**Assignment 5 - Neural networks   
Due Monday, April 18th at 11:00 AM CST.**

**Note : No submission will be accepted after 11:00 AM CST, April 25. This is INCLUDING the LATE DAYS.**

**Part A - Programming**

For this part of the assignment, you will be writing code to train and test a single-layer neural network using backpropagation. Specifically, you should assume:

1. Your code is intended for binary classification problems.
2. All of the attributes are numeric.
3. The neural network has connections between input and output nodes with no hidden layers and one bias unit and one output node.
4. For training the neural network, use n-fold **stratified**cross validation (using sigmoid activation function).
5. Set initial weights for all units including bias as 0.1.
6. Use a threshold value of 0.5. If the sigmoidal output is less than 0.5, take the prediction to be the class listed first in the ARFF file in the class attributes section; else take the prediction to be the class listed second in the ARFF file.

**File Format:**

Your program should read files that are in the ARFF format. In this format, each instance is described on a single line. The feature values are separated by commas, and the last value on each line is the class label of the instance. Each ARFF file starts with a header section describing the features and the class labels. Lines starting with '%' are comments. See the link above for a brief, but more detailed description of the ARFF format. Your program needs to handle only numeric attributes, and simple ARFF files (i.e. don't worry about sparse ARFF files and instance weights). Your program can assume that the class attribute is named 'class' and it is the last attribute listed in the header section.   
  
Use the following data set for your program : [sonar.arff](http://pages.cs.wisc.edu/~dpage/cs760/sonar.arff)

**Specifications:**

The program should be callable from command line as follows:   
**neuralnet trainfile num\_folds learning\_rate num\_epochs**  
  
Your program should print the output in the following format for each instance in the source file (in the same order in which the instances appear in the source file)   
**fold\_of\_instance predicted\_class actual\_class confidence\_of\_prediction**  
  
If you are using a language that is not compiled to machine code (e.g. Java), then you should make a small script called neuralnet.sh that accepts the command-line arguments and invokes the appropriate source-code program and interpreter. More instructions below!

**Part B - Analysis**

In this section, you will draw graphs for analysing the performance of neural network (using sonar.arff as the data set) with respect to certain parameters.

1. Plot accuracy of the neural network constructed for 25, 50, 75 and 100 epochs.   
   (With learning rate = 0.1 and number of folds = 10)
2. Plot accuracy of the neural network constructed with number of folds as 5, 10, 15, 20 and 25.   
   (With learning rate = 0.1 and number of epochs = 50)
3. Plot ROC curve for the neural network constructed with the following parameters:   
   (With learning rate = 0.1, number of epochs = 50, number of folds = 10)

Combine all the three graphs in a single PDF file named wisc\_id\_analysis.pdf.

**Submission Instructions**

Create an executable that calls your program. Instructions to create the executable can be found [here](http://pages.cs.wisc.edu/~dpage/cs760/instructions.html).   
  
Collect all files needed to run your code (including the executable) and the PDF containing graphs into a folder named Wiscusername\_HW5.   
Compress the above folder as a zip file named Wiscusername\_HW5.zip.   
  
Upload this zip file under Assignment 5 in the course [moodle](https://ay15-16.moodle.wisc.edu/prod/mod/assign/view.php?id=117401).   
  
**You need to ensure that your code will run, when called from the command line as described above, on the CS department Linux machines.**