Q-LEARNING and SARSA

Update Q-table using both Q-learning and SARSA algorithms.

- $Q(s,a) = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} Q(1,1) & Q(1,2) \\ Q(2,1) & Q(2,2) \end{bmatrix}$
- $\alpha = 0.1$ and $\gamma = 0.5$
- $(s, a, r, s') = (1, 2, 3, 2)$
- $a' = \pi_\epsilon(s') = 2$

Q-learning. The Q-learning algorithm consists in:

(1) Initialize $Q(s,a)\ \forall\ s, a$ in an arbitrary way and $Q(end - state,)$ to 0
(2) Iterate and for each episode, after selecting the initial state s:
- Find the action a according to the policy
- Observe reward r and next state s'
- Update Q as: $Q(s,a) = Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
- Set s as s'

So in this specific case we have $a = 2, s = 1$ – this means that $Q(s,a) = Q(1,2) = 2$ –, $r = 3$ and also given the next state $s' = 2$ we have $max_{a'}Q(s',a') = Q(2,2) = 4$.
So putting everything together we obtain:

$$Q(1,2) = Q(1,2) + 0.1[r + 0.5Q(2,2) - Q(1,2)] = 2 + 0.1[3 + 0.5 * 4 - 2] = 2 + 0.3 = 2.3$$

And the new matrix:

$$Q(s,a) = \begin{bmatrix} 1 & 2.3 \\ 3 & 4 \end{bmatrix}$$

SARSA. The SARSA algorithm instead is defined as follow:

(1) Initialize $Q(s,a)\ \forall\ s, a$ in an arbitrary way and $Q(end - state,)$ to 0
(2) Iterate and for each episode, after selecting the initial state s and the action a according to the policy derived from Q:
- Compute the action selected
- Observe reward r and next state s'
- Given the observed s' select action a' according to the policy derived from Q
- Update Q as: $Q(s,a) = Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$
- Set s as s' and a as a'

Differently from before now, with the SARSA algorithm, we don't have to find the $max_{a'}Q(s',a')$ because a' is already give and equal to 2.

Then it's possible to compute:

$$Q(1,2) = Q(1,2) + 0.1[3 + 0.5Q(2,a') - Q(1,2)] = 2 + 0.1[3 + 0.5Q(2,2) - 2] =$$
$$= 2 + 0.1[3 + 0.5 * 4 - 2] = 2.3$$

And again we obtain:

$$Q(s,a) = \begin{bmatrix} 1 & 2.3 \\ 3 & 4 \end{bmatrix}$$

n-STEP ERROR / TD ERROR

Prove that the n-step error can also be written as a sum of TD errors if the value estimates don't change from step to step.

$$G_{t:t+n} - V_{t+n-1}(S_t) = \sum_{k=t}^{t+n-1} \gamma^{k-t}\delta_k$$

where $G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{n-1}R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$

Before starting notice that if the value estimates doesn't change from step to step, then we have that $V_k(S_i) = V_{k+1}(S_i)$. So from now on I'll write $V_k(s)$ just as $V(S)$
Let's compute the $\sum_{k=t}^{t+n-1} \gamma^{k-t}\delta_k$, knowing that $\delta_k = R_{k+1} - \gamma V(S_{k+1}) + V(S_k)$:

$$\begin{aligned}\sum_{k=t}^{t+n-1} \gamma^{k-t}\delta_k = R_{t+1} + \gamma V(S_{t+1}) - V(S_t) + \\ + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) - \gamma V(S_{t+1}) + \\ + \cdots + \\ + \gamma^{n-1}R_{t+n} + \gamma^n V(S_{t+n}) - \gamma^{n-1}V(S_{t+n-1})\end{aligned}$$

It's easy to notice that there are multiple elements that cancel eachother out, giving back the following:

$$\begin{aligned}\sum_{k=t}^{t+n-1} \gamma^{k-t}\delta_k = R_{t+1} + \cancel{\gamma V(S_{t+1})} - V(S_t) + \\ + \gamma R_{t+2} + \cancel{\gamma^2 V(S_{t+2})} - \cancel{\gamma V(S_{t+1})} + \\ + \cdots + \\ + \gamma^{n-1}R_{t+n} + \gamma^n V(S_{t+n}) - \cancel{\gamma^{n-1}V(S_{t+n-1})} = \\ = R_{t+1} - V(S_t) + \gamma R_{t+2} + \cdots + \gamma^{n-1}R_{t+n} + \gamma^n V(S_{t+n}) = \\ \textit{[Since the value is the same for each step]} \\ = R_{t+1} - V_{t+n-1}(S_t) + \gamma R_{t+2} + \cdots + \gamma^{n-1}R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n}) = \\ = G_{t:t+n} - V_{t+n-1}(S_t)\end{aligned}$$