<13. Policy-based Reinforcement Learning Methods>

* policy-based methods

- value function과 상관없이, state와 action을 mapping 하는 parameterized policy를 직접적으로 학습
- policy를 추출하여 method가 act함.

* Learning a parameterized policy

- policy 매개변수에 대해서, some scalar performance measure의 기울기 기반으로 매개변수 학습
  - θ : policy's parameter vector
  - policy : $\pi_\theta(a|s)$, performance measure : $J(\theta) = v^{\pi_\theta}(s)$
  - $J(\theta)$를 최대화하는 방향으로 학습. $\underbrace{\theta_{\pi'} = \theta_\pi + \alpha \cdot \nabla_\theta J(\theta_\pi)}_{\text{gradient ascent}}$
  - $J(\theta)$를 최대화하는 $\pi_\theta(a|s)$의 θ 발견이 목적
    - In episodic environment, $J(\theta) = v_N^{\pi_\theta}(s_0)$
    - In continuing environment, $J(\theta) = \sum_{s \in S} \mu^{\pi_\theta}(s) \cdot v^{\pi_\theta}(s)$

      ↳ π_θ : stationary policy

      $\mu^{\pi_\theta}(s)$: state distribution

- most common policy parameterization way : soft-max preferences
  - $\pi_\theta(a|s) = \dfrac{e^{h(s,b,\theta)}}{\sum_b e^{h(s,b,\theta)}}$
  - $h(s,b,\theta)$: parameterized numerical preferences for each state-action pair and can be parameterized arbitrarily.

* Policy Gradient Theorem

- policy 변수 θ에 대한 performance measure의 gradient는 $\pi_\theta(a|s)$, $\mu^{\pi_\theta}(s)$, $Q^{\pi_\theta}(s,a)$로 표현됨
  - $J(\theta) = \sum_{s\in S} \mu^{\pi_\theta}(s) \cdot V^{\pi_\theta}(s) = \underline{\sum_{s\in S} \mu^{\pi_\theta}(s) \cdot \sum_{a\in A} \pi_\theta(a|s)\, Q^{\pi_\theta}(s,a)}$

$$\nabla_\theta J(\theta) \propto \sum_{s\in S} \mu^{\pi_\theta}(s) \cdot \sum_{a\in A} Q^{\pi_\theta}(s,a)\, \nabla_\theta \pi_\theta(a|s)$$

$$= \sum_{s\in S} \mu^{\pi_\theta}(s) \cdot \sum_{a\in A} \pi_\theta(a|s) \cdot Q^{\pi_\theta}(s,a) \cdot \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}$$

$$= E_{\pi_\theta}\left[\sum_{a\in A} \pi_\theta(a|S_t) \cdot Q^{\pi_\theta}(S_t,a) \cdot \nabla_\theta \ln(\pi_\theta(a|S_t))\right]$$

$$= E_{\pi_\theta}[Q^{\pi_\theta}(S_t,A_t)\, \nabla_\theta \ln \pi_\theta(A_t|S_t)].$$

* 장점과 단점

- 장점
  - Better convergence properties
  - Effective in high-dimensional or continuous action space
  - Can learn stochastic (randomized) policies
- 단점
  - Easy to converge to a local rather than global optimum
  - Evaluating a policy is typically inefficient and high variance.

* Monte-Carlo Policy Gradient: REINFORCE

- $\nabla_\theta J(\theta) \propto E_{\pi_\theta}[Q^{\pi_\theta}(S_t,A_t)\, \nabla_\theta \ln \pi_\theta(A_t|S_t)] = E_{\pi_\theta}[G_t\, \nabla_\theta \ln \pi_\theta(A_t|S_t)]$.
  (by Policy Gradient Theorem)
- G_t : <u>total</u> (discount) rewards or (costs) from the decision epoch t to the end of episode based on sample trajectory and $Q^{\pi_\theta}(S_t,A_t) = E_{\pi_\theta}[G_t|S_t,A_t]$
- gradient ascent: $\underline{\theta_{\pi'} = \theta_\pi + \alpha \cdot G_t \cdot \nabla_\theta \cdot \ln \pi_\theta(A_t|S_t)}$

* Algorithm step

① 초기화: policy 매개변수 θ를 초기화

② given policy π_θ에 대해서, sample episode trajectory $s_1, a_1, r, s_2, a_2, \dots, s_T$ 생성

③ epoch $t = 1, \dots, T$에 대해

③-1: total reward $G_t \leftarrow \sum_{k=t}^{T} \gamma^{k-t} \cdot r_k$ 추정.

③-2: policy 매개변수 $\theta \leftarrow \theta + \alpha \cdot G_t \nabla_\theta \cdot \ln \pi_\theta(a_t | s_t)$ 업데이트.

④ 또 다른 sample episode trajectory가 있다면, ②으로 돌아가서 진행.

* Some Implications

- $\nabla_\theta \ln \pi_\theta(A_t|S_t)$는 A_t의 S_t 재방문 확률을 증가시키는 방향임
- 이로 인해 위 direction 따르는 policy 매개변수 양과 총 보상이 비례 & action probability에 반비례
- This update는 가장 높은 보상을 지향하는 방향으로 policy 매개변수가 움직이도록 한다.

* Revised policy algorithm

- action value 와 arbitrary baseline $b(s)$ 비교 위해 아래와 같이 일반화 가능.
  - $\nabla_\theta J(\theta) \propto \sum_{s \in S} \mu^{\pi_\theta}(s) \sum_{a \in A} \left(Q^{\pi_\theta}(s,a) - b(s) \right) \nabla_\theta \pi_\theta(a|s)$
  - $b(s)$는 state에 따른 어떠한 함수가 될 수 있음.
  - The above equation remains valid because the subtracted quantity is zero:

    $\sum_{a \in A} b(s) \cdot \nabla_\theta \pi_\theta(a|s) = b(s) \sum_{a \in A} \nabla_\theta \pi_\theta(a|s) = b(s) \cdot \nabla_\theta 1 = 0.$
- update policy for the policy parameter is revised as:
  - $\theta_{\pi'} = \theta_\pi + \alpha \cdot \left(G_t - b(S_t) \right) \nabla_\theta \ln \pi_\theta(A_t|S_t)$
  - 이렇게 하면 update의 기댓값은 그대로 두면서 분산을 줄일 수 있다.

* Actor-critic

- policy와 value function 두 개를 학습하는 것을 actor-critic method 라고 함
  - actor: learned policy, critic: learned value function
- Actor-Critic은 2개의 모델로 구성됨
  - actor model: 정책 π_θ에 대해 θ를 업데이트
  - critic model: value function ($V_w(s)$ or $Q_w(s,a)$) 결정하는 매개변수 w를 업데이트.

* One-step Actor-Critic method (episodic environment)

① 초기화: 정책 변수 θ, state value function 변수 w, initial state s 초기화 $(n=1)$

② $t=1,\dots,T$에 대해서

②-1: 주어진 state s에 대해, 주어진 policy π_θ에 따라 take action a.

②-2: reward r과 state s' 관찰

②-3: temporal difference $\Delta \leftarrow r+\gamma\cdot V_{w,t+1}(s') - V_{w,t}(s)$ 를 관찰

②-4: 정책 변수 $\theta \leftarrow \theta + \alpha^\theta \Delta \nabla_\theta \ln \pi_\theta(a|s)$ 업데이트

②-5: 가치함수 변수 $w \leftarrow w + \alpha^w \Delta \nabla_w V_{w,t}(s)$를 업데이트

②-6: $s \leftarrow s'$

③ $n<N$이면, $n\leftarrow n+1$로 업데이트 & step ②로.

* Actor-Critic with Eligibility Traces method (episodic environment)

① 초기화 : trace-decay rates $\lambda^\theta, \lambda^w \in [0,1]$. 정책 변수 θ, 가치함수 변수 w 초기화. $n=1$

② 초기 상태 s 초기화. $z^\theta \leftarrow 0$, $z^w \leftarrow 0$

③ epoch $t=1,\dots,T$ 에 대해서

③-1 : 주어진 state s에 대해서, given policy π_θ에 따라 take action a

③-2 : 보상 r과 다음 state s'를 관찰

③-3 : temporal difference $\Delta \leftarrow r + \delta \cdot V_{w,t+1}(s') - V_{w,t}(s)$ 관찰

③-4 : 정책 변수를 다음과 같이 업데이트 :

$z^\theta \leftarrow \lambda^\theta z^\theta + \nabla_\theta \ln \pi_\theta(a|s)$ (z에 축적 시킴)

$\theta \leftarrow \theta + \alpha^\theta \Delta z^\theta$

③-5 : 가치함수 변수를 다음과 같이 업데이트.

$z^w \leftarrow \lambda^w z^w + \nabla_w V_{w,t}(s)$

$w \leftarrow w + \alpha \cdot \Delta z^w$

③-6 : $s \leftarrow s'$

④ $n < N$ 이라면, $n \leftarrow n+1$로 업데이트하고 step ②.

* Actor-Critic with Eligibility Traces method (continuing environments)

① 초기화 : trace-decay rates $\lambda^\theta, \lambda^w \in [0,1]$, 정책 변수 θ, 가치함수 변수 w 초기화

② 초기 상태 s 초기화. $z^\theta \leftarrow 0$, $z^w \leftarrow 0$, $\bar{r} = 0$

③ epoch $t = 1,\dots,\infty$ 에 대해서

③-1 : Given State s, policy π_θ에 대해서, take an action a.

③-2 : reward r과 next state s' 관찰

③-3 : temporal difference $\Delta \leftarrow r - \bar{r} + \delta \cdot V_w(s') - V_w(s)$ 관찰

③-4 : average reward per stage $\bar{r} \leftarrow \bar{r} + \alpha^{\bar{r}} \cdot \Delta$ 업데이트

③-5 : 정책 변수 θ 업데이트 : $z^\theta \leftarrow \lambda^\theta z^\theta + \nabla_\theta \ln \pi_\theta(a|s)$, $\theta \leftarrow \theta + \alpha \cdot \Delta z^\theta$

③-6 : 가치함수 변수 w 업데이트 : $z^w \leftarrow \lambda^w z^w + \nabla_w V_w(s)$, $w \leftarrow w + \alpha^w \cdot \Delta z^w$

③-7 : $s \leftarrow s'$