Reinforcement Learning - 3

AI & OPTIMIZATION LAB

김 태 민

INU 인천대학교
INCHEON NATIONAL UNIVERSITY

Contents

I On-Policy와 Off-Policy의 차이

- **On-Policy란?**
  - 행동을 선택하는 Policy와 학습을 하는 Policy가 같아야 학습이 가능
    - → 1번이라도 학습을 해서 Policy improvement를 시킨 순간, 그 Policy가 했던 과거의 Experience들은 모두 사용 불가
    - → 현재 **데이터를 취득한 Policy와 학습되는 Policy가 같아야 한다.**
    - → 데이터 효율성이 떨어진다.
    - → **예시 : SARSA**

- **Off-Policy란?**
  - 행동을 선택하는 Policy와 학습하는 Policy가 같지 않아도 학습이 가능
    - → 과거의 Policy를 통해 취한 행동을 포함한 **경험데이터를 통해서 현재의 Policy 학습 가능**
    - → 사람이 한 데이터도 학습을 시킬 수 있다.
    - → 과거 데이터로만 학습하는 경우는 Offline-RL이라고 부른다.
    - → **예시 : Q-learning**

On-Policy와 Off-Policy의 차이

- **On-Policy와 Off-Policy 알고리즘 정리**

	Value Based	Policy Based	Actor-Critic
On-Policy	• Monte Carlo Learning • TD(0) • SARSA	• REINFORCE • REINFORCE with Advantage	• A2C • A3C • TRPO • PPO
Off-Policy	• Q-Learning • DQN • Double DQN • Dueling DQN		• DDPG • TD3 • SAC • IMPALA

- **Trust Region Policy Optimization (TRPO)**
  - TRPO는 **Stochastic Policy 기반의 Policy Optimization 기법**이다.
  - Trust Region은 Performance가 상승하는 방향으로 업데이트를 보장할 수 있는 구간을 뜻함.
  - TRPO는 이를 이용해 더 나은 Policy로 업데이트하기 위한 Optimization 기법에 대한 방법론

Algorithm 1 Policy iteration algorithm guaranteeing non-decreasing expected return η

Initialize π_0.

for $i = 0, 1, 2, \ldots$ until convergence **do**

Compute all advantage values $A_{\pi_i}(s, a)$.

Solve the constrained optimization problem

$$\pi_{i+1} = \arg\max_{\pi} \left[L_{\pi_i}(\pi) - C D_{\mathrm{KL}}^{\max}(\pi_i, \pi) \right]$$

→ Policy improvement

where $C = 4\epsilon\gamma/(1-\gamma)^2$

and $$L_{\pi_i}(\pi) = \eta(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a|s) A_{\pi_i}(s, a)$$

→ Policy evaluation

end for

- **Trust Region Policy Optimization (TRPO)를 위한 사전 준비**

Stochastic Policy에 대한 Expected discounted reward 식

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right], \text{ where}$$
$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t).$$

- π의 Discounted Reward의 기대값

Kakade & Langford (2002)의 이론과
TRPO의 Appendix에 따르면

Expectation of new Policy

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \ldots \sim \tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t)\right]$$
$$= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a)$$

- $\tilde{\pi}$은 Policy π에 대해 Advantage를 취한 Policy
- 다음 식은 그 때의 Expected Return
- 이 식의 의미
  - → Nonnegative expected advantage를 가지면 Policy performance의 상승을 보장할 수 있다.
  - → 하지만 추정과 근사의 과정에서 error가 존재해 모든 state가 Nonnegative expected advantage를 충족시키기는 어렵다.

- **Trust Region Policy Optimization (TRPO)를 위한 사전준비**

Local approximation of new Policy

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a).$$

L을 Maximize하는 New Policy를 찾아가는 과정으로 최적화가 진행

- 다음 식은 Advantage Policy $\tilde{\pi}$ 로부터 직접 sample을 얻기 어려우므로, 간접적으로 Local policy π로부터 얻으려는 식이다.

Conservative policy iteration

$$\eta(\pi_{new}) \geq L_{\pi_{old}}(\pi_{new}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$
$$\text{where } \epsilon = \max_s |\mathbb{E}_{a\sim\pi'(a|s)}[A_\pi(s,a)]|.$$

- 기존의 Policy를 업데이트함에 있어 Approximation error에 의해 Improvement가 보장되지 못할 수 있는 상황을 미연에 방지할 수 있음을 알려주는 식
- α는 기존 Policy와 새로운 Policy간의 업데이트 비율을 결정하는 parameter로 사용

위 식을 변형하여 Policy iteration에 사용

- **Theorem 1**

Theorem 1

Let $\alpha = D_{TV}^{max}(\pi_{old}, \pi_{new})$,

$D_{TV}^{max}(\pi, \tilde{\pi}) = \max_s D_{TV}(\pi(.|s)||\tilde{\pi}(.|s))$,

$\eta(\pi_{new}) \geq L_{\pi_{old}}(\pi_{new}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2$

where $\epsilon = \max_{s,a}|A_\pi(s,a)|$.

Total variation divergence, KL divergence사이의 관계식

$$D_{TV}(p||q)^2 \leq D_{KL}(p||q)$$

- α를 기존 Policy와 새로운 Policy간의 Distance measure로, ε을 적절히 변형해 다음 식을 도출

- KL Divergence란?
  - 두 확률분포의 차이를 계산하는 데 사용하는 함수
  - 어떤 이상적인 분포에 대해, 그 분포를 근사하는 다른 분포를 사용해 샘플링을 한다면 발생할 수 있는 정보 엔트로피 차이를 계산한다.
  - 결과적으로, p와 q의 Cross entropy에서 p의 entropy를 뺀 값

Policy iteration을 진행하는 알고리즘 식

$$\eta(\tilde{\pi}) \geq L_\pi(\tilde{\pi}) - CD_{KL}^{max}(\pi, \tilde{\pi}),$$

$$\text{where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2.$$

- **Policy iteration algorithm**

Algorithm 1 Policy iteration algorithm guaranteeing non-decreasing expected return η

Initialize π_0.

for $i = 0, 1, 2, \ldots$ until convergence **do**

Compute all advantage values $A_{\pi_i}(s, a)$.

Solve the constrained optimization problem

$$\pi_{i+1} = \arg\max_{\pi} \left[L_{\pi_i}(\pi) - C D_{\mathrm{KL}}^{\max}(\pi_i, \pi) \right]$$

→ Policy improvement

where $C = 4\epsilon\gamma/(1-\gamma)^2$

and $$L_{\pi_i}(\pi) = \eta(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a|s) A_{\pi_i}(s, a)$$

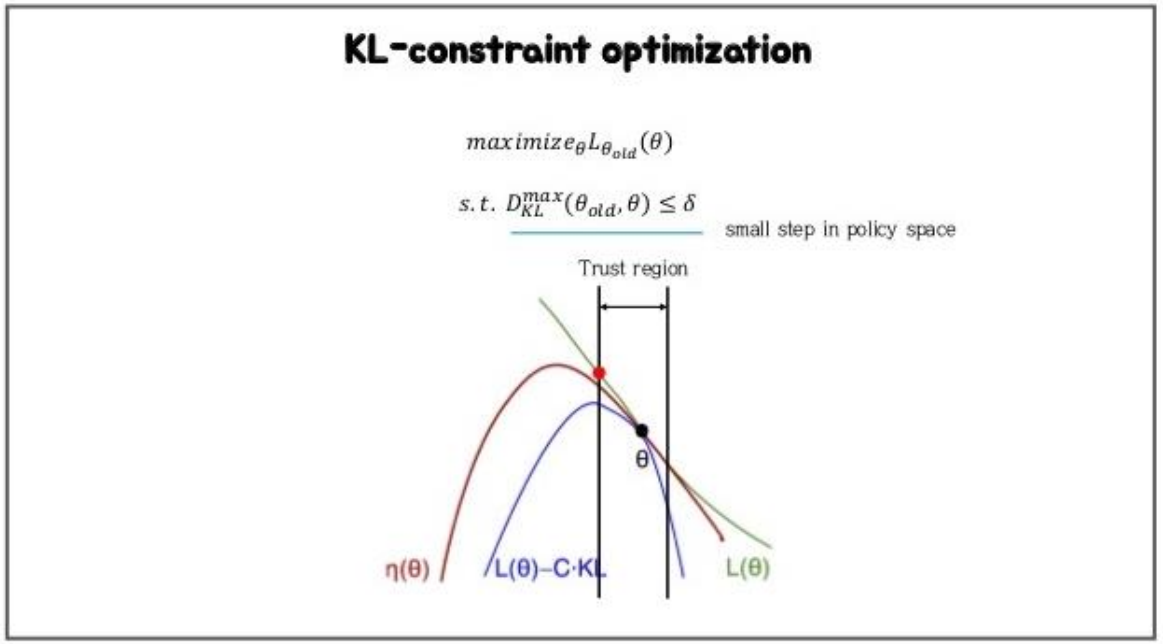
→ Policy evaluation

end for

- **Optimization of Parameterized Policies**

$$\eta(\theta) \geq L_{\theta_{old}}(\theta) - CD_{KL}^{max}(\theta_{old}, \theta)$$

- 위 식을 보면 Surrogate Function인 우변의 Maximization을 통해 η의 개선이 보장된다.
- 하지만, C가 실제로는 매우 큰 값이 들어가게 되면서 lower bound에 대한 step size가 매우 작아지는 문제 발생
- 그래서 KL divergence를 이용해 penalty를 주는 방식이 아닌 Trust region constraint를 구현했다.



TRPO

- **Trust Region Policy Optimization (TRPO)의 Idea**

Objective Function

$$\underset{\theta}{\text{maximize}}\ L_{\theta_{\text{old}}}(\theta)$$
$$\text{subject to } D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta.$$

Surrogate Objective Function with Monte-Carlo

$$\underset{\theta}{\text{maximize}}\ \mathbb{E}_{\sim\rho_\theta,\sim}\left[\frac{\pi_\theta(a|s)}{q(a|s)} Q_{\theta_{old}}(s,a)\right]$$
$$\text{subject to } \mathrm{E}_{s\sim\rho_{\theta_{old}}}[D_{KL}(\pi_{\theta_{old}}(.|s)||\pi_\theta(.|s))] \leq \delta.$$

- 위 목적함수에서 제약식이 모든 State에 대해 만족해야하는데, 현실적으로 모든 State에 대한 KL Divergence를 계산하는 것은 어려움
- 이를 Practical하게 바꾸기 위해, 몬테카를로 기법을 사용해 Surrogate objective function을 최적화

- 목적함수의 Maximize를 위한 θ값을 찾기 위해, KL Divergence의 Hessian을 구하고, 이를 기반으로 Fisher Information Matrix를 구해 최적화를 진행
  - → Hessian을 구한다는 것은 2차 미분 값들을 구한다는 뜻
    - → **TRPO와 PPO의 차이점 (PPO는 1차 미분만으로 계산이 끝난다.)**

PPO

- **Proximal Policy Optimization Algorithm (PPO)**
  - PPO의 Concept은 TRPO의 Surrogate Objective Function을 푸는 과정이 복잡하니, Clipping 등의 방법으로 단순화 시켜, 1차 미분으로 Approximate 하는 것
  - 정책 + 가치 기반 강화학습 알고리즘
  - 확률적 경사 상승법을 사용하여 Surrogate 목적 함수 최적화
  - 다수의 epoch 동안 미니배치 업데이트 수행

  - **장점**
    - 비교적 단순한 구현
    - 다양한 환경에서 평균적으로 좋은 성능
    - 낮은 샘플 복잡도
    - 짧은 연산 시간

- **Proximal Policy Optimization Algorithm (PPO)의 특징**
  - TRPO의 Surrogate objective Function을 최대화하는 것이 목표

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right]$$

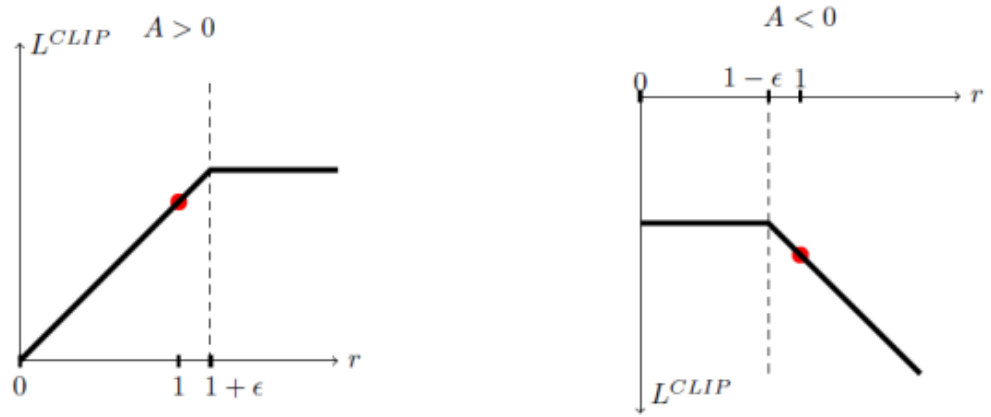
Probability ratio가 과도하게 커지면서 학습에 실패하거나 성능이 저하되는 문제 발생

  - PPO는 Clipping 기법을 통해 간단하게 이 문제를 해결

→ Probability ratio를 숫자 1에서부터 멀리 떨어져 있는(기존 Policy와 많이 다른) Policy에 Penalty를 주는 방법

- **Clipped Surrogate Objective Function**
  - 계산적으로 효율적인 Penalty를 적용하고 과도한 Policy 업데이트를 방지

$$L^{CLIP}(\theta) = \hat{E}_t[\min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$$



  - 입실론 값은 Hyperparameter로 0.1이나 0.2정도를 사용한다.
  - Clipping으로 믿을 수 있는 부분에서만 업데이트를 하기 때문에, 안정적으로 쌓은 데이터를 여러 번 사용할 수 있다.

- **Generalized Advantage Estimation (GAE)**
  - A3C와 같은 PG방식의 알고리즘에서는 T time step동안의 Policy에 대해 Sample을 얻고 업데이트
  - Time step T까지만 고려하는 Advantage estimator를 사용

$$\hat{A}_t = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V(s_T)$$

  - PPO에서는 이의 Generalized 버전인 GAE의 Truncated version 사용 (람다가 1인 경우 위와 동일)

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1},$$

$$where \quad \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

PPO

- **Network Update**

Actor Network Update

$$\max L_t^{CLIP} = \hat{E}_t[\min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$$

Critic Network Update

$$\min L_t^{VF} = (V_\theta(s_t) - V_t^{targ})^2$$

Actor - Critic Network in PPO

$$\max L_t^{CLIP+VF+S}(\theta) = \hat{E}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)]$$

c_1, c_2: coefficients, S: entropy bonus

- **Network Update**

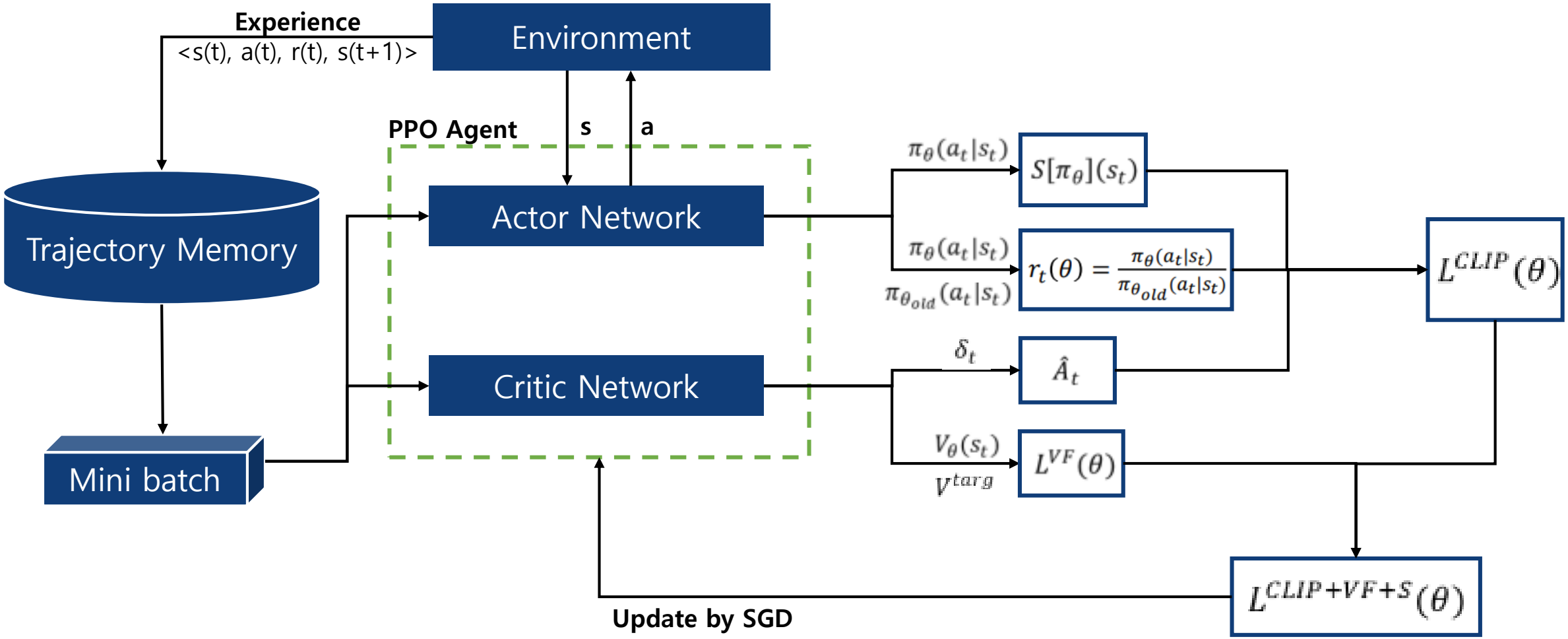
Algorithm 1 PPO, Actor-Critic Style

for iteration=$1, 2, \ldots$ **do**
  for actor=$1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{\text{old}} \leftarrow \theta$
end for

- 매 iteration마다 N actor가 T timestep만큼의 데이터를 모아 업데이트하는 방식

  → 따라서, N*T개의 데이터를 이용해 Surrogate loss를 형성하고, minibatch SGD를 적용해 이를 업데이트

- **PPO 구조**


Experience
<s(t), a(t), r(t), s(t+1)>
Environment
PPO Agent
s
a
Actor Network
Critic Network
Trajectory Memory
Mini batch
$\pi_\theta(a_t|s_t)$
$S[\pi_\theta](s_t)$
$\pi_\theta(a_t|s_t)$
$\pi_{\theta_{old}}(a_t|s_t)$
$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$
$L^{CLIP}(\theta)$
δ_t
$\hat{A}_t$
$V_\theta(s_t)$
V^{targ}
$L^{VF}(\theta)$
$L^{CLIP+VF+S}(\theta)$
Update by SGD

참고문헌

- https://newsight.tistory.com/250
- https://data-newbie.tistory.com/543
- https://engineering-ladder.tistory.com/69
- https://ropiens.tistory.com/82
- https://github.com/CUN-bjy/rl-paper-review
- Trust region policy optimization, J. Schulman 외 4명, International conference on machine learning, 2015
- https://hyunw.kim/blog/2017/10/27/KL_divergence.html
- Proximal policy optimization algorithms, J. Schulman 외 4명, arXiv preprint arXiv:1707.06347, 2017
- 파이썬과 케라스로 배우는 강화학습, 위키북스
- 수학으로 풀어보는 강화학습 원리와 알고리즘, 위키북스
- 강화학습 / 심층강화학습 특강, 위키북스

감사합니다