KOREA UNIVERSITY
SEJONG CAMPUS

강화학습

강화학습 소개

고려대학교 세종캠퍼스 인공지능사이버보안학과
구 자 훈

목차

PREVIEW

- 아이의 걸음마 학습
  - 아이는 매 순간 행동을 취함
  - 좋은 행동은 즐거움을 느끼는 양의 보상, 나쁜 행동은 넘어져 아픔을 느끼는 음의 보상
  - 양의 보상을 받은 행동은 늘리고, 음의 보상을 받은 행동은 줄임
  - 이 과정을 반복함으로써 신나게 노는 아이로 성장

그림 1-1 인간의 강화 학습

- 이런 고도의 지능적인 학습 과정을 기계가 흉내 낼 수 있을까?
  - 강화 학습은 긍정적인 답 제공

1.1 강화 학습이란?

- 동물은 자신에게 유리한 행동을 학습할 수 있어야 생존
- 학습 능력이 뛰어난 개체는 그렇지 않은 개체보다 더 나은 생존 가능성

- 동물의 강화학습을 살펴봤을 때, 컴퓨터로 이를 어떻게 모방할 수 있을까

1.1.1 동물의 강화 학습

- 스키너 상자Skinner box
  - 설정
    - 파란 불일 때, 손잡이를 누르면 맛있는 치즈라는 양의 보상
    - 빨간 불일 때, 손잡이를 누르면 감전에 따른 음의 보상
  - 쥐의 강화 학습
    - 처음에는 반반의 확률로 랜덤 선택
    - 점점 파란 불은 강화되고 빨간 불은 억제

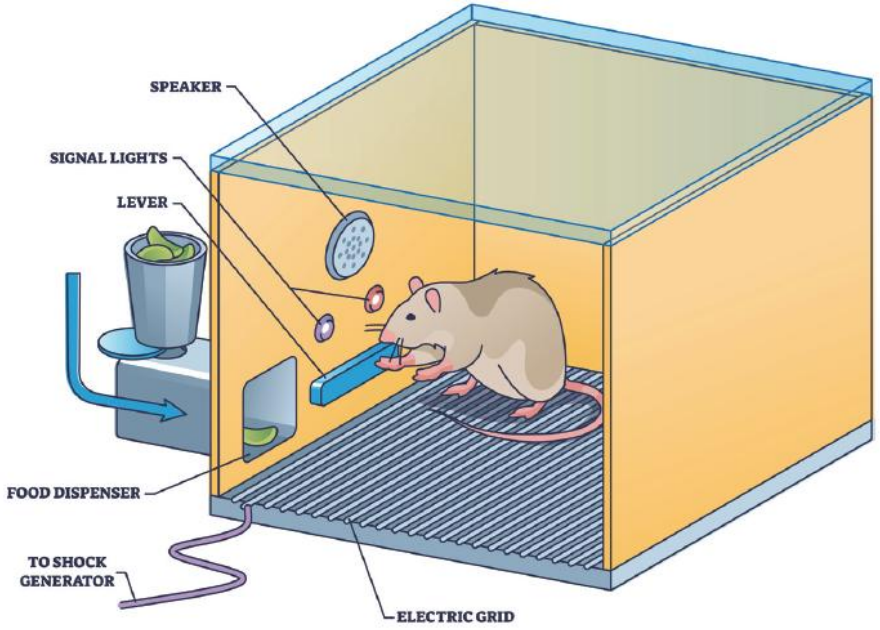


그림 1-2 스키너 상자

1.1.1 동물의 강화 학습

- 인간의 강화 학습
  - 걸음마, 수영, 자전거 타기, 자동차 운전 등
  - 상대의 마음을 얻는 일 등
- 학습하는 주체, 주체와 상호작용하는 환경, 주체가 취하는 행동, 행동에 따라 변하는 상태, 주어진 보상
- 에이전트agent와 환경environment의 상호작용interaction
  - 에이전트는 오랜 시간 환경과 상호작용하면서 보상을 최대화하는 행동을 학습함

1.1.2 기계의 강화 학습

- 수학을 통한 문제 정의와 해결방법 구상
- 에이전트가 환경과 상호작용하는 방식
  - 에이전트가 행동action을 선택하면,
  - 환경은 행동에 따라 상태state를 변화시키고 보상reward을 제공
  - 순간time-step 0에서 시작하여 1, 2, 3, ...으로 진행

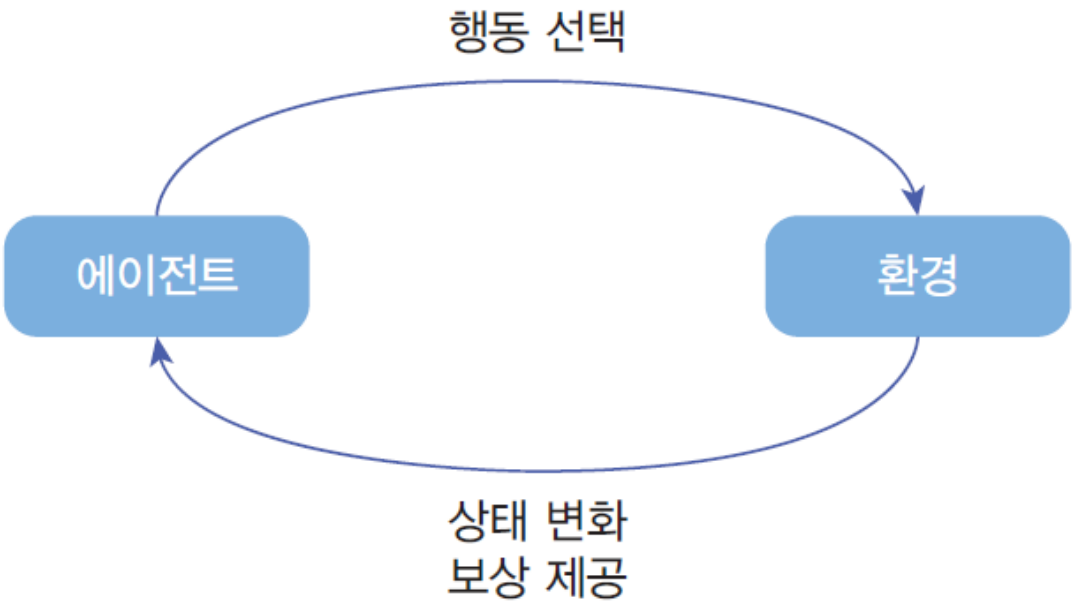


그림 1-3 강화 학습에서 에이전트와 환경의 상호작용

1.1.2 기계의 강화 학습

- 에이전트와 환경의 구분
  - 과업에 따라 혼란스럽거나 애매한 경우 있음
  - [그림 1-1]의 걸음마 배우기
    - 바닥을 환경, 아이를 에이전트로 하면 부적절함 ← 왜?
    - 아이의 몸을 환경, 아이의 두뇌를 에이전트로 설정해야 함 ← '왼발에 힘 ' 이라는 행동을 취하면, 몸의 상태가 바뀌고 보상이 주어짐
  - 이족 보행 로봇: 제어 장치가 에이전트, 로봇의 나머지 부분이 환경
  - 자율주행: 제어 박스가 에이전트, 차량의 나머지 부품과 도로는 환경
  - 바둑: 돌을 둘 곳을 정하는 프로그램이 에이전트, 바둑판은 환경

그림 1-1 인간의 강화 학습

1.1.2 기계의 강화 학습

행동과 상태, 보상이란?

- 바둑의 사례
  - 행동: 돌을 둘 곳 좌표 {(1,1}, (1,2), ..., (1,19), (2,1), ..., (19,19)} ← 361개 요소의 집합
  - 상태: 19x19 격자 상황
  - 보상: 게임 진행되는 동안 0, 승패가 결정된 종료 상태에서 이기면 +1, 지면 -1

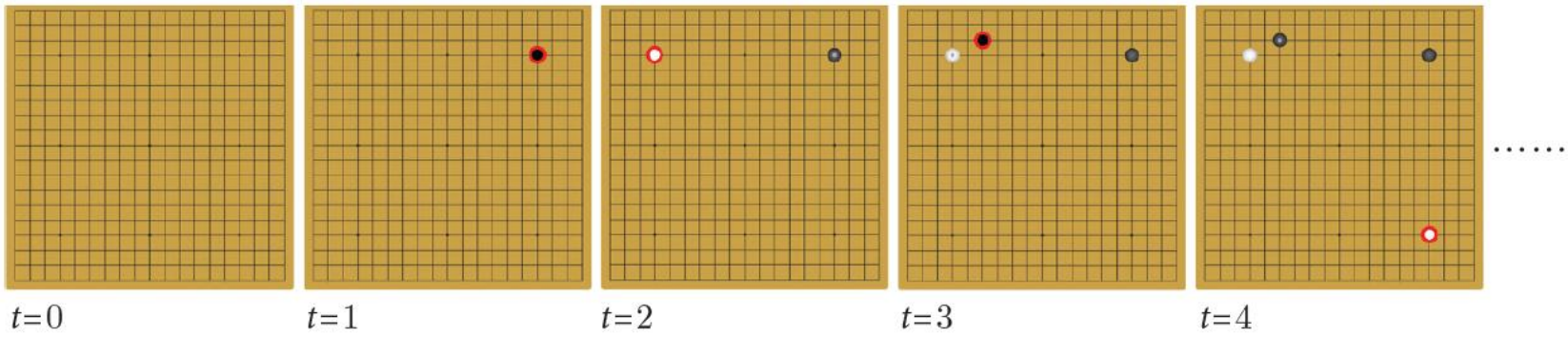


그림 1-4 바둑 게임의 상태 변화

- 바둑은 지연 보상delayed reward
  - 방대한 상태 공간을 효율적으로 표현하고 빠르게 처리하는 방법, 지연된 보상의 희소한 정보를 효과적으로 학습하는 방법이 강화학습의 핵심

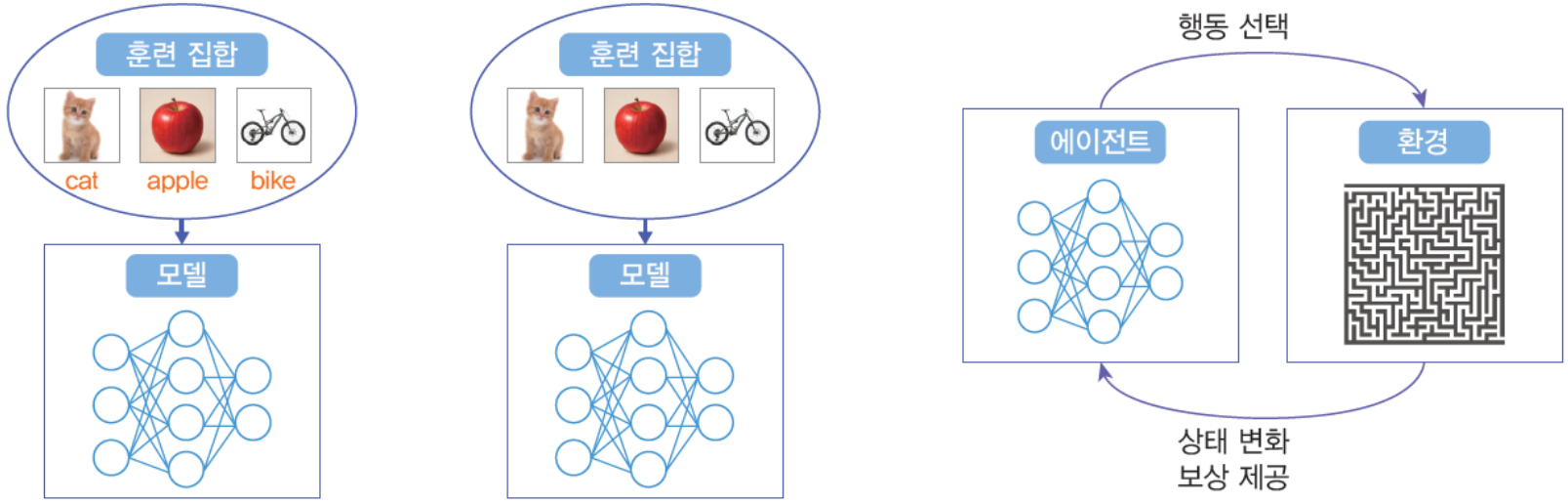
1.1.2 기계의 강화 학습

- 현대 강화 학습 기법
  - 가능한 상태의 수가 사실상 무한인데 어떻게 표현하고 처리할까?
  - 지연된 보상의 희소한 정보를 가지고 어떻게 효과적으로 학습할까?
  - 누적 보상을 최대화하는 행동을 어떻게 결정할까?

1.2 기계 학습=지도 학습+비지도 학습+강화 학습

기계 학습

- 지도 학습supervised learning: 참값(레이블)ground truth(GT)을 가진 샘플로 구성된 훈련 집합 사용
  - 영상의 경우 분류, 검출, 분할, 추적 등
- 비지도 학습unsupervised learning: 참값이 없는 샘플로 구성된 훈련 집합 사용
  - 차원 축소, 특징 추출, 생성 모델 등
- 강화 학습: 에이전트와 환경의 상호작용에서 발생하는 데이터를 활용한 학습



(a) 지도 학습 (b) 비지도 학습 (c) 강화 학습

그림 1-5 기계 학습의 종류

1.2 기계 학습=지도 학습+비지도 학습+강화 학습

- 강화 학습이 데이터를 생성하고 소비하는 과정(자율 학습self-learning)
  - 에이전트는 정책policy을 통해 행동을 선택. 처음에는 랜덤하게 정책 초기화하고 시작
  - 에피소드episode라 부르는 데이터 생성. 처음에는 나쁜 품질의 에피소드 데이터
  - 에피소드 데이터로 정책을 개선
  - 개선된 정책으로 조금 나은 에피소드 생성
  - 반복하면서 최적 정책optimal policy으로 수렴

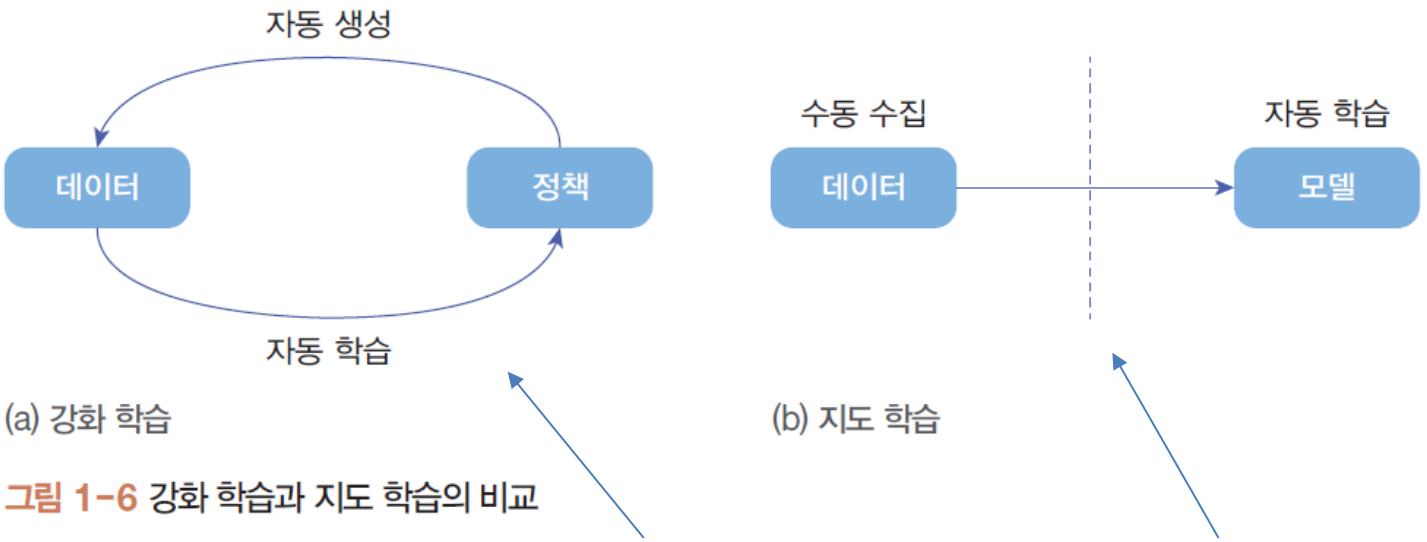


그림 1-6 강화 학습과 지도 학습의 비교

데이터 수집과 학습이 밀접합

데이터 수집과 학습이 분리됨

1.3 성공 사례와 응용 분야

- 인공지능의 혁신으로 여겨지는 많은 성공 스토리
- 최근 응용 분야가 급속 확대
  - 자율주행과 지능 로봇 등
  - 인공일반지능artificial general intelligence(AGI)을 구현하는 핵심 기술

1.3.1 성공 사례

- TD-Gammon
  - Gerald Tesauro는 1990년대 초반부터 백개먼을 플레이하는 프로그램 제작
  - 초기에는 지도 학습으로 신경망을 학습하는 NeuroGammon. 낮은 성능
  - 이후 강화 학습에 주목하여 시간차 학습 알고리즘 적용한 TD-Gammon. 세계 챔피언 수준
  - 시간차 학습은 5장

1.3.1 성공 사례

- 아타리 게임
  - Mnih는 49종 아타리 게임을 플레이하는 인공지능 프로그램 개발. 전문가 수준 이상
  - 210x160 컬러 화면 영상이 상태, 입력 장치 조작이 행동에 해당
  - 7장에서 DQN 학습 알고리즘으로 실습 수행

그림 1-7 강화 학습의 성공 사례(왼쪽부터 아타리 게임, 알파고, 파쿠르 로봇)

1.3.1 성공 사례

- 알파고
  - 프로 기사의 기보 데이터로 지도 학습 수행한 후, 자기 대전self-play 통한 강화 학습 적용
  - MCTS와 결합하여 강력해짐
  - 이후 알파고 제로, 알파제로, 뮤제로로 발전
  - 13.3절에서 상세하게 설명
- 파쿠르 로봇
  - 장애물 뛰어넘기, 공중제비, 바닥 기기, 철봉 매달리기 등을 수행하는 로봇
  - 예상치 못한 지형지물 대처하려면 일반화 능력 필수
  - PPO 학습 알고리즘(10장) 사용한 사례
- ChatGPT
  - 대규모 언어 모델을 얼라이닝하는데 강화 학습 활용
  - ChatGPT, Claude, Gemini 등에 적용

1.3.2 응용 분야

- 다양한 응용 분야로 확대하는 중
  - 게임
  - 로봇
  - 자연어 처리
  - 컴퓨터 비전
  - 자율주행
  - 에너지
  - 추천
  - 운영체제와 클라우드 컴퓨팅
  - 경제
  - 항공
  - 의료
  - 농업
  - 과학 발견
  - …

1.4 간략 역사

표 1-1 강화 학습의 간략 역사

연도	사건
1947	몬테카를로 방법 탄생[Metropolis1987, Eckhardt1987]
1957	리처드 벨만(Richard E. Bellman), 『Dynamic Programming』 출간[Bellman1957]
1958	프랭크 로젠블랫(Frank Rosenblatt), 퍼셉트론 제안[Rosenblatt1958]
1959	Checkers 프로그램에서 시간차 아이디어 창안[Samuel1959, Sutton2018(16.2절)]
1983	강화 학습으로 CartPole 과업 해결 및 행동가-비평가 알고리즘 창안[Barto1983]
1986	데이비드 루멜하트(David E. Rumelhart), 다층 퍼셉트론을 설명한 『Parallel Distributed Processing』 출간[Rumelhart1986]
1988	시간차 아이디어를 강화 학습의 주요 알고리즘으로 정립[Sutton1988]
1989	Q-러닝 창안[Watkins1989]
1992	백개먼을 플레이하는 TD-Gammon 공개[Tesauro1992, Tesauro1995] REINFORCE 알고리즘 탄생[Williams1992]
1993	바둑 프로그램에 처음으로 몬테카를로 방법 적용[Brugmann1993]
1994	Sarsa 알고리즘 창안[Rummery1994]
1998	컨볼루션 신경망을 실용화한 LeNet-5 공개[LeCun1998]
1999	정책 그레이디언트 정리 제안[Sutton1999]

1.4 간략 역사

2002	UCB 공식 탄생[Auer2002]
2003	Bullet Physics 라이브러리 공개[Coumans2015]
2006	UCB를 트리 탐색에 적용하는 UCT 창안[Kocsis2006] 몬테카를로 트리 탐색 알고리즘 탄생[Coulom2006]
2012	AlexNet이 ILSVRC 대회 우승하여 딥러닝 시대 엶[Krizhevsky2012] 아타리 게임 에뮬레이터 ALE 공개[Bellemare2013] MuJoCo 라이브러리 공개[Todorov2012]
2013	DQN으로 아타리 게임 정복[Mnih2013, Mnih2015]
2015	구글이 텐서플로 공개[Abadi2016] UC Berkeley가 가사 노동 로봇 BRETT 공개[Yang2015] TRPO 알고리즘 창안[Schulman2015]
2016	알파고가 이세돌을 이김[Silver2016] OpenAI가 Gym 라이브러리를 공개[Brockman2016] DeepStack 프로그램이 2인 포커 게임에서 전문가를 이김[Moravcik2017]
2017	PPO 알고리즘 창안[Schulman2017] 알파고 제로가 알파고를 100:0으로 이김[Silver2017] OpenAI Five 초기 버전을 The International 2017 대회에서 공개 사람 피드백을 강화 학습에 적용하는 RLHF 기법 제안[Christiano2017] PyBullet 라이브러리 공개[Coumans2016]
2018	AlphaStar가 StarCraft II 게임에서 세계 챔피언을 이김[Vinyals2019]

1.4 간략 역사

2019	OpenAI Five가 Dota 2 게임에서 세계 챔피언을 이김[Berner2019] OpenAI가 루빅스 큐브를 맞추는 로봇 공개[Akkaya2019] UC Berkeley가 레고 조립이 가능한 SOLAR 로봇 공개[Zhang2019] Pluribus 프로그램이 6명 포커 게임에서 세계 챔피언을 이김[Brown2019] 사람 피드백을 LLM에 적용[Ziegler2019] 엔비디아가 Isaac Sim 최초 공개[Zhou2023]
2020	컬링 로봇 Curly가 인간 컬링 팀을 이김[Won2020]
2021	OpenAI가 Gym의 관리 권한을 Farama 재단에 이관 Farama 재단의 Gymnasium 서비스 개시[Towers2024] 테슬라가 Optimus 로봇 공개[Malik2023] 딥마인드가 MuJoCo를 인수하고 무료 공개[Towers2024]
2022	OpenAI가 ChatGPT를 대중에 공개[Bubeck2023] 딥마인드가 MuJoCo를 오픈소스로 공개[DeepMind2022]
2023	앤스로픽이 Claude를 대중에 공개[Bai2022] 구글이 Gemini를 대중에 공개[Gemini-team2023]
2024	학술 논문을 자동 생성하는 The AI Scientist 공개[Lu2024] 구글이 탁구 로봇 공개[D'Ambrosio2024] 알파폴드에 강화 학습을 적용한 연구[Aderinwale2024] 딥마인드가 Genie와 Genie2 공개[Bruce2024, Parker-Holder2024]
2025	민첩한 휴머노이드 ASAP 로봇 공개[He2025] Sutton과 Barto가 2024년 Turing Award 수상(https://awards.acm.org/about/2024-turing) Gemini Robotics 공개[DeepMind2025]

PREVIEW

- 강화 학습은 누적 보상의 기댓값, 즉 기대 이득expected return을 최대화
  - 동물은 때때로 순간 보상을 희생하고, 기대 이득을 최대화 (단기보상 / 장기보상)
  - 바둑에서도 대마를 위해 종종 작은 집을 희생

그림 2-1 순간 보상을 희생함으로써 이득을 최대화하는 누 떼

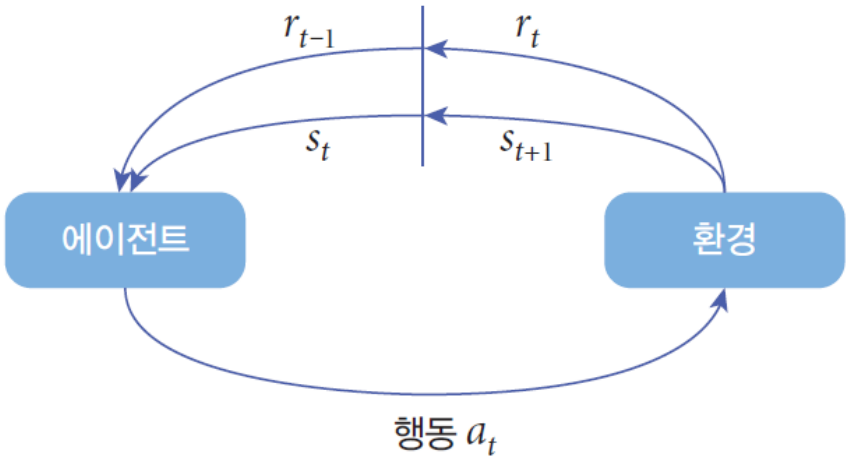
  - 강화 학습 문제를 수학적 공식화하고, 알고리즘으로 변환하고, 파이썬 프로그래밍 시작
  - 예제를 통한 강화학습의 용어와 공식의 직관적 이해 유도

2.1 환경과 상호작용하는 에이전트

- 에이전트와 환경의 구성 요소와 상호작용 방식을 수학적으로 정의
- 마르코프 결정 프로세스로 표현

2.1.1 마르코프 결정 프로세스

- 마르코프 결정 프로세스(Markov decision process, MDP)
  - 에이전트는 행동action 선택. 환경은 상태state 변화와 보상reward 제공
  - 순간 0에서 시작하여, $1,2,3,\ldots,T$로 진행



(a) 사이클로 표현

$$[-,s_0]a_0[r_0,s_1]a_1[r_1,s_2]a_2\cdots\cdots[r_{t-1},s_t]a_t[r_t,s_{t+1}]a_{t+1}\cdots\cdots[r_{T-2},s_{T-1}]a_{T-1}[r_{T-1},s_T]$$

(b) 펼쳐서 표현

그림 2-2 마르코프 결정 프로세스

2.1.1 마르코프 결정 프로세스

- 에피소드 e 생성
  - 순간 t에서 환경은 상태 s_t에 있음
  - 에이전트가 행동 a_t를 선택하면, 다음 순간 $t+1$로 넘어가고 환경은 상태 s_{t+1}로 전환되고 보상 r_t를 제공
  - 이런 과정을 순차적으로 $t = 0,1,2,\cdots,T$까지 반복 (T는 종료 상태에 도달한 순간)

$$e=[-,s_0]a_0[r_0,s_1]a_1[r_1,s_2]a_2\cdots[r_{T-2},s_{T-1}]a_{T-1}[r_{T-1},s_T] \qquad (2.1)$$

  - 에피소드의 일부를 궤적trajectory이라 부름

논리적으로는 행동을 수행한 후에 상태가 바뀌고 보상이 발생하므로 $[-,s_0]a_0[r_1,s_1]a_1[r_2,s_2]\cdots$로 표기하는 것이 합리적이다. 하지만 현대적인 학습 알고리즘은 현재 상태 s에서 행동 a를 취했을 때 발생하는 보상 r, 즉 s-a-r을 단위 정보로 활용하여 학습한다. 따라서 $[-,s_0]a_0[r_1,s_1]a_1[r_2,s_2]\cdots$로 표기하면 단위 정보가 s_t-a_t-r_{t+1}이 되어 보상을 저장하는 배열의 인덱스가 오른쪽으로 한 칸 밀려 코딩할 때 혼란이 발생한다. 이런 이유 때문에 현대적인 강화 학습 알고리즘은 주로 식 (2.1)의 표기를 사용한다.

2.1.1 마르코프 결정 프로세스

- 상태, 행동, 보상 공간
  - $\mathcal{S}$는 발생 가능한 모든 상태의 집합
  - $\mathcal{A}$는 발생 가능한 모든 행동의 집합
  - $\mathcal{R}$은 발생 가능한 모든 보상의 집합

$$\left.\begin{aligned} s_t &\in \mathcal{S} \\ a_t &\in \mathcal{A} \\ r_t &\in \mathcal{R} \end{aligned}\right\} \quad (2.2)$$

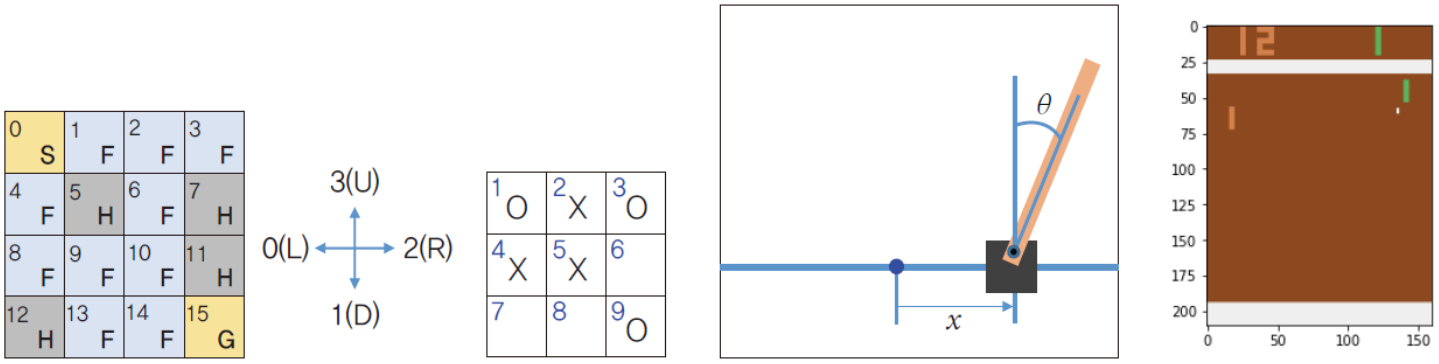
- 강화 학습은 누적 보상accumulated reward, 즉 이득return R을 최대화함
  - e는 에피소드, z는 순간 t에서 시작하는 궤적

$$\left.\begin{aligned} R(e) &= \sum_{i=0,1,\cdots,T-1} r_i \\ R(z) &= \sum_{i=t,t+1,\cdots,T-1} r_i \end{aligned}\right\} \quad (2.3)$$

- 강화 학습의 최종 목표
  - 기대 이득expected return을 최대화. 즉 R의 기댓값 $\mathbb{E}(R)$을 최대화

2.1.2 과업 사례

- 이산 과업discrete task과 연속 과업continuous task
  - 행동을 이산 값으로 표현하면 이산 과업, 연속 값으로 표현하면 연속 과업
  - Gymnasium 라이브러리에서 환경 제공



(a) FrozenLake (b) 틱택토 (c) CartPole (d) 퐁 게임

그림 2-3 이산 과업 사례

(a) Pendulum (b) BipedalWalker (c) MuJoCo

그림 2-4 연속 과업 사례

2.1.2 과업 사례

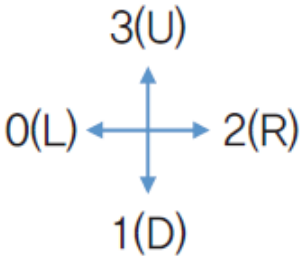
- 이산 과업: [예제 2-1] FrozenLake
  - 4x4 보드에서 0번 S 칸에서 출발하여 15번 G 칸에 도달하는 것이 목표
  - G 칸에 도달하면 보상 1을 받고 종료. H 칸에 도달하면 보상 0을 받고 종료
  - 상태 공간의 크기 $|\mathcal{S}|$는 16, 행동 공간의 크기 $|\mathcal{A}|$는 4인 아주 작은 과업

$\mathcal{S}=$

$\mathcal{A}=$

$\mathcal{R}=$

0 S	1 F	2 F	3 F
4 F	5 H	6 F	7 H
8 F	9 F	10 F	11 H
12 H	13 F	14 F	15 G



2.1.2 과업 사례

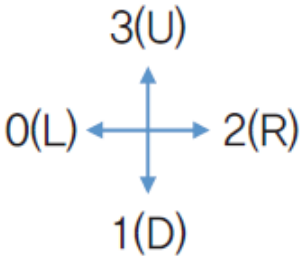
- 이산 과업: [예제 2-1] FrozenLake
  - 4x4 보드에서 0번 S 칸에서 출발하여 15번 G 칸에 도달하는 것이 목표
  - G 칸에 도달하면 보상 1을 받고 종료. H 칸에 도달하면 보상 0을 받고 종료
  - 상태 공간의 크기 $|\mathcal{S}|$는 16, 행동 공간의 크기 $|\mathcal{A}|$는 4인 아주 작은 과업

$$\mathcal{S}=\{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$$

$$\mathcal{A}=\{0(\mathrm{L}),1(\mathrm{D}),2(\mathrm{R}),3(\mathrm{U})\}$$

$$\mathcal{R}=\{1,0\}$$

0 S	1 F	2 F	3 F
4 F	5 H	6 F	7 H
8 F	9 F	10 F	11 H
12 H	13 F	14 F	15 G



2.1.2 과업 사례

- 이산 과업: [예제 2-2] 틱택토
  - O와 X가 번갈아 수를 두는데, 가로, 세로, 대각 방향으로 연속 세 칸을 먼저 차지하면 승리
  - 상태 공간의 크기 $|\mathcal{S}|$는 $3^9 = 19683$보다 작음. 행동 공간 크기 $|\mathcal{A}|$는 9인 아주 작은 과업

$\mathcal{S}$ =

$\mathcal{A}$ =

$\mathcal{R}$ =

1 O	2 X	3 O
4 X	5 X	6
7	8	9 O

2.1.2 과업 사례

- 이산 과업: [예제 2-2] 틱택토
  - O와 X가 번갈아 수를 두는데, 가로, 세로, 대각 방향으로 연속 세 칸을 먼저 차지하면 승리
  - 상태 공간의 크기 $|\mathcal{S}|$는 $3^9 = 19683$보다 작음. 행동 공간 크기 $|\mathcal{A}|$는 9인 아주 작은 과업

$$\mathcal{S} = \{----------, O--------, \cdots\cdots, OXOXX---O, \cdots\}$$

$$\mathcal{A} = \{1,2,3,4,5,6,7,8,9\}$$

$$\mathcal{R} = \{1,0,-1\}$$

1 O	2 X	3 O
4 X	5 X	6
7	8	9 O

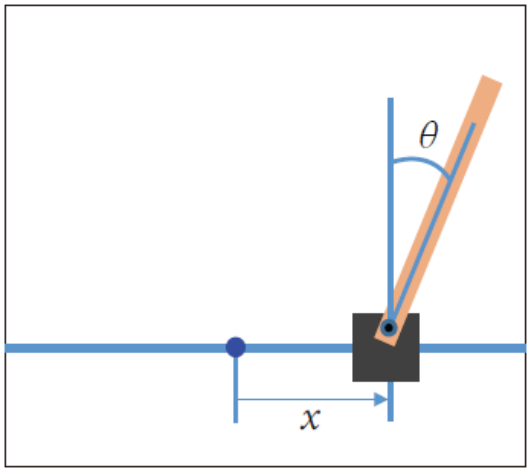
2.1.2 과업 사례

- 이산 과업: [예제 2-3] CartPole
  - 좌우로 움직이는 수레 위에서 가능한 한 막대를 오래 세우는 과업
  - 기운 각도 θ가 ± 12도를 넘거나 원점에서 2.4 이상 멀어지면 실패로 간주하고 종료 (terminated=True). 매 순간 보상 1을 줌
  - 쓰러지지 않고 500 순간 이상을 버티면 승리로 간주하고 종료 (truncated=True)
  - 게임은 [-0.05,0.05] 사이의 랜덤한 위치에서 시작
  - 상태는 4차원 실수 공간. 행동은 2차원 정수 공간

$\mathcal{S}=$

$\mathcal{A}=$

$\mathcal{R}=$



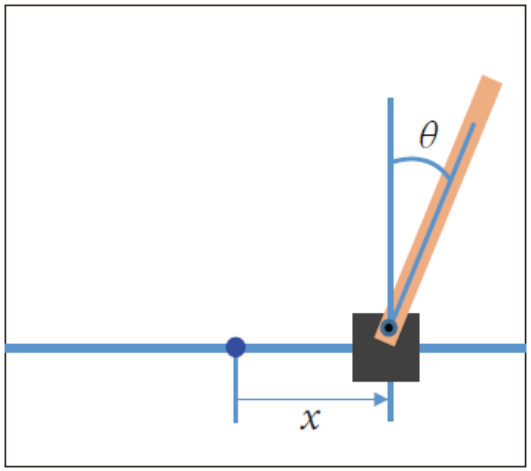
2.1.2 과업 사례

- 이산 과업: [예제 2-3] CartPole
  - 좌우로 움직이는 수레 위에서 가능한 한 막대를 오래 세우는 과업
  - 기운 각도 θ가 ± 12도를 넘거나 원점에서 2.4 이상 멀어지면 실패로 간주하고 종료 (terminated=True). 매 순간 보상 1을 줌
  - 쓰러지지 않고 500 순간 이상을 버티면 승리로 간주하고 종료 (truncated=True)
  - 게임은 [-0.05,0.05] 사이의 랜덤한 위치에서 시작
  - 상태는 4차원 실수 공간. 행동은 2차원 정수 공간

$$\mathcal{S}=\{x,\vec{x},\theta,\vec{\theta}\mid x,\vec{x},\theta,\vec{\theta}\text{는 실수}\}$$

$$\mathcal{A}=\{0(\text{왼쪽}),1(\text{오른쪽})\}$$

$$\mathcal{R}=\{1\}$$



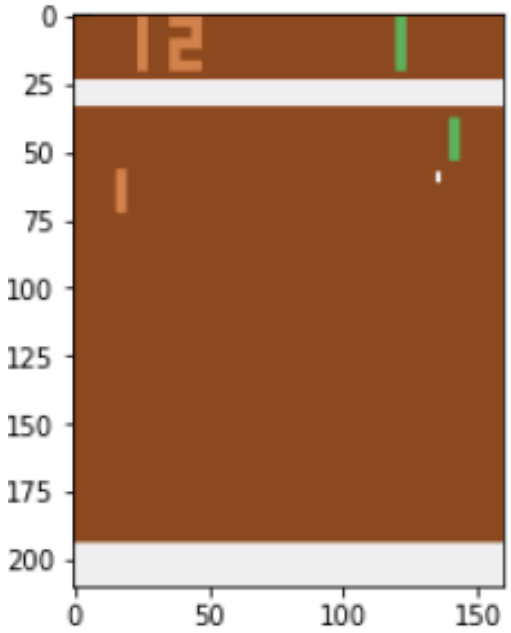
2.1.2 과업 사례

- 이산 과업: [예제 2-4] 퐁 게임
  - 에이전트는 녹색 배트를 위아래로 움직여 공을 되받아 쳐내야 함
  - 210x160 컬러 화면 영상이 상태에 해당. 행동은 배트를 {정지, 위, 아래}의 세 가지
  - 게임 도중에는 보상 0, 승패가 결정되면 이기면 1 지면 -1

$\mathcal{S}=$

$\mathcal{A}=$

$\mathcal{R}=$



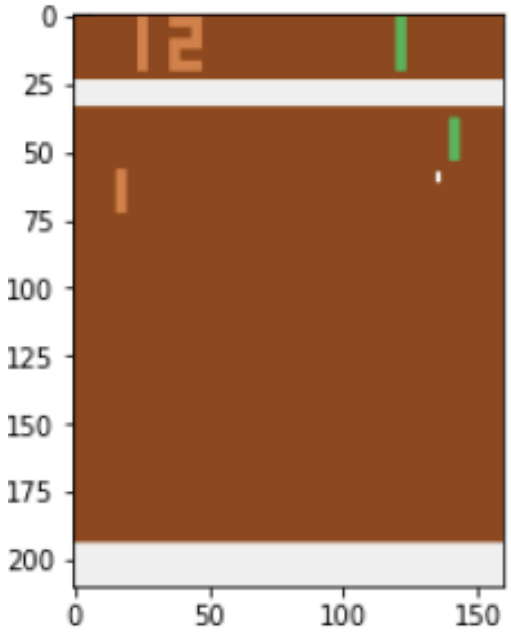
2.1.2 과업 사례

- 이산 과업: [예제 2-4] 퐁 게임
  - 에이전트는 녹색 배트를 위아래로 움직여 공을 되받아 쳐내야 함
  - 210x160 컬러 화면 영상이 상태에 해당. 행동은 배트를 {정지, 위, 아래}의 세 가지
  - 게임 도중에는 보상 0, 승패가 결정되면 이기면 1 지면 -1

$$\mathcal{S}=\{\text{화면}1,\text{화면}2,\cdots\cdots\}$$

$$\mathcal{A}=\{0(\text{정지}),2(\text{위}),3(\text{아래})\}$$

$$\mathcal{R}=\{0,1,-1\}$$



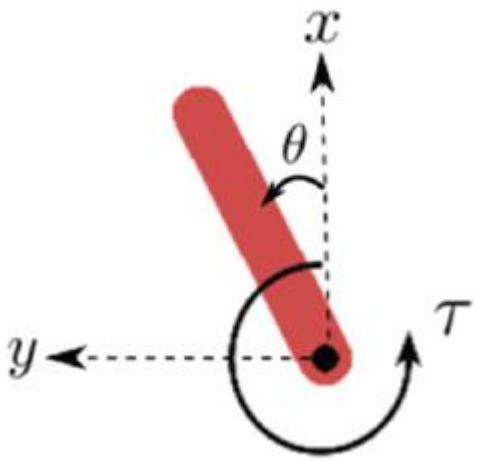
2.1.2 과업 사례

- 연속 과업: [예제 2-5] Pendulum
  - 막대에 힘을 가해 위쪽을 향하게 한 다음 오래 머물게 하는 것이 목표
  - 막대의 끝 좌표 x와 y, 각을 나타내는 θ가 3차원 상태 공간을 구성(3차원 실수 공간)
  - 행동은 막대에 가하는 힘 τ(1차원 실수 공간) → τ가 실수 이므로 연속 과업
  - 보상은 θ와 τ로 계산함

$$\mathcal{S}=\{(x,y,\theta)\,|\,x,y,\theta\text{는 실수}\}$$

$$\mathcal{A}=\{\tau\,|\,\tau\text{는 실수}\}$$

$$\mathcal{R}=\left\{r\,|\,r=-\left(\theta^2+0.1\times\left(\frac{d\theta}{dt}\right)^2+0.001\times\tau^2\right)\right\}$$



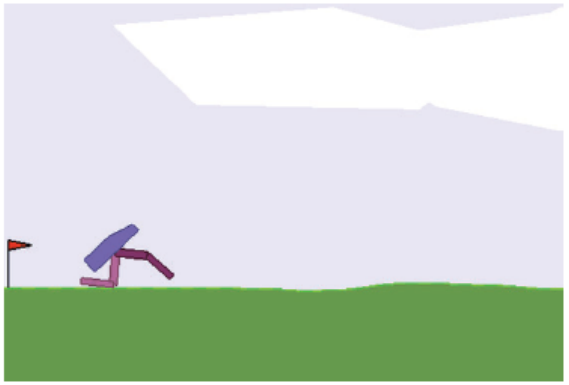
2.1.2 과업 사례

- 연속 과업: [예제 2-6] BipedalWalker
  - 무릎과 엉덩이 각각에 2개 관절을 가진 이족 보행 로봇이 넘어지지 않고 오래 걸어야 함
  - 상태는 10개 라이더 센서에서 들어오는 신호를 포함하여 위치와 속도를 나타내는 24개 변수(24차원 실수 공간)
  - 행동은 4개 관절에 가하는 회전력 (4차원 실수 공간) → 연속 과업
  - 보상은 힘을 적게 들여 멀리 갈수록 큼

$$\mathcal{S}=\{(x_1,x_2,\cdots,x_{24}) \mid x_i\text{는 실수}\}$$

$$\mathcal{A}=\{(\tau_1,\tau_2,\tau_3,\tau_4) \mid \tau_i\text{는 실수}\}$$

$$\mathcal{R}=\{r \mid r=\text{앞으로 전진한 거리}-\text{모터에 가한 힘의 합, 넘어지면 } -100\}$$

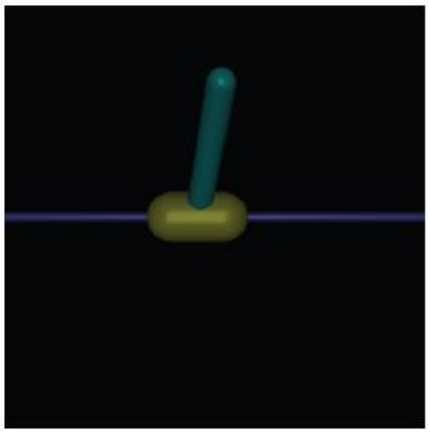


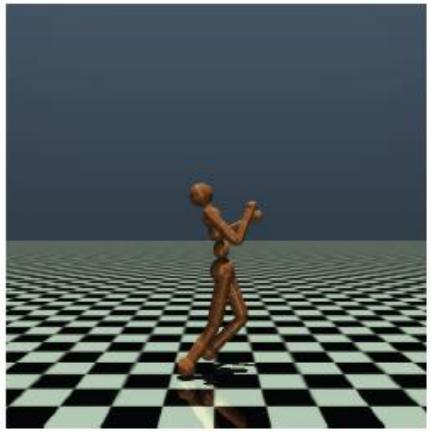
2.1.2 과업 사례

- 연속 과업: [예제 2-7] MuJoCo
  - MuJoCo는 로봇 및 동물의 신체 동작을 시뮬레이션하는 라이브러리
  - 여기서는 InvertedPendulum과 Humanoid를 예시함
  - InvertedPendulum
    - CartPole과 비슷한데, 행동은 수레에 가하는 힘으로서 실수(CartPole은 왼쪽 또는 오른쪽만 선택하기 때문에 정교한 제어가 불가능)

$$\mathcal{A}=\{\tau \mid -3\leq\tau\leq 3\text{인 실수}\}$$

  - Humanoid
    - 상태 공간은 376차원 실수 공간이고 행동 공간은 17차원 실수 공간





2.1.3 MDP의 성질

- MDP는 마르코프 성질Markov property을 만족한다고 가정
  - 마르코프 성질은 다음 순간의 상태 확률은 직전 순간만 고려한 상태 확률과 같다는 가정

$$p(s_{t+1} \mid s_0, a_0, s_1, a_1, \cdots, s_t, a_t) = p(s_{t+1} \mid s_t, a_t) \qquad (2.4)$$

  - 날씨는 마르코프 성질을 만족하지 않음
  - 틱택토와 바둑은 마르코프 성질 만족
    - 현재 상태만 보고 다음 수를 결정할 수 있음
  - 자율주행
    - 차량의 위치만 사용하면 만족 못하지만, 위치, 속도, 방향 등을 사용하면 만족함

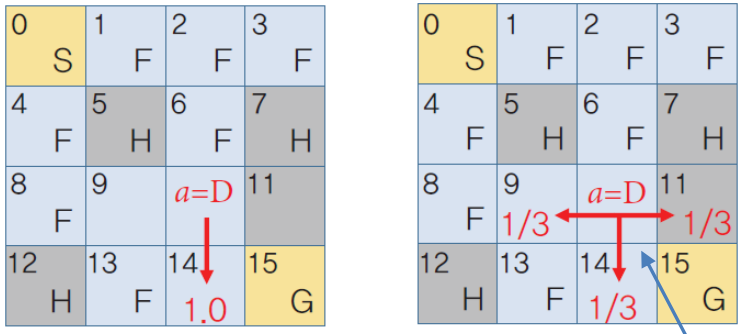
→강화 학습에서는 마르코프 성질을 만족하도록 상태를 표현하는 일이 중요

2.1.3 MDP의 성질

- 상태 전이 확률state transition propability(MDP 역학dynamics)
  - 현재 순간 t의 상태 s_t에서 행동 a_t를 취했을 때, 다음 순간 $t+1$에서 상태 s_{t+1}과 보상 r_t가 발생할 확률

$$p(s_{t+1}, r_t \mid s_t, a_t), \text{ 줄여서 쓰면 } p(s', r \mid s, a) \quad (2.5)$$

- 결정론 과업과 스토캐스틱 과업
  - 행동 a_t를 취했을 때, 하나의 상태로 전이하면 결정론이고 확률 분포에 따라 여러 상태로 전이하면 스토캐스틱 과업
  - 틱택토와 바둑은 결정론 과업
  - Gymnasium의 FrozenLake 과업은 결정론 또는 스토캐스틱 판으로 설정 가능



(a) 결정론 판 (b) 스토캐스틱 판

얼음이 미끄러워 여러 상태로 전이 가능

그림 2-5 FrozenLake 과업의 결정론 판과 스토캐스틱 판

2.1.3 MDP의 성질

- 모델 자유와 모델 기반
  - MDP 역학의 확률 분포를 사용하여 학습하는 알고리즘은 모델 기반(3장의 동적 프로그래밍이 여기 해당)
  - MDP 역학이 제공하는 상태 s'와 보상 r 정보를 사용하여 학습 하는 알고리즘은 모델 자유(4장 이후 몬테카를로, Sarsa, Q-러닝, DQN, REINFORCE, TRPO, PPO, DDPG, TD3, SAC는 모델 자유)
- 환경이 처한 상황을 모두 관찰하는 경우와 일부만 관찰하는 경우
  - 틱택토와 바둑은 전자
  - 퐁 게임은 후자(후자를 partially observable MDP(POMDP)라고 부름)
  - POMDP에서 상태 표현에 주의 필요함
    - 퐁 게임의 경우 현재 화면 상태만으로는 정보가 부족하여 학습이 불가능
    - Mnih는 최근 네 장의 영상을 쌓아 상태 표현. $p(s_{t+1}, r_t|s_t, a_t)$ 대신 $p(s_{t+1}, r_t|s_{t-3}, s_{t-2}, s_{t-1}, s_t, a_t)$를 사용하는 셈