Assignment 6

CSG120, Fall 2003
Due: Thursday, Dec. 11
Note: This assignment will not be accepted late.

Part I. Computer exercises

For this part you will not need to write any programs of your own. You should use the programs found in the `programs/learn` subdirectory of the course web site. Consult the file `README.txt` to get an idea of what's there and how to run it. (If you have any additional questions about these programs, feel free to ask me as well.)

1. Run the decision tree learning program on the PlayTennis data taken from the Mitchell handout. (This data, suitably formatted for all the learning programs, is found in the file `play-tennis-data.lisp`.) Draw the tree generated by the program and confirm that it matches the one in the handout.

2. Run the perceptron algorithm on the same PlayTennis data (by using the same file `play-tennis-data.lisp`). The program automatically uses a 1-out-of-3 encoding for each of the 3-valued attributes Outlook and Temperature, while using a single node for each of Humidity and Wind. Is the data linearly separable using this representation?

3. Run the perceptron algorithm on a different representation of the PlayTennis data, this time using a single node for each attribute, with the Outlook attribute values encoded as Sunny = 0, Overcast = 1/2, and Rain = 1, while Temperature is encoded as Cool = 0, Mild = 1/2, and Hot = 1. (To do this, simply use the file `play-tennis-alt-data.lisp`.) Is the data in this form linearly separable?

4. Run the backpropagation algorithm on the same encoding of the PlayTennis data used in the previous problem (using 2 hidden units). Briefly comment on the results.

5. There are 22 unseen examples. Pick a few of these and determine how the decision tree, the perceptron network using the 8-dimensional input encoding, and the multilayer network using the 4-dimensional input encoding classify them. In particular, try to find an attribute vector that is classified differently by two of these classifiers. For this attribute vector, which classifier seems to you to give a more reasonable answer?

What you should turn in for this part: Hardcopy of all relevant dribble files and appropriate written commentary recording your observations and answers to the questions above.

Part II. Pencil-and-paper exercises.

6. Give decision trees to represent the following boolean functions:

- $A \wedge \neg B$
- $A \vee (B \wedge C)$
- $(A \wedge B) \vee (C \wedge D)$

7. (a) Design a 2-input perceptron that implements the boolean function $A \vee \neg B.$ (b) Design a 2-layer network of perceptrons that implements $A \ \mathrm{XOR} \ B.$

Part III. (Optional) Extra-credit pencil-and-paper exercises

The motivation for the following problems is the material in Section 18.4, Ensemble Learning (pp. 664-668).

8. Boosting algorithms like AdaBoost (Figure 18.10) require the use of *weighted* training data. Assume that every training example (x_i, y_i) in a given set of training data has a corresponding nonnegative weight w_i. (You may assume these weights are normalized to add up to 1 if you wish, but it may not be necessary.) The more usual situation of unweighted data can be thought of as being a special case where all the data are assumed to have equal weights.

Give a generalization of the information gain criterion that can be used to select the best attribute to split on when creating a decision tree for such weighted training data. (Think of the use of weighted training data as follows: Suppose that a fictitious training set is constructed by replicating all the actual training examples many times. The weight on any particular training example should correspond to the number of times this example appears in this fictitious training set relative to the number of times the other examples appear.)

9. A *decision stump* is a decision tree having just one test, and each of its leaves specifies a definite class associated with the corresponding value of the attribute tested. In what follows, assume that there are just 2 classes (+ and -).

a. How many different possible decision stumps are there for data having n boolean input attributes? Prove your answer.

b. Generalize this to data having n discrete-valued attributes, where for each i, attribute i has r_i possible values. In particular, how many possible decision stumps are there for the PlayTennis instance (i.e., input attribute vector) space?

10. A classifier obtained through the use of a boosting algorithm like AdaBoost works by taking a weighted majority of the classifications of individual classifiers in the ensemble for any given instance (where the fixed weights on the individual classifiers are determined by the boosting algorithm along with the individual classifiers themselves). Here we examine the range of functions that can be represented by the result of a boosting algorithm using decision stump learning as the base learner.

Prove or disprove the following claim: *If all attributes (input and output) are boolean-valued, then the representational power of a weighted majority of decision stumps is exactly the same as the representational power of a perceptron. In other words, the classifications produced by any weighted majority of decision stumps correspond exactly to linearly separable boolean functions.*