ML BOOTCAMP

—

Vivek Patidar

20JE1098

Department : Mathematics and Computing

What i learned:

This whole project is really a starting point in my journey towards my newly developed hobby: machine learning.

Before this project even begins, I barely know any language but I am so fascinating about the machine learning that before coming to this college I know what I have to pursue this amazing branch of computer science as a result i land up to learn Machine learning

When I get to know about the cyberlabs ML department I take it as a good chance to make myself more familiar with the ML.

but for this project I have to learn from very basic python as i was untouched by any kind of coding or programming i started all this in the vacations i get after the JEE advance and then i started learning python, due to less time I did my best to make myself more comfortable with the python.

But throughout this amazing journey I enjoyed every part of it.

Speds almost 5 to 6 hour a day on my laptop trying to learn and implement the algorithm with all my dedication i do make a hard work which i myself can see in me as before this woc begins i just started my programming journey but even in this stage i grabs decent knowledge which i think is a good startup for me.

And I wanna continue my newly developed hobby and keep going in this field, I also have a few projects in my mind in which machine learning can be a right way to achieve that.

Goals

1. Implement Linear Regression on MNIST dataset
2. Implement Logistic Regression on MNIST dataset
3. Implement KNN on MNIST dataset
4. Implement 2- layer neural network
5. Implement n- layer neural network

Linear Regression

I used this algorithm and made my model to train on the 19999 dataset and do the prediction on the 9999 dataset which are already divided into test and train data labels .

Then I used an iteration loop which then made the linear regression model and updated the parameters according to the gradient descent method which we were taught by the andrew ng course.

And I used numpy, matplotlib and pandas in each and every algorithm as basic libraries to do certain tasks.

Logistic Regression

In this i used the same training and testing data but this time i divide the dataset into a way so that i can run a one vs all classifier for each digit whose image is given in that 19999 data points with 784 pixelsI have divided this data into a 0 for not the particular digit and 1 for the digit.

And then I tested my algorithm on the test data from where I have calculated the accuracy which comes out to be around 90%.

KNN

This was the easiest algorithm for me. In this i do not need to train my model for every training data points but rather i just have to test on the test dataset to the model and find the nearest neighbours with k = 3 , i used this value of k after testing my model on different values of k which then ters out that the value 3 for k results in a good amount of accuracy compared to all others, i also mentioned it in the sklearn version also,

So i used the 3 most nearest neighbours and found which is the most occured digit in them and assigned that test data point to be that only labels and repeat this process for all the 9999 times.

2 layered neural network

The architecture of my model is that I have 784 elements in the input layer and I take 100 and 50 in the hidden layers respectively and the output is of 10 units which gives the maximum value to the label of that particular testing sample.

784 >> 100 >> 50 >> 10

Then I make many functions like the forward propagation , back propagation, updating the parameters and all other kinda functions which i used in my main loop.

Then while updating the previous parameters i used the gradient descent which i learned from the andrew ng course.

N layer neural network

Similar to the 2 layers in one i also used the same backpropagation and forward propagation algorithm in these also but used vector format to store all my parameters and all architecture whose defining matrix is supplied by the user of n layers.

The number of layers in the model as well as the length of each layer is also specified by the user.

And then after uses the same gradient descent algorithm on that also to update the parameters after the iterations by using the vectors.

But i could not finish this algorithm.

Accuracies

	My model	With sklearn
Linear regression	9.32%	25.16%
Logistic regression	90.64%	97.40%
KNN	96.10%	94.62%
2 layer Neural Network	10%	94.77%

Reason for the different accuracy:

Linear regression: may be due to overtraining of the model

Logistic regression: it may be due to malfunctioning for calculating the accuracy as all other things are running well.

Knn: it already achieved the sklearn mark maybe the difference is because of, I am missing some code for the sklearn part.

Neural network: It is due to not taking the batch gradient descent which makes my code slow and not properly using the vector forms as i am learning it but not able to implement it fully in my code.

Where to find my code

I have uploaded all of my code on the github account whose link was already shared by me

In that github account in the private repository named ML_bootcaamp_vivek i used its branches as a room for each algorithm but as i uploaded some more code in it which do not belongs to the specific code and there is also some previous version also there.

But I make it to get myself a better understanding of that particular algorithm by starting the usage of the algorithm on my own small data and then try it on the bigger dataset like MNIST.

So i am mentioning the final file name in this :

1. Linear regression: FINAL_LinearRegression.ipynb
2. Logistic regression: FINAL_LogisticRegression.ipynb,,LogisticRegression_using_sklearn
3. KNN: FINAL_KNN.ipynb and sklearn_KNN.ipynb
4. 2 layered neural network : FINAL_neural_network.ipynb
5. N layered neural network : Draft_Neural_network_n_layer.ipynb