CSCI 4350/5350

Milan Zanussi

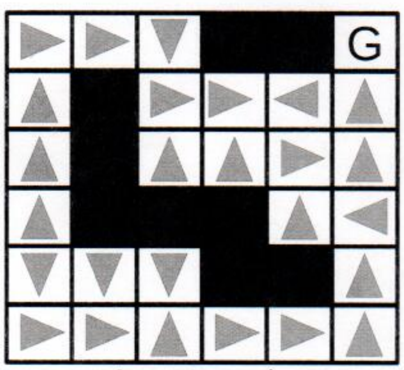
Homework 10

Due: Tue. Dec. 4, 11:00 PM

1. Use *policy iteration* to determine the answers to the following questions. Note that the reward for reaching the goal state, G, is 1, and a reward of 0 is given for all other states [r(G) = 1 and r(s) = 0 for all s != G] and the value of a state, V(s), is the sum of all future rewards obtained starting from that state [V(s) = r(s) + r(s+1) + r(s+2) + … + r(s+n)].

   a. (5 points) Determine the value of all states, **V(s)**, below and write their values in the grid provided, then calculate the new policy, **π_{new}(s)**, and indicate the ***new*** actions in the grid provided:

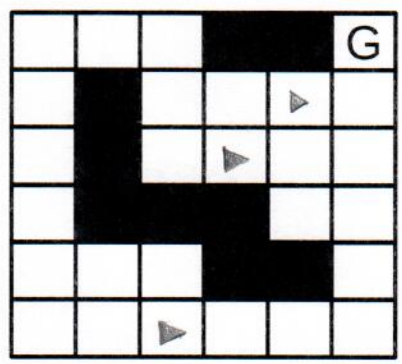
π_{old}(s)



V(s)

0	0	0			G
0		0	0	0	1
0		0	0	1	1
0				1	1
0	0	0			1
0	0	0	1	1	1

π_{new}(s)



   b. (2 points) Will recalculating V(s) and π(s) (performing another iteration of the algorithm) result in additional changes to the policy? Why or why not?

   No, because the new V(s) function will be 1 for each state, and thus the algorithm will not revise any policies.

   c. (2 points) Would this also be true if we defined V(s) using *discounted* future rewards [i.e. $V(s) = \gamma^0 r(s) + \gamma^1 r(s+1) + \gamma^2 r(s+2) + \ldots + \gamma^n r(s+n)$]? Why or why not?

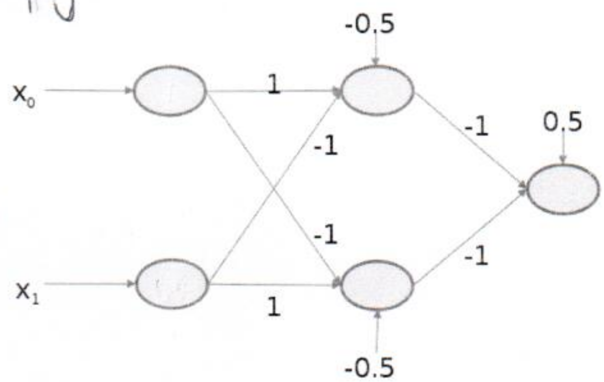
   No, because the discounting would require this problem to have the optimal policy for getting to the goal on each step, so we would need to revise more policies to make all paths optimal.

2. (2 points) An activation function is used to transform a neural units' *net input* into a corresponding *rate code* output.

3. (2 points) What is the most commonly used *learning algorithm* for updating weights in a feed-forward neural network?

Error Backpropagation

4. Given the neural network architecture on the right which uses a *threshold* activation function ($f(net_i) = 0$ if $net_i <= 0$ and $f(net_i) = 1$ if $net_i > 0$), answer the following questions:



a. (1 point each) What is the output of the network for each of the four input patterns:

i. [0 0] 0

ii. [0 1] 1

iii. [1 0] 1

iv. [1 1] 0

b. (1 point) What would be the output of the network for an input pattern of [1, 0.4]?

0

c. (2 points) Does the input/output from question (b) make sense? Why or why not?

In some sense, because the activation function has two possible dichotomies. We can interpret $net_i > 0$ as true and $net_i \leq 0$ as false, and then this neural network just becomes an XOR gate with the previously stated interpretations corresponding to true or false inputs.