Multi-arm Bandit
다중 선택

Hello!

I am Woojin Park

Creative Advocate Intern at Unity Technologies

강화학습 (RL)

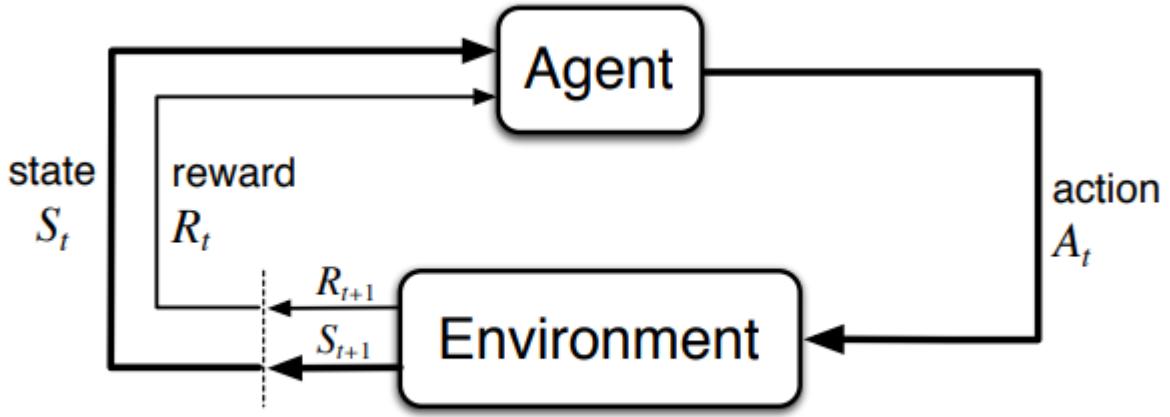


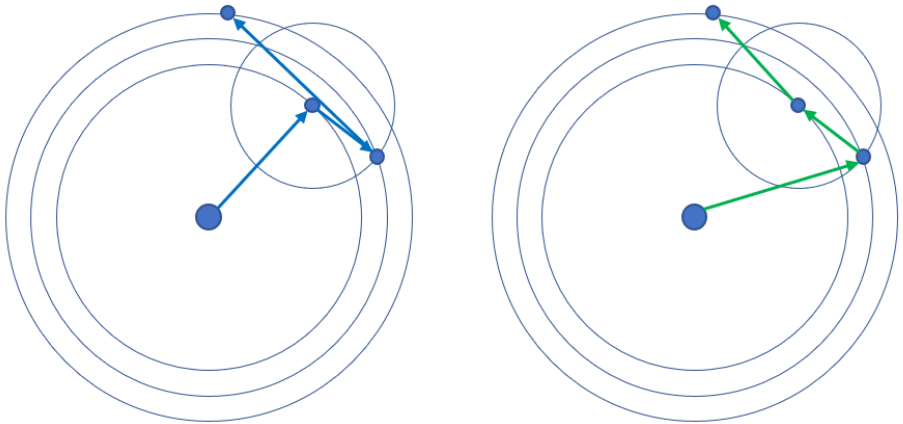
Figure 3.1: The agent–environment interaction in reinforcement learning.

Action-Value Method

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{N_t(a)}}{N_t(a)}. \tag{2.1}$$

$$A_t = \arg\max_a Q_t(a), \tag{2.2}$$

Path Planning

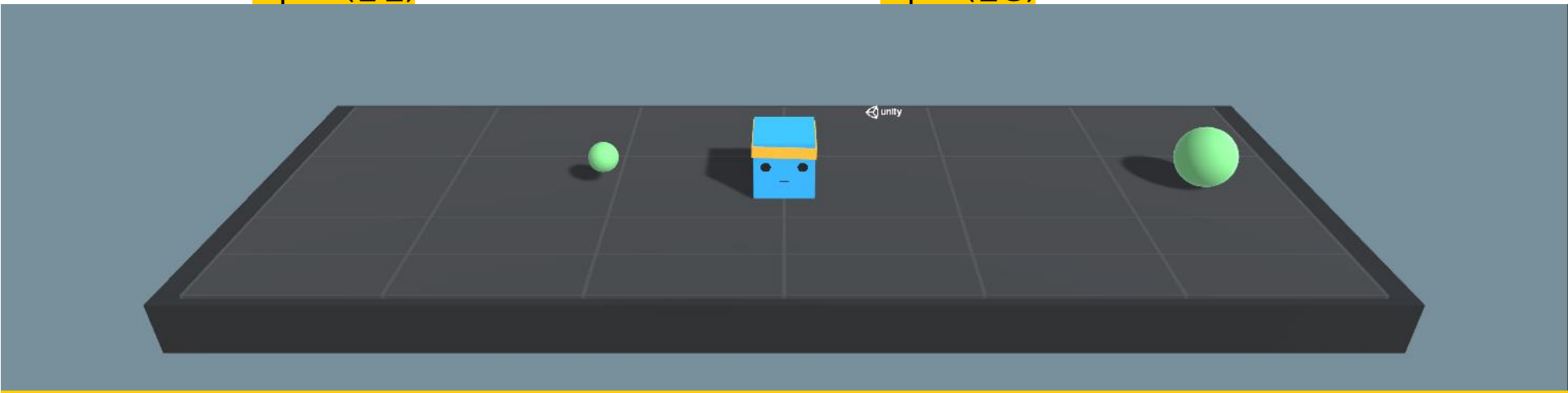


- *탐욕(근시안)적인 행동은 일반적으로 준최적(**Suboptimal**) 결과이다. 즉 최적(**Optimal**)의 결과를 보장할 수 없다.*

탐험과 활용

Explore (탐험)

Exploit (활용)

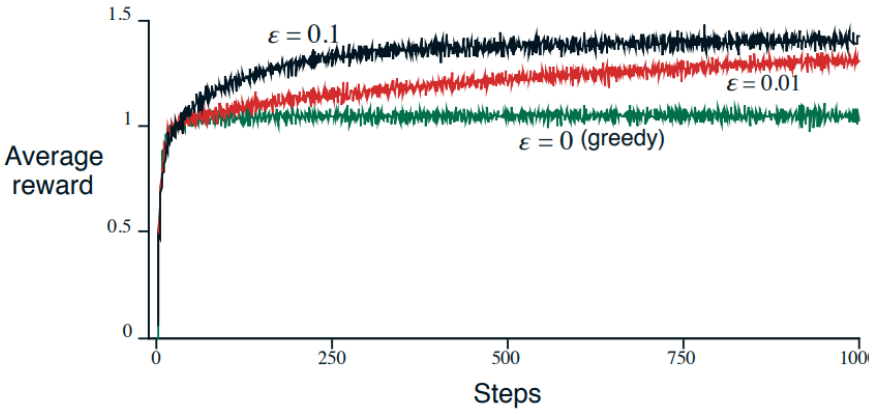


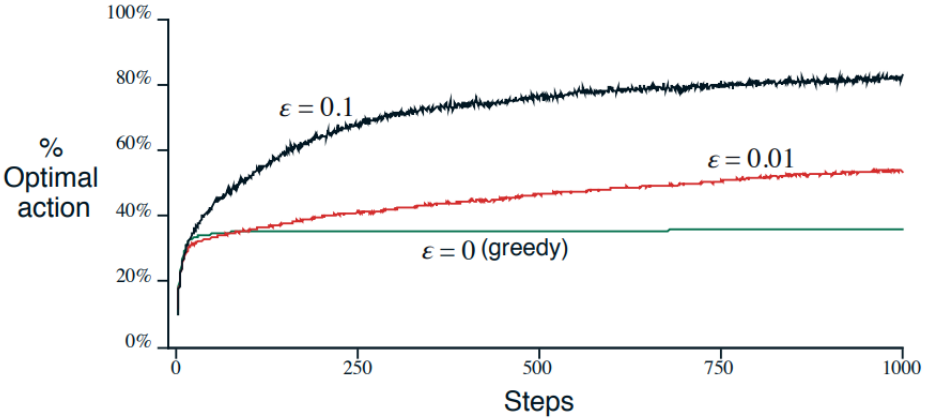
- *좌측 이동* +1, *우측 이동* -1
- *작은 공* +5, *큰 공* +100

Greedy $A_t = \arg\max_a Q_t(a),$

Epsilon-Greedy

Epsilon-Greedy (1-ε)





탐험과 활용의 필요성 (1)

Stationary (정상)

시간 t에 따른 보상 값의 확률 분포 변화 없음

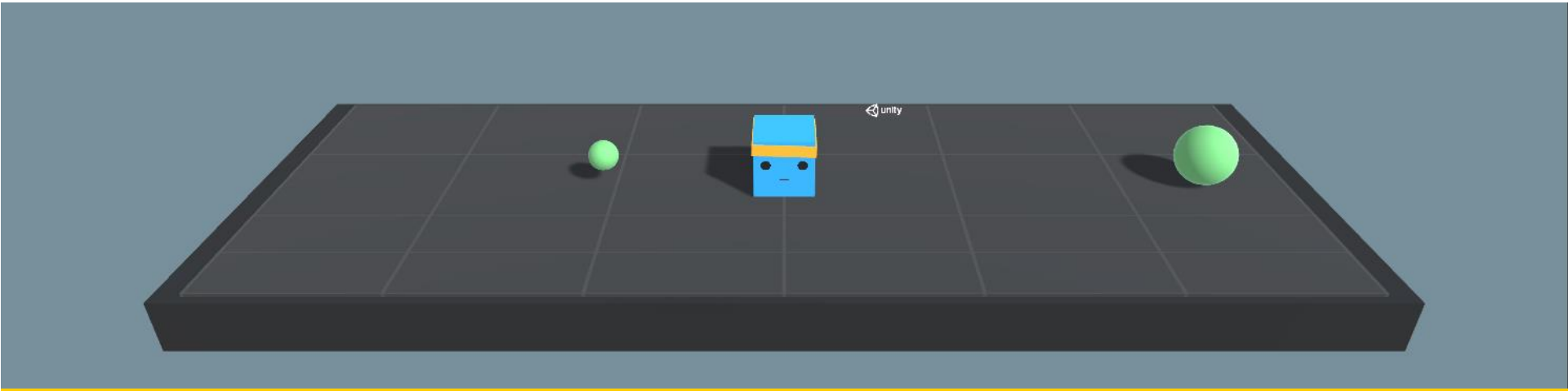
Nonstationary (비정상)

시간 t에 따른 보상 값의 확률 분포 변화 있음

실생활의 모든 문제!!!

- *좌측 이동* +1, *우측 이동* -1
- *작은 공* +5, *큰 공* +100

Stationary의 예시

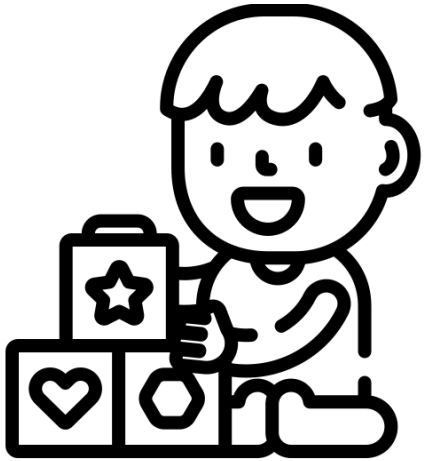


- *좌측 이동* +1, *우측 이동* -1
- *작은 공* +5, *큰 공* +100

NonStationary의 예시

오후 5시 : +10 점

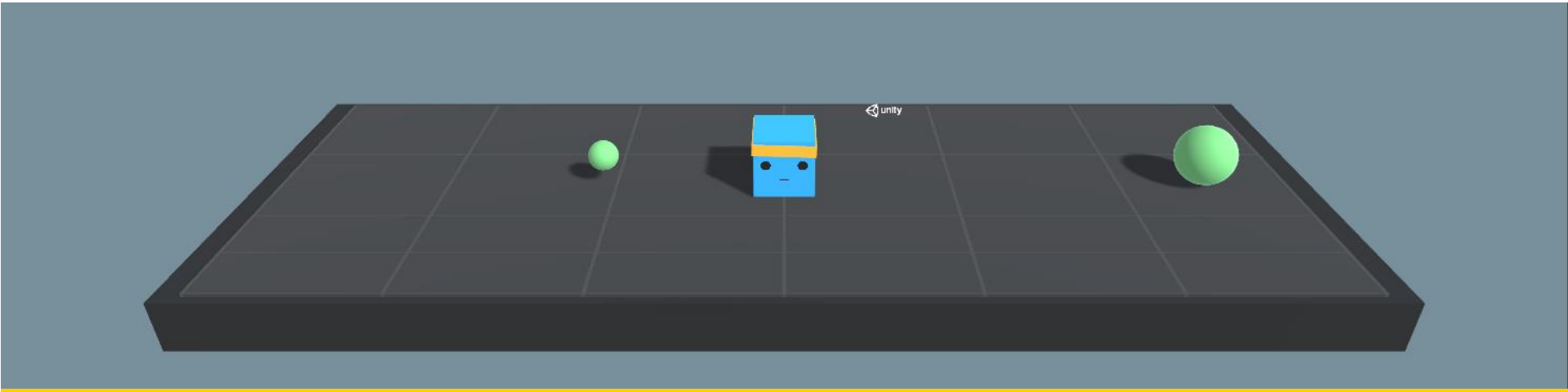
새벽 3시 : -10000 점



NonStationary의 예시

탐험과 활용의 필요성 (2)

Model Free 즉 MDP를 모르는 경우가 대다수…



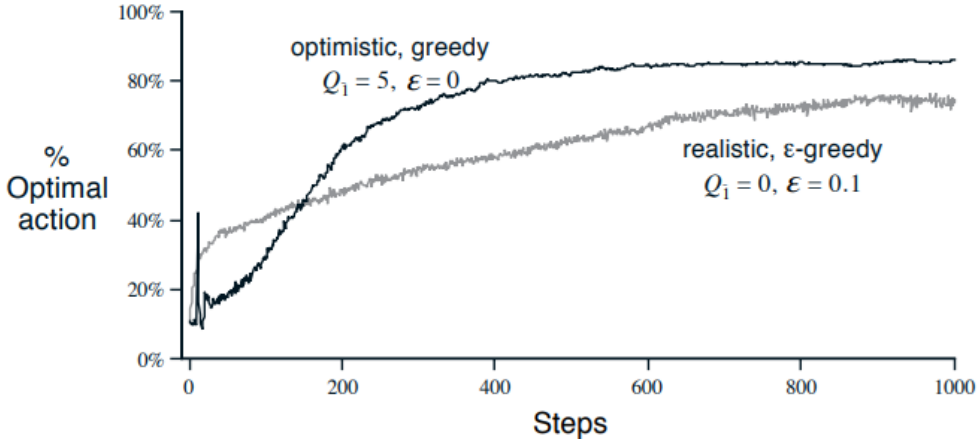
탐험과 활용의 필요성 (2)

- 구성: 보상을 받기 위해 에이전트는 왼쪽 또는 오른쪽으로만 이동할 수 있는 선형 이동 문제입니다.
- 목표: 보상을 가장 많이 받는 상태로 이동합니다.
- 에이전트: 하나의 에이전트가 존재합니다.
- 보상 함수:
  - -0.01 스텝마다 감점
  - +0.1 작은 공 위치에 도착할 때 (Suboptimal)
  - +1.0 큰 공 위치에 도착할 때 (Optimal)
- Behavior Parameters:
  - 벡터 관측 : 1
  - Actions: 이산적 행동 1 - 분기(Branch) 3개 (좌측 이동, 정지, 우측 이동)
  - 관측 : 없음
- Float Properties: 없음
- Benchmark Mean Reward: 0.93

Tracking a Nonstationary Problem

$$
\begin{aligned}
Q_{k+1} &= Q_k + \alpha\Big[R_k - Q_k\Big] \\
&= \alpha R_k + (1-\alpha)Q_k \\
&= \alpha R_k + (1-\alpha)\left[\alpha R_{k-1} + (1-\alpha)Q_{k-1}\right] \\
&= \alpha R_k + (1-\alpha)\alpha R_{k-1} + (1-\alpha)^2 Q_{k-1} \\
&= \alpha R_k + (1-\alpha)\alpha R_{k-1} + (1-\alpha)^2 \alpha R_{k-2} + \\
&\qquad \cdots + (1-\alpha)^{k-1}\alpha R_1 + (1-\alpha)^k Q_1 \\
&= (1-\alpha)^k Q_1 + \sum_{i=1}^{k} \alpha(1-\alpha)^{k-i} R_i. \qquad (2.6)
\end{aligned}
$$

Optimistic Initial Values



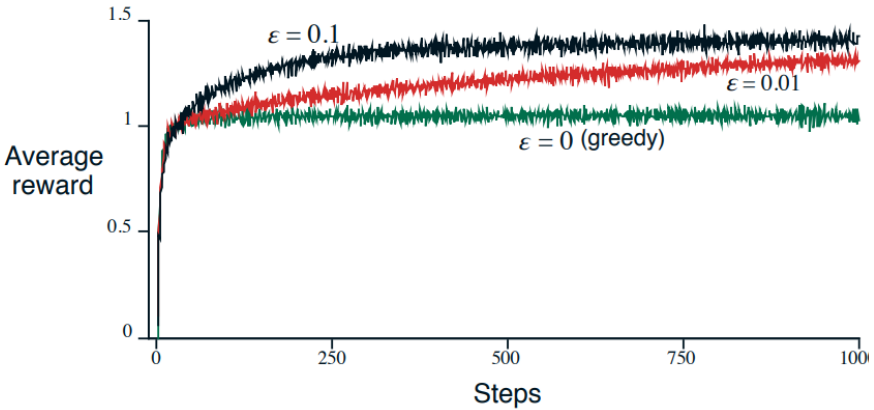
- *긍정적 초기값, 초반에 높은 q 값을 부여함으로써 탐험 유도*
- *시작할 때만 효과적, 정상 문제에만 효과적임*

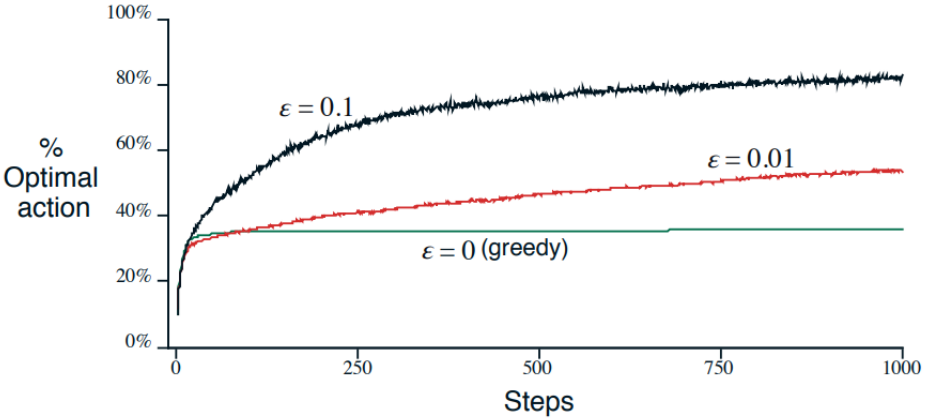
Upper Confidence Bound Action Selection

$$A_t = \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right], \tag{2.8}$$

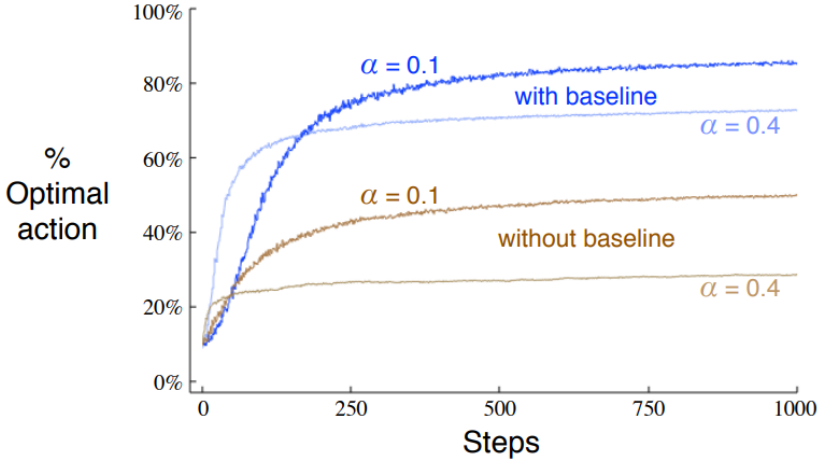
- **Epslion Greedy** *중 단순하게 비탐욕적인 행동이 아니라 최적행동의 참재력에 따라 비탐욕적 행동 선택*
- **lnt(***자연로그***), N(** *행동* **a** *의 횟수*), **c**는 *하이퍼 파라미터*
- *에피소드에 따라* **t**는 *계속 증가*, **Nt**는 *행동* **a** *에 따라 증가*

Epsilon-Greedy (1-ε)



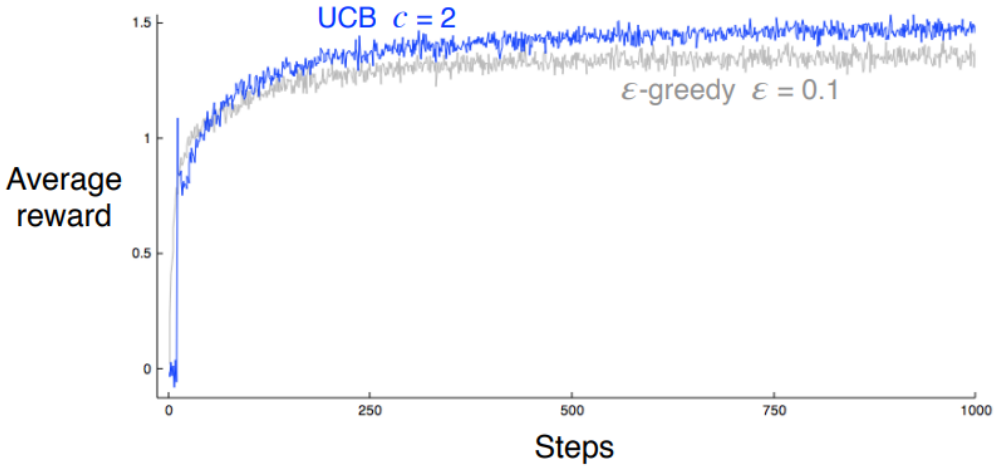


경사도 다중 선택 알고리즘



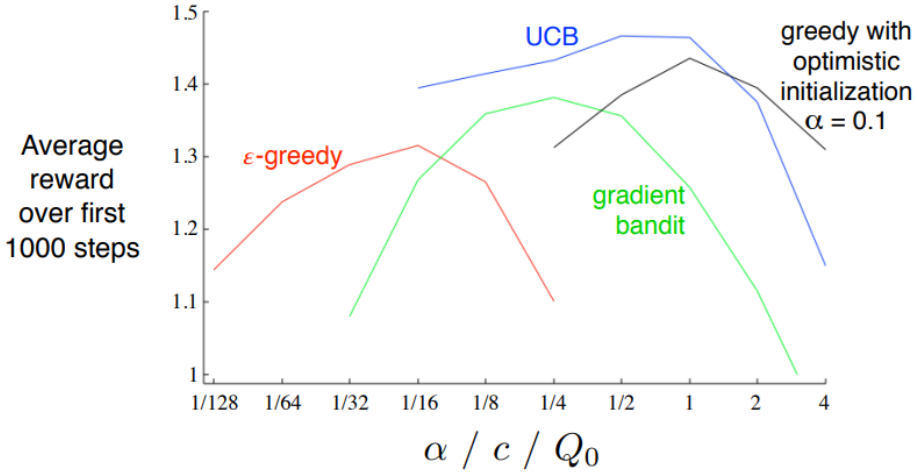
- *보상이 아닌 선호도를 추정하고 확률적으로 선호되는 행동 선택*

Upper Confidence Bound Action Selection



- ***State Size가 커질수록 더 어려워짐***

Summary


Average reward over first 1000 steps
1.5
1.4
1.3
1.2
1.1
1
UCB
greedy with optimistic initialization
α = 0.1
ε-greedy
gradient bandit
1/128
1/64
1/32
1/16
1/8
1/4
1/2
1
2
4
α / c / Q0

Thanks!

Any questions ?