University at Buffalo
The State University of New York

INTRODUCTION TO MACHINE LEARNING
CSE 574

FALL-2016

PROJECT-3 REPORT

SUBMITTED BY:

ARUN CHANDRA PENDYALA(50207136)

CLASSIFICATION

INTRODUCTION:

The aim of this project is to implement classification algorithms and evaluate the models – logistic regression, single hidden layer neural network(NN) and convolutional neural network(CNN). The tasks involved were to recognize a 28 x 28 grayscale handwritten digit image and identify it as one among the set of 10 digits ranging from 0 to 9. The project also involved testing the MNIST trained models on the provided USPS dataset.

DATASETS:

MNIST DATA

It is a large collection of handwritten digits used widely for training and testing in the fields of machine learning and image processing. The dataset was of the overall size – 70,000 x 784 after it was imported into the python code where D = 784 corresponds to 28x28 feature set of the digits. For this project , the dataset was divided as follows:

i) Training dataset – 50,000 x 784
ii) Validation dataset – 10,000 x 784
iii) Testing dataset – 10,000 x 784

The target value set is of the size 70,000 x 1 corresponding to the labels associated with each digit.

USPS DATA:

The USPS datasets consists of images of digits in folders marked from 0 to 9. The dataset is processed and loaded using the Python imaging library(PIL). There are totally 2000 images in the collection with 200 images in each digit folder. This dataset is used for testing the performance of MNIST-trained dataset on it. This test dataset is of the size 19999 x 784.

IMPLEMENTATION:

LOGISTIC REGRESSION:

The concept of multiclass logistic regression
Stochastic gradient descent has been used for the updation of weights
1 of K coding scheme is used to represent the target value t into a vector of length K. The various calculations performed are as shown below:

$$p(\mathcal{C}_k|\mathbf{x}) = y_k(\mathbf{x}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \qquad a_k = \mathbf{w}_k^\top \mathbf{x} + b_k$$

$$\nabla_{\mathbf{w}_j} E(\mathbf{x}) = (y_j - t_j)\,\mathbf{x} \qquad \mathbf{w}_j^{t+1} = \mathbf{w}_j^t - \eta \nabla_{\mathbf{w}_j} E(\mathbf{x})$$

The updation of weights is done for multiple iterations for better performance of the system. Instead of using vectorized form, the weights were updated iteratively by taking datapoints one at time. Classification of digits is done by:

$$C = \arg\max_i y_i.$$

SINGLE HIDDEN LAYER NEURAL NETWORK:

This is the simplest form of neural network with only layer of hidden nodes between input and output layers. The logistic sigmoid function has been used as the activation function in this project. Stochastic gradient descent has been used for the updation of weights similar to that of logistic regression. Even in this, 1 of K coding scheme is used to represent the target value t into a vector of length K and updation of weights is done for multiple iterations for better convergence. Instead of using vectorized form, the weights were updated iteratively by taking datapoints one at time.The calculations are performed as shown below:

$$z_j = h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + b_j^{(1)}\right)$$

$$\delta_k = y_k - t_k$$

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + b_k^{(2)}$$

$$\delta_j = h'(z_j) \sum_{k=1}^{K} w_{kj} \delta_k$$

$$y_k = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

$$\frac{\partial E}{\partial w_{ji}^{(1)}} = \delta_j x_i, \qquad \frac{\partial E}{\partial w_{kj}^{(2)}} = \delta_k z_j \qquad \mathbf{w}^{t+1} = \mathbf{w}^t - \eta \nabla_{\mathbf{w}} E(\mathbf{x})$$

CONVOLUTIONAL NEURAL NETWORKS:

Two layers of neural networks – layer of convolutional subunits(which consider the overlapping regions) and subsampling units are used in the convolutional neural networks instead of just fully connected neural networks layer subunits.The network has several feature maps and submapping. The final layer has softmax output. The whole network is trained using backpropagation.
The tensorflow module is used for convolutional neural networks, which is an interface for expressing machine learning algorithms and an implementation for executing them.

RESULTS:

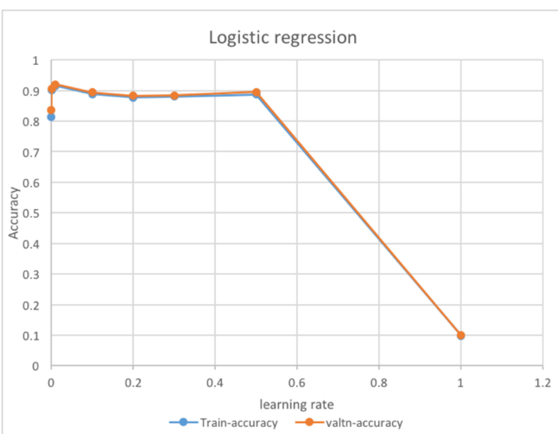
LOGISTIC REGRESSION:

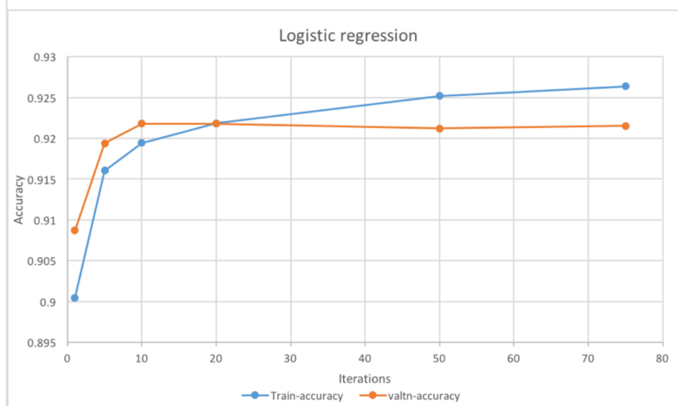
Tuning Learning rate and number of iterations:

The learning rate was varied from 0.0001 to 1 and it was then observed that the training and validation accuracies were good at learning rate = 0.001. At learning rate = 0.001 , the number of iterations were varied and it was observed at number of iterations = 10 , the accuracies for training and validation datasets were good and the computation time was also reasonable.

LOGISTIC REGRESSION		
	5 iterations	
Learning rate	Accuracy	
	Training	validation
0.0001	0.8146	0.8356
0.001	0.90164	0.9058
0.01	0.91526	0.9196
0.1	0.8881	0.8946
0.2	0.87704	0.8825
0.3	0.88084	0.8846
0.5	0.88704	0.8953
1	0.09864	0.0991

LOGISTIC REGRESSION		
	eta = 0.01	
Iterations	Accuracy	
	Training	Validation
1	0.90048	0.9087
5	0.91606	0.9194
10	0.91944	0.9218
20	0.92182	0.9218
50	0.9252	0.9212
75	0.92638	0.9215





SINGLE HIDDEN LAYER NEURAL NETWORK:

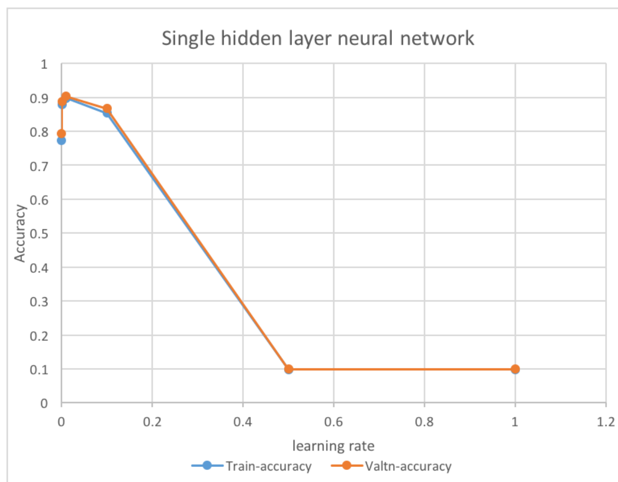
Tuning learning rate , number of iterations and number of hidden layer nodes:

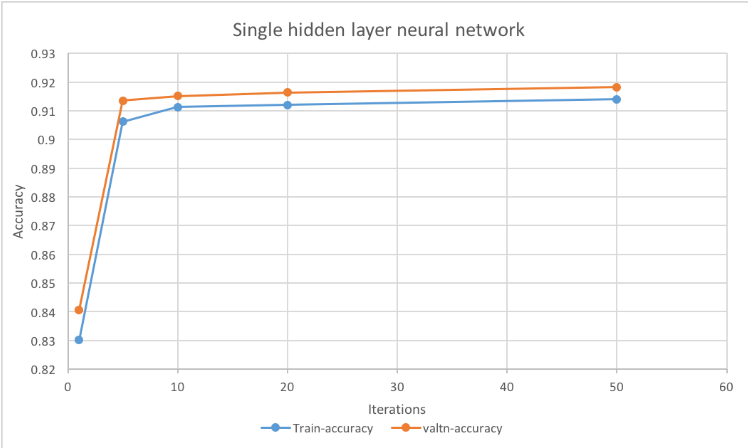
Just like in the previous case of logistic regression, the learning rate was varied from 0.0001 to 1 , it was observed that the performance was very good relatively at $\eta = 0.01$. At this learning rate, the number of iterations was varied. It was the performance got better with the increase in the number of iterations but it was computationally time consuming and the performance didn't improve significantly beyond number of iterations = 20. So it was set to 20 and the number of nodes in the hidden layer was then varied.The number of hidden nodes was hen chosen as 100.

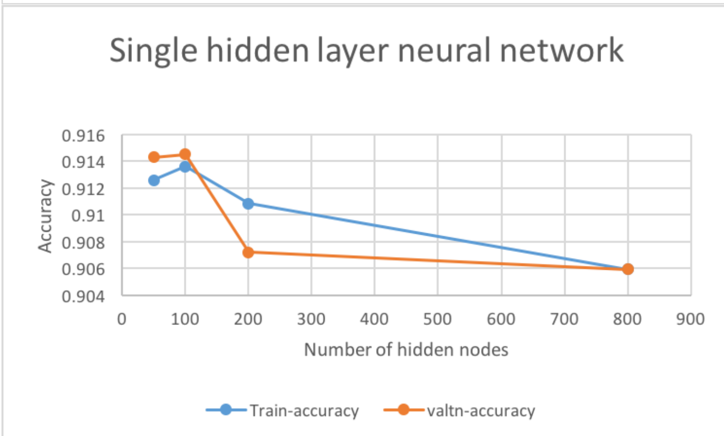
Neural network		
	5 iterations, 50 hidden nodes	
Learning rate	Accuracy	
	Training	validation
0.0001	0.77284	0.7937
0.001	0.8789	0.8887
0.01	0.89716	0.903
0.1	0.85372	0.8671
0.5	0.09864	0.0991
1	0.09864	0.0991

Neural Network		
	eta = 0.01	
Iterations	Accuracy	
	Training	Validation
1	0.8301	0.8407
5	0.90618	0.9135
10	0.9113	0.9151
20	0.91202	0.9164
50	0.91404	0.9182

Neural Network		
	eta=0.01 and 20 iterations	
Number of hidden nodes	Accuracy	
	Training	Validation
50	0.9126	0.9143
100	0.9136	0.9145
200	0.91084	0.9072
800	0.90592	0.90592







CONVOLUTIONAL NEURAL NETWORK:

The following results were obtained for CNN:
1) 1000 steps:
step 0, training accuracy 0.02
step 100, training accuracy 0.82
step 200, training accuracy 0.94
step 300, training accuracy 0.9
step 400, training accuracy 0.98
step 500, training accuracy 0.94
step 600, training accuracy 1
step 700, training accuracy 0.96
step 800, training accuracy 0.88
step 900, training accuracy 1
test accuracy 0.9641
test accuracy for USPS data 0.487624
2)5000 steps:
step 4000, training accuracy 1
step 4100, training accuracy 0.98
step 4200, training accuracy 0.98
step 4300, training accuracy 1
step 4400, training accuracy 0.98

step 4500, training accuracy 1
step 4600, training accuracy 0.96
step 4700, training accuracy 1
step 4800, training accuracy 1
step 4900, training accuracy 0.98
test accuracy 0.9869
test accuracy for USPS data 0.567978
The accuracies can be increased further by increasing the number of steps at the expense of high computational load. The computation time for 5000 steps was itself very high.

USPS DATA AND FREE LUNCH THEOREM :

In the context of machine learning, the free lunch theorem states that there is no one model that works best for every problem. The assumptions of a great model for one problem may not hold for another problem, so it is common in machine learning to try multiple models and find one that works best for a particular problem.
A model is a simplified representation of reality, and the simplifications are made to discard unnecessary detail and allow us to focus on the aspect of reality that we want to understand. These simplifications are grounded on assumptions; these assumptions may hold in some situations, but may not hold in other situations. This implies that a model that explains a certain situation well may fail in another situation.
SOURCE: https://chemicalstatistician.wordpress.com/2014/01/24/machine-learning-lesson-of-the-day-the-no-free-lunch-theorem/

The accuracies observed for the following models:
1) Logistic regression – 0.319999999
2) Single Hidden layer neural network – 0.3668
3) Convoltuional neural network - 0.487624
The values observed are quite low. This suggests that the free lunch theorem holds true for this scenario.

RESULTS:

############ 1. logistic regression ###########
Accuracy- training set
0.92034
Accuracy - validation set
0.9232
Accuracy - testing set
0.9132
Accuracy - testing USPS data
0.319999999
############ 2.single layer neural network ###########
Accuracy- training set
0.9136
Accuracy - validation set
0.9145
Accuracy - testing set
0.9145

Accuracy - testing USPS data
0.3668
############ 3.convolutional neural network ###########
step 4900, training accuracy 0.98
test accuracy 0.9869
test accuracy for USPS data 0.567978

CONCLUSION:

The following results have been obtained in this project:

	Training accuracy	Validation Accuracy	Testing Accuracy
Logistic regression	0.92034	0.9232	0.9132
Single hidden layer Neural Network	0.9136	0.9145	0.9083
Convolutional neural network	0.98	-	0.9869

As expected, the convolutional neural network showed the best results for both the datasets – MNIST and USPS datasets. But it was very time consuming as it involved computations in multiple layers.