate: 0.1, Number of iterations: 100



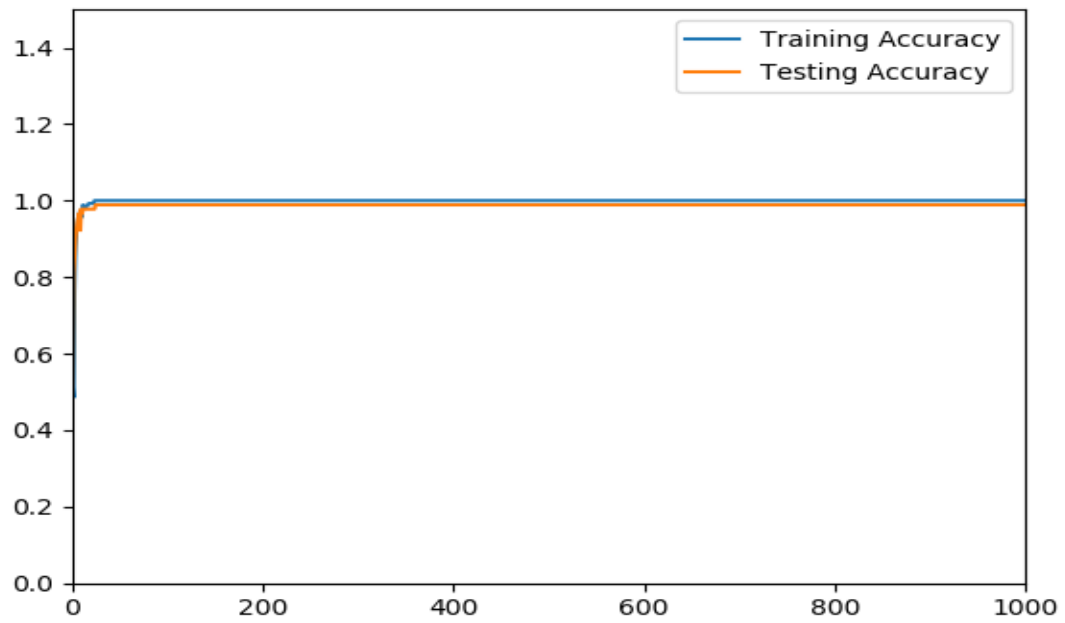
Learning rate: 0.1, Number of iterations: 1000



Learning rate: 0.5, Number of iterations: 100



Learning rate: 0.5, Number of iterations: 1000

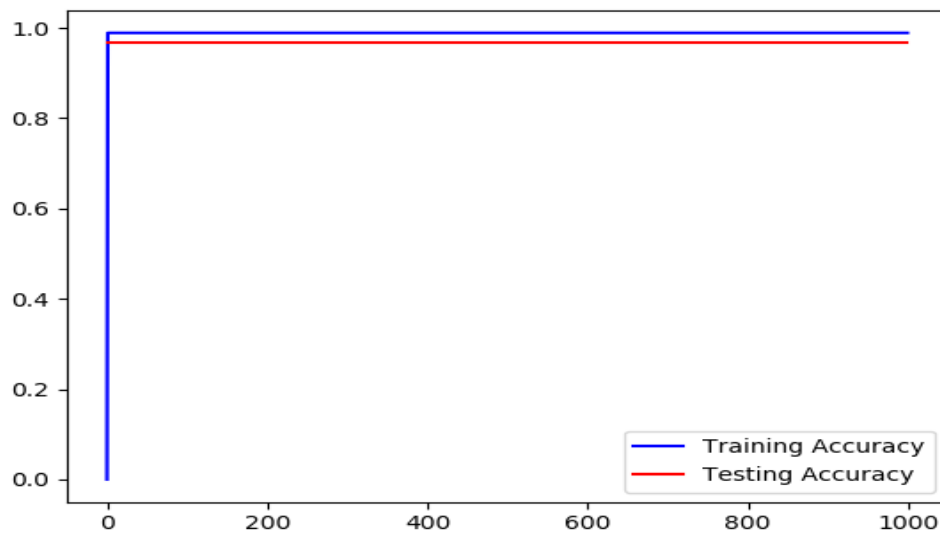
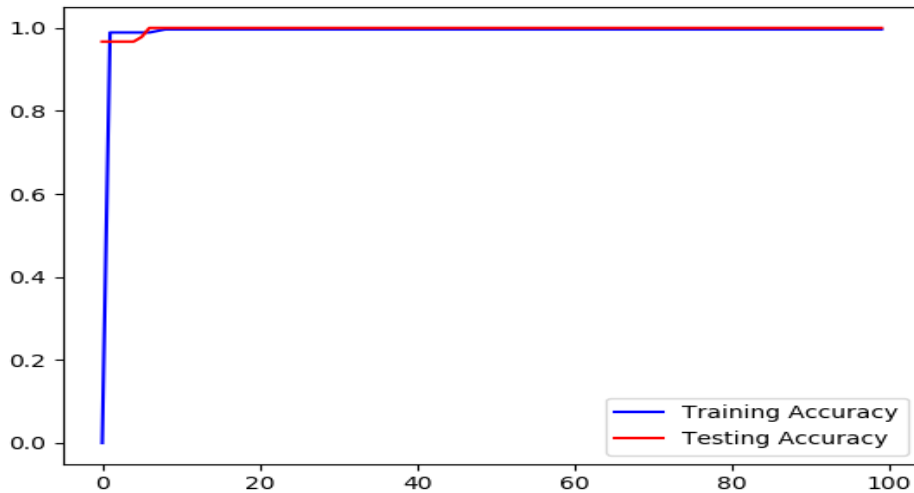


SVM

The first plot is obtained when we plot Training accuracy and Testing accuracy for digits data with lambda as 0.01, learning rate as 0.1 and 100 iterations.

The second plot is obtained when we plot Training accuracy and Testing accuracy with lambda as 0.1, learning rate as 0.01 and 1000 iterations.

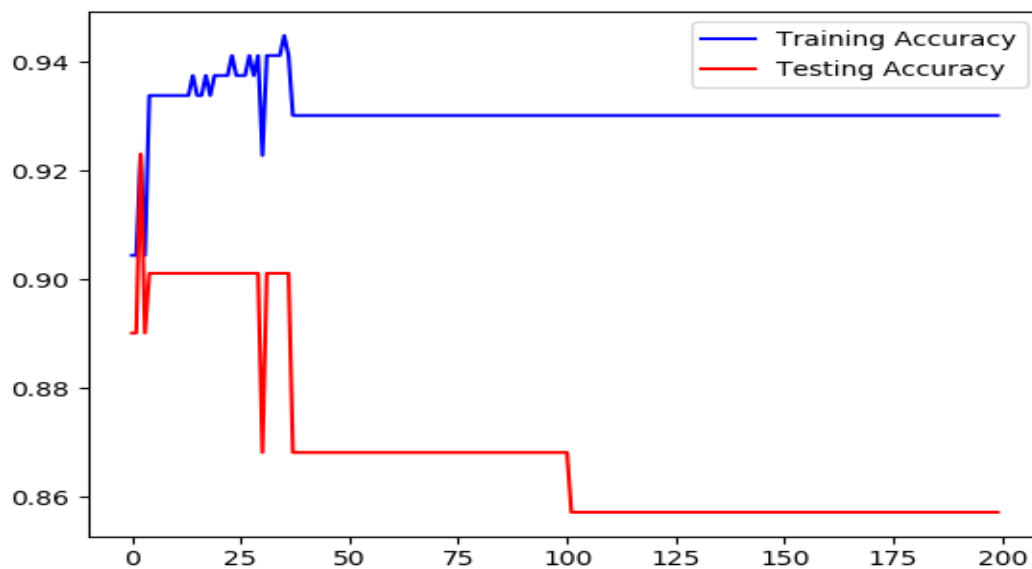
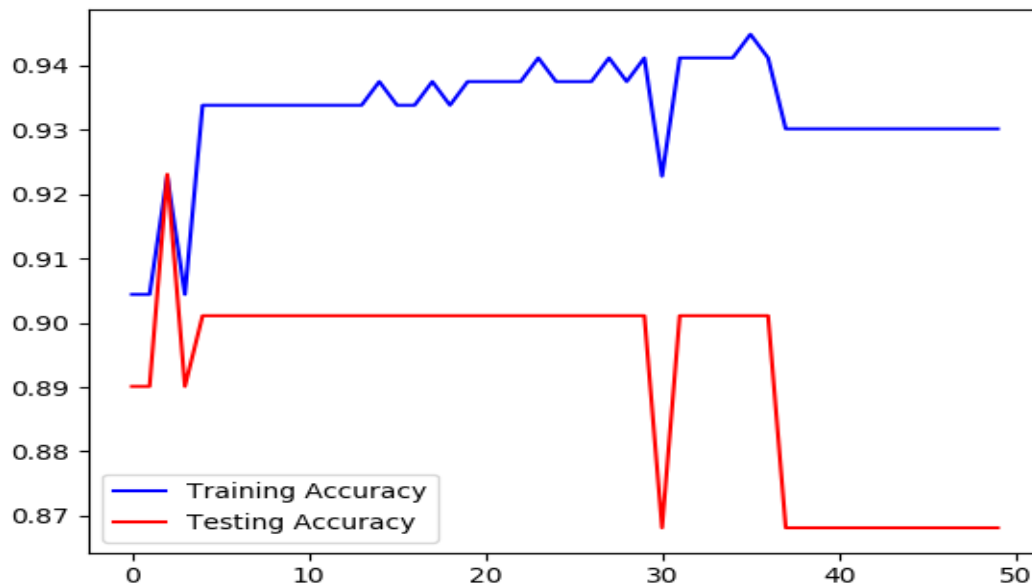
Thus, from the plots it can be said that as we increase regularization parameter and decrease the learning rate, the difference in training and testing accuracy increases slightly.



Adaboost

The following plots are obtained when we plot Training accuracy and Testing accuracy for digits data with 50 and 200 iterations respectively.

As seen from the plot, there is a spike in Testing accuracy in the initial iterations. Also, there is a big enough margin in the training and testing accuracy which proves that Adaboost never overfits.



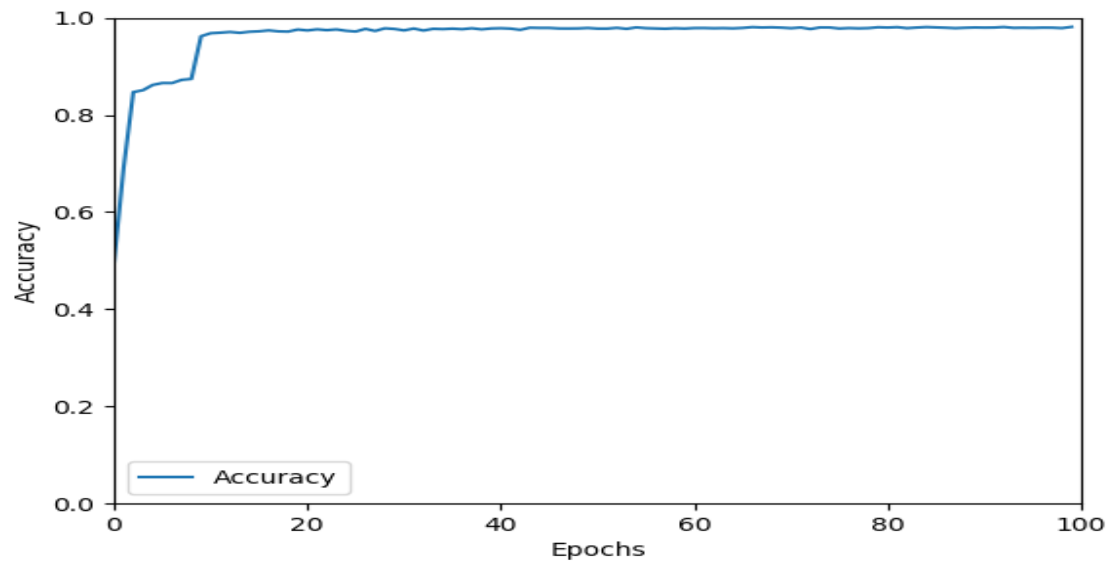
Comparison of Neural Networks, SVM and Adaboost:

	Neural Networks	SVM	Adaboost
Pros	<ul style="list-style-type: none">-Extremely fast and flexible at learning important features from basically any data structure, without having to manually derive features.	<ul style="list-style-type: none">-Needs fewer hyperparameters to get a reasonably accurate model.-Global optimum guaranteed.-The results are more interpretable.	<ul style="list-style-type: none">-Easily interpretable strong classifier formed by linearly combining weak classifiers-Works really well for small size datasets
Cons	<ul style="list-style-type: none">-Don't provide a regularization parameter to avoid overfitting.-Due to the randomness of the learning algorithm, the results are not reproducible.-Hard to determine relevance of features	<ul style="list-style-type: none">-Choosing the kernel and regularisation parameters can often lead to severe overfitting if we over-optimize the model selection criterion	<ul style="list-style-type: none">-The weights are usually predetermined while in NN, they can be optimized.-Unlike NN, adaboost doesn't work well if we want to learn complex structures.

TensorFlow

When we use 2 layer Neural Network with Relu and Softmax as the activation functions, the final accuracy of the code on MNIST data is about 98%.

When we plot accuracy against the number of iterations, the following plot is obtained. As seen from it, the accuracy is low in the initial iterations but as the model gets trained, it improves and finally converges around 98%.



When we plot the values of loss function against the number of iterations, it can be seen from the figure that the loss function is high initially. As the backpropagation algorithm adjusts the values of variables, the loss gradually decreases.

