Homework #7: Chapters 18

The following exercises are due on Friday, May 6 at 1pm. Note, this homework is continued on the reverse side of the paper.

1. *[10 pts.]* Assume you have 3 inputs x_1,x_2,and x_3 each of which can have a value of 0 or 1. Create the smallest decision tree you can for the parity function (i.e., the value of the function is 1 if and only if there is an odd number of ones in the input) on these inputs. Create the smallest decision tree you can for the majority function (i.e., the value is 1 if and only if there are more 1's than 0's in the input).

2. *[30 pts.]* Consider the training set in Table 1 below.

Example	**Type**	**Garage**	**Bedrooms**	**Bathrooms**	**Goal Predicate**
X_1	TownHouse	Yes	2	1.5	Yes
X_2	Condo	No	2	2	No
X_3	Apartment	Yes	2	1	No
X_4	TownHouse	Yes	4	2	No
X_5	TownHouse	Yes	3	1.5	Yes
X_6	Condo	Yes	1	1	No
X_7	TownHouse	No	3	1.5	No
X_8	Apartment	Yes	3	1	No

Table 1. Housing Training Set

   a) *[5 pts.]* Calculate the entropy for the training set.

   b) *[20 pts.]* Calculate the information gain for each of the four attributes: *type*, *garage*, *bedrooms*, and *bathrooms*. **Hint:** If you don't have a calculator capable of doing base 2 logarithms, you can calculate them using the natural logarithm: $\log_2 x = \ln x / \ln 2$.

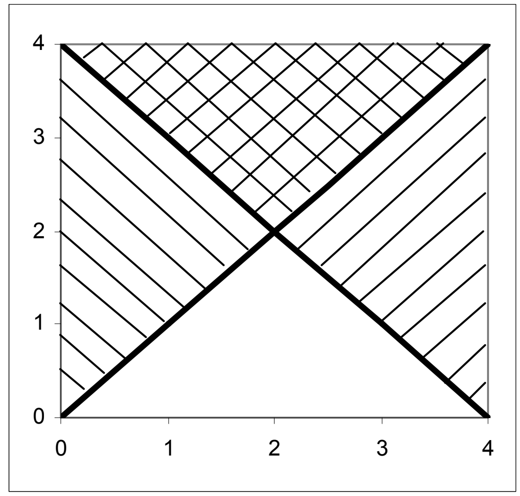
   c) *[5 pts.]* Based on your findings in part b), draw a partial decision tree that includes the attribute on which the first test should be performed and its immediate child nodes. You may use a question mark for the attribute of any nodes that cannot be completely classified by the first attribute test.

3. *[10 pts.]* Suppose you are running a learning experiment on a new algorithm for Boolean classification. You have a data set consisting of 100 positive examples and 100 negative examples. You plan to use leave-one-out cross-validation and compare your algorithm to a baseline function, a simple majority classifier. Note, a majority classifier is given a set of training data and then always outputs the class that is in the majority in the training set, regardless of the input. You expect the majority classifier to score about 50% on leave-one-out-cross-validation, but to your surprise, it scores zero every time. Can you explain why?

4. *[20 pts.]* For this problem you will construct a series of neural networks, building up to a neural network that can recognize a function that is not linearly separable. All of the neural networks will have two inputs: x and y. Both x and y are real numbers and their values are restricted such that $0 \leq x \leq 4$ and $0 \leq y \leq 4$. Use threshold functions for all nodes in your

neural networks and assume that the threshold function returns 1 when its input is ≥ 0 and returns 0 otherwise. Do not use a learning algorithm to develop the neural networks in this problem.

a) *[6 pts.]* Create a single layer feed-forward neural network (a perceptron) to recognize when the input y is greater than or equal to the input x (i.e. y≥x). This corresponds to the area in the graph above the line from the lower left to upper right.

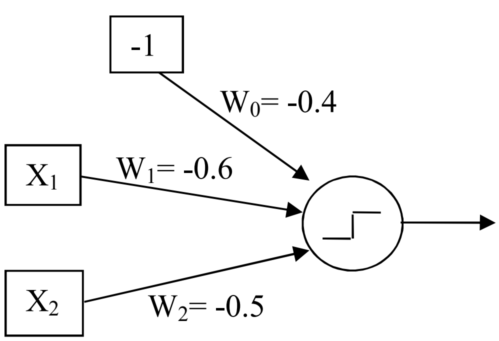


b) *[6 pts.]* Create a single layer feed-forward neural network (a perceptron) to recognize when the input y is greater than or equal to 4 minus the input x (i.e. y≥4-x). This corresponds to the area in the graph above the line from the upper left to lower right.

c) *[8 pts.]* Create a multilayer feed-forward neural network to recognize when y is ≥ both x and 4-x. This corresponds to the double hash marked area in the graph. Obviously, this is not a linearly separable function.

Hints: Approach part c by considering how to combine your answers from parts a and b. Test your neural networks thoroughly to be sure that they perform as expected for various combinations of inputs in the range given (especially near boundaries of the function and input ranges)

5. *[30 pts.]* Use the perceptron learning rule (equation 18.7, p. 724; ch18-learning.pptx, slide #6) to teach the perceptron below to recognize the boolean NAND function. Assume that a **threshold activation function** is being used (i.e., the function returns 1 when its input is ≥ 0 and returns 0 otherwise). For initial weights, use $W_0 = -0.4$, $W_1 = -0.6$, and $W_2 = -0.5$. For the learning rate, use $\alpha = 0.1$. Use only the examples in the table to the right of the network in your learning process (note, Y is the correct output for the example). Stop the training once the weights remain unchanged for one full pass through the examples. The examples must be used in the order given by the table below. Start again with the first example whenever you exhaust all of the examples but have not yet reached the stopping criteria. Show all of the intermediate calculations and values (not just the answer or the updated weights after each example).



Training Set

X_1	X_2	Y
0	1	1
1	1	0
0	0	1
1	0	1