동아대학교
DONG-A UNIVERSITY

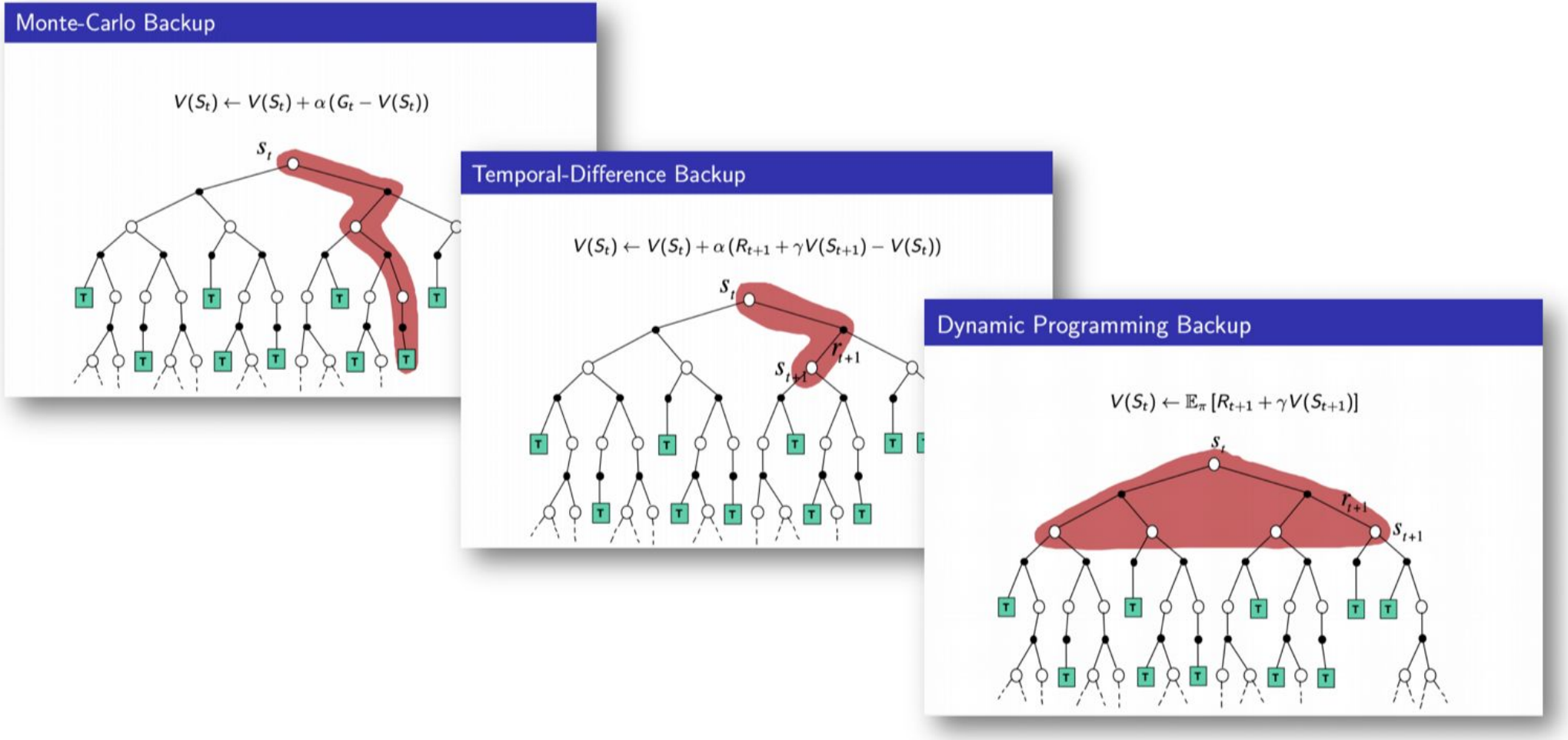
[실증적SW개발프로젝트]

RLHF기반 로봇 팔 제어 프로그램 개발

2143841 권은주
1824751 진현석
2051505 조현진

CONTENTS

MC vs. TD vs. DP


Monte-Carlo Backup
$V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$
S_t
Temporal-Difference Backup
$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$
S_t
r_{t+1}
S_{t+1}
Dynamic Programming Backup
$V(S_t) \leftarrow \mathbb{E}_\pi[R_{t+1} + \gamma V(S_{t+1})]$
S_t
r_{t+1}
S_{t+1}

On-policy vs. off-policy

Target policy: 현재 학습하고 있는 policy (π)

Behavior policy: 행동을 결정하는 policy (μ)

만약 이 둘이 다르면 off-policy, 일치하면 on-policy

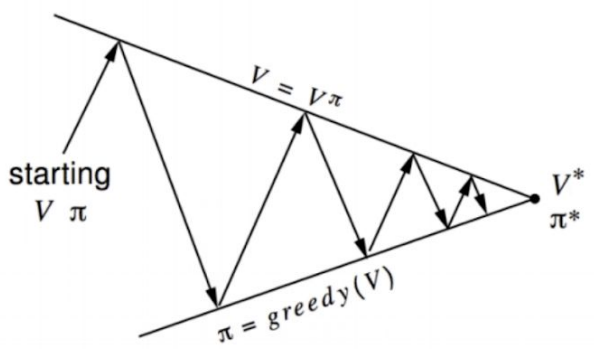
- On-policy learning
  - "Learn on the job"
  - Learn about policy π from experience sampled from π
- Off-policy learning
  - "Look over someone's shoulder"
  - Learn about policy π from experience sampled from μ

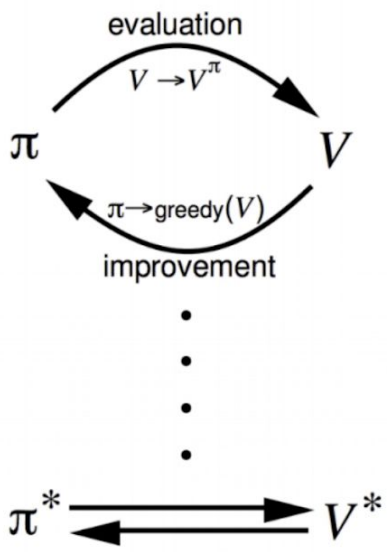
https://www.davidsilver.uk/teaching/

- On-policy이 일반적으로 더 단순
- Off-policy은 추가적인 개념과 notation이 필요
- Off-policy가 보통 variance가 보통 더 크고 수렴하는 데 오래 걸림. 하지만 bias는 낮아짐.
- Off-policy는 on-policy를 special case로 하는
  더 강력하고 일반적으로 적용 가능한 학습임
- 예를 들어, human expert로부터 만들어진 데이터를 이용해서도 학습에 적용 가능

On-policy Monte-Carlo control

Generalised Policy Iteration (Refresher)





Policy evaluation Estimate v_π
e.g. Iterative policy evaluation

Policy improvement Generate $\pi' \geq \pi$
e.g. Greedy policy improvement

Policy evaluation Monte-Carlo policy evaluation, $V = v_\pi$?

Policy improvement Greedy policy improvement?

Model-free control에 있어서 State value function을 이용한 policy를 개선하는 것의 문제점:

V(s)에 기반한 policy update에는 model이 여전히 필요함

$$\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ \mathcal{R}_s^a + \mathcal{P}_{ss'}^a V(s')$$

Key 아이디어: Q(s, a)에 기반한 policy update는 model-free!

$$\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ Q(s, a)$$

"The Monte Carlo methods for this are essentially the same as just presented for state values, except now we talk about visits to a state–action pair rather than to a state."

SB textbook, p. 96

Example of Greedy Action Selection

"Behind one door is tenure - behind the other is flipping burgers at McDonald's."

- There are two doors in front of you.
- You open the left door and get reward 0
  $V(left) = 0$
- You open the right door and get reward $+1$
  $V(right) = +1$
- You open the right door and get reward $+3$
  $V(right) = +2$
- You open the right door and get reward $+2$
  $V(right) = +2$

$\vdots$

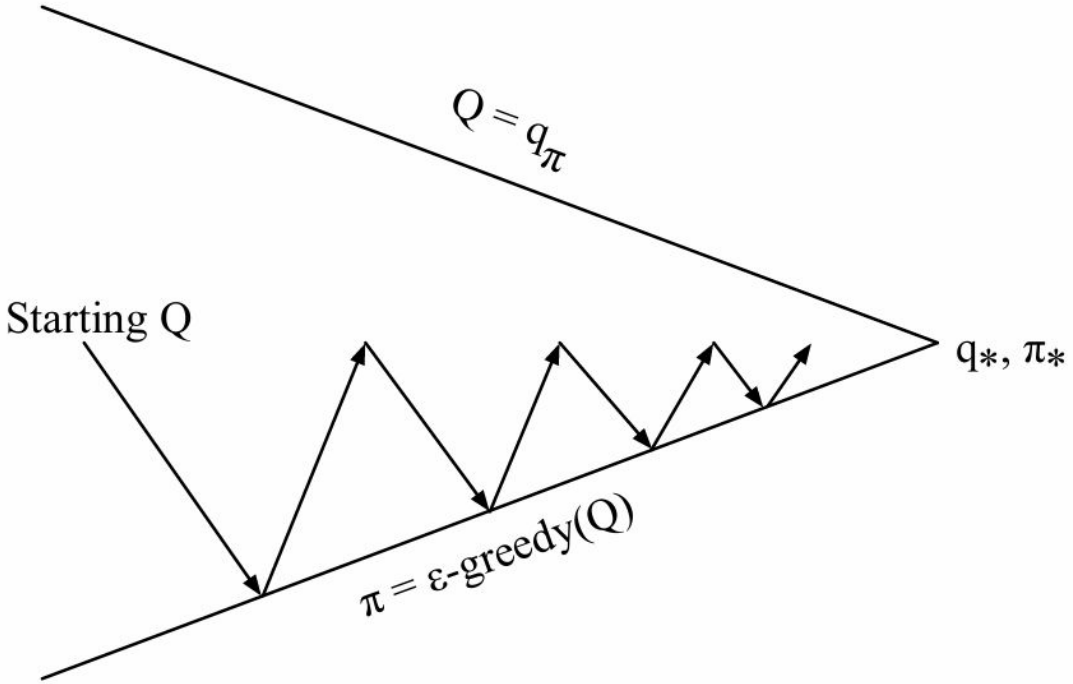
- Are you sure you've chosen the best door?

ϵ-Greedy Exploration

- Simplest idea for ensuring continual exploration
- All m actions are tried with non-zero probability
- With probability $1-\epsilon$ choose the greedy action
- With probability ϵ choose an action at random

$$\pi(a|s) = \begin{cases} \epsilon/m + 1 - \epsilon & \text{if } a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q(s,a) \\ \epsilon/m & \text{otherwise} \end{cases}$$

Monte-Carlo Control



Every episode:

Policy evaluation Monte-Carlo policy evaluation, $Q \approx q_\pi$

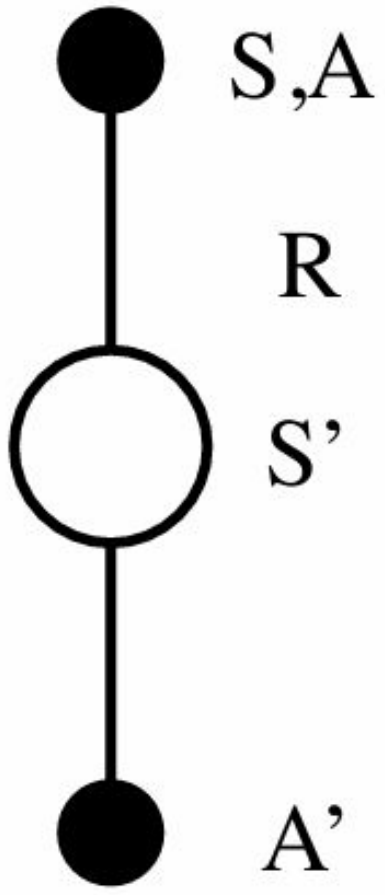
Policy improvement ϵ-greedy policy improvement

▪ Temporal-Difference Control

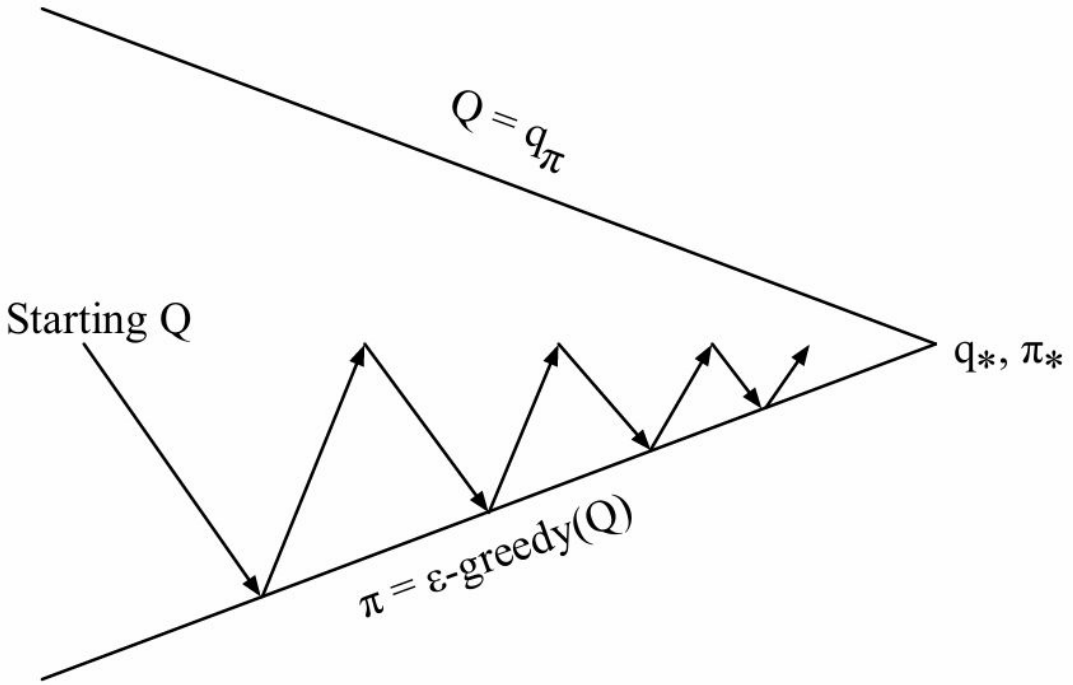
Sarsa

$$Q(S,A) \leftarrow Q(S,A) + \alpha\left(R + \gamma Q(S',A') - Q(S,A)\right)$$

state S에서 action A를 해 reward R을 받고 state S'에 가 action A'를 한다.
그 즉시 value function을 업데이트



On-Policy Control With Sarsa

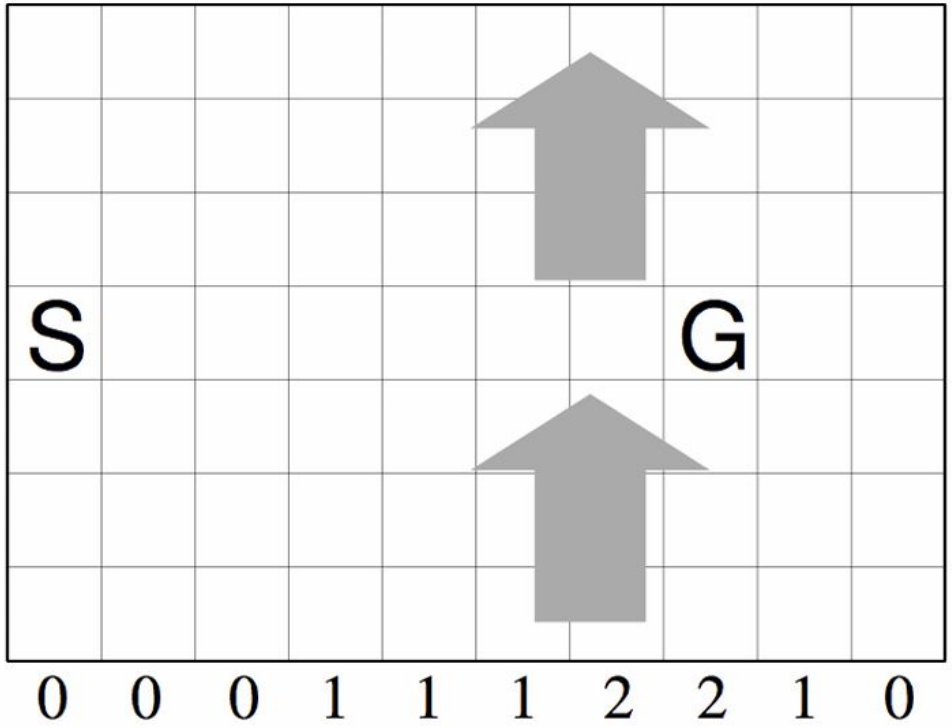


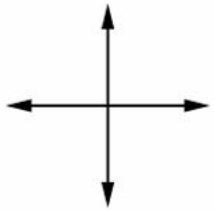
Every time-step:

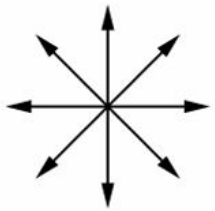
Policy evaluation Sarsa, $Q \approx q_\pi$

Policy improvement ϵ-greedy policy improvement

Windy Gridworld Example

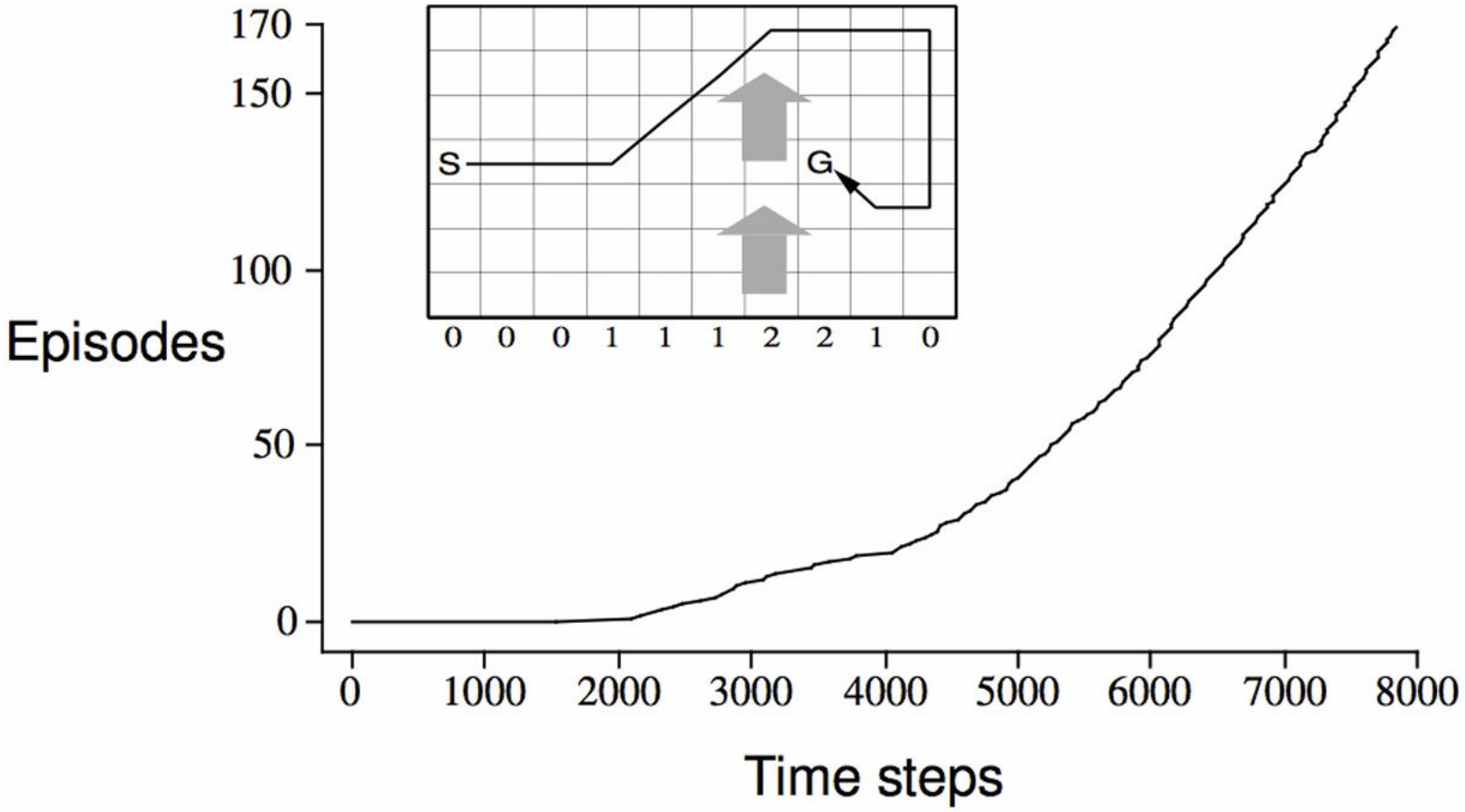






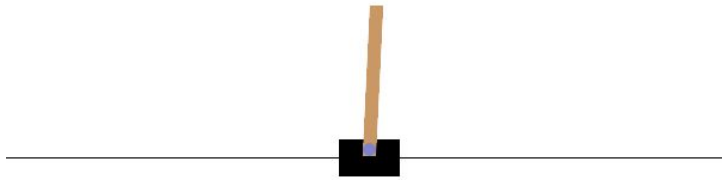
- Reward = -1 per time-step until reaching goal
- Undiscounted

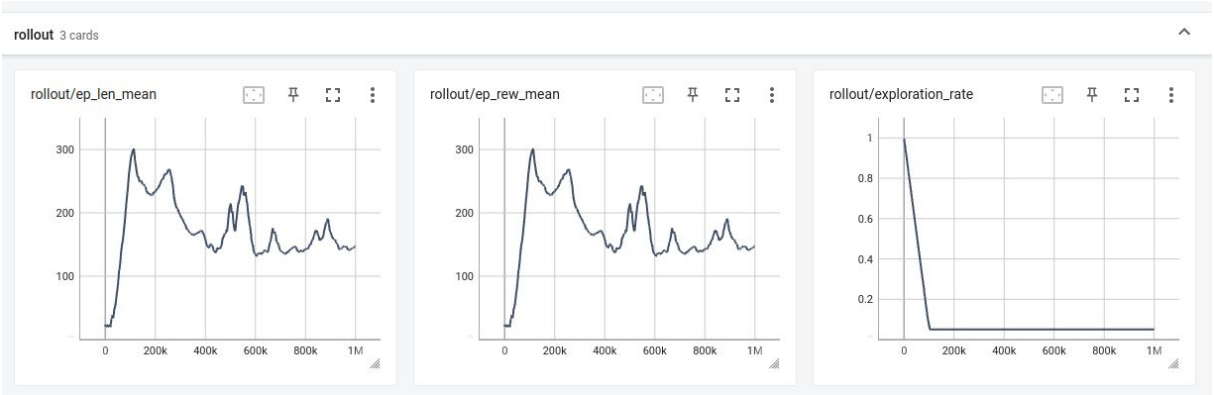
Sarsa on the Windy Gridworld



강화학습 환경과 알고리즘의 원리를 파악하기 위해 간단한 예제로 실습을 진행함.

1. Cartpole, MountainCar 예제 실습





2. Github에 실습 내용 업로드

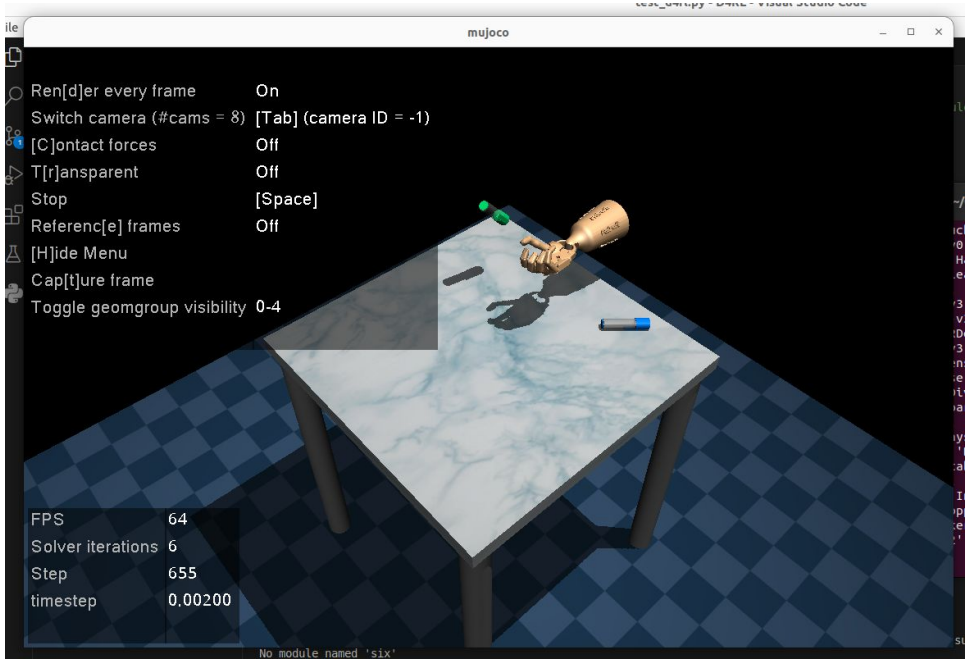
AI_Project_CoRLHF / RL_study / 6주차 / **실습** /

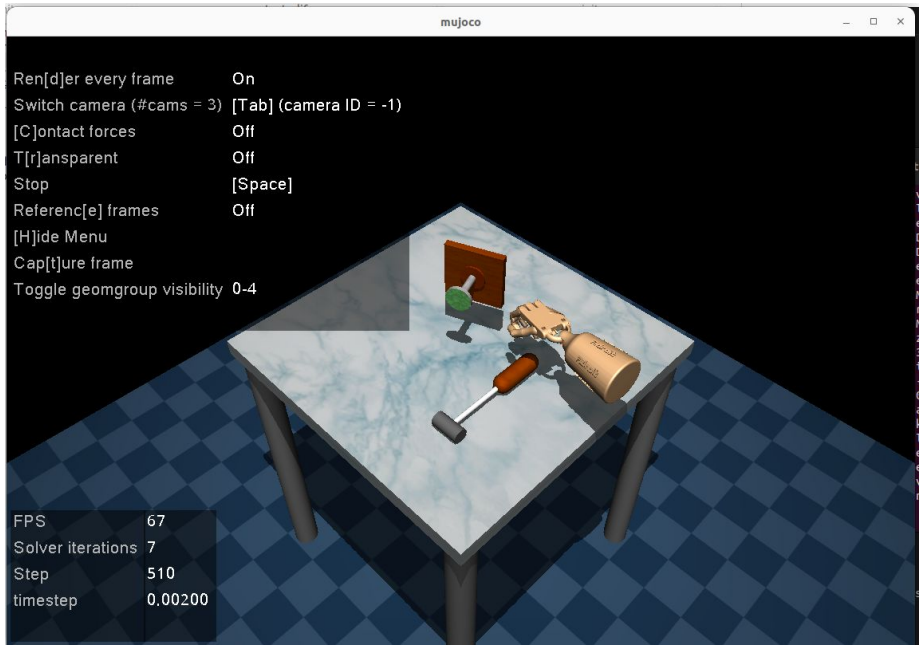
eunjuyummy Rename Pong_a2c_1e6.gif to CartPole_DQN_1

Name
..
CartPole.ipynb
CartPole.md
CartPole_DQN_1e6.gif

RLIF, RLHF 비교 실험을 진행할 가상 환경을 구축함.

1. Ardoit Pen (base), Ardoit Hammer (1차)





2. Github에 환경 구축 방법, 실행 코드 업로드

- Environments document.pdf
- environment_test.py
- register_environment.py

실증적AI프로젝트 금주 활동내역

주제: RLHF를 이용한 협동 로봇 제어 프로그램 개발

1.
2.
3.

	팀장 (권은주)	팀원 1 (조현진)	팀원 2 (진현석)
금주 개인별 활동내역	1. 5주차 개념정리, 스터디 2. 실험 환경 구축	1. 5주차 개념정리, 스터디 2. 강화학습 실습	1. 5주차 개념정리, 스터디 2. 강화학습 실습

차주 활동계획

1. RLIF 코드 분석 (4/16~4/19)

- train_rlif_main(학습 실행 파일), expert file, agent, model , utils 분석 후 코드 구조도 생성

2. Ardoit pen environment RLIF 알고리즘 적용 (4/20~4/21)

QUESTIONS & ANSWERS

Dept. of AI, Dong-A University

권은주 (kkkoj4284@donga.ac.kr)

진현석 (cpu132465@donga.ac.kr)

조현진 (gkfkgkdh@naver.com)

Github (https://github.com/eunjuyummy/AI_Project_CoRLHF)