CS37300 Homework	— Homework 5—	Last update: November 3, 2017
https://www.cs.purdue.edu/homes/ribeirob/courses/Fall2017		Due: Nov 13, 2017 (06:00am)

Instructions and Policy: Each student should write up their own solutions independently. You need to indicate the names of the people you discussed a problem with; ideally you should discuss with no more than two other people.

- YOU MUST INCLUDE YOUR NAME IN THE HOMEWORK
- The answers (without the python scripts) MUST be in submitted in a single PDF file via Blackboard.
- The python scripts will be submitted separately via turnin at data.cs.purdue.edu.
- Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary.
- Theoretical questions MUST include the intermediate steps to the final answer.
- Zero points in any question where the python code answer doesn't match the answer in the PDF.
- Python code answers MUST adhere to the format described below.

There are TWO (2) questions in this homework.

Python code guidelines

Turn in each question of your homework in a separate python file named `hw5-X.py`, where X is the question number.

Your code is REQUIRED to run on either Python 2 or Python 3 at scholar.rcac.purdue.edu. Preferably use Python 3 (Python 2 will also be accepted). The TA's will help you with the use of the scholar cluster. If the name of the executable is incorrect, it won't be graded. Please make sure you didn't use any library/source explicitly forbidden to use. If such library/source code is used, you will get 0 pt for the coding part of the assignment. If your code doesn't run on scholar.rcac.purdue.edu, then even if it compiles in another computer, your code will still be considered not-running and the respective part of the assignment will receive 0 pt.

Q1 (40 pts): [Bagging and Boosting]

Bagging is generally used to help stabilize classifiers with unstable learning algorithms (optimal score searching algorithms). A classifier has a stable learning algorithm if, by changing the training data, the predicted class labels in the test data don't change. For instance, the predictions of a decision tree might significantly change with a small change in the training data. This definition depends on the amount of data, of course. Classifiers that are unstable with 10^3 training examples may be stable with 10^9 examples. Bagging works by aggregating the answers of unstable classifiers trained over multiple training datasets. These multiple datasets are often not independent, generally sampled with replacement from the same training data.

Boosting works by converting weak classifier (very simple models) to strong ones (models that can describe complex relationships between the inputs and the class labels). A weak learner is a classifier whose output of an test example attributes x_i is only slightly correlated with its true class t_i. That is, the weak learner classifies the data better than random, but not much better than random. In boosting, weak learners are trained sequentially in a way that the current learner gives more emphasis to the examples that past learns made mistakes on.

(a) (20pts) Suppose we decide to use a large deep feedforward network as a classifier with a small training dataset. Assume the network can perfectly fit the training data but we want to make sure it is accurate in our test data (without having access to the test data). Would you use boosting or bagging to help improve the classification accuracy? Describe what would be the problem of using the other approach.

(b) (20pts) Download the code at
`https://www.cs.purdue.edu/homes/ribeirob/courses/Fall2017/data/mistery_classifier.zip` .
Read the code and answer the following questions:
Note: You don't need to run the code, but if you want to run it on the scholar.rcac.purdue.edu cluster, you will need to command `module load anaconda/5.0.0-py36`

(i) (10 pts) Which classifier is this? Why are the trees used in this classifier so shallow? Describe how the classifier works using pseudo-code.
Hint: It uses decision trees but this is not a decision tree classifier.

(ii) (10 pts) What happens if we have mislabeled data? Why mislabeled data could be a problem?
Hint: We are looking for an asnwer that uses the training weights of the examples at each iteration as a justification.

Q2 (60 pts): [Deep Learning / Python] In class we have seen the training of a one layer feedforward neural network using Stochastic Gradient Descent (SGD)
`https://www.cs.purdue.edu/homes/ribeirob/courses/Fall2017/lectures/MiniBatch_GD_FeedForward_ReLU.html`
We also observed that the code is missing something important: the bias term of each neuron. You are requested to perform the following modifications to the original code:

(a) **(20 pts)** Add a bias term to the parameters of the hidden layer neurons (the bias goes before the activation of the hidden layer neurons). Do the same to the input that activates the output layer neurons.
(1) Describe the new equations of your neural network as we did in section "**Define the forward pass**" at `https://www.cs.purdue.edu/homes/ribeirob/courses/Fall2017/lectures/MiniBatch_GD_FeedForward_ReLU.html`.
(2) Report the new accuracy after 10 iterations (at PDF) and the new test scatter plot in line 11 (at PDF).
(Python Source) Turn in your python source as hw5-2.py (this is your code with the biases). Make sure that the only ouput is

```
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Iteration 7
Iteration 8
Iteration 9
Accuracy after 10 iterations:  (your accuracy)
```

Replacing "(your accuracy)" by whatever you get with your new algorithm.
Hint: *For each neuron:* If the neuron activation function is $\sigma()$ and the input to the neuron is z, output of the neuron is $\sigma(z)$. The bias is an extra parameter w_0, such that the new neuron output is $\sigma(z+w_0)$.

(b) **(20 pts)** Using the previous code (with the bias) and the original code, report the average and standard deviation accuracies of 20 runs with 100 iterations each. Plot these values using a boxplot. Is there a significant difference? Yes/no, why? (explain using the boxplots)
Hint: Example code to plot boxplots in python:
`https://www.cs.purdue.edu/homes/ribeirob/courses/Fall2017/hw/hw5/boxplot.html`

(c) **(20 pts)** Divide the original training data into 3000 examples for training, remaining for validation. Using your code (the one with the bias), plot the learning curves (training and validation accuracy) for training data sizes of 100, 500, 1000, 2000, and 3000. The learning curves should use the zero-one loss (accuracy).
Hint: The plot has 100, 500, 1000, 2000, and 3000 in the horizontal axis. And two curves: the accuracy over the training data, and the accuracy over the test data.

(d) **(Extra points +10 pts)** Add one extra fully connected layer to the neural network (with the same number of neurons as the hidden layer). Redo (a) and (b).