Classifications through Fully Connected Neural Networks

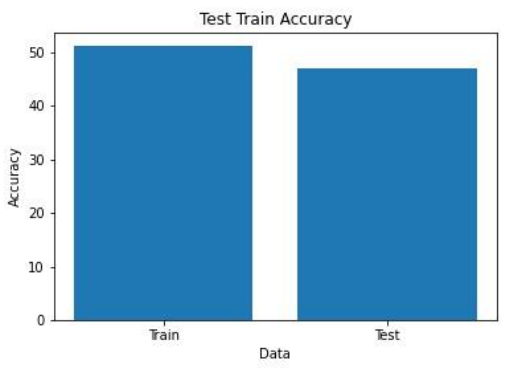
Task 2: Binary Classification

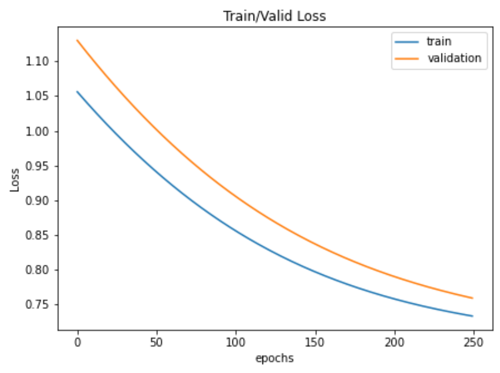
Neural Network Architecture:

This task required binary classification of red and blue circles as 0 or 1. Data size taken is 1000 samples which is further split into train valid and test as per [70:10:20] split. A fully connected network with input size of 2, one hidden layer of 3 neurons and an output layer with single output. Network has two set of weights W1 and W2. Dimensions of W1 are [2+1,3], third one added for the bias factor. W2 is [3+1,1]. Sigmoid activation is applied on the both the layers and loss calculated through cross entropy and network is learned through batch gradient decent.

Network Functions and Accuracy:

Each input sample is a 1x2 array but in order to match the dimensions and compute correct calculations as per the formula, W.T*x +b, I add a third column of ones to the data so that each sample is 1x3 array where the third value is 1. Similar operation is performed for on the output of hidden layer to match the W2. Sigmoid is applied at both the layer outputs to introduce non linearity. The results of activations are stored in network global variables which are then used while backpropagating the loss. During back propagation, the last row of W2 is dropped in order to match the dimensions while matrix multiplication. The training accuracy obtained for 250 epochs and 0.001 learning rate, is 51.29% whereas 47.0% on test data. An improve in accuracy was seen with reshuffling the data and other random set of weight initializations. Trained model is saved for further use. The train and validation losses curve and train test accuracy bar chart are bellow.

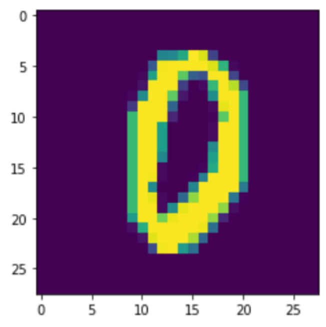
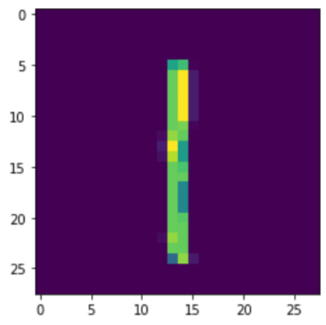
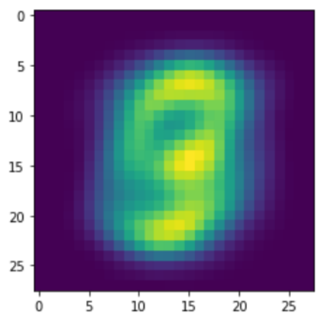




Task 3: Multiclass classification

Data:

This task required classification of MNIST digits from 0-9. The data set provided had 55955 train and test examples with each class present in it. Train dataset was further split into train and valid as per [85:15] ratio but shuffled beforehand to minimize the chances of bias. Train and test labels are one hot encoded so that it can be compared with the output(SoftMax) of network. Data is normalized by mean scaling every image and vectorized to perform fast NumPy matrix multiplications. Non normalized dataset sample images along with the mean image are as follows.

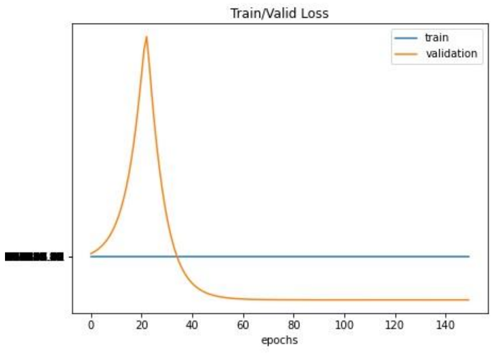
Network Architecture:

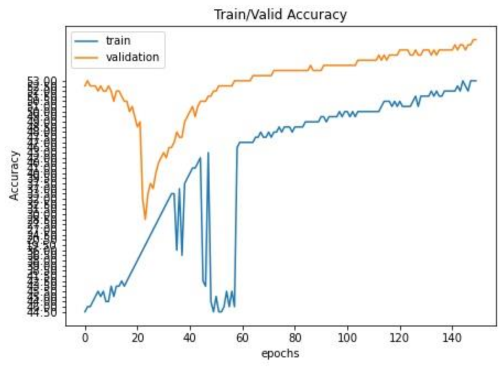
The Neural network for this task has input layer of 785 neurons, 128 1st hidden layer neurons, 64 2nd hidden layer neurons and 10 output neurons. To connect this network, 3 set of weight matrices W1[785x128], W2[128x64] and W3[64x10] along with three set of bias vectors, b1[128x1], b2[64x1] and b3[10x1] are initialized randomly. As the data is large and computations for 2 hidden layers is required, I opt to do mini batch gradient decent with varying batch sizes to check for accuracy. SoftMax activation function is used on the output of 1st and 2nd hidden layers whereas SoftMax is applied on the output of final layer. The activations of the hidden layers are stored in global class variables that help in backprop. Cross entropy loss is computed after every mini batch and the averaged to get loss for one epoch. Accuracy is also computed for every mini batch and then averaged to get the accuracy for one epoch.

Experiments and Accuracy of Mean Subtracted Data:

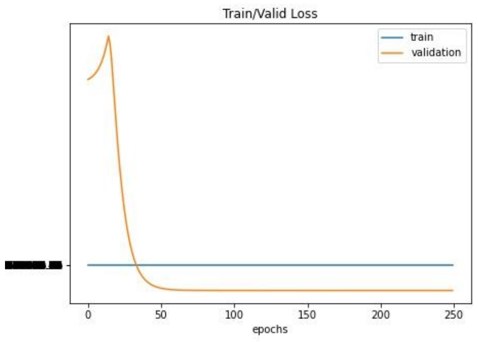
Accuracy reached 100% in 5 epochs with batch sizes[10000-20000] but learning rate 0.001. Test data gave 10% accuracy. By decreasing the learning rate to 0.000001, it reached 51% with batch size= 20000 in 150 epochs and test data achieved 10%. Upon only changing the batch size to 25000, accuracy reached 71% in 65 epoch and then straight 100%.

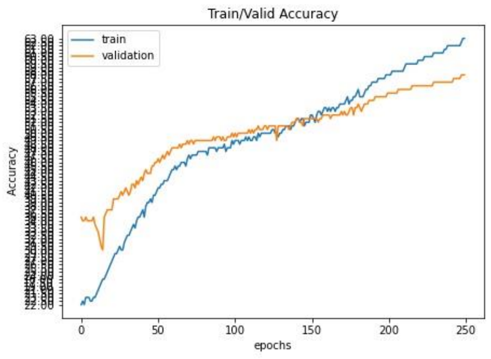
The accuracy and loss curves for 51% training accuracy are:

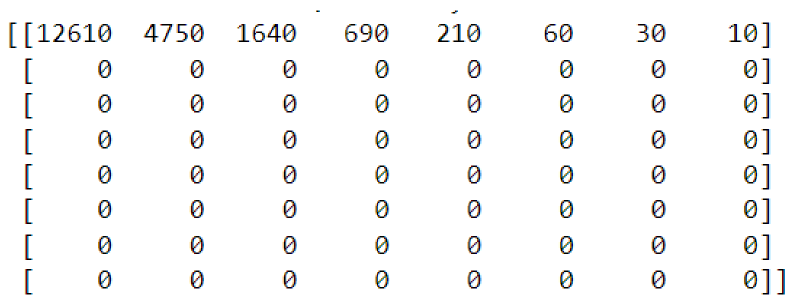




Training the same model upto 250 epochs achieved 63% train accuracy and same 10% test accuracy.



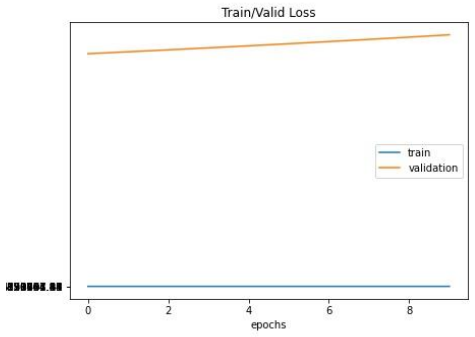


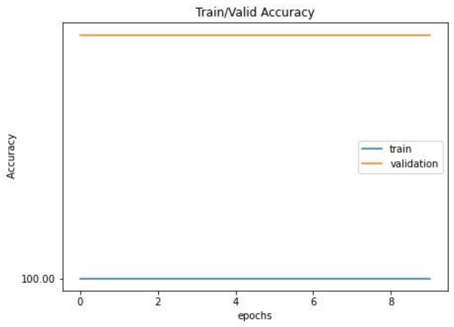


```
[[12610  4750  1640   690   210    60    30    10]
 [    0     0     0     0     0     0     0     0]
 [    0     0     0     0     0     0     0     0]
 [    0     0     0     0     0     0     0     0]
 [    0     0     0     0     0     0     0     0]
 [    0     0     0     0     0     0     0     0]
 [    0     0     0     0     0     0     0     0]
 [    0     0     0     0     0     0     0     0]]
```

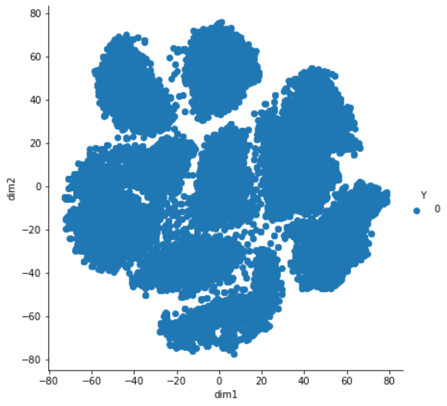
Experiments and Accuracy of Non Mean Subtracted Data:

Model was overfitting on any kind of hyperparameters given.





The TSNE plot genrated for the input data is as follows:


dim2
dim1
Y
● 0