CS472 Foundations of Artificial Intelligence
Fall 2000 Assignment 5

Due Monday, November 27 at the beginning of class.

Collaboration: You are allowed to work in groups of 2-4 students for this assignment. Collaboration between groups is not allowed. Working alone is fine. In all cases, you must write up the solutions to the problem set yourself; no collaboration is allowed for the write-up. The last part of this assignment involves programming. **As noted below, only one member of the group should turn in the programming portion for the group.**

1. **Decision Trees (30 points)**

   (a) (20 points) Consider the following (entirely fictional) scenario:

   After much discussion, and in response to increasing complaints from students, the professors in Cornell's CS department have decided to abolish all problem sets and final exams!! In place of the normal coursework, they will use a machine learning algorithm to determine whether or not a student will pass the course. In particular, they decide to use Quinlan's ID3 algorithm for decision tree induction (as discussed in class and in R&N chapter 18) and have begun gathering historical data regarding past Cornell students. The table below shows this historical data. It lists eight (8) examples of passing(+) and failing(-) students in terms of the three features that the department has found to be most important in determining the ultimate success of the student: (1) his/her favorite type of music, (2) hair color, and (3) ability to play ice hockey.

	Class	Hockey Skills	Hair Color	Favorite Music
1	-	expert	blond	classical
2	-	novice	brown	classical
3	+	novice	blond	rap
4	-	novice	brown	rap
5	-	expert	brown	rap
6	+	novice	red	rap
7	-	novice	blond	classical
8	+	expert	blond	rap

   Show the decision tree that would be created by an ID3-style algorithm for decision tree induction given all of the above examples as training instances. Show all of the information gain (or average disorder) computations that you used to compute the tree.

   (b) (10 points) Give an example of a training set where the decision tree learning algorithm using information gain produces a tree much bigger than the smallest possible tree. Explain why this happens.

2. **Computational Learning Theory (30 points)** For the questions in this section, you'll need to read Section 18.6 of R&N.

   (a) (20 points) Show that k-CNF (k literals per clause, k fixed) formulas are PAC learnable. Hint: Start the algorithm with the conjunction of all possible clauses of length k.

   (b) (10 points) Russell & Norvig, problem 18.8.

3. **Neural Network Learning (30 points)**

   (a) (10 points) Construct by hand a neural netowrk that computes the XOR function of two inputs. Be sure to specify what sort of units you are using.

   (b) (20 points) Consider the network in Fig. **??**. We would like to train it to learn the complement of the XOR function of x_1 and x_2. Consider the backpropagation training method as described in lecture or in R&N, chapter 19.

   Let the training set be:

x1	x2	x3	class
0	0	-1	1
0	1	-1	0
1	0	-1	0
1	1	-1	1

Let the initial weights be
$w_{1,4} = 2, w_{2,4} = -2, w_{3,4} = 2, w_{1,5} = 1, w_{2,5} = -1, w_{3,5} = -2, w_{4,6} = 2, w_{5,6} = -2$, and $w_{3,6} = 2$.

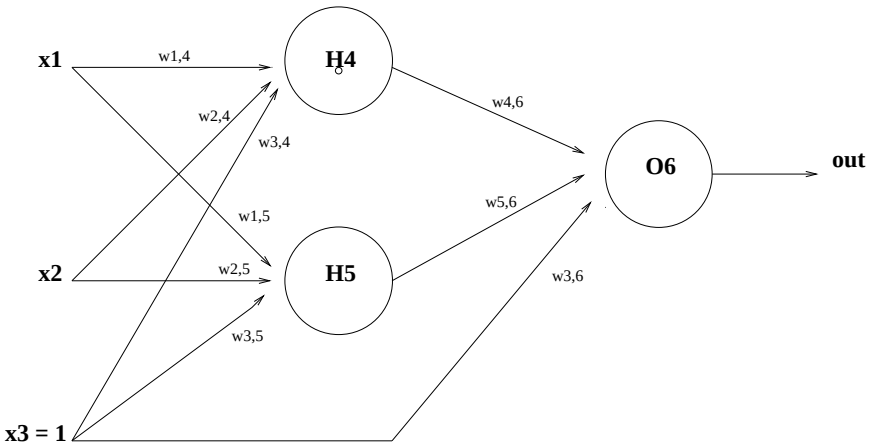

Figure 1: Network for training.

Calculate the new set of weights after considering the first and second examples. (Assume that the units implement the sigmoid function.) Show your intermediate results. Use a learning rate,α, of 1.

4. **Comparison of Learning Algorithms (20 pts)**

You've just designed three machine learning systems that can predict whether the stock market will go up or down tomorrow given as input a set of attribute-value pairs that describe the movement of today's market. The first system is a fully connected, feedforward, backpropagation-trained neural network. The second system is an instance-based (i.e, case-based) algorithm that uses a nearest-neighbor similarity/distance metric for case retrieval. The third is a decision tree trained using the ID3/C4.5 algorithm for top-down induction of decision trees. Assume that all systems achieve equal accuracy on all of your test data. In the end, you manage to convince skeptical traders from a prominent Wall Street trading firm agree to use your systems for stock market prediction.

(a) (5 pts.) Which system will the traders prefer when they want to be able to make predictions about tomorrow's market as quickly as possible? Why?

(b) (5 pts.) Which system will they prefer when they want a justification of the system's reasoning? Why?

(c) (10 pts.) The traders have gathered new training data and want you to update the concept description derived by each system to include the new training data. Compare how each of the three systems could handle this task.

5. **Perceptron Learning (60 points)**

(a) (40 points) Implement a perceptron in the programming language of your choice to learn the following examples. (Note: you may not use any predefined neural network package.)

class	height	hair	eyes
+	short	dark	blue
-	short	light	blue
+	tall	dark	brown
+	tall	dark	blue
-	short	light	brown
+	tall	light	blue
-	short	dark	brown
-	tall	light	brown

Start with initial weights of 0; run your perceptron on the above data set. For this part of the assignment,

i. Experiment with different learning rates. Briefly summarize the results of the experiments.

ii. Using the learning rate of your choice, give the outputs on each example above at the beginning, at the end, and after any three intermediate epochs. (Be sure to state the epoch number and the learning rate.) ****Note*** (11/20/2000)*: Given the data set above, you will probably find that the net converges in only a couple of epochs (possibly regardless of the learning rate). If this is the case, then, for this part of the question, just show the weights after each epoch...we know that there may be fewer than the requested THREE intermediate epochs.

iii. Turn in your well-documented source code.

(b) (10 points) Now consider the following example:

class	A	B	C
-	0	0	0
+	0	0	1
+	0	1	0
-	0	1	1
+	1	0	0
-	1	0	1
-	1	1	0
-	1	1	1

Is it perceptron learnable? If so give a set of weights/threshold that will learn it, otherwise state why it isn't perceptron learnable.

(c) (10 points) What are the advantages/disadvantages of using a **local encoding** vs. a **distributed encoding** for feature vectors? (See p. 577 of Russell & Norvig.) When would you use one of the encoding compared to the other?