Homework 6

Due: November 26, 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 6 should be submitted as hw06.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:

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

Question 1 - 10 points

Train a multi-layer perceptron using backpropagation for the following two dimensional dataset. We have provided some code to implement the MLP, but anything missing you will need to code up yourself (independently - without using online code sources) and add it to the code. There are two python files, one with the MLP implementation (labeled mlp.py) and another for instantiating the MLP and using it (run.py). If all the proper code is added in the correct manner, only run.py needs to be run. MLP.py should **NOT** be executed directly. It is a module you import into your run.py file.

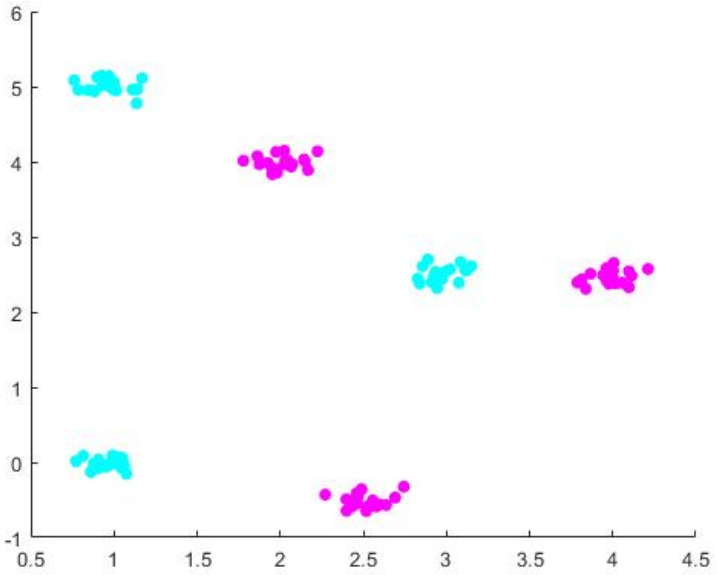

Figure 1: 2-D dataset

For the report, use confusion matrices to demonstrate the accuracy of your MLP, and discuss the methods you used to achieve the highest accuracy. How many nodes do you have on your hidden layer? How did changing the learning rate change your results? What role does increased different number of iterations have on accuracy?