Policy Gradient algorithm

24.02.22

정상혁

Policy Gradient algorithm

DQN : Q-Network (CNN)

- Learns the optimal action value function

$$Q(s, a; \theta) \approx Q^*(s, a)$$

- state space : continuous
  action space : **discrete** (not large)

- Policy

$$\pi(a \mid s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{m} & , if\ a = \underset{a'}{argmax}\ Q(s, a'; \theta) \\ \frac{\epsilon}{m} & , otherwise\ (m-1\ actions) \end{cases}$$

- Loss function

$$L(\theta) = [r_{t+1} + \gamma \max_a Q(s_{t+1}, a; \theta) - Q(s_t, a_t; \theta)]^2$$

- Weight update

$$\theta := \theta - \alpha \nabla_\theta L(\theta)$$

Policy gradient algorithm

- Directly learns the optimal policy

$$\pi_\theta(a \mid s)$$

- state space : continuous
  action space : **continuous**

Trajectory : $\tau = s_0, a_0, r_1, s_1, a_1, r_2, \cdots, s_{T-1}, a_{T-1}, r_T, s_T$

Total reward : $r(\tau)$

- Objective function

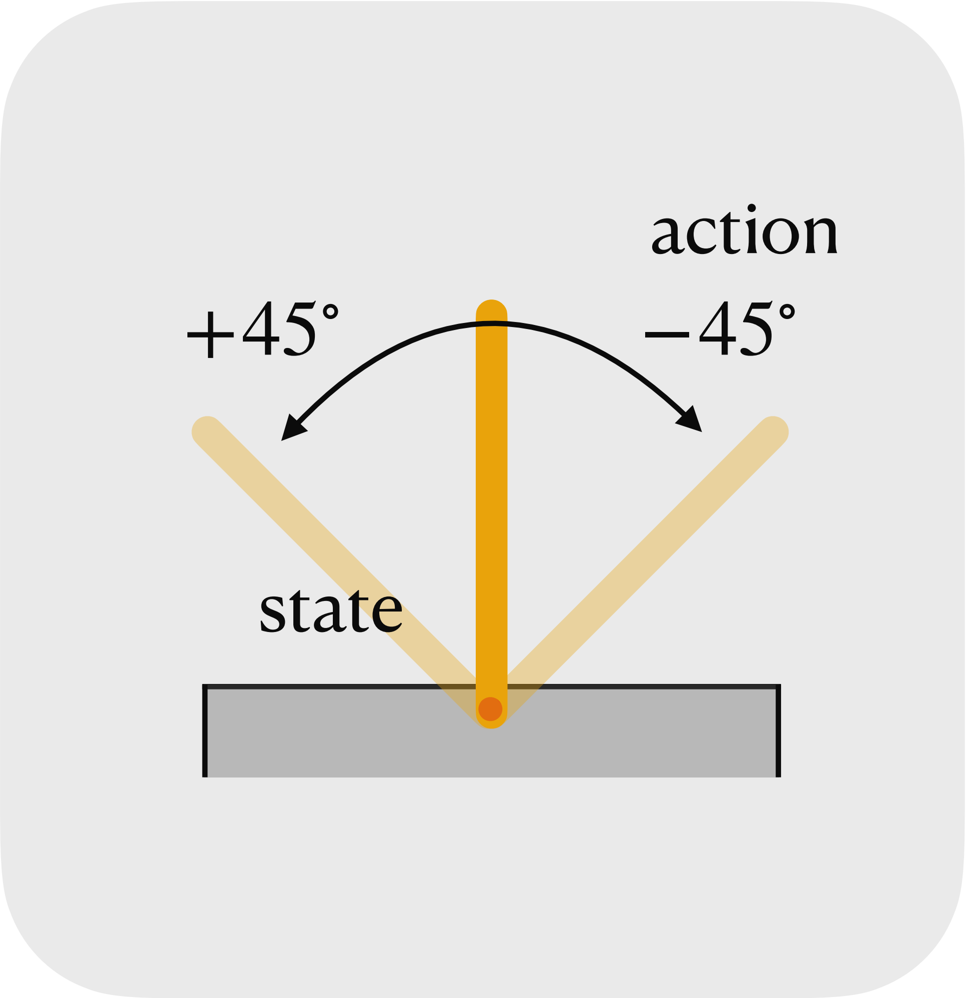
$$J(\theta) = E_{\pi_\theta}[r(\tau)] = \int p(\tau; \theta) r(\tau)\, d\tau$$

(where $p(\tau; \theta) = \pi_\theta(\tau)$ is the pdf of τ)

- Weight update

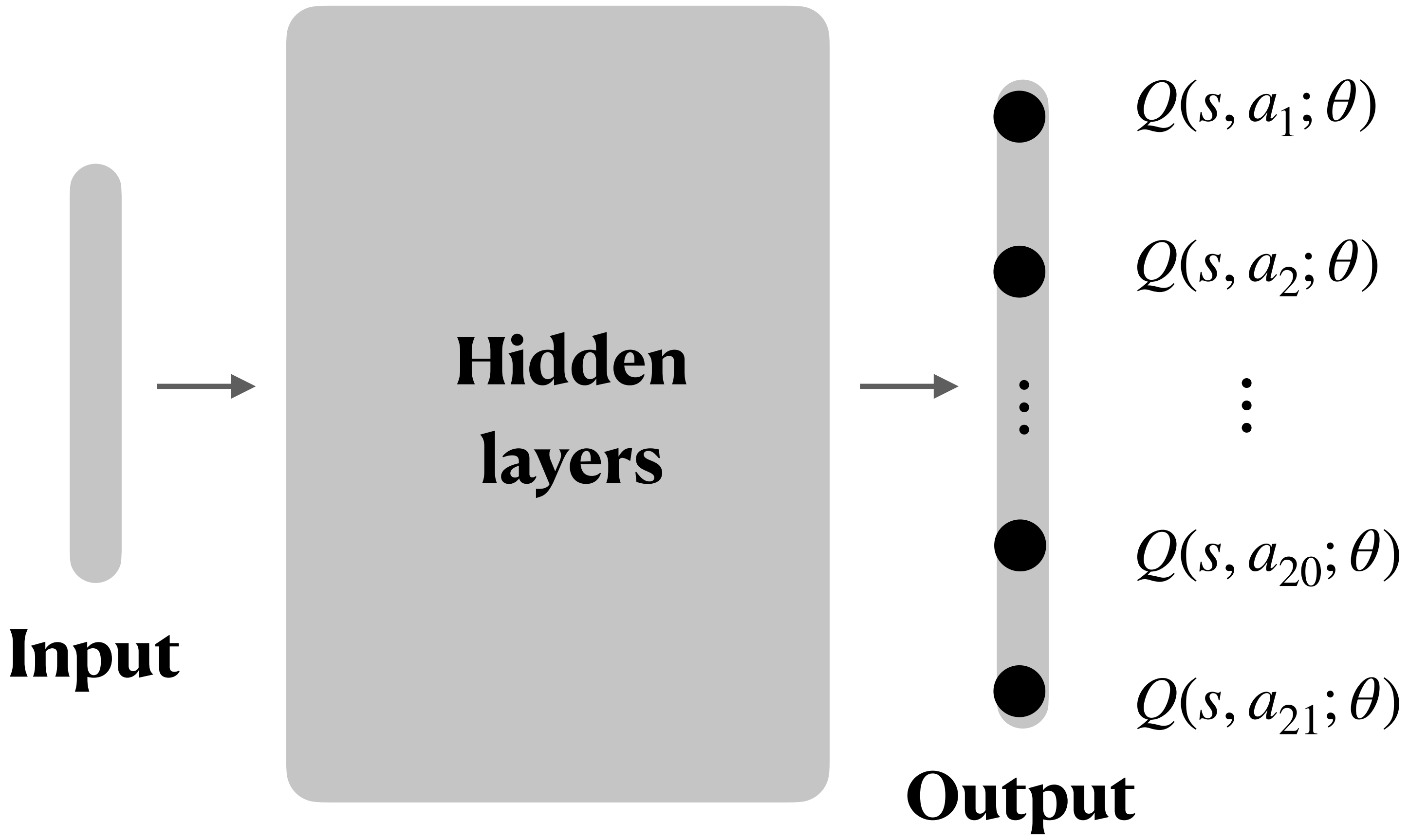
$$\theta := \theta + \alpha \nabla_\theta J(\theta)$$

Continuous action space



- **DQN** : $Q(s, a; \theta)$

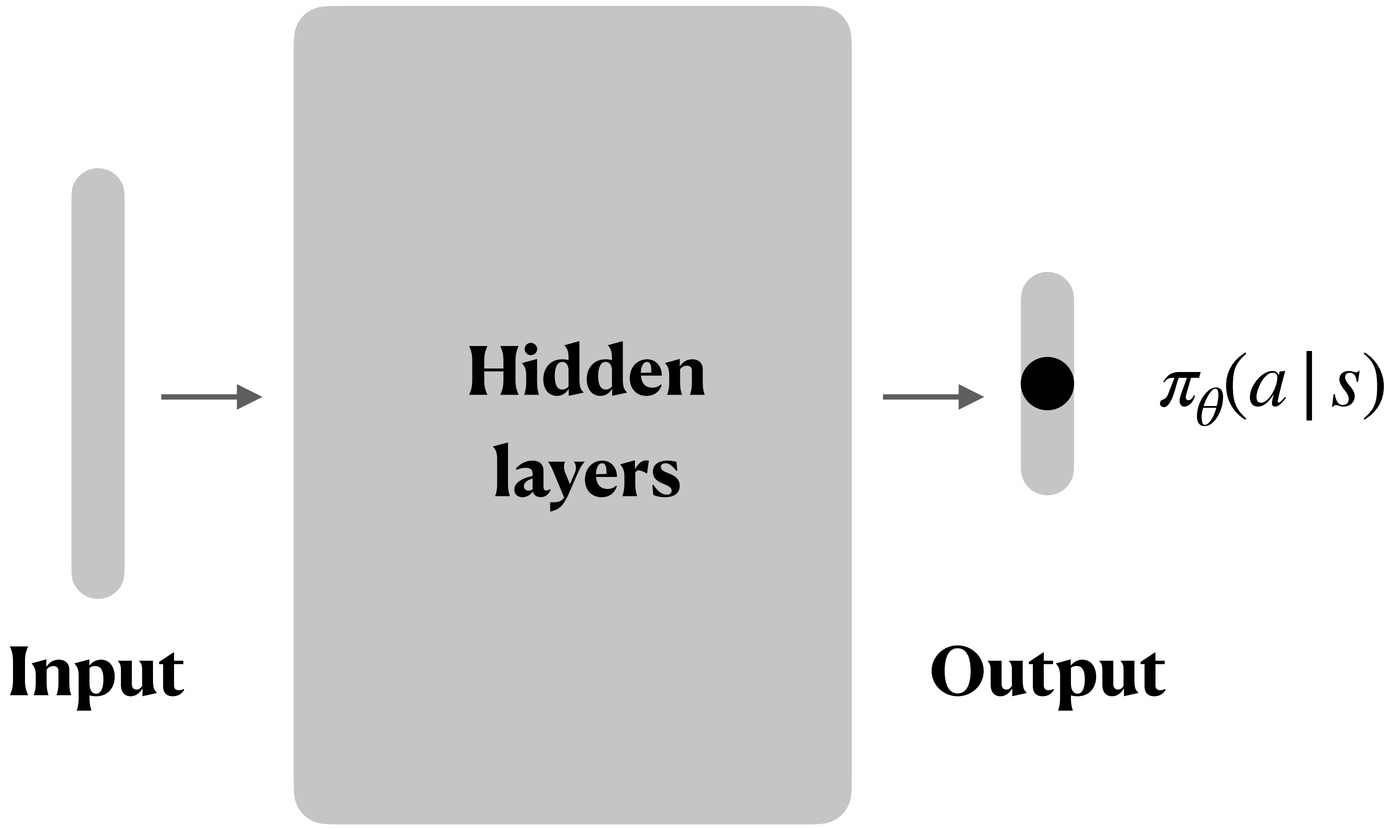
state space : continuous
action space : **discrete** (not large)



(where $a_1 = -10, a_2 = -9, \cdots, a_{20} = +9, a_{21} = +10$)

- **Policy gradient algorithm** : $\pi_\theta(a \mid s)$

state space : continuous
action space : **continuous**



$\pi_\theta(a \mid s) \in [-10, 10]$

Policy Gradient Theorem

[Policy Gradient Theorem]

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\pi_\theta}[r(\tau)] = \mathbb{E}_{\pi_\theta}\left[r(\tau) \textstyle\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]$$

The derivative of the expected total reward is the expectation of the product of total rewards and summed gradients of log of the policy π_θ.

Advantages

- Do not need to know $p(\tau;\theta) = \pi_\theta(\tau)$ nor $p(s_{t+1} \mid s_t, a_t)$
- Expectation can be approximated by sampling

$$\begin{aligned}
\nabla_\theta E_{\pi_\theta}[r(\tau)] &= \nabla_\theta \int \pi_\theta(\tau) r(\tau)\, d\tau \\
&= \int \nabla_\theta \pi_\theta(\tau)\, r(\tau)\, d\tau \\
&= \int \pi_\theta(\tau) \frac{\nabla_\theta \pi_\theta(\tau)}{\pi_\theta(\tau)}\, r(\tau)\, d\tau \\
&= \int \pi_\theta(\tau)\, r(\tau)\, \nabla_\theta log \pi(\tau)\, d\tau \\
&= E_{\pi_\theta}[r(\tau)\, \nabla_\theta log \pi_\theta(\tau)] \\
&= E_{\pi_\theta}[r(\tau)\, \nabla_\theta \sum_{t=0}^{T-1} log\, \pi_\theta(a_t \mid s_t)]
\end{aligned}$$

- Trajectory : $\tau = s_0, a_0, r_1, s_1, a_1, r_2, \cdots, s_{T-1}, a_{T-1}, r_T, s_T$

$$\begin{aligned}
\pi(\tau) &= p(s_0)\boxed{P(a_0 \mid s_0)P(s_1 \mid s_0, a_0)}\boxed{P(a_1 \mid s_1)P(s_2 \mid s_1, a_1)} \cdots \boxed{P(a_{T-1} \mid s_{T-1})P(s_T \mid s_{T-1}, a_{T-1})} \\
&= p(s_0)\ \Pi_{t=0}^{T-1}\, \pi(a_t \mid s_t)\, p(s_{t+1} \mid s_t, a_t)
\end{aligned}$$

($\because$ Markov property)

$P(s_2 \mid s_0, a_0, s_1, a_1) = P(s_2 \mid s_1, a_1)$

$$\begin{aligned}
\because\ \nabla_\theta log\, \pi_\theta(\tau) &= \nabla_\theta log(\ p(s_0)\ \Pi_{t=0}^{T-1}\, \pi_\theta(a_t \mid s_t)\, p(s_{t+1} \mid s_t, a_t)\ ) \\
&= \underbrace{\cancel{\nabla_\theta log\, p(s_0)}}_{=0} + \nabla_\theta \sum_{t=0}^{T-1} log\, \pi_\theta(a_t \mid s_t) + \underbrace{\cancel{\nabla_\theta \sum_{t=0}^{T-1} log\, p(s_{t+1} \mid s_t, a_t)}}_{=0}
\end{aligned}$$

Policy Gradient Theorem

$$\nabla_\theta J(\theta) = \nabla_\theta E_{\pi_\theta}[r(\tau)] = E_{\pi_\theta}\left[\, r(\tau) \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \right]$$

Trajectory : $\tau = s_0, a_0, r_1, s_1, a_1, r_2, \cdots, s_{T-1}, a_{T-1}, r_T, s_T$

- **Expectation**

  Can be approximated by sampling a large number of trajectories
  Unbiased approximation
  MCMC(Markov Chain Monte Carlo)

- **Total reward :** $r(\tau)$ $\rightarrow$ **Discounted return :** G_t

  $r(\tau)$ adds high variance

  Rewards before the time step t don't contribute anything

$$E_{\pi_\theta}\left[\, \sum_{t=0}^{T-1} r(\tau)\ \nabla_\theta \log \pi_\theta(a_t \mid s_t) \right]$$

$$\Downarrow$$

$$E_{\pi_\theta}\left[\, \sum_{t=0}^{T-1} \underline{G_t}\ \underline{\nabla_\theta \log \pi_\theta(a_t \mid s_t)} \right]$$

s_t 에서 선택한 a_t 가 얼마나 좋은지

$J(\theta)$ 를 maximize 하는 학습 방향