CSE 574
Fall 2016
University at Buffalo
Project 3
Classification

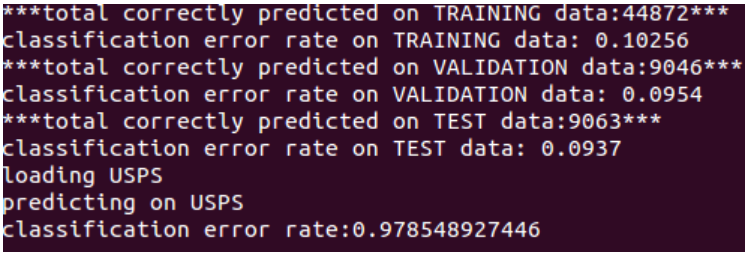
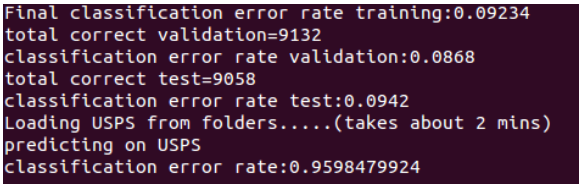
11 December 2016

Submitted by,
Jayaraj Sajjanar
50208475

Introduction

Project was to classify MNIST hand written digits using logistic regression, neural network and Convolutional neural network. The project was implemented in python.

Results

		Classification Eror Rate	Screenshot
Logistic Regression	Train	0.10256	
	Validation	0.0954	
	Test	0.0937	
	USPS		
Neural Network	Train	0.09234	
	Validation	0.0868	
	Test	0.0942	
	USPS		
CNN	Train		
	Validation	0.034	Accuracy – 96.56
	Test	0.037	Accuracy – 96.21
	USPS		

Description

For logistic regression, the code was written in python. Bias was set as an array of 1's. Weights were set as an array of 0s. About 100 epochs were run and the iterations were stopped when highest accuracy was achieved.

Similarly for neural network. Bias was set as 1's. Weights were randomly set. Multiple iterations were run till an acceptable accuracy was achieved on training. This happened to be that 88% accuracy was achieved in 5 mins i.e. 35 iterations. And took about 15 mins i.e. 103 iterations to achieve 90%

accuracy.

Screen shots have been attached for the same.

USPS data was imported from the folders. The data was reshaped to 20*20. The data was normalised by subtracting 255 and dividing 255. And the padding was done. Even then abysmal prediction results were obtained.

Could not install tensorflow on local machine. So I installed it on aws. I received an resource exhausted error. So had to use python 3.5 on friends system.

I followed the tutorial on
https://www.tensorflow.org/versions/r0.12/tutorials/mnist/pros/index.html#build-a-multilayer-convolutional-network for doing convolutional neural network.

Hyper parameter tuning

Hyper paramters like bias and weights and number of epochs and learning rate were experimented on. Optimum values were chosen after observing the results. For example, when the weights were set randomly, the accuracy was better. And when learning rate was set higher, the convergence was faster.

In neural network w1 was set as random values of shape (784*784) and w2 as 10*784.
i.e. a network was built of hidden layer of length 784.

Conclusion

The results and observations complied the no free lunch theorem. The weights that were obtained on MNIST were applied to the USPS and the prediction results were abysmal.