Policy Gradient

이다경

INDEX

1. Policy-based Reinforcement Learning

2. Policy Objective Function

    - Finite different policy gradient
    - Monte-Carlo policy gradient
    - Actor-Critic policy gradient

Value-based vs Policy-based

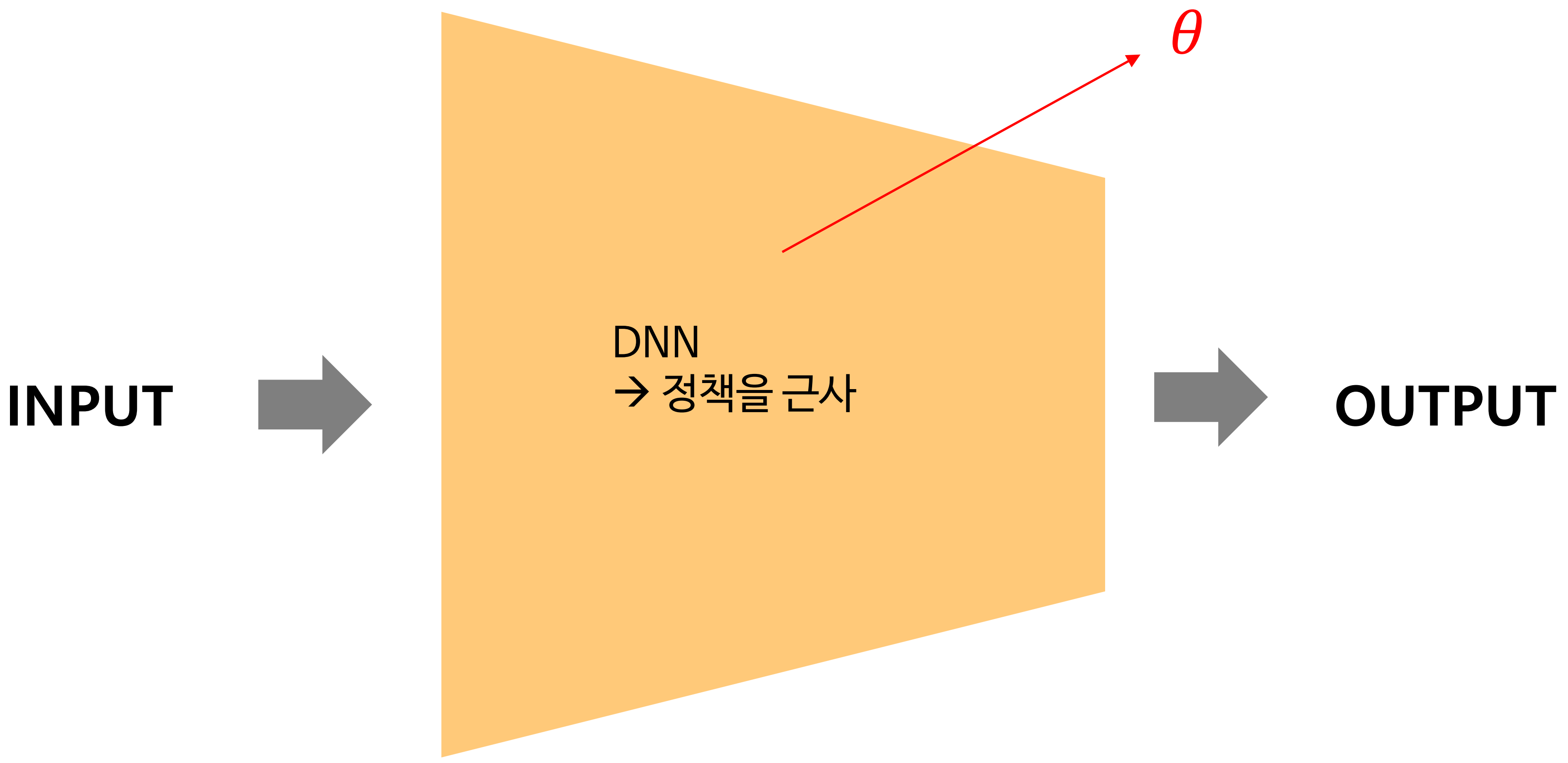
Value-based

Q라는 action-value function에 초점을 맞추어
Q function을 구하고 이것을 토대로 policy를 구하는 방법

ex) DQN : DNN을 이용하여 Q-function을 approximate 하고
이를 통해 policy를 구하는 방법

Policy-based

Policy 자체를 approximate
즉, DNN에서 나오는 것이 value가 아닌 policy자체!



정책 $\pi_\theta(a|s) = P[A_t = a|S_t = s, \theta]$

Policy-based 장단점

장점

1. 수렴
value-based에서는 value를 바탕으로 policy를 계산하기 때문에, value가 조금만 달라져도 policy가 크게 변화
→ 수렴에 불안정

But policy-based에서는 policy자체가 함수화 되어 학습하면서 조금씩 변화하기 때문에 안정적이고 부드럽게 수렴

2. Stochastic한 policy정책
→ "가위바위보" (가위, 바위, 보를 1/3씩 내는 것이optimal) 처럼 stochastic한 policy 배울 수 있음

단점

1. Local optimum에 빠질 수도
2. Variance가 높음

Policy Objective Function

Q function이 output으로 나오는 DNN이 아니라 Policy가 output으로 나오는 DNN을 만들 것!
DNN을 update할 기준이 DQN에서는 target과 현재 Q function 값의 차이였다면,
Policy Gradient에서는 Objective Function(=$J(\theta)$) 정의

Policy Gradient에서 목표는 Objective Function을 최대화 시키는 theta(policy의 parameter vector)을 찾아내는 것.
→ 그 방법이 gradient descent

Objective Function의 gradient를 구하는 세 가지 방법
1. Finite different policy gradient
2. Monte-Carlo policy gradient
3. Actor-Critic policy gradient

Policy Objective Function

에이전트가 정책 π_θ 에 따라서 가게 되는 "경로 " 를 생각해보자!

경로(trajectory) = 에이전트와 환경이 상호작용한 흔적

$$\tau = s_0, a_0, r_1, s_1, a_1, r_2, \dots, s_T$$

$J(\theta)$ = 경로 동안 받을 것이라고 기대하는 보상의 합(경로가 매번 달라지므로)

$$J(\theta) = \boldsymbol{E}\left[\sum_{t=0}^{T-1} r_{t+1} \mid \pi_\theta\right] = \boldsymbol{E}[r_1 + r_2 + r_3 + \cdots + r_T \mid \pi_\theta]$$

Policy Objective Function

$J(\theta)$를 기준으로 θ(정책신경망)을 어떻게 업데이트할 것인가?

→ θ에 대한 $J(\theta)$의 경사를 따라 올라가다(Gradient Ascent)

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

$\nabla_\theta J(\theta)$ = Policy Gradient

Policy Objective Function
1. Finite Difference Policy Gradient

- For each dimension $k \in [1, n]$
  - Estimate kth partial derivative of objective function w.r.t. θ
  - By perturbing θ by small amount ϵ in kth dimension

$$\frac{\partial J(\theta)}{\partial \theta_k} \approx \frac{J(\theta + \epsilon u_k) - J(\theta)}{\epsilon}$$

- Numerical Method
- 간단하지만, 비효율적인 방법
- But Policy가 미분 가능하지 않더라도 작동

Policy Objective Function

2. Monte-Carlo Policy Gradient

$$\nabla_\theta J(\theta) \sim \sum_{t=0}^{T-1} \nabla_\theta log \pi_\theta(a_t|s_t)\, G_t$$

Policy Objective Function

2. Monte-Carlo Policy Gradient

$$\nabla_\theta J(\theta) \sim \sum_{t=0}^{T-1} \nabla_\theta log \pi_\theta(a_t|s_t)\, G_t$$

$$\tau = s_0, a_0, r_1, s_1, a_1, r_2, \dots, s_T$$

$$J(\theta) = E\left[\sum_{t=0}^{T-1} r_{t+1}|\pi_\theta\right] = E_\tau[r_1|\pi_\theta] + E_\tau[r_2|\pi_\theta] + E_\tau[r_3|\pi_\theta] + \cdots$$

$$= \sum_{t=0}^{T-1} \underbrace{P(s_t, a_t|\tau)}_{(s_t, a_t)\text{일 확률}} \underbrace{r_{t+1}}_{(s_t, a_t)\text{일 때의 값}}$$

$$\boldsymbol{E}[f(x)] = \sum_x p(x) f(x)$$

Policy Objective Function

2. Monte-Carlo Policy Gradient

$$\nabla_\theta J(\theta) \sim \sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)\, G_t$$

양변에 미분 취하기

$$\nabla_\theta J(\theta) = \nabla_\theta E\left[\sum_{t=0}^{T-1} r_{t+1}|\pi_\theta\right]$$

$$= \boldsymbol{\nabla_\theta} \sum_{t=0}^{T-1} P(s_t, a_t|\tau) r_{t+1}$$

$$= \sum_{t=0}^{T-1} \boldsymbol{\nabla_\theta} P(s_t, a_t|\tau) r_{t+1} \qquad = \sum_{t=0}^{T-1} P(s_t, a_t|\tau) \frac{\boldsymbol{\nabla_\theta} P(s_t, a_t|\tau)}{P(s_t, a_t|\tau)} r_{t+1}$$

$$= \sum_{t=0}^{T-1} P(s_t, a_t|\tau) \boldsymbol{\nabla_\theta log} P(s_t, a_t|\tau) r_{t+1}$$

$$\frac{dlogx}{dx} = \frac{1}{x}$$

$$\frac{dlogf(x)}{dx} = \frac{df(x)/dx}{f(x)}$$

Policy Objective Function

2. Monte-Carlo Policy Gradient

$$\nabla_\theta J(\theta) \sim \sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)\, G_t$$

$$\sum_{t=0}^{T-1} P(s_t, a_t|\tau)\boldsymbol{\nabla_\theta log}P(s_t, a_t|\tau) r_{t+1}$$

$$= E_\tau[\sum_{t=0}^{T-1} \underline{\boldsymbol{\nabla_\theta log}P(s_t, a_t|\tau) r_{t+1}}]$$

$$\boldsymbol{E}[f(x)] = \sum_x p(x) f(x)$$

$$P(s_t, a_t|\tau) = \mathrm{P}(s_0, a_0, r_1, s_1, a_1, r_2, \dots, s_t, a_t|\theta)$$

$$= \mathrm{P}(s_0)\pi_\theta(a_0|s_0)\mathrm{P}(s_1|s_0, a_0)\pi_\theta(a_1|s_1)\mathrm{P}(s_2|s_1, a_1) \dots$$

$$logP(s_t, a_t|\tau)$$

$$= \log[\mathrm{P}(s_0)\pi_\theta(a_0|s_0)\mathrm{P}(s_1|s_0, a_0)\pi_\theta(a_1|s_1) \dots$$
$$+\mathrm{P}(s_t|s_{t-1}, a_{t-1})\pi_\theta(a_t|s_t)]$$

$$= \log\mathrm{P}(s_0) + \log\pi_\theta(a_0|s_0) + \log\mathrm{P}(s_1|s_0, a_0) + log\pi_\theta(a_1|s_1) \dots$$
$$+\log\mathrm{P}(s_t|s_{t-1}, a_{t-1}) + log\pi_\theta(a_t|s_t)$$

Policy Objective Function

2. Monte-Carlo Policy Gradient

$$\nabla_\theta J(\theta) \sim \sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)\, G_t$$

$$\boldsymbol{\nabla_\theta log} P(s_t, a_t|\tau)$$

$$= \boldsymbol{\nabla_\theta}[\cancel{\log P(s_0)} + \log\pi_\theta(a_0|s_0) + \cancel{\log P(s_1|s_0, a_0)} + log\pi_\theta(a_1|s_1) \ldots$$
$$+\cancel{\log P(s_t|s_{t-1}, a_{t-1})} + log\pi_\theta(a_t|s_t)]$$

$$= \boldsymbol{\nabla_\theta}\log\pi_\theta(a_0|s_0) + \boldsymbol{\nabla_\theta} log\pi_\theta(a_1|s_1) + \cdots + \boldsymbol{\nabla_\theta}\ log\pi_\theta(a_t|s_t)$$

$$\sum_{t=0}^{T-1} \boldsymbol{\nabla_\theta log} P(s_t, a_t|\tau) r_{t+1}$$

$$= \sum_{t=0}^{T-1} r_{t+1}[\boldsymbol{\nabla_\theta}\log\pi_\theta(a_0|s_0) + \boldsymbol{\nabla_\theta} log\pi_\theta(a_1|s_1) + \cdots + \boldsymbol{\nabla_\theta}\ log\pi_\theta(a_t|s_t)]$$

$$= \sum_{t=0}^{T-1} r_{t+1}\left(\sum_{t'=0}^{t} \boldsymbol{\nabla_\theta}\log\pi_\theta(a_{t'}|s_{t'})\right) \quad = \sum_{t=0}^{T-1} \boldsymbol{\nabla_\theta}\log\pi_\theta(a_t|s_t) \sum_{t'=t+1}^{T} r_{t'}$$

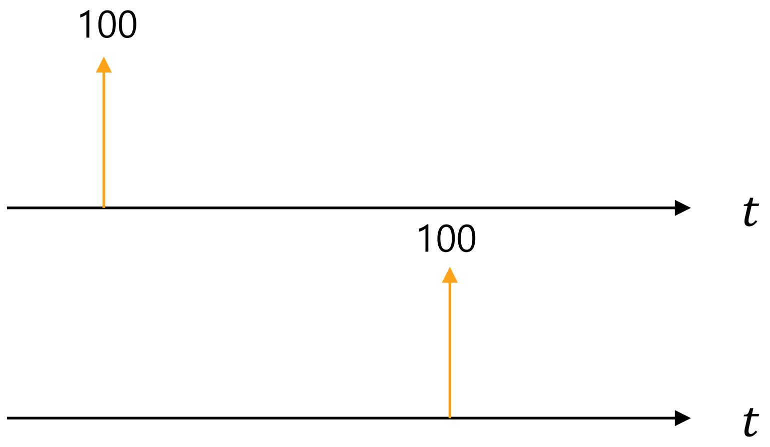
Policy Objective Function

2. Monte-Carlo Policy Gradient

$$\nabla_\theta J(\theta) \sim \sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)\, G_t$$

$$\boldsymbol{\nabla_\theta J(\theta)} = E_\tau\left[\sum_{t=0}^{T-1} \boldsymbol{\nabla_\theta} \log\pi_\theta(a_t|s_t) \sum_{t'=t+1}^{T} r_{t'}\right]$$

이때, $\sum_{t'=t+1}^{T} r_{t'}$ 와 같은 단순 보상의 합 → 문제!



감가율(discount factor) $0 \le \gamma \le 1$ 도입

Policy Objective Function

2. Monte-Carlo Policy Gradient

$$\boxed{\nabla_\theta J(\theta) \sim \sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)\, G_t}$$

$$\boldsymbol{\nabla_\theta J(\theta)} = E_\tau[\sum_{t=0}^{T-1} \boldsymbol{\nabla_\theta} \log\pi_\theta(a_t|s_t) \sum_{t'=t+1}^{T} r_{t'}]$$

$$\sim E_\tau[\sum_{t=0}^{T-1} \boldsymbol{\nabla_\theta} \log\pi_\theta(a_t|s_t) \sum_{t'=t+1}^{T} \gamma^{t'-t-1} r_{t'}]$$

$$\sum\nolimits_{t'=t+1}^{T} \gamma^{t'-t-1} r_{t'} = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots + \gamma^{T-t-1} r_T \rightarrow G_t$$

Policy Objective Function

2. Monte-Carlo Policy Gradient

$$\nabla_\theta J(\theta) = \boldsymbol{E}_{\boldsymbol{\tau}}\left[\sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)\, G_t\right]$$

강화학습에서는 E[]를 계산
하지 않고 Sampling

$$\nabla_\theta J(\theta) \sim \sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)\, G_t$$

$$\theta \longleftarrow \theta + \alpha \boldsymbol{\nabla_\theta J(\theta)} = \theta + \alpha \boxed{\sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)\, G_t}$$

Policy Objective Function

2. Monte-Carlo Policy Gradient

1. 한 에피소드를 현재 정책에 따라 실행

2. Trajectory를 기록

3. 에피소드가 끝난 뒤 G_t 계산

4. Policy gradient를 계산하여 정책 업데이트

5. (1~4) 반복

* Policy gradient = $\sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)\, G_t$

에피소드 마다 업데이트 → Monte-Carlo Policy gradient

Policy Objective Function

2. Monte-Carlo Policy Gradient

function REINFORCE

  Initialize θ arbitrarily

  for each episode $\{s_0, a_0, r_1, s_1, a_1, r_2, \dots, s_T\} \sim \pi_\theta$ **do**

    for t = 1 too T-1 **do**

      $\theta \longleftarrow \theta + \boldsymbol{\alpha \nabla_\theta} \log \pi_\theta(a_t|s_t) G_t$

    end for

  end for

  return $\boldsymbol{\theta}$

end function

Policy Objective Function

3. Actor-Critic Policy Gradient

Monte-Carlo Policy gradient는 한 에피소드가 끝나야지만 업데이트!
→ Actor-Critic Policy Gradient는 time-step마다 업데이트 하는 방법
→ DNN을 두 개 만들어서 Policy뿐 아니라 Q function도 approximate해서 gradient를 구하자!

Actor는 Policy를, Critic은 Q-function을 approximate

Policy Objective Function

3. Actor-Critic Policy Gradient

$$\tau = s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_T$$

$$\nabla_\theta J(\theta) = \boldsymbol{E}_{\boldsymbol{\tau}}\left[\sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)\, G_t\right]$$

Expectation을 쪼개자 → $(s_0 \sim\ a_t)$ + $(r_{t+1} \sim r_T)$

$$\sum_{t=0}^{T-1} E_{s_0,a_0,\ldots,s_t,a_t}[\boldsymbol{\nabla_\theta}\log\pi_\theta(a_t|s_t)]\, E_{r_{t+1},s_{t+1},\ldots,s_T,r_T}[G_t]$$

Q-function

$\boldsymbol{Q_{\pi\theta}(s_t, a_t)}$

Policy Objective Function

3. Actor-Critic Policy Gradient

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

$$\sim \sum_{t=0}^{T-1} \boldsymbol{\nabla_\theta} \log \pi_\theta(a_t|s_t) \underline{Q_{\pi\boldsymbol{\theta}}(s_t, a_t)}$$

알 수 있다면, 매 time-step마다 업데이트 가능!

$$Q_{\pi\boldsymbol{\theta}}(s_t, a_t) \approx \underline{Q_{\boldsymbol{W}}(s_t, a_t)}$$

Q-function approximate한 DNN으로

$$\sim \sum_{t=0}^{T-1} \boldsymbol{\nabla_\theta} \log \pi_\theta(a_t|s_t) Q_{\boldsymbol{W}}(s_t, a_t)$$

Policy Objective Function

3. Actor-Critic Policy Gradient - Baseline

State value function을 일종의 평균으로 사용해서
현재의 행동이 평균적으로 얻을 수 있는 value보다 얼마나 더 좋은 것인가를 계산
→ variance문제 해결

$$\boldsymbol{\nabla_{\theta} J(\theta)} \sim \log \pi_{\theta}(a_t|s_t) Q_{\boldsymbol{W}}(s_t, a_t)$$

$$\boldsymbol{\nabla_{\theta} J(\theta)} \sim \log \pi_{\theta}(a_t|s_t) (Q_{\boldsymbol{W}}(s_t, a_t) - V_v(s_t))$$

원래는 왼쪽 1, 오른쪽 2
→ 왼쪽 -1, 오른쪽 1

Policy Objective Function
3. Actor-Critic Policy Gradient - Baseline

이미 사용하고 있는 DNN이 2개! 따라서 V까지 DNN하면 너무 비효율적

$$Q(s_t, a_t) = \boldsymbol{E}[r_{t+1} + \gamma V(s_{t+1}) | s_t, a_t]$$ 이용

$$\boldsymbol{\nabla_\theta J(\theta)} \sim \boldsymbol{\nabla_\theta} \log \pi_\theta(a_t | s_t)(\underbrace{r_{t+1} + \gamma V(s_{t+1})}_{\text{큐함수}} - \underbrace{V_v(s_t)}_{\text{베이스라인}})$$

Policy Objective Function

3. Actor-Critic Policy Gradient - Baseline

1. Actor

- 정책을 근사 : θ
- $\boldsymbol{\nabla_\theta} \log \pi_\theta(a_t|s_t)(r_{t+1} + \gamma V(s_{t+1}) - V_v(s_t))$ 로 업데이트

2. Critic

- 가치함수(Value function)을 근사 : v
- $(r_{t+1} + \gamma V(s_{t+1}) - V_v(s_t))^2$ 의 오차함수로 업데이트

Actor의 loss function

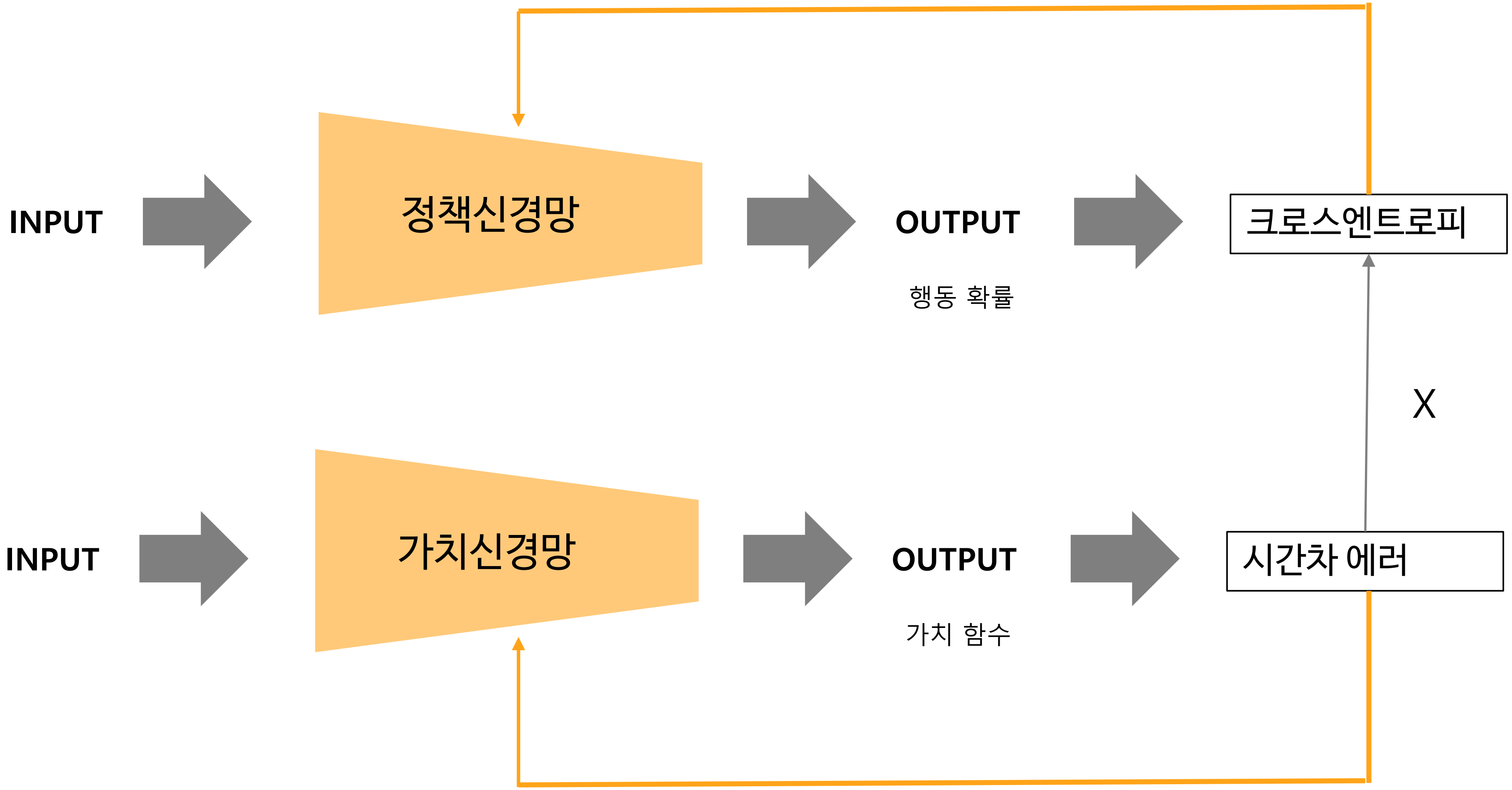
$$\underbrace{\boldsymbol{\nabla_\theta} \log \pi_\theta(a_t|s_t)}_{\text{크로스 엔트로피}}\underbrace{(r_{t+1} + \gamma V(s_{t+1}) - V_v(s_t))}_{\text{시간차 에러}}$$

Critic의 loss function

$$\underbrace{(r_{t+1} + \gamma V(s_{t+1}) - V_v(s_t))^2}_{\text{시간차 에러}}$$

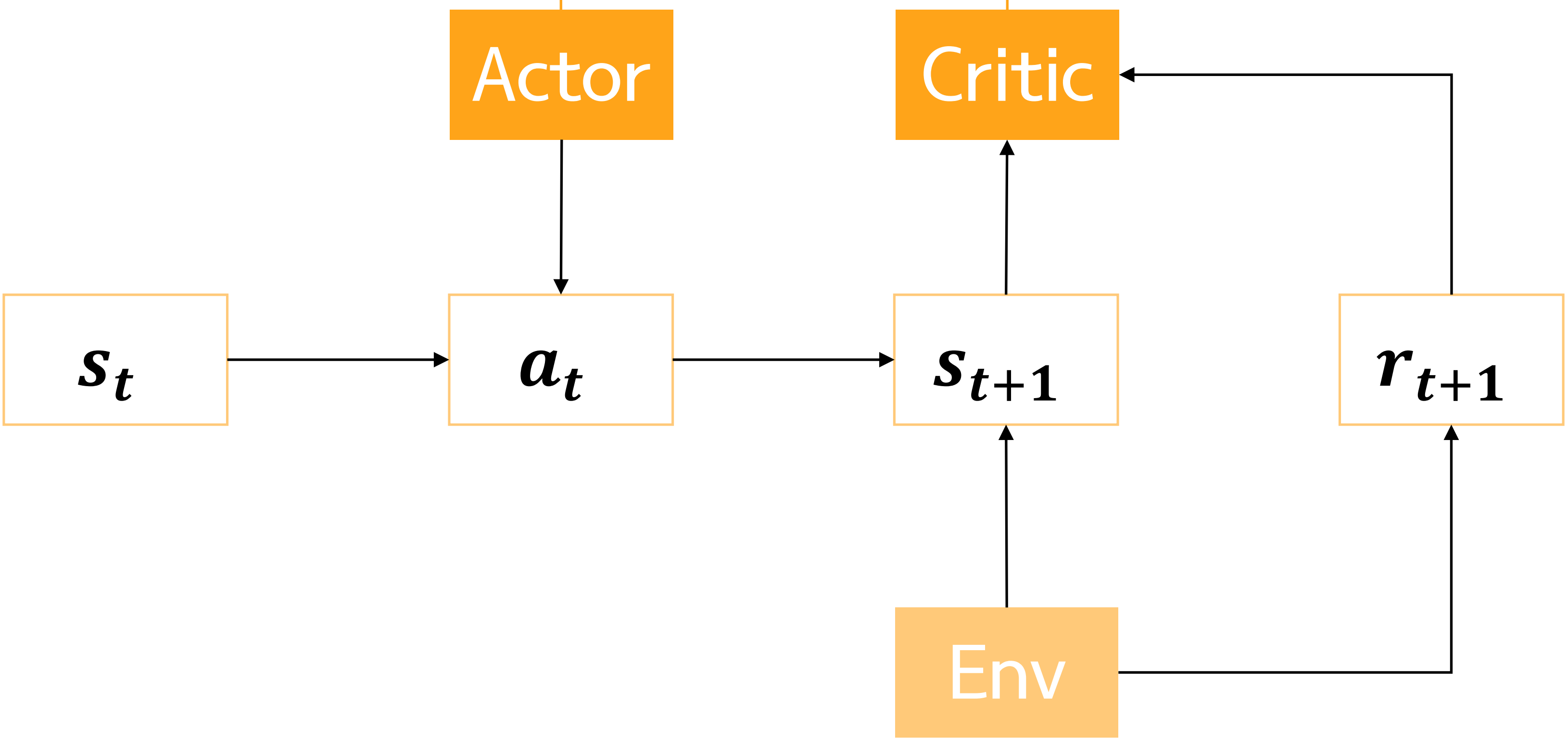
Policy Objective Function

3. Actor-Critic Policy Gradient - Baseline



Policy Objective Function
3. Actor-Critic Policy Gradient - Baseline

$$\nabla_{\boldsymbol{\theta}} \log \pi_{\theta}(a_t|s_t)(r_{t+1} + \gamma V(s_{t+1}) - V_v(s_t))$$



https://www.youtube.com/watch?v=gINks-YCTBs

http://www.modulabs.co.kr/RL_library/3305

https://github.com/dennybritz/reinforcement-learning/tree/master/PolicyGradient