CSE 326/426 (Spring 2019) Project 3

Due on 11:55pm, May 1, 2019

Goal: Implement a feedforward neural work for classification and regression

Instruction:

- Read PRML Section 5.1 and 5.3 about forward and backward passes over feedforward neural networks (also known as multilayer perceptron, MLP).
- Go through our slides to understand the shapes and computations of the gradientis during backpropagation. You're suggested to write down the gradient equations in matrix and vector form, and figure out the meaning of each element in those matrices/vectors and how they are computed. Be vigilant when dealing with the indices of layers, units, and examples.
- Now start coding. The package contains 5 files, as shown below.

```
.
 src
     problem1.py
     problem2.py
     problem3.py
     test1.py
     test2.py
```

- In problem1.py, you will implement 4 activation functions (sigmoid, tanh, softmax, and identity), and 2 loss functions (cross entropy loss and mean square error, MSE). The gradients of some of these functions will be implemented as well. If a function raises "NotImplementedError", there is no need to do anything to that function.
- In problem 2.py, you will implement the forward and backward passes of a neural network using the building blocks from problem1.py. Then implement the train function, using the forward and backward functions, to implement gradient descent training of the network.
- Files test$*$.py ($* = 1, 2$) will unit-test the correctness of your implementations in the corresponding problem$*$.py files. For example, after you implement problem1.py file, run

  ```
  nosetests -v test1
  ```

  to test the correctness of your implementation of problem1.py. Note that passing the tests does not mean your implementations are entirely correct: the test can catch only a limited number of mistakes.

  If you use Anaconda (highly recommended), there should NOT be any packages you need to install. Otherwise, you will need to install nosestest for such unit test.

Grading: This project has 100 points in total. The number of points for each functions are printed when you run nosetests. 25 points go to the training quality, such as speed, accuracy, and completeness, when running problem3.py. The project counts towards 6% of the final grade (30% for all projects: 6% for projects 1-3, and 12% for the team project).

Submitting: There is no hand-written report required, and your submission should include the ONLY file

```
<your_LIN>_P3.tgz
```

which, when decompressed, contains the same file trees shown above but with functions in problem∗.py implemented. Submit the file to Coursesite.