Homework 4

Due: October 30, 2018, 11:59 PM EST

Instructions

Your homework submission must cite any references used (including articles, books, code, websites, and personal communications). All solutions must be written in your own words, and you must program the algorithms yourself. If you do work with others, you must list the people you worked with. If you solve any problems by hand just digitize that page and submit it (make sure the problem is labeled).

Your programs must be written in Python. All code must be able to compile and run for full credit. Comment all code following proper coding conventions. Remember, if we can't read it, we can't grade it! (For more information on python coding standards, refer to: *https://www.python.org/dev/peps/pep-0008/*)

You should submit your assignment via Github. Submit your solutions as a PDF named "hw(hw #).pdf". For example, homework 4 should be submitted as hw04.pdf. If the assignment requires coding, submit your working code as a .py file with the same name.

If you have any questions address them to:

- Connor McCurley (TA) – cmccurley@ufl.edu
- Xiaolei Guo (TA) – suninth@ufl.edu
- Daniel Wells (TA) – dwells@ufl.edu

Question 1 - 2 points

Consider the following multi-layer perceptron.

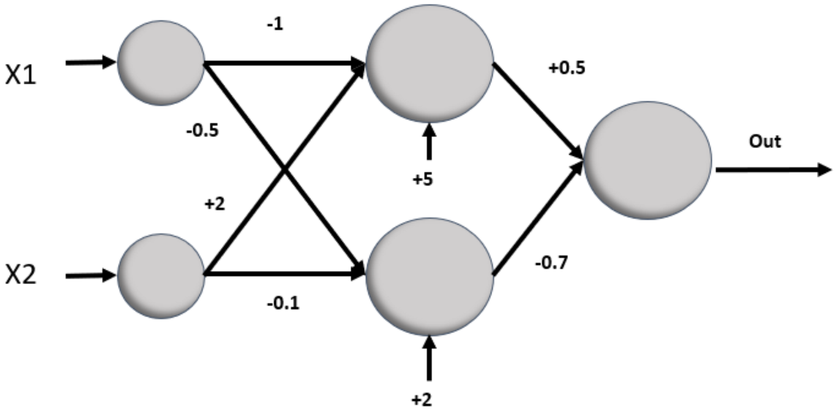

Figure 1: Two layer multi-layer perceptron

For the set of input data

$$\boldsymbol{X} = \left\{ \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -2 \\ -3 \end{bmatrix}, \begin{bmatrix} -5 \\ -1 \end{bmatrix}, \begin{bmatrix} -2 \\ 2 \end{bmatrix}, \begin{bmatrix} 6 \\ -2 \end{bmatrix}, \begin{bmatrix} 4 \\ 3 \end{bmatrix} \right\}$$

calculate the output vector $\boldsymbol{y} \in \mathbb{R}^6$. You can assume that all activation functions $\phi(\nu)$ are rectified linear units (ReLU), where

$$\phi(\nu) = \begin{cases} \nu & for\, \nu > 0 \\ 0 & otherwise \end{cases}$$

Question 2 - 8 points

For the following two-dimensional data sets, do the following:

1. For each data set, design **two** different neural networks by hand which can discriminate between the classes (each class indicated by a different color).

2. Code your solutions in python and verify that the network learns a decision boundary similar to what you designed. For this section, you should include plots showing the data and the corresponding decision boundaries learned by the networks.

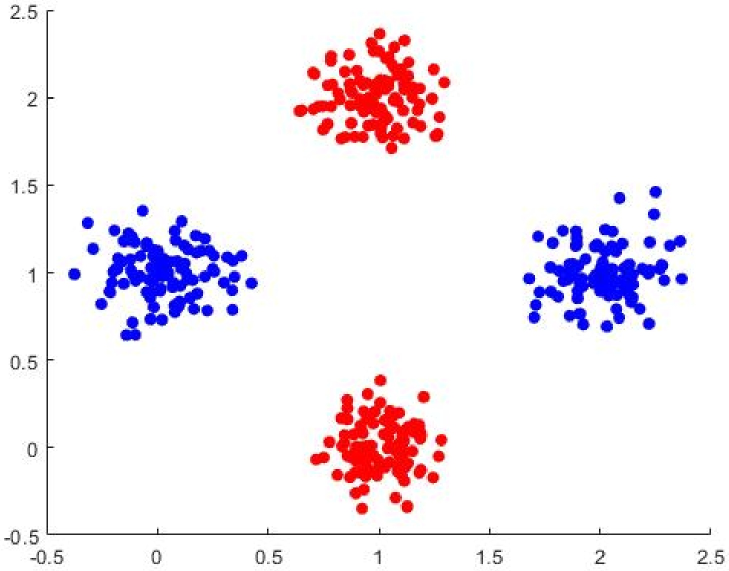

Figure 2: Data set 1

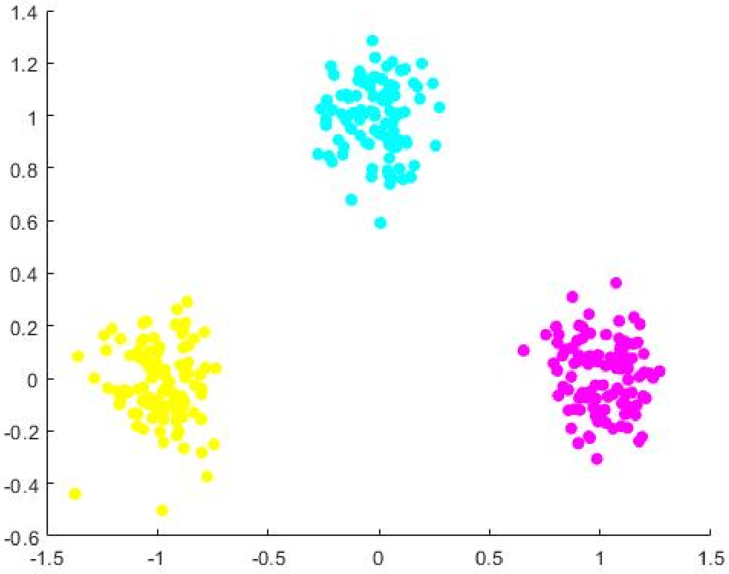

Figure 3: Data set 2

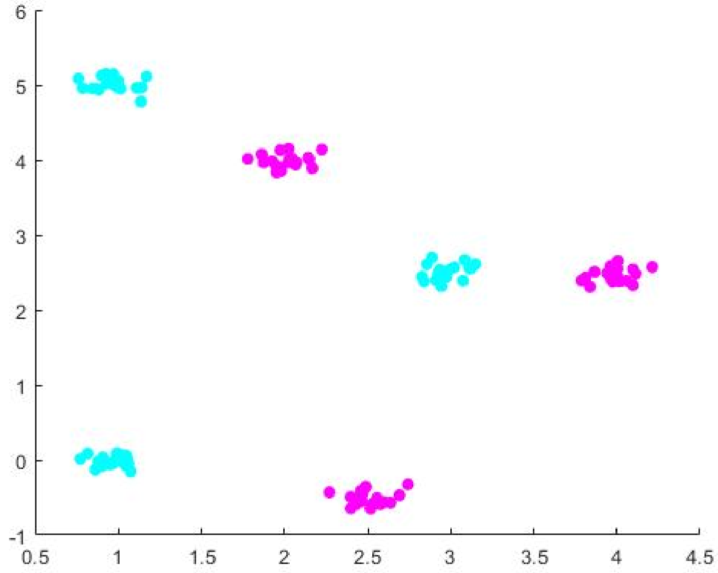

Figure 4: Data set 3