1081 Deep Learning – Homework 2

Due: Nov. 24, 2019, 11:55pm

1. (20%) Please load 'data.mat' into your Python code, where you will find $x, y \in R^{1001}$. Please use the loss function (L1 Norm) below to find the curve $y = \theta_0 + x\theta_1 + x^2\theta_2$ that best approximates y based on x. (upload your source code to moodle, and show the results, including a plot where the line is overlaid over the given data in the report)

$$Loss\ Function: f(\boldsymbol{\theta}) = \sum_{i=1}^{N} |y_i - (\theta_0 + x\theta_1 + x^2\theta_2)|$$

2. (20%) In 'train.mat,' you can find 2-D points X=[x1, x2] and their corresponding labels Y=y. Please use logistic regression $h(\boldsymbol{\theta}) = \frac{1}{1+e^{-\boldsymbol{\theta}^T x}}$ to find the decision boundary (optimal $\boldsymbol{\theta}^*$) based on 'train.mat." Report the test error on the test dataset 'test.mat.' (percentage of misclassified test samples) Hint: you can use "LogisticRegression" in Python.

3. (25%) Please use a gradient descent method to solve Question 2. (show your code, decision boundary, and test error on the test dataset)

4. (35%) Please use a neural network (with its architecture shown below) to find the decision boundary based on 'train.mat." The activation function must be used in the two hidden layers and the output layer. You are required to use a **gradient descent** method to train the network. Report the test error on the test dataset 'test.mat.' (percentage of misclassified test samples) Note that if you use any off-the-shelf functions to construct and optimize your network model, the most points you can get is merely 5%.