

# Cmpe 462 - HW3

Şahin Batmaz

2012400117

## QUESTION 1

First, the data is splitted into 3 vectors for 3 classes. Then each class data is divided into training and test datasets. I wanted to have around 60% of the class data to be training set and rest to be the test set. The sizes of the classes are 131,135 and 134. Since the number of data points in classes are not same and are not divisible for finding the 60%, I fixed the training dataset size as 80 which is around 60% of a class data. To select 80 points in classes, for each class, 80 random numbers are generated such that minimum available number is 1 and maximum available number is the size of the class. These values are treated as indices and training sets are obtained with them. The remaining points are constructed the test datasets.

Output values are calculated with the following formulas:

- Input Layer : x vector  $\rightarrow [1 \ x_1 \ x_2]$  (First value 1 is the bias)
- Hidden Layer : z vector  $\rightarrow [1 \ z_1 \ z_2 \ z_3 \ z_4 \ z_5]$  (First value 1 is the bias)
- Output Layer : o vector  $\rightarrow [o_1 \ o_2 \ o_3]$
- Weights between input and hidden layer : w matrix which is 5 by 3.
- Weights between hidden and output layer : v matrix which is 3 by 6.

- 
- z vector = x vector \* w matrix
  - o vector = z vector \* v matrix
- 

To process output layer, I used softmax formula. The vector is specified with p.

- p vector = softmax(o vector)
  - softmax formula =  $\exp(o \text{ vector}) / \sum(\exp(o \text{ vector}))$
- 

Then for the error function I used cross entropy.

- error =  $-\sum(c_i \log(p_i))$   
where  $c_i$  is 1 if 'i' is the class that the data point coming from, otherwise 0.
- 

Then I simplified the formula by eliminating the '0' cases.

- error =  $-\log(p_j)$   
where j is the class that the current point is coming from.

# REPORT

There are 4 files submitted; 0-prepare.m, 1-functions.m, 2\_1-gaussians.m, 2\_2-knn.m. To run 'gaussian' and 'knn' scripts, 'prepare' and 'functions' scripts must be run first.

## 1. 0-prepare.m

In this script, points2d.dat file is read. Each class is extracted into separate vectors. From these vectors; training and test datasets are randomly constructed. Then training and test datasets are combined in trainingList and testList respectively. With this script, data is prepared to be used.

## 2. 1-MLPerceptron.m

- x,z,o and p vectors are initialized.
- v matrix and w matrix are initialized as v\_ih and w\_hk.
- Hyperbolic tangent function is implemented.
- Calculating functions are implemented for z,o and p.
- Weight calculator functions are implemented by using the derivations done on hand.
- Then the model is run on the dataset and the following iterations are followed:
  - x,z,o and p vectors are calculated.
  - Backpropagation values are calculated for all weights.
  - At the end, weights are updated.
  - Meanwhile, the error is counted for the dataset for the current iteration.

NOTE: MY CODE **DOES NOT WORK**. THE POINTS ARE ALWAYS CLASSIFIED AS THE SECOND CLASS AND THERE IS NO IMPROVEMENT OBSERVED. I COULD NOT FIND THE PROBLEM. I MADE THE DERIVATIONS MANY TIMES. I BELIEVE I FOLLOWED THE APPROPRIATE APPROACH BUT I MADE MISTAKES IN SOME PLACES.