Machine Learning

Project 3

Report

Puneeth Pepalla

50206906

Puneethp

Aim: To implement and evaluate classification algorithms. The classification task is to recognize a 28⇥28 grayscale handwritten digit image and identify it as a digit among 0 to 9.

TASKS INVOLVED

1. **Implementing logistic regression, train it on the MNIST digit images and tune hyperparameters.**

2. **Implement single hidden layer neural network, train it on the MNIST digit images and tune hyperparameters.**

3. **Use a publicly available convolutional neural network package, train it on the MNIST digit images and tune hyperparameters**

4. Test your MNIST trained models on USPS test data and compare the performance with that of the MNIST data.

TASK-I: Logistic Regresssion:

Step:1- Extracting feature values and labels is done using a library called cpickle.

Step-2: Since it is already partitioned we use the training data set to train the model.

Step-3: Training model parameters.

First Training data set consists of a tuple. The first one is the training features and the second one is the training labels. This training set consists of 50000 examples.

So we split the tuple and extract the features and arrays into two different numpy arrays. And bias vector can be added right here by using numpy.insert

Step-4: Testing MNIST trained models on USPS test data and compare the performance with that of the MNIST data.

One Hot Representation:

This is one of the important steps of the project in which we take the array that consists of the labels and represent them with 0's and 1's (can be called binary form). This is done using a simple logic.

Example:3 can be represented as 0001000000, 5 can be represented as 0000010000, 9 as 0000000001

Now we consider the weight matrix and fill it up with random values. Since there are 785 features and 10 labels, the size of the matrix would be 785*10. This is done using np.rand.random().

Now according to the standard hypothesis equation, we multiply the weights matrix and features matrix

By using numpy.dot

$$p\left(\mathcal{C}_k|\mathbf{x}\right) = y_k\left(\mathbf{x}\right) = \frac{\exp\left(a_k\right)}{\sum_j \exp\left(a_j\right)}$$

Now using softmax function we have to calculate the probability as above for each iteration.

The weights should be updated for a certain number of iterations until all the parameters are tuned perfectly.

After calculating the dot product and bias vector, the probabilities have to be calculated.

The weights updation is done by applying derivative to the above formula.

$$E\left(\mathbf{x}\right) = -\sum_{k=1}^{K} t_k \ln y_k$$

Here we use learning rate eta(n) for updating weights. We set this value of eta to 0.5 and modify it according to the change in weights and finally choose the optimal value for it.

Now for each iteration the formula for updating weights can be represented as follows:

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

Where n is the learning rate.

Now recording all the values obtained for weights and calculating the accuracy of the model

First we fix a value for the learning rate and test the model on training data set, validation data set and test data set.

If we obtain more value of accuracy(100%) for training data then that might lead to overfitting of data

So, we choose the optimum value for the value of learning rate for which the accuracy is highest.

Tablar representation for the values of learning rate and Efficiency

Learning rate (training set)	Efficiency
0.5	83
0.3	88
0.65	91
0.4	86
0.8	89

These are the obtained values of accuracy for different values of eta(learning rate) for training data set in logistic Regression.

Learning rate (validation set)	Efficiency
0.5	**81**
0.3	**87**
0.7	**92**
0.9	**83**

These are the obtained values of accuracy for different values of eta(learning rate) for validation data set in logistic Regression.

Learning rate (Test set)	Efficiency
0.5	**83**
0.3	**84**
0.6	**92**

0.9	**80**

These are the obtained values of accuracy for different values of eta(learning rate) for test data set in logistic Regression.

Single Layered Neural Networks:

In Single neural networks we take one input layer, one hidden layer and one output layer.

We use two different set of weights and update them **simultaneously.**

We iterate for every value of training data, that is number of iterations are 50000

Each node in the input layer receives one features of each input, multiplies with the first set of weights and outputs the value to 1000 nodes of the hidden layer.

The size if the input layer is 784*M where M is the number of nodes in the hidden layer and 784 is the number of features.

In this case let us assume the number of nodes in the hidden layer are 1000.

Now each node of the hidden layer takes 784 inputs and outputs 10 values to the output layer since the number of nodes in the output layer are 10(0-9).

The size of first weight matrix is 784*1000

The size of the second weight matrix is 1000*10.

Now for each iteration we update both the weight matrices simultaneously until they are tuned.

After iterating through all the data, we calculate the final values of weights and calculate the corresponding accuracy(testing the model on training, validation and test data set)

The efficiency of the model is obtained.

The equations involved in Single Layered Neural Networks:

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

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

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

The values of efficiencies obtained for different values of eta(Learning Rate)

Learning rate (Training set)	**Efficiency**
0.03	**84**
0.04	**89**
0.07	**91**
0.09	**83**

These are the obtained values of accuracy for different values of eta(learning rate) for training data set in Single Neural Networks.

Learning rate (validation set)	**Efficiency**
0.05	**81**
0.03	**88**
0.07	**90**
0.09	**83**

These are the obtained values of accuracy for different values of eta(learning rate) for validation data set in Single Neural Networks.

Learning rate (test set)	**Efficiency**
0.05	**81**
0.03	**85**
0.067	**93**

0.09	**89**

These are the obtained values of accuracy for different values of eta(learning rate) for test data set in Single Neural Networks.

Convolution Neural Networks:

I have used tensorflow interface for implementation of cnn. This is done using a virtual machine.