

# 기초부터 시작하는 강화학습

---

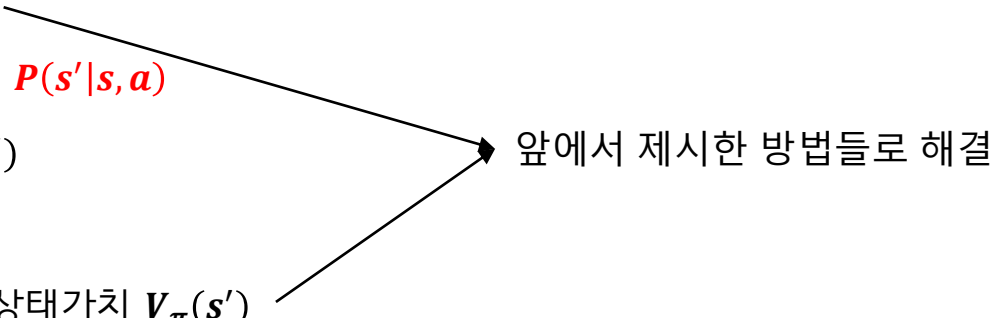
## 몬테카를로, 시간차학습

# 0. 목차

---

- 1) 몬테카를로 방법
  - 1.1) 몬테카를로 방법의 Prediction
  - 1.2) 몬테카를로 방법의 Control
- 2) 시간차 학습
  - 2.1) 시간차 학습의 Prediction
  - 2.2) 시간차 학습의 Control: SARSA(On-policy)
  - 2.3) 시간차 학습의 Control: Q-learning(Off-policy)
  - 2.4) SARSA와 Q-learning의 차이점

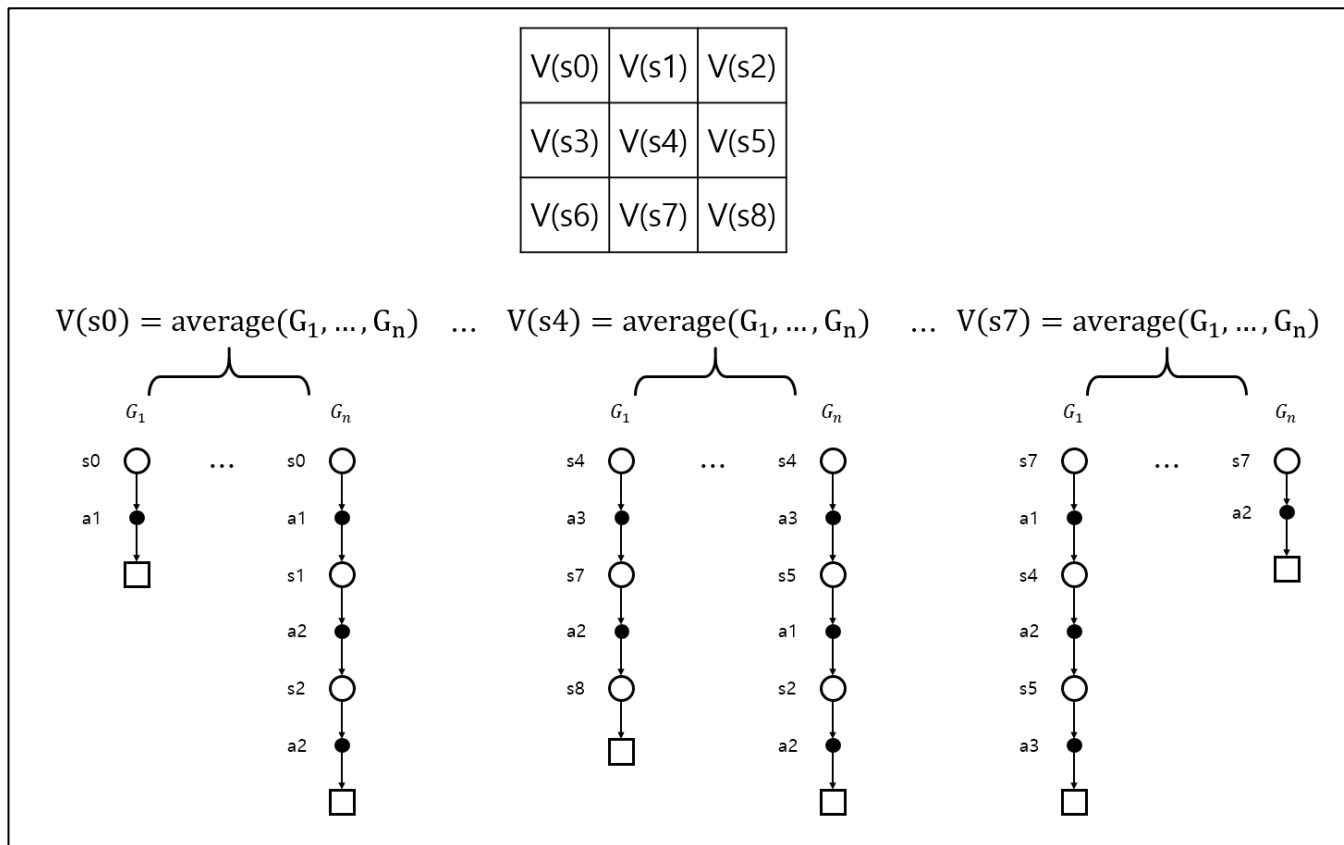
# 1. 몬테카를로 방법

- 상태가치함수:  $V_{\pi}(s) = \sum_a \pi(a|S) \sum_{s'} P(s'|s, a) [r(s, a, s') + \gamma V_{\pi}(s')]$
- 상태가치를 구하기 위해 알아야할 요소들
  - 1) 정책  $\pi(a|S)$
  - 2) 상태전이확률  $P(s'|s, a)$
  - 3) 보상  $r(s, a, s')$
  - 4) 감가율  $\gamma$
  - 5) 다음 상태의 상태가치  $V_{\pi}(s')$

앞에서 제시한 방법들로 해결
- 위와 같은 환경에 대한 정보를 모두 알고 있는 상태에서 강화학습을 푸는 알고리즘: **모델 기반 알고리즘(Model-based algorithm)**
- 상태전이확률을 몰라도 되는 알고리즘: **모델 프리 알고리즘(Model-free algorithm)**
- 모델 프리 알고리즘 종류: **몬테카를로 방법, 시간차 학습**

# 1. 몬테카를로 방법

- 몬테카를로 방법의 상태가치 계산 방법



## 1.1 몬테카를로 방법의 Prediction

- 몬테카를로 방법은 탐색적인 방법을 이용해 **상태가치함수와 행동가치함수를 학습**
- 또한, **경험으로 상태전이확률을 대신**한다.
- 몬테카를로 방법
  - => 모든 상태(도착지점 제외)에서 **에피소드**를 시작하고, 에피소드별로 얻은 수익  $G$ 를 저장
- 1) 모든 단계에서 행동은 가능한 행동들 중 무작위 선택
- 2) 지정된 횟수( $n$ 번)만큼 에피소드가 끝나면 수익  $G$ 들의 평균을 각 상태마다 계산
- 3) 각 상태마다 계산된 평균 수익  $G$ 를 그 상태의 상태가치로 저장
- 몬테카를로 방법의 상태가치함수

$$V(s) = \text{average}(G_1, G_2, \dots, G_n)$$

## 1.1 몬테카를로 방법의 Prediction

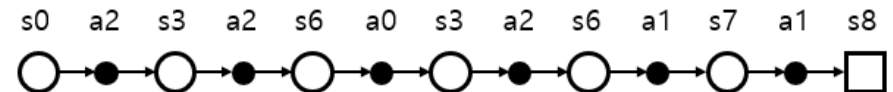
- 몬테카를로 방법이 제대로 학습하기 위한 전제조건

- 1. 모든 상태에서 시작할 수 있어야 한다.
- 2. 에피소드는 반드시 끝이 있어야 한다.

- 에피소드에서 같은 상태를 2번 지나가는 경우

- 1. First-visit 몬테카를로 방법

- 첫 번째로 도착한 상태의 보상만 참고



중복 상태를 가지는 에피소드

- 2. Every-visit 몬테카를로 방법

- 모든 상태의 보상을 수익에 참고

- 1번과 2번 모두 동일한 결과로 수렴하지만, 계산시간에서 차이가 난다.

## 1.1 몬테카를로 방법의 Prediction

---

- 알고리즘: First-visit 몬테카를로 방법의 Prediction
- 입력:
- 초기화:
  - $\pi \leftarrow$  평가할 정책
  - $V \leftarrow$  임의의 상태 가치 함수
  - $\text{Return}(s) \leftarrow$  빈 리스트(모든  $s \in S$ 에 대해)
- 반복:
  - 정책  $\pi$ 를 이용해 에피소드 생성
  - 에피소드에 출현한 각 상태  $s$ 에 대해:
    - $G \leftarrow$  처음  $s$ 에 의해 발생한 수익
    - $G$ 를  $\text{Return}(s)$ 에 추가(append)
    - $V(s) \leftarrow \text{average}(\text{Return}(s))$

## 1.1 몬테카를로 방법의 Prediction

---

- 앞에서 상태가치를 구할 때, 얻은 전체 수익의 평균으로 계산하였다.
- 샘플링이 많아지는 경우 메모리가 많이 차지 하기 때문에 아래와 같이 변경

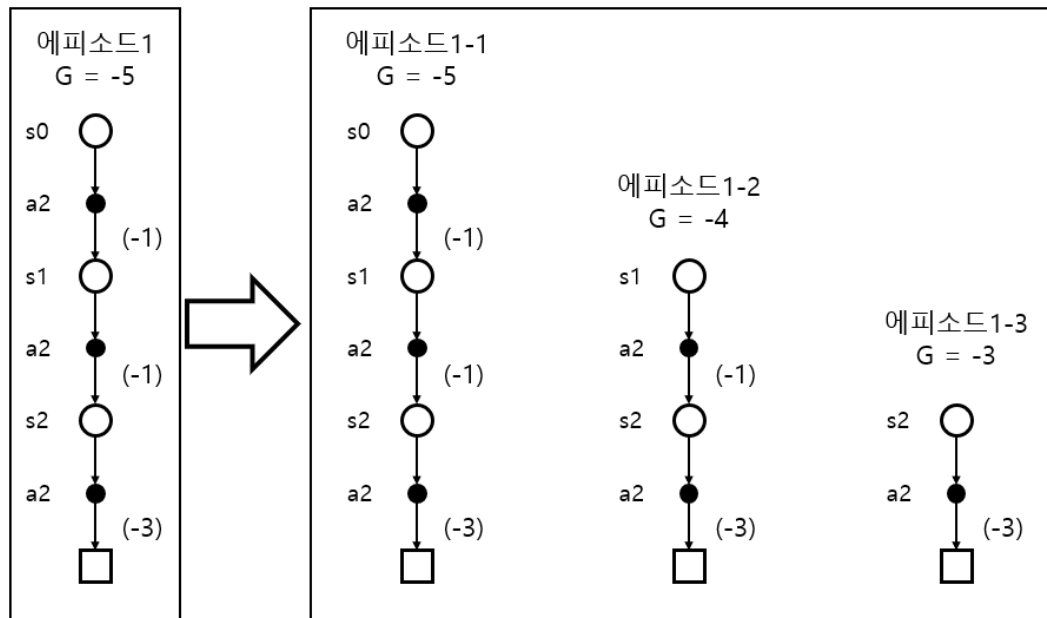
$$V(S_t) \leftarrow V(S_t) + \frac{1}{n+1} [G_t - V(S_t)]$$

- 새로운 상태가치를 기준으로  $[G_t - V(S_t)] = 0$ 이 되도록 상태가치  $V(S_t)$ 를 학습



## 1.1 몬테카를로 방법의 Prediction

- 모든 상태에서 시작을 할 수 없고 처음 상태에서만 시작이 가능한 문제인 경우
  - 에피소드 분리를 통해 해결





## 1.2 몬테카를로 방법의 Control

- 탐욕정책(greedy policy)을 해결한 방법:  $\epsilon$  - greedy 정책

- $$\pi(s, a) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A(s)|} & (a = A^*) \\ \frac{\epsilon}{|A(s)|} & (a \neq A^*) \end{cases}, |A(s)|: \text{상태 } s \text{에서 가능한 행동의 개수}, 0 \leq \epsilon \leq 1$$

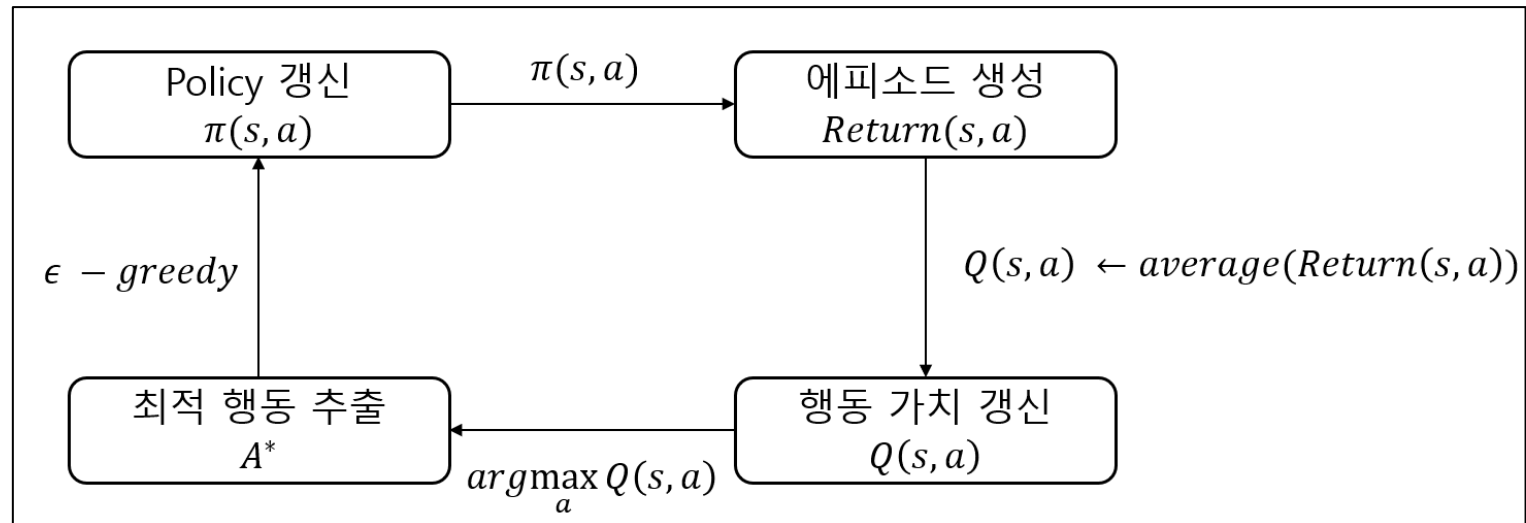
- 예)  $|A(s)| = 4$ 개(동서남북),  $A^* =$  서쪽 이동(최적 행동)
- $\epsilon = 1$ , 최적 행동과 상관없이 모든 행동의 선택될 확률이 0.25로 동일 (즉, 무작위 선택)
- $\epsilon = 0$ , 최적 행동만 선택되어 탐욕정책

## 1.2 몬테카를로 방법의 Control

- 알고리즘: 몬테카를로 방법의 Control
- 모든  $s \in S, a \in A(S)$ 에 대해 초기화:
  - $Q(s, a) \leftarrow$  임의의 값
  - $\text{Return}(s, a) \leftarrow$  빈 리스트
  - $\pi(s, a) \leftarrow$  임의의  $\epsilon$ -탐욕정책
- 무한 반복:
  - (a)  $\pi$ 를 사용해 에피소드 1개 생성
  - (b) 에피소드에 출현한 각  $s, a$ 에 대해:
    - $R \leftarrow s, a$ 의 처음 발생한 수익
    - $R$ 을  $\text{Return}(s, a)$ 에 추가
    - $Q(s, a) \leftarrow \text{average}(\text{Return}(s, a))$
  - (c) 에피소드 안의 각  $s$ 에 대해
    - $a^* \leftarrow \arg \max_a Q(s, a)$
    - 모든  $a^* \in A(S)$ 에 대해
    - $$\pi(s, a) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A(S)|} & (a = a^*) \\ \frac{\epsilon}{|A(S)|} & (a \neq a^*) \end{cases}$$

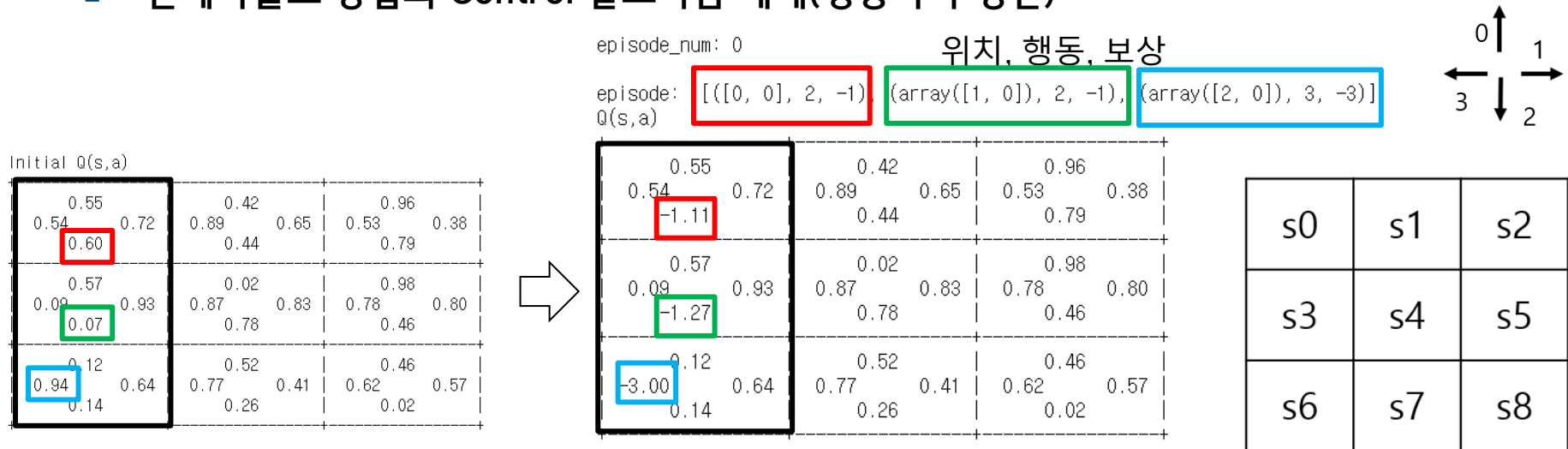
## 1.2 몬테카를로 방법의 Control

- 몬테칼로로 방법의 Control 알고리즘 과정



## 1.2 몬테카를로 방법의 Control

### ■ 몬테카를로 방법의 Control 알고리즘 예제(행동가치 갱신)



S0에서 행동(아래) 갱신

$$G = -1 + (0.09) * -1 + (0.09)^2 * -3$$

$$= -1.1143$$

```
# Incremental mean : Q(s,a) ← average(Return(s,a))
Q_table[i,j,action] += 1 / Q_visit[i,j,action] * (G - Q_table[i,j,action])
```

$$Q(s_0, \text{down}) += \frac{1}{\text{visit}(S_0, \text{down})} * (G - Q(s_0, \text{down}))$$

$$Q(s_0, \text{down}) = 0.6 + \frac{1}{1} * (-1.143 - 0.6)$$

$$= -1.143$$

## 1.2 몬테카를로 방법의 Control

### ■ 몬테카를로 방법의 Control 알고리즘 예제(정책 갱신)

Initial Q(s,a)

0.55	0.42	0.96
0.54 0.72	0.89 0.65	0.53 0.38
0.60	0.44	0.79
0.57	0.02	0.98
0.09 0.93	0.87 0.83	0.78 0.80
0.07	0.78	0.46
0.12	0.52	0.46
0.94 0.64	0.77 0.41	0.62 0.57
0.14	0.26	0.02

Initial optimal\_a

→	←	↑
→	←	↑
←	←	←

Initial Policy

0.20	0.20	0.40	0.20	0.20
0.20 0.40	0.40 0.20	0.20 0.20	0.20 0.20	0.20
0.20	0.20	0.20	0.20	0.20
0.20	0.20	0.40	0.20	0.20
0.20 0.40	0.40 0.20	0.20 0.20	0.20 0.20	0.20
0.20	0.20	0.20	0.20	0.20
0.20	0.20	0.20	0.20	0.20
0.40 0.20	0.40 0.20	0.40 0.20	0.40 0.20	0.20
0.20	0.20	0.20	0.20	0.20

policy

0.25	0.25	0.25
0.25 0.25	0.25 0.25	0.25 0.25
0.25	0.25	0.25
0.25	0.25	0.25
0.25 0.25	0.25 0.25	0.25 0.25
0.25	0.25	0.25
0.25	0.25	0.25
0.25 0.25	0.25 0.25	0.25 0.25
0.25	0.25	0.25

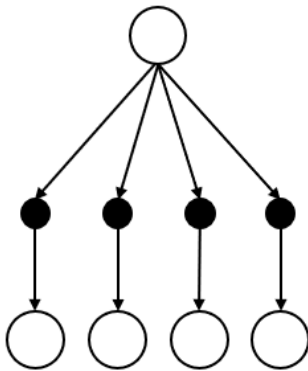


```
# 모든 a ∈ A(S) 에 대해서 :
# 새로 계산된 optimal_a 를 이용해서 행동 선택 확률 policy (π) 갱신
epsilon = 1 - epi / max_episode
```

$$\begin{aligned} \epsilon &= 1 - \frac{\text{episode\_num}}{\text{max\_episode}} \\ &= 1 - \frac{0}{10000} \end{aligned}$$

$$\pi(s, a) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A(S)|} (a = a^*) \\ \frac{\epsilon}{|A(S)|} (a \neq a^*) \end{cases}$$

## 2. 시간차 학습

### ■ 학습방법별 비교

	동적계획법	몬테카를로법	시간차 학습
환경 정보	필요(model-based)	불필요(model-free)	불필요(model-free)
가치함수 계산	상태전이확률	샘플링	샘플링
학습 단위	Time Step	Episode	Time Step
백업 다이어그램			



## 2.1 시간차 학습의 Prediction

- 1) 상태가치함수:  $V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$
- 2) 시간차 학습의 수익:  $G_t = r_{t+1} + \gamma V(S_{t+1})$
- 3) 시간차 학습의 상태가치함수:  $V(S_t) \leftarrow V(S_t) + \alpha[r_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$
- 시간차 학습의 상태가치  $V(S_t)$ 가 수렴하는 조건

$$r_{t+1} + \gamma V(S_{t+1}) = V(S_t)$$

- 시간차 학습의 여러 방법이 있지만, 여기서는 오직 연결된 다음 상태만의 상태가치를 이용해 상태가치를 구하는 방법(TD(0))을 이용한다.
- 시간차 학습 = Temporal Difference learning = TD

## 2.1 시간차 학습의 Prediction

---

- 알고리즘: TD(0)의 Prediction
- 초기화:
  - $\pi \leftarrow$  평가할 정책
  - $V \leftarrow$  임의의 상태 가치 함수
- 각 에피소드에 대해 반복:
  - $s$ 를 초기화
  - 에피소드의 각 스텝에 대해 반복:
    - $a \leftarrow$  상태  $s$ 에서 정책  $\pi$ 에 의해 결정된 행동
    - 행동  $a$ 를 취한 후 보상  $r$ 과 다음 상태  $s'$ 를 관측
    - $V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$
    - $s \leftarrow s'$
  - $s$ 가 마지막 상태라면 종료

## 2.1 시간차 학습의 Prediction

### ■ 상태가치 갱신 예제

반복횟수 1번

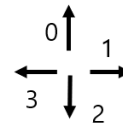
-0.06	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00

epi: 2, action: 1

-0.07	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00

epi: 2, action: 0

-0.07	-0.03	0.00
0.00	0.00	0.00
0.00	0.00	0.00



### ■ 상태가치 갱신

```
# V(s) ← V(s) + α[r + γV(s') - V(s)]
V[pos[0],pos[1]] += alpha * (reward + gamma * V[observation[0],observation[1]] - V[pos[0],pos[1]])
```

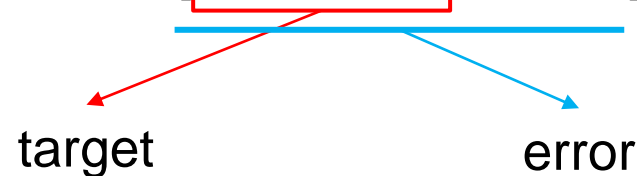
$$\begin{aligned}
 V(s_0) &= V(s_0) + 0.01 * (-1 + 0.9 * V(s_1) - V(s_0)) \\
 &= -0.06 + 0.01 * (-1 + 0.9 * 0 + 0.06) \\
 &= -0.0694
 \end{aligned}$$

s0	s1	s2
s3	s4	s5
s6	s7	s8

## 2.2 시간차 학습의 Control: SARSA(On-policy)

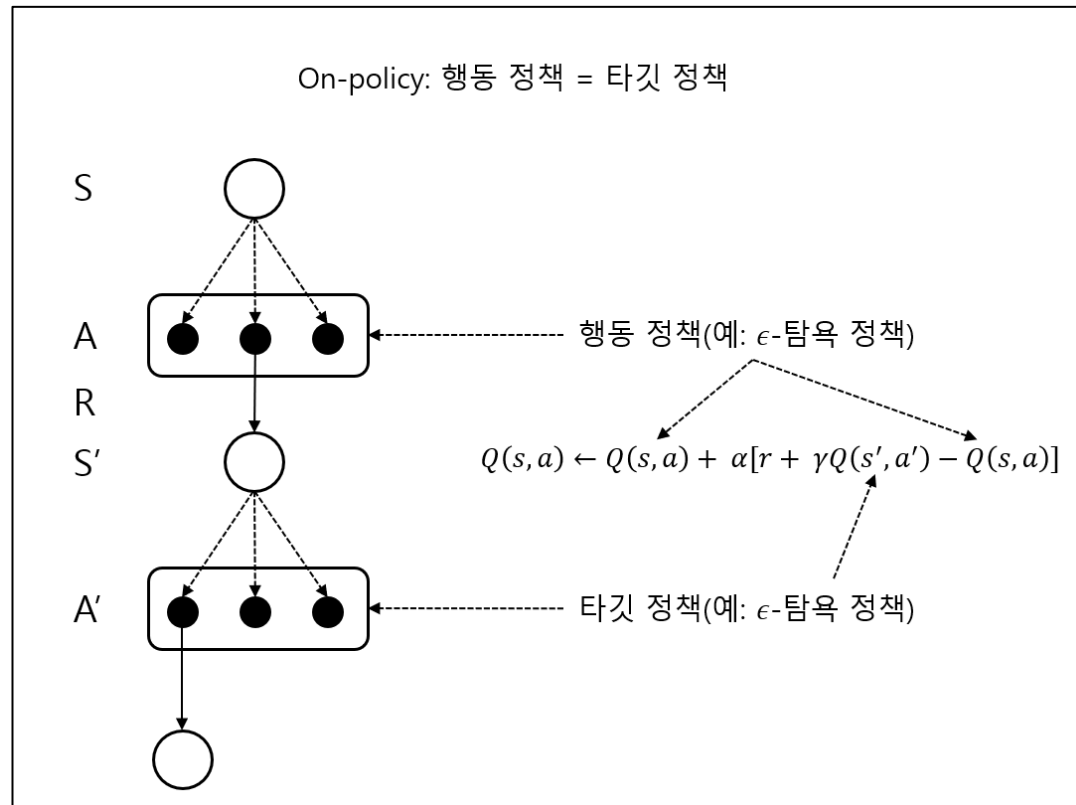
- 최적 정책을 학습하는 TD(0) Control에서는 두 가지의 행동을 선택하는 정책이 있다.
  - 1. 행동 정책(Behavior Policy): 현재 상태  $s$ 에서 가능한 행동들 중에서  $a$ 를 선택하는 정책
  - 2. 타깃 정책(Target Policy): 행동가치함수를 학습하기 위해 다음 상태  $s'$ 에서 가능한 행동들 중에서  $a'$ 를 선택하는 정책

- SARSA의 행동가치함수:  $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$



## 2.2 시간차 학습의 Control: SARSA(On-policy)

- SARSA의 특징
  - 행동 정책과 타겟 정책이 동일



## 2.2 시간차 학습의 Control: SARSA(On-policy)

- 알고리즘: TD(0) SARSA
- 모든  $s \in S$ ,  $a \in A(s)$ 에 대해 초기화:
  - $Q(s, a) \leftarrow$  임의의 값
  - $Q(\text{terminal\_state}, \cdot) = 0$
- 각 에피소드에 대해 반복:
  - $s$ 를 초기화
  - $s$ 에서 행동 정책으로 행동  $a$ 를 선택(예:  $\epsilon$ -탐욕정책)
  - 에피소드의 각 스텝에 대해 반복:
    - 행동  $a$ 를 보상  $r$ 과 다음 상태  $s'$ 를 관측
    - $s'$ 에서 타깃 정책으로 행동  $a'$ 를 선택(예:  $\epsilon$ -탐욕정책)
    - $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$
    - $s \leftarrow s'; a \leftarrow a'$

$s$ 가 마지막 상태라면 종료



## 2.3 시간차 학습의 Control: Q-learning(Off-policy)

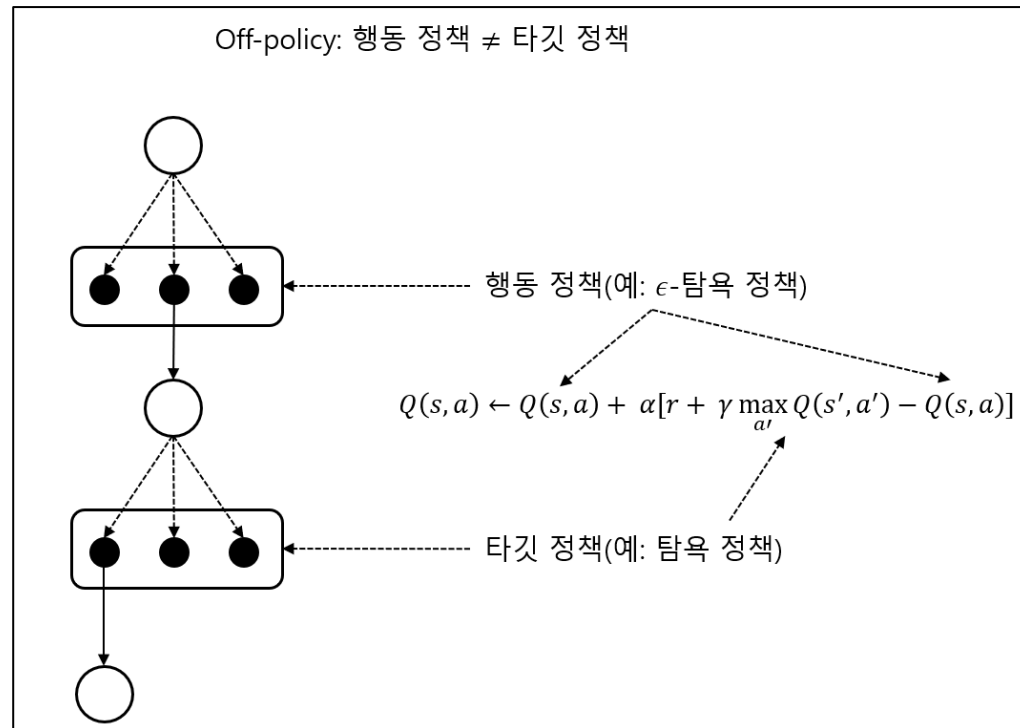
- Q-learning의 행동가치함수:  $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$

- Q-learning의 특징

- 행동 정책과 타깃 정책이 다름

target

error





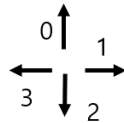
## 2.3 시간차 학습의 Control: Q-learning(Off-policy)

- 알고리즘: TD(0) Q-learning
- 모든  $s \in S$ ,  $a \in A(S)$ 에 대해 초기화:
  - $Q(s, a) \leftarrow$  임의의 값
  - $Q(\text{terminal\_state}, \cdot) = 0$
- 각 에피소드에 대해 반복:
  - $s$ 를 초기화
  - 에피소드의 각 스텝에 대해 반복:
    - $s$ 에서 행동 정책으로 행동  $a$ 를 선택(예:  $\epsilon$ -탐욕정책)
    - 행동  $a$ 를 보상  $r$ 과 다음 상태  $s'$ 를 관측
    - $s'$ 에서 타깃 정책으로 행동  $a'$ 를 선택(예: 탐욕정책)
    - $Q(s, a) \leftarrow Q(s, a) + \alpha \left[ r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$
    - $s \leftarrow s'$

$s$ 가 마지막 상태라면 종료

## 2.3 시간차 학습의 Control: Q-learning(Off-policy)

### ■ Q-learning 예제



pos: [0, 0]  
cur\_action: 2, next\_action: 1

pos: [1 0]  
cur\_action: 2, next\_action: 3

Q-learning : initial\_Q(s,a)

0.55	0.42	0.96
0.54	0.72	0.89
0.60	0.44	0.79

0.57	0.02	0.98
0.09	0.93	0.87
0.07	0.78	0.83

0.12	0.52	0.00
0.94	0.64	0.77
0.14	0.26	0.41



0.55	0.42	0.96
0.54	0.72	0.89
0.53	0.44	0.79

0.57	0.02	0.98
0.09	0.93	0.87
0.07	0.78	0.83

0.12	0.52	0.00
0.94	0.64	0.77
0.14	0.26	0.41



0.55	0.42	0.96
0.54	0.72	0.89
0.53	0.44	0.79

0.57	0.02	0.98
0.09	0.93	0.87
0.05	0.78	0.83

0.12	0.52	0.00
0.94	0.64	0.77
0.14	0.26	0.41



pos: [2 0]  
cur\_action: 3, next\_action: 3

0.55	0.42	0.96
0.54	0.72	0.89
0.53	0.44	0.79

0.57	0.02	0.98
0.09	0.93	0.87
0.05	0.78	0.83

0.12	0.52	0.00
0.64	0.64	0.77
0.14	0.26	0.41

### ■ 행동가치 갱신

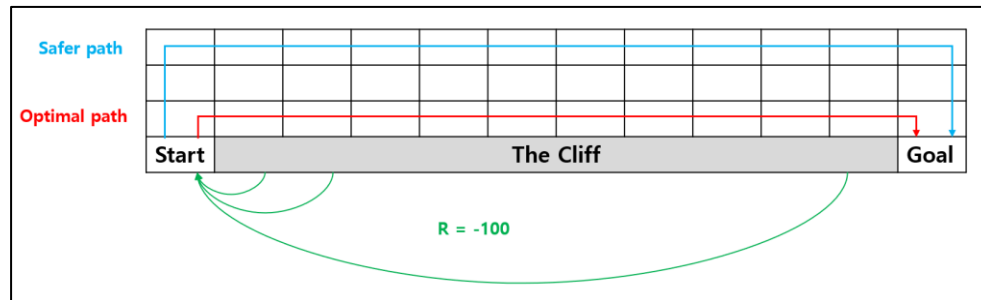
```
# Q(s,a) ← Q(s,a) + α[r + γ max_{a'} Q(s',a') - Q(s,a)]
Q_table[pos[0],pos[1],action] += alpha * (reward + gamma * Q_table[observation[0], observation[1],next_action] -
                                             Q_table[pos[0],pos[1],action])
```

$$\begin{aligned}
 Q(s_0, a) &= Q(s_0, a) + \alpha [r + \gamma \max_{a'} Q(s_3, a') - Q(s_0, a)] \\
 &= 0.6 + 0.1 * [-1 + 0.9 * 0.93 - 0.6] \\
 &= 0.5237
 \end{aligned}$$

s0	s1	s2
s3	s4	s5
s6	s7	s8

## 2.4 SARSA와 Q-learning의 차이점

- Sutton 교수의 책(Reinforcement Learning 2<sup>nd</sup> 예제)



- 4 x 12 격자 존재, 양쪽 끝에 Start 지점과 Goal 지점 존재
- 아래쪽에 절벽이 있어 에이전트가 떨어지면 -100의 보상을 받고 Start 지점으로 이동.
- SARSA는 안전한 경로를 학습하고 Q-learning은 최적 경로를 학습한다.
- Max 값을 추구하는 Q-learning은 절벽에 떨어지는 것을 감수하고 최단 경로를 탐색하지만, SARSA는 절벽 바로 위의 상태들은  $R=-100$ 의 영향을 받아 멀리 돌아가는 안정적인 경로 탐색

## 2.4 SARSA와 Q-learning의 차이점

- (예제) 4 x 7 격자 존재, 양쪽 끝 아래에 Start 지점과 Goal 지점 존재

s0	s1	s2	s3	s4	s5	s6
s7	s8	s9	s10	s11	s12	s13
s14	s15	s16	s17	s18	s19	s20
s21	s22	s23	s24	s25	s26	s27

- s21: Start 지점, s27: Goal 지점, s22 ~ s26: Cliff 지점
- 격자 밖의 보상: -3, Goal 보상: 1, Cliff 보상: -100, 그 외: -1
- Cliff 지점에 도착하면 보상 -100을 받고 Start 지점으로 이동
- Epsilon: 0.8

## 2.4 SARSA와 Q-learning의 차이점

- SARSA가 멀리 돌아 가도록 학습하는 이유
- 현재 agent가 s14에 있다고 가정

s0	s1	s2	s3	s4	s5	s6
$-1 + \frac{-6}{4}$	s8	s9	s10	s11	s12	s13
$-3 + \frac{-10}{4}$ s14	$-1 + \frac{-103}{4}$	s16	s17	s18	s19	s20
$-1 + \frac{-107}{4}$	s22	s23	s24	s25	s26	s27

## 2.4 SARSA와 Q-learning의 차이점

- Q-learning이 최단 거리로 학습하는 이유
- 현재 agent가 s14에 있다고 가정

	s0	s1	s2	s3	s4	s5	s6
	-1 + -1	s8	s9	s10	s11	s12	s13
-3 + -1	s14	-1 + -1	s16	s17	s18	s19	s20
	-1 + -1	s22	s23	s24	s25	s26	s27

- 오른쪽이 Goal 지점과 가깝기 때문에 보상을 더 받을 것이다.

## 2.4 SARSA와 Q-learning의 차이점

### ■ SARSA 방법 적용

SARSA : Q(s,a)																											
-23.62	-22.74	-21.81	-21.00	-12.85	-7.54	-5.80	-23.70	-21.87	-22.03	-20.87	-20.78	-16.70	-18.63	-12.27	-15.39	-7.96	-10.63	-5.24	-5.31	-6.29							
-25.97	-26.15	-34.18	-28.30	-18.31	-7.76	-4.75	-22.27	-21.37	-19.87	-18.78	-12.52	-5.97	-4.80	-27.93	-31.27	-27.58	-30.99	-31.59	-23.17	-26.94	-27.28	-24.67	-13.15	-20.83	-5.52	-7.65	-6.07
-44.01	-64.96	-68.11	-62.32	-54.27	-24.90	-4.19	-31.69	-25.20	-24.40	-24.08	-18.50	-9.27	-3.30	-50.97	-53.75	-44.75	-70.82	-57.21	-50.15	-58.13	-42.65	-60.47	-23.41	-29.02	-2.61	-8.93	-3.18
-80.18	-180.32	-173.84	-161.60	-172.53	-150.47	1.51	-46.89	0.58	0.13	0.59	0.68	0.25	1.07	-83.48	-203.09	0.67	0.93	0.18	0.72	0.00	0.02	0.96	0.27	0.57	0.58	0.56	0.80
-99.51	0.32	0.29	0.83	0.74	0.59	0.92																					
SARSA :optimal policy																											
→	→	→	→	→	→	↓	↑	↑	↑	↑	↑	→	↓	↑	↑	↑	↑	↑	↑	→	→	→	→	→	→	↓	
↑	↑	↑	↑	↑	↑	↓	↑	↑	↑	↑	↑	↑	↓	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↓
↑	↑	↑	↑	↑	↑	↓	↑	↑	↑	↑	↑	↑	↓	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↓
↑	S	The Cliff										G	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↓

The Cliff

## 2.4 SARSA와 Q-learning의 차이점

### ■ Q-learning 방법 적용

Q-learning : $Q(s,a)$																											
-6.70	-6.08	-5.52	-4.89	-4.59	-3.70	-3.02	-6.68	-4.07	-4.65	-3.45	-4.08	-2.73	-3.55	-2.38	-3.04	-1.19	-2.52	-1.40	-1.65	-3.20							
-4.00	-3.33	-2.61	-1.83	-1.38	0.02	0.08	-4.62	-3.32	-4.01	-2.59	-3.34	-1.79	-2.61	-0.97	-1.97	-0.01	-1.08	1.12	-0.29	-1.55							
-3.99	-3.32	-2.58	-1.77	-0.92	0.09	0.96	-5.31	-2.56	-3.31	-1.74	-2.57	-0.82	-1.76	0.22	-0.87	1.39	0.11	2.73	1.13	0.42							
-3.98	-102.98	-102.99	-103.00	-103.02	-103.05	4.26	-3.31	0.58	0.13	0.59	0.68	0.25	3.54	-5.98	-102.98	0.67	0.93	0.18	0.72	0.00	0.02	0.96	0.27	0.57	0.58	3.80	3.49
-5.98	0.32	0.29	0.83	0.74	0.59	2.56	-5.98	0.32	0.29	0.83	0.74	0.59	2.56	-5.98	0.32	0.29	0.83	0.74	0.59	2.56	-5.98	0.32	0.29	0.83	0.74	0.59	2.56
Q-learning : optimal policy																											
↓	↓	↓	↓	→	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓							
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓							
→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→							
↑ S	The Cliff														↓ G												



# 기초부터 시작하는 강화학습

---

## 몬테카를로, 시간차학습

### 감사합니다