Name: Krishna Chaitanya Naragam

MAV ID: 1001836274

Assignment 2

Logistic Regression

In statistics, the logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1, with a sum of one. [Wikipedia]

The task in problem 1 is to use the given multi-variate normal distribution parameters to generate train and test data for two classes and use logistic regression to create a model that fits the produced data.

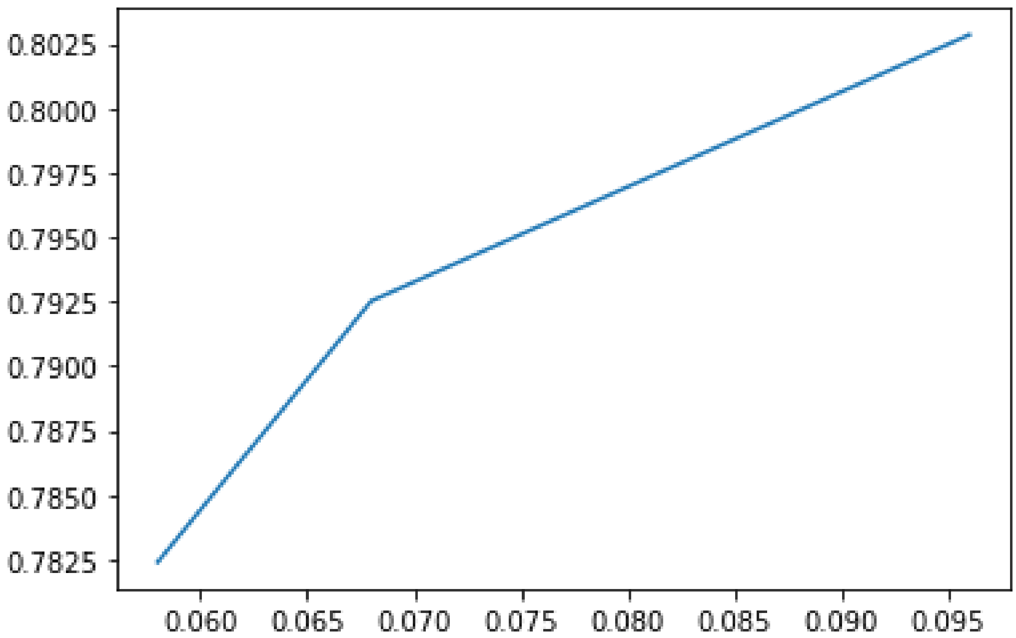
Gradient Descent

Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. To find a local minimum of a function using gradient descent, we take steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point. [Wikipedia]

The task was to learn the number of iterations it took to learn the model given a learning rate. It was observed that, as the learning rate increased, the number of iterations required to converge decreased. It was coincidental in my case that the accuracy has not changed significantly as the learning rate changed. It could be completely possible to skip the global minima or even a local minimum while fitting because the algorithm has reached the max number of iterations. If the learning rate is too high, it could wander around the minimum point and never reach it as the steps were too high. The observed learning rate vs the number of iterations were,

0.0001	-	10000 (Max iterations)
0.001	-	10000 (Max iterations)
0.01	-	10000 (Max iterations)
0.1	-	7176
1	-	760

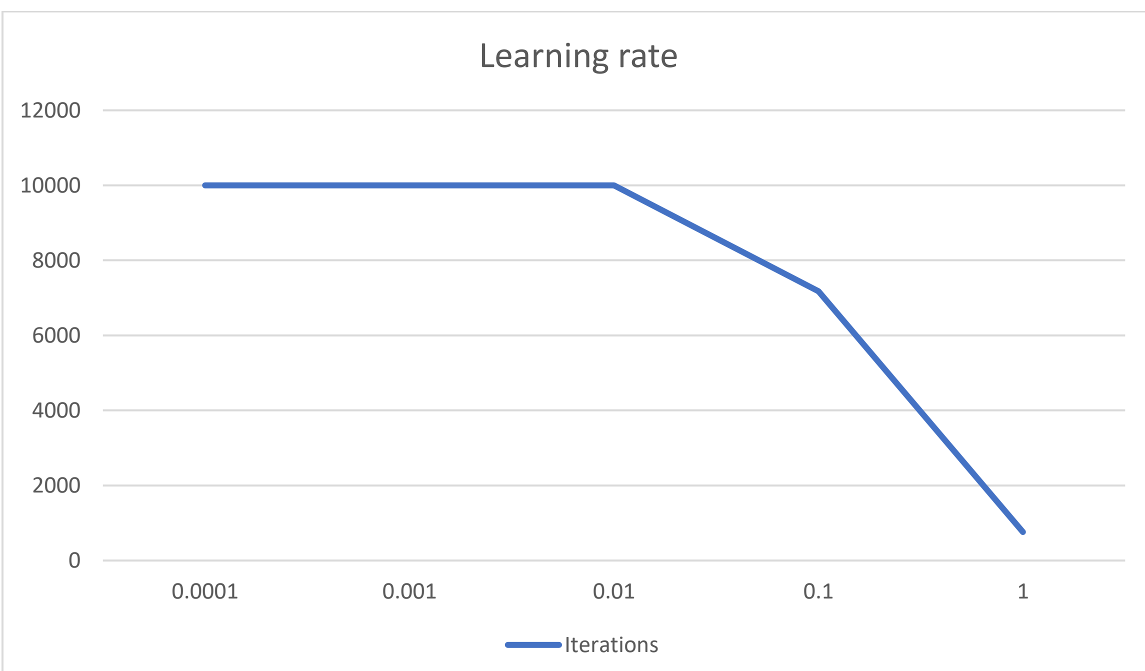
The ROC curve is,



With False positive rate as X axis and True positive rate as Y axis.

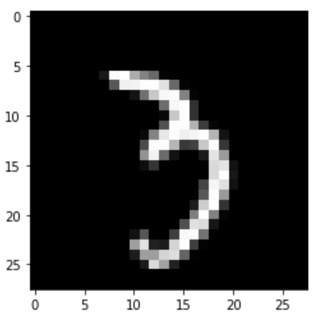
Area under the curve is, 0.030

The learning rate curve is,



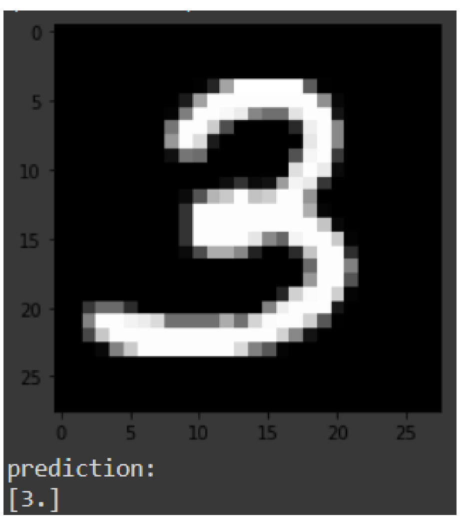
The task in problem 2 was a classic machine learning problem to classify handwritten digits into digital numbers. It has many applications in handwriting detection, number plate recognition etc. The dataset used was MNIST, which is available publicly in many places on the internet. One of such sources is MXNET which was used in my solution. The second task was to filter the data to only contain 5 digits, 0 to 4. Which was easy to achieve programmatically by using the filter function of the mxnet object.

The image is just a matrix, with values ranging from 0-255 depending on how white the pixel is. On plotting, the matrix, we get image as follows.



This is then given as input to the model to make a prediction. The matrix is flattened and normalized. Normalization here refers to dividing all the elements of the matrix by 255 to reduce computations. The model is then trained on the samples provided and the tested on the test data that was collected from MXNET's MNIST data. The model that was trained in the process produced a training accuracy of 0.9604523 and testing accuracy of 0.96205485.

Testing prediction on one of the images,



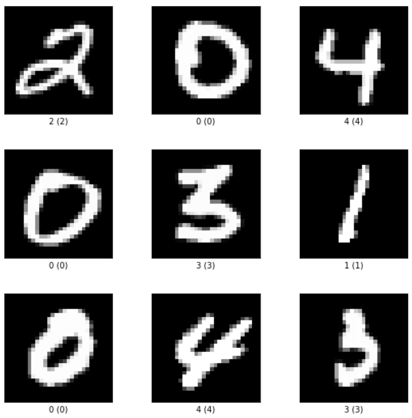
The classification was right in this case.

The reduction in training images, helped increase the speed of training.

The problem 3 uses the same dataset as that of problem 2. The engine now would be an artificial neural network that would predict the same given training.

This time I used tensorflow to create the neural network. The dataset thistime is form the tensorflow datasets itself. That has the MNIST dataset.

Sample of images loaded form the dataset,



The first network, that was asked was to build a sigmoid network, where the network consisted of 1 hidden layer and a total of 3 sigmoid layers. The second network consisted of two ReLU layers and a SoftMax layer at the end. With the ReLU network, the training time was much less as the gradient calculation is easier for this than the later. The accuracy was comparatively similar but the ReLU outperformed sigmoid by 2%. The testing accuracy was 97% for sigmoid, while for ReLU, it was, 99%. The ReLU seems to be a better choice for this dataset given the ease in complexity of network and reduced number of calculations that need to be performed.

In comparison to problem 2, where logistic regression was used, artificial neural networks seem to perform better in terms of training and prediction times and the accuracy. As the number of tunable metrics in ANNs is much higher.