

CS401 Artificial Intelligence

Programming Assignment No 2 (Sections C, D, E)

Spring 2018

Assigned on 21/03/2019

Deadline Part a) 24/03/2013 before 11:55 p.m.

Part b, c) 31/03/2013 before 9:00 a.m.

Submission: Online on SLATE and presentation/viva

Weight 4%

Digit Recognition Using Supervised Learning**Instructions:****Following rules will be enforced for this assignment**

- This assignment will be done independently using either MATLAB/OCTAVE or PYTHON 3.5 or later as the implementation platforms.
- We will not use any library/built-in implementations of the methods for digit classification.

In this assignment we are going to search and download the MNIST digit recognition dataset of hand written digits and use some of the supervised learning methods discussed in class to train models for predicting class of a test instance.

Part a) Using K-NN classifier to predict labels of an input image.**[5 + 5 Points]**

Our first job in this assignment is to compare performance of K-Nearest Neighbors classifier for the MNIST digit recognition dataset. We are going to figure out the best value of K (i.e. value of K that gives the best test accuracy) for this problem by using validation data kept separate for this purpose.

For this part we are going to use i) **cosine similarity** and ii) **Euclidean distance** (also known as L2-norm) for computing the nearest neighbors of an input instance using the raw images (pixel values).

Part b)

In this part we are going to train a system of ten neurons (i.e. **sigmoid unit**) each trained to discriminate a single digit from the rest of the digits. We will use the gradient descent algorithm for finding optimal weights of each of the neurons in the system. In general we are going to write a script/code for

- learning a threshold unit using the perceptron learning rule **[5 Points]**
- learning a perceptron with sigmoid activation function using gradient descent **[5 Points]**
- learning a neuron with squashing (i.e. **tanh**) activation function using gradient descent **[5 Points]**

Part c)

Compare our results with the results using Histogram of Oriented Gradients (HOG features) as descriptors/features to represent your digit images and using SVM to classify the digits. (You can use any available implementation of the HOG based digit recognition using SVM) **[5 Points]**