CSE/ECE 343/543: Machine Learning
Assignment-3

Max Marks:210 **Due Date**: 11:59PM Nov. 12, 2018

Instructions

- Keep collaborations at high level discussions. Copying/Plagiarism will be dealt with strictly.
- Start early, solve the problems yourself. Some of these questions may be asked in Quiz/Exams.
- Submission Instructions: Submissions will be through backpack. Create a single *firstname-A3.zip* file containing a report **A3.pdf**, your source folder **A3-src** and theory questions solutions **A3-theory.pdf**. Report all your theory solutions and outputs of all programming questions e.g intrinsic and extrinsic parameters,figures, images etc in **A3.pdf**. List name of all the functions/scripts that you have implemented along with the two line summary in **A3.pdf**. Put all your programming functions/scripts in **A3-src**. You are allowed to use *numpy*, *scipy* and *matplotlib* only, unless specified otherwise. In case of any doubt, initiate a discussion on backpack or drop an email to Priyabrata {`priyabrata17043@iiitd.ac.in`} **and** Aman {`aman15012@iiitd.ac.in`} with the subject line [**ML18-A3-Doubt**]. Emails with other subject lines may suffer delays in response.
- Report(A3.pdf) is **required**. 50% of the total points of the programming question will be deducted if the results are not reported in A3.pdf
- Late submission penalty: As per course policy.

PROGRAMMING QUESTIONS

1. (100 points) **Neural Networks.**

   i) Implement the forward and back propagation algorithms to train an artificial neural network for MNIST (attached) from scratch. You are not permitted to use any external libraries in this part.

      a) (*60 points*) The arguments to your function would be the number of layers and the number of nodes in each layer. Assume a sigmoid activation function in each layer and softmax in the output layer. You have to construct the following neural network architectures from the above:

         i) 1 hidden layer - [100 units]

         ii) 3 hidden layers - [100, 50, 50 units]

         Implement forward and backward propagation, which can be used for a general fully-connected neural network with softmax layer as the output. Use this function to train the two architectures and test them using the provided MNIST subset. Report accuracy as the evaluation metric in this part. Save the weights of your best models. Please ensure that your code is general and does not hard code equations.

b) (*20 points*) How would you diagnose if your model has overfitted or underfitted? Show plots/graphs and any other evidence to establish that your model is well-trained and can generalize well. Give proper justification with challenges you faced and counter-measures you took.

c) (*10 points*) Implement ReLU for the hidden layers and repeat part A and B using these activations at every layer (except output). You should still use softmax at the output.

d) (*10 points*) Implement part A using the sklearn. Report the accuracy, as compared to what you received with your own network. Explain reasons for an observed difference in accuracies, if any. .

2. (50 points) Use the binary CIFAR 10 subset (attached) for this part. Use the existing AlexNet Model from PyTorch (pretrained on ImageNet) as a feature extractor for the images in the CIFAR subset. You should use the fc8 layer as the feature, which gives a 1000 dimensional feature vector for each image. Train a linear SVM from the sklearn library over these extracted feature vectors for classification. Report the test accuracy along with the confusion matrix and the ROC curve.

THEORY QUESTIONS

3. (10 points) Assume you have a given data-set with some labeling, which you try to train using a neural network of n layers. Each input datum is a m-dimensional array, with each value in the range [500, 1000]. Your peer 'X' uses sigmoid activation in the networks layers, but is unable to train the model successfully. Assuming that there is no problem in the model's architecture, what could be the possible problem? Explain in terms of the activation function and back propagation. What would happen if X used ReLU instead? Would the problem worsen or get better? Suggest a data pre-processing technique (for both settings) to remedy this problem. After deciding the architecture, you wish to initialize the network. You decide to initialize all the weights as zeros. Is there a problem with his approach? Suggest a better method for initialization of weights. Which loss function between MSE and Cross-entropy should he use if it is a classification problem; justify.

4. (10 points) When used for classification, the quadratic (squared error) cost function poses a "*learning slowdown*" problem (learning is slow when the error is large). Explain mathematically how the cross-entropy cost function solves this problem.

5. (10 points) Can a neural net of arbitrary depth using just linear activation functions be used to model the XOR truth table? Can you mathematically prove this classifier equivalent to be same as another classifier discussed in class?