**Jonathan Skaggs**

**CS 478 Perceptron**

**Arff Files**

Linearly separable data:

% 1. Title: Testing Not Linearly Separable Database  
  
@RELATION ls  
  
@ATTRIBUTE x Continuous  
@ATTRIBUTE y Continuous  
@ATTRIBUTE class {class1,class2}  
  
@DATA  
-0.8, 0.2, class1  
 0.8, -0.2, class2  
-0.5, -0.4, class1  
 0.6, 0.3, class2  
 0.1, -0.2, class1  
 0.2, -0.1, class2  
-1.0, -0.1, class1  
 1.0, 0.8, class2  
%  
%  
%

Not linearly separable data:

% 1. Title: Testing Linearly Separable Database  
  
@RELATION nls  
  
@ATTRIBUTE x Continuous  
@ATTRIBUTE y Continuous  
@ATTRIBUTE class {class1,class2}  
  
@DATA  
 0.3, 0.2, class1  
-0.4, 0.5, class2  
-0.6, -0.2, class1  
 0.7, -0.4, class2  
-0.3, -0.3, class1  
 0.9, -0.7, class2  
 0.4, 0.6, class1  
-0.4, 0.7, class2  
%  
%  
%

**Stopping Criteria**

Stuff

**Learning Rates**

Try it with a couple different learning rates and discuss the effect of learning rate, including how many epochs are completed before stopping. (For these situations learning rate should have minimal effect, unlike with the Backpropagation lab).

**Graphs**

Graph the instances and decision line for the two cases above (with LR=.1). For all graphs always label the axes!

**Voting Dataset**

Use the perceptron rule to learn this version of the voting task. This particular task is an edited version of the standard voting set, where we have replaced all the “don’t know” values with the most common value for the particular attribute. Randomly split the data into 70% training and 30% test set. Try it five times with different random 70/30 splits. For each split report the final training and test set accuracy and the # of epochs required. Also report the average of these values from the 5 trials. You should update after every instance. Remember to shuffle the data order after each epoch. By looking at the weights, explain what the model has learned and how the individual input features affect the result. Which specific features are most critical for the voting task, and which are least critical? Do one graph of the average misclassification rate vs epochs (0th – final epoch) for the training set. In our helps page is some help for doing graphs. As a rough sanity check, typical Perceptron accuracies for the voting data set are 90%-98%.

**Iris Dataset**

Create 1 perceptron for each pair of output classes, where the training set only contains examples from the 2 classes. Run all perceptrons on novel data and set the class to the label with the most wins (votes) from the perceptrons. In case of a tie, use the net values to decide.