CSCI/ARTI 8950 Machine Learning

Assignment Number 4: Due Thursday 3/23/2006 **(in class)**

You have two options for this assignment. You should choose and do only one. No extra credit will be given for doing both.

1. **Option 1** [**80 points**] Do programming problem 4.11 on page 125 of the text book. You should refer to the text book's web pages for the code, data and tasks to do. The tasks can be found starting on page 3 of the file: **http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-8/faceimages/docs/hw97.ps**

   You should note the following:

   - You are only required to do Part I (required). You should do the work on your own. You can discuss with other students, however, and especially get advice if you run into trouble with compilation or so.
   - You can use the code given and I strongly recommend it or, if you prefer, you can use a suitable **Back-propagation** neural network code of your choice. If you do use another code, you still have to use the given data and do the same tasks. You should also provide me with clear documentation about the network you use so I can grade your assignment in a reasonable amount of time. If you use the given code you should be alert to the fact that it assumes you are running on the CMU file system. You may have to change some directory names to match your local path. I advise you to start with the Makefile and to use gcc instead of cc as your preferred C compiler. You should also explore the use of the "alias" and "ln -s" commands which may help you change program names and directory paths easily on UNIX systems.

2. **Option 2** [**80 points**] For this assignment you need to create or use a suitable **Back-propagation** neural network package. I recommend the WEKA package mentioned in class but there are many other packages available for free download on the web. You should choose a data set from the UCI repository (preferably the same one you used for the decision tree homework so you can compare the performance) for this assignment (please don't forget to mention which one you used!). You should train a neural network using back-propagation. You should experiment to find the best number of hidden units and use a validation set (if the data set is large enough) or cross-validation to stop the training before over-fitting happens. You should turn in your code if you wrote your own or just the name and web location of the package you used as well as the settings and any code modifications. You should describe the network structure and number of nodes in each layer as well as any special settings you used such as momentum terms or special error functions. You should also report the training, validation and testing/cross-validation error as well as trainging time (CPU time is prefered but the number of epochs will suffice).