12기 정규세션

ToBig's 11기 한재연

강화학습 기초

contents

Contents

Unit 01 | 개요

- Reinforcement Learning

행동심리학의 "강화"

- 시행착오를 통해 학습하는 방법
- 이전에 배우지 않았지만 직접 시도하면서 행동과 결과로 나타나는 좋은 보상 사이의 상관 관계를 학습 하고, 좋은 보상을 얻게 해주는 행동을 점점 더 많이 하는 것

Unit 01 | 개요

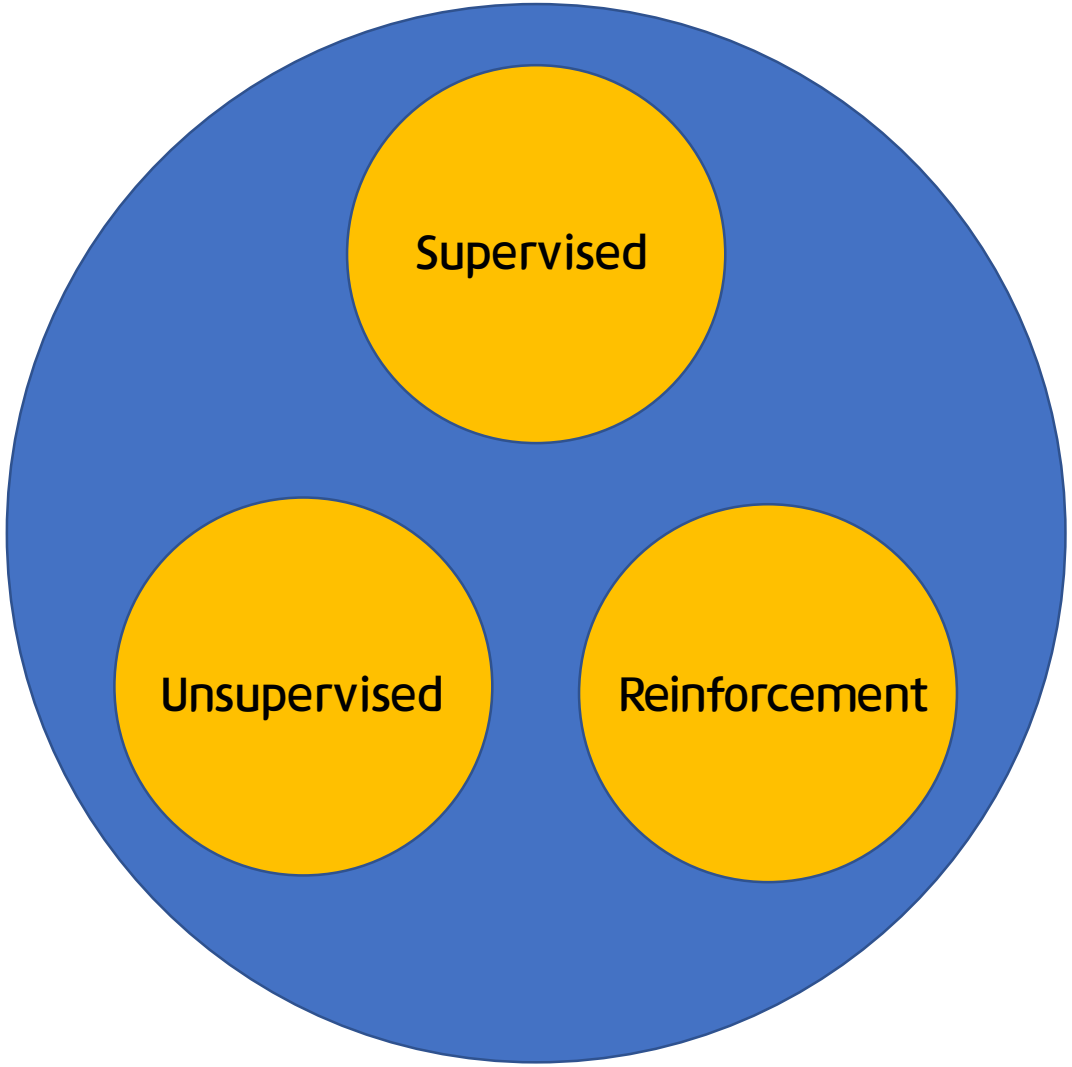
- Reinforcement Learning

스키너 상자 실험

먹이(보상)에 의해 강화

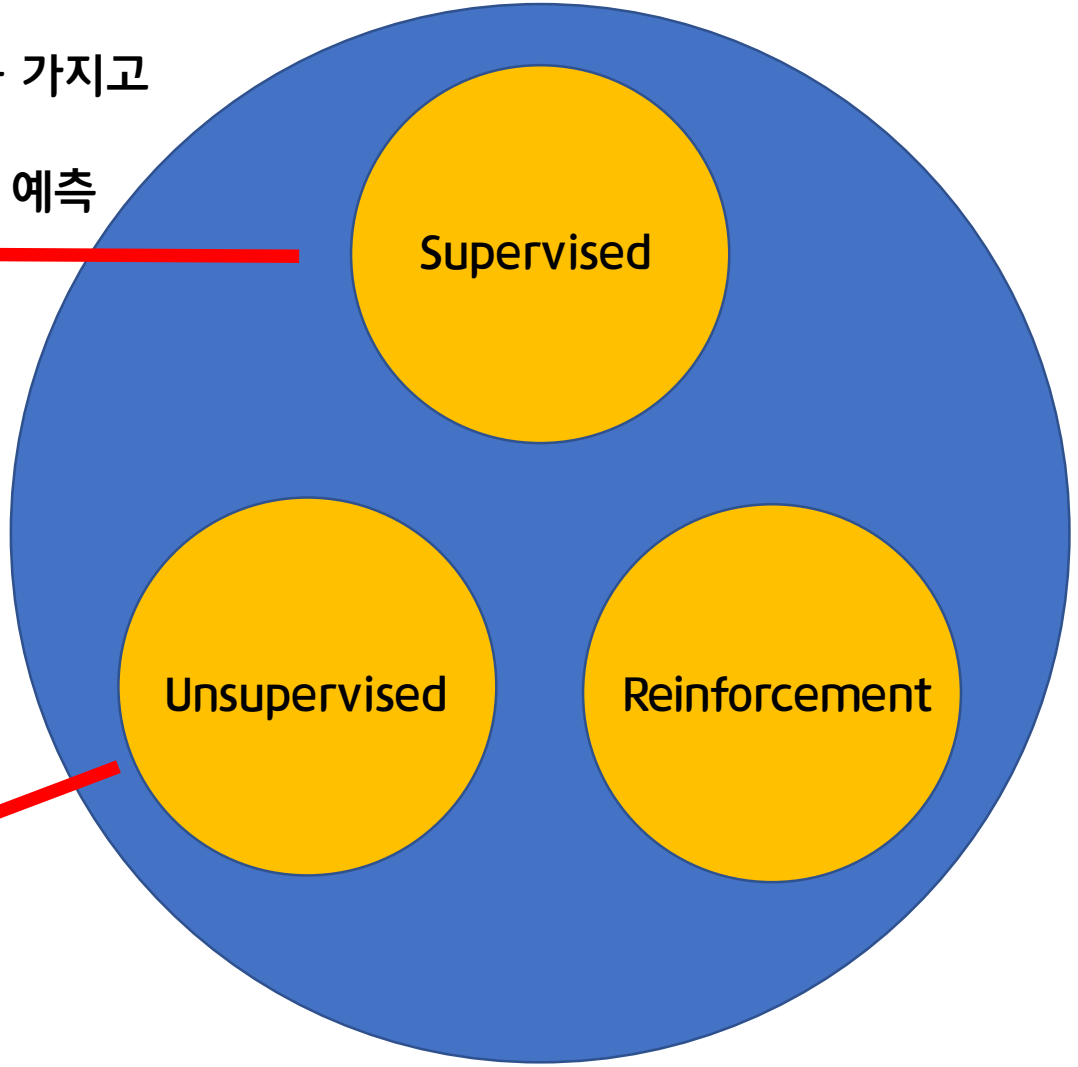
1. 배고픈 상태의 흰 쥐를 스키너 상자에 넣는다.
2. 흰 쥐는 스키너 상자 안에서 돌아다니다가 우연히 지렛대를 누르게 된다.
3. 지렛대를 누르자 먹이가 나온다.
4. 지렛대와 먹이 간의 상관관계를 알지 못하는 쥐는 다시 상자 안을 돌아다닌다.
5. 다시 우연히 지렛대를 누른 흰 쥐는 또 먹이가 나오는 것을 보고 지렛대를 누르는 행동을 자주 하게 된다.
6. 이러한 과정이 반복되면서 흰 쥐는 지렛대를 누르면 먹이가 나온다는 사실을 학습하게 된다.

Unit 01 | 개요



Unit 01 | 개요

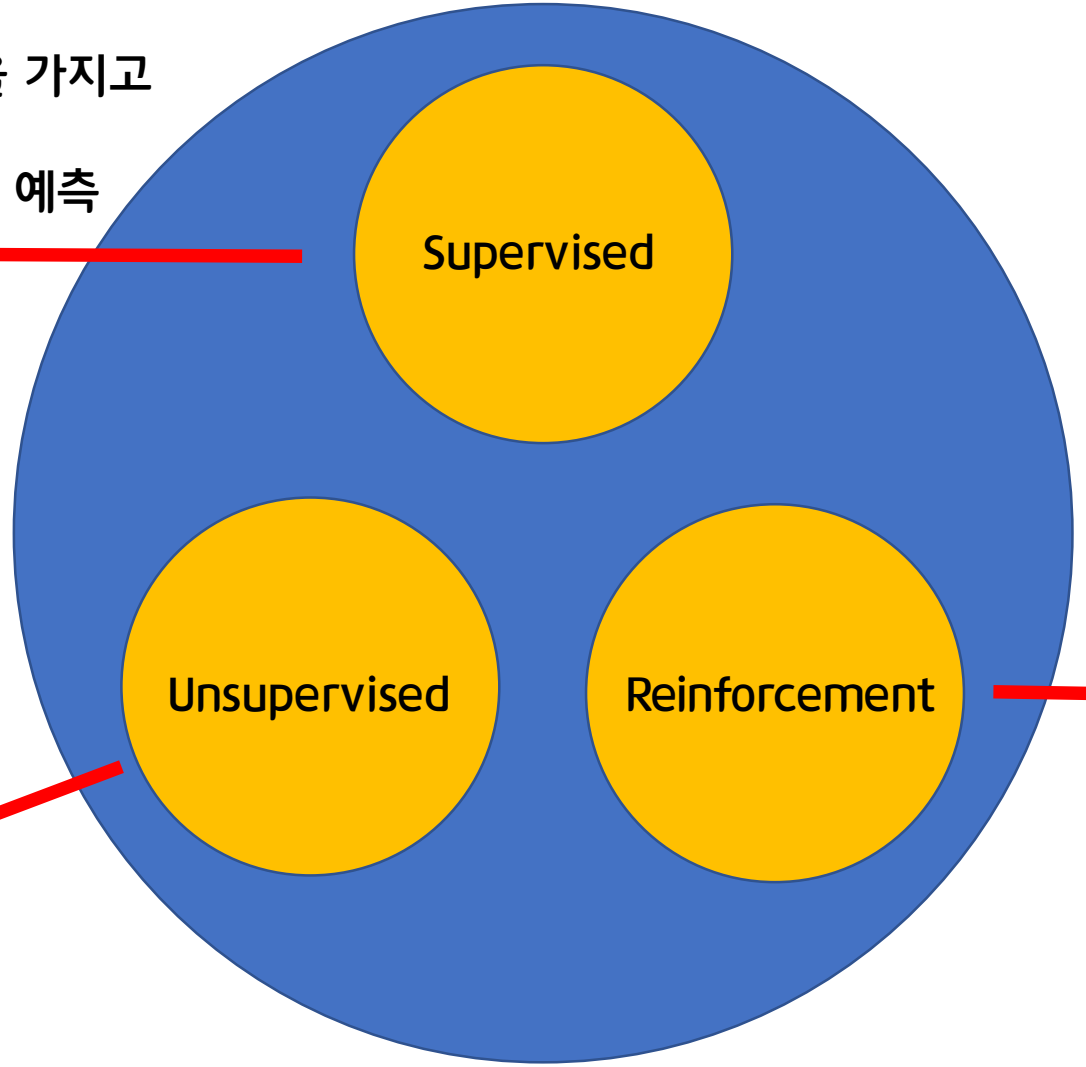
데이터와 그에 대한 라벨(정답) 을 가지고 학습
입력으로 데이터를 받으면 정답을 예측
분류 문제, 회귀 문제



정답이 없음
데이터의 숨은 구조 파악
Clustering, 차원 축소

Unit 01 | 개요

데이터와 그에 대한 라벨(정답) 을 가지고
학습
입력으로 데이터를 받으면 정답을 예측
분류 문제, 회귀 문제



정답이 없음
데이터의 숨은 구조 파악
Clustering, 차원 축소

행동, 정책, 상태, 보상을 가지고
특정 환경에 대한 문제를 해결
경험을 통해 학습
보상을 최대화 하도록 행동하는
것이 목표

Unit 01 | 개요

응용 사례

Atari Breakout

바둑

Unit 01 | 개요

- Reinforcement Learning

"어떤 **환경** 안에서 정의된 **에이전트**가 현재의 **상태**를 인식하여, 선택 가능한 행동들 중 **보상**을 최대화(최소화)하는 **행동** 혹은 행동 순서를 선택하는 방법", 위키백과

목적: 보상의 합을 최대화하는 "최적의 행동양식, 정책" 을 학습

순차적인 행동 결정 문제를 해결하는데 쓰일 수 있는 학습 방법

이 때 환경의 정보를 잘 알지 못한다는 가정이 들어간다.

Unit 01 | 개요

앞으로 자주 쓰게 될 용어들

- 에이전트: 학습 시킬 주체
- 에피소드: 상황이 종료 될 때 까지의 과정 (게임을 예시로 들면, 게임 한 판)
- 타임스텝: 행동 한 번을 선택 하는 시간의 단위
- 정책: 특정 상태 s 에서 특정 행동 a 를 취할 확률 $\pi(a \mid s)$

contents

Unit 02 | MDP & 벨만 방정식

- 그렇다면 모든 판단 문제를 정책, 상태, 행동, 보상 등을 가지고 풀 수 있을까?
- 매우 추상적
- 내가 해결하려는 문제를 하나의 수학적인 틀을 가지고 표현해야 한다.
- Markov Decision Process (MDP)

MDP 의 구성 요소

Unit 02 | MDP & 벨만 방정식

보상 함수
타임스텝 t, 상태 s 에서 행동 a 를 취할 때 에이전트가 받을 t+1 번째 보상의 기댓값

$$R_s^a = \mathrm{E}[\mathrm{R}_{t+1} \mid \mathrm{S}_t = \mathrm{s}, \mathrm{A}_t = \mathrm{a}]$$

상태 변환 확률
상태s 에서 행동a 를 취했을때 다음 상태가s' 일 확률

$$P_{ss'}^a = \mathrm{P}[S_{t+1} = \mathrm{s}' \mid \mathrm{S}_t = \mathrm{s}, \mathrm{A}_t = \mathrm{a}]$$

Unit 02 | MDP & 벨만 방정식

예제: 그리드 월드

목표: 선택된 지점에서 출발하여 지정된 지점까지 도착하기

상태: 각 격자의 위치

행동: 상하좌우

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	+1 (4,4)

Unit 02 | MDP & 벨만 방정식

예제: 그리드 월드

목표: 선택된 지점에서 출발하여 지정된 지점까지 도착하기

상태: 각 격자의 위치

행동: 상하좌우

이 때 $R_{(4,3)}^{Right}$은?

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	$^{+1}$(4,4)

Unit 02 | MDP & 벨만 방정식

예제: 그리드 월드

목표: 선택된 지점에서 출발하여 지정된 지점까지 도착하기

상태: 각 격자의 위치

행동: 상하좌우

이 경우 상태 변환 확률은 1

현실에서는 과연 상태 변환 확률이 항상 1일까?

상태 변환 확률을 미리 알 수 있을까?

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4) +1

Unit 02 | MDP & 벨만 방정식

예제: 그리드 월드

목표: 선택된 지점에서 출발하여 지정된 지점까지 도착하기

상태: 각 격자의 위치

행동: 상하좌우

(4, 3) 에서 오른쪽으로 행동을 취할 때, 에이전트가 (4, 4) 로 갈 상태 변환 확률 $P^{Right}_{(4,3)(4,4)}$가 0.8 이라고 하자.

이 때 $R^{Right}_{(4,3)}$은?

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	+1 (4,4)

Unit 02 | MDP & 벨만 방정식

감가율
나중에 받는 보상의 가치를 감소시키는 정도
$\gamma \in [0,1]$

내가 만약 1억원 복권에 당첨되었다면, 당장 받고 싶을까 1년 후에 받고 싶을까?

당장 받을 수 있는 1억원 복권과 1년 후에 받을 수 있는 복권 중 어떤 것을 선택할까?

K 타임스텝 이 후 받은 보상
$\gamma^k R$

Unit 02 | MDP & 벨만 방정식

예제: 그리드 월드

목표: 선택된 지점에서 출발하여 지정된 지점까지 도착하기

서로 다른 두 에피소드에 대해 생각해 보자

- (1, 2) – (1, 3) – (2, 3) – (3, 3) – (3, 4) – (4, 4)
- (2, 1) – (2, 2) – (2, 3) – (2, 4) – (3, 4) – (3, 3) – (4, 3) – (4, 4)

에이전트가 첫 번째 에피소드를 선택하게끔 학습시킨 다면 더 좋은 성능의 모델이라 할 수 있다.

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4) +1

Unit 02 | MDP & 벨만 방정식

가치 함수
각 상태의 가치

가치 함수로 어떤 정책이 더 좋은 정책인지 판단을 할 수도 있고, 극단적으로는 인접한 상태 중 가장 높은 가치로만 행동을 취하는 방식으로 정책을 정할 수 있을 것이다.
(탐욕 정책)

반환값
한 번의 에피소드에 대해서 에이전트가 받은 보상의 합

Unit 02 | MDP & 벨만 방정식

예제: 그리드 월드

목표: 선택된 지점에서 출발하여 지정된 지점까지 도착하기

서로 다른 두 에피소드에 대해 생각해 보자

- (1, 2) – (1, 3) – (2, 3) – (3, 3) – (3, 4) – (4, 4)
- (2, 1) – (2, 2) – (2, 3) – (2, 4) – (3, 4) – (3, 3) – (4, 3) – (4, 4)

각 에피소드의 반환값은?
감가율을 감안한다면?

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	+1 (4,4)

Unit 02 | MDP & 벨만 방정식

반환값
한 번의 에피소드에 대해서 에이전트가 받은 보상의 합

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

위 식은 몇 개의 정보가 필요할까? 몇 번의 연산이 필요할까?

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

Unit 02 | MDP & 벨만 방정식

가치 함수

각 상태의 가치 = 상태 s 에서 시작될 수 있는 모든 에피소드에 대한 반환값의 기댓값

$$
\begin{aligned}
v(s) &= E[G_t \mid S_t = s] \\
&= E[R_{t+1} + \gamma\, G_{t+1} \mid S_t = s] \\
&= E[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s]
\end{aligned}
$$

Unit 02 | MDP & 벨만 방정식

가치 함수
각 상태의 가치 = 상태 s 에서 시작될 수 있는 모든 에피소드에 대한 반환값의 기댓값

$$v(s) = E[R_{t+1} + \gamma v(S_{t+1}) | S_t = s]$$

각 에피소드로 에이전트가 갈 확률은 정책에 의존 할 것이다.
정책을 고려한 가치 함수 필요

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

벨만 기대방정식

Unit 02 | MDP & 벨만 방정식

예제: 그리드 월드

목표: 선택된 지점에서 출발하여 지정된 지점까지 도착하기

다음 에피소드를 에이전트가 취할 확률은?

- (1, 2) – (1, 3) – (2, 3) – (3, 3) – (3, 4) – (4, 4)

무작위 정책일 경우?
그렇지 않을 경우?

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4) +1

Unit 02 | MDP & 벨만 방정식

최적의 정책이라면 항상 갈 수 있는 상태 중 가장 높은 가치로만 이동 할 것이다.
이 때의 가치는?

$v_*(\mathrm{s}) = \mathrm{E}[\mathrm{R}_{t+1} + \gamma v_*(\mathrm{S}_{t+1}) | \mathrm{S}_t = \mathrm{s}]$, **이 때** $v_*(\mathrm{S}_{t+1}) = \max_{\pi} v_\pi(\mathrm{S}_{t+1})$

벨만 최적 방정식

이 경우 정책은?

$$\pi_*(a \mid s) = \begin{cases} 1 \; if \; a = argmax \; \mathrm{q}_*(\mathrm{s},\mathrm{a}) \\ 0 \; otherwise \end{cases}$$

Unit 02 | MDP & 벨만 방정식

큐 함수
가치 함수의 행동 버전
상태 s 에서 행동 a 를 취했을 때 반환값의 기댓값

$$
\begin{aligned}
q(s,a) &= E[G_t \mid S_t = s, A_t = a] \\
&= E[R_{t+1} + \gamma\, G_{t+1} \mid S_t = s, A_t = a] \\
&= E[R_{t+1} + \gamma\, q(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a]
\end{aligned}
$$

Unit 02 | MDP & 벨만 방정식

가치 함수, 큐 함수, 정책, 감가율, 그리고 상태 변환 확률의 관계 with 벨만 기대 방정식

$$v_\pi(s) = \sum_a \pi(a \mid s) q_\pi(s, a)$$

$$q_\pi(s, a) = R_s^a + \gamma \sum_s P_{ss'}^a v_\pi(s)$$

기댓값을 이용한 정의와는 다르게, **계산이 가능하다.**

Contents

Unit 03 | 정책 & 가치 이터레이션

가치함수가 무엇인지는 알았다. 그렇다면 어떻게 구할까? 벨만 기대 방정식을 살펴보자.

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

Unit 03 | 정책 & 가치 이터레이션

가치함수가 무엇인지는 알았다. 그렇다면 어떻게 구할까? 벨만 기대 방정식을 살펴보자.

$$v_\pi(\mathrm{s}) = \mathrm{E}[\mathrm{R}_{\mathrm{t}+1} + \gamma v_\pi(\mathrm{S}_{\mathrm{t}+1}) \mid \mathrm{S}_\mathrm{t} = \mathrm{s}]$$

$$= \sum_a \pi(a \mid s) q_\pi(s,a) = \sum_a \pi(a \mid s)(R_s^a + \gamma v_\pi(\mathrm{s}'))$$

Why? $q_\pi(s,a) = R_s^a + \gamma \sum_s P_{ss'}^a v_\pi(s)$

Unit 03 | 정책 & 가치 이터레이션

가치함수가 무엇인지는 알았다. 그렇다면 어떻게 구할까? 벨만 기대 방정식을 살펴보자.

$$v_\pi(s) = \sum_a \pi(a \mid s)(R_s^a + \gamma v_\pi(s'))$$

위 식을 **타임스텝**을 기준으로 본다면?

$$v_\pi(s_t) = \sum_a \pi(a_t \mid s_t)(R_{t+1} + \gamma v_\pi(s_{t+1}))$$

Unit 03 | 정책 & 가치 이터레이션

모든 에피소드의 마지막이 될 수 있는 상태의 가치를 구한다음, 아래의 수식을 이용하여 마지막에서 두 번째, 마지막에서 세 번째… 의 타임스텝에 대한 가치를 구하는 연산을 무한이 반복하면, 참 가치함수를 구할 수 있다!

DP 식 접근

$$v_\pi(s_t) = \sum_a \pi(a_t|\, s_t)(\mathrm{R}_{t+1+}\, \gamma v_\pi(s_{t+1}))$$

Unit 03 | 정책 & 가치 이터레이션

모든 에피소드의 마지막이 될 수 있는 상태의 가치는 어떻게 구할까?
그리드 월드 입장에선 항상 도착 지점 (4, 4) 가 마지막 상태.

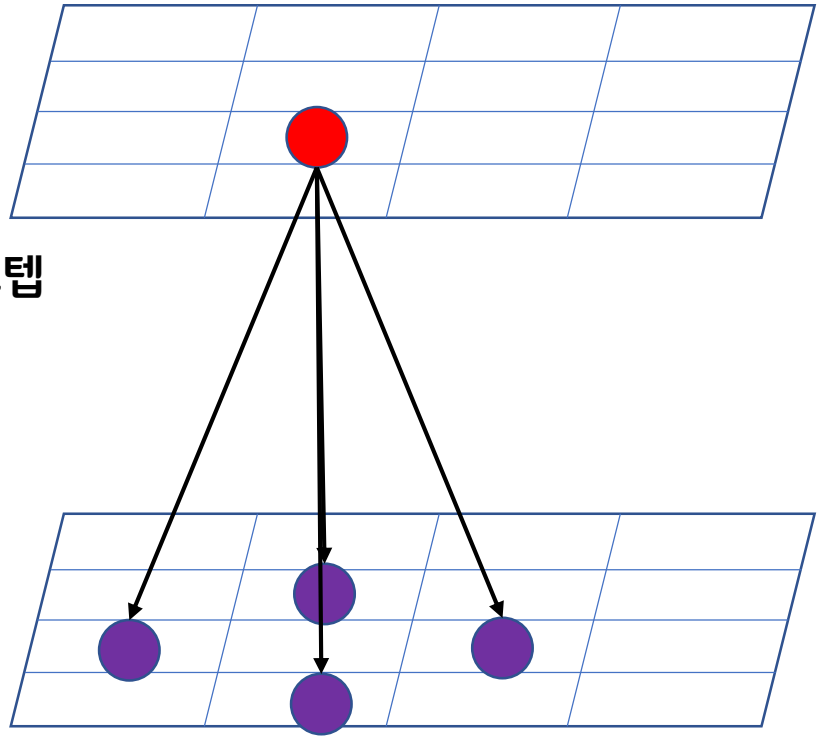
마지막 상태에서는 에피소드가 더 이상 뻗어나가지 못하므로, 이 상태의 보상 1이 바로 가치가 된다.

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4) +1

Unit 03 | 정책 & 가치 이터레이션

메모리에 t+1 번째 타임스텝까지 계산된 가치들이 저장되어있다 가정하면, 이전에 점화식으로 표현된 가치함수의 정의를 그대로 사용하여 t 번째 타임스텝까지 계산할 수 있다.

$$v_\pi(s) = \sum_a \pi(a \mid s)(\mathrm{R}_{t+1+} \gamma v_\pi(s'))$$



t 번째 타임스텝

t+1 번째 타임스텝

Unit 03 | 정책 & 가치 이터레이션

이렇게 각 상태에 대한 가치를 구한다면, 이 가치함수 값들은 정책을 평가하는데 쓰일 수 있다.

우리의 목표는 좋은 정책을 찾는 것이기 때문에, 구한 가치를 이용하여 정책 발전을 시켜야 한다.

즉, 정책 이터레이션은 정책 평가와 정책 발전 두 단계로 나누어진다.

정책 발전은 어떻게?

Unit 03 | 정책 & 가치 이터레이션

이렇게 각 상태에 대한 가치를 구한다면, 이 가치함수 값들은 정책을 평가하는데 쓰일 수 있다.

우리의 목표는 좋은 정책을 찾는 것이기 때문에, 구한 가치를 이용하여 정책 발전을 시켜야 한다.

즉, 정책 이터레이션은 정책 평가와 정책 발전 두 단계로 나누어진다.

정책 발전은 어떻게? 탐욕 정책 발전

Unit 03 | 정책 & 가치 이터레이션

정책 이터레이션

1. 처음엔 정책을 무작위 정책으로 둔다.

$$\pi \leftarrow random$$

2. DP 로 가치함수를 구한다. (정책 평가)

3. 최대의 큐함수값을 가지는 행동으로 정책을 바꾼다. (탐욕 정책 발전)

$$\pi \leftarrow \pi', where\ \pi'(s) = argmax_a q_\pi(s,a) = argmax_a(R_s^a + \gamma v_\pi(s'))$$

4. 2번을 시행한다.

Unit 03 | 정책 & 가치 이터레이션

어짜피 탐욕 정책을 쓸거면, 차라리 벨만 기대 방정식이 아닌 벨만 최적 방정식을 써도 되지 않을까?

가치 이터레이션

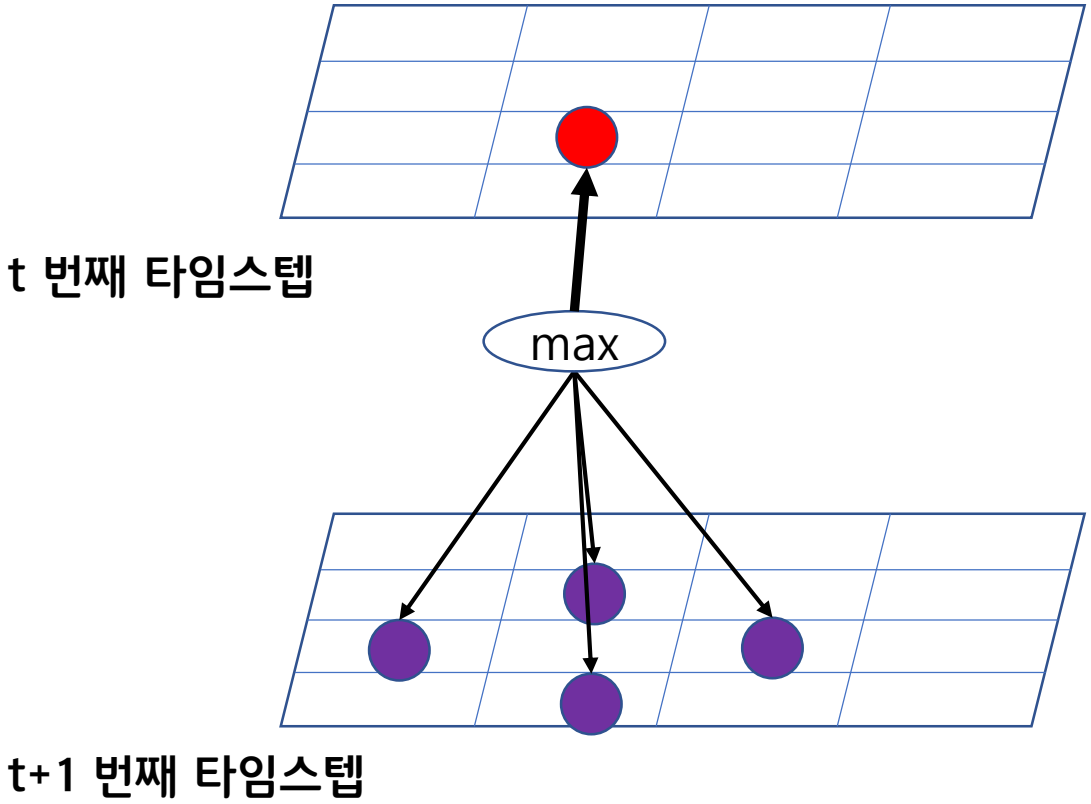
이미 정책은 고정되어 있으므로, 벨만 최적 방정식에 따른 가치함수만 구하면 끝!

Unit 03 | 정책 & 가치 이터레이션

정책 이터레이션에서와는 다르게 인접한 상태로 가는 큐함수에 대한 값만 가져온다.

이 시행을 여러 번 반복하면, 벨만 최적 방정식에 따른 참 가치함수에 근접한 값이 나올거고, 이를 이용하여 에이전트는 탐욕 정책으로 행동한다.

$$v_*(s) = max_a(R_{t+1+} \gamma v_*(s'))$$



Unit 03 | 정책 & 가치 이터레이션

정책 이터레이션을 이용하여 정책을 결정하거나, 가치 이터레이션을 이용하여 가치를 구하는 두 가지 다른 접근으로 학습을 시도하였고, 전체적으로 DP 알고리즘이 쓰였다.

그러나 사실 정책/가치 이터레이션은 현실에서는 많은 한계가 있기에 쓰이지 않는다.

정책/가치 이터레이션의 메모리 사용량이나 연산량은 상태의 개수에 의존한다. 그러나 우리가 풀려는 많은 현실의 문제는 상태의 개수가 매우 많다.
(ex 그리드 월드의 크기가 더 크다면? 3차원이라면? 바둑)

Unit 03 | 정책 & 가치 이터레이션

따라서 사실상 정책/가치 이터레이션은 강화학습 알고리즘이라고 보기 어렵고, 앞으로 다양한 근사 방법과 인공신경망을 이용한 알고리즘 (DQN, A3C 등) 을 본격적인 강화학습 알고리즘으로 본다. (강화학습 세미나에서 함!!)

과제

9주차 알고리즘 세션의 주제는 DP 입니다. 따라서 다음 시간에는 정책/가치 이터레이션을 직접 구현할 것입니다.

그러나 복잡한 수식이 많았기에 한번에 이해 하는 데는 사실 어렵습니다. 원활한 진행을 위해 이번에 나온 모든 개념들을 복습하는 시간을 가지셔야 직접 구현 할 때 잘 따라 오실 수 있을겁니다.

따라서 과제는 복습입니다.

Ref.

- 투빅스 10기 이민주 님의 강화학습 자료
- 파이썬과 케라스로 배우는 강화학습, 위키북스

들어주셔서 감사합니다.