15-381 - Fall 2001
Homework 5
Due: Thursday, November 15, 2001

Problem 1: Decision Tree Learning Exercise 18.3, textbook.

Problem 2: Decision Tree Learning

You have a decision tree algorithm and you are trying to figure out which attribute is the best to test on first. You are using the "information gain" metric.

- You are given a set of 128 examples, with 64 positively labeled and 64 negatively labeled.
- There are three attributes, Home_Owner, In_Debt, and Rich.
- For 64 examples, Home_Owner is true. The Home_Owner=true examples are 1/4 negative and 3/4 positve.
- For 96 examples, In_Debt is true. Of the In_Debt=true examples, 1/2 are positive and half are negative.
- For 32 examples, Rich is true. 3/4 of the Rich=true examples are positive and 1/4 are negative

(a) What is the entropy of the initial set of examples?

(b) What is the information gain of splitting on the Home_Owner attribute as the root node?

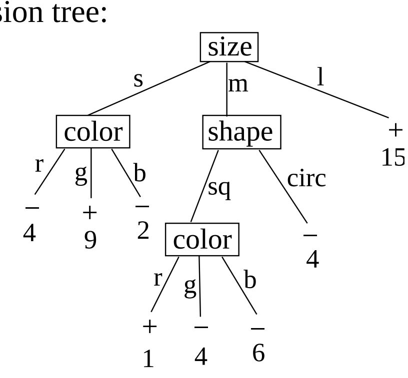
(c) What is the information gain of splitting on the In_Debt attribute as the root node?

(d) What is the information gain of splitting on the Rich attribute as the root node?

(e) Which attribute do you split on?

Problem 3: Decision Tree Learning

Consider the following decision tree:



(a) Using the above decision tree, classify the following examples. Note that the numbers next to the +/- classification tell how many examples reached that point during the training of the tree (you will need that information for the next part of the problem).

Size	Color	Shape	Smell	Classification
small	green	square	pine	
large	blue	square	pine	
medium	green	circle	rotten egg	
medium	red	square	lemon	

(b) Very often, testing examples miss values for some attributes. Consider the following procedure for handling missing values:

Suppose an example X has a missing value for attribute A, and that the decision tree tests for A at a node that X reaches. One way to handle this case is to pretend that the example has all possible values for the attributes, but to weight each value according to its frequency among all of the examples that reach that node in the decision tree. Classification is achieved by following all branches at any node for which a value is missing, and should multiply the weights along each path.

Using the same decision tree, apply the procedure above, and show your work, to classify the following examples with missing attributes:

- blue, circle, rotten egg
- medium, red

(c) Suppose that an attribute splits the set of examples E into subsets E_i, and each subset has p_i positive examples and n_i negative examples. Show that unless the ratio $\frac{p_i}{p_i+n_i}$ is the same for all i, the attribute has strictly positive information gain.

Problem 4: Inductive Learning

(a) What is cross-validation and why is it needed?

(b) Decision tree learning and neural network learning are *supervised* inductive learning algorithms. What does this mean? How do *unsupervised* inductive learning algorithms acquire concepts?

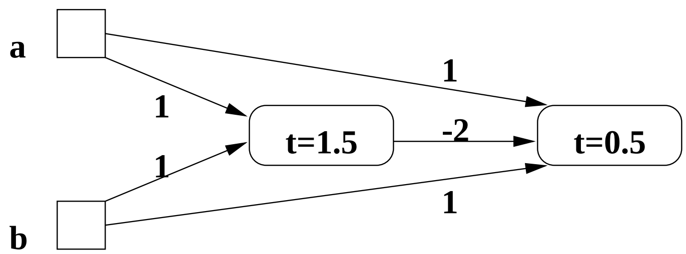
(c) Why does neural network training need to use many times the same set of training examples?

(d) We discussed techniques in decision tree learning to handle attributes with continuous values. In particular, you implemented a one-split method that allowed for a continuous attribute to be split at most once at each decision node. Which candidate thresholds did you consider for the possible split? Discuss the average increase in complexity, if N splits are needed and considered as a function of the average number of candidate split points.

Problem 5: Perceptrons and Neural Nets

(a) Draw a step-thresholded perceptron that computes the boolean implication function, $a \Rightarrow b$.

((b)] Modify the neural network below that computes $a \oplus b$, ($\oplus$ stands for XOR.) so it computes the boolean equivalence, $a \iff b$. Your modification **cannot change the current topological connections**, but it can change weights, thresholds, and/or adding additional units and connections.



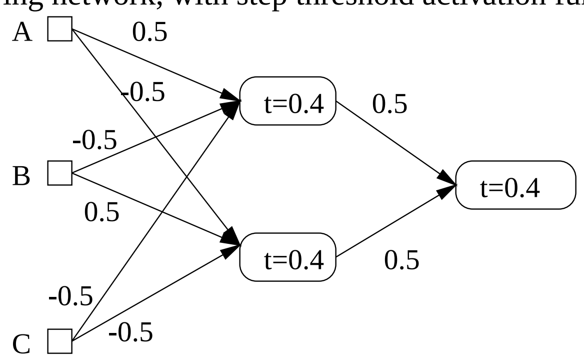
(c) Show that any propositional boolean formula representable as the conjunction of literals can be represented by a one-layer step-thresholded perceptron. *Hint: Consider the conjunction of m positive literals and k negative literals, namely $a_1 \wedge \ldots \wedge a_m \wedge \neg b_1 \wedge \ldots b_k$.*

(d) Show that any propositional boolean formula representable as the disjunction of literals can be represented by a one-layer step-thresholded perceptron. *Hint: Consider the disjunction of m positive literals and k negative literals, namely $a_1 \vee \ldots \vee a_m \vee \neg b_1 \vee \ldots b_k$.*

(c) Can any propositional boolean formula be represented by a thresholded two-layer neural network?

Problem 6: Neural Networks

In class, you have seen that a two layered neural network can represent arbitrary boolean functions. Consider the following network, with step threshold activation functions;



(a) Which of the following functions does it represent. Justify your answer.

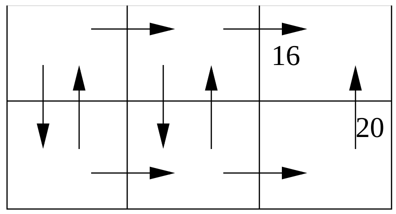
$$(A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee \neg C) \ :$$
$$(A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \ :$$
$$(\neg A \vee B \vee C) \wedge (A \vee \neg B \vee C) \ :$$
$$(\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge C) \ :$$

(b) Show how a two input network would represent $\neg A \oplus \neg B$ using a single hidden unit.

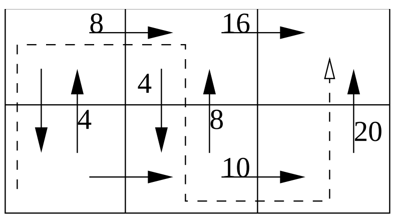
(c) What search strategy does backpropagation use (e.g., depth-first, breadth-first, hill-climbing, etc)? Explain your answer.

Problem 7: Reinforcement Learning

For this problem you will use the following deterministic world. Allowable moves are shown by arrows and the numbers indicate the reward for performing that action. If there is no number, the reward is 0.

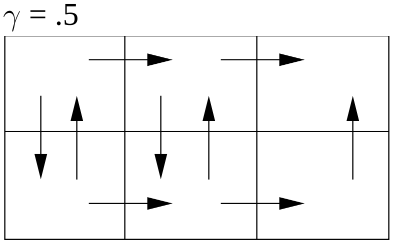


(a) Given the values for Q shown below, show all the changes in the Q estimates when the agent takes the path shown by the dotted line (note that it starts in the left most cell and it follows the direction of the dotted arrow.) Consider $\gamma = .5$.

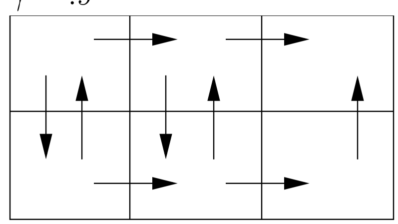


(b) Show all the final optimal Q values for γ=0.5 and for γ=0.9.

γ = .5

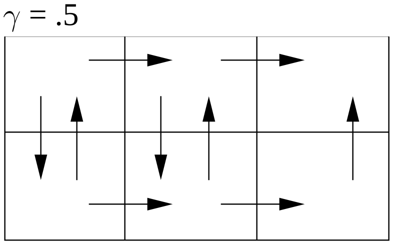


$\gamma = .9$



(c) Given the Q values above, show all the V^* values of **each state**, **and mark one optimal policy,** for each value of γ.

γ = .5



$\gamma = .9$