Instructions:

1. Your assignments WILL be tested for plagiarism
2. No late submissions will be accepted.
3. Use Matlab for this assignment and submit your all functions and main files.
4. Use LaTex for writing your analysis Report. LaTex file along with pdf should be submitted (BONUS).
5. Your submission folder should be named on your roll number and should be submitted in .rar file.

Problem 1

(Simple Neural Network)

For this part, you should not use any machine learning libraries. You should implement the backpropagation algorithm yourself.

The dataset for this assignment is from MNIST database. It is a database of handwritten digits. We have split the date set into training set and test set stored in .mat files. There are 60,000 training samples in train.mat, 10,000 test samples in test.mat. Each sample is a handwritten digit represented by a 28 by 28 greyscale pixel image. Each pixel is a value between 0 and 1 with a value of 0 indicating white. Each sample used in the dataset is a feature vector of length 784(28x28=784). The value of a label is the digit it represents, e.g, a label of value 8 indicates the sample represents the digit 8.

The network you are working on in this assigment has 3 layers: one input layer, one hidden layer and one output layer. The activation function used in your hidden layer should be the basic sigmoid function and at output should be the softmax function.

1) Select a learning rate of 0.2.
2) Assign random initial weights and biases to the neurons. Each initial weight or bias is a random floating-point number drawn from the standard normal distribution (mean 0 and variance 1).
3) Create a neural net of size [784,30,10]. This network has three layers: 784 neurons in the input layer, 30 neurons in the hidden layer, and 10 neurons in the output layer.

For each training example in batch, use backpropagation to calculate a gradient estimate, which as you saw in class consists of following steps:

1. Feed forward the input to get the activations of the output layer.
2. Calculate derivatives of the cost function for that input with respect to the activations of the output layer.
3. Calculate the errors for all the weights and biases of the neurons using backpropogation.
4. Update weights (and biases)

Tasks:

a) Devise your program using Sum of squared errors (SSE) and report the roc curves and model accuracy
b) Devise your program using cross entropy as your cost function and report the roc curves and model accuracy

c) Compare the results of both parts (a) and (b)
d) (Using either SSE or cross entropy) Report the accuracy with 2 other values (other than 30) of hidden layer nodes and comment on the results.
e) (Using either SSE or cross entropy) Report the quantitative results with change in learning rate value (You can chose 0.1 and 0.8).

Problem 2

(Short Questions)

Please give the precise answers to following questions.

1. Three advantages of using neural network as auto encoder.
2. How CNN works? Write in clear steps. (Define any term you use)
3. Comment on how regularized gradient descent affect model overfitting.