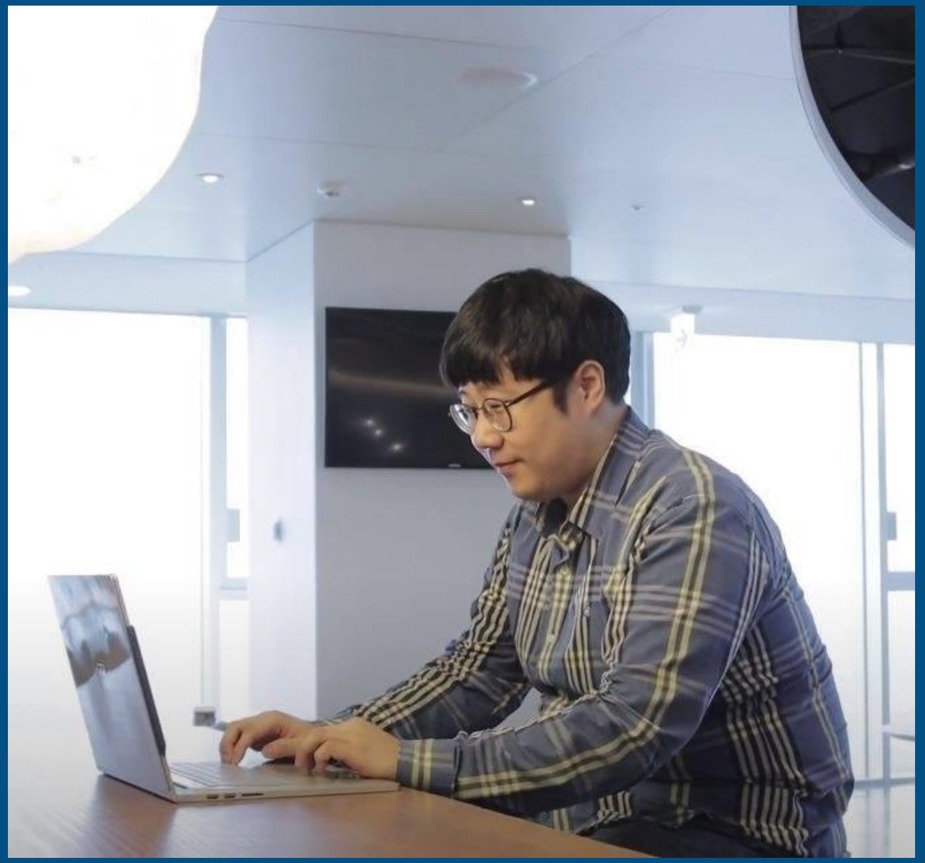
전북과학고등학교 특강

강화학습 기초

Chris Ohk
utilForever@gmail.com

강사 소개

- 옥찬호 (Chris Ohk)
  - 경북대학교 컴퓨터공학과 학사 (08)
  - KAIST 전산학부 석사 (13)
  - 넥슨 코리아 게임 프로그래머
  - Microsoft Developer Technologies MVP
  - C++ Korea, Reinforcement Learning KR 관리자
  - IT 전문서 집필 및 번역 다수
    - 게임샐러드로 코드 한 줄 없이 게임 만들기 (2013)
    - 유니티 Shader와 Effect 제작 (2014)
    - 2D 게임 프로그래밍 (2014), 러스트 핵심 노트 (2017)
    - 모던 C++ 입문 (2017), C++ 최적화 (2019)



교재

- 주 교재
  - 파이썬과 케라스로 배우는 강화학습 (위키북스, 2020)
- 부 교재
  - Reinforcement Learning, An Introduction - Second Edition (MIT Press, 2018)
    한국어 : 단단한 강화학습 (제이펍, 2020)
  - Deep Reinforcement Learning Hands-On - Second Edition (Packt, 2020)
  - 바닥부터 배우는 강화 학습 (영진닷컴, 2020)

스터디 진행 계획

- Day 1 (7/21)
  - What is Reinforcement Learning?
  - MDP (Markov Decision Process)
  - Value Function and Q-Function
- Day 2 (7/22)
  - Bellman Equation
  - Dynamic Programming
  - Exercise #1

스터디 진행 계획

- Day 3 (7/28)
  - Policy Evaluation
  - SARSA
  - Q-Learning
  - Exercise #2
- Day 4 (7/29)
  - Assignment #1

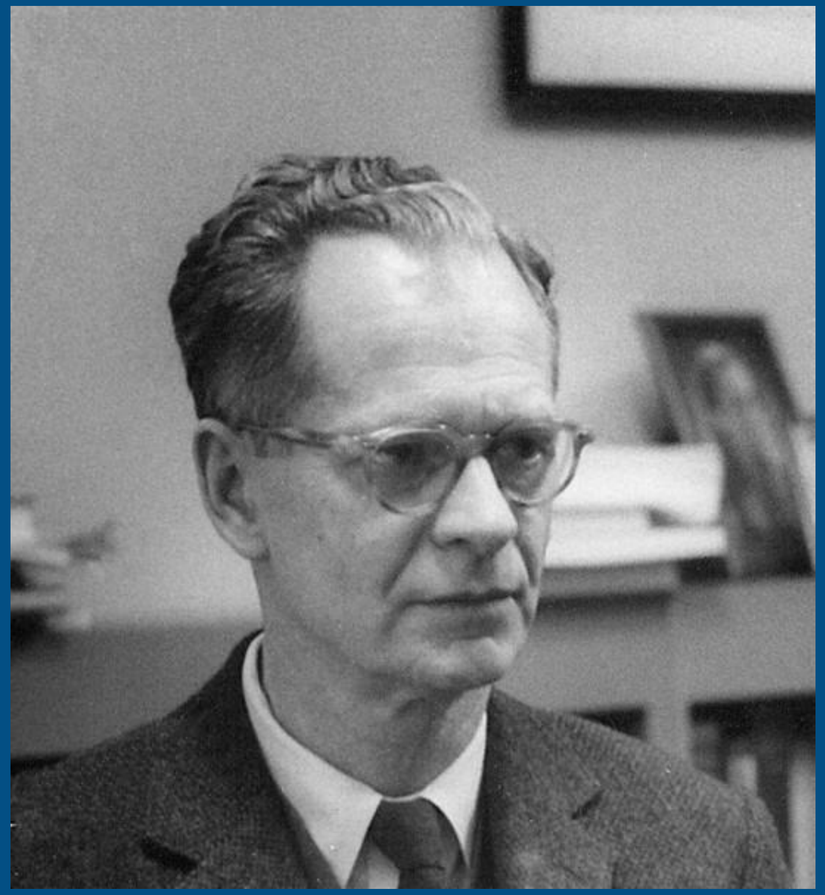
스터디 진행 계획

- Day 5 (8/2)
  - Deep Learning with PyTorch
- Day 6 (8/5)
  - Deep SARSA
  - Policy Gradient
  - Exercise #3

스터디 진행 계획

- Day 7 (8/9)
  - DQN (Deep Q-Network)
  - Exercise #4
- Day 8 (8/12)
  - Assignment #2

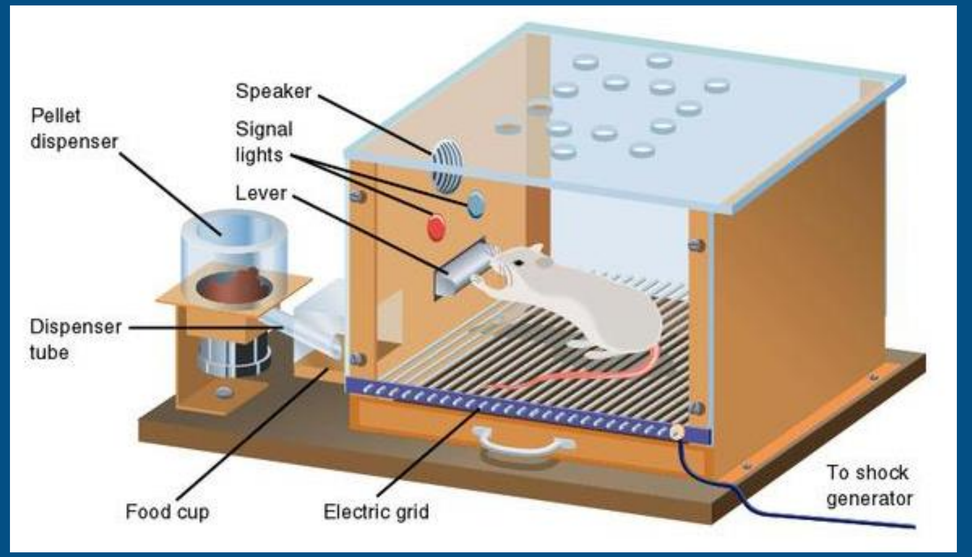
강화학습이란?



B. F. Skinner (1904~1990)

스키너의 강화 연구

1. 굶긴 쥐를 상자에 넣는다.
2. 쥐는 돌아다니다가 우연히 상자 안에 있는 지렛대를 누르게 된다.
3. 지렛대를 누르자 먹이가 나온다.
4. 지렛대를 누르는 행동과 먹이와의 상관관계를 모르는 쥐는 다시 돌아다닌다.
5. 그러다가 우연히 쥐가 다시 지렛대를 누르면 쥐는
   이제 먹이와 지렛대 사이의 관계를 알게 되고
   점점 지렛대를 자주 누르게 된다.
6. 이 과정을 반복하면서 쥐는 지렛대를 누르면
   먹이를 먹을 수 있다는 것을 학습한다.



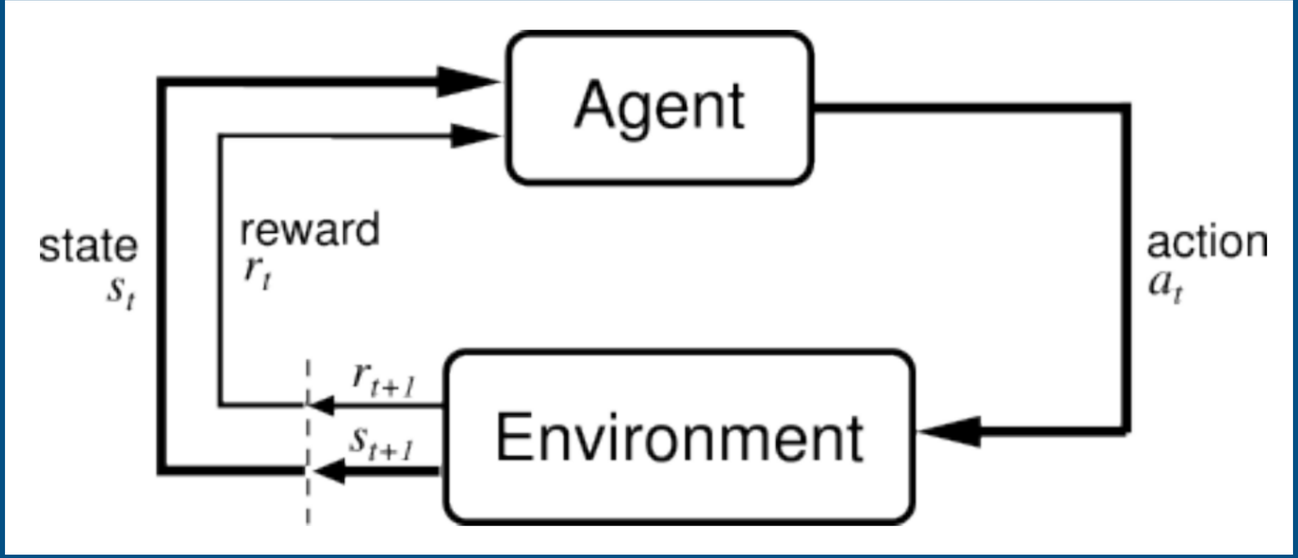
우리 주변에서의 강화

아이가 첫걸음을 떼는 과정도 일종의 강화라고 할 수 있다.

1. 아이는 걷는 것을 배운 적이 없다.
2. 아이는 스스로 이것저것 시도해 보다가 우연히 걷게 된다.
3. 자신이 하는 행동과 걷게 된다는 보상 사이의 상관관계를 모르는 아이는 다시 넘어진다.
4. 시간이 지남에 따라 그 관계를 학습해서 잘 걷게 된다.

강화학습

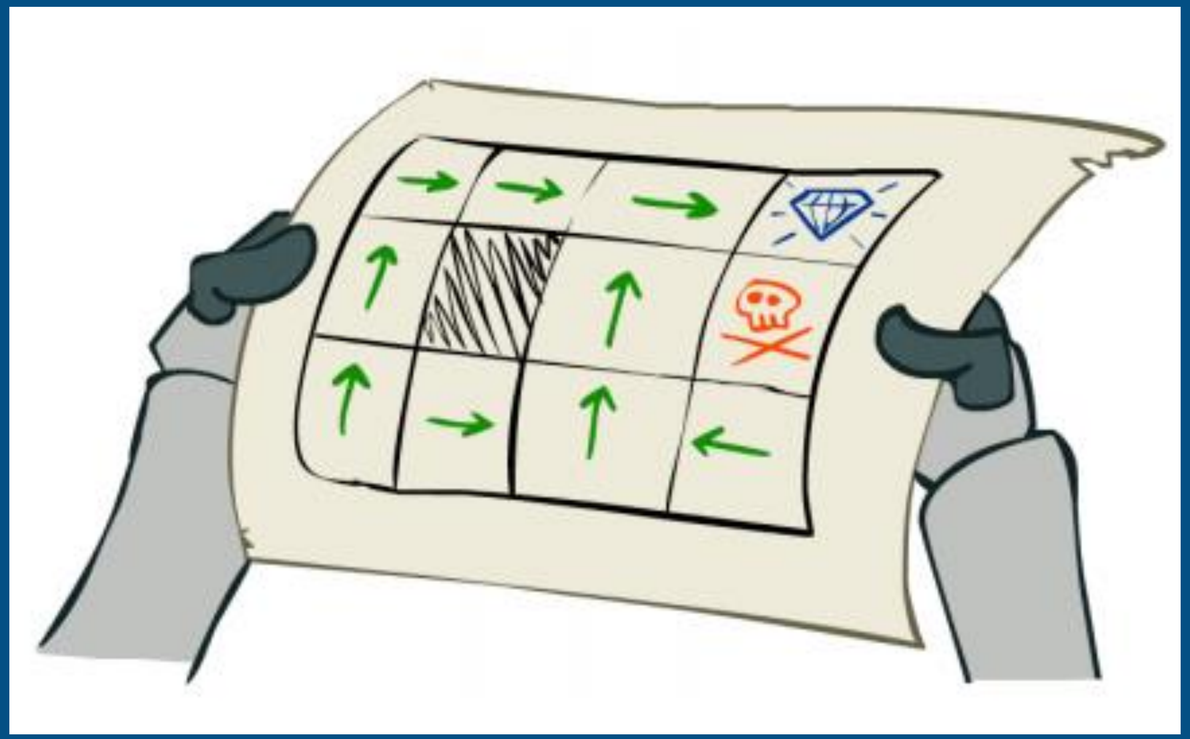
- 에이전트는 사전 지식이 없는 상태에서 학습함
- 에이전트는 자신이 놓인 환경에서 자신의 상태를 인식한 후 행동
- 환경은 에이전트에게 보상을 주고 다음 상태를 알려줌
- 에이전트는 보상을 통해 어떤 행동이 좋은 행동인지 간접적으로 알게 됨



강화학습 문제

결정을 순차적으로 내려야 하는 문제에 강화학습을 적용한다.

이 문제를 풀기 위해서는 문제를 수학적으로 정의해야 한다.



강화학습 문제

수학적으로 정의된 문제는 다음과 같은 구성 요소를 가진다.

1. 상태 (State)
   현재 에이전트의 정보 (정적인 요소 + 동적인 요소)
2. 행동 (Action)
   에이전트가 어떠한 상태에서 취할 수 있는 행동
3. 보상 (Reward)
   에이전트가 학습할 수 있는 유일한 정보, 자신이 했던 행동을 평가할 수 있는 지표
   강화학습의 목표는 시간에 따라 얻는 보상의 합을 최대로 하는 정책을 찾는 것
4. 정책 (Policy)
   순차적 행동 결정 문제에서 구해야 할 답
   모든 상태에 대해 에이전트가 어떤 행동을 해야 하는지 정해놓은 것

MDP

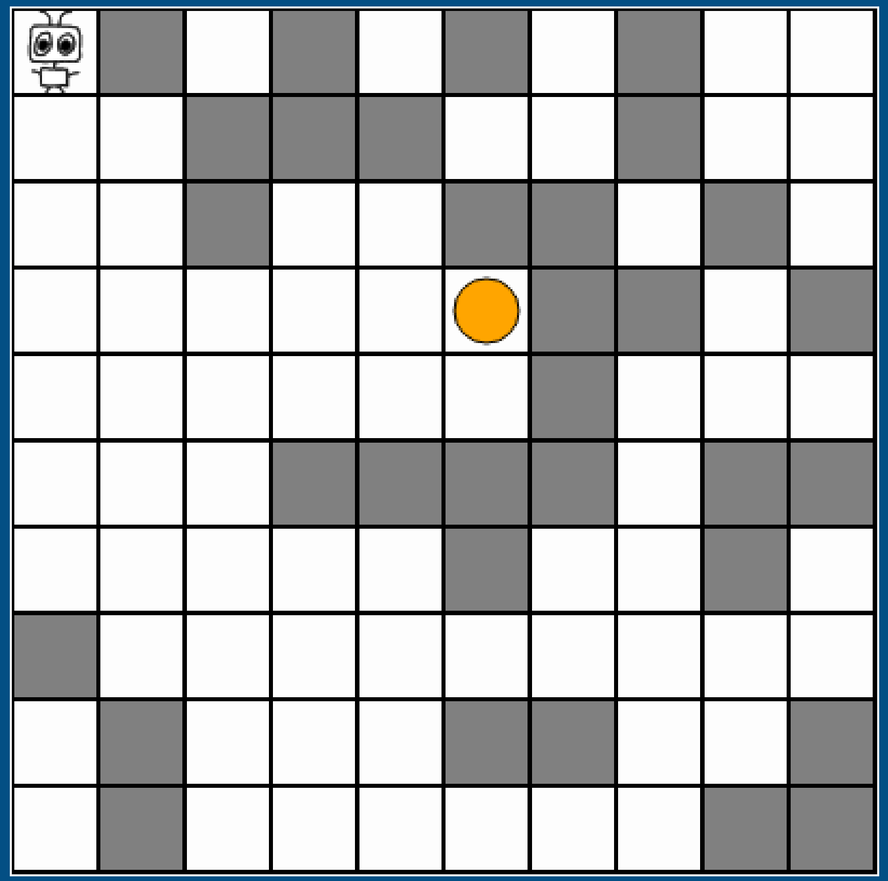
강화 학습은 순차적으로 행동을 계속 결정해야 하는 문제를 푸는 것

→ 이 문제를 수학적으로 표현한 것이 MDP(Markov Decision Process)

- MDP의 구성 요소
  - 상태
  - 행동
  - 보상 함수
  - 상태 변환 확률 (State Transition Probability)
  - 감가율 (Discount Factor)

그리드 월드

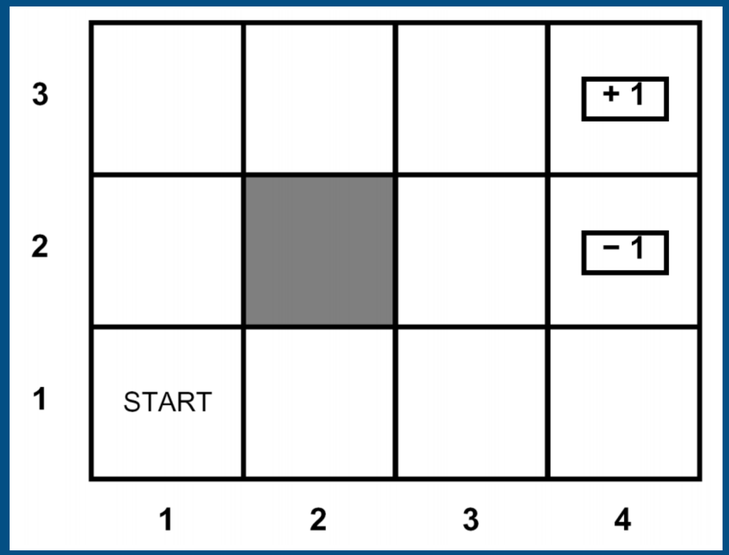
격자로 이뤄진 환경에서 문제를 푸는 각종 예제



상태

에이전트가 관찰 가능한 상태의 집합 : S

- 그리드 월드에서 상태의 개수는 유한
- 그리드 월드에 상태가 5개 있을 경우, 수식으로 표현하면
  $S = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5)\}$
- 그리드 월드에서 상태는 격자 상의 각 위치(좌표)
- 에이전트는 시간에 따라 상태 집합 안에 있는 상태를 탐험한다.
  이 때 시간을 t, 시간 t일 때의 상태를 S_t라고 표현한다.
- 예를 들어, 시간이 t일 때 상태가 $(1, 3)$이라면 $S_t = (1, 3)$



상태

에이전트가 관찰 가능한 상태의 집합 : $\boldsymbol{S}$

- 어떤 t에서의 상태 S_t는 정해진 것이 아니다.
- 때에 따라서 $t = 1$일 때 $S_t = (1, 3)$일 수도 있고 $S_t = (4, 2)$일 수도 있다.

"상태 = 확률 변수(Random Variable)"

상태

$$S_t = s$$

"시간 t에서의 상태 S_t가 어떤 상태 s다."

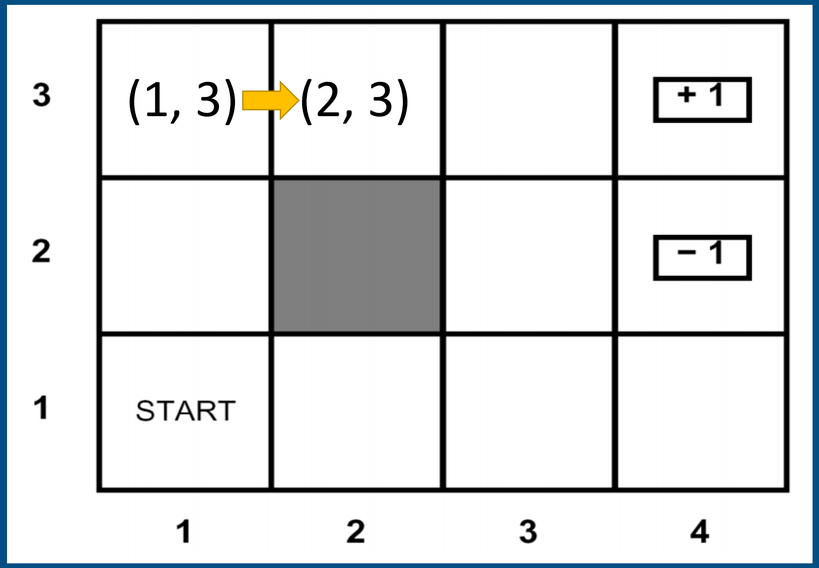
행동

에이전트가 상태 S_t에서 할 수 있는 가능한 행동의 집합 : A

- 보통 에이전트가 할 수 있는 행동은 모든 상태에서 같다.

$$A_t = a$$

- "시간 t에 에이전트가 특정한 행동 a를 했다."
- t라는 시간에 에이전트가 어떤 행동을 할 지는 정해져 있지 않으므로 A_t처럼 대문자로 표현한다.
- 그리드 월드에서 에이전트가 할 수 있는 행동은 $A = \{\text{up}, \text{down}, \text{left}, \text{right}\}$
- 만약 시간 t에서 상태가 (1, 3)이고 $A_t = \text{right}$라면 다음 시간의 상태는 (2, 3)이 된다.



보상 함수

에이전트가 학습할 수 있는 유일한 정보

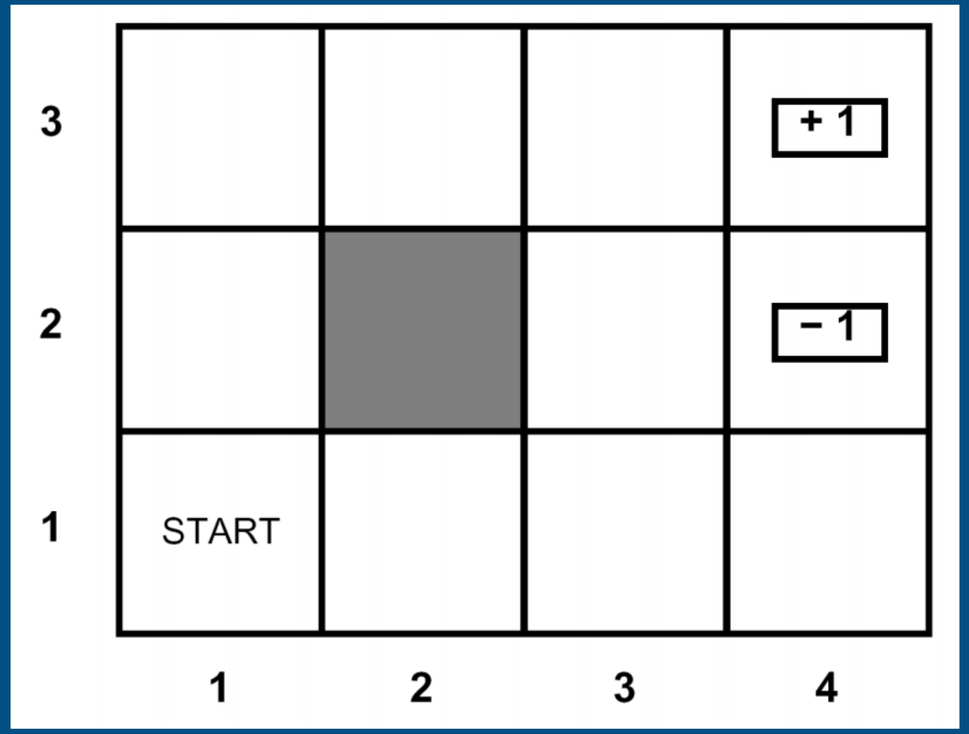
- 보상 함수 (Reward Function)

$$R_s^a = \mathrm{E}[R_{t+1}|S_t = s, A_t = a]$$

- 시간 t일 때 상태가 $S_t = s$이고 그 상태에서 행동이 $A_t = a$를 했을 경우 받을 보상에 대한 기댓값(Expectation) E
- 에이전트가 어떤 상태에서 행동한 시간 : t
  보상을 받는 시간 : $t + 1$
- 이유 : 에이전트가 보상을 알고 있는게 아니라 환경이 알려주기 때문
  에이전트가 상태 s에서 행동 a를 하면 환경은 에이전트가 가게 되는 다음 상태 s'와 에이전트가 받을 보상을 에이전트에게 알려준다. 이 시점이 $t + 1$이다.

보상 함수

에이전트가 학습할 수 있는 유일한 정보



상태 변환 확률

에이전트가 어떤 상태에서 어떤 행동을 취하면 상태가 변한다.

하지만 어떤 이유로 인해 다음 상태로 변하지 못할 수도 있다.

→ 상태의 변화에는 확률적인 요인이 들어간다.

이를 수치적으로 표현한 것이 상태 변환 확률!

$$P_{ss'}^{a} = \mathrm{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

감가율

에이전트는 항상 현재 시점에서 판단을 내리기 때문에
현재에 가까운 보상일수록 더 큰 가치를 갖는다.

보상의 크기가 100일 때, 에이전트가 현재 시각에 보상을 받을 때는
100의 크기 그대로 받아들이지만 현재로부터 일정 시간이 지나서
보상을 받으면 크기가 100이라고 생각하지 않는다.

에이전트는 그 보상을 얼마나 시간이 지나서 받는지를 고려해
감가시켜 현재의 가치로 따진다.

감가율

감가율

A 은행

“당첨금 1억 원을 지금 당장 드리겠습니다.”

vs

B 은행

“지금 당장 받으면 막쓰다가 탕진할 가능성이 크니,
10년 후에 당첨금 1억 원을 드리겠습니다.”

감가율

A 은행
“당첨금 1억 원을 지금 당장 드리겠습니다.”

≠

B 은행
“지금 당장 받으면 막쓰다가 탕진할 가능성이 크니,
10년 후에 당첨금 1억 원을 드리겠습니다.”

감가율

A 은행

“당첨금 1억 원을 지금 당장 드리겠습니다.”

≒

B 은행

“지금 당장 받으면 막쓰다가 탕진할 가능성이 크니,
10년 후에 당첨금 1억 원에 이자까지 드리겠습니다.”

감가율

우리는 이자를 통해 나중에 받을 보상에
추가적인 보상을 더해 현재의 보상과 같게 만든다.

→ 반대로 말하면 같은 보상이면 나중에 받을수록 가치가 줄어든다.
이를 수학적으로 표현한 개념이 “감가율(Discount Factor)”

감가율

감가율 : 시간에 따라서 감가하는 비율

$$\gamma \in [0, 1]$$

현재의 시간 t로부터 시간 k가 지난 후에 받는 보상이 R_{t+k}라면

현재 그 보상의 가치는 $\gamma^{k-1} R_{t+k}$와 같다.

즉, 더 먼 미래에 받을 수록 에이전트가 받는 보상의 크기는 줄어든다.

정책

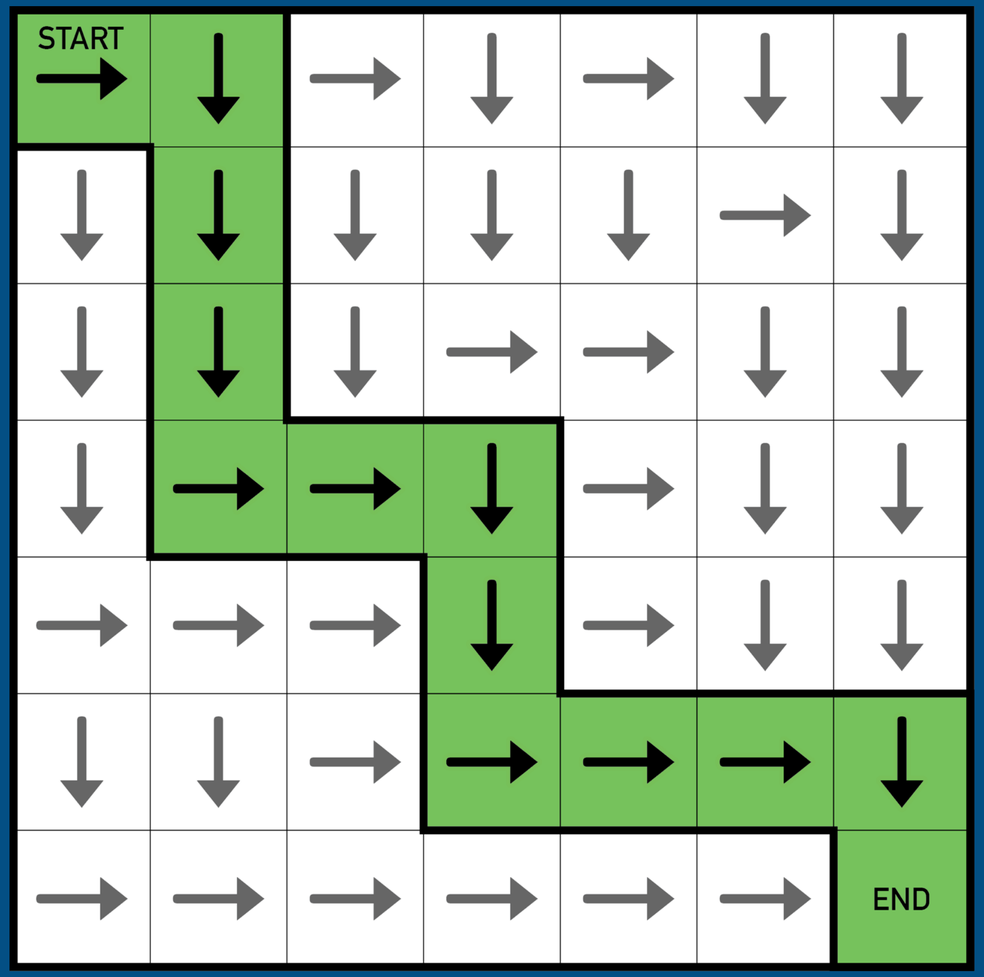
모든 상태에서 에이전트가 할 행동

- 상태가 입력으로 들어오면 행동을 출력으로 내보내는 일종의 함수
- 하나의 행동만을 나타낼 수도 있고, 확률적으로 $a_1 = 10\%$, $a_2 = 90\%$로 나타낼 수도 있다.

$$\pi(a|s) = \mathrm{P}[A_t = a|S_t = s]$$

- 시간 t에 에이전트가 $S_t = s$에 있을 때 가능한 행동 중에서 $A_t = a$를 할 확률
- 강화 학습 문제를 통해 알고 싶은 것은 정책이 아닌 “최적 정책”

정책



가치함수

우리가 지금까지 한 일 : 문제를 MDP로 정의

→ 에이전트는 MDP를 통해 최적 정책을 찾으면 된다.

하지만 에이전트가 어떻게 최적 정책을 찾을 수 있을까?

가치함수

에이전트 입장에서 어떤 행동을 하는 것이 좋은지를 어떻게 알 수 있을까?

→ 현재 상태에서 앞으로 받을 보상을 고려해서 선택해야 좋은 선택!

하지만 아직 받지 않은 보상들을 어떻게 고려할 수 있을까?

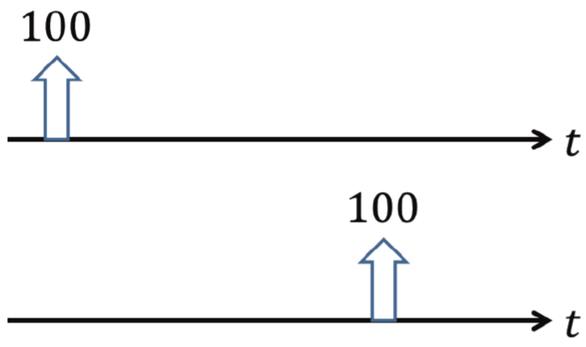
→ 에이전트는 가치함수를 통해 행동을 선택할 수 있다.

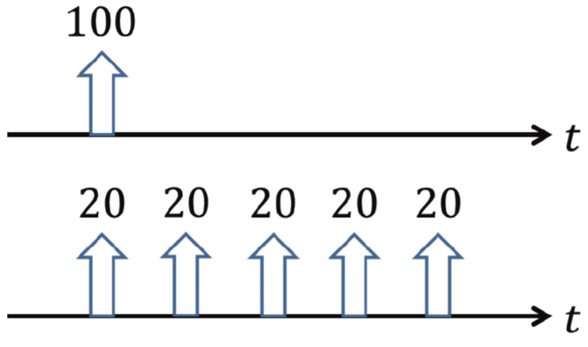
가치함수

현재 시간 t부터 에이전트가 행동을 하면서 받을 보상을 모두 더해보자.

$$R_{t+1} + R_{t+2} + R_{t+3} + R_{t+4} + R_{t+5} + \cdots$$

하지만 이 수식에는 3가지 문제가 있다.



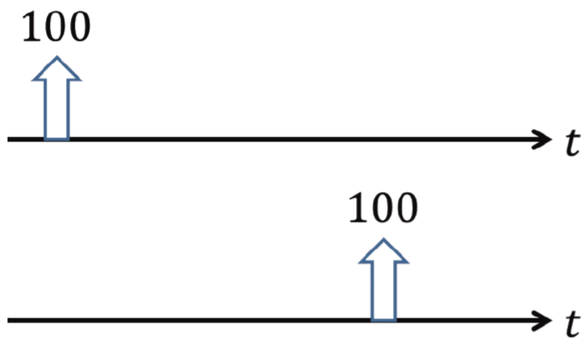


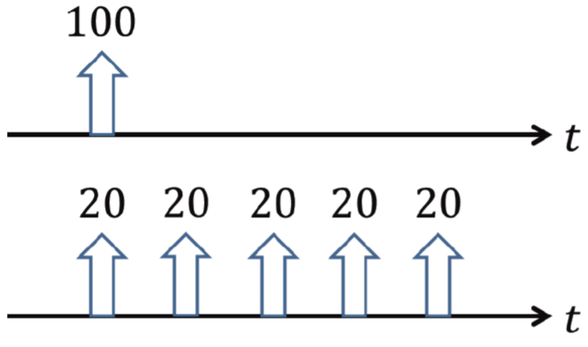
$$0.1 + 0.1 + \cdots = \infty$$

$$1 + 1 + \cdots = \infty$$

가치함수

1. 현재에 받은 보상 100과 미래에 받는 보상 100을 똑같이 취급한다.
2. 보상 100을 1번 받을 때와 보상 20을 5번 받을 때를 구분하지 못한다.
3. 시간이 무한대라면 0.1을 받아도, 1을 받아도 합이 무한대다.





$$0.1 + 0.1 + \cdots = \infty$$
$$1 + 1 + \cdots = \infty$$

가치함수

단순한 보상의 합으로는 판단을 내리기 어려우므로,
좀 더 정확한 판단을 위해 감가율을 고려한다.

$$R_{t+1} + R_{t+2} + R_{t+3} + R_{t+4} + R_{t+5} + \cdots$$

$$R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \gamma^4 R_{t+5} + \cdots$$

가치함수

이 값을 반환값(Return) G_t라고 한다.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

예를 들어, 에피소드를 $t = 1$부터 5까지 진행했다면

$$G_1 = R_2 + \gamma R_3 + \gamma^2 R_4 + \gamma^3 R_5 + \gamma^4 R_6$$

$$G_2 = R_3 + \gamma R_4 + \gamma^2 R_5 + \gamma^3 R_6$$

$$G_3 = R_4 + \gamma R_5 + \gamma^2 R_6$$

$$G_4 = R_5 + \gamma R_6$$

$$G_5 = R_6$$

가치함수

에이전트는 에피소드가 끝난 후에야 반환값을 알 수 있다.
하지만 에피소드가 끝날 때까지 기다려야 할까?

때로는 정확한 값을 얻기 위해 끝까지 기다리는 것보다
정확하지 않더라도 현재의 정보를 토대로 행동하는 것이 나을 때가 있다.

→ 가치함수 = 어떠한 상태에 가면 받을 것이라고 예상되는 값

$$v(s) = E[G_t \mid S_t = s]$$

가치함수

$$v(s) = E[G_t \mid S_t = s]$$

$$v(s) = E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid S_t = s]$$

$$v(s) = E[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \cdots) \mid S_t = s]$$

$$v(s) = E[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

가치함수

- 반환값으로 나타내는 가치함수

$$v(s) = E[R_{t+1} + \gamma G_{t+1} | S_t = s]$$

- 가치함수로 표현하는 가치함수의 정의

$$v(s) = E[R_{t+1} + \gamma v(S_{t+1}) | S_t = s]$$

여기까지는 가치함수를 정의할 때 정책을 고려하지 않음

가치함수

하지만 정책을 고려하지 않으면 안된다.

- 상태에서 상태로 넘어갈 때 에이전트는 무조건 행동을 해야 하고 각 상태에서 행동을 하는 것이 에이전트의 정책이기 때문
- 정책에 따라서 계산하는 가치함수는 당연히 달라질 수밖에 없음
- MDP에서의 가치함수는 항상 정책을 고려

가치함수

벨만 기대 방정식(Bellman Expectation Equation)

$$v_\pi(s) = E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$$

- 현재 상태의 가치 함수와 다음 상태의 가치함수 사이의 관계를 말해주는 방정식
- 강화학습은 벨만 방정식을 어떻게 풀어나가느냐의 스토리

큐함수

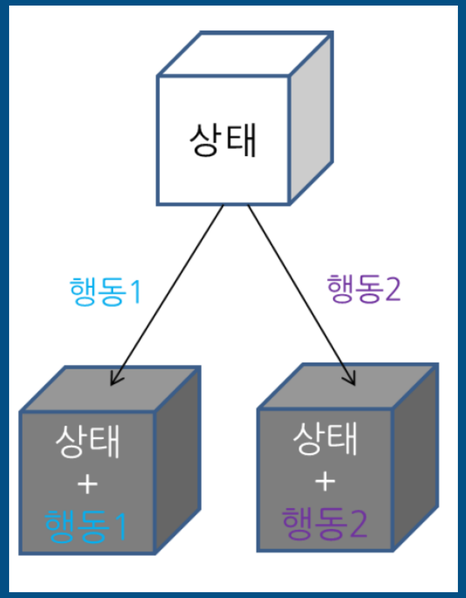
가치함수는 말 그대로 “함수” → 입력/출력이 무엇인지 알아야 한다.

- 지금까지 설명한 가치함수는 상태 가치함수
- 에이전트는 가치함수를 통해 다음에 어떤 상태로 가야할 지 판단할 수 있다.
- 어떤 상태로 가면 좋을지 판단한 후에 그 상태로 가기 위한 행동을 따져볼 것이다.

큐함수

하지만…

- 어떤 상태에서 각 행동에 대해 따로 가치함수를 만들어서 그 정보를 얻어올 수 있다면 에이전트는 굳이 다음 상태의 가치 함수를 따져보지 않아도 어떤 행동을 해야할 지 선택할 수 있다.
- 이처럼 어떤 상태에서 어떤 행동이 얼마나 좋은지 알려주는 함수를 행동 가치함수라고 한다.
  → 큐함수(Q Function)!



큐함수

가치함수와 큐함수 사이의 관계

$$v_\pi(s) = \sum_{a \in A} \pi(a \mid s) q_\pi(s, a)$$

1. 각 행동을 했을 때 앞으로 받을 보상인 큐함수 $q_\pi(s, a)$를 $\pi(a \mid s)$에 곱한다.
2. 모든 행동에 대해 큐함수와 정책을 곱한 값을 더하면 가치함수가 된다.

큐함수는 강화학습에서 중요한 역할을 한다.

- 강화학습에서 에이전트가 행동을 선택하는 기준으로
  가치함수보다는 보통 큐함수를 사용한다.
- 그 이유는 뒤에서 설명할 예정

큐함수

큐함수 또한 벨만 기대 방정식의 형태로 나타낼 수 있다.

$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | \, S_t = s, A_t = a]$$

- 가치함수의 식과 다른 점은 조건문에 행동이 더 들어간다는 점

감사합니다!

스터디 듣느라 고생 많았습니다.