1. 강화학습 소개 (Introduction to Reinforcement Learning)

순천향대학교 컴퓨터공학과

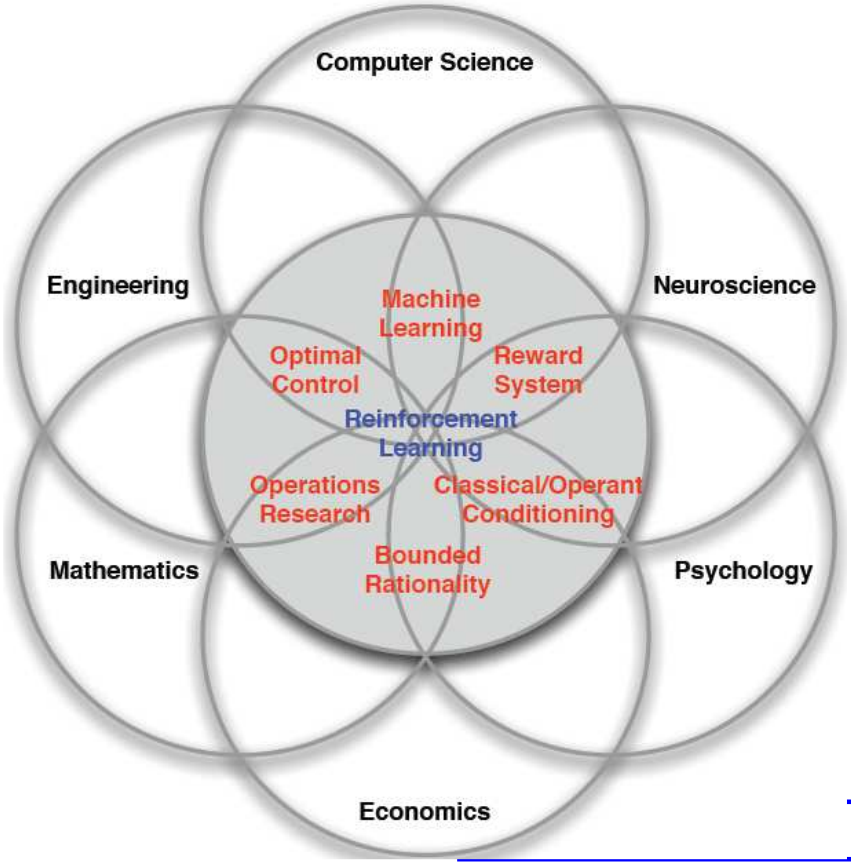
이 상 정

학습 내용

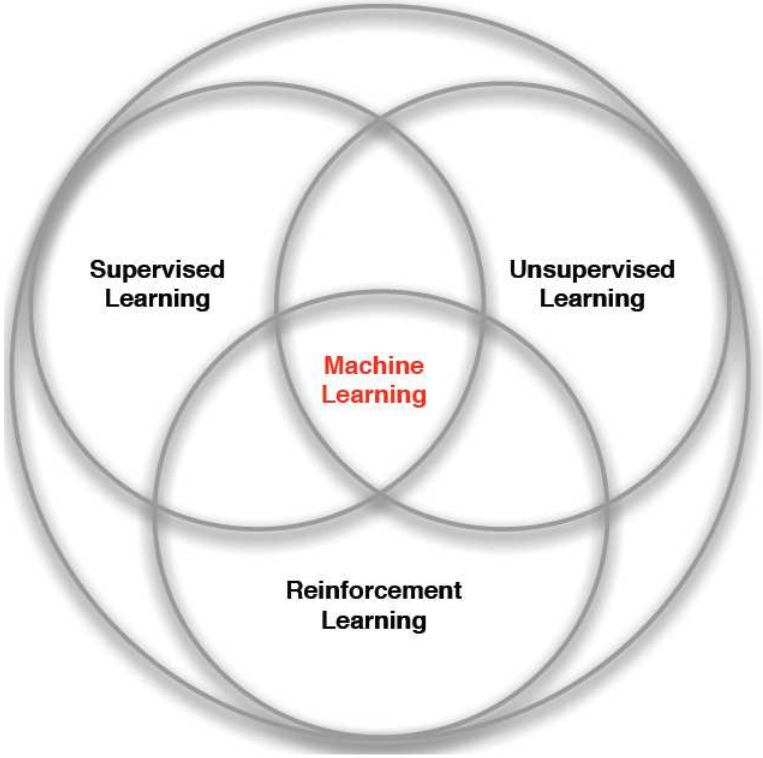
1. 강화학습이란?
2. 강화학습의 기본
3. 강화학습 에이전트
4. 강화학습의 기본 문제

1. 강화학습이란?

강화학습의 다양한 측면

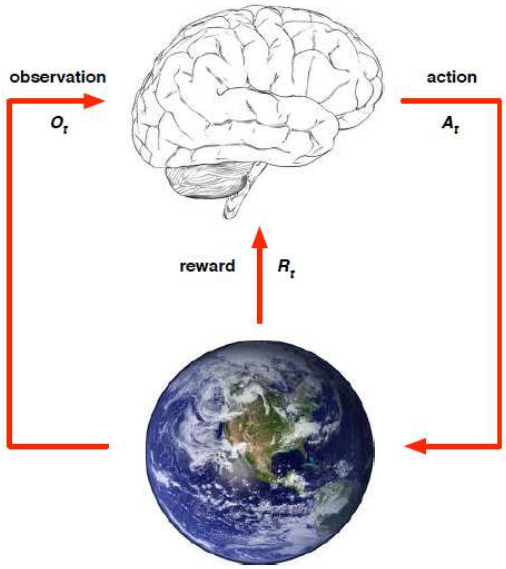


머신러닝의 분야



강화학습의 주요 개념

- **환경과의 상호작용을 통한 비지도 학습**
  - 환경(environment)을 탐색하는 에이전트(agent)가 현재의 상태(state)를 관찰(observation)하여 어떤 행동(action)을 수행
  - 에이전트는 특정 행동을 수행 후 환경으로부터 보상(reward)을 얻게 됨
    - 선택 행동의 보상의 결과(좋은/나쁜 결과인지)는 즉시 또는 먼 미래에 제공
  - 에이전트가 앞으로 누적될 보상을 최대화하는 일련의 행동으로 정의되는 정책(policy)을 탐색



강화학습 응용 예

- 체스, 바둑, 보드 게임 등
- 자율주행
- 휴머노이드 로봇
- 모형 헬리콥터 조종
- 아타리 게임
- 투자 포트폴리오 관리

.....

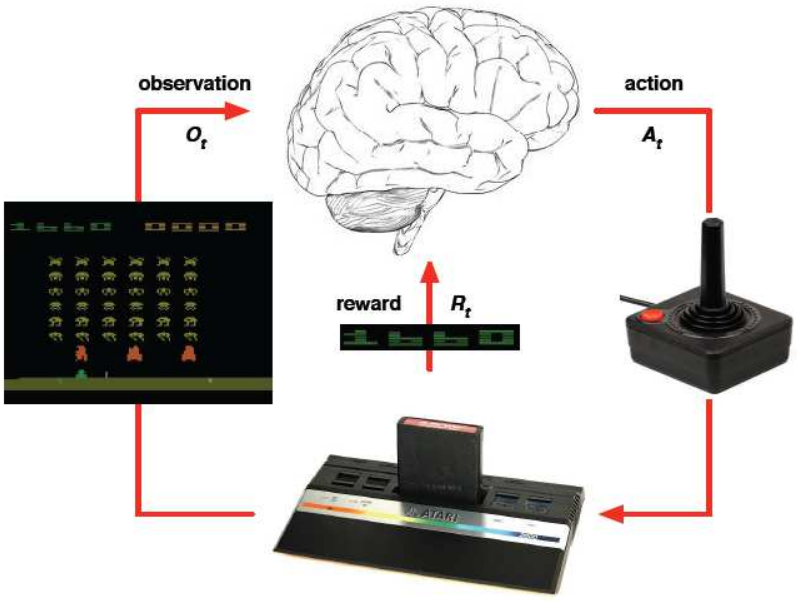
AlphaGo

1000 에피소드 학습 후

3000 에피소드 학습 후

5000 에피소드 학습 후

아타리 게임 (Atari Game) 예



- 게임을 규칙을 모름
- 게임을 수행하는 상호 작용으로 직접 학습
- 조이스틱에서의 행동을 선택하고, 픽셀과 점수를 관찰

2. 강화학습의 기본

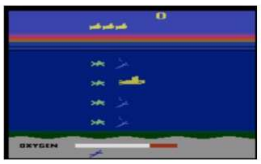
보상 (Reward)

- 특정 행동을 선택한 결과가 보상(reward) R_t로 제공
  - 스칼라 숫자 값으로 제공
    - 좋은 결과이면 양의 값, 나쁜 결과이면 음의 값
  - 스텝 t 에서 에이전트가 선택한 행동이 얼마나 좋은지를 나타냄
  - 에이전트는 누적되는 보상(cumulative reward)을 최대화하는 것을 목표로 함

- 강화학습은 아래와 같은 보상의 가설(Reward Hypothesis)에 기반
  - 모든 목표는 기대되는 누적된 보상 (expected cumulative reward)을 최대화하는 것으로 기술될 수 있다.

보상 예

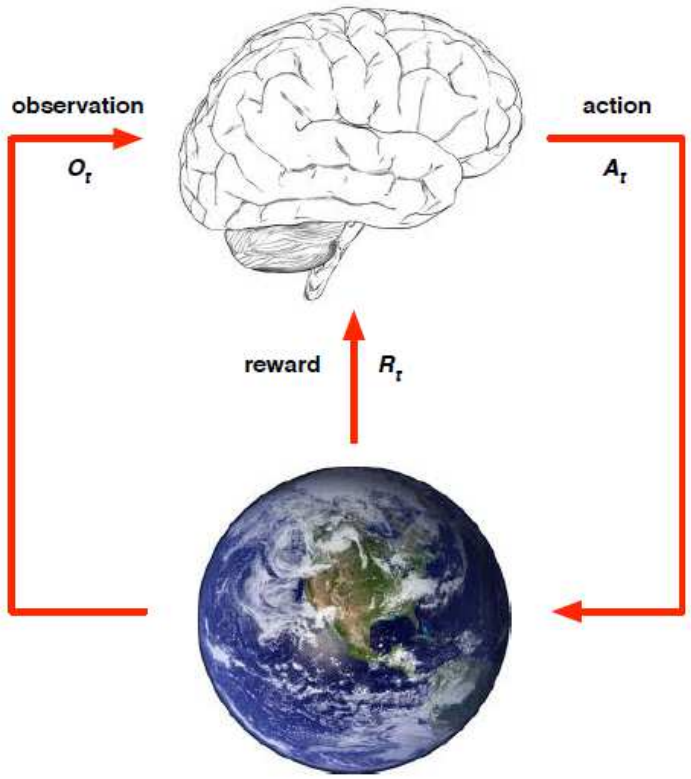
- 모형 헬리콥터 조종
  - 의도된 궤적을 따를 때 +ve 보상
  - 추락 시 –ve
- 바둑 게임
  - 승리/패배 시 +/- 보상
- 휴머노이드 로봇 걷기 제어
  - 앞으로 걸을 때 +ve 보상
  - 넘어질 때 –ve 보상
- 아타리 게임
  - 점수가 증가/감소 시 +/- 보상
- 투자 포트폴리오 관리
  - 은행에 돈이 증가할 때마다 +ve



순차적인 의사 결정 (Sequetial Decision Making)

- 강화학습은 미래에 받게될 모든 보상의 총합 (누적된 보상)을 최대화를 목표로 순차적으로 의사를 결정
- 현재의 선택된 행동이 장기적인 관점의 먼 미래에 영향
  - 보상의 결과가 즉시 나타나지 않고 지연
  - 현재의 즉시 발생하는 보상을 희생하고 장기적인 보상의 관점에서 행동을 선택하는 것이 올바를 수 있음

에이전트와 환경



- **행동**의 주체 **에이전트**는 특정 **환경** 속에서 존재
- 시간 t에서 **에이전트**
  - 행동(A_t) 수행
  - 환경을 관찰(O_t)
  - 보상(R_t)을 받음
- **환경**
  - 행동(A_t)을 받음
  - 관찰(O_t)을 내보냄
  - 보상(R_t) 을 내보냄

히스토리와 상태 (History and State)

- **히스토리(history)**는 **관찰(observations), 행동(actions), 보상(rewards)**의 시퀀스

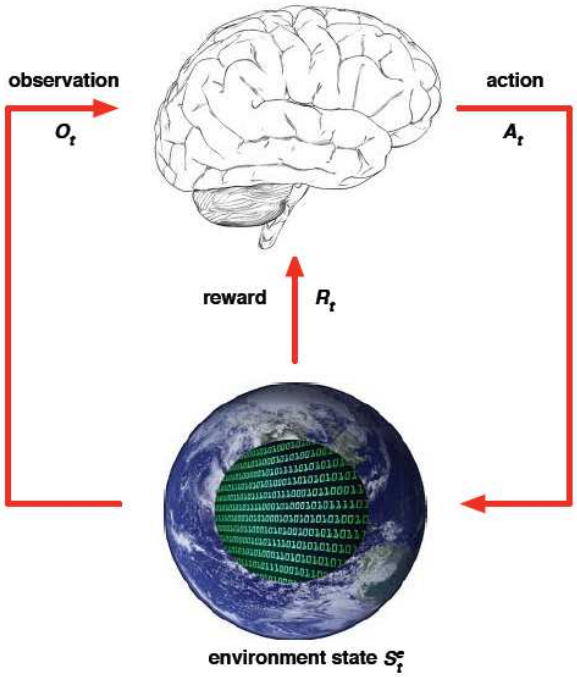
$$H_t = O_1, R_1, A_1, \ldots, A_{t-1}, O_t, R_t$$

  - 시간 t까지 관측 가능한 모든 변수들
  - 히스토리에 의존하여 다음 상황 선택
    - **에이전트**는 **행동**을 선택
    - **환경**은 **관찰/보상**을 선택

- **상태(state)**는 다음 상황을 결정하기 위해 사용되는 정보
  - 상태는 히스토리의 함수

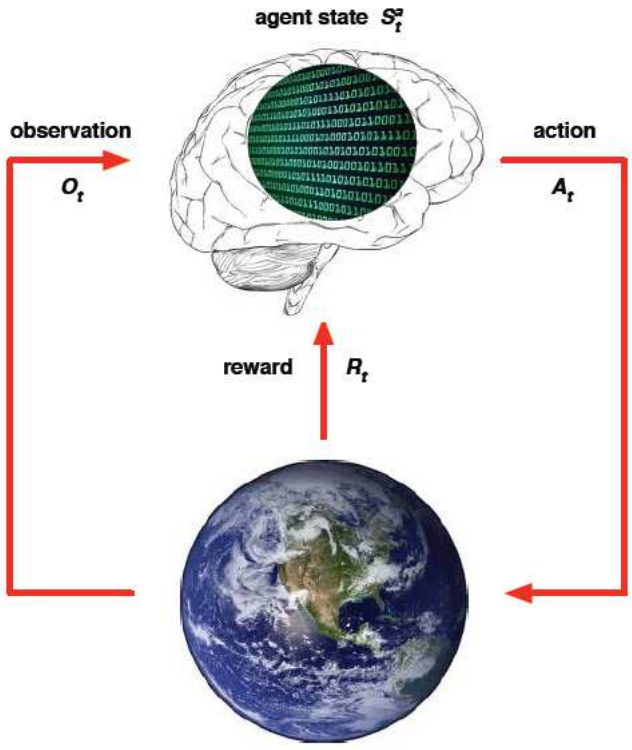
$$S_t = f(H_t)$$

환경의 상태 (Environment State)



- 환경의 상태 S_t^e 는 환경의 모든 정보를 표현
  - 환경이 다음의 관찰/보상을 선택하기 위해 사용하는 정보
- 일반적으로 에이전트는 모든 환경에 대한 정보를 볼 수 없음
  - 에이전트가 환경의 정보를 볼 수 있더라도 좋은 행동을 선택하는 데 무관한 정보

에이전트의 상태 (Agent State)



- 에이전트의 상태 S_t^a 는 에이전트의 내부를 표현
  - 에이전트가 다음의 행동을 선택하기 위해 사용하는 정보
- 강화학습 알고리즘이 사용하는 정보
  - 히스토리의 함수

$$S_t^a = f(H_t)$$

정보 상태 (Information State)

- 정보 상태 (마르코프 상태 라고도 함)는 히스토리의 모든 유용한 정보를 포함

Definition

A state S_t is Markov if and only if

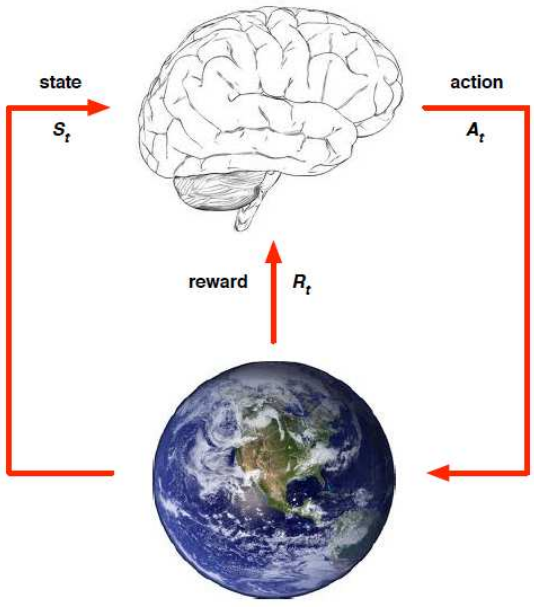
$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, \ldots, S_t]$$

- 미래는 현재 상태 이전의 과거에 독립적

$$H_{1:t} \rightarrow S_t \rightarrow H_{t+1:\infty}$$

  - 현재 주어진 상태는 과거의 모든 히스토리 정보를 포함하고 있기 때문에 현재의 정보가 중요하며, 과거의 정보들은 의미가 없음
  - 미래는 확정된 것이 아니기 때문에 확률적으로 접근

완전한 관찰 환경 (Fully Observable Environments)



- 완전한 관찰성 (full observability)
  - 에이전트가 직접적으로 환경의 모든 정보를 알 수 있음

$$O_t = S_t^a = S_t^e$$

  - 에이전트의 상태 = 환경의 상태 = 정보 상태
  - 이를 마르코프 의사결정 과정 (Markov Decision Process, MDP) 이라고 함

부분 관찰 환경 (Partially Observable Environmants)

- **부분 관찰성 (partial observility)**
  - 에이전트가 환경에서 간접적으로 정보를 관찰
    - 로봇의 카메라는 절대 위치를 알려주지 않음
    - 주식 거래자는 현재의 가격만을 관찰
    - 포커 플레이어는 공개된 카드만을 관찰
  - 에이전트의 상태 ≠ 환경의 상태
  - 이를 부분 관찰 마르코프 의사 결정 프로세스 (Partially Observable Markov Decision Process, POMDP) 라고 함
- **에이전트가 현재 시점의 상태를 구축해야함**
  - 히스토리 전체를 사용

$$S_t^a = H_t$$

  - 과거 발생한 환경의 상태 확률에 기반

$$S_t^a = (\mathbb{P}[S_t^e = s^1], \ldots, \mathbb{P}[S_t^e = s^n])$$

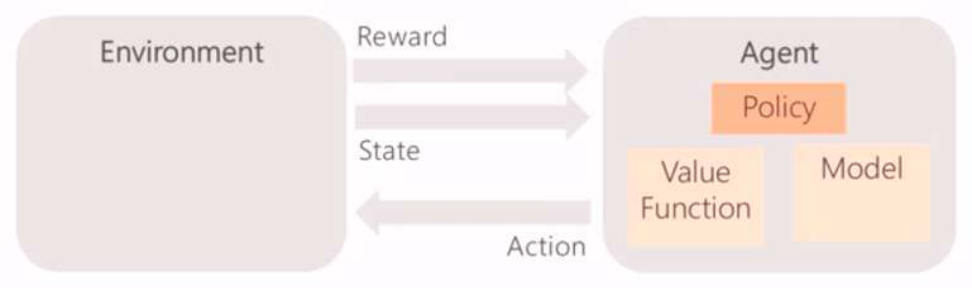
  - 머신러닝 RNN

$$S_t^a = \sigma(S_{t-1}^a W_s + O_t W_o)$$

3. 강화학습 에이전트

에이전트 주요 구성 요소

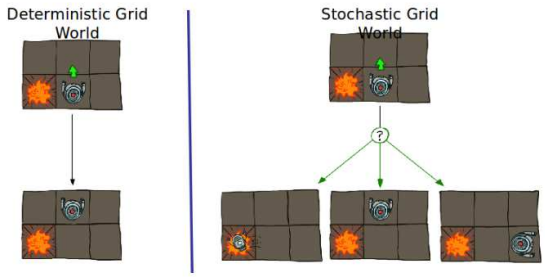
- 강화학습 에이전트의 주요 구성 요소
  - 정책 (Policy)
    - 특정 상태에서 에이전트의 행동을 결정
  - 가치함수 (Value Function)
    - 각 상태와 행동의 가치를 평가
  - 모델 (Model)
    - 행동 후 환경의 전체적인 형태를 기술



정책 (Policy)

- 정책 (Policy)은 특정 상태에서 에이전트의 행동을 결정
  - 최적의 정책을 탐색하는 것이 강화학습의 목표
- 정책은 상태로부터 행동을 매핑
  - 결정론적 정책 (deterministic policy)
    - 정형화된 규칙에 의해 행동을 결정
    - 같은 상태에서는 항상 같은 행동으로 결정

$$a = \pi(s)$$



  - 확률론적 정책 (stochastic policy)
    - 확률적으로 행동을 결정
    - 같은 상태에서 항상 같은 행동을 결정하지는 않음

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

가치함수 (Value Function)

- 가치함수 (Value Function)는 각 상태와 행동의 가치를 평가
  - 가치함수는 미래의 보상을 예측하여 각 상태 좋음과 나쁨을 평가
  - 상태-가치함수(State Value Function)은 상태만을 고려하여 보상을 계산
    - 할인율을 고려한 전체 보상을 계산

$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \mid S_t = s \right]$$

  - 상태와 행동까지 모두 고려하는 경우는 행동-가치함수(Action Value Function) 이라고 함

할인율 (Discount Factor)

- 현재 얻게 되는 보상이 미래에 얻게 될 보상보다 얼마나 더 중요한지를 나타내는 값으로 0과 1사이의 값

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

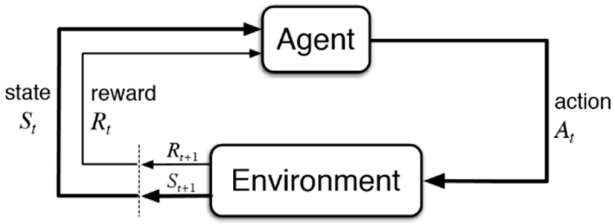
  - 스텝 t에서 미래를 포함한 전체 보상

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

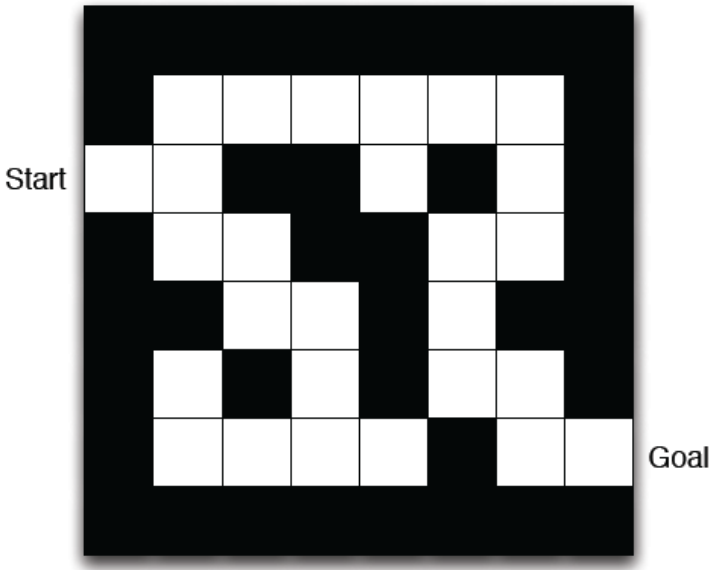
모델 (Model)

- 모델 (Model)은 행동 후 환경의 전체적인 형태를 기술
  - 환경이 다음에 무엇을 할 것인지를 예측
  - 다음 상태, 보상 등을 예측

$$\mathcal{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$$
$$\mathcal{R}^a_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$$

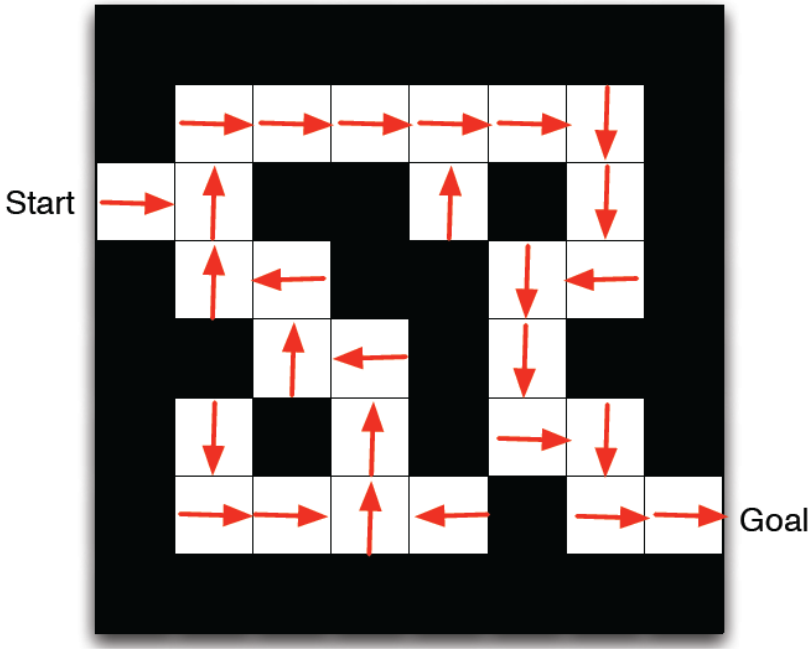


미로(Maze) 예



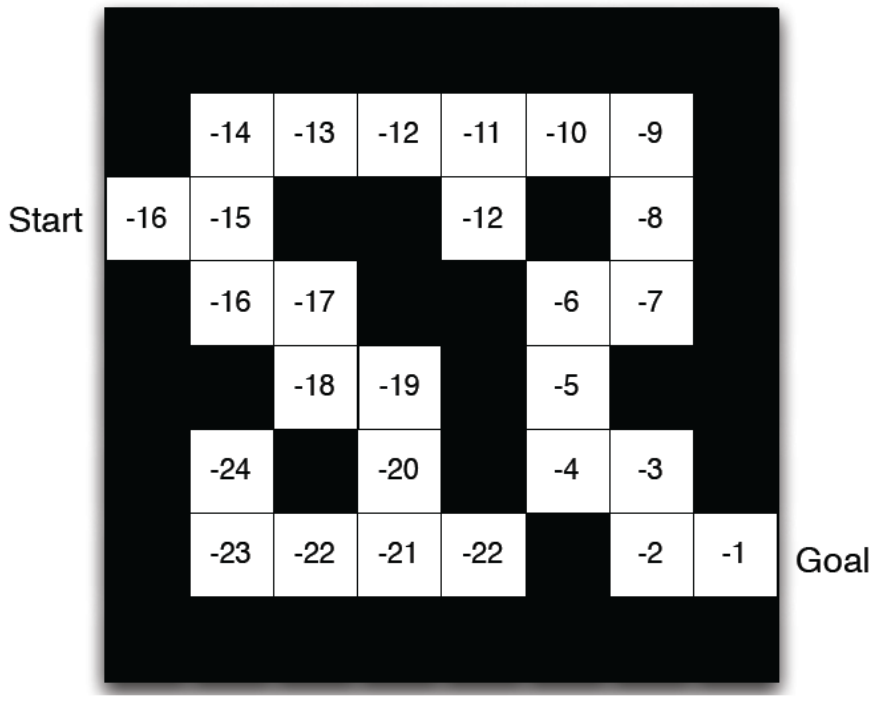
- 보상 (Reward)
  - 시간 스텝 당 -1
- 행동 (Actions)
  - N, E, S W
- 상태 (States)
  - 에이전트의 위치

미로 예: 정책 (Policy)



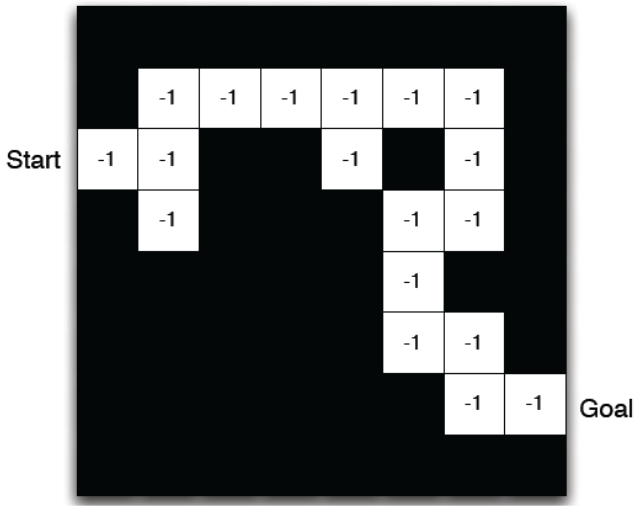
- 화살표가 각 상태 s에서 **정책** $\pi(s)$ 를 나타냄

미로 예: 가치함수



- 숫자가 각 상태 s에서 **가치** $v_\pi(s)$ 를 나타냄

미로 예: 모델 (Model)

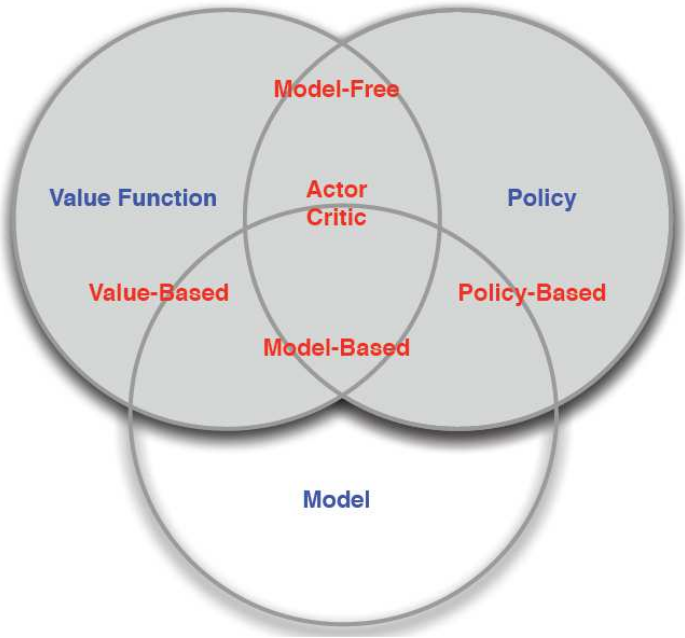


- 에이전트는 환경의 내부 모델을 갖고 있을 수도 있음
  - 다이내믹 (dynamics)
    - 행동이 상태를 어떻게 변경하는가?
  - 보상 (rewards)
    - 각 상태에서의 보상은 얼마인가?
- 모델이 불완전할 수 있음

- 그리드 배치는 전이 모델(transitiomn model) $\mathcal{P}^a_{ss'}$ 를 나타냄
- 숫자는 각 상태 s에서의 직접 보상(immediate reward) $\mathcal{R}^a_s$ 를 나타냄 (이 예에서는 모든 행동 a에 대해 동일)

강화학습 에이전트의 종류

- 정책(policy), 가치함수(value function) 기반 분류
  - 가치 기반형 (Value based) 에이전트
    - 정책 없음
    - 가치함수 (value function) 사용
  - 정책 기반 (Policy based) 에이전트
    - 정책 사용
    - 가치함수 없음
  - Actor Critic 에이전트
    - 정책 사용
    - 가치함수 사용
- 모델(model) 기반 분류
  - 비모델 (Model free) 에이전트
    - 정책 and/or 가치함수
    - 모델 없음
  - 모델 기반 (Model based) 에이전트
    - 정책 and/or 가치함수
    - 모델



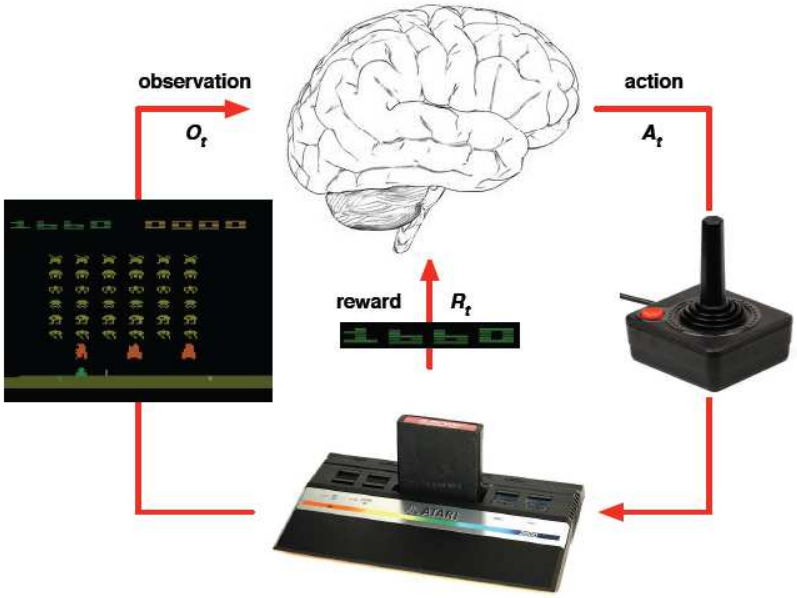
4. 강화학습의 기본 문제

학습과 계획

- 순차 의사 결정(sequential decision making)에는 강화학습과 계획의 두 가지 방식

- 강화학습 (Reinforcement Learning)
  - 초기에 환경에 대해 알지 못함
  - 에이전트는 환경과 상호작용을 통해 환경을 파악
  - 에이전트는 정책을 생성하고 개선
- 계획 (Planning)
  - 환경의 모델이 알려짐, 모델 기반 에이전트
  - 에이전트 (상호작용 없이) 모델을 가지고 계산을 수행
  - 에이전트는 정책을 개선
  - 동적 계획법 (dynamic programming)이 이에 해당

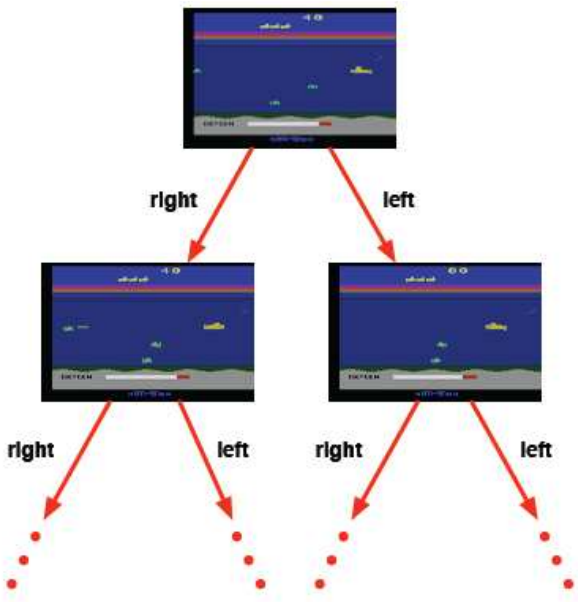
아타리 게임 (Atari Game) 예: 강화학습



- 게임을 규칙을 모름
- 게임을 수행하는 상호작용으로 직접 학습
- 조이스틱에서의 행동을 선택하고, 픽셀과 점수를 관찰

아타리 게임 (Atari Game) 예: 계획법

- 게임의 규칙이 알려짐
- 에이전트가 모델을 완전히 파악하여 에뮬레이터에게 질의 가능
  - 상태 s에서 행동 a를 수행하면:
    - 다음 상태는?
    - 점수는?
- 최적의 정책을 찾으려고 계획
  - 예, 트리 검색



강화학습의 탐색과 활용

- 탐색(exploration)은 환경에 대한 정보를 파악하는 과정
- 활용(exploitation)은 보상을 최대화하기 위해 알려진 정보를 이용하는 과정
- 레스토랑 선택 예
  - 활용: 자신이 선호하는 레스토랑 선택
  - 탐색: 새로운 레스토랑 시도

- 탐색-활용 딜레마 (exploration-exploitation dilemma)
  - 이미 알고 있는 정보만 이용하면 새로운 정보를 찾게 될 기회를 놓쳐서 더 좋은 정책으로 개선할 기회를 얻지 못함
- 탐색과 활용을 적절하게 혼용해서 최적의 정책을 결정해야 함

예측과 제어

- 예측(prediction)은 주어진 정책을 사용하여 미래의 결과를 평가하고 행동하는 것

- 제어(control)는 가장 최적의 정책을 찾기 위해 최적화하는 것

- 제어 문제 해결을 위해서는 먼저 예측 문제의 해결이 필요

참고 자료

- Richard Sutton, An Introduction to Reinforcement Learning, 2017
  - http://incompleteideas.net/book/bookdraft2017nov5.pdf
  - 단단한 강화학습, 김성우 옮김, 제이펍
- David Silver - UCL Course on RL, 2015
  - http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
  - Lecture 1: Introduction to Reinforcement learning
- 기타
  - RL (강화학습) 기초 – 2. Reinforcement Learning 소개
    - http://daeson.tistory.com/m/312?category=710652
  - 모두의연구소 - 강화학습 그리고 OpenAI
    - http://www.modulabs.co.kr/RL_library/1705
  - RL - Introduction to Deep Reinforcement Learning
    - https://medium.com/@jonathan_hui/rl-introduction-to-deep-reinforcement-learning-35c25e04c199