Machine Learning
Homework 8 : Neural Networks and Logistic Regression
(due Midnight April 19)

Instructions

1. The deadline for full score is Midnight April 19. You can get 50% credit for late submission (Midnight April 20).
2. Total marks = 15
3. You have to type the assignment using a word processing engine, create a pdf and upload on the form. Please note that only pdf files will be accepted.
4. All code/Jupyter notebooks must be put up as secret gists and linked in the created pdf submission. Again, only secret gists. Not public ones.
5. Any instances of cheating/plagiarism will not be tolerated at all.
6. Cite all the pertinent references in IEEE format.
7. The least count of grading would be 0.5 marks.

1. (a) Logistic Regression
   - i. Implement a function for binary logistic regression using gradient descent **[2 marks]**
   - ii. Show the usage of your implementation on the IRIS dataset. We will only be making use of sepal-length and petal-width as the two features. We have only two classes - Setosa and Not-Setosa. **[1 marks]**
   - iii. Plot the decision boundary **[1 marks]**
   - iv. Compare your implementation against sklearn's Logistic Regression **[1 marks]**

   (b) Neural Networks
   - i. Implement a neural network class that can be instantiated with:
     - A. an input data matrix X containing samples as rows and features as columns
     - B. a list containing number of hidden units in each hidden layer
     - C. a list containing activation function to be used in eacyh layer: sigmoid, softmax, ReLU, or identity (or linear)
     - D. a cost function

     As an example: let us say we have an input data matrix of shape 100X3, we use 2 layers and the number of hidden units is: [4, 2, 1] where the last number (1) indicates number of units in the output layer, the activations we use for the three layers are: ['ReLU','ReLU','Linear'] **[1 marks]**
   - ii. For this class define a method forward propagation **[2 marks]**
   - iii. For this class define a method backward propagation which provides the derivative of weights wrt cost function **[2 marks]**
   - iv. Implement a gradient descent based method to update the weights **[1 mark]**

   (c) Show the usage of your defined neural network on:
   - i. MNIST dataset where you shuffle the data once, and then use first 50% of the data for training, next 20% for validation and last 30% of the data for testing. You must return the confusion matrix and overall test accuracy. You may choose the number of layers and activations as per your choice **[2 marks]**

ii. Housing price dataset as you’ve been using thus far in earlier assignments where you shuffle the data once, and then use first 50% of the data for training, next 20% for validation and last 30% of the data for testing. You must return the RMSE and MAE. You may choose the number of layers and activations as per your choice **[2 marks]**