***Report***

Name : Ganesh Talwar

Admission number:22je0344

Gmail: 22je0344@iit.ism.ac.in , [ganeshtalwar.dcmc@gmail.com](mailto:ganeshtalwar.dcmc@gmail.com)

Contact No.: 9464278910

Project: ML Bootcamp

Branch: Computer Science and Engineering

***Project definition*:** In this project I have to make python library inwhich I have all the functions for linear regression , logistic regression model , neural network and k nearest neighbour algorithm . in my project the name of my library mlal1. Also in this project I have used only numpy , pandas ,jupyter notebooks.

* **Linear regression model:**

In the linear regression model I made some functions in this model that are

Cost\_function\_linear : in order to compute the cost of the model

z\_score\_normalisation(to do feature scaling )

gradient\_descent\_calculate\_linear(in order to calculate the dw,db)

gradient\_linear( in to update my parameters)

output\_label( in order to find the output of by applying final w,b on test data)

then I initialized w as a array of zeroes and b as 0,then I tried gradient descent on it with iterations equal to 10000 and different alphas and tried to reduce the cost as much possible

Alpha iterations Cost

1 10000 5.53245078e+107

1 1000 9.46217322e+13

0.9 10000 4714.07834306

0.9 1000 4714.07834306

0.8 10000 4714.07834306

0.8 1000 4714.07834306

0.5 10000 4719.07834306

0.5 10000 4719.07834306

This shows that the cost converges at the above value so even if we change alpha cost is not changing much so minimum cost will be 4714.07834306

**Polynomial regression model**

In the polynomial regression model I made some functions in this model that are

Cost\_function\_linear : in order to compute the cost of the model

z\_score\_normalisation(to do feature scaling )

gradient\_descent\_calculate\_linear(in order to calculate the dw,db)

gradient\_linear( in to update my parameters)

In this model I made a n degree polynomial so I used another function

power\_function( to declare the value of n)

output\_label( in order to find the output of by applying final w,b on test data)

in this I tried different values of degree and different values of alpha after doing this I fixed the value of number of iteration equal to 1000 and alpha as 0.02 and then tried till 6th degree model to reduce cost to minimum. Then I initialized w as array of zeroes and b as 0 and then applied gradient descent on the model on it to find final w,b and tested on test data.

Cost R2score

degree 2: 2.52e+12 0.1628

degree 3: 1.18e+12 0.3887

degree 4: 1.32e+12 0.4856

degree 5: 8.41e+12 0.674

degree 6: 6.67e+12 0.743

after this analysis I decided that the 6th degree suits the data best.

* ***K-nearest algorithm***

In the k nearest algorithm our task was to predict value of the digit by seeing the images . images of data was given in terms of pixel I have to use a KNN algorithm in order to find the most correct digits. KNN deends on fact that similar things appear at same place in graph so for this I used functions:

cal\_eucleadian\_distance(in order to calculate the distance between any two points)

cal\_pred(sorting the values of distances find which label is coming maximum number of times in top k rows of that sorted array)

* ***logistic regression***

In this we were asked to predict value of the digit by seeing the images and image pixels were given in the dataset. In this I used the one vs all model in this particular case I made 10 model and then I did the one hot encoding of the training data . then I train the model with sigmoid and gradient descent algorithm then I initialized the value of w,b as the random values .

I tried different values of alpha on each model so that the cost of all the model is reduced as much possible in this I used the following functions

cost\_function\_logistic( to find the cost when I trained the model )

gradient\_descent\_calculate\_logistic( to find the change in the value of w,b )

gradient\_logistic( to run the gradient descent and update the parameters w,b)

sigmoid( in order to calculate the sigmoid of a function)

after trying I fixed the values of iterations =100 and 1alphas will be after trying over the 10 models I got an accuracy of 60 percent.

* ***Neural network***

With 1 hidden layer:

In this we were asked to predict value of the digit by seeing the images and image pixels were given in the dataset. In this I used the one vs all model in this particular case I made 10 model and then I did the one hot encoding of the training data. After completing my training set I then made a class and applied forward( in this I used numpy.tanh() for hidden layer and softmax function for the output layer) and backward propagation. After doing his I applied gradient descent on it with iterations equal to15 and alpha as 0.001 but accuracy came only 60% but then I instead of initializing w as np.random.rand I initialized it as 0.001\*np.random.rand , iterations as 15 , instead of np.tanh() I used the relu function and the accuracy became 69% Then I predicted on the test dataset.

In this I used functions:

\_\_init\_\_(constructor to initialize w1,b1,w2,b2)

Forward

Backward

Train

Predict

With n layer:

In this I used the one vs all model in this particular case I made 10 model and then I did the one hot encoding of the training data. After completing my training set I then made a class and applied forward( in this I used numpy.tanh() for hidden layer and softmax function for the output layer) and backward propagation. After doing his I applied gradient descent on it with iterations equal to15 and alpha as 0.001 but in my code some error is coming in back propagation it works for 1 value of w then again start giving error.

In this I used functions:

\_\_init\_\_(constructor to initialize w1,b1,w2,b2)

Forward

Backward

Train

Predict