

이론부터 실전까지 AI에이전트 완벽 마스터

-8장 강화학습과 에이전트-

이 하 늘(haneul@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목차

- 강화학습 소개
- 심층 강화학습
- LLM과 강화학습 모델의 상호작용

강화학습 소개

- 개요

- 지능형 에이전트가 환경 내에서 최적의 행동을 찾을 수 있도록 보상을 이용해 학습하는 방법

- 주요 요소 (1/2)

- 에이전트 (agent)

- 환경과 상호작용하며 목표를 달성하기 위해 최적의 의사결정을 내리는 주체

- e.g., 슈퍼마리오 게임: 마리오

- 환경 (environment)

- 에이전트의 행동에 반응하여 보상과 새로운 상태를 제공하는 외부 시스템

- e.g., 슈퍼마리오 게임: 슈퍼마리오 게임 자체

강화학습 소개

- 주요 요소 (2/2)

- 상태 (state, S_t)

- 특정 시점의 환경 조건이나 구성으로써, 다음에 발생할 일을 예측하기 위해 활용되는 정보
 - e.g., 슈퍼마리오 게임: 플레이하는 게임 속 화면

- 행동 (action, A_t)

- 에이전트가 환경에서 수행할 수 있는 결정이나 움직임

- 보상 (reward, R_t)

- 에이전트가 행한 현재 행동이 얼마나 좋은지를 나타내는 정량화된 지표
- 에이전트가 미래에 더 나은 행동을 선택하기 위해 학습에 사용

강화학습 소개

• 주요 원리

1) 에피소드 (episode): 에이전트가 처음 상태에서 최종 상태까지 도달할 때까지의 과정

- 에이전트가 환경과 반복적으로 상호작용하며, 그 과정에서 얻는 보상을 통해 행동 방식을 학습
- 에이전트는 한 에피소드¹⁾ 내 각 행동으로부터 받는 보상의 합을 최대화하기 위해 행동 전략을 지속적으로 개선하여 더 나은 결정을 내리는 것이 목표
 - ✓ 강화학습의 궁극적인 목표는 에이전트가 시간에 따라 누적되는 총 보상을 최대화하는 정책을 찾는 것

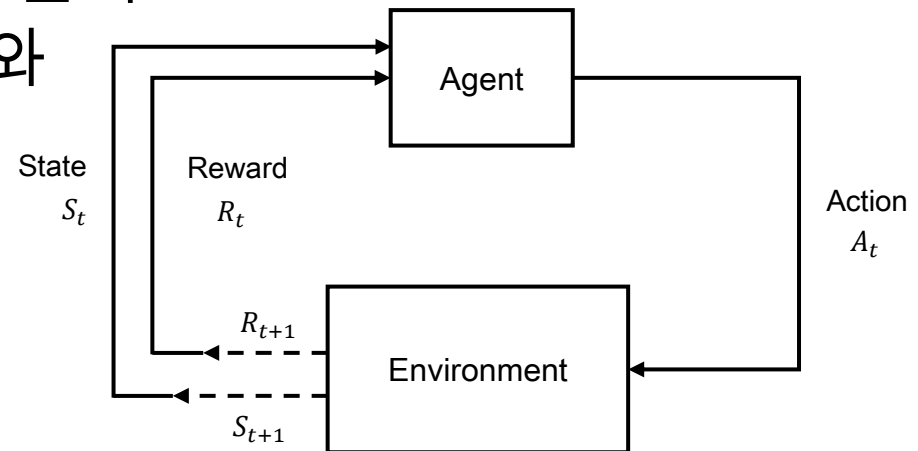
강화학습 소개

- 과정

- 행동의 결과가 다음 상태에 영향을 미치며, 이후의 행동 결정에도 영향을 줌

$$S_t \rightarrow A_t \rightarrow R_{t+1}, S_{t+1}$$

- 환경은 에이전트에게 상태 S_t 를 제공
- 에이전트는 이에 따라 행동 A_t 를 선택
- 환경은 (S_t, A_t) 에 따른 보상 R_{t+1} 와 다음 상태 S_{t+1} 을 제공



강화학습 소개

- 정책 (policy, π)
 - 주어진 상태에서 어떤 행동을 선택할지 정해주는 규칙이나 함수
- 에이전트의 정책 개선
 - 탐색(exploitation)
 - 새로운 행동을 시도해 환경에 대한 지식을 넓히는 과정
 - 학습 초반에 주로 사용
 - 탐색에 비중을 많이 두면 최적의 정책을 찾는 것이 늦어질 수 있음
 - 활용(exploration)
 - 이미 학습한 정책을 바탕으로 최적이라고 판단되는 행동을 선택하는 과정
 - 학습 후반에 주로 사용
 - 활용에 비중을 많이 두면 최적의 정책을 찾지 못할 수 있음

두 과정의 균형이 적절히 유지되는 것이 강화학습의 중요한 문제

강화학습 소개

- 멀티 암드 밴딧 문제 (Multi-Armed Bandits) (1/9)
 - 개요
 - 총 K 개 슬롯머신(Bandit)의 각 레버(Arm)가 서로 다른 성공 확률 분포를 가질 때, 일정 기간 동안 기대되는 총 보상을 최대화하는 문제
 - 어떤 레버가 가장 좋은 보상을 주는지를 학습해 해당 레버를 더 자주 선택하도록 행동을 최적화해야 함



강화학습 소개

- 멀티 암드 밴딧 문제 (2/9)

- 목표

- 각 기계의 예상 수익에 대해 더 많은 정보를 얻기 위한 “탐색”과 가장 높은 기대 수익을 갖는 기계를 “활용”하는 것 사이의 균형을 맞춰야 함

- 접근법

- 탐욕 알고리즘 (greedy algorithm)
 - 한 번씩 모두 탐색한 후, 점수가 좋은 슬롯 머신을 활용
- ϵ -탐욕 알고리즘 (ϵ -greedy algorithm)
 - 확률에 따라 랜덤으로 슬롯 머신을 활용
- 상한 신뢰 경계 (UCB, upper confidence bound)
 - 좋은 수익률을 보이며 최적의 선택이 될 가능성이 있는 슬롯 머신 탐색하여 활용

강화학습 소개

• 멀티 암드 밴딧 문제 (3/9)

• 탐욕 알고리즘 (greedy algorithm)

- 한 번씩 모두 탐색한 후, 점수가 좋은 슬롯 머신을 활용하는 알고리즘
- 시간 i 에서 추정 수익(가치) $Q_{t(a)}$ 가 가장 큰 행동을 선택

$$Q_{t(a)} = \frac{t \text{ 시점 이전까지 행동 } a \text{를 취했을 때 얻는 보상 총합}}{t \text{ 시점 이전까지 행동 } a \text{를 취한 횟수}} = \frac{\sum_{i=1}^{t-1} R_i \cdot 1_{\{A_i=a\}}}{\sum_{i=1}^{t-1} 1_{\{A_i=a\}}}$$

$$A_t = \operatorname{argmax}_a Q_t(a)$$

- $Q_{t(a)}$: 슬롯 머신 a 의 기대 수익 (가치)
 - t : 현재 플레이 횟수
 - R_i : i 번째 플레이에서 얻은 수익 (보상)
 - $A_i = a$: i 번째 플레이를 슬롯 머신 a 에서 수행
 - $1_{\{A_i=a\}}$: 시간 단계 i 에서 행동 a 가 선택되었는지 나타내는 변수
- $$1_{\{A_i=a\}} = \begin{cases} 1, & \text{시간 단계 } i \text{에서 행동 } a \text{를 선택할 때} \\ 0, & \text{그렇지 않을 때} \end{cases}$$

강화학습 소개

- 멀티 암드 밴딧 문제 (4/9)
 - 탐욕 알고리즘 (greedy algorithm)
 - 장점
 - 매우 단순하고 구현하기 쉬움
 - 단점
 - 탐색을 한 번만 함
 - 한번의 플레이(탐색) 결과가 그 슬롯 머신의 일반적인 수익률을 나타낼 수 없음

강화학습 소개

• 멀티 암드 밴딧 문제 (5/9)

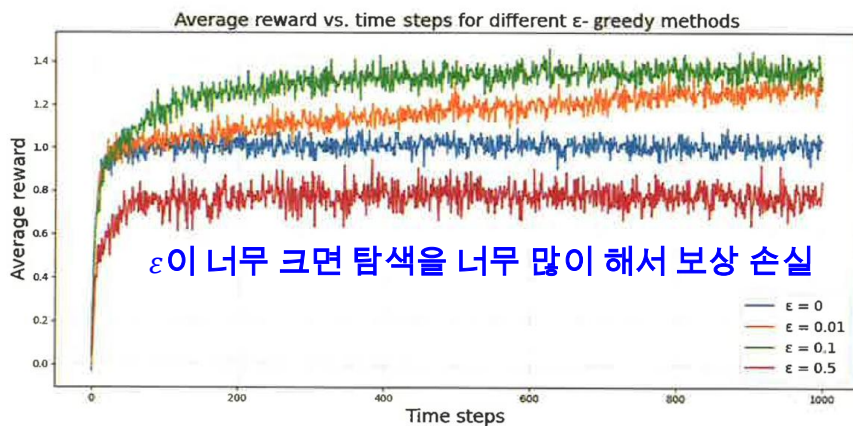
• ϵ -탐욕 (ϵ -greedy) 알고리즘

• 확률(ϵ)에 따라 랜덤으로 슬롯 머신을 활용하는 알고리즘

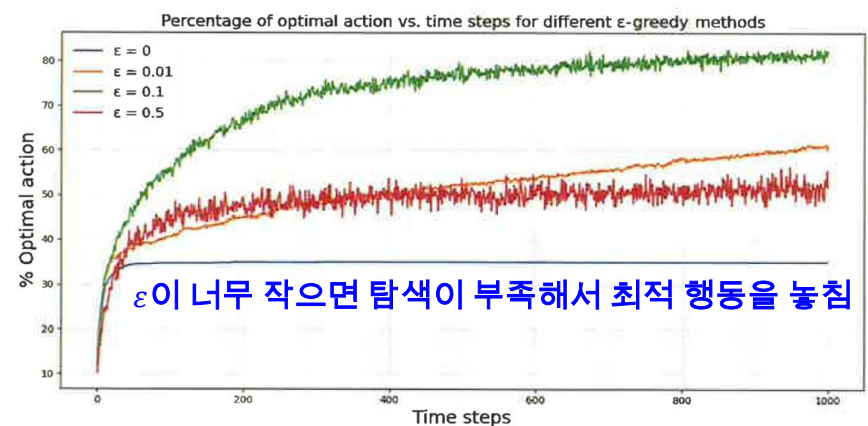
- e.g., $\epsilon = 0.1$ 이라면, 90%의 확률로 현재 가장 가치가 높은 행동을 선택(활용)하고, 10%의 확률로 아직 시도하지 않은 행동을 탐색

- $Q_t(a)$: 슬롯 머신 a 의 기대 수익 (가치)
- t : 현재 플레이 횟수
- A_t : t 번째 플레이를 슬롯 머신 a 에서 수행

$$A_t = \begin{cases} \operatorname{argmax}_a Q_t(a), & \text{with probability } 1 - \epsilon \quad \text{활용} \\ \text{a random action,} & \text{with probability } \epsilon \quad \text{탐색} \end{cases}$$



서로 다른 탐욕 방법에서 시간 단계별 평균 보상



서로 다른 탐욕 방법에서 시간 단계별 최적 행동 선택 비율

강화학습 소개

- 멀티 암드 밴딧 문제 (6/9)
 - ϵ -탐욕 (ϵ -greedy) 알고리즘
 - 장점
 - 별도의 복잡한 탐색 전략 없이 ϵ 하나로 탐색 비율을 조절
 - 환경을 탐색하면서도, 최적의 행동을 선택할 수 있도록 함
 - 단점
 - ϵ 확률로 좋지 않은 행동도 무작위로 선택
 - 행동 공간이 매우 크면 무작위 탐색으로 좋은 행동을 발견하기 어려움

• 멀티 암드 밴딧 문제 (7/9)

• 상한 신뢰 경계 (UCB, Upper Confidence Bound)

- 좋은 수익률을 보이며 최적의 선택이 될 가능성이 있는 슬롯 머신 탐색하여 활용
- 무작위성이 아닌, 각 행동의 잠재적 최적성과 불확실성을 참고

클수록 불확실성이 큰 행동을

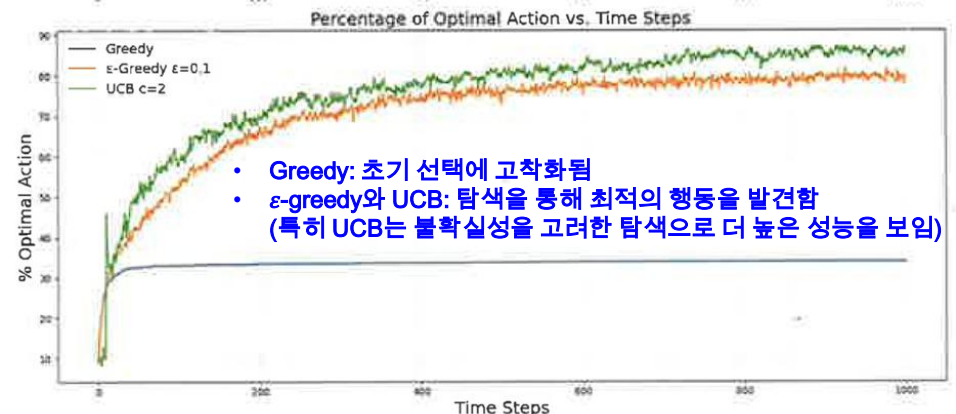
적극적으로 탐색

$$A_t = \operatorname{argmax}_a \left[Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$$

활용에 사용되는 현재까지의 추정 가치
값이 클수록 좋은 값으로 판단

탐색 보너스

많이 선택한 행동일수록 $N_t(a)$ 가 커져 탐색 보너스가 작아짐



강화학습 소개

- 멀티 암드 밴딧 문제 (8/9)
 - 상한 신뢰 경계 (UCB)
 - 장점
 - 현재 좋은 행동을 활용하고, 충분히 시도하지 않은 행동(불확실성)은 탐색해볼 수 있음
 - ϵ -greedy처럼 무작위로 탐색하지 않고, 정보가 부족한 행동 ($N_t(a)$ 이 작은 행동)을 우선적으로 탐색함
 - 단점
 - 다른 알고리즘에 비해 계산 복잡도 증가
 - 탐색 계수 c 에 대한 설정이 필요함
 - c 가 너무 크면, 탐색을 지나치게 많이 수행
 - c 가 너무 작으면, 탐색이 부족해질 수 있음
 - 가능한 행동이 많아지면 각 행동을 충분히 탐색하기 어려움

강화학습 소개

- 멀티 암드 밴딧 문제 (9/9)
- 알고리즘 비교

구분	Greedy	ϵ -Greedy	UCB
기본 원리	현재 가장 가치가 높은 행동만 선택	대부분 최선의 행동을 선택하되, 일정 확률로 무작위 탐색	보상과 불확실성을 함께 고려하여 행동 선택
행동 선택	$A_t = \arg \max_a Q_t(a)$	$1 - \epsilon$: 최선 행동 ϵ : 무작위 행동	$A_t = \arg \max_a \left[Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$
탐색 방식	탐색하지 않음	무작위 탐색	불확실성이 높은 행동을 우선 탐색
활용 방식	현재 최선의 행동에 집중	현재 최선의 행동을 주로 활용	높은 평균 보상을 가진 행동을 활용
탐색 제어	없음	ϵ 으로 조절	c 로 조절
장점	구현이 가장 간단하고 계산량이 적음	간단하게 탐색·활용의 균형 확보	무작위 탐색보다 체계적인 탐색 가능
단점	잘못된 초기 선택에 고착될 수 있음	좋지 않은 행동도 무작위로 탐색	행동 공간이 크면 계산 및 탐색 부담 증가
적합한 상황	이미 최적 행동을 충분히 알고 있는 경우	간단한 환경 및 기본적인 탐색이 필요한 경우	행동별 불확실성을 고려한 효율적인 탐색이 필요한 경우

강화학습 소개

- 마르코프 결정 과정 (MDP, Markov Decision Process) (1/9)

- 개요

- 에이전트가 현재 상태를 관찰하고 행동을 선택하면, 그 행동의 결과로 보상을 받고 다음 상태로 이동하는 의사결정 과정

- 요소

- 상태 (S): 가능한 모든 상태의 집합
- 행동 (A): 에이전트가 취할 수 있는 모든 행동의 집합
- 보상 (R): 현재 (S, A)쌍에 대한 보상의 분포
- 전이확률 (P): 현재 (S, A)쌍에 대한 다음 상태의 분포
- 할인 계수 (γ): 미래의 보상을 현재 가치로 환산할 때 사용하는 비율 (0~1)
 - 0에 가까울수록 당장의 보상에 더 큰 가치를 매겨 근시안적인 판단을 함
 - 1에 가까울수록 미래의 보상에 큰 가치를 매겨 미래지향적인 판단을 함

강화학습 소개

• 마르코프 결정 과정 (MDP) (2/9)

• 전이확률 (P)

• 필요성

- 실제 환경의 동역학을 수학적으로 모델링
- 불확실성이 있는 환경에서 행동의 결과 예측
- 에이전트가 취할 수 있는 최적의 전략 수립의 근거

- 동역학: 움직이는 물체에 작용하는 힘과 그에 따른 운동의 관계를 다루는 분야
- 대문자(S_t, A_t, R_t): 특정 값이 정해지지 않은 확률변수
- 소문자(s, a, r): 확률변수가 가질 수 있는 구체적인 값

• 상태 전이 확률

- 현재 상태 s 에서 행동 a 를 선택했을 때, 다음 상태가 s' 이 될 확률

$$p(s' | s, a) = P[S_{t+1} = s' | S_t = s, A_t = a] = \sum_{r \in R} p(s', r | s, a)$$

• 상태-보상 전이 확률

- 현재 상태 s 에서 행동 a 를 선택했을 때, 다음 상태가 s' 이 되고 보상 r 을 얻을 확률

$$p(s', r | s, a) = P[S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a]$$

강화학습 소개

• 마르코프 결정 과정 (MDP) (Markov Decision Process)

- γ : 할인 계수
- 대문자(S_t, A_t, R_t): 특정 값이 정해지지 않은 확률변수
- 소문자(s, a, r): 확률변수가 가질 수 있는 구체적인 값

• 보상 (R)

• 상태-행동 쌍에 대한 기대 보상

- 특정 상태 s 에서 행동 a 를 했을 때 다음에 받을 보상 R_{t+1} 의 기대값

$$r(s, a) = E[R_{t+1} | S_t = s, A_t = a] = \sum_{r \in R} \sum_{s' \in S} r p(s', r | s, a)$$

가능한 모든 다음 상태 s' 과 가능한 모든 보상 r 에 대해서 r 에 해당 확률을 곱한 뒤 합산

• 상태-행동-다음 상태(삼중항) 기대 보상

- 특정 상태 s 에서 행동 a 를 해서 s' 이 되었을 때 받을 보상 R_{t+1} 의 기대값

$$r(s, a, s') = E[R_{t+1} | S_t = s, A_t = a, S_{t+1} = s'] = \frac{\sum_{r \in R} r p(s', r | s, a)}{p(s' | s, a)}$$

상태 s 에서 s' 이 되면서 보상 r 을 받는 모든 경우의 기대값
상태 s 에서 행동 a 를 취해 s' 이 될 확률

• 누적 할인 보상 (G_t)

- 상태 S_t 에서 시작하여 종료 시점 T 까지 미래에 얻는 보상을 할인하여 합산한 값

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^T R_T \quad (0 \leq \gamma \leq 1)$$

강화학습 소개

• 마르코프 결정 과정 (MDP) (4/9)

• 가치 함수

• E_π : 정책 π 를 따랐을 때 예측되는 기대값

- 에이전트가 특정 상태에 있거나 특정 상태에서 특정 행동을 선택했을 때, 앞으로 얻을 누적 보상의 기대값을 평가하는 함수
 - 현재 상태가 s 일 때, 정책 π 를 계속 따랐을 경우 앞으로 얻을 누적 할인 보상의 기대값

상태 s 가 얼마나 좋은가?

$$v_\pi(s) = E_\pi[G_t | S_t = s] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right]$$

- 상태 s 에서 행동 a 를 선택한 후, 이후 정책 π 를 따랐을 때 얻을 누적 할인 보상의 기대값

상태 s 에서 행동 a 를 선택하는 것이 얼마나 좋은가?

$$q_\pi(s, \underline{a}) = E_\pi[G_t | S_t = s, \underline{A_t = a}] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, \underline{A_t = a}\right]$$

- $\pi(a|s)$: 현재 s 에서 정책 π 에 따라 행동 a 를 선택할 확률
- $p(s', r|s, a)$: 행동 a 를 했을 때 다음 상태 s' , 보상 r 이 발생할 확률

• 마르코프 결정 과정 (MDP) (5/9)

• 가치 함수의 재귀적 표현: 벨만 방정식 (bellman equation)

- 현재 가치 $v_\pi(s)$ 를 즉시 보상 R_{t+1} 과 할인된 다음 상태의 가치 $v_\pi(S_{t+1})$ 로 표현한 방정식

• 방정식 유도

- 가치 함수의 정의 상태 s 에서 정책 π 를 따를 때 얻을 누적 보상의 기대값

- $v_\pi(s) = E_\pi[G_t | S_t = s]$

• 누적 보상의 재귀적 표현

현재의 누적 보상은 즉시 보상과
이후의 누적 보상으로 나눌 수 있음

- $G_t = R_{t+1} + \gamma G_{t+1}$

• 벨만 방정식 표현

현재 상태의 가치를 즉시 보상과
다음 상태의 가치로 표현

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

- MDP의 전이 확률을 이용했을 때 표현

가능한 행동·다음 상태·보상의 확률을 고려하여
상태의 가치를 계산

- $v_\pi(s) = \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) [r + \gamma v_\pi(s')]$

강화학습 소개

- 마르코프 결정 과정 (MDP) (6/9)
- 동적 프로그래밍 (DP, Dynamic Programming)
 - 환경의 전이확률과 보상 함수를 알고 있는 경우, 벨만 방정식을 반복적으로 적용하여 가치 함수와 최적 정책을 계산하는 방법

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

벨만 방정식: 현재 상태의 가치는 현재 보상과 다음 상태의 가치로 표현

현재 가치 함수를 반복적으로 갱신하면 각 상태의 가치가 수렴하지 않을까?

$$V(s) \rightarrow V'(s) \rightarrow V''(s) \rightarrow \dots$$

강화학습 소개

- 마르코프 결정 과정 (MDP) (7/9)
- 동적 프로그래밍 (DP, Dynamic Programming)
 - 정책 평가(policy evaluation)
 - 주어진 정책 π 를 따를 때 각 상태의 가치를 계산하는 과정

$$v_{\pi}(s) = \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a)[r + \gamma v_{\pi}(s')]$$

정책은 그대로 두고, 가치 함수만 계산

- 정책 π 가 주어짐
 - 각 상태에서 정책 π 에 따라 행동
 - 현재 보상과 다음 상태의 가치 계산을 더함
 - $v_{\pi}(s)$ 업데이트
 - 모든 상태의 가치가 수렴할 때까지 반복
- $V(s) \rightarrow V'(s) \rightarrow V''(s) \rightarrow \dots$

강화학습 소개

- 마르코프 결정 과정 (MDP) (8/9)

- 동적 프로그래밍 (DP, Dynamic Programming)

- 정책 개선(policy improvement)

- 각 상태 s 에서 행동 a 를 선택한 후 이후 정책 π 를 따를 때의 가치 $q_{\pi}(s, a)$ 를 비교하여, 가장 좋은 행동을 선택하도록 정책을 개선

$$\pi \rightarrow \pi' \rightarrow \pi'' \rightarrow \dots$$

$$q_{\pi}(s, a) = \sum_{s', r} p(s', r | s, a) [r + \gamma v_{\pi}(s')]$$

가치 함수는 그대로 두고, 정책을 개선

- 정책 반복(policy iteration)

- 정책 평가와 정책 개선을 반복하여 최적 정책을 찾음

$$\pi \rightarrow v_{\pi} \rightarrow \pi' \rightarrow v_{\pi'} \rightarrow \pi'' \rightarrow \dots$$

$$\pi'(s) = \operatorname{argmax}_a q_{\pi}(s, a)$$

강화학습 소개

- 마르코프 결정 과정 (MDP) (9/9)
- 동적 프로그래밍 (DP, Dynamic Programming)
 - 한계
 - 환경의 전이확률과 보상 함수 등 환경 모델을 사전에 알아야 함
 - e.g., 로봇이 어떤 행동을 할 때, “이 행동을 하면 80% 확률로 A 상태로 가고, 20% 확률로 B 상태로 간다.”는 사실을 알고 있어야 함
 - 상태·행동 공간이 커질수록 모든 상태와 행동에 대한 계산량이 급격히 증가
 - 실제 환경에서는 정확한 환경 모델을 얻기 어려움

심층 강화 학습

- 개요

- 강화학습과 딥러닝을 결합한 하위 분야

- 필요성

- 기존 강화학습에서는 정책과 가치 함수를 단순한 함수로 표현하여, 상태 및 행동 공간이 커질수록 일반화에 실패
- 정책과 가치 함수를 신경망으로 표현하여, 이미지, 비디오, 연속적인 작업과 같은 고차원 상태 공간을 다룰 수 있게함

- 분류

- 환경 모델의 사용 여부에 따른 분류
- 데이터를 수집하는 정책에 따른 분류

심층 강화 학습

- 환경 모델의 사용 여부에 따른 분류
 - 모델 기반 접근법 (model-based learning)
 - 환경의 전이 모델이나 보상 모델을 학습하거나 알고 있다고 가정
 - 학습한 환경 모델을 이용해 미래 상태/보상을 예측
 - e.g., DQN, REINFORCE, PPO, 액터-크리틱
 - 모델 프리 접근법 (model-free learning)
 - 실제 경험을 통해 직접 가치 함수 또는 정책을 학습
 - e.g., 알파제로

구분	모델 기반 접근법	모델 프리 접근법
핵심	미래 상태·보상을 예측	경험으로 정책/가치 함수 학습
장점	• 계획 및 미래 예측 가능 • 실제 환경과의 상호작용을 줄일 수 있음 • 복잡한 작업을 계획하는 데 용이함	• 구현과 파라미터 조정이 비교적 간단 • 확장성이 높고, 계산 복잡도가 낮음
단점	• 고차원 환경에서 정확한 모델 구축이 어려울 수 있음 • 계산 비용이 큼	시행착오가 필요하므로 많은 경험이 필요할 수 있음

심층 강화 학습

- 데이터를 수집하는 정책에 따른 분류
 - 온-폴리시 (on-policy)
 - 데이터를 수집하고 학습하는 데 동일한 정책 사용
 - e.g., REINFORCE, PPO, 액터-크리틱
 - 오프-폴리시 (off-policy)
 - 학습에 사용하는 정책(목표 정책)과 데이터를 수집하는 정책(행동 정책)을 분리하여 사용
 - e.g., DQN, 알파제로

구분	온-폴리시 (on-policy)	오프-폴리시 (off-policy)
장점	• 결정을 내리는 데 사용되는 현재 정책을 직접 평가하고 개선 • 학습된 정책이 실제 환경과의 상호작용을 통해 최적화되므로 에이전트가 안정적이고 분산에 덜 취약	• 과거 데이터나 다른 정책이 생성한 데이터를 활용 가능해 샘플 효율성이 높음 • 행동 정책과 목표 정책을 분리하여, 탐색은 자유롭게 하면서도 학습은 목표 정책을 최적화하는 데 집중
단점	• 과거 정책 데이터를 버려 샘플 효율성이 낮아, 복잡한 환경에서는 방대한 데이터 필요 • 탐색 능력이 제한적이어서 안정적인 환경에 적합	• 현재 정책이 취한 행동으로부터 직접 학습하지 않기 때문에 행동 정책과 목표 정책 간의 불일치가 업데이트의 분산을 키워 학습 안정성을 떨어뜨릴 수 있음
예시	• 챗봇이 사용자 질문에 더 나은 답변을 학습하는 경우, 챗봇이 특정 정책으로 답변을 제공하고 사용자 피드백을 활용해 정책 최적화	• 음악 추천 시스템에서 행동 정책이 새로운 장르나 최신 음반을 탐색하도록 유도 • 목표 정책은 사용자 맞춤형 추천 성능 최적화

심층 강화 학습

- Q-러닝 (Q-Learning)과 심층 Q-네트워크 (DQN)

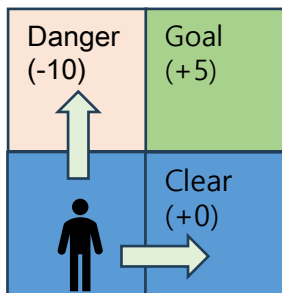
- Q-러닝

- 개요

- 상태 s 에서 행동 a 를 선택했을 때 얻을 수 있는 행동 가치를 출력하는 함수

- 특징

- 상태-행동 값을 저장하는 조희 테이블인 Q -테이블 사용
 - Q -값은 특정 상태에서 특정 행동을 취하고 이후 최적 전략을 따른다고 가정했을 때 얻을 수 있는 기대 미래 보상을 나타냄
 - 일반적으로는 가장 높은 Q 값을 가진 행동을 선택하지만, 탐색 정도를 조절함 (ϵ -greedy)으로써 새로운 행동을 시도할 가능성을 남겨둠



상태

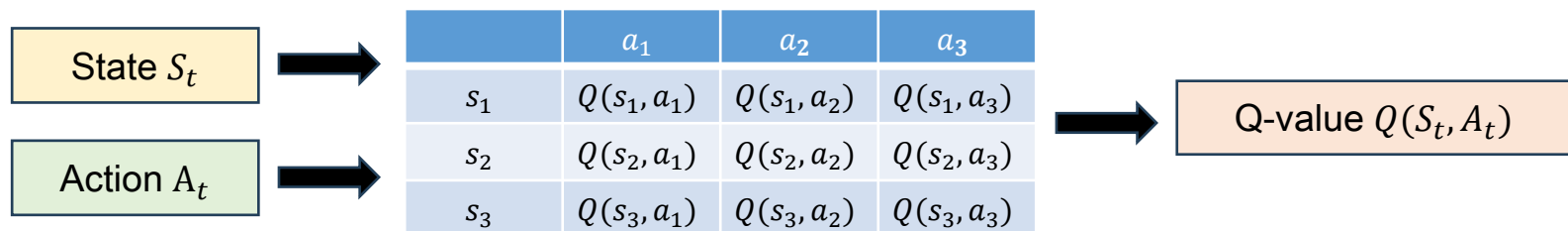
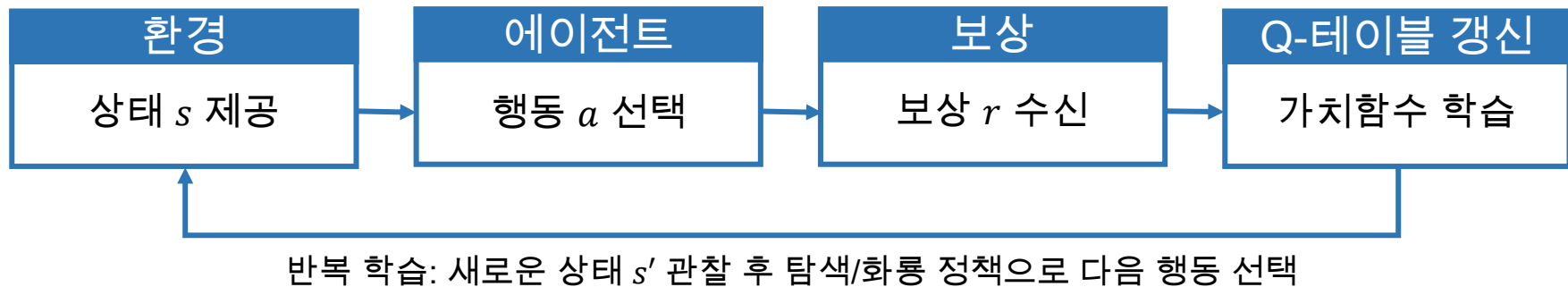
행동

	Left	Right	Up	Down
(1,1)	0	0	0	0
(1,2)	0	0	0	0
(2,1)	0	0	0	0
(2,2)	0	0	0	0

심층 강화 학습

- Q-러닝 (Q-Learning)과 심층 Q-네트워크 (DQN)

- Q-러닝
 - 동작 방식



심층 강화 학습

- Q-러닝 (Q-Learning)과 심층 Q-네트워크 (DQN)
 - Q-러닝
 - 장점
 - 환경 모델 없이 학습 가능 (모델 프리 접근법)
 - 최적 정책을 직접 학습할 수 있음
 - 오프-폴리시이므로 과거 경험이나 다른 정책에서 얻은 경험 활용 가능
 - 단점
 - 연속적인 상태 공간에서는 직접 적용하기 어려움
 - 복잡한 환경에서는 모든 상태-행동의 Q -값을 테이블로 저장하기 어려움

심층 강화 학습

- Q-러닝 (Q-Learning)과 심층 Q-네트워크 (DQN)
- 심층 Q-네트워크
 - 개요
 - Q-러닝에서 사용하는 Q-테이블을 심층신경망으로 대체한 알고리즘
 - 특징
 - 모델 프리 접근법 및 오프-폴리시
 - 복잡하고 고차원적인 상태를 입력으로 처리 가능
 - 대표적인 가치 기반 심층 강화 학습

심층 강화 학습

• Q-러닝 (Q-Learning)과 심층 Q-네트워크 (DQN)

• 심층 Q-네트워크

• 구성요소 및 동작

• Main Network

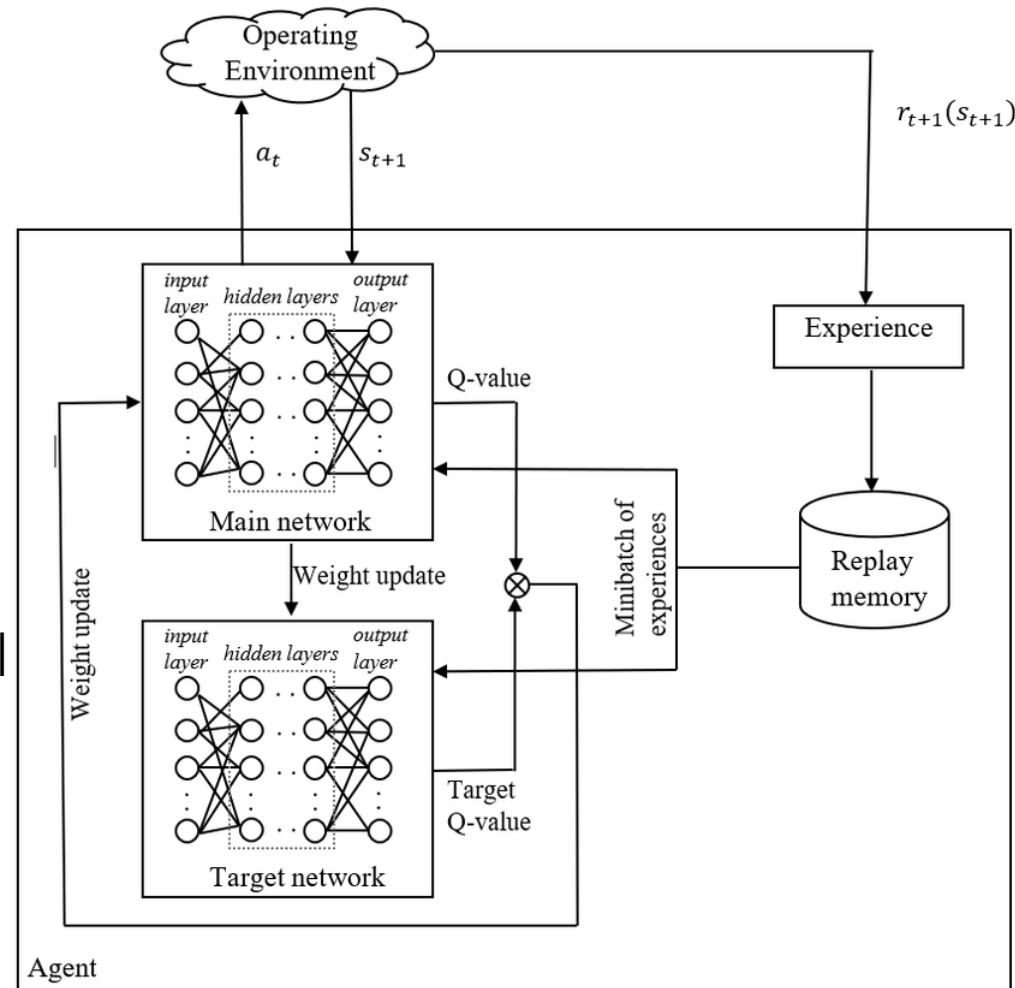
- 현재 상태 s_t 를 입력으로 받아 각 행동에 대한 Q-value를 출력
- $Q(s_t, a)$ 를 근사하는 신경망

• Target Network

- TD target 계산에 사용되는 별도의 신경망
- 일정 주기마다 Main network의 가중치를 복사

• Replay Memory

- 경험 $(s_t, a_t, r_{t+1}, s_{t+1})$ 을 저장
- 저장된 경험 중 일부를 minibatch로 무작위 샘플링하여 학습



심층 강화 학습

- Q-러닝 (Q-Learning)과 심층 Q-네트워크 (DQN)

- 심층 Q-네트워크

- 장점

- Q-테이블 없이 신경망을 이용해 복잡한 상태 공간 처리 가능
 - 이미지, 게임과 같은 고차원 입력에서도 학습 가능

- 단점

- 신경망 학습으로 인해 Q-러닝보다 계산량이 큼
 - 학습이 불안정해질 수 있음
 - 많은 경험 데이터가 필요할 수 있음

심층 강화 학습

- REINFORCE 알고리즘

- 개요

- Q값을 학습하는 대신 정책 자체를 신경망으로 표현하고 직접 최적화하는 정책 기반 알고리즘

- 특징

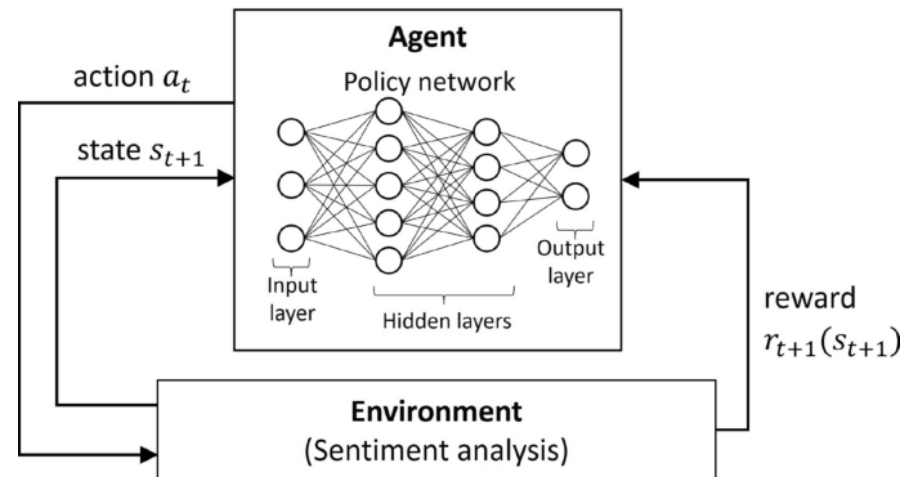
- 모델 프리 접근법 및 온-폴리시
- 정책 신경망이 상태를 입력받아 행동의 확률을 출력

심층 강화 학습

• REINFORCE 알고리즘

• 동작

1. 현재 정책 π_θ 에 따라 환경과 상호작용
2. Episode를 수행하며 경험 수집
3. 각 시점에서 얻은 누적 보상 G_t 계산
4. 높은 누적 보상을 얻은 행동의 확률을 높이는 방향으로 정책 업데이트



심층 강화 학습

- REINFORCE 알고리즘

- 장점

- 정책을 직접 최적화
- 연속적인 행동 공간에도 적용 가능
- 확률적인 정책을 자연스럽게 표현

- 단점

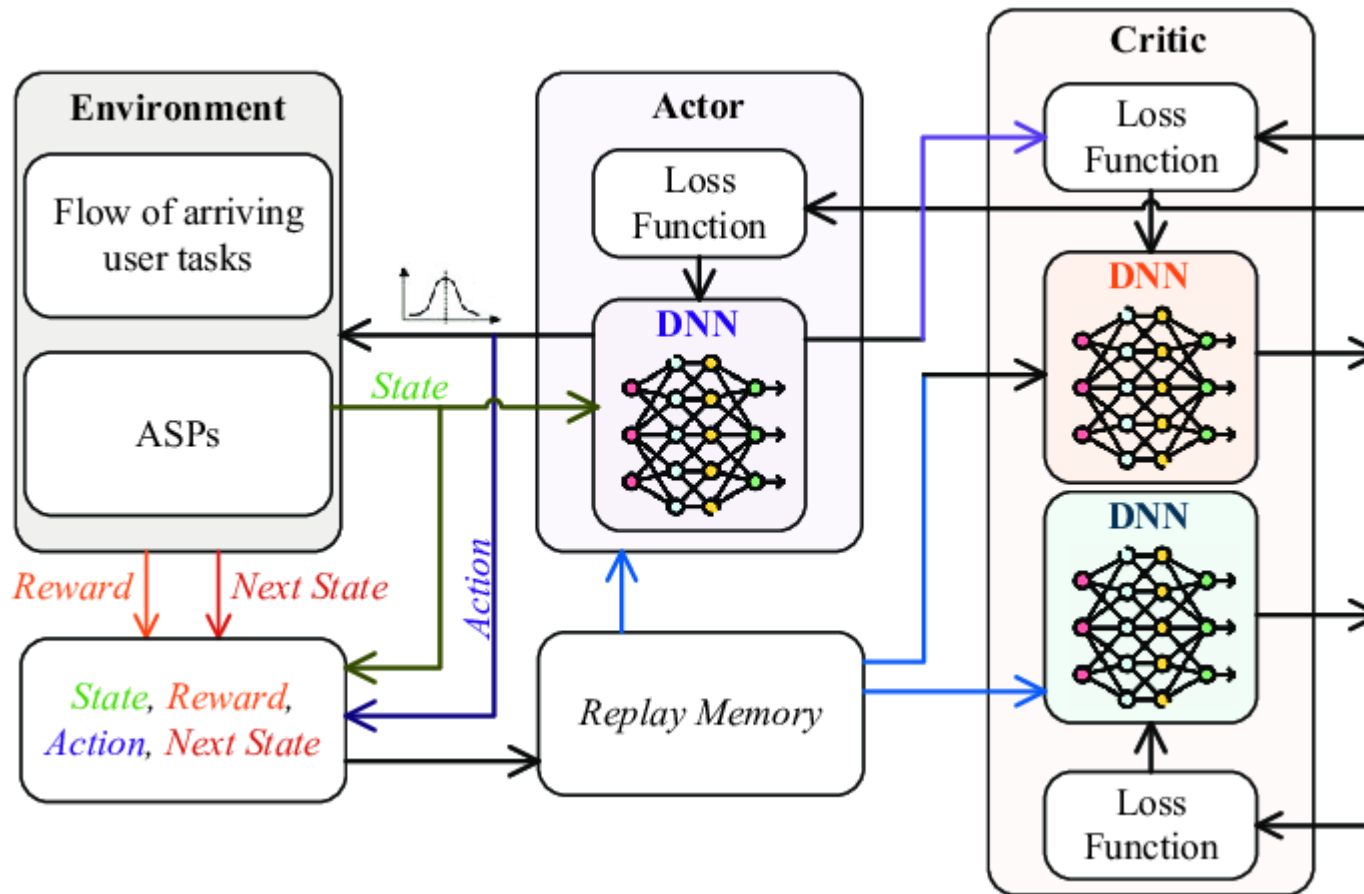
- Episode가 끝나야 전체 누적 보상을 알 수 있는 경우가 많음
- 보상에 따른 정책 업데이트의 분산이 클 수 있음
- 학습이 불안정하거나 데이터 효율이 낮을 수 있음

심층 강화 학습

- 액터-크리틱(Actor-Critic) 구조
 - 개요
 - 정책 기반 방법과 가치 기반 방법의 아이디어를 결합하여, 어떤 행동을 할지 결정(Actor), 그 행동이 얼마나 좋은지 평가(Critic)하는 두 개의 구성요소를 함께 학습하는 알고리즘
 - 특징
 - 정책과 가치 함수의 분리 가능
 - 연속적인 행동 공간에 대한 적용 가능

심층 강화 학습

- 액터-크리틱(Actor-Critic) 구조
- 동작원리



*DU, Hongyang, et al. "Enabling AI-generated content (AIGC) services in wireless edge networks".
arXiv preprint arXiv:2301.03220, 2023.

심층 강화 학습

- 액터-크리틱(Actor-Critic) 구조
 - 장점
 - 정책과 가치 함수를 함께 활용
 - REINFORCE보다 낮은 분산으로 정책을 학습할 수 있음
 - Actor와 Critic의 역할을 분리하여 효율적인 학습 가능
 - 단점
 - Actor와 Critic을 동시에 학습해야 하므로 구조가 복잡
 - Critic의 가치 추정이 부정확하면 Actor 학습에도 영향을 줄 수 있음
 - 하이퍼파라미터 및 학습 안정성에 주의 필요

심층 강화 학습

- PPO (Proximal Policy Optimization)

- 개요

- 정책이 한 번의 업데이트에서 지나치게 크게 변하는 것을 제한하여 안정적인 학습을 유도하는 정책 기반 알고리즘

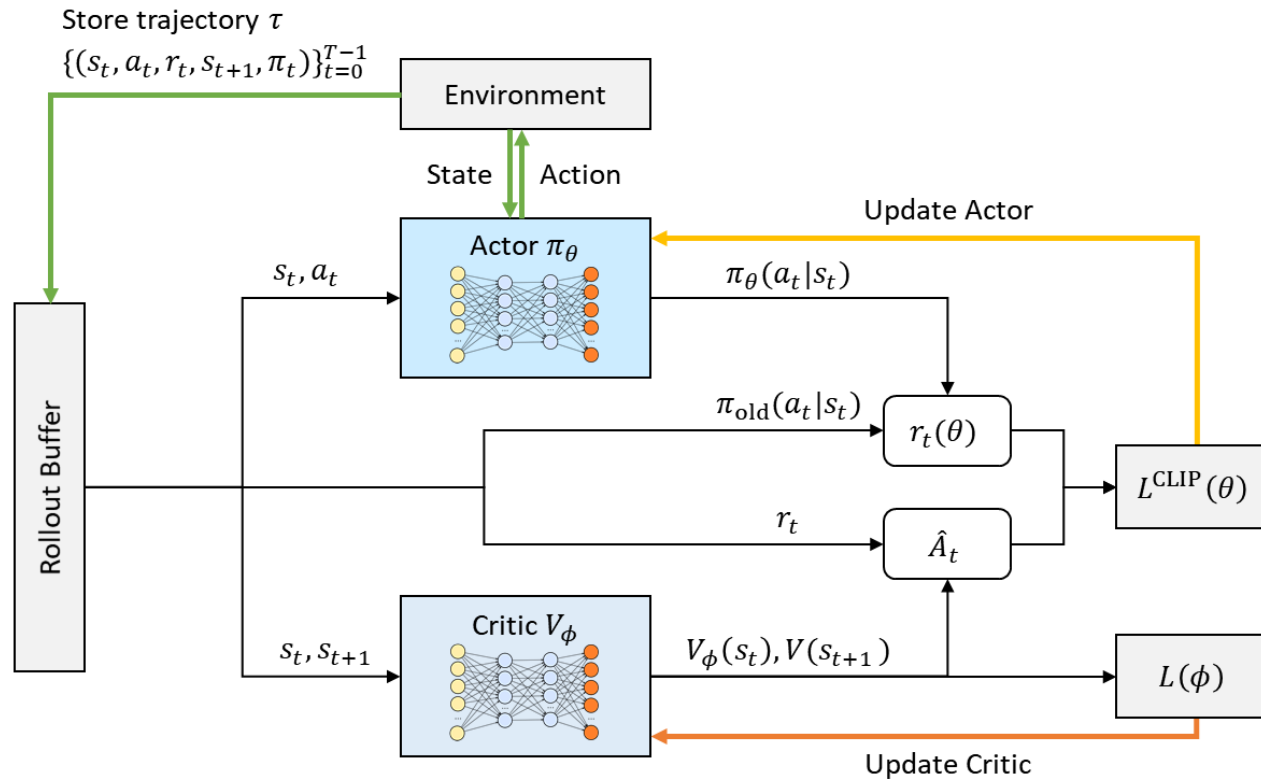
- 특징

- 모델 프리 접근법 및 온-폴리시
- 정책 업데이트의 크기를 제한
- REINFORCE 알고리즘 계열의 정책 경사 방법을 보다 안정적으로 개선

심층 강화 학습

- PPO (Proximal Policy Optimization)

- 동작 원리



*<https://ai-com.tistory.com/entry/RL-%EA%B0%95%ED%99%94%ED%95%99%EC%8A%B5-%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-5-PPO>.

심층 강화 학습

- PPO (Proximal Policy Optimization)

- 장점

- 정책 업데이트를 제한하여 학습 안정성 향상
- 구현이 비교적 단순
- 다양한 강화학습 문제에서 널리 활용

- 단점

- 기본적으로 온-폴리시이므로 과거 데이터를 자유롭게 재사용하기 어려움
- 정책을 반복적으로 업데이트하기 위해 많은 환경 상호작용이 필요할 수 있음

심층 강화 학습

- 알파제로(AlphaZero)

- 개요

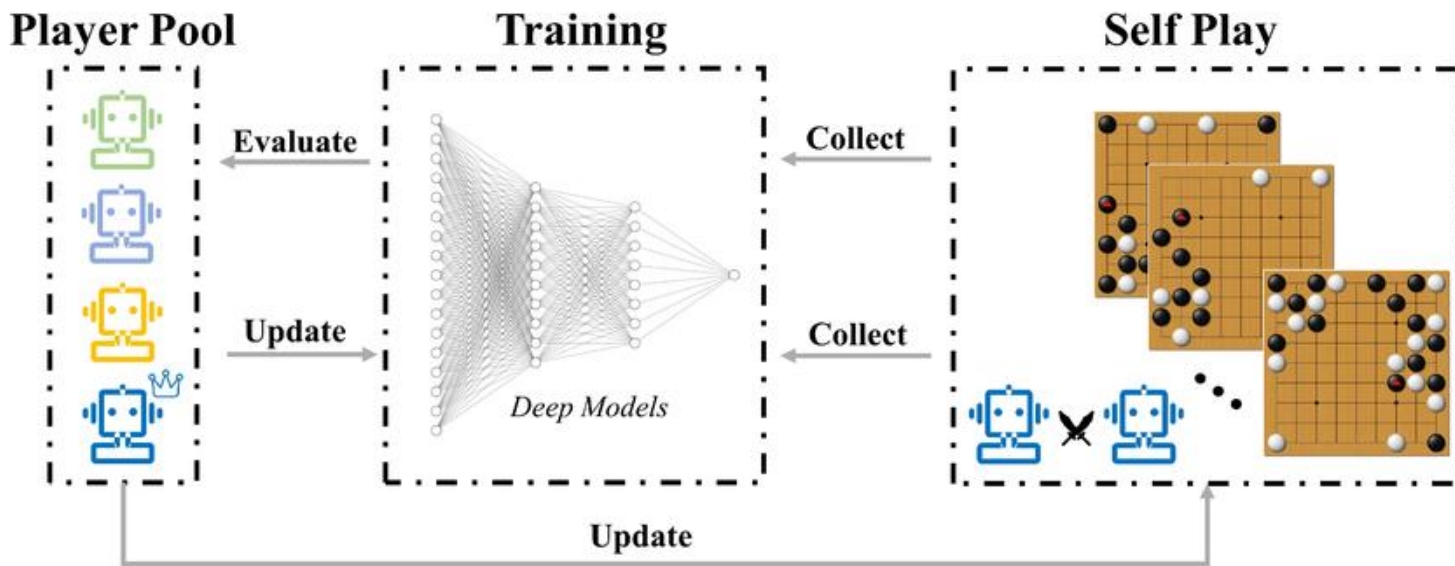
- 신경망 + Monte Carlo Tree Search(MCTS) + 자기대국(Self-play)을 결합한 강화학습 방법

- 특징

- 신경망을 이용해 현재 게임 상태에서 유망한 행동과 상태의 가치를 평가
- MCTS를 통해 가능한 행동들을 탐색하여 실제 선택할 행동을 결정
- 게임을 자기 자신과 반복적으로 플레이하면서 학습 데이터를 생성
- 생성된 데이터를 이용해 신경망을 업데이트하고, 더 강력해진 모델로 다시 Self-play를 수행

심층 강화 학습

- 알파제로(AlphaZero)
- 동작 원리



*GAO, Yifan; WU, Lezhou. "Efficiently mastering the game of nogo with deep reinforcement learning supported by domain knowledge".
Electronics, vol. 10, no.13, 2021.

심층 강화 학습

- 알파제로(AlphaZero)

- 장점

- 인간이 직접 정답 행동을 제공하지 않아도 학습 가능
- 자기대국을 통해 높은 수준의 전략을 스스로 발견
- 탐색과 학습을 결합하여 복잡한 게임 의사결정에 강력

- 단점

- 많은 계산 자원과 자기대국 데이터 필요
- MCTS를 수행하는 데 상당한 계산량 필요
- 게임처럼 명확한 상태·행동·승패 구조가 있는 환경에 특히 적합

심층 강화 학습

• 알고리즘 비교

알고리즘	On-policy	Off-policy	Model-free	Model-based
DQN		✓	✓	
REINFORCE	✓		✓	
PPO	✓		✓	
Actor-Critic	✓		✓	
AlphaZero		✓		✓

심층 강화 학습

- 과제와 미래 전망 (1/3)
 - 본 적 없는 환경에서의 일반화
 - 학습하지 않은 새로운 환경·조건에서의 일반화가 어려움
 - 특정 시뮬레이션 환경에 과도하게 의존할 경우 실제 환경 적용에 한계
 - 다양한 환경에서도 안정적으로 성능을 유지할 수 있어야 함
 - 보상 함수 설계
 - 보상 함수가 에이전트의 행동과 학습 성능에 큰 영향을 미침
 - 복잡한 환경에서는 적절한 보상 함수 설계가 어려움
 - 희소 보상 환경에서 특히 어려움

심층 강화 학습

- 과제와 미래 전망 (2/3)

- 모델 기반 계획의 누적 오류

- 모델 기반 강화학습은 환경 모델의 예측에 의존
- 예측 범위가 길어질수록 모델 오류가 누적
- 복잡하고 고차원적인 환경에서 실제 적용에 제약

- 멀티태스크 학습

- 하나의 에이전트가 여러 작업을 동시에 수행하도록 학습하기 어려움
- 여러 작업을 수행할 수 있는 범용적인 에이전트 설계가 필요
- 작업 간 지식 공유 및 효율적인 학습이 중요

심층 강화 학습

- 과제와 미래 전망 (3/3)
 - 멀티모달 강화 학습
 - 이미지·텍스트·음성 등 다양한 형태의 정보를 동시에 처리해야 함
 - 여러 모달리티의 정보를 통합하여 복잡한 환경과 상호작용할 수 있는 정책을 학습해야 함
 - 멀티모달 정보를 효과적으로 통합하는 방법이 필요

LLM과 강화학습 모델의 상호작용

- 유형

- 강화학습으로 강화된 LLM

- 하나 이상의 자연어 처리 작업에서 강화학습을 활용해 LLM의 성능을 향상시키는 경우

- LLM으로 강화된 강화학습

- 자연어 처리가 아닌 작업을 수행하는 강화학습 알고리즘을 훈련시키는 데 LLM을 사용하는 경우

- 강화학습과 LLM의 결합

- 서로 간에 훈련하거나 파인튜닝하지 않고, 강화학습 모델과 LLM을 함께 사용해 특정 기술 집합을 계획하는 방식

LLM과 강화학습 모델의 상호작용

- 강화학습으로 강화된 LLM
 - 강화학습은 LLM의 파인튜닝, 프롬프트 엔지니어링, 정렬에 활용됨
 - LLM은 시퀀스에서 다음 단어를 예측하도록 훈련되나, 이로 인해 LLM의 학습 목표와 인간 가치 사이에 불일치 발생
 - LLM이 편향된 텍스트를 생성하거나 유해한 콘텐츠를 출력하거나 지시를 제대로 따르지 않을 수 있음

LLM과 강화학습 모델의 상호작용

- 강화학습으로 강화된 LLM
 - 인간 피드백 기반 강화학습 (RLHF, Reinforcement learning from human feedback)
 - 보상을 사람의 선호로부터 추론하여 LLM 학습
 - 과정
 - 인간 피드백 데이터 수집
 - 데이터 기반으로 보상 모델 학습
 - 강화학습을 통해 LLM 파인튜닝
 - 특징
 - LLM과의 상호작용은 프롬프트를 통해 이루어지며, LLM이 수행할 작업에 대한 명확한 지시 사항을 모두 담고 있어야 함
 - 정책 네트워크를 훈련시켜 원하는 프롬프트를 생성할 수 있으며, 생성된 프롬프트는 일반적으로 여러 LLM에 전이가 가능함

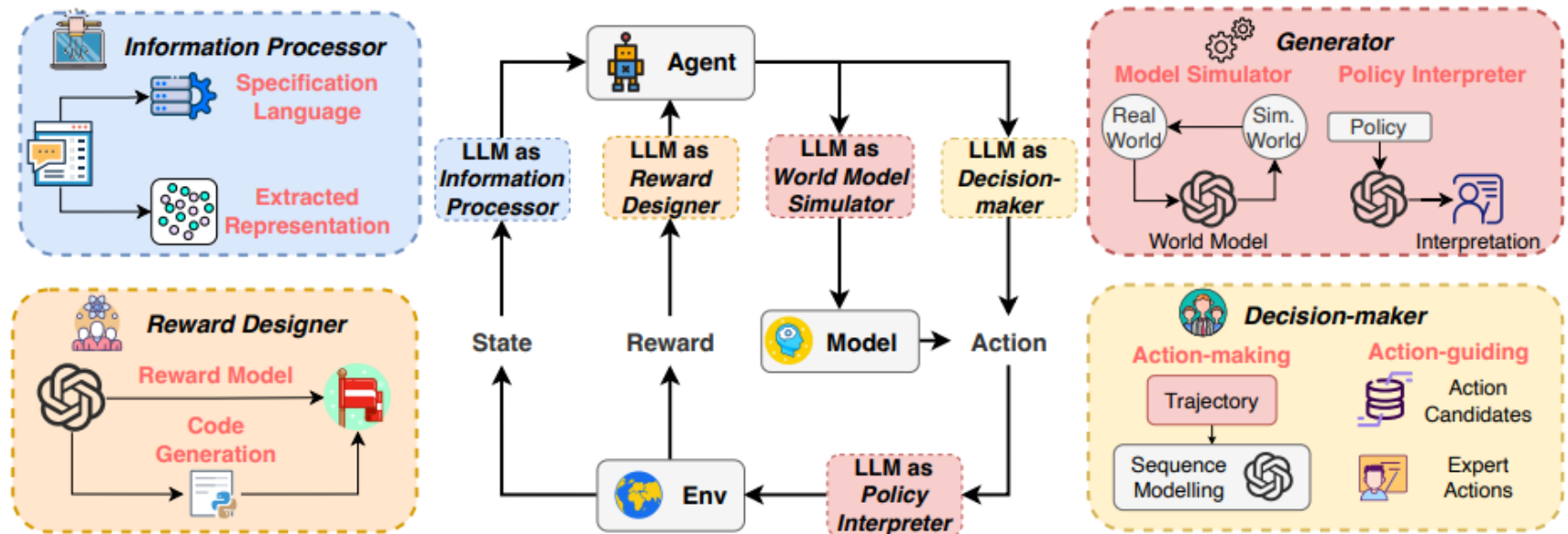
LLM과 강화학습 모델의 상호작용

- LLM으로 강화된 강화학습

- 사전 학습된 LLM이 가진 멀티모달 정보 처리, 생성, 추론, 기타 고차원 인지 능력을 활용해 강화학습 에이전트를 지원
- LLM은 추론과 계획 수립 능력이 있어 학습 성능 향상 가능
- LLM의 일반화 능력으로 다양한 상황에 적응할 수 있으며, 사전학습 단계에서 축적한 지식을 기반으로 여러 도메인이나 작업을 전이할 수 있음
- 사전 학습된 모델은 지식을 지속적으로 확장하거나 새로운 능력을 습득하기 어려우나, LLM을 활용함으로써 이를 보완할 수 있음

LLM과 강화학습 모델의 상호작용

- LLM으로 강화된 강화학습
 - LLM이 삽입되는 지점에 따라 LLM의 역할이 구분됨
 - 정보처리자 (Information Processor)
 - 보상설계자 (Reward Designer)
 - 의사결정자 (Decision-maker)
 - 생성자 (Generator)

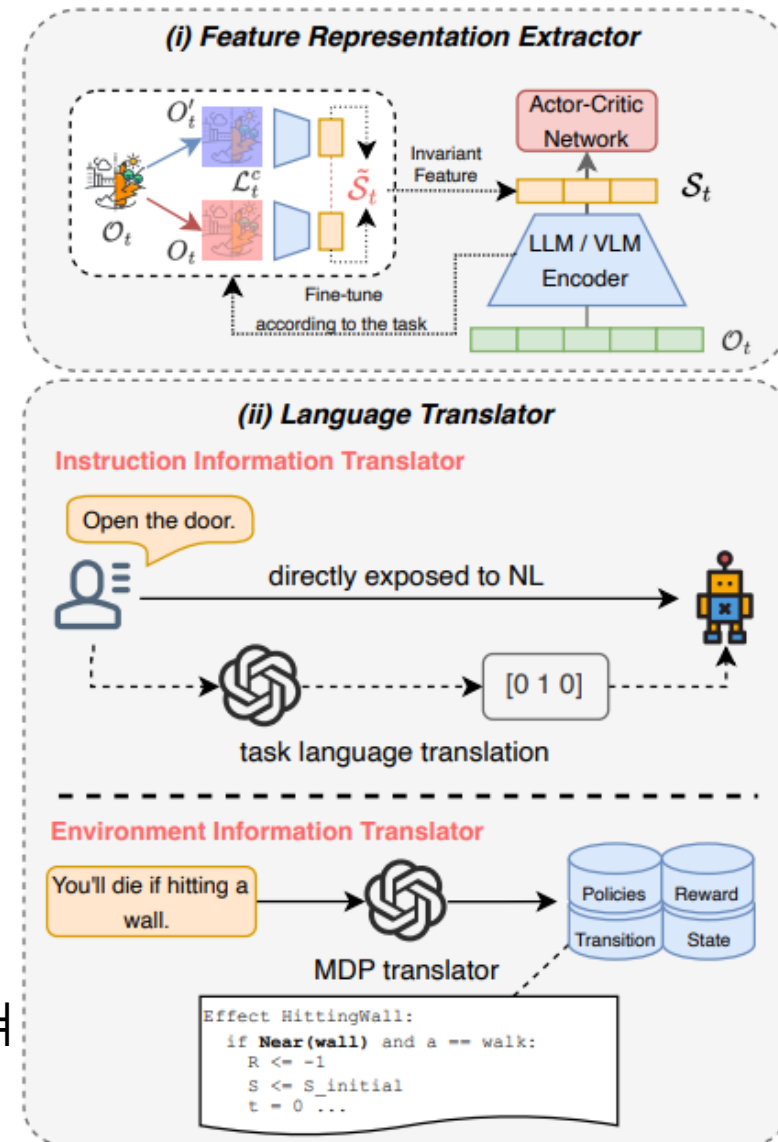


LLM과 강화학습 모델의 상호작용

- LLM으로 강화된 강화학습
 - 정보처리자로서의 LLM (1/2)
 - 에이전트와 환경 사이에서, 복잡한 정보를 에이전트가 이해하고 학습하기 쉬운 형태로 변환하는 LLM
 - 정보: 텍스트, 이미지·영상, 자연어 지시, 과거 관찰 정보 등
 - 필요성
 - 중요 특징 추출
 - 정보의 차원을 줄이고 복잡성을 낮출 수 있음
 - 에이전트의 학습을 보다 빠르게 할 수 있음
 - 정보의 표준화 및 형식 변환
 - 에이전트가 일관된 형태의 정보를 입력받도록 지원
 - 과거 정보의 요약 및 압축
 - 에이전트가 과거에 관찰한 정보를 LLM이 요약 해 긴 과거 정보를 압축된 표현으로 제공
 - 환경 정보의 의미 해석
 - 환경에서 얻은 정보를 에이전트가 이해할 수 있는 형태로 변환

LLM과 강화학습 모델의 상호작용

- LLM으로 강화된 강화학습
- 정보처리자로서의 LLM (2/2)
 - 특징 표현 추출기로 사용
 - 환경에서 얻은 관측 O_t 가 이미지와 같이 복잡한 정보라면, LLM/VLM Encoder가 의사결정에 중요한 특징 $\tilde{S}_t$ 를 추출
 - 언어 번역기로 사용
 - Instruction Information Translator
 - 자연어 지시를 에이전트가 사용할 수 있는 형태로 변환
 - Environment Information Translator
 - 환경이 제공하는 정보를 LLM이 해석하여 에이전트가 활용할 수 있는 MDP 형태로 변환

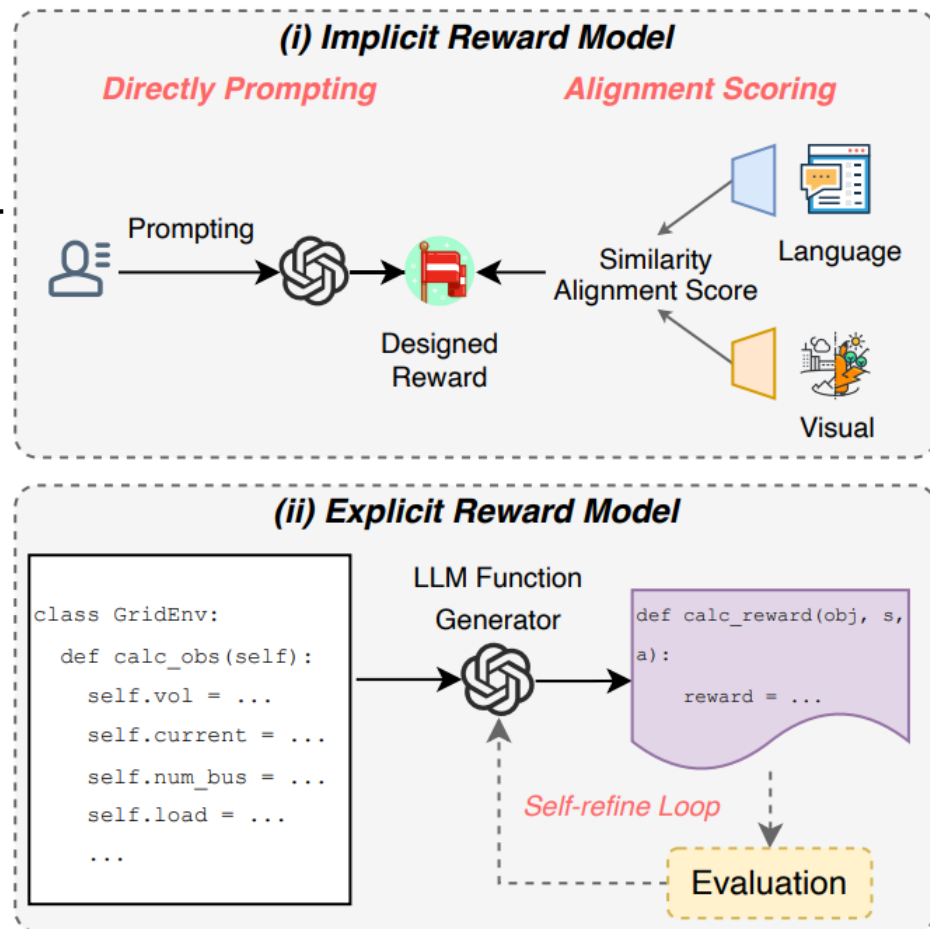


LLM과 강화학습 모델의 상호작용

- LLM으로 강화된 강화학습
 - 보상설계자로서의 LLM (1/2)
 - 에이전트가 달성해야 할 목표를 정의하고, 에이전트의 행동 결과를 평가하여 적절한 보상 또는 벌점을 부여하는 LLM
 - 필요성
 - 복잡한 문제에서는 명확한 보상 함수를 설계하기 어려움
 - 보상이 희소하게 발생하는 환경은 학습에 필요한 피드백이 부족
 - 사람이 자연어로 기대하는 행동이나 목표를 설명하면, LLM이 이를 바탕으로 보상 또는 평가 기준을 생성

LLM과 강화학습 모델의 상호작용

- LLM으로 강화된 강화학습
 - 보상설계자로서의 LLM (2/2)
 - 암묵적 보상 모델
 - LLM에게 에이전트가 기대되는 행동을 자연어로 설명하고, LLM이 행동을 평가하는 방식
 - 명시적 보상 모델
 - LLM의 코드 생성 능력을 활용하여 보상 함수를 직접 생성하는 방식



LLM과 강화학습 모델의 상호작용

- LLM으로 강화된 강화학습
 - 의사결정자로서의 LLM
 - 현재 상태와 주어진 정보를 바탕으로 에이전트가 수행할 행동을 결정하는 LLM
 - 필요성
 - 강화학습 에이전트는 좋은 행동을 찾기 위해 많은 시행착오를 겪어야 하는 샘플 비효율성을 가짐
 - 강화학습 에이전트는 행동 공간이 크다면 가능한 행동을 하나씩 시도해야 하는 탐색 비효율성을 가짐
 - LLM은 새로운 환경에서 모든 행동을 시행착오로 학습하는 대신에, 사전 지식과 추론 능력을 활용해 유망한 행동을 선택함

LLM과 강화학습 모델의 상호작용

- LLM으로 강화된 강화학습

- 의사결정자로서의 LLM

- Action-making

- LLM이 강화학습 정책 자체의 일부가 되어 행동을 직접 생성하는 방식

- Action-guiding

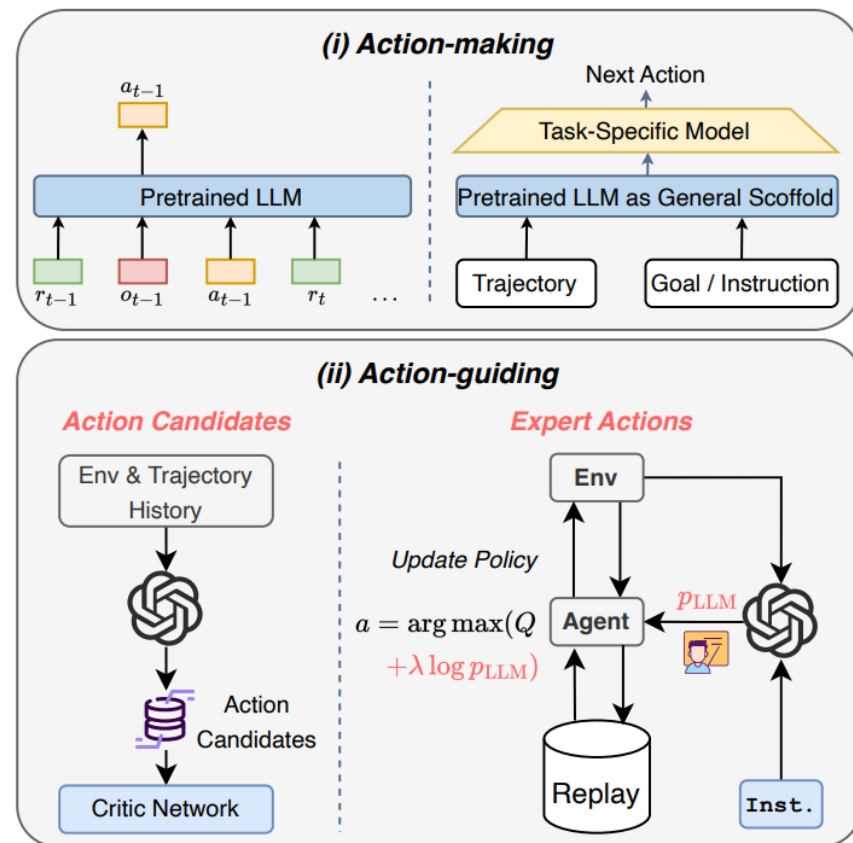
- LLM이 RL 에이전트의 행동 선택을 보조하고, 최종 행동 결정을 유도하는 방식

- Action Candidates

- LLM이 현재 상황과 작업을 이해하여 유망한 행동 후보를 생성

- Expert Actions

- LLM이 전문가 수준의 행동이나 행동 계획을 생성



LLM과 강화학습 모델의 상호작용

- LLM으로 강화된 강화학습
 - 생성자로서의 LLM
 - 강화학습에 필요한 경험·궤적·정책 등의 정보를 생성하여 에이전트의 학습과 계획을 지원하는 LLM
 - 필요성
 - 실제 환경에서 충분한 경험 데이터를 수집하려면 많은 시간과 비용이 필요하지만, LLM을 통해 학습에 필요한 궤적을 생성하여 실제 환경과의 상호작용을 줄여 샘플 효율성을 높임
 - 생성된 궤적을 활용하여 에이전트의 학습 및 계획을 지원함으로써 미래 예측 및 시뮬레이션이 가능
 - 에이전트의 행동이나 정책에 대한 설명을 생성하여 사람이 에이전트의 의사결정을 이해할 수 있도록 지원

LLM과 강화학습 모델

- LLM으로 강화된 강화학습

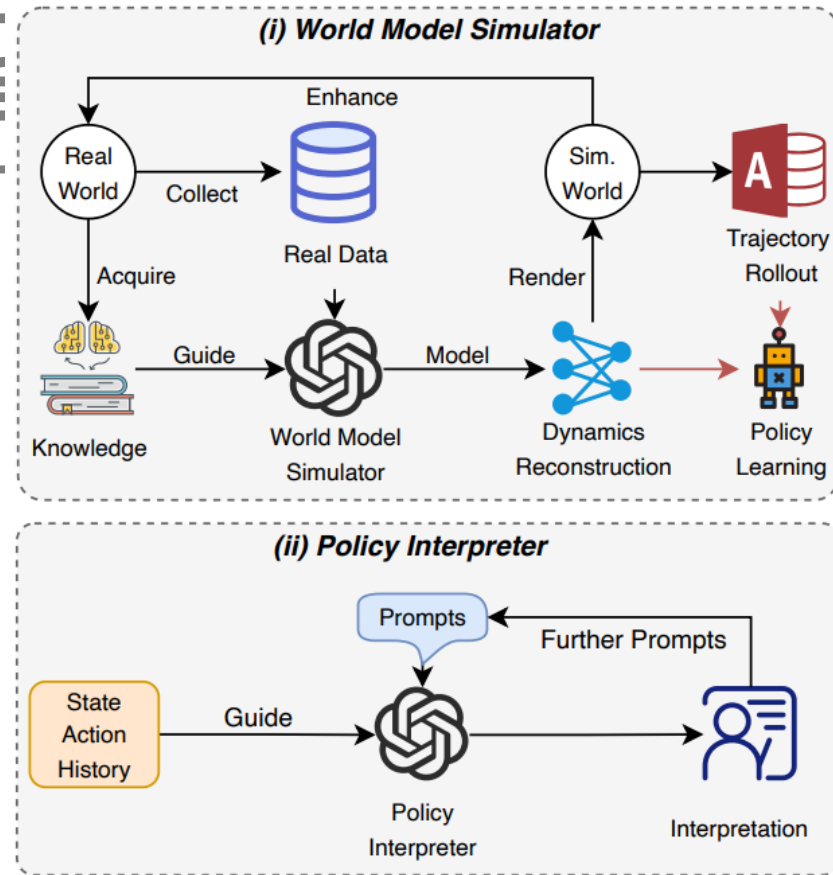
- 생성자로서의 LLM

- World Model Simulator

- LLM이 환경의 동적 특성을 학습한 World Model처럼 동작하여, 실제 환경과 상호작용하지 않고도 미래 궤적을 생성

- Policy Interpreter

- 주어진 상태와 행동에 대해 에이전트의 행동을 인간에게 설명함으로써 에이전트의 안전성을 확인
 - 설명가능한 강화학습(XRL, explainable RL)
 - 에이전트의 행동 및 정책을 인간이 명확하게 이해할 수 있도록 자연어로 설명
 - 사람이 에이전트의 의사결정 과정을 이해하는 데 도움



*CAO, Yuji, et al. "Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods." *IEEE Transactions on Neural Networks and Learning Systems*, 36.6: 9737-9757, 2024.

LLM과 강화학습 모델의 상호작용

- LLM으로 강화된 강화학습

- 활용 가능성

- 다양한 종류의 데이터를 분석하여, 정보 통합 및 지시 사항 이해, 목표 및 보상 구조 개선 작업에 활용
- e.g., 로봇틱스, 자율주행, 헬스케어 추천 시스템, 에너지 관리

- 한계

- LLM의 성능에 크게 의존하여, 기존의 LLM의 문제 (e.g., 할루시네이션, 편향 문제)를 이어받음
- 작업이 복잡하거나 데이터에 노이즈가 많을 경우 LLM이 잘 못 해석할 가능성 존재
- LLM 도입 시 학습과 추론 단계에서 계산 비용이 증가하며 지연시간 증가

Thanks!

이 하 늘(haneul@pel.sejong.ac.kr)