**Capstone Lab 10**

**Neural Nets Classification**

You should have already downloaded and installed Weka.

Download the Iris and Vowel datasets.

(http://dml.cs.byu.edu/~cgc/docs/mldm\_tools/Assignments/Datasets/iris.arff, http://dml.cs.byu.edu/~cgc/docs/mldm\_tools/Assignments/Datasets/vowel.arff)

1. The backpropagation algorithm in Weka is under the “Classify” tab, Click on “Choose”, expand “functions”, and select “MultilayerPerceptron”.

2. Use the backpropagation algorithm on the Iris problem, with a random 70/30 split.

* With a single hidden layer and a fixed number of hidden nodes (of your choice), experiment with different learning rates. Graph test set accuracy over time (i.e., number of iterations/epochs) for several different learning rates. Based on these results, select a reasonable learning rate.
* With the learning rate selected and a single hidden layer, experiment with different numbers of hidden nodes, starting from 1 and adding each time until you see no improvement on the training set's accuracy. For each choice of number of hidden nodes, graph test set accuracy over time.
* Record your best number of hidden nodes (i.e., the one resulting in highest accuracy on the test set).

3. Use the backpropagation algorithm on the Vowel problem, with a random 75/25 split. (Note: Make sure you ignore the "Train or Test" attribute).

* Repeat the above experiments.
* With the learning rate selected, induce a 2-hidden layer neural network with 6 hidden nodes in the first layer and 4 hidden nodes in the second. Graph test set accuracy over time.

4. Using only the best number of hidden nodes as recorded above for 1 hidden layer and the same training/test splits, re-run your backpropagation algorithm with the momentum term option to induce a neural network for both Iris and Vowel. Graph training and test set accuracy over time.

5. Analyze the data you have collected and briefly answer the following questions:

* Discuss the effect of different learning rates on the algorithm's performance.
* Discuss the effect of different numbers of hidden units on the algorithm's performance (1-hidden layer case).
* Compare your recorded best numbers of hidden nodes for each problem with the following heuristic value: *H*=*N*/(10(*I*+*O*)), where *N* is the size of the (training) data set, *I* is the number of network inputs and *O* is the number of outputs.
* How did the momentum term affect the learner's behavior (number of epochs to convergence, final accuracy, etc.)?