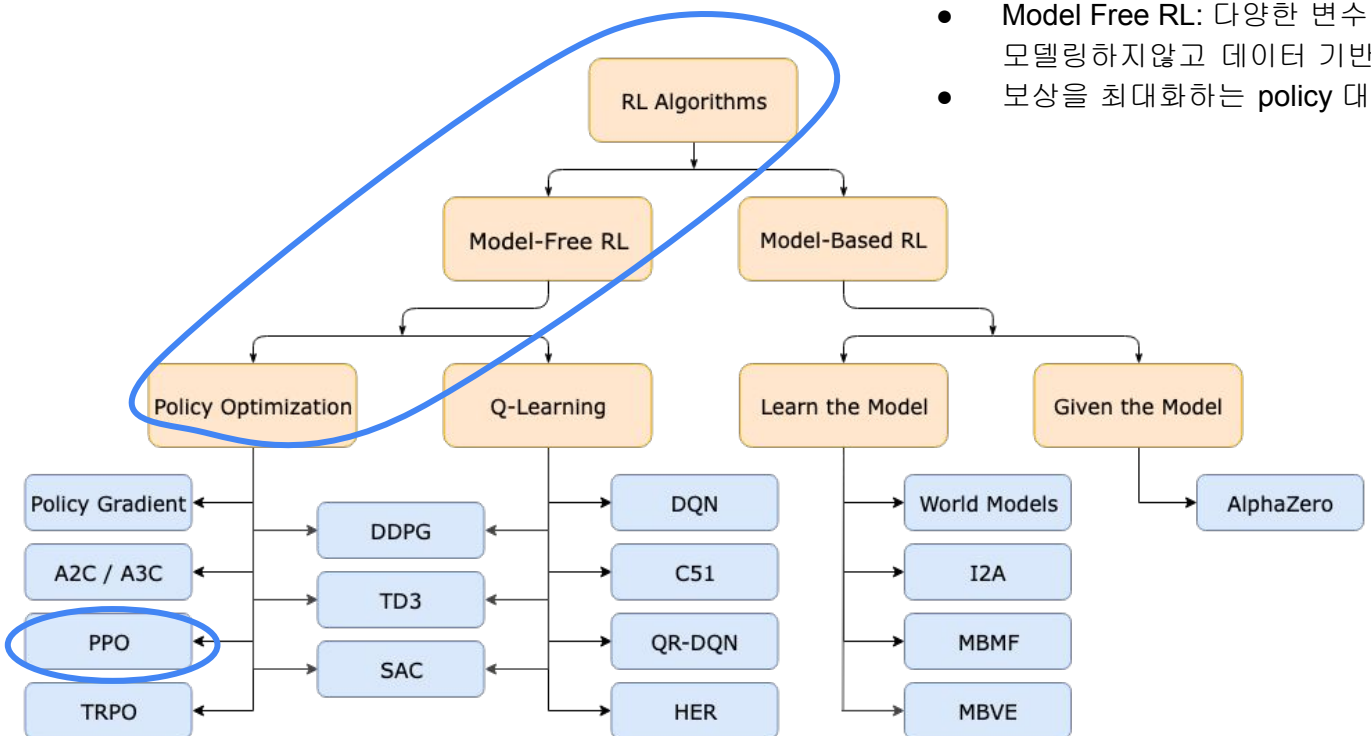
MMORPG RL 개념과 설계 방법 구상

Joanne

강화학습 모델링 설계 방법

RL(reinforcement learning) 방법론 분류

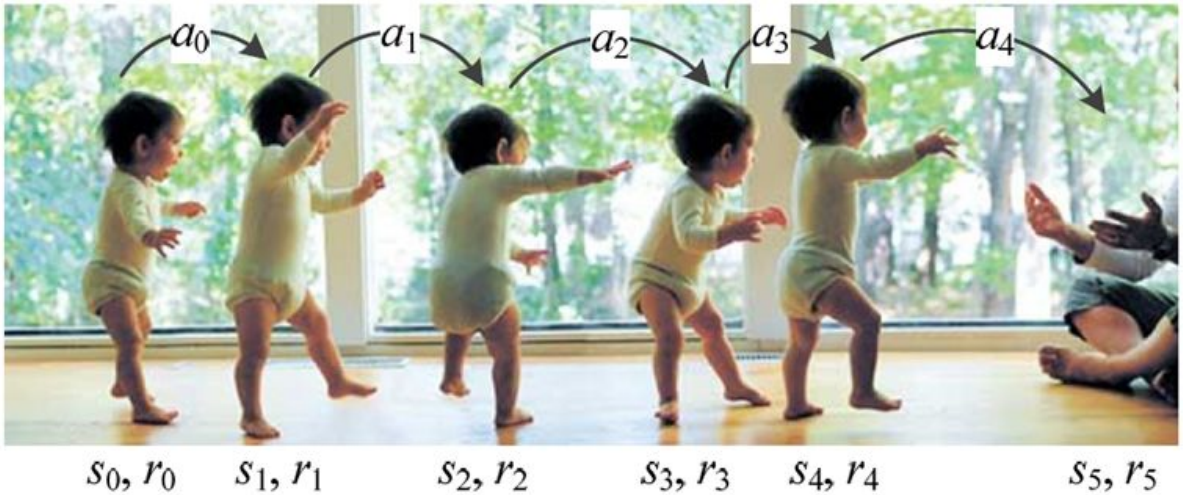


- Model Free RL: 다양한 변수 및 예기치 못한 상황들에 대해 일일이 모델링하지않고 데이터 기반 의사결정
- 보상을 최대화하는 policy 대로 반영하여 업데이트

강화학습 모델링 **MDP(markov decision process)** 설계 방법

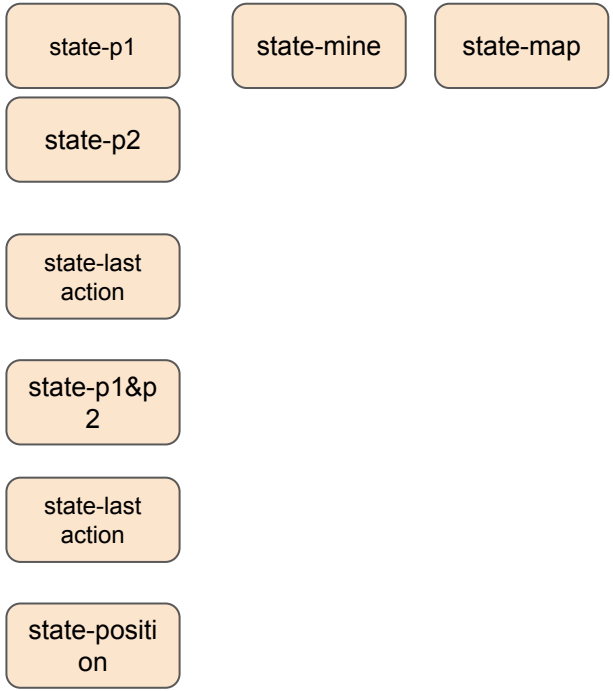
강화학습 문제 해결 방식

MDP 기본 구성: 상태(state), 행동(action), 보상(reward)



[그림 9-2] 강화 학습의 핵심 개념인 상태 s_i, 행동 a_i, 보상 r_i

데이터 (상태)



a. 플레이어 데이터:

- **스탯**: 아이템, 매직 포인트(MP), 체력(HP), 공격력(Damage).
- **위치 및 분포**: 플레이어들이 보스 주변에 클러스터 형태로 모여 있는지 여부
- **행동**: 가장 최근 공격 방식 및 stats

b. 보스 데이터:

- **스탯**: 체력(HP), 사용 가능한 스킬, 스킬 쿨타임, 에너지(MP)
- **최근 전투 결과**: 보스가 받은 피해량 및 성공적인 공격 수
- **현재 상황**: 전투 진행 시간(초기, 중반, 후반)

c. 환경 데이터:

- **맵 요소**: 장애물 유무, 던전 크기.
- **플레이어 위치**: 전체 분포 및 개별 플레이어 간 거리.

데이터 (상태)

state-p1

state-p2

state-last action

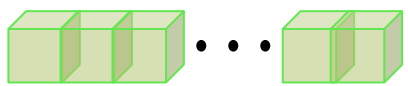
state-p1&p2

state-last action

state-position

주변 상황을 나타내는 피처

Player1 자신 stats



p1 자기 주변 상황에 대한 묘사

주변 상황을 나타내는 피처

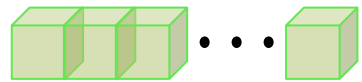
Player2 자신 stats

p2 자기 주변 상황에 대한 묘사

state-boss

주변 상황을 나타내는 피처

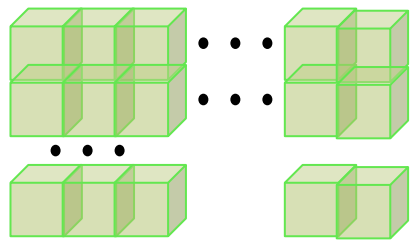
Player1 자신 stats



Boss 자기 주변 상황에 대한 묘사

주변 상황을 나타내는 피처

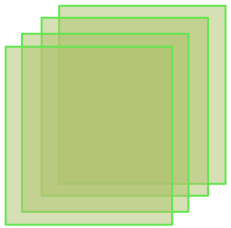
boss 주변 player



Boss와 플레이어 주변 상황에 대한 묘사

state-map

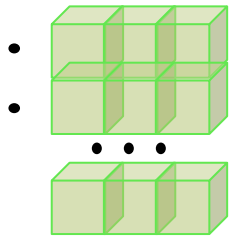
주변상황에 대한 맵 채널



상태와 행동

세부 항목 종합적으로 고려

입력 신경망



state-mine

state-p1

state-p2

state-last action

state-p1&p2

state-last action

state-map

encoder network

encoder network

encoder network

encoder network

encoder network

encoder network

계층적 행동 선택

행동 선택기
(공격 or 방어?)

분류

공격 skill 선택기

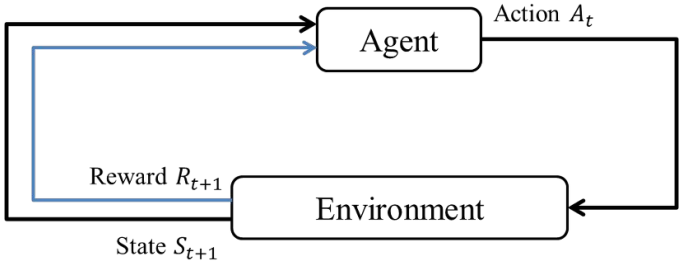
공격 대상 플레이어 선택기

이동 위치/거리 선택기

출력 신경망
(행동으로 변환)

보상

출력 행동 변환



이미지 출처: Sunilkumar, Abishek & Bahrpeyma, Fouad & Reichelt, Dirk. (2024). An overview of the applications of reinforcement learning to robot programming: discussion on the literature and the potentials. 10.33968/2024.54.

$$E\left[\sum_{t=0}^{\infty}\gamma^t R_{a_t}(s_t, s_{t+1})\right]$$

Reward R(x) = **α** *r_1(x) + **β***r_2(x) + **γ** *r_3(x) +... **ω***r_n(x)

보상(R) 체계 (예시)

- r_1(x): Boss가 일정 시간 안에 플레이어를 공격할 수 있도록 유도
- r_2(x): 가장 최근에 자기를 공격한 플레이어를 제거 하도록 유도
- r_3(x): 특정 시간 주기적으로 알/몬스터를 소환
- r_4(x): Boss의 체력을 일정 수준 이상으로 유지
- r_5(x): 플레이어 거리가 아주 멀지 않게 유지
- **α, β,γ,ω :** 학습 결과 기반 튜닝이 필요한 가중치

확률적으로 최선에 가까운 행동 출력

$$P_{ss'} = Pr(S_{t+1}=s' \mid S_t=s) \longrightarrow P_{ss'}^{a} = Pr(S_{t+1}=s' \mid S_t=s, A_t=a)$$

Policy와 Transition probability를 다음과 같이 구분

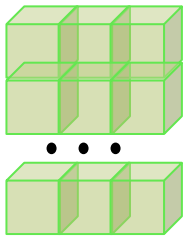
- State(s) : 상태값
- Policy : state s에서 action a를 할 확률
- Transition probability : state s에서 action a를 해서

state s'로 전이할 확률 (i.e. 상태 s에서 s'으로 전이될 확률

장점: 안정적이다. 정답이 불분명하고 데이터가 다변하는 환경에서 안정적인 학습 결과 기대 가능

단점: 학습시간이 오래걸린다.. (매우 많은 데이터를 오랫동안 학습해야함)

강화학습 입출력 구조



state-mine

state-p1

state-p2

state-last action

state-p1&p 2

state-last action

state-map

encoder network

encoder network

encoder network

encoder network

encoder network

encoder network

계층적 행동 선택

행동 선택기 (공격 or 방어?)

공격

공격 skill 선택기

공격 대상 P 선택기

이동 위치/거리 선택기

출력 행동 변환