강화학습 구현해보기

김권현

숨은원리

2017. 12. 14(목), 18:30-20:20

차례

강화학습 이론

강화학습의 두 줄기

- 가치 기반(Value based)
- 정책 기반(Policy based)
- 액터-크리틱(Actor-Critic)

가치 기반 강화 학습

- 상태 가치 함수 $V_\pi(s)$: 정책 π를 따를 때, 상태 s에서 시작해서 기대되는 총 할인된 보상의 합

$$V_\pi(s) = \mathbb{E}_\pi\left[R + \gamma R' + \gamma^2 R'' + \cdots \middle| S = s\right]$$

- 상태-행동 가치 함수 $Q_\pi(s, a)$: 정책 π를 따를 때, 상태 s, 행동 a에서 기대되는 총 할인된 보상의 합

$$Q_\pi(s, \boldsymbol{a}) = \mathbb{E}_\pi\left[R(s, \boldsymbol{a}) + \gamma R' + \gamma^2 R'' + \cdots \middle| S = s, A = \boldsymbol{a}\right]$$

가치 기반 강화 학습: 최적의 정책 구하기

- **점진적 정책 개선Policy Iteration** : 주어진 정책에 대해 상태-행동 가치 함수를 구하고, 정책을 개선하고, 다시 상태-행동 가치 함수를 구하는 식으로 반복한다.
- **점진적 가치 개선Value Iteration** : 벨만 최적 방정식을 만족하는 가치 함수 값을 점진적 갱신을 통해 구한다.

$$Q_*(s, a) = \mathbb{E}\left[R + \gamma \max_{a'} Q_*(S', a') \middle| S = s, A = a\right]$$

결정론적 환경의 예

상태state	행동action		
	a	b	c
A	$30(\to B)$	$-30(\to C)$	$10(\to A)$
B	$-20(\to C)$	$30(\to B)$	$-10(\to A)$
C	$20(\to A)$	$-10(\to A)$	$10(\to C)$

- 보상reward $r(s, a)$, $r(s, a, s')$
- 할인율discount factor : 미래의 보상과 현재의 보상을 비교하는 방법 (이자, 인플레이션, 사망율 등의 해석)
- 반환값return : 즉각보상과 할인된 미래 보상의 총합
- 정책policy : 주어진 상태에서 어떤 행동을 할 것인가? 결정론적 정책 $\pi(s)$, 확률적인 정책$\pi(a|s)$
- 끝없이 지속되는 과제continous task, 끝이 있는 과제episodic task

결정론적 환경의 예

상태state	행동action		
	a	b	c
A	$30(\to B)$	$-30(\to C)$	$10(\to A)$
B	$-20(\to C)$	$30(\to B)$	$-10(\to A)$
C	$20(\to A)$	$-10(\to A)$	$10(\to C)$

결정론적 정책의 예.[1]

$$\pi_1(A) = a, \pi_1(B) = a, \pi_1(C) = a$$

이때, 상태 A에서 즉각보상과 할인된 미래 보상의 총합은,

$$V_1(A) = R(A, a) + \gamma^1 R(B, a) + \gamma^2 R(C, a) + \gamma^3 R(A, a) + \cdots$$

[1]확률적 정책으로 표현한다면, $\pi_1(a|A) = 1, \pi_1(b|A) = 0, \pi_1(c|A) = 0, \pi_2(a|B) = 1, \pi_1(b|B) = 0, \pi_1(c|B) = 0, \pi_1(a|C) = 1, \pi_1(b|C) = 0, \pi_1(c|C) = 0$

결정론적 환경의 예: 가치 함수

상태state	행동action		
	a	b	c
A	$30(\to B)$	$-30(\to C)$	$10(\to A)$
B	$-20(\to C)$	$30(\to B)$	$-10(\to A)$
C	$20(\to A)$	$-10(\to A)$	$10(\to C)$

정책 π_1 을 따를 때, 상태 A, B, C의 가치 함수를 구해보면,

$$V_1(A) = R(A,a) + \gamma^1 R(B,a) + \gamma^2 R(C,a) + \gamma^3 R(A,a) + \gamma^4 R(B,a) + \gamma^5 R(C,a) + \cdots$$
$$V_1(B) = R(B,a) + \gamma^1 R(C,a) + \gamma^2 R(A,a) + \gamma^3 R(B,a) + \gamma^4 R(C,a) + \gamma^5 R(A,a) + \cdots$$
$$V_1(C) = R(C,a) + \gamma^1 R(A,a) + \gamma^2 R(B,a) + \gamma^3 R(C,a) + \gamma^4 R(A,a) + \gamma^5 R(B,a) + \cdots$$

결정론적 환경의 예: 가치 함수

상태state	행동action		
	a	b	c
A	$30(\rightarrow B)$	$-30(\rightarrow C)$	$10(\rightarrow A)$
B	$-20(\rightarrow C)$	$30(\rightarrow B)$	$-10(\rightarrow A)$
C	$20(\rightarrow A)$	$-10(\rightarrow A)$	$10(\rightarrow C)$

정책 π_1 을 따르를 때, 상태 A, B, C의 가치 함수를 구해보면,

$$V_1(A) = R(A,a) + \gamma^1 R(B,a) + \gamma^2 R(C,a) + \gamma^3 R(A,a) + \gamma^4 R(B,a) + \gamma^5 R(C,a) + \cdots$$
$$V_1(B) = R(B,a) + \gamma^1 R(C,a) + \gamma^2 R(A,a)) + \gamma^3 R(B,a) + \gamma^4 R(C,a) + \gamma^5 R(A,a) + \cdots$$
$$V_1(C) = R(C,a) + \gamma^1 R(A,a) + \gamma^2 R(B,a) + \gamma^3 R(C,a) + \gamma^4 R(A,a) + \gamma^5 R(B,a) + \cdots$$

결정론적 환경의 예: 가치 함수

$$V_1(A) = R(A,a) + \gamma^1 R(B,a) + \gamma^2 R(C,a) + \gamma^3 R(A,a) + \gamma^4 R(B,a) + \gamma^5 R(C,a) + \cdots$$
$$V_1(B) = R(B,a) + \gamma^1 R(C,a) + \gamma^2 R(A,a)) + \gamma^3 R(B,a) + \gamma^4 R(C,a) + \gamma^5 R(A,a) + \cdots$$
$$V_1(C) = R(C,a) + \gamma^1 R(A,a) + \gamma^2 R(B,a) + \gamma^3 R(C,a) + \gamma^4 R(A,a) + \gamma^5 R(B,a) + \cdots$$

반복적인 부분을 활용하면, 다음과 같이 다시 쓸 수 있다.

$$\begin{aligned} V_1(A) &= R(A,a) + \gamma \cdot \boldsymbol{R(B,a)} + \gamma \cdot \boldsymbol{\gamma^1 R(C,a)} + \gamma \cdot \boldsymbol{\gamma^2 R(A,a)} + \gamma \cdot \boldsymbol{\gamma^3 R(B,a)} + \\ &= R(A,a) + \gamma(\boldsymbol{R(B,a) + \gamma^1 R(C,a) + \gamma^2 R(A,a) + \gamma^3 R(B,a) + \cdots}) \end{aligned}$$
$$\boldsymbol{V_1(B) = R(B,a) + \gamma^1 R(C,a) + \gamma^2 R(A,a) + \gamma^3 R(B,a) + \cdots}$$

결정론적 환경의 예: 벨만 가치 함수 방정식

$$
\begin{aligned}
V_1(A) &= R(A,a) + \gamma \cdot \boldsymbol{R(B,a)} + \gamma \cdot \boldsymbol{\gamma^1 R(C,a)} + \gamma \cdot \boldsymbol{\gamma^2 R(A,a)} + \gamma \cdot \boldsymbol{\gamma^3 R(B,a)} + \\
&= R(A,a) + \gamma(\boldsymbol{R(B,a)} + \boldsymbol{\gamma^1 R(C,a)} + \boldsymbol{\gamma^2 R(A,a)} + \boldsymbol{\gamma^3 R(B,a)} + \cdots) \\
\boldsymbol{V_1(B)} &= \boldsymbol{R(B,a)} + \boldsymbol{\gamma^1 R(C,a)} + \boldsymbol{\gamma^2 R(A,a)} + \boldsymbol{\gamma^3 R(B,a)} + \cdots
\end{aligned}
$$

이를 활용하면, 다음과 같은 상태 가치 함수 방정식을 얻는다.

$$
\begin{aligned}
V_1(A) &= R(A,a) + \gamma \boldsymbol{V_1(B)} \\
\boldsymbol{V_1(B)} &= R(B,a) + \gamma V_1(C) \\
V_1(C) &= R(C,a) + \gamma V_1(A)
\end{aligned}
$$

벨만 방정식을 풀기

상태state	행동action		
	a	b	c
A	$30(\to B)$	$-30(\to C)$	$10(\to A)$
B	$-20(\to C)$	$30(\to B)$	$-10(\to A)$
C	$20(\to A)$	$-10(\to A)$	$10(\to C)$

우리는 $R(A,a)$, $R(B,a)$, $R(C,a)$를 모두 알고 있으므로, 만약 할인율 $\gamma = 0.9$라면,

$$\begin{aligned} V_1(A) &= 30 + 0.9 V_1(B) \\ V_1(B) &= -20 + 0.9 V_1(C) \\ V_1(C) &= 20 + 0.9 V_1(A) \end{aligned}$$

벨만 방정식을 풀기(행렬 표현)

앞에서 구한 벨만 방정식

$$\begin{aligned} \boldsymbol{V}_1(\boldsymbol{A}) &= 30 + 0.9\boldsymbol{V}_1(\boldsymbol{B}) \\ \boldsymbol{V}_1(\boldsymbol{B}) &= -20 + 0.9\boldsymbol{V}_1(\boldsymbol{C}) \\ \boldsymbol{V}_1(\boldsymbol{C}) &= 20 + 0.9\boldsymbol{V}_1(\boldsymbol{A}) \end{aligned}$$

이를 행렬식으로 표현해보자. $\boldsymbol{V}_1 = \begin{pmatrix} \boldsymbol{V}_1(\boldsymbol{A}) \\ \boldsymbol{V}_1(\boldsymbol{B}) \\ \boldsymbol{V}_1(\boldsymbol{C}) \end{pmatrix}$ 로 놓으면,

$$\begin{aligned} \boldsymbol{V}_1 &= \begin{pmatrix} R(A,a) \\ R(B,a) \\ R(C,a) \end{pmatrix} + \gamma \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \boldsymbol{V}_1 \\ &= \boldsymbol{R}_1 + 0.9\boldsymbol{T}_1\boldsymbol{V}_1. \end{aligned}$$

벨만 방정식을 푸는 두 가지 방법

$$\boldsymbol{V}_1 = \boldsymbol{R}_1 + 0.9\boldsymbol{T}_1\boldsymbol{V}_1$$

- 행렬 방정식을 푼다.

$$\begin{aligned}(\boldsymbol{I} - 0.9\boldsymbol{T}_1)\boldsymbol{V}_1 &= \boldsymbol{R}_1 \\ \boldsymbol{V}_1 &= (\boldsymbol{I} - 0.9\boldsymbol{T}_1)^{-1}\boldsymbol{R}_1\end{aligned}$$

- 반복적인 갱신update을 이용한다.

$$\begin{aligned}\boldsymbol{V}_1^{(1)} &= \boldsymbol{R}_1 + 0.9\boldsymbol{T}_1\boldsymbol{V}_1^{(0)} \\ \boldsymbol{V}_1^{(2)} &= \boldsymbol{R}_1 + 0.9\boldsymbol{T}_1\boldsymbol{V}_1^{(1)} \\ &\vdots\end{aligned}$$

벨만 방정식의 해: 반복적인 갱신

첫 번째 갱신에서 가치 함수 V_1 은 다음과 같이 결정된다.

$$V_1^{(1)}(A) = R(A,a) + \gamma V_1^{(0)}(B)$$
$$V_1^{(1)}(B) = R(B,a) + \gamma V_1^{(0)}(C)$$
$$V_1^{(1)}(C) = R(C,a) + \gamma V_1^{(0)}(A)$$

두 번째 갱신에서 가치 함수 V_2 은 다음과 같다.

$$V_1^{(2)}(A) = R(A,a) + \gamma V_1^{(1)}(B)$$
$$V_1^{(2)}(B) = R(B,a) + \gamma V_1^{(1)}(C)$$
$$V_1^{(2)}(C) = R(C,a) + \gamma V_1^{(1)}(A)$$

그리고 다음과 같이 다시 쓸 수 있다.

$$\begin{aligned} V_1^{(2)}(A) &= R(A,a) + \gamma(R(B,a) + \gamma V_1^{(0)}(C)) \\ &= R(A,a) + \gamma R(B,a) + \gamma^{\mathbf{2}} V_1^{(0)}(C) \end{aligned}$$

상태-행동 가치 함수Q-function

주어진 정책, 주어진 상태에서 "행동"을 평가하기

$$V_1(A) = R(A,a) + \gamma^1 R(B,a) + \gamma^2 R(C,a) + \gamma^3 R(A,a) + \gamma^4 R(B,a) + \gamma^5 R($$
$$Q_1(A,a) = R(A,a) + \gamma^1 R(B,a) + \gamma^2 R(C,a) + \gamma^3 R(A,a) + \gamma^4 R(B,a) + \gamma^5 R($$
$$Q_1(A,b) = R(A,b) + \gamma^1 R(C,a) + \gamma^2 R(A,a) + \gamma^3 R(B,a) + \gamma^4 R(C,a) + \cdots$$
$$Q_1(A,c) = R(A,c) + \gamma^1 R(A,a) + \gamma^2 R(B,a) + \gamma^3 R(C,a) + \gamma^4 R(A,a) + \gamma^5 R($$

이를 가치 함수를 활용하여 표현해보면,

$$\begin{aligned} V_1(A) &= R(A,a) + \gamma^1 V_1(B) \\ Q_1(A,a) &= R(A,a) + \gamma^1 V_1(B) \\ Q_1(A,b) &= R(A,b) + \gamma^1 V_1(C) \\ Q_1(A,c) &= R(A,c) + \gamma^1 V_1(A) \end{aligned}$$

상태-행동 가치 함수Q-function 벨만 방정식

$$V_1(A) = R(A, a) + \gamma^1 V_1(B)$$
$$Q_1(A, a) = R(A, a) + \gamma^1 V_1(B)$$
$$Q_1(A, b) = R(A, b) + \gamma^1 V_1(C)$$
$$Q_1(A, c) = R(A, c) + \gamma^1 V_1(A)$$

여기서 $Q_1(A, a) = V_1(A)$를 활용하면, 다음의 방정식을 구할 수 있다.

$$Q_1(A, a) = R(A, a) + \gamma^1 Q_1(B, a)$$
$$Q_1(A, b) = R(A, b) + \gamma^1 Q_1(C, a)$$
$$Q_1(A, c) = R(A, c) + \gamma^1 Q_1(A, a)$$
$$Q_1(B, a) = R(B, a) + \gamma^1 Q_1(C, a)$$
$$Q_1(B, b) = R(B, b) + \gamma^1 Q_1(B, a)$$
$$\vdots$$

상태-행동 가치 함수Q-function 벨만 방정식의 해 구하기

- 역행렬을 이용하여 해를 구하기
- 반복적으로 갱신update하기

상태-행동 가치 함수Q-function 활용하여 최상의 정책 구하기 : 정책을 반복적으로 개선하기

Table: 정책 π_1 을 따를 때 Q-함수값

state	action		
	a	*b*	*c*
A	**104.**059	**72.**28782	**103.**6531
B	**82.**28782	**104.**059	**83.**65314
C	**113.**6531	**64.**05904	**112.**2878

Q-함수를 활용하여 개선된 정책 π_2 를 다음과 같이 정할 수 있다.

$$\pi_2(A) = a, \pi_2(B) = \mathbf{b}, \pi_2(C) = a$$

그리고 이에 대해 다시 Q-함수를 구하고, 다시 정책을 개선하는 과정을 반복함으로써 최상의 정책에 이를 수 있다.

상태-행동 가치 함수Q-function를 활용하여 최상의 정책 구하기 : 최적 벨만 방정식

$$
\begin{aligned}
Q_*(A,a) &= R(A,a) + \gamma^1 Q_*(B,?) \\
Q_*(A,b) &= R(A,b) + \gamma^1 Q_*(C,?) \\
Q_*(A,c) &= R(A,c) + \gamma^1 Q_*(A,?) \\
Q_*(B,a) &= R(B,a) + \gamma^1 Q_*(C,?) \\
Q_*(B,b) &= R(B,b) + \gamma^1 Q_*(B,?)
\end{aligned}
$$

최상의 정책이라면 행동은 $Q_*(s,a)$를 최대로 하는 행동을 선택할 것이다.

$$
\begin{aligned}
Q_*(A,a) &= R(A,a) + \gamma^1 max_{a'} Q_*(B,a') \\
Q_*(A,b) &= R(A,b) + \gamma^1 max_{a'} Q_*(C,a') \\
Q_*(A,c) &= R(A,c) + \gamma^1 max_{a'} Q_*(A,a') \\
Q_*(B,a) &= R(B,a) + \gamma^1 max_{a'} Q_*(C,a') \\
Q_*(B,b) &= R(B,b) + \gamma^1 max_{a'} Q_*(B,a')
\end{aligned}
$$

최적 벨만 방정식의 해 구하기

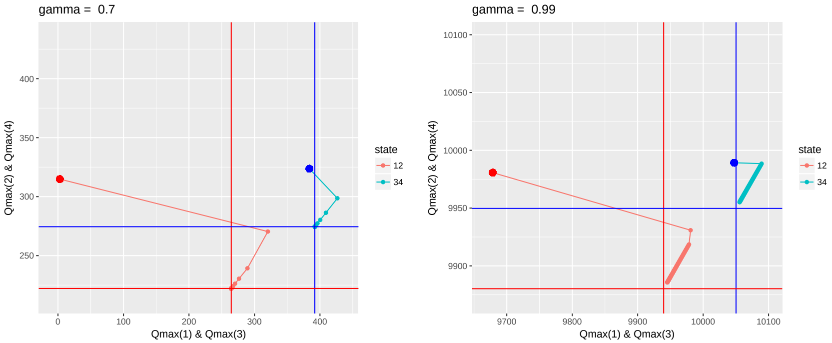
$$
\begin{aligned}
Q_*(A,a) &= R(A,a) + \gamma^1 max_{a'} Q_*(B,a') \\
Q_*(A,b) &= R(A,b) + \gamma^1 max_{a'} Q_*(C,a') \\
Q_*(A,c) &= R(A,c) + \gamma^1 max_{a'} Q_*(A,a') \\
Q_*(B,a) &= R(B,a) + \gamma^1 max_{a'} Q_*(C,a') \\
Q_*(B,b) &= R(B,b) + \gamma^1 max_{a'} Q_*(B,a') \\
&\vdots
\end{aligned}
$$

위의 방정식은 역행렬을 활용하여 풀 수 없지만, 반복적인 갱신을 활용하여 풀 수 있다!

반복적인 갱신의 예

상태state	행동action		
	a	b	c
A	$100(\rightarrow B)$	$-10(\rightarrow C)$	$-20(\rightarrow C)$
B	$0(\rightarrow A)$	$50(\rightarrow B)$	$30(\rightarrow D)$
C	$200(\rightarrow D)$	$70(\rightarrow A)$	$50(\rightarrow B)$
D	$-100(\rightarrow C)$	$0(\rightarrow A)$	$0(\rightarrow C)$

반복적인 갱신의 예: γ의 효과



동기적Synchronous 갱신

$$\boldsymbol{V}^{(1)}(\boldsymbol{A}) = 30 + 0.9V^{(0)}(B)$$
$$\boldsymbol{V}^{(1)}(\boldsymbol{B}) = -20 + 0.9V^{(0)}(C)$$
$$\boldsymbol{V}^{(1)}(\boldsymbol{C}) = 20 + 0.9V^{(0)}(A)$$

$$V^{(2)}(A) = 30 + 0.9\boldsymbol{V}^{(1)}(\boldsymbol{B})$$
$$V^{(2)}(B) = -20 + 0.9\boldsymbol{V}^{(1)}(\boldsymbol{C})$$
$$V^{(2)}(C) = 20 + 0.9\boldsymbol{V}^{(1)}(\boldsymbol{A})$$

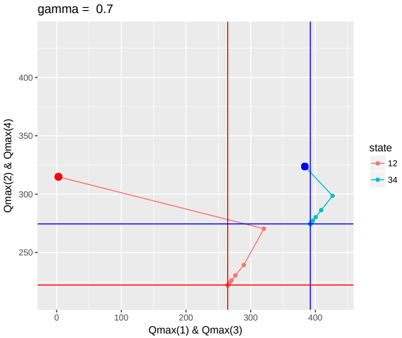
비동기적Asynchronous 갱신

$$\begin{aligned}
\boldsymbol{V^{(1)}(A)} &= 30 + 0.9V^{(0)}(B) \\
V^{(1)}(B) &= -20 + 0.9V^{(0)}(C) \\
V^{(1)}(C) &= 20 + 0.9\boldsymbol{V^{(1)}(A)}
\end{aligned}$$

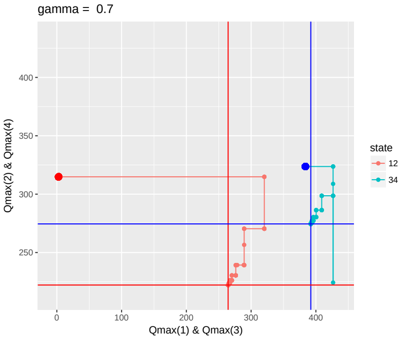
$$\begin{aligned}
V^{(2)}(A) &= 30 + 0.9V^{(1)}(B) \\
V^{(2)}(B) &= -20 + 0.9V^{(1)}(C) \\
V^{(2)}(C) &= 20 + 0.9V^{(2)}(A)
\end{aligned}$$

동기적, 비동기적 갱신의 예: Q_*

상태state	행동action		
	a	b	c
A	$100(\to B)$	$-10(\to C)$	$-20(\to C)$
B	$0(\to A)$	$50(\to B)$	$30(\to D)$
C	$200(\to D)$	$70(\to A)$	$50(\to B)$
D	$-100(\to C)$	$0(\to A)$	$0(\to C)$



Synchronous update



Asynchronous update

확률적 환경, 확률적 정책에서 벨만 방정식

상태 가치 함수

$$
\begin{aligned}
V_1(s) &= \mathbb{E}_{S' \sim p(S'|s,a'), a' \sim \pi_1(a'|s)} \left[R(s, a') + \gamma V_1(S') | S = s \right] \\
&= \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) \left[r + \gamma V_1(s') \right]
\end{aligned}
$$

상태-행동 가치 함수

$$
\begin{aligned}
Q_1(s, a) &= \mathbb{E}[R(s, a)] + \gamma \mathbb{E}_{s' \sim \mathbb{P}(s'|s,a)} \left[V(s') \right] \\
&= \mathbb{E}[R(s, a)] + \gamma \mathbb{E} \left[\sum_{a'} \pi(a'|s') \sum_{s'', r} p(s'', r|s', a) \left[r + \gamma V_1(s'') \right] \right] \\
&= \mathbb{E}[R(s, a)] + \gamma \mathbb{E} \left[\sum_{a'} \pi(a'|s') Q_1(s', a') \right]
\end{aligned}
$$

환경을 모를 때

상태 전이 확률 $\mathbb{P}(s'|s, a)$를 모를 때,
어떻게 주어진 정책에 대한 상태 가치 함수를 구하고,
더 나아가, 최상의 정책을 찾아낼 수 있을까?

주어진 상태에서 주어진 정책에 따라 행동을 여러 번 해보기

몬테카를로 (MC; Monte Carlo) 방법 : 주어진 상태 s 에서 주어진 정책 π 에 따라 행동을 계속함으로써 얻어지는 보상의 할인된 총합 (G) 을 구한다. 그리고 이를 여러 번 반복해서 평균을 구한다.

$$
\begin{aligned}
G &= R_{(0)} + \gamma R_{(1)} + \gamma^2 R_{(2)} \cdots \gamma^T R_{(T)} \\
v(s) &= \mathbb{E}_{\pi|s}\Big[G\Big] \\
\hat{v(s)} &= \sum_{i=1}^{n} g_i \Big/ n
\end{aligned}
$$

- 끝이 있는 과제에서만 사용할 수 있는 방법이다.
- 분산이 크다.

환경을 모를 때 주어진 정책에 대한 상태 가치 함수 학습하기

시간차Temporal Differenc 방법을 사용하여 상태 가치 함수 갱신update하기

- DP fullbackup

$$\begin{aligned} V_1(s) &= \mathbb{E}_{S' \sim p(S'|s,a'), a' \sim \pi_1(a'|s)} \left[R(s, a') + \gamma V_1(S') | S = s \right] \\ &= \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) \left[r + \gamma V_1(s') \right] \end{aligned}$$

- TD update(step-size:α)

$$V_1(s) = V_1(s) + \alpha \left[r + \gamma V_1(s') - V_1(s) \right]$$

모평균의 추정

주어진 표본 $y_1, y_2, y_3, \cdots$ 에 대해 최소 분산 선형 추정량BLUE; Best Linear Unbiased Estimator은 다음의 표본 평균이다.[2]

$$\hat{\mu}_n = \frac{y_1 + y_2 + \cdots + y_n}{n}$$

만약 $y_1, y_2, \cdots$ 가 순차적으로 관찰된다면, 다음의 반복적인 갱신을 활용할 수 있다.

$$\begin{aligned}
\hat{\mu}_1 &= y_1 \\
\hat{\mu}_2 &= \frac{y_1 + y_2}{2} = \frac{\hat{\mu}_1 + y_2}{2} \\
\hat{\mu}_3 &= \frac{y_1 + y_2 + y_3}{3} = \frac{2\hat{\mu}_2 + y_3}{3} = \frac{2}{3}\hat{\mu}_2 + \frac{1}{3}y_3
\end{aligned}$$

[2] Gauss-Markov 정리

순차적인 모평균의 추정

$$
\begin{aligned}
\hat{\mu_1} &= y_1 \\
\hat{\mu_2} &= \frac{y_1 + y_2}{2} = \frac{\hat{\mu_1} + y_2}{2} \\
\hat{\mu_3} &= \frac{y_1 + y_2 + y_3}{3} = \frac{2\hat{\mu_2} + y_3}{3} = \frac{2}{3}\hat{\mu_2} + \frac{1}{3}y_3 \\
&= \left(1 - \frac{\mathbf{1}}{\mathbf{3}}\right)\hat{\mu_2} + \frac{\mathbf{1}}{\mathbf{3}}y_3 \\
&= \hat{\mu_2} + \frac{\mathbf{1}}{\mathbf{3}}(y_3 - \hat{\mu_2}) \\
&\vdots \\
\hat{\mu_n} &= \hat{\mu_{n-1}} + \frac{\mathbf{1}}{\boldsymbol{n}}(y_n - \hat{\mu_{n-1}})
\end{aligned}
$$

순차적인 모평균의 추정

$\hat{\mu_n} = \hat{\mu_{n-1}} + \frac{\mathbf{1}}{\boldsymbol{n}}(y_n - \hat{\mu_{n-1}})$ 의 다음과 같이 α_n(step-size) 를 활용하여 표현할 수 있다.

$$\boldsymbol{\alpha_n} = \frac{1}{n}$$

$$\hat{\mu_n} = \hat{\mu_{n-1}} + \boldsymbol{\alpha_n}(y_n - \hat{\mu_{n-1}})$$

순차적인 모평균의 추정: 고정 학습률

$$\hat{\mu_n} = \hat{\mu_{n-1}} + \boldsymbol{\alpha}(y_n - \hat{\mu_{n-1}})$$

고정 학습률을 사용하는 경우, 이를 가중치 평균weighted average으로 다시 표현하면 다음과 같다.

$$\begin{aligned}
\hat{\mu_1} = \hat{\mu_0} + \alpha(y_1 - \hat{\mu_0}) &= (1-\alpha)\hat{\mu_0} + \alpha y_1 \\
\hat{\mu_2} = \hat{\mu_1} + \alpha(y_2 - \hat{\mu_1}) &= (1-\alpha)\boldsymbol{\hat{\mu_1}} + \alpha y_2 \\
&= (1-\alpha)\Big[\boldsymbol{(1-\alpha)\hat{\mu_0} + \alpha y_1}\Big] + \alpha y_2 \\
&= (1-\alpha)^2\hat{\mu_0} + (1-\alpha)\alpha y_1 + \alpha y_2 \\
&\vdots \\
\hat{\mu_n} \quad &= \boldsymbol{(1-\alpha)^n}\hat{\mu_0} + \boldsymbol{(1-\alpha)^{n-1}}\alpha y_1 + \cdots + \boldsymbol{\alpha y_n}
\end{aligned}$$

Stationary/Non-stationary 조건: 학습률의 조정

- 학습률 $\alpha_n = \frac{1}{n}$ 은 모평균이 고정되어 있을 때 최소분산 가중치 평균이지만, 모평균이 변화할 경우에는 이를 반영하지 못한다.

$$\mathbb{V}\mathrm{ar}[\hat{\mu}_n] = \left(\frac{1}{n}\right)^2 \mathbb{V}\mathrm{ar}[y_1] + \cdots + \left(\frac{1}{n}\right)^2 \mathbb{V}\mathrm{ar}[y_n]$$

- 고정 학습률 $\alpha_n = \alpha$ 는 그 크기가 클수록 모평균의 변화를 반영할 수 있지만, 모평균이 고정되어 있을 경우에는 모평균으로 수렴하지 못하는 단점이 있다.

$$\mathbb{V}\mathrm{ar}[\hat{\mu}_n] = \left((1-\alpha)^{n-1}\alpha\right)^2 \mathbb{V}\mathrm{ar}[y_1] + \cdots + \alpha^2 \mathbb{V}\mathrm{ar}[y_n]$$

- 두 학습률의 장점을 결합하여 $\alpha_n = \max(\frac{1}{n}, \alpha)$ 으로 설정할 수도 있다.

순차적인 모평균의 추정: 변동 학습률

$$\hat{\mu_n} = \hat{\mu_{n-1}} + \boldsymbol{\alpha_n}(y_n - \hat{\mu_{n-1}})$$

고정 학습률을 사용하는 경우, 이를 가중치 평균weighted average으로 다시 표현하면 다음과 같다.

$$\begin{aligned}
\hat{\mu_1} &= \hat{\mu_0} + \alpha_1(y_1 - \hat{\mu_0}) = (1-\alpha_1)\hat{\mu_0} + \alpha_1 y_1 \\
\hat{\mu_2} &= \hat{\mu_1} + \alpha_2(y_2 - \hat{\mu_1}) = (1-\alpha_2)\boldsymbol{\hat{\mu}_1} + \alpha_2 y_2 \\
&= (1-\alpha_2)\left[\boldsymbol{(1-\alpha_1)\hat{\mu_0} + \alpha_1 y_1}\right] + \alpha_2 y_2 \\
&= (1-\alpha_2)(1-\alpha_1)\hat{\mu_0} + (1-\alpha_2)\alpha_1 y_1 + \alpha_2 y_2 \\
&\vdots \\
\hat{\mu_n} &= \left[\prod_{i=1}^{n}(1-\alpha_i)\right]\hat{\mu_0} + \left[\prod_{i=2}^{n}(1-\alpha_i)\right]\alpha_1 y_1 + \cdots + \alpha_n y
\end{aligned}$$

환경을 모를 때 주어진 정책에 대한 상태-행동 가치 함수 학습하기

시간차Temporal Differenc 방법을 사용하여 상태-행동 가치 함수 갱신update하기

- DP fullbackup

$$Q_1(s,a) = \mathbb{E}[R(s,a)] + \gamma \mathbb{E}\left[\sum_{a'} \pi(a'|S') Q_1(S',a')\right]$$

- TD update(step-size:α)
  - sarsa

$$Q_1(s,a) = Q_1(s,a) + \alpha\left[r + \gamma Q_1(s',a') - Q_1(s,a)\right]$$

TD update for Q-function

행위자가 π_1 을 따라 환경과 상호작용하여 (s, a, r, s') 을 얻을 때:

- sarsa

$$Q_1(s,a) = Q_1(s,a) + \alpha\Big[r + \gamma Q_1(s', a') - Q_1(s,a)\Big]$$

- expected sarsa(on-policy)

$$Q_1(s,a) = Q_1(s,a) + \alpha\Big[r + \gamma \mathbb{E}_{\pi_1} Q_1(s', A') - Q_1(s,a)\Big]$$

- expected sarsa(off-policy)

$$Q_2(s,a) = Q_2(s,a) + \alpha\Big[r + \gamma \mathbb{E}_{\pi_2}[Q_2(s', A')] - Q_2(s,a)\Big]$$

- Q-learning(off-policy)

$$Q_*(s,a) = Q_*(s,a) + \alpha\Big[r + \gamma \max_{a'} Q_*(s', a') - Q_*(s,a)\Big]$$

sample mean

확률변수 X의 평균을 구하고자 한다.

$$\mathbb{E}[X] = \sum_{i=1}^{N_x} x_i p(x_i) \quad (N_x : \text{서로다른}x\text{값의수})$$

만약 $p(x_i)$를 모른다면 표본 평균을 쓸 수 있다:

$$\widehat{\mathbb{E}[X]} = \sum_{i=1}^{N} x_i \Big/ N \quad (N : \text{표본의크기})$$

sample mean 예

x_i	1	2	3	4	5
$P(x_i)$	0.4	0.2	0.2	0.1	0.1

- 평균

$$\mathbb{E}[X] = 1 \cdot 0.4 + 2 \cdot 0.2 + 3 \cdot 0.2 + 4 \cdot 0.1 + 5 \cdot 0.1 = 2.3$$

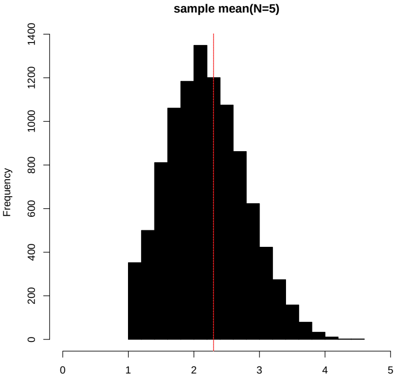
- 표본 평균으로 모평균 추정: Sampling with replacement($N = 5$)

Prob.	x_1	x_2	x_3	x_4	x_5	$\widehat{\mathbb{E}[X]}$
$(0.4)^5 = 0.01024$	1	1	1	1	1	1
$(0.4)^4(0.2) = 0.00512$	1	1	1	1	2	1.2
$(0.4)^4(0.2) = 0.00512$	1	1	1	1	3	1.4
$(0.4)^4(0.1) = 0.00256$	1	1	1	1	4	1.6
$(0.4)^4(0.1) = 0.00256$	1	1	1	1	5	1.8
$(0.4)^3(0.2)(0.4) = 0.00512$	1	1	1	2	1	1.2
$(0.4)^3(0.2)^2 = 0.00256$	1	1	1	2	2	1.4
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

sample mean 예

X	1	2	3	4	5
$P(X)$	0.4	0.2	0.2	0.1	0.1

- 표본크기 5에서 표본 평균의 분포(복원 추출)



Importance sampling

x_i, y_i	1	2	3	4	5
$P_X(x_i)$	0.4	0.2	0.2	0.1	0.1
$P_Y(y_i)$	0.1	0.1	0.2	0.2	0.4

$\mathbb{E}[Y]$를 구하고 싶은데, $P(Y)$를 모른다.
X를 sampling할 수 있으며, $P(y)/P(x)$를 알 수 있다면,

$$\widehat{\mathbb{E}[Y]} = \sum_{i=1}^{N} \frac{P_Y(x_i)}{P_X(x_i)} x_i \bigg/ N \quad (N: \text{표본의크기})$$

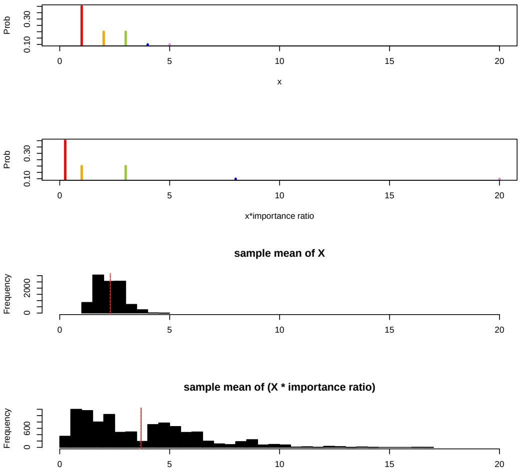
Importance sampling

x_i, y_i	1	2	3	4	5	$\mathbb{E}$	$\mathbb{V}\mathrm{ar}$
$P_X(x_i)$	0.4	0.2	0.2	0.1	0.1	2.3	1.81
$P_Y(y_i)$	0.1	0.1	0.2	0.2	0.4	3.7	1.81
ISR	1/4	1/2	1	2	4	1	1.275
$x_i \times \mathrm{ISR}$	0.25	1	3	8	20	**3.7**	**36.695**

$\mathbb{E}[Y]$를 구하고 싶은데, $P(Y)$를 모른다.
X를 sampling할 수 있으며, $P(y)/P(x)$를 알 수 있다면,

$$\widehat{\mathbb{E}[Y]} = \sum_{i=1}^{N} \frac{P_Y(x_i)}{P_X(x_i)} x_i \Big/ N \quad (N : \text{표본의크기})$$

Importance sampling



Weighted(Self-Normalized) Importance sampling

x_i, y_i	1	2	3	4	5	$\mathbb{E}$	$\mathbb{V}\mathrm{ar}$
$P_X(x_i)$	0.4	0.2	0.2	0.1	0.1	2.3	1.81
$P_Y(y_i)$	0.1	0.1	0.2	0.2	0.4	3.7	1.81
ISR	1/4	1/2	1	2	4	**1**	1.275
$x_i \times$ ISR	0.25	1	3	8	20	3.7	**36.695**

Importance sampling estimator

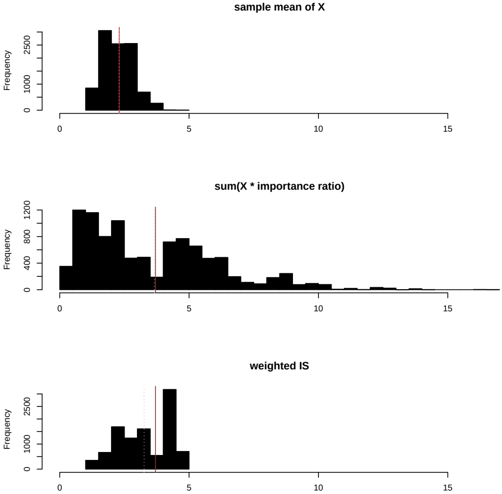
$$\widehat{\mathbb{E}[Y]} = \sum_{i=1}^{N} \frac{P_Y(x_i)}{P_X(x_i)} x_i \Big/ N \quad (N : \text{표본의크기})$$

Weighted importance sampling estimator

$$\widehat{\mathbb{E}[Y]} = \sum_{i=1}^{N} w_i x_i \Big/ \boldsymbol{\sum_{i=1}^{N} w_i} \quad \left(w_i = \frac{P_Y(x_i)}{P_X(x_i)}\right)$$

$$= \sum_{i=1}^{N} \frac{w_i}{\sum_{i=1}^{N} w_i} x_i$$

Weighted(Self-Normalized) Importance sampling



Off-policy MC method

- Importance sampling
- Weighted Importance sampling
- Per-decision importance sampling
- Weighted per-decision importance sampling

Off-policy TD method

π_1(behavior policy)에 따라 얻어진 (s, a, r, s')에 대해서, π_2(target policy)의 Q-함수를 구하고자 한다면:

- DP fullbackup

$$Q_2(s,a) = \mathbb{E}[R(s,a)] + \gamma \mathbb{E}\Big[\sum_{a'} \pi_2(a'|S')Q_2(S',a')\Big]$$

- On-policy TD update(sarsa, step-size:α)

$$Q_1(s,a) = Q_1(s,a) + \alpha\Big[r + \gamma Q_1(s',a') - Q_1(s,a)\Big]$$

- Off-policy TD update(sarsa, step-size:α)

$$Q_2(s,a) = Q_2(s,a) + \alpha \frac{\pi_2(a'|s')}{\pi_1(a'|s')}\Big[r + \gamma Q_2(s',a') - Q_2(s,a)\Big]$$

- expected sarsa(off-policy)

$$Q_2(s,a) = Q_2(s,a) + \alpha\Big[r + \gamma \mathbb{E}_{\pi_2}[Q_2(s',A')] - Q_2(s,a)\Big]$$

모평균의 추정 : Importance sampling

주어진 표본 $y_1, y_2, y_3, \cdots$ 에 대해 Importance sampling estimator는 다음과 같다.

$$\hat{\mu}_X = \frac{w_1 y_1 + w_2 y_2 + \cdots + w_n y_n}{n}, \quad w_i = f_X(y_i)/f_Y(y_i)$$

만약 $y_1, y_2, \cdots$ 가 순차적으로 관찰된다면, 다음의 반복적인 갱신을 활용할 수 있다.

$$\begin{aligned} \hat{\mu}_1 &= w_1 y_1 \\ \hat{\mu}_2 &= \frac{w_1 y_1 + w_2 y_2}{2} = \frac{\hat{\mu}_1 + w_2 y_2}{2} \\ \hat{\mu}_3 &= \frac{w_1 y_1 + w_2 y_2 + w_3 y_3}{3} = \frac{2\hat{\mu}_2 + w_3 y_3}{3} = \frac{2}{3}\hat{\mu}_2 + \frac{1}{3} w_3 y_3 \end{aligned}$$

순차적인 모평균의 추정 : Importance sampling

$$
\begin{aligned}
\hat{\mu}_1 &= w_1 y_1 \\
\hat{\mu}_2 &= \frac{w_1 y_1 + w_2 y_2}{2} = \frac{\hat{\mu}_1 + w_2 y_2}{2} \\
\hat{\mu}_3 &= \frac{w_1 y_1 + w_2 y_2 + w_3 y_3}{3} = \frac{2\hat{\mu}_2 + w_3 y_3}{3} = \frac{2}{3}\hat{\mu}_2 + \frac{1}{3} w_3 y_3 \\
&= \left(1 - \frac{\mathbf{1}}{\mathbf{3}}\right) \hat{\mu}_2 + \frac{\mathbf{1}}{\mathbf{3}} w_3 y_3 \\
&= \hat{\mu}_2 + \frac{\mathbf{1}}{\mathbf{3}} (w_3 y_3 - \hat{\mu}_2) \\
&\vdots \\
\hat{\mu}_n &= \hat{\mu_{n-1}} + \frac{\mathbf{1}}{\boldsymbol{n}} (w_n y_n - \hat{\mu_{n-1}})
\end{aligned}
$$

순차적인 모평균의 추정 : Importance sampling

$\hat{\mu_n} = \hat{\mu_{n-1}} + \frac{\mathbf{1}}{\boldsymbol{n}}(w_n y_n - \hat{\mu_{n-1}})$ 의 다음과 같이 α_n(step-size) 를 활용하여 표현할 수 있다.

$$\boldsymbol{\alpha_n} = \frac{1}{n}$$
$$\hat{\mu_n} = \hat{\mu_{n-1}} + \boldsymbol{\alpha_n}(w_n y_n - \hat{\mu_{n-1}})$$

순차적인 모평균의 추정 (Importance sampling): 고정 학습률

$$\hat{\mu_n} = \hat{\mu_{n-1}} + \boldsymbol{\alpha}(w_n y_n - \hat{\mu_{n-1}})$$

고정 학습률을 사용하는 경우, 이를 가중치 평균weighted average으로 다시 표현하면 다음과 같다.

$$\begin{aligned}
\hat{\mu}_1 = \hat{\mu}_0 + \alpha(w_1 y_1 - \hat{\mu}_0) &= (1-\alpha)\hat{\mu}_0 + \alpha w_1 y_1 \\
\hat{\mu}_2 = \hat{\mu}_1 + \alpha(w_2 y_2 - \hat{\mu}_1) &= (1-\alpha)\boldsymbol{\hat{\mu}_1} + \alpha w_2 y_2 \\
&= (1-\alpha)\left[\boldsymbol{(1-\alpha)\hat{\mu}_0 + \alpha w_1 y_1}\right] + \alpha w_2 y_2 \\
&= (1-\alpha)^2\hat{\mu}_0 + (1-\alpha)\alpha w_1 y_1 + \alpha w_2 y_2 \\
&\vdots \\
\hat{\mu}_n &= \boldsymbol{(1-\alpha)^n}\hat{\mu}_0 + \boldsymbol{(1-\alpha)^{n-1}}\alpha w_1 y_1 + \cdots + \boldsymbol{\alpha} w_n y_n
\end{aligned}$$

모평균의 추정 : Weighted Importance sampling

주어진 표본 $y_1, y_2, y_3, \cdots$ 에 대해 Importance sampling estimator는 다음과 같다.

$$\hat{\mu}_X = \frac{w_1 y_1 + w_2 y_2 + \cdots + w_n y_n}{\sum_{i=1}^{n} w_i}, \quad w_i = f_X(y_i)/f_Y(y_i)$$

만약 $y_1, y_2, \cdots$ 가 순차적으로 관찰된다면,

$$\hat{\mu}_1 = \frac{w_1 y_1}{w_1} = y_1$$

$$\hat{\mu}_2 = \frac{w_1 y_1 + w_2 y_2}{w_1 + w_2} = \frac{w_1 \hat{\mu}_1 + w_2 y_2}{w_1 + w_2} = \frac{(w_1 + w_2 - w_2)\hat{\mu}_1 + w_2 y_2}{w_1 + w_2}$$

$$\hat{\mu}_3 = \frac{w_1 y_1 + w_2 y_2 + w_3 y_3}{w_1 + w_2 + w_3} = \frac{(w_1 + w_2)\hat{\mu}_2 + w_3 y_3}{w_1 + w_2 + w_3}$$

$$= \hat{\mu}_2 - \frac{w_3}{w_1 + w_2 + w_3}\hat{\mu}_2 + \frac{w_3 y_3}{w_1 + w_2 + w_3} = \hat{\mu}_2 - \frac{w_3 y_3 - w_3 \mu_2}{w_1 + w_2 + w_3}$$

정책 기반 강화 학습

- **정책Policy** : 주어진 상태 s에서 행동 a를 택할 확률

$$\pi(s, a) = \mathbb{P}(A = a | S = s)$$

- 어떤 모수 θ의 함수로 정책을 나타낼 수 있다. $\pi_\theta(s, a)$

$$\text{예)}\ \pi_\theta(s, a) = \theta^a (1 - \theta)^{1-a} \ (a = 0, 1)$$

정책 기반 강화 학습 : 최적의 정책 구하기

- 정책을 판단하는 두 가지 기준
  - 할인된 보상의 총합 : $\mathbb{E}_{\pi,\mu}\left[R + \gamma R' + \gamma^2 R'' + \cdots\right]$
  - 평균 보상 : $\mathbb{E}_{\pi,\mu}\left[\lim_{n\to\infty} \frac{R+R'+R''+\cdots+R^{(n)}}{n}\right]$
- 정책이 θ의 함수라면,
  최적의 정책은 $\nabla_\theta \mathbb{E} = 0$을 만족할 것이다.

정책 기반 강화 학습 : REINFORCE

$J(\theta) = \mathbb{E}[R + \gamma R + \gamma R^2 + \cdots | \pi(\theta)]$ 라고 하면,

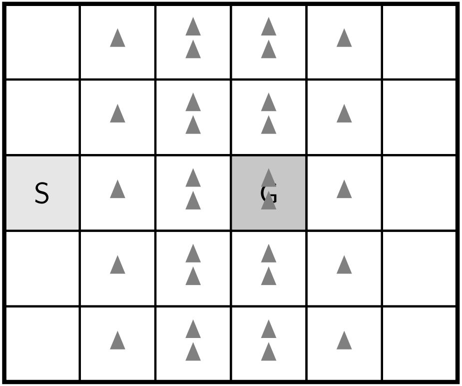
$$
\begin{aligned}
\nabla_\theta J(\theta) &= \sum_s d_{\pi_\theta}(s) \sum_a \nabla_\theta \pi(a|s) Q_\pi(s, a, \theta) \\
&= \sum_s d_{\pi_\theta}(s) \sum_a \frac{\pi(a|s)}{\pi(a|s)} \nabla_\theta \pi(a|s) Q_\pi(s, a, \theta) \\
&= \sum_s d_{\pi_\theta}(s) \sum_a \pi(a|s) \nabla_\theta \log \pi(a|s) Q_\pi(s, a, \theta) \\
&= \mathbb{E}_\theta \left[\nabla_\theta \log \pi(a|s) Q_\pi(s, a, \theta) \right]
\end{aligned}
$$

정책 기반 강화 학습 : REINFORCE

$$\theta_{t+1} \leftarrow \theta_t + \alpha \left[\nabla_\theta \log \pi_\theta (a|s) G_t \right]$$

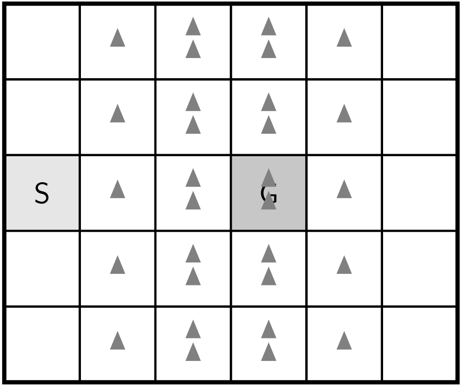
실습: 바람부는 격자 세계

결정론적 환경 실습: 바람부는 격자 세계 Windy01_DP.py



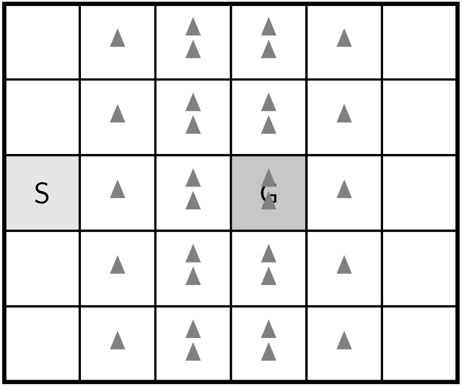
- 행위자는 S(tart)에서 출발한다.
- 모든 가능한 행동의 집합 $\mathcal{A} = \{\uparrow, \leftarrow, \downarrow, \rightarrow\}$
- 바람에 의해 저절로 1 또는 2만큼 위로 이동한다.

확률적 환경 실습: 바람부는 격자 세계 Windy02_DP.py



- 바람의 세기는 확률적으로 결정되면, $0, 1, 2$ 또는 $1, 2, 3$의 확률이 모두 $1/$ 이다.

TD 학습 실습: 바람부는 격자 세계 `Windy02_TD.py`



- 행동에 의한 상태 변화와 보상을 전혀 모르는 경우
- TD 학습법 : $V(s) \leftarrow V(s) + \alpha \left[r + \gamma V(s') - V(s) \right]$

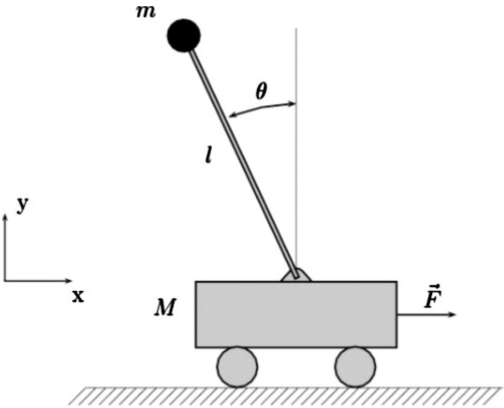
실습: CNN으로 MNIST 숫자 인식하기

실습: MNIST 숫자 인식하기 mnist_CNN.py

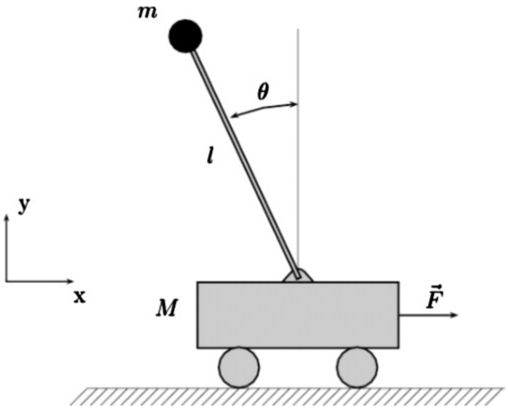
```
model = Sequential()

model.add(Conv2D(32, (3, 3), activation='relu',
  input_shape=(28,28,1)))
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))
model.add(Flatten())

model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))
```

실습: 카트폴

카트폴 수직으로 세우기



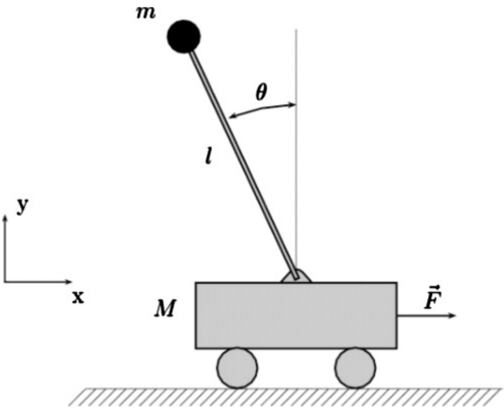
키보드 체험: 카트폴 수직으로 세우기
Cartpole_keyboard.py



- 키보드 <1>을 누르면 카트를 오른쪽으로 밀게 된다.
- 아무것도 누르지 않으면 카트를 왼쪽으로 민다.

REINFORCE: 카트폴 수직으로 세우기

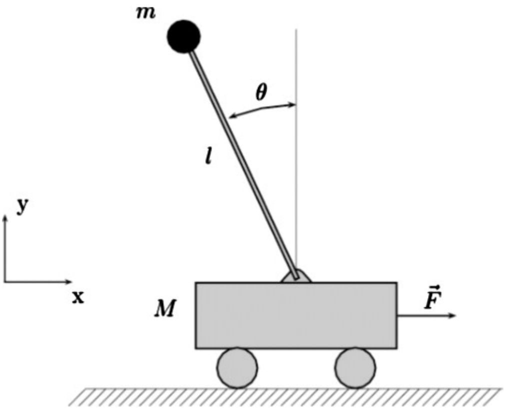
Cartpole_REINFORCE.py



- 한 에피소드가 끝난 후 보상의 총합을 바탕으로 정책망의 파라미터를 갱신한다.

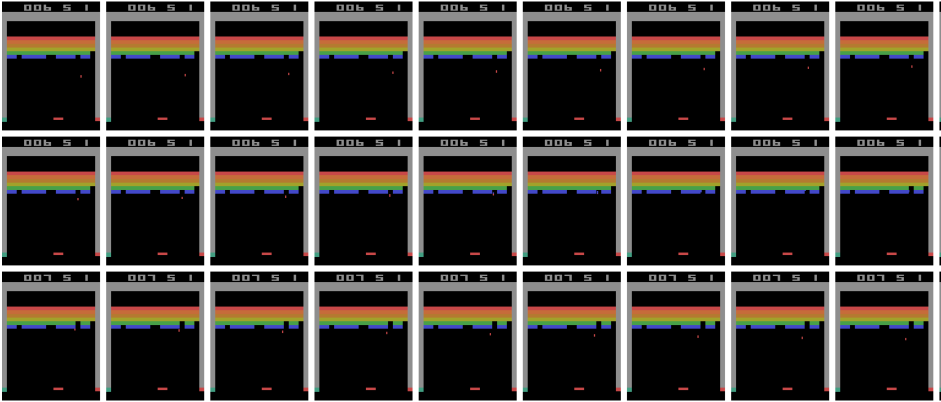
Actor-Critic: 카트폴 수직으로 세우기

Cartpole_ActorCritic.py



- 매 시간 크리티의 평가를 바탕으로 정책망(Actor)의 파라미터를 갱신한다.

마무리 : ATARI 시연



감사합니다!