Homework #4

CS 539, Fall 2022

100 points total [6% of your final grade]

Due: October 25, 2022 by 11:59pm
[no submission will be accepted after October 27, 2022 at 11:59pm]

Delivery: Submit via Canvas

Part 1. Softmax regression and neural network [80 points]

In this part, you will implement softmax regression (in problem1.py) and fully connected neural network (in problem2.py) with stochastic gradient descent in python3.

We provide the following files:

a) problem1.py - You will implement several functions of softmax regression. Do not change the input and the output of the functions.
b) test1.py - This file includes unit tests. Run this file by typing 'nosetests -v test1.py' in the terminal. No modification is required.
c) Problem2.py - You will implement several functions of neural network. Do not change the input and the output of the functions.
d) test2.py - This file includes unit tests. Run this file by typing 'nosetests -v test2.py' in the terminal. No modification is required.

Part 2. Using neural nets to recognize handwritten digits [20 points]

In this part, you will deal with the MNIST Database (http://yann.lecun.com/exdb/mnist/). The MNIST Database is a collection of samples of handwritten digits from many people, originally collected by the National Institute of Standards and Technology (NIST), and modified to be more easily analyzed computationally. We will use a tutorial and sample software provided:

- Read CHAPTER 1: Using neural nets to recognize handwritten digits from "Neural Networks and Deep Learning" by Michael A. Nielsen, Determination Press, 2015.
- Download the samples provided with the chapter. (Note that the data files represent a further modification of the original NIST data). The orginal software linked in the tutorial was written in python 2. Therefore, instead, download this software written in python 3: https://github.com/chengfx/neural-networks-and-deep-learning-for-python3. This software package also include the MNIST Database.
- Following his development, and using the Python software provided (again, use the python 3 version), run the 30-epoch, 30 hidden-unit, η=3.0 neural network example and save your results in a table. Do this three times, and use the median performance as your "result".
- Then, experiment with at least 3 alternative network hyper-parameters and topologies (e.g.,different # of epochs, # of hidden units, η, activation functions).
- Save and summarize the results and report them.
- Through the experiment, what is the best configuration? What did you learn?
- Any idea to further improve the prediction rate?

What to turn in:

- Submit to Canvas your problem1.py, problem2.py and pdf document for part 2.
- This is an individual assignment, but you may discuss general strategies and approaches with

other members of the class (refer to the syllabus for details of the homework collaboration policy).