강화학습

KAIG 세미나 (2019-03-19)

김수환

강화학습의 예시

강화학습 -> DeepMind -> **AlphaGo**

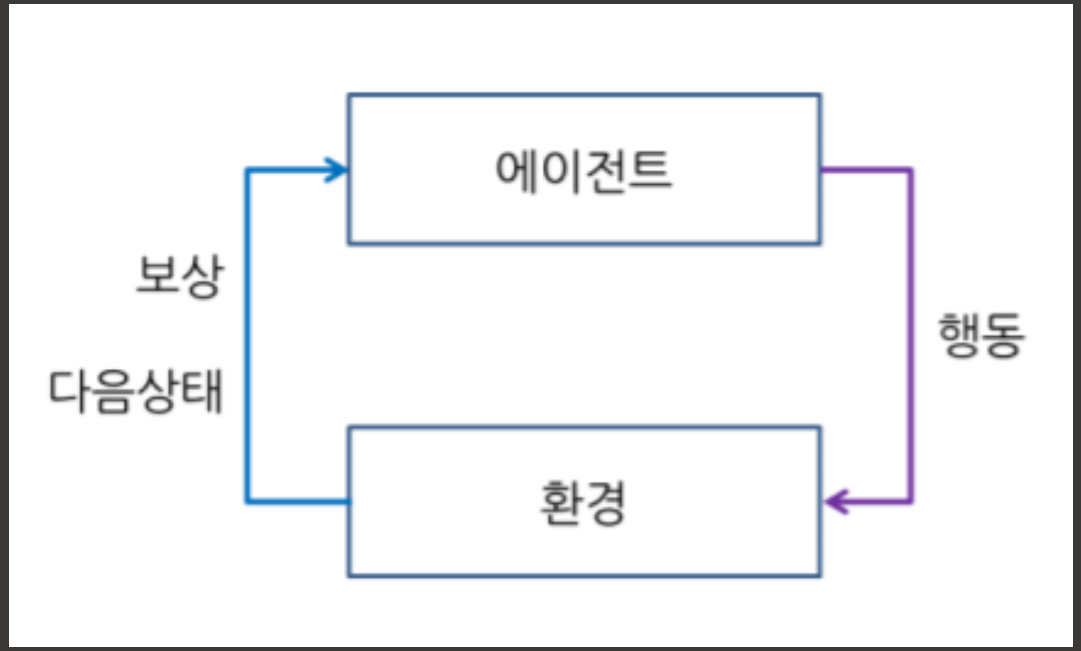
강화학습의 예시

DeepMind -> **Atari** -> **DQN**

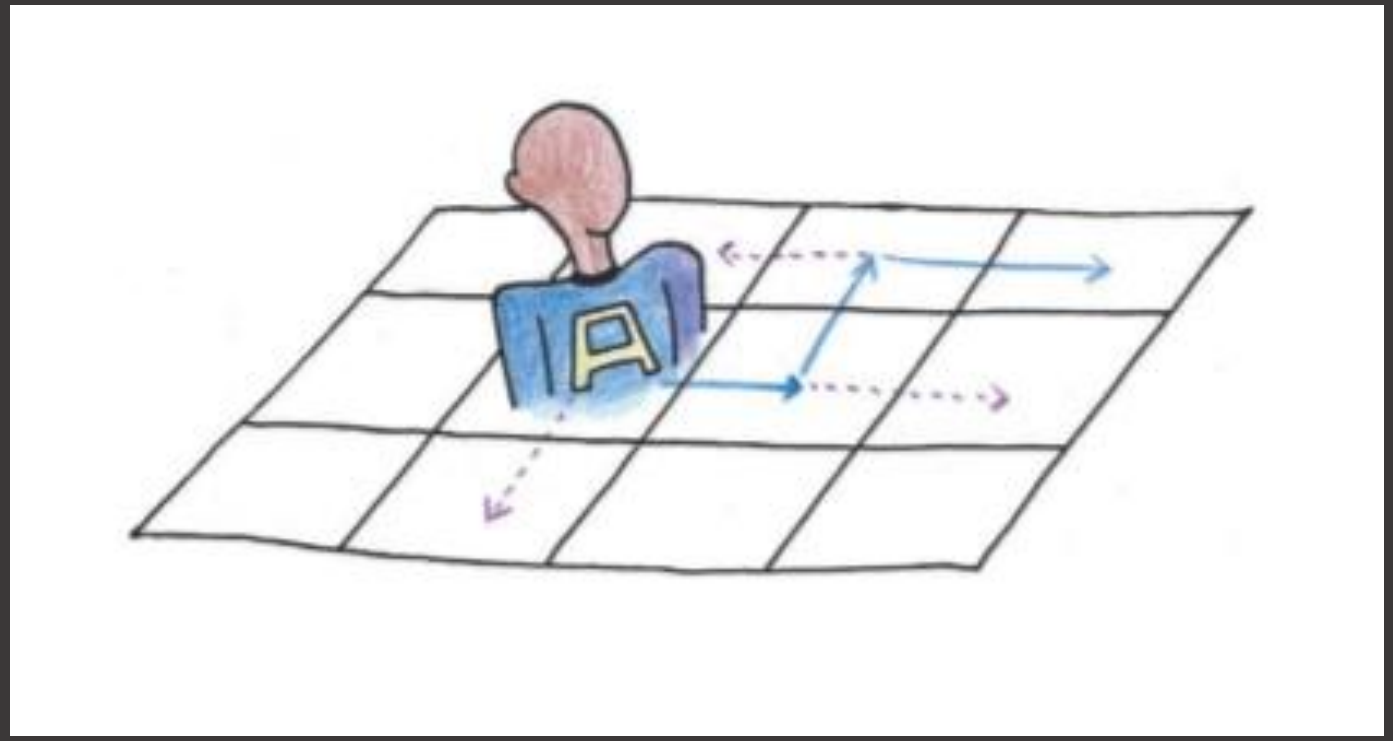
게임 화면으로 게임 플레이하는 법을 학습

학습을 위해 필요한 것

- **핵심은 에이전트와 환경의 상호작용**



에이전트 -> 상태를 관찰, 행동을 선택, 목표지향



환경 -> 에이전트를 제외한 나머지

판단하는 아이라는 주체를 빼고 길과 자전거와 아이의 몸 또한 환경이 된다

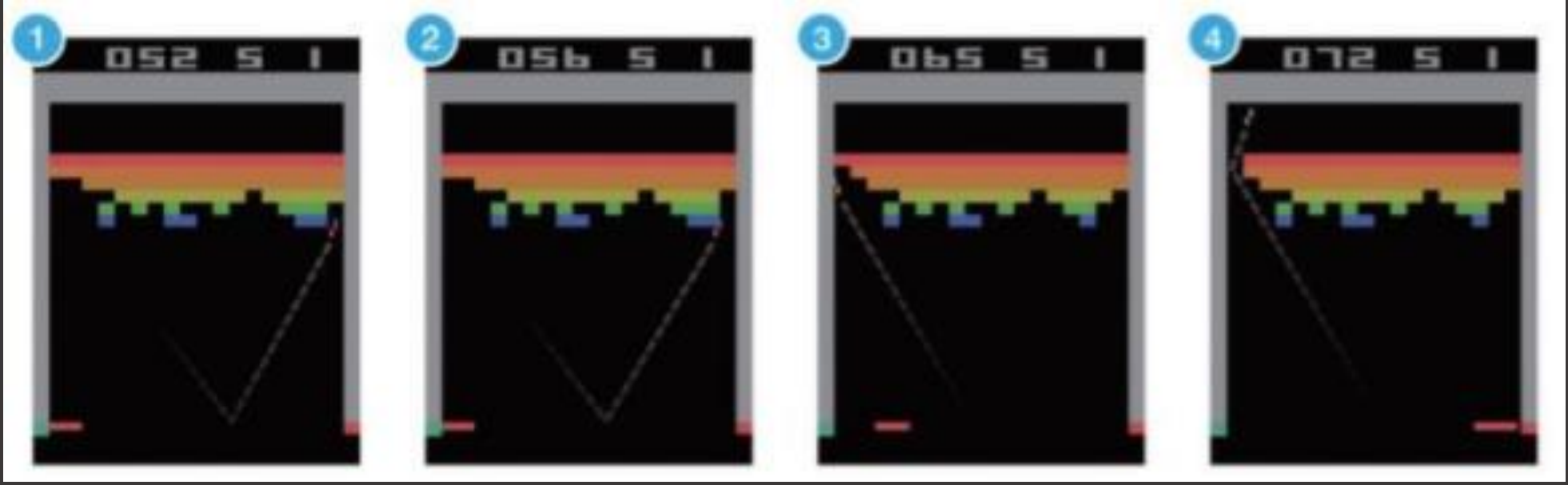
에이전트가 무엇을 관찰?

상태(state), 보상(reward)

상태(s) -> 현재 상황을 나타내는 정보

에이전트가 탁구를 치려면 탁구공의 위치, 속도, 가속도와 같은 정보가 필요

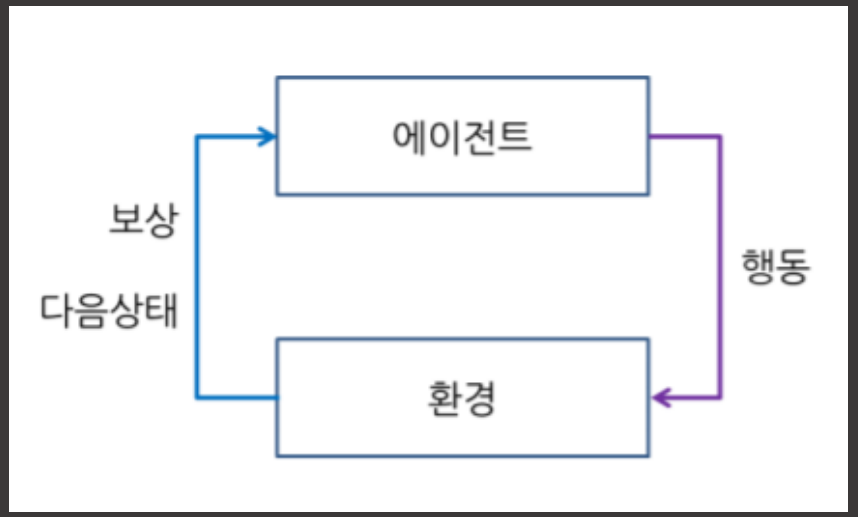
보상(r) -> 행동의 좋고 나쁨을 알려주는 정보



보상은 에이전트가 달성하고자 하는 목표에 대한 정보를 담고 있다

에이전트와 환경의 상호작용 과정

1. 에이전트가 환경에서 자신의 상태를 관찰
2. 그 상태에서 어떠한 기준에 따라 행동을 선택
3. 선택한 행동을 환경에서 실행
4. 환경으로부터 다음 상태와 보상을 받음
5. 보상을 통해 에이전트가 가진 정보를 수정

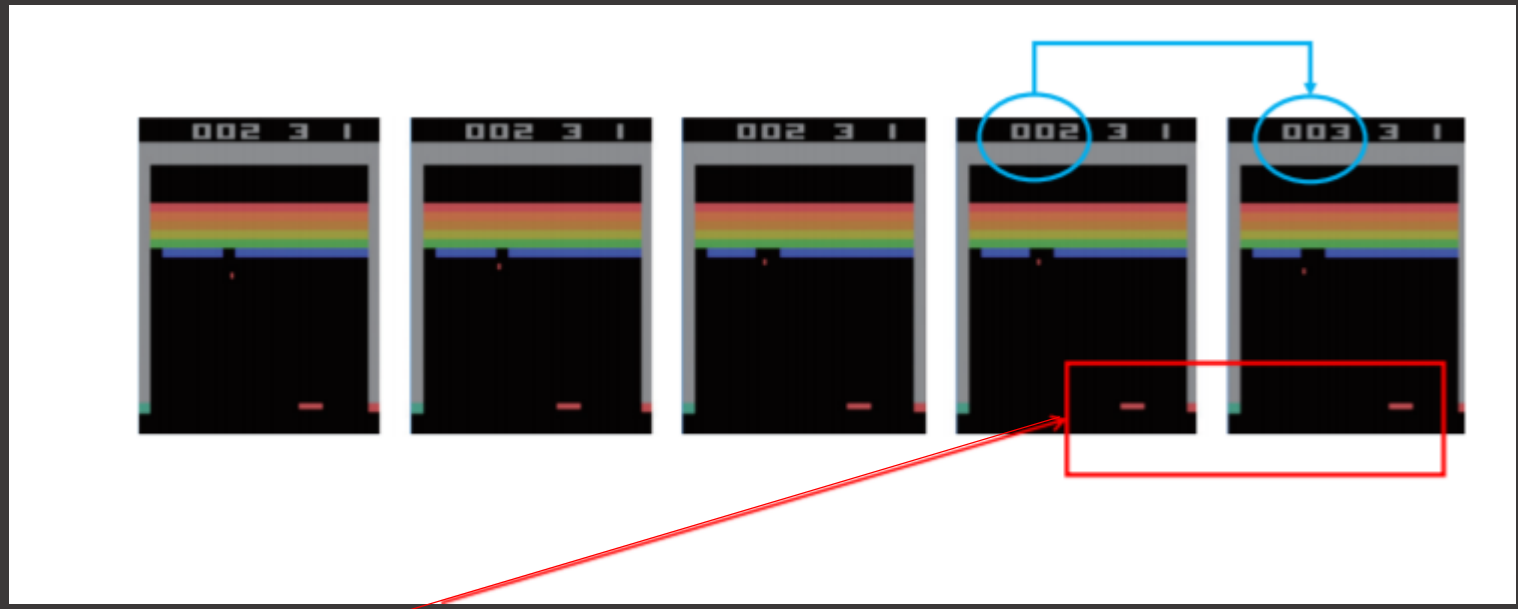


어떻게 행동을 선택?

판단 기준의 필요 -> 가치함수(Value Function)

가치함수 (Value Function)

- 만약 즉각적인 보상만을 고려해서 행동을 선택한다면?



이 행동만이 좋은 행동이고 나머지는 아니다?

가치함수 (Value Function)

- 보상은 딜레이(Delay) 된다
- 어떤 행동이 그 보상을 얻게 했는지 명확하지 않다

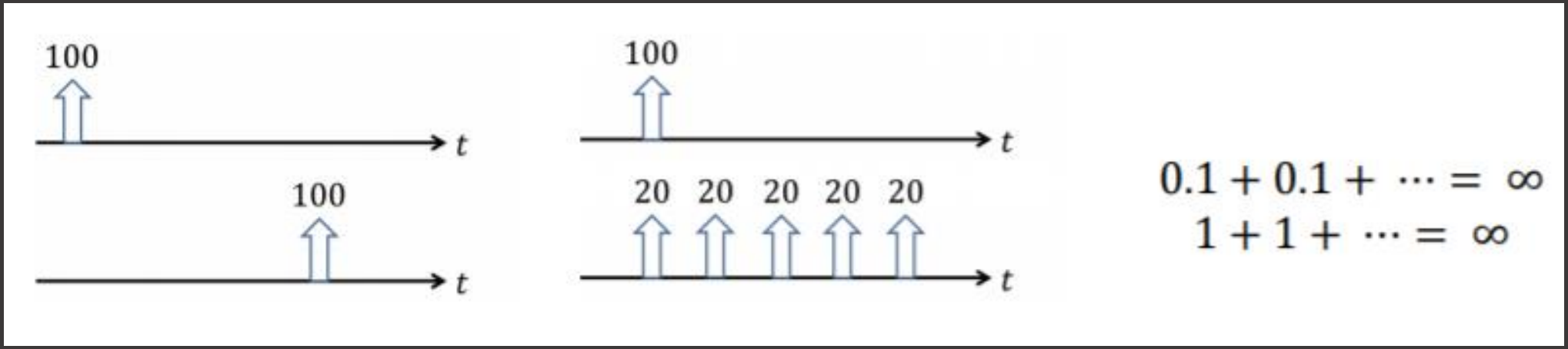
그렇다면 앞으로 받을 보상을 싹 다 더해보자

현재 시간 = t

보상의 합 = $R_{t+1} + R_{t+2} + \cdots + R_T$

가치함수 (Value Function)

- 에이전트는 항상 현재에 판단을 내리게 때문에 현재에 가까운 보상일수록 더 큰 가치를 가짐
- 시간이 지나서 받는지를 고려하기 위해 감가율 γ 개념 도입(discount factor)



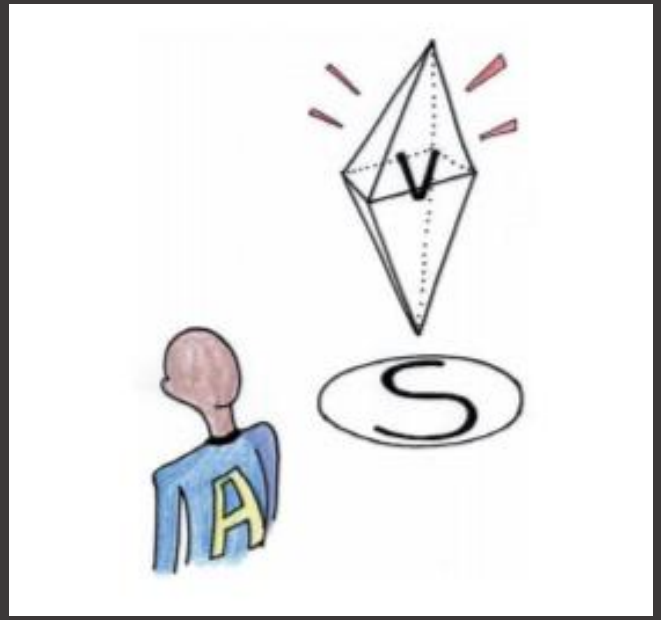
감가율 $0 \leq \gamma \leq 1$

보상의 합 = $R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t-1} R_T$

가치함수 (Value Function)

- 하지만 아직 보상을 받지 않았는데....? 미래에 받을 보상을 어떻게 알지?

지금 상태에서 미래에 받을 것이라 기대하는 보상의 합 = 가치함수



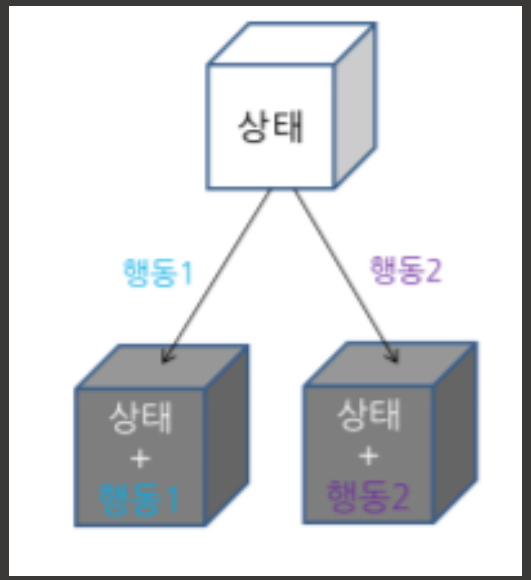
가치함수 v(s) = $\mathrm{E}[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s]$

보상에 대한 기댓값

큐함수(Q function)

- 하지만 내가 알고 싶은 건 '어떤 행동이 좋은가'인데?

지금 상태(s)에서 이 행동(a)을 선택했을 때 미래에 받을 것이라 기대하는 보상의 합
= 큐함수



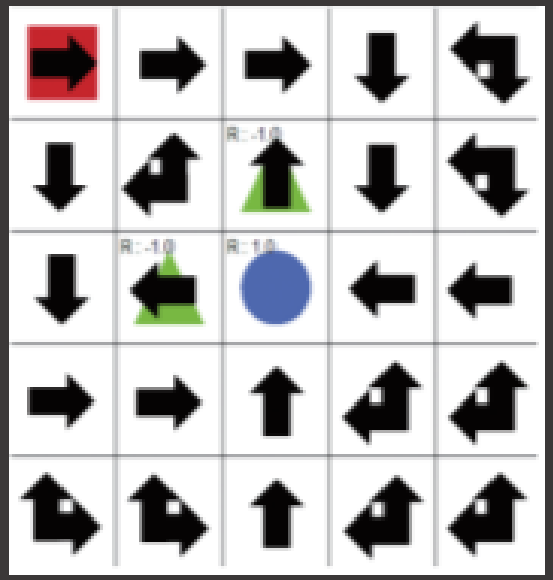
큐함수 q(s,a) = E[$R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s, A_t = a$]

정책(Policy)

- 미래에 대한 기대 -> 내가 어떻게 행동할 것인지를 알아야 함
- 각 상태에서 에이전트가 어떻게 행동할 지에 대한 정보

상태 s에서 행동 a를 선택할 확률

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



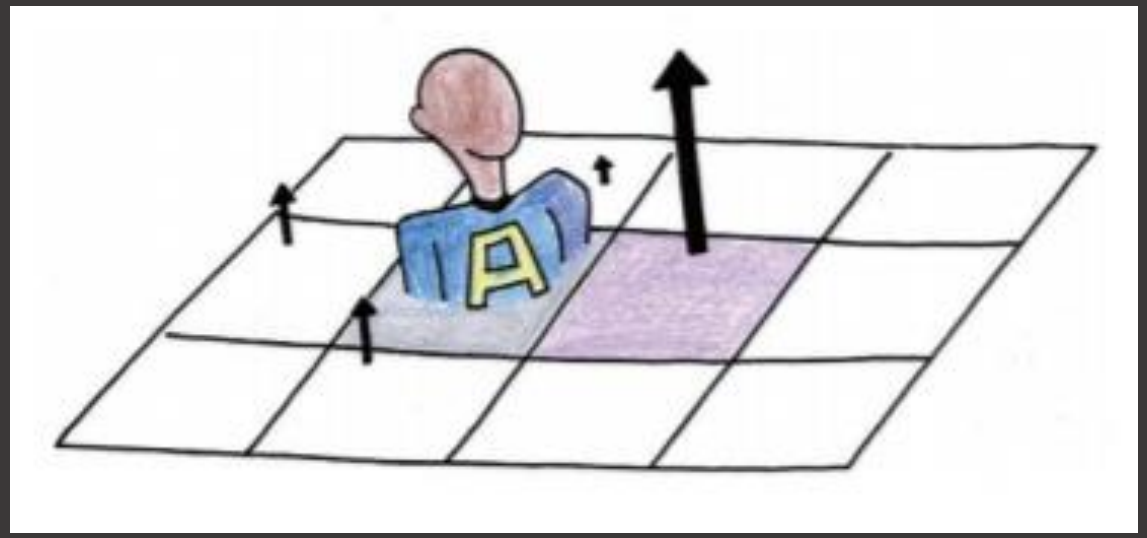
가치함수 $v_\pi(s) = E_\pi[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s]$
큐함수 $q_\pi(s,a) = \mathbf{E}[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s, A_t = a]$

큐함수를 통해 어떻게 행동을 선택?

greedy Policy

greedy Policy

- 지금 상태에서 선택할 수 있는 행동 중에 큐함수가 가장 높은 행동을 선택



greedy Policy $\pi'(s) = argmax_a q_\pi(s,a)$

어떻게 학습? 큐함수의 업데이트

벨만 방정식(Bellman equation)

- 기존 큐함수의 정의 : q_π(s,a) = $E_\pi[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s, A_t = a]$
  -> 위의 식으로부터 기댓값을 알려면 앞으로의 모든 보상을 고려해야함 -> 불가능

- 한 번에 계산하는 것이 아닌, 순차적으로 풀어가는 방법 고려
  -> 현재 상태의 가치함수와 다음 상태의 가치함수 사이의 관계를 식으로 표현

기존 큐함수 : q_π(s,a) = $E_\pi[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s, A_t = a]$

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

벨만 기대 방정식(Bellman expectation equation)

브레이크아웃의 MDP와 학습 방법

오픈 에이아이 짐의 브레이크아웃 게임 화면

브레이크아웃의 MDP와 학습 방법

1. MDP

상태 : 게임 화면 (총 4개의 연속적인 화면을 받는다)

행동 : 제자리, 왼쪽, 오른쪽, 발사(게임 시작시만 사용)

보상 : 벽돌이 하나씩 깨질 때마다 (+1점)
더 위쪽의 벽돌을 깰수록 (더 큰 보상)
아무것도 깨지 않을 때(0점)
공을 놓쳐 목숨을 잃을 경우 (-1점)

브레이크아웃의 MDP와 학습 방법

2. 학습

처음 에이전트는 무작위로 움직이다가, 우연히 공을 쳐서 벽돌을 깨면 '환경(게임)'으로 부터 1만큼의 보상을 받는다. (화면 인식은 CNN으로)

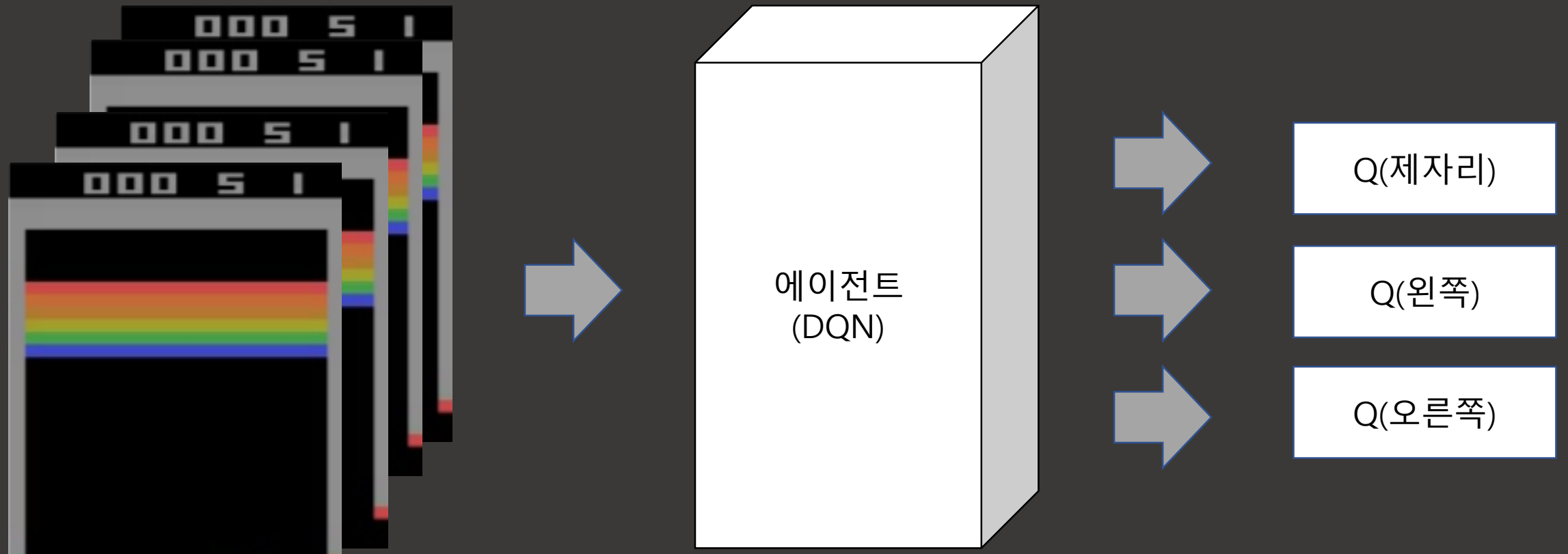
위의 내용을 통해 인공신경망(딥러닝)을 학습시킨다.

입력으로 화면(4개)이 들어오면, 인공신경망은 출력으로 에이전트가 할 수 있는 행동 중 큐함수가 가장 높은 행동을 출력으로 내보낸다.

이러한 인공신경망을 $DQN^{Deep-Q-Learning}$이라고 한다.

브레이크아웃의 MDP와 학습 방법

2. 학습



브레이크아웃 게임의 심층 신경망

브레이크아웃의 MDP와 학습 방법

2. 학습

1. 에이전트는 4개의 연속된 게임 화면을 입력으로 받는다.
2. 처음에는 아무것도 모르므로 임의로 행동을 취한다.
3. 그에 따라 보상을 받게 되면 그 보상을 통해 학습한다.
4. 결국 사람처럼 혹은 사람보다 잘하게 된다.

브레이크아웃의 MDP와 학습 방법

3. 문제점

1. 사람이라면 처음 게임을 시작할 때, 게임의 규칙을 보고 어느 정도 게임에 대한 사전지식을 가지고 시작하지만, 강화학습 에이전트는 규칙을 전혀 모른채 시작한다.
   -> 규칙을 몰라도 학습할 수 있다는 장점이자, 초반의 느린 학습의 원인

2. 사람은 하나를 학습하면 다른 곳에도 그 학습이 영향을 미치지만, 현재 강화학습 에이전트는 항상 바닥부터 학습을 해야한다.
   -> 현재 및 미래 강화학습 분야의 연구 분야로서 지속적으로 해결해야 할 과제