PROBLEM 2

(5.1) Consider the following MDP. It has states {0,1,2,3,4} with 4 as the starting state. In every state, you can take one of two possible actions: walk (W) or jump (J). The Walk action decreases the state by one. The Jump action has probability 0.5 of decreasing the state by two, and probability 0.5 of leaving the state unchanged. Actions will not decrease the state below zero: you will remain in state 0 no matter which action you will take (i.e., state 0 is a terminal state). Jumping in state 1 leads to state 0 with probability 0.5 and state 1 with probability 0.5. This definition leads to the following transition functions:

- For states $k \geq 1$, $T(k, W, k-1) = 1$
- For states $k \geq 2$, $T(k, J, k-2) = T(k, J, k) = 0.5$
- For state $k = 1$, $T(k, J, k-1) = T(k, J, k) = 0.5$

The reward gained when taking an action is the distance travelled squared: $R(s, a, s') = (s - s')^2$. The discount factor is $\gamma = 0.5$.

(a) **(4 points)** Suppose we initialize $Q_0(s, a) = 0$ for all $s \in \{0, 1, 2, 3, 4\}$ and $a \in \{J, W\}$. Evaluate the Q-values $Q_1(s, a)$ after exactly one Q-value iteration.

	s_0	s_1	s_2	s_3	s_4
J	0	0.5	2	2	2
W	0	1	1	1	1

(b) **(4 points)** What is the policy that we would derive from $Q_1(s, a)$?

s_1	s_2	s_3	s_4
W	J	J	J

(c) **(2 points)** What are the values $V_1(s)$ corresponding to $Q_1(s, a)$?

s_0	s_1	s_2	s_3	s_4
0	1	2	2	2

(d) **(4 points)** Will the policy change after the second iteration? If your answer is "yes", briefly describe how

No

6.036 Fall 2018 Final Review
Solution Explanations

We provide more detailed explanations for some problems in the Final Review Problems pdf. The solutions are provided in the Final Review Problems Solutions pdf. If you feel an explanation is insufficient, reach out on Piazza or at office hours.

2) a) The W action is deterministic and always gives a reward of 1, except in the s_0 state.

$$Q_1(s_1, W) = Q_1(s_2, W) = Q_1(s_3, W) = Q_1(s_4, W) = 1$$

Note that the Q-values for s_0 will always be 0 for both actions. We can use $V(s) = \max_a Q(s, a)$. Some sample calculations:

$$V_1(s_0) = R(s_0, a) + \gamma[T(s_0, a, s_0)V_0(s_0)] = 0$$

$$\begin{aligned} Q_1(s_1, J) &= R(s_1, J) + \gamma[T(s_1, J, s_0)V_0(s_0) + T(s_1, J, s_1)V_0(s_1)] \\ &= 0.5 * 1^2 + 0.5[0.5 * 0 + 0.5 * 0] = 0.5 \\ Q_1(s_2, J) &= R(s_2, J) + \gamma[T(s_2, J, s_2)V_0(s_2) + T(s_2, J, s_0)V_0(s_0)] \\ &= 0.5 * 2^2 + 0.5[0.5 * 0 + 0.5 * 0] = 2 \end{aligned}$$

s_3 and s_4 are analogous so s_2.

b) For each state, we choose the actions with the higher Q-values (these are taken as V-values) to get the policy.

c) The corresponding V_1-values for s are simply the highest values of the Q_1-values for s and a. From $V(s) = \max_a Q(s, a)$.

d) Even though the V-values change after iteration 2 (with the exception of $Q_i(s_0, a) = V(s_0) = 0$ for all i, a), the relative ordering stays the same, which means we keep the same policy. It should be possible to realize this without doing any calculations beyond the Q_2 values. Here are some sample calculations:

$$V_2(s_0) = R(s_0, a) + \gamma[T(s_0, a, s_0)V_1(s_0)] = 0$$

s_0 is a terminal state, so it's not included in the policy.

$$Q_2(s_1, J) = R(s_1, J) + \gamma[T(s_1, J, s_0)V_1(s_0) + T(s_1, J, s_1)V_1(s_1)]$$

$$= 0.5 * 1^2 + 0.5[0.5 * 0 + 0.5 * 0.5] = 0.5 + 0.5^3 = 0.625$$

$Q_2(s_1, W) = R(s_1, W) + \gamma[T(s_1, W, s_0)V_1(s_0)] = 1 * 1^2 + 0.5[1 * 0] = 1$

since $V(s_0)$ stays 0, $Q(s_1, W)$ stays 1 and $V(s_1)$ stays 1 with W as the optimal action.

$$Q_2(s_2, J) = R(s_2, J) + \gamma[T(s_2, J, s_2)V_1(s_2) + T(s_2, J, s_0)V_1(s_0)]$$

$$= 0.5 * 2^2 + 0.5[0.5 * 2 + 0.5 * 0] = 2 + 0.5 = 2.5$$

$Q_2(s_2, W) = R(s_2, W) + \gamma[T(s_2, W, s_1)V_1(s_1)] = 1 * 1^2 + 0.5[1 * 1] = 1.5$

$$V_2(s_2) = \max\{Q_2(s_2, J), Q_2(s_2, W)\} = 2.5$$

s_3 and s_4 are analogous to s_2, and all three of these states will keep J as the optimal action in future iterations. Therefore the policy does not change.

4) a) The optimal policy will be the one that moves towards higher rewards, so S5 will move left while all the other states will move right (towards S5).

b) one value iteration: 1
two value iterations: 1 + 0.5(10) = 6
infinite value iterations: 1 + $(\sum_{i=1}^{\infty} 0.5^i)(10) = 11$

5) a) Solutions are fine.

b) If we cannot tune hidden layer weights, then we should not be able to solve an XOR-like distribution with just the input and output layers. Note that we can, in theory, solve the other distribution with a single linear separator.

c) The network outputs +1 because $f(z_1) = 2$, $f(z_2) = 0$, $W_1 = W_2 = 1$, and $W_0 = -1$ for $(b, 2)$.

d) Recall that only outgoing weights of b and 2 inputs are active for the input $(b, 2)$ and could potentially be updated. However, since $f(z_2) = 0$, there is no backpropagation through it. Therefore only weights from b to the $f(z_1)$ hidden unit, 2 to the $f(z_1)$ hidden unit, and the $f(z_1)$ unit to the output unit are updated.

6) a) Note that the transition probabilities are all 1, simplifying the calculations. With $\gamma = 0$, none of the future rewards matter, so we can use the rewards given in the table at face value to determine the optimal policy. EXPLORE from HIGH and LOW gives higher rewards than CHARGE from HIGH and LOW. EXPLORE and CHARGE give the same reward from EMPTY, so they are equally preferable.

b) Changing γ from 0 will make future rewards matter. While the optimal policy for HIGH will not change, the policies for LOW and EMPTY could change. Note that the policy for EMPTY won't