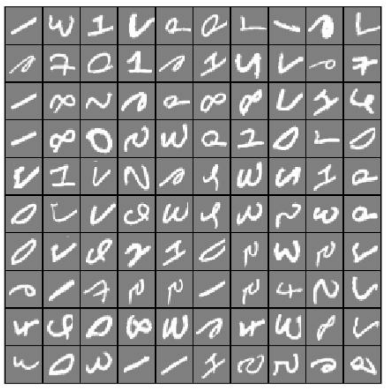
ASSIGNMENT 2

BT3410-Analysis and Interpretation of Biological Data

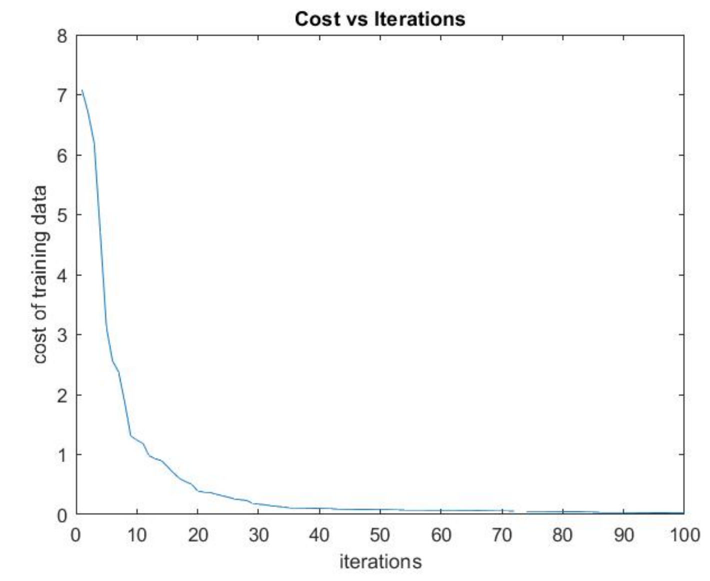
SR Nikitha
BE18B011
April 24, 2021

PROBLEM 1

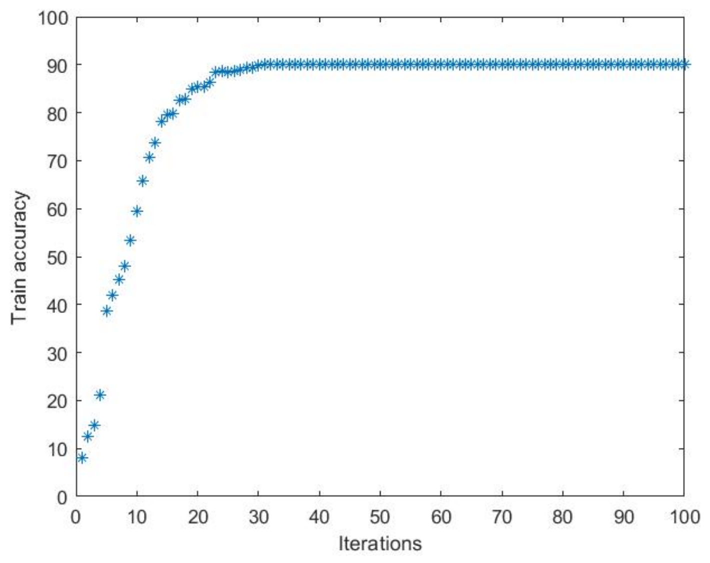
The given dataset has images of 800 images of size 28*28 that can be used for training. This is a small set compared to the original MNIST dataset of 60,000 images. The colour values of pixels of an image are given as a linear vector. The trainLabels matrix contains the labels of the train images. A subset of the given dataset can be visualised as follows:

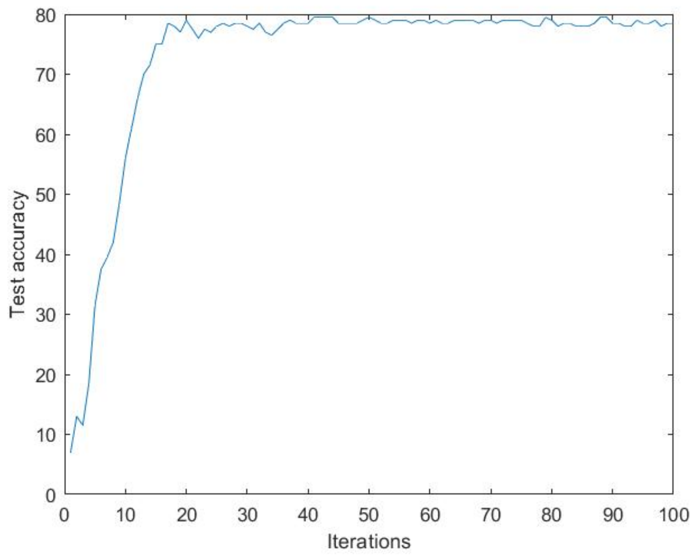


On training the neural network for 100 iterations, a maximum accuracy of 80.5 percent is obtained on the test data. The accuracy on the training data was 99 percent. Plotting the variation of cost function with each iteration of training,



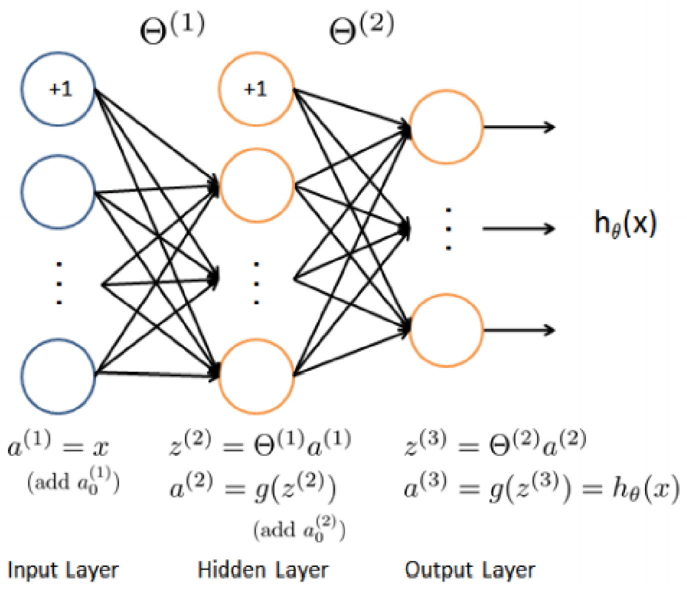
The variation of train and test accuracy with each iteration can be plotted as follows,





a) The network architecture is made of 3 layers : input, hidden and output layers. The size of each layer is,

1. Input layer : 784
2. Hidden layer : 200
3. Output layer : 10 (0 to 9)



b) The activation function used on the nodes of the network is sigmoid function.

SIGMOID

```
function g = sigmoid(z)
%SIGMOID Compute sigmoid function

g = 1.0 ./ (1.0 + exp(-z));
end
```

c) The loss function used for computing loss is the logistic loss function. Logistic loss function is defined as,

$$Cost(h_\theta(x), y) = -y log(h_\theta(x)) - (1 - y) log(h_\theta(x))$$

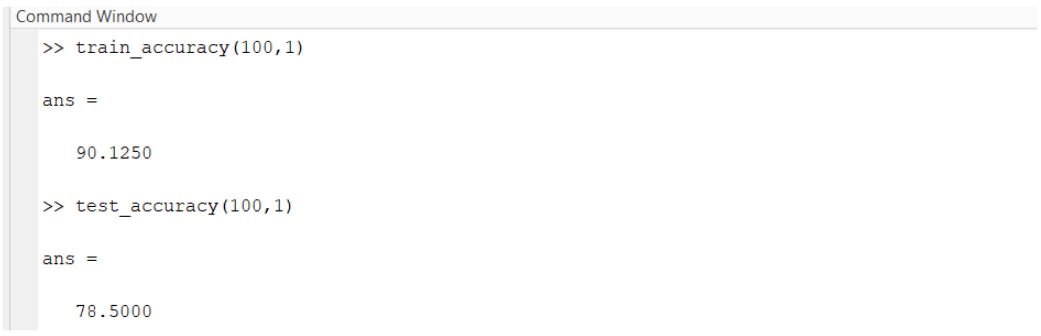
This is for a two class classification model. So the cost function here would have a loss function over all the classified labels from 0 to 9. The complete loss function including regularisation is,

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[-y_k^{(i)}\log((h_\theta(x^{(i)}))_k) - (1 - y_k^{(i)})\log(1 - (h_\theta(x^{(i)}))_k)\right] + \frac{\lambda}{2m}\left[\sum_{j=1}^{25}\sum_{k=1}^{400}(\Theta_{j,k}^{(1)})^2 + \sum_{j=1}^{10}\sum_{k=1}^{25}(\Theta_{j,k}^{(2)})^2\right].$$

d) The learning algorithm is gradient descent with random parameter initialisation. I have used the optimisation function fmincg which is a parallel to fminunc function in matlab but memory optimised for large datasets like this dataset. (source code attached)

e) The learning rate used is 0.002

f) The accuracy of test data obtained is 78.5% and the loss value after 100 interations was 5.344676e-02 .


Command Window

```
>> train_accuracy(100,1)

ans =

   90.1250

>> test_accuracy(100,1)

ans =

   78.5000
```

**All graphs and analysis done on Matlab. All the code files are attached in the folder.