Computer Science 300 – Spring 2014
Assignment #4

100 points **Due: Monday, 4/28/14 at 11:59PM**

Part A (50 points)
The code for this part of the assignment is available in the file `A4.zip` on D2L. All questions deal with implementing back propagation for a neural network.

1. (20 points) The back propagation algorithm has been partially implemented for you in `ANN.java`. The `forward_feed` method needs to be completed. Provide the code that runs a forward feed through the network (It can be done in 8-10 lines altogether).

2. (15 points) Similarly, the `train` method has a small block that needs completion. Provide the code to complete this. (Again, it can be done in 4-10 lines altogether). At this point, you should have a complete implementation of back propagation.

3. (15 points) Three training sets, `and`, `or` and `xor` have been provided. In `Driver.java`, choose values for α and β for all three training sets. Your values should be "saved" by making the appropriate modifications in `Driver.java`. You can run the training by giving "`xor`" or the other example sets as a command line argument to the `Driver` class.

Part B (50 points)
This is an investigative assignment in which you will download the machine learning software, WEKA, and use it to compare the performance of various learning algorithms on an identical large data set.

Download WEKA from `http://www.cs.waikato.ac.nz/ml/weka/`. Unzip it to a directory, navigate to that directory and verify that you can start up WEKA by typing

```
java -jar weka.jar
```

Examine the data sets that come with the WEKA distribution in the data directory.

Now go to `http://archive.ics.uci.edu/ml/datasets.html` and download the `Breast Cancer Wisconsin (Diagnostic)` data set. Examine the format this data is in, and convert it to the `.arff` format of WEKA. You may need to write a Java parser/converter for this.

Start up WEKA, choose the "`Explorer`" option and load this data set. Verify that all the instances and their attributes have successfully loaded. Now run the following learning algorithms on this data set, using *10-fold cross validation* to assess performance:

1. Run the Decision Tree learning algorithm on your data set. Turn off the pruning feature when training. Explore WEKA's features to see if you can map the real data set to a nominal data set and improve the performance of your decision tree.

2. Run the Naive Bayes learning algorithm on your data set.

3. Run the MultiLayer perceptron learning algorithm on your data set.

For all algorithms, make sure you have obtained the best training results you can by tuning any necessary parameters. When you are satisfied with your output, copy the output from all three learning algorithms into a *plain text file*. Scanning through your text file, it should be clear where the output of each algorithm ends and the other begins.
At the end of your text file, provide the following:

- in CAPITAL LETTERS, the algorithm (with its parameters) you would choose to classify your data based on the results of your exploration
- and what criterion you based this choice on.

Submission
You may do this assignment with a partner, if you choose. If so, it is expected that you both understands all parts of the code you submit, and you will both receive identical grades.

Submit the following files, zipped up in a single folder called `Assignment4.zip` to the `Assignment4` drop box on D2L by the deadline specified:

1) A `Readme` file with the following information in it:
   a) Name of partner, if any:
   b) Statement that you understood all the code that was submitted, even if done by your partner: (Yes/No)
   c) Part A:
      - Q1: Successfully completed: Yes/No/Yes but...
      - Q2: Successfully completed: Yes/No/Yes but...
      - Q3: Successfully completed: Yes/No/Yes but...

   d) Part B:
      - Successfully completed Decision Trees: Yes/No/Any Issues
      - Successfully completed Naïve Bayes: Yes/No/Any Issues
      - Successfully completed Multilayer Perceptron: Yes/No/Any Issues
2) `ANN.java` for Part A, Question 1 and 2.
3) `Driver.java` for Part A, Question 3.
4) Plain Text Output file for Part 2.