CS 4700: Foundations of Artificial Intelligence
Ungraded Homework
Solutions

1. Neural Networks:
    a. There are 2^{2^n} distinct Boolean functions over *n* inputs. Thus there are 16 distinct Boolean functions over 2 inputs. How many of these are representable by a perceptron?
    b. Imagine you have a single-node neural network with two inputs whose activation function is the logistic function, and with only two data points ((0,0),1) and ((1,1),0). If all weights are initially set to 0, the learning rate is set to 1, and you make two passes through the data, give the values of the weights after each update.
    c. Textbook, problem 18.16a
    d. Textbook, problem 18.22a
    e. Textbook, problem 18.23
2. Clustering:
    a. Give an example where k-means clustering forms different clusters depending on which data points are randomly chosen as the initial centroids for each cluster. Your example should not rely on how you break ties if you have multiple points that are the same distant from multiple centroids. (Or, stated differently, you can make this happen even if the distance between every pair of points is distinct.)
3. MDPs and reinforcement learning:
    a. There are *N* cities along a major highway that forms a big loop. The cities are numbered 1 through N. If you go clockwise from city *i* you get to city *i*+1 and if you go counter-clockwise you get to city *i*-1, except that counter-clockwise from city 1 puts you in city *N* and clockwise from city N puts you in city 1. You start in city 1. Each day you can either
        - do the STAY action, in which case you'll be in that city the following day
        - do the CLOCKWISE or COUNTER-CLOCKWISE actions, in which case with probability pi you'll wind up in the next city in the selected direction the next day, but with probability (1-pi) your car won't start in which case the city you're in will be unchanged the next day.

        All even numbered cities give reward ri = 1, and all odd numbered cities give reward ri=0, if you are in that city at the start of a given day.

        i. If for all cities pi = 1 and the discount factor γ = 0.5:
            1. What is the value of $U^{\pi}(1)$ for the policy π that says to execute STAY in all states?
            2. What is the value of $U^{\pi}(N)$ for this policy?
            3. What is the optimal value U(1) for city 1? What policy does it imply?
            4. What is the optimal value U(N) for city N? What policy does it imply?
        ii. If N=3:
            1. Show the values of U for the first two iterations of value iteration.

2. Show the values of π for the first two iterations of policy iteration, assuming that for each state π is initially set to CLOCKWISE.

b. Consider a problem with 3 states A, B, and C, where A is the initial state, and where you have two actions, X and Y, that you can do in each state. Imagine we use Q-learning on this problem with N=6, learning rate $\alpha = 0.5$, and $R_{MAX} = 10$ (using the f function given in class). Because, for some initial period of time, n<N each time f is called, at the last step of each Q-Learning call many different actions will be tied with value R_{MAX} (where we assume ties are broken at random). Imagine that as a result of executing the Q-Learning procedure you proceed through the following sets of states/actions/rewards:

State	Reward	Action	Resulting State
A	3	X	B
B	2	X	C
C	0	Y	B
B	2	X	C
C	1	X	B

i. What are the final values of Q(s,a) after executing the updates at each step (there should be 5 updates)?
ii. Given these Q values, what policy would they yield for the 3 states?

4. Search:

a. Agents A and B are locations on an NxN grid. They both know where each of them is on the grid. They each have at most five actions they can take, UP, DOWN, LEFT, RIGHT, and STOP, where each action does what you would expect it to do. If any of these actions would cause them to hit a wall then that action is not permissible in that location. They need to wind up in the same location, they don't care where. They both take actions simultaneously – they do not alternate turns. The problem is to come up with a sequence of actions for the two agents that is as short as possible.

i. Formally state this as a (single agent) state space search problem.
ii. Give a (non-trivial) admissible heuristic for this problem.
Which of the following are guaranteed to find a shortest solution?
1. Depth-first search
2. Breadth-first search
3. Hillclimbing using an admissible heuristic
4. A* with a heuristic that returns 0 for all states