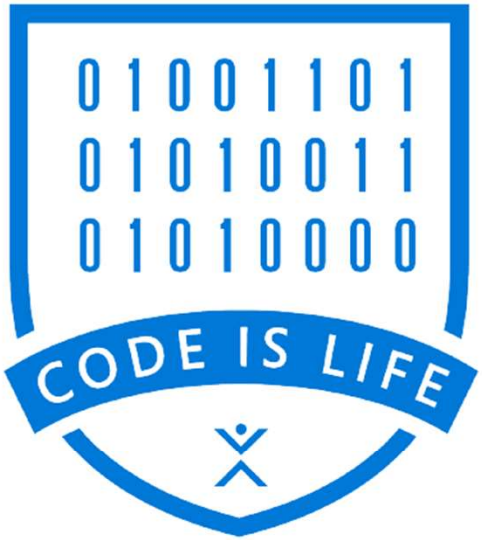
Microsoft Student Partners

강화학습 기초

처음 만나는 강화학습



Microsoft

강화학습이란?

강화학습 소개 영상

강화학습이란?

강화 학습

- 인공지능의 한 분야
- 바둑 인공지능 "알파고"에서 사용된 방법

인공 지능과 강화 학습

인공 지능이란?

인공 지능 人工知能

컴퓨터 인간의 지능이 가지는 학습, 추리, 적응, 논증 따위의 기능을 갖춘 컴퓨터 시스템. 전문가 시스템, 자연 언어의 이해, 음성 번역, 로봇 공학, 인공 시각, 문제 해결, 학습과 지식 획득, 인지 과학 따위에 응용한다.

표준국어대사전

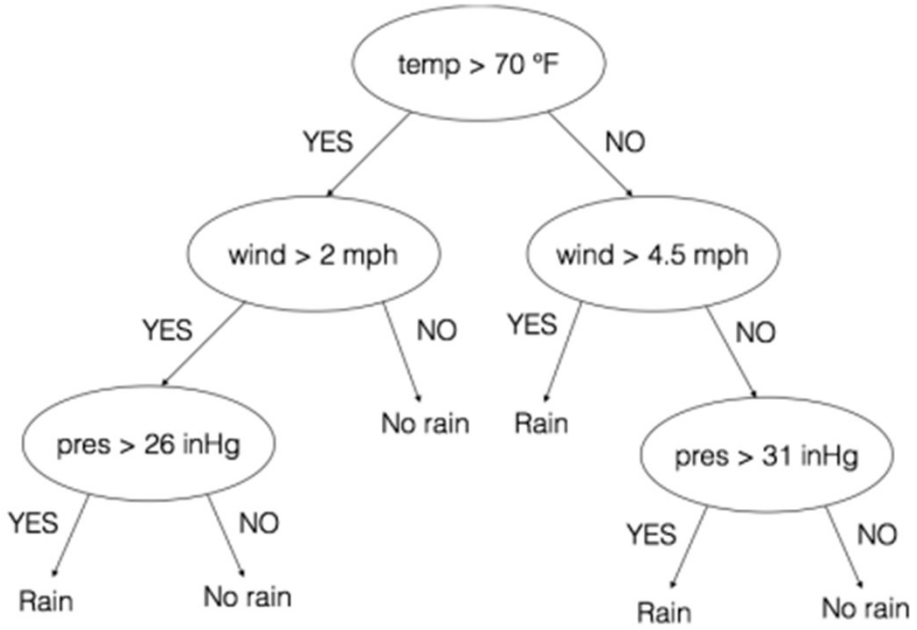
인공 지능과 머신 러닝

인공 지능의 분류

- 규칙 기반 (Rule-based)
  - 사람이 일일이 지정해 준 규칙에 따라 프로그램이 동작
- 머신 러닝(Machine learning)
  - 많은 데이터의 관찰을 통해, 프로그램이 스스로 최적의 방법을 터득

인공 지능과 머신 러닝

머신 러닝이 필요한 이유

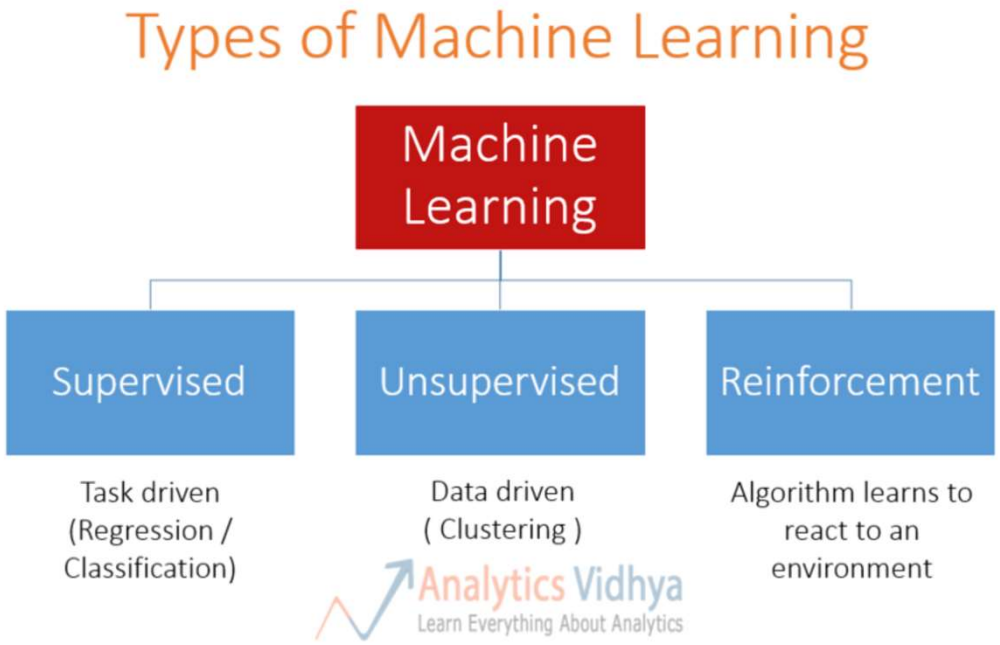


인공 지능과 머신 러닝

머신 러닝이 필요한 이유

머신 러닝의 분류 (환경에 대한 상호 작용을 중심으로)

머신 러닝의 분류



머신 러닝의 분류 (환경에 대한 상호 작용을 중심으로)

- 지도 학습 (Supervised Learning)
  - '정답'이 주어짐
  - 입력 값으로 예측한 예측 값과 정답의 차이를 확인하고, 이에 맞추어 예측 모델을 학습

- 비 지도 학습 (Unsupervised Learning)
  - '정답'이 주어지지 않음. 데이터만 주어짐
  - 정답 없이 데이터 만을 이용해서, 분류 기준을 학습 (무엇으로 분류 되었는지는 모른다!)

- 강화 학습
  - '특정 상황에서의 보상' 이 주어짐
  - 지금 당장의 '정답' 없이 '현재 상황'에서의 보상만을 보고 미래를 위한 학습을 진행해야 하는 경우!

강화 학습

- 자전거를 배우는 경우
  - ~~"이 자전거는 ~~~한 시스템이고 ~~~한 dynamics를 가지고 있어서 만약 10도 정도 기울었을 때는 핸들을 반대로 ~~~한 각속도로 틀어줘야한다. 근데 너의 몸무게가 얼마나 되지?"~~
  - 아무것도 모르고 핸들을 돌려 가며 어떻게 하면 넘어지고 어떻게 하면 똑바로 가는지 학습했을 것 입니다.

나와 자전거(기울어짐, 속도…) (상태)

핸들 왼쪽으로, 오른 쪽으로 (행동)

기울어졌던 자전거가 중심을 잡는다 (보상)

강화 학습

- 인생이란 강화 학습이 아닐까…
  - 여러분은 왜 개발자가 되기로 하셨나요?
  - 80세의 저는, 개발자가 되기로 결정한 것을 후회하지는 않을까요?
    - ~~공무원 시험을 봤어야 했는데…~~
    - 인생에서, 미래의 정답을 지금 알 수는 없습니다.
  - 컴퓨터 공학과를 선택하기 전, 스무 살의 나 (상태)
  - 흥미? 취업? 연봉? (보상)
  - 개발자가 되자! 컴공 진학, 개발 공부… (액션)

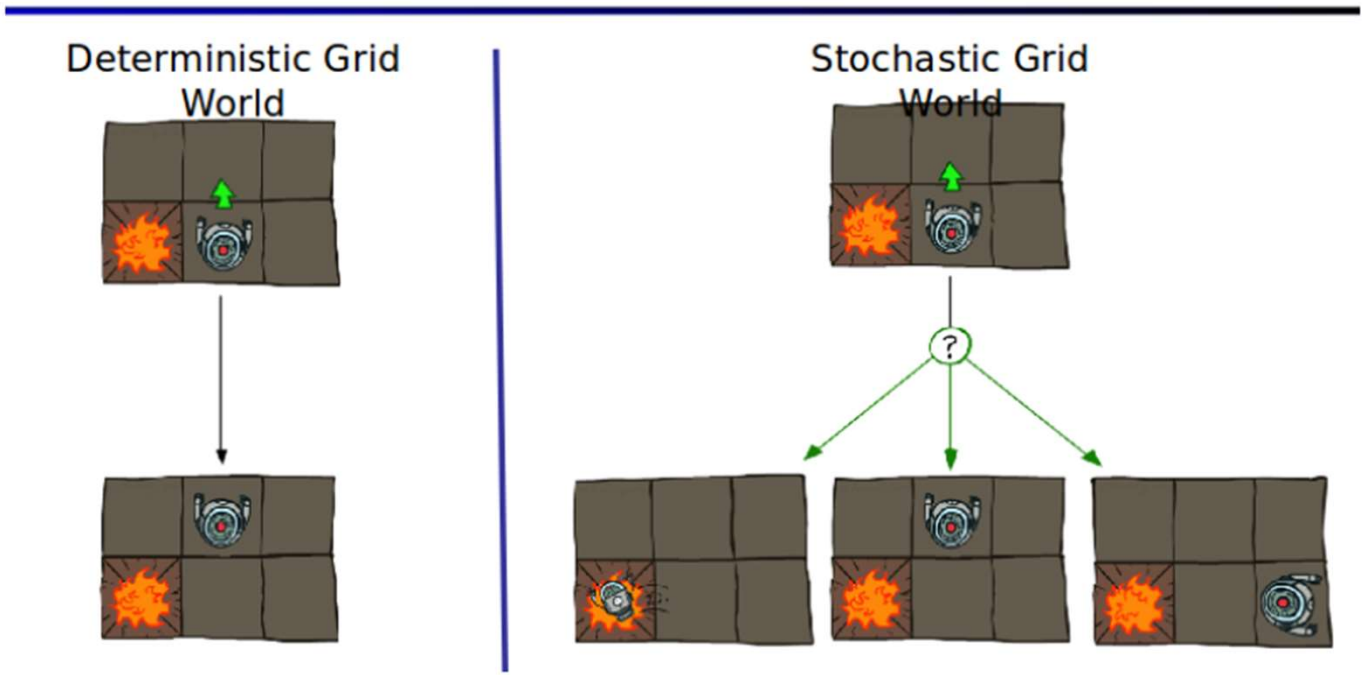
강화 학습

강화 학습 3요소

- 상태 (State)
  - Agent(주인공)은 어떤 '상태'에 있을 수 있다
  - Ex) 게임에서의 위치, 바둑에서의 바둑판 상황
- 행동 (Action)
  - 어떤 '상태'에서는 어떤 '행동'을 통해 행동에 해당하는 다음 '상태'로 이동할 수 있다.
  - Ex) 왼쪽으로 이동, 오른쪽으로 이동, (4,13)위치에 바둑돌 놓기
- 보상 (Reward)
  - 어떤 '행동'을 취하면 보상이 따른다.
  - 슈퍼마리오가 동전을 먹으면 점수가 올라간다

강화 학습

강화 학습

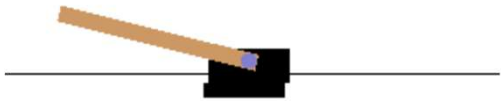


강화 학습의 기본 원리

Observation

현재 Agent의 상태를 파악

이미지 그 자체를 이용해 관찰할 수도, 특징 점들의 집합이 될 수도

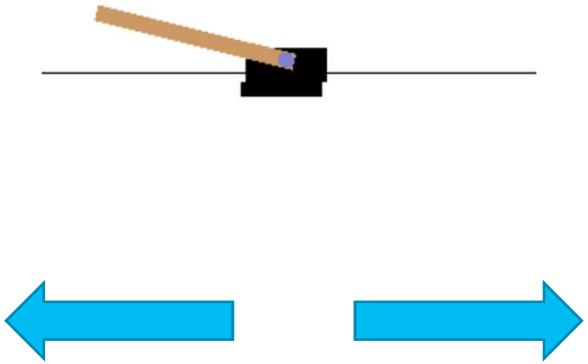


강화 학습의 기본 원리

Action Space

가능한 Action의 집합

검정 카트의 현재 상황에서 가능한 Action은 카트를 왼쪽, 오른쪽으로 움직이는 것



강화 학습의 기본 원리

Q-table

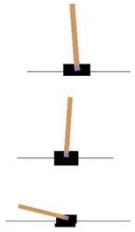
Q-value는 각 State에서의 선택 가능한 Action에 따라 받을 것으로 기대되는 보상

Q-value는 state와, action에 따라 정해진다. (Q(state, action))

Q-value는 보상 그 자체가 아닌 보상의 "기댓값"으로, Action을 선택하는 기준이 된다.

State에서 Action을 선택할 때 참고한다 (가장 큰 Q-value를 가지는 Action을 선택한다)

Q-value는 Q-Table에 기록된다

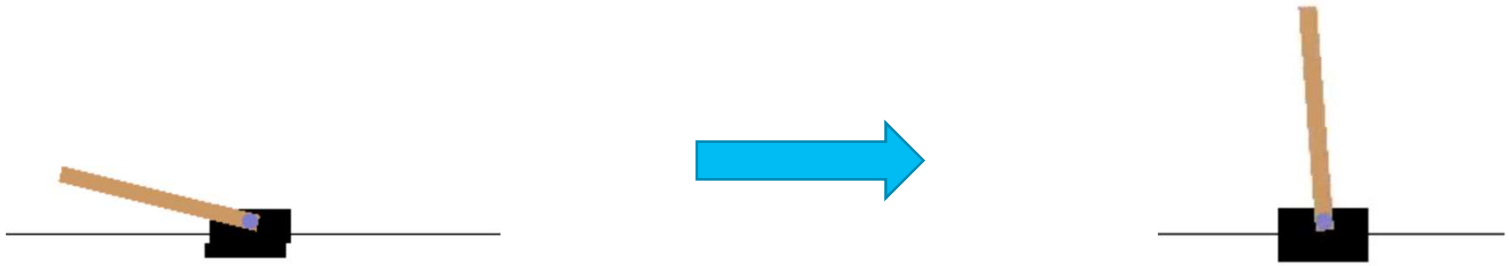


State ₩ Action	<-	->
State 1	0.5	0.3
State 2	-0.3	0.7
State3	1.8	-2.1

강화 학습의 기본 원리

Select Action

현재 state에서, Q-value값이 가장 큰 Action을 찾아, 그 Action을 행하고 다음 state로 이동한다!



Action ⇦ 선택

강화 학습의 기본 원리

강화 학습 알고리즘

알고리즘 [편집]

수식으로 표현하면, 강화 학습 모델은 다음과 같이 구성된다.

1. 환경 상태 집합, S;
2. 행동 집합, A;
3. 포상($\in \mathbb{R}$)의 집합;.

매 시점 t에 에이전트는 자신의 상태(state) $s_t \in S$와 가능한 행동(action) $A(s_t)$를 가지고 있다.

에이전트는 어떤 행동 $a \in A(s_t)$ 을 취하고, 환경으로부터 새로운 상태 s_{t+1}와 포상(reward) r_{t+1}을 받는다. 이 상호작용에 기반해서 강화 학습 에이전트는 누적된 포상값 R을 최대화 하는 정책(policy) π: $S \to A$을 개발한다.

종료 상태(terminal state)가 존재하는 MDPs에서는 $R = r_0 + r_1 + \cdots + r_n = \sum_{t=1}^{n} r_t$이고, 그렇지 않은 MDPs에서는 $R = \sum_{t=1}^{n} \gamma^t r_t$가 된다. 여기서 γ는 미래의 포상이 현재에 얼마나 가치 있는지를 표현하는 할인율(discount factor)로 0과 1사이의 값이다.

- Agent가 관찰을 통해 자신의 '상태'를 파악
- 현재 '상태'에서 가능한 '행동' 중 어떤 '행동'을 취해야 '보상'이 가장 큰 지를 살피고
- 가장 큰 '보상' 을 주는 '행동'을 통해 다음 '상태'로 이동한다.
- 최종 목표는 보상의 총합을 최대화 하는 optimal policy를 찾는 것!
- 현재 '상태' 이전의 '상태'와 '행동'들에는 영향을 받지 않는다

Q 러닝

그렇다면 Q-value는 어떻게 알지?

강화 학습을 통해 Q-value를 학습하자!

Q 러닝

$$Q : S \times A \to \mathbb{R}$$

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} \right)$$

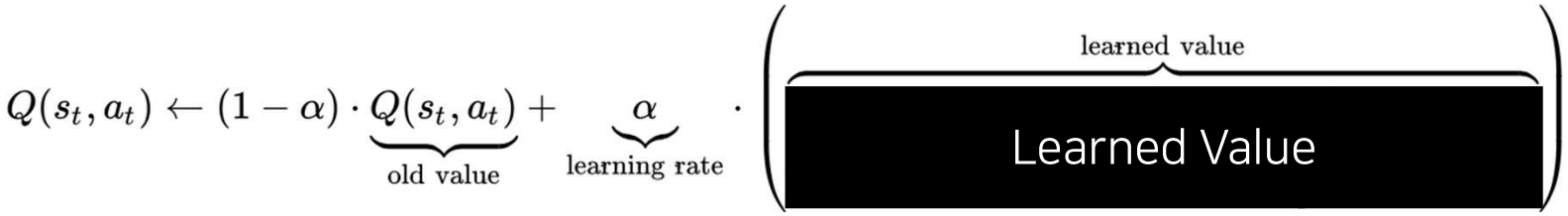
매 step마다

현재의 Q-value를 참고해 Action을 선택하고

다음 state로 이동하면서

그리고 동시에 Q-value를 update한다 (Q-value를 학습)

Q 러닝



Q-value의 update 계산에 담긴 의미

- Old value와 learned value를 적정 비율로 섞는다 (learning rate로 조절)
  - α는 0과 1 사이의 수

Q 러닝

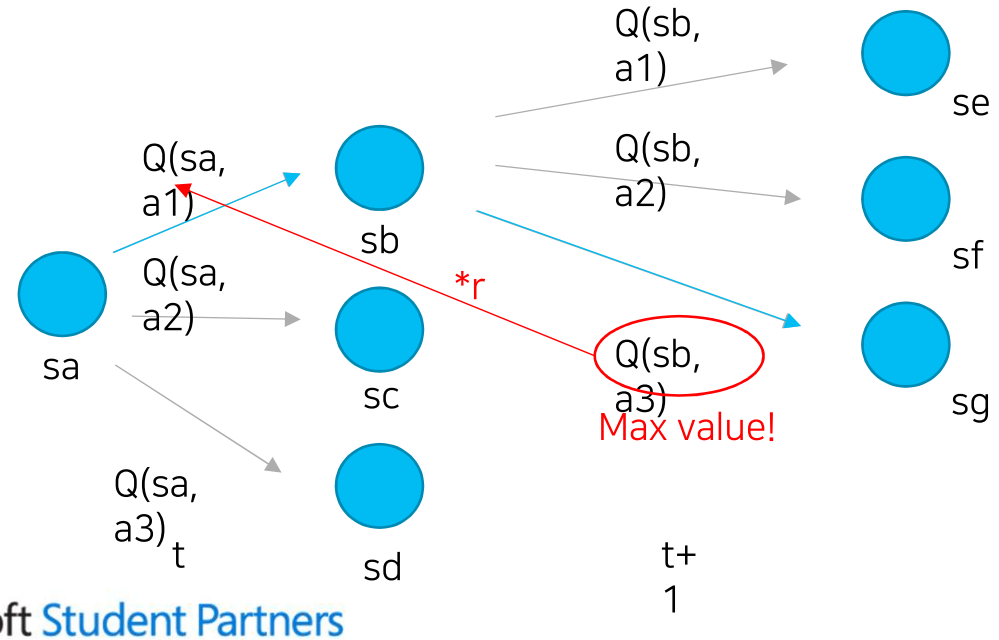
$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \text{Expected Reward}}^{\text{learned value}} \right)$$

Q-value의 update 계산에 담긴 의미

- Old value와 learned value를 적정 비율로 섞는다 (learning rate로 조절)
- learned value
  - Action을 취하면 지금 당장 받는 보상(reward)
  - 그리고 미래에 받을 것으로 기대되는 보상 (expected reward)가 고려 대상이다

Q 러닝

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} \right)$$



해당 Action을 선택한다면 앞으로 받을 수 있는 보상의 기댓값

State sa에서 Action a1을 선택했을 때 State sb에 도달한다고 하면

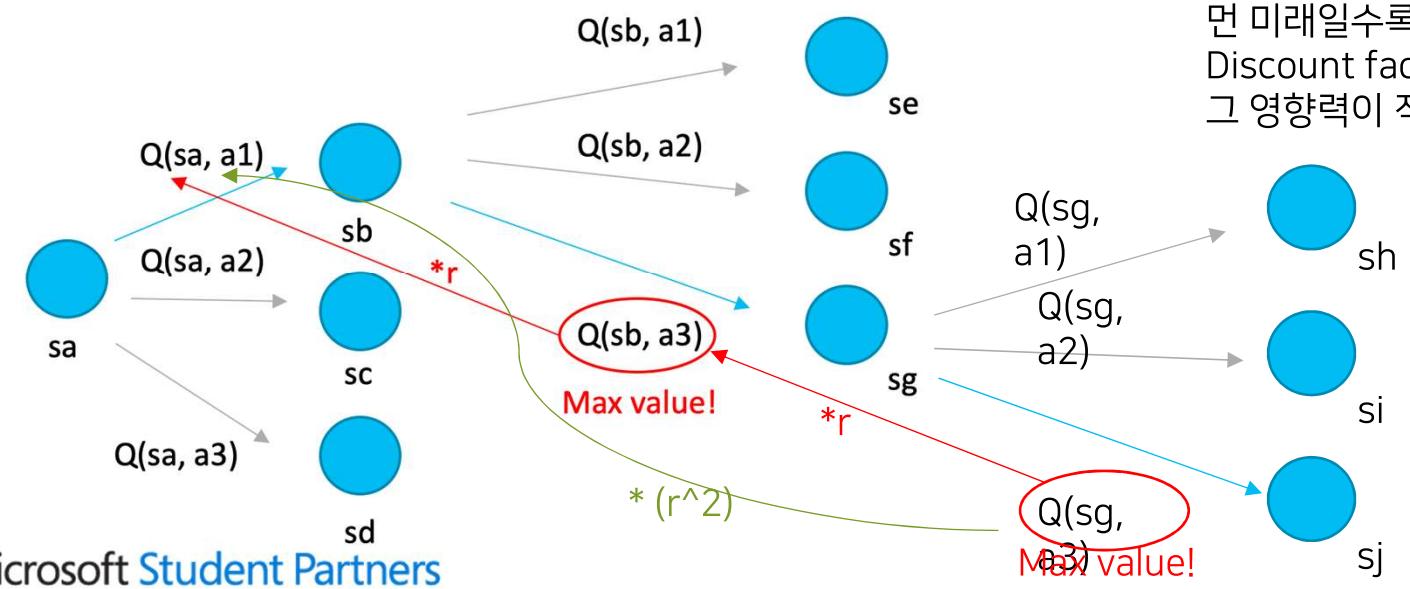
State sb에서 선택할 것으로 기대되는 Q(sb, a3)가 Q(sa, a1)에 반영된다

sa에서 a1을 택한다면
sb에서 a3을 택했을 때의
기대되는 보상(Q-value)도
함께 기대할 수 있다!

Q 러닝

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} \right)$$

더 미래의 Q-value도 반영이 되나,
먼 미래일수록
Discount factor배 만큼
그 영향력이 작아진다



Q 러닝

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

Q-value의 update 계산에 담긴 의미

- Old value와 learned value를 적정 비율로 섞는다 (learning rate로 조절)
- learned value
  - 현재의 보상과 미래의 보상으로 이루어진 보상의 기댓값
  - 미래의 Q-value는 먼 미래일수록 그 영향력이 적다 (discount factor)

Episode와 Step

강화 학습 알고리즘

- Simulation
  - 환경에 대한 학습을 시작해서
  - 환경에 대해 배워가는 전 과정
  - Q-value는 초기값을 가진다
  - 새로운 게임 캐릭터를 키워서 만렙에 도달하기까지

Episode와 Step

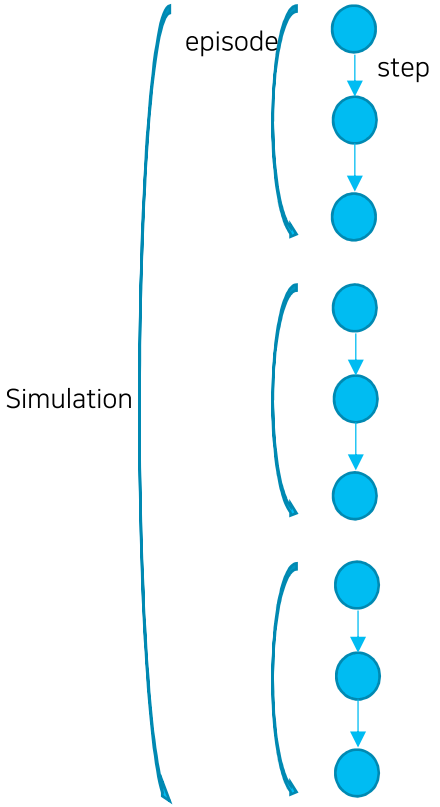
강화 학습 알고리즘

- Episode
  - 가장 처음의 State (t=0)로 부터 시작
  - 매 Step마다 Action을 통해 State를 이동
  - "종료 State"에 도달하면 끝이 남
  - State, Action, Reward의 연속
  - Q-table은 초기화 하지 않고 계속 학습
  - 하나의 simulation은 여러 번의 episode를 통해 학습을 한다
  - 게임 한 판!

Episode와 Step

강화 학습 알고리즘

- Step
  - State에서 Q-table을 참고해 Action을 선택해 다음 state로 이동하는 과정
  - 매 step마다 Q-value를 update한다
  - 하나의 episode는 여러 번의 step으로 이루어진다



강화 학습의 적용 사례

- 자율 주행 차

- 게임

Q & A

Q & A

Microsoft Student Partners

Microsoft

자료 출처

- One fourth Lab (https://medium.com/@onefourthlabs)
- 모두의 연구소