Indian Institute of Technology, Kanpur
Department of Electrical Engineering
Introduction to Reinforcement Learning (EE932)
Theory Assignment 4

Deadline: Submission **NOT** Required | 2023-24 Quarter 4 | Max points: 10

1. Tick all the correct options. Q-Learning algorithm is: **(2 pt)**

   (a) Off-Policy
   (b) Online
   (c) Offline
   (d) On-Policy

   Answer: (a), (b)

2. Memory replay buffer and Target Q-network are two important concepts used in DQN. Answer which concept is used to solve the correlated training data problem and which is used for solving the non-stationary target data problem. **(2 pt)**

   Answer:

   Memory replay buffer: Correlation problem
   Target Q-Network: Non-stationarity problem.

3. Consider an MDP with two states A and B and two actions a and b in each state. Assume $\gamma = 0.8$ and $\alpha = 0.2$, $\epsilon = 0.1$. Suppose the initial Q-values are shown in the table below. **(2 pt)**

$Q(A, a)$	2.0
$Q(A, b)$	2.0
$Q(B, a)$	4.0
$Q(B, b)$	2.0

Suppose that we were initially in state A, we took action b, received reward 1, and moved to state B, and then took action b again to get a reward of -1 and landed up in state A. In other words, the trajectory observed so far $S_0, A_0, R_1, S_1, A_1, R_2, S_2$ is

$$A, b, 1, B, b, -1, A.$$

Suppose this was the data corresponding to a Q-learning algorithm. Which Q-table items would have changed after the first update of Q-learning, and what is its new value?

Answer:
Trajectory: $A, b, 1, B, b, -1, A.$

$$Q_{new}(S_t, A_t) = Q_{old}(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_{a'} Q_{old}(S_{t+1}, a') - Q_{old}(S_t, A_t)]$$

The first update of Q-learning corresponds to $Q(S_0, A_0)$, i.e., $Q(A, b)$. The new value can be calculated as shown below.

$$Q_{new}(A, b) = 2 + 0.2[1 + 0.8 * \max\{Q_{old}(B, a), Q_{old}(B, b)\} - Q_{old}(A, b)]$$
$$Q_{new}(A, b) = 2 + 0.2[1 + 0.8 * \max\{4, 2\} - 2] = 2.44$$

4. In the context of reinforcement learning, Monte Carlo function approximation is used to estimate the value function $V(s; \mathbf{w})$. Suppose we represent states with a feature vector $\phi(s)$ and approximate the value function as a linear combination of these features: **(2 pts)**

$$V(s; \mathbf{w}) = \mathbf{w}^\top \phi(s)$$

Given the following feature representations and rewards received by the agent for state s_2 during three episodes:

- **Feature Vector for s_2:** $\phi(s_2) = [1, 2]$
- **Weights Vector:** $\mathbf{w} = [0.5, -0.5]$

Rewards received for s_2 in three episodes:

- **Episode 1:** Reward = -1
- **Episode 2:** Reward = 2
- **Episode 3:** Reward = -2

Using Monte Carlo function approximation, update the weights $\mathbf{w}$ after these three episodes with a learning rate $\alpha = 1$.

What are the updated weights $\mathbf{w}$?

(A) [0.49, -0.51]

(B) [0.52, -0.48]

(C) [0.55, -0.45]

(D) [0.4625, -0.575]

Answer: (D)

Monte Carlo function approximation updates the weights based on the error between the observed return and the estimated value. The update rule is:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \left(G_t - V(s_t; \mathbf{w})\right) \phi(s_t)$$

Where G_t is the return (reward) at time t.

1. **Episode 1:**

$$\begin{aligned} G_1 &= -1 \\ V(s_2; \mathbf{w}) &= 0.5 \times 1 + (-0.5) \times 2 = 0.5 - 1 = -0.5 \\ \delta &= G_1 - V(s_2; \mathbf{w}) = -1 - (-0.5) = -0.5 \\ \mathbf{w} &= [0.5, -0.5] + 0.1 \times (-0.5) \times [1, 2] = [0.5, -0.5] + [-0.05, -0.1] = [0.45, -0.6] \end{aligned}$$

2. **Episode 2:**

$$\begin{aligned} G_2 &= 2 \\ V(s_2; \mathbf{w}) &= 0.45 \times 1 + (-0.6) \times 2 = 0.45 - 1.2 = -0.75 \\ \delta &= G_2 - V(s_2; \mathbf{w}) = 2 - (-0.75) = 2.75 \\ \mathbf{w} &= [0.45, -0.6] + 0.1 \times 2.75 \times [1, 2] = [0.45, -0.6] + [0.275, 0.55] = [0.725, -0.05] \end{aligned}$$

3. **Episode 3:**

$$
\begin{aligned}
G_3 &= -2 \\
V(s_2; \mathbf{w}) &= 0.725 \times 1 + (-0.05) \times 2 = 0.725 - 0.1 = 0.625 \\
\delta &= G_3 - V(s_2; \mathbf{w}) = -2 - 0.625 = -2.625 \\
\mathbf{w} &= [0.725, -0.05] + 0.1 \times (-2.625) \times [1, 2] \\
&= [0.725, -0.05] + [-0.2625, -0.525] \\
&= [0.4625, -0.575]
\end{aligned}
$$

Therefore, the updated weights $\mathbf{w}$ are:

D) [0.4625, -0.575]

5. In the context of reinforcement learning, Q-learning with function approximation is used to estimate the action-value function $Q(s, a; \mathbf{w})$. Suppose we represent state-action pairs with a feature vector $\phi(s, a)$ and approximate the Q-function as a linear combination of these features: **(2pt)**

$$Q(s, a; \mathbf{w}) = \mathbf{w}^\top \phi(s, a)$$

Given the following feature representations and rewards received by the agent during one episode in a two-state MDP (Markov Decision Process):

- **Feature Vector for state** A**:** $\phi(A) = [1, 0]$
- **Feature Vector for state** B**:** $\phi(B) = [0, 1]$
- **Feature Vector for action** x**:** $\phi(x) = [1]$
- **Feature Vector for action** y**:** $\phi(y) = [0]$
- **Initial Weights Vector:** $\mathbf{w} = [0.5, 0.5, 0.5]$

The feature vector for a state-action pair (s, a) is obtained by concatenating the feature vectors of the state and the action.

Rewards and transitions during the episode: $A, x, 2, B, x, -1, A$

Using Q-learning with a learning rate $\alpha = 0.1$ and a discount factor $\gamma = 0.9$, update the weights $\mathbf{w}$ after this episode. What are the updated weights $\mathbf{w}$?

(A) [0.55, 0.45, 0.5]

(B) [0.6, 0.4, 0.5]

(C) [0.65, 0.35, 0.5]

(D) [0.69, 0.4, 0.59]

Answer: (D)

Q-learning with function approximation updates the weights based on the error between the observed return and the estimated Q-value. The update rule is:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \left(r + \gamma \max_{a'} Q(s', a'; \mathbf{w}) - Q(s, a; \mathbf{w}) \right) \phi(s, a)$$

Where:

- r is the reward received after taking action a in state s.
- s' is the next state.

- a' is the action taken in the next state.

1. **Step 1:** $(A, x, 2, B)$

$$
\begin{aligned}
\phi(A, x) &= [\phi(A), \phi(x)] = [1, 0, 1] \\
Q(A, x; \mathbf{w}) &= \mathbf{w}^\top \phi(A, x) = [0.5, 0.5, 0.5]^\top [1, 0, 1] = 0.5 + 0 + 0.5 = 1.0 \\
\phi(B, x) &= [\phi(B), \phi(x)] = [0, 1, 1] \\
Q(B, x; \mathbf{w}) &= \mathbf{w}^\top \phi(B, x) = [0.5, 0.5, 0.5]^\top [0, 1, 1] = 0 + 0.5 + 0.5 = 1.0 \\
\phi(B, y) &= [\phi(B), \phi(y)] = [0, 1, 0] \\
Q(B, y; \mathbf{w}) &= \mathbf{w}^\top \phi(B, y) = [0.5, 0.5, 0.5]^\top [0, 1, 0] = 0 + 0.5 + 0 = 0.5 \\
\max_{a'} Q(B, a'; \mathbf{w}) &= \max\{Q(B, x; \mathbf{w}), Q(B, y; \mathbf{w})\} = \max\{1.0, 0.5\} = 1.0
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{w} &\leftarrow \mathbf{w} + \alpha \left(2 + 0.9 \times 1.0 - 1.0\right) [1, 0, 1] \\
\mathbf{w} &= [0.5, 0.5, 0.5] + 0.1 \times (2 + 0.9 - 1.0)[1, 0, 1] \\
\mathbf{w} &= [0.5, 0.5, 0.5] + 0.1 \times 1.9[1, 0, 1] = [0.5, 0.5, 0.5] + [0.19, 0, 0.19] = [0.69, 0.5, 0.69]
\end{aligned}
$$

2. **Step 2:** $(B, x, -1, A)$

$$
\begin{aligned}
Q(B, x; \mathbf{w}) &= \mathbf{w}^\top \phi(B, x) = [0.69, 0.5, 0.69]^\top [0, 1, 1] = 0 + 0.5 + 0.69 = 1.19 \\
\phi(A, x) &= [\phi(A), \phi(x)] = [1, 0, 1] \\
Q(A, x; \mathbf{w}) &= \mathbf{w}^\top \phi(A, x) = [0.69, 0.5, 0.69]^\top [1, 0, 1] = 0.69 + 0 + 0.69 = 1.38 \\
\phi(A, y) &= [\phi(A), \phi(y)] = [1, 0, 0] \\
Q(A, y; \mathbf{w}) &= \mathbf{w}^\top \phi(A, y) = [0.69, 0.5, 0.69]^\top [1, 0, 0] = 0.69 + 0 + 0 = 0.69 \\
\max_{a'} Q(A, a'; \mathbf{w}) &= \max\{Q(A, x; \mathbf{w}), Q(A, y; \mathbf{w})\} = \max\{1.38, 0.69\} = 1.38
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{w} &\leftarrow \mathbf{w} + \alpha \left(-1 + 0.9 \times 1.38 - 1.19\right) [0, 1, 1] \\
\mathbf{w} &= [0.69, 0.5, 0.69] + 0.1 \times (-1 + 1.242 - 1.19)[0, 1, 1] \\
\mathbf{w} &= [0.69, 0.5, 0.69] + 0.1 \times (-0.948)[0, 1, 1] \\
&= [0.69, 0.5, 0.69] + [0, -0.0948, -0.0948] \\
&= [0.69, 0.4052, 0.5952]
\end{aligned}
$$

Therefore, the updated weights $\mathbf{w}$ are:

D) [0.69, 0.4052, 0.5952]