Reinforcement Learning introduction_2

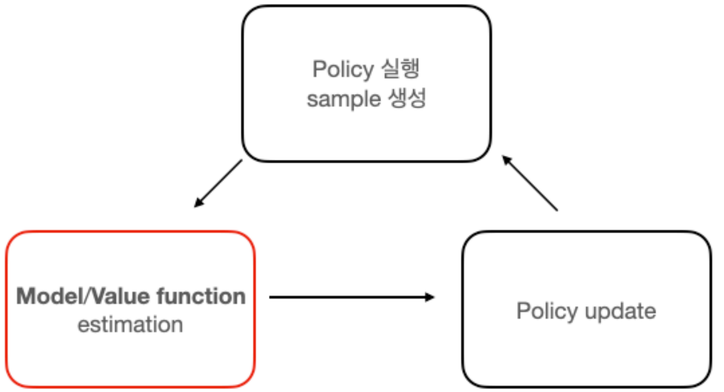
정규헌

Reinforcement Learning

이전 내용 요약

- **강화학습의 특징**
  - **순차적의사 결정 문제 :** 행동을 하고, 환경이 변하고, 또 행동을 하고..
  - **Reward hypothesis :** reward 누적 합을 최대화 시키는 문제
- **Reward**
- **Environment**
- **Agent**
  - **Policy :** Agent의 Action을 결정
  - **Value function :** Agent의 state에 따른 총 reward의 기댓값
  - **Model :** Agent의 내부에서 환경이 어떻게 변할지 예측

강화학습의 공통적인 iteration

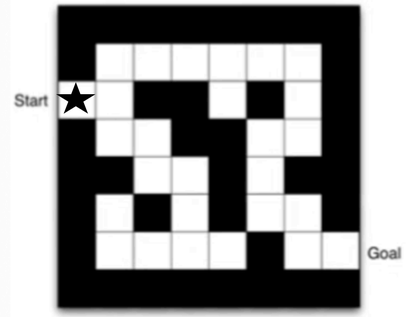


Reinforcement Learning

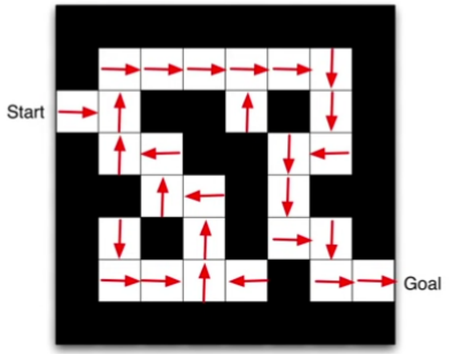
이전 내용 요약

- **강화학습의 특징**
  - **순차적의사 결정 문제 :** 행동을 하고, 환경이 변하고, 또 행동을 하고..
  - **Reward hypothesis :** reward 누적 합을 최대화 시키는 문제
- **Reward**
- **Environment**
- **Agent**
  - **Policy :** Agent의 Action을 결정
  - **Value function :** Agent의 state에 따른 총 reward의 기댓값
  - **Model :** Agent의 내부에서 환경이 어떻게 변할지 예측

목표 : Start로 부터 Goal 지점까지 한 칸씩 이동

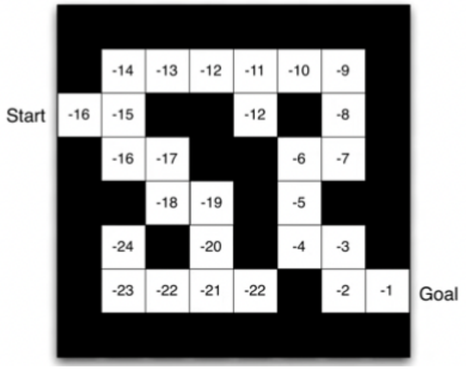


- Rewards: -1 per time-step
- Actions: N, E, S, W
- States: Agent's location



Policy

화살표 : Agent의 최적 Policy



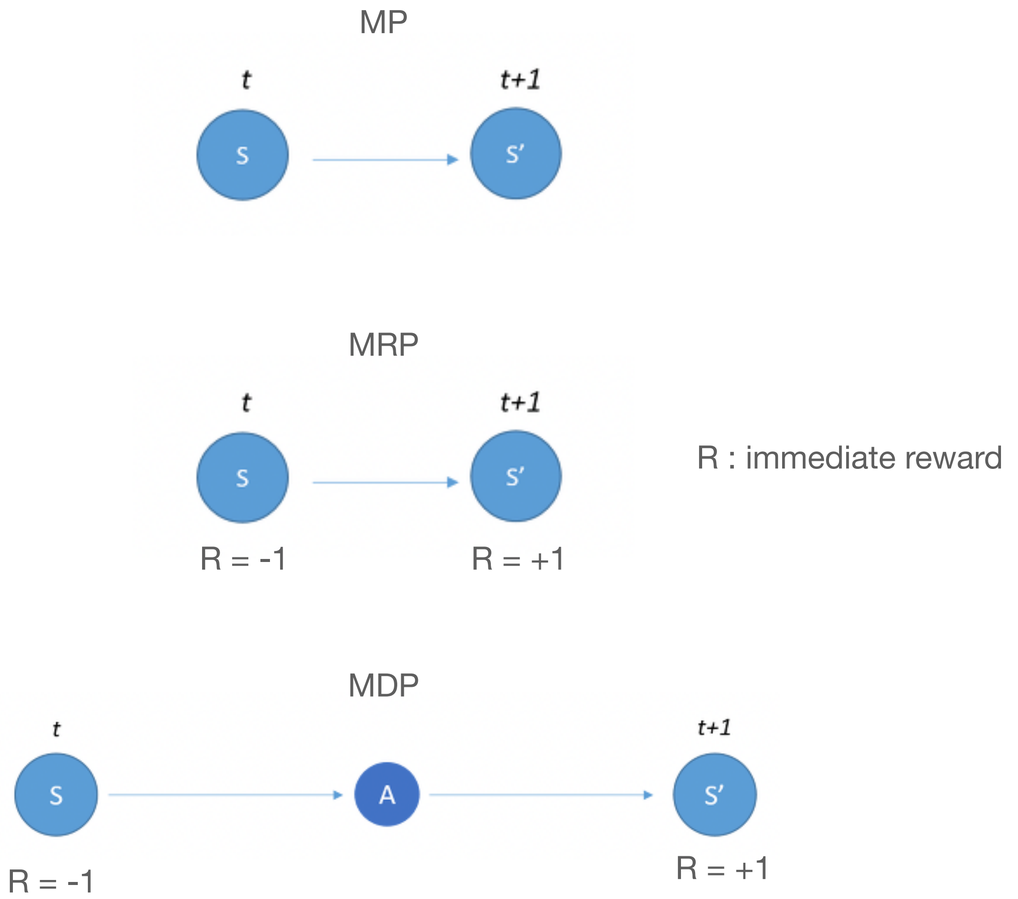
Value function

Value function을 통해 얻은
Agent의 state(위치)에 따른 누적 reward 기대값

Reinforcement Learning

MDP(Markov Decision Process)

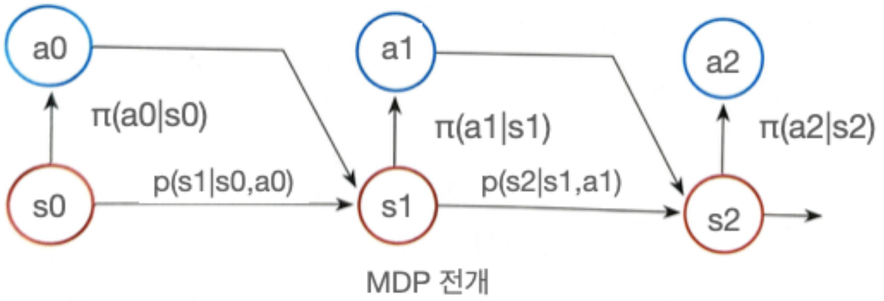
- **강화학습은 순차적 의사결정 문제를 푸는 것 !**
  - 순차적 의사결정 문제를 **수학적 모델 MDP**로 표현
- **MP**(Markov process/Markov Chain)
  - 미래의 state는 바로 직전 state에만 영향을 받음
- **MRP**(Markov Reward Process) : MP에 reward 개념이 추가
- **MDP : MRP에 Action 추가**
  - 미래의 state는 바로 직전 state + action에만 영향을 받음
  - state가 있다면 history는 필요 없다
  - state는 미래에 대한 충분한 표현형이다



Reinforcement Learning

MDP(Markov Decision Process)

- 강화학습은 **MDP** 문제를 푸는것
  - MDP에서 Reward를 Maximize 하는 것
  - **policy** (정책)
    - π(at | st) = P(at | st)
  - **State transition probability**(상태 천이 확률) - transition
    - 이전 state, action을 취할때 다음 state로 이동할 확률
    - P(st | s(t-1), a(t-1))
- MDP의 수학적 특징
  1. *P(a1 | s0, a0, s1) = P(a1 | s1)*
  2. *P(s2 | s0,a0, s1,a1) = P(s2 | s1,a1)*

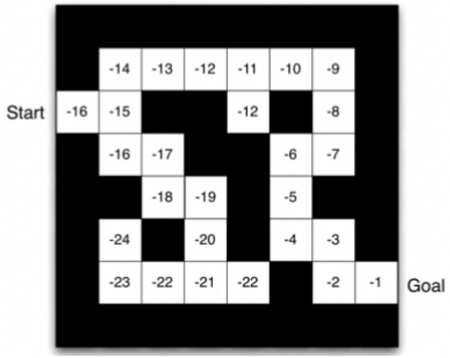


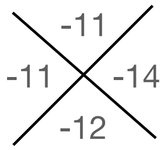
MDP 전개

s0,a0,s1,a1,s2,a2, ……

Reinforcement Learning

MDP(Markov Decision Process)





- **Return = Expected Reward**
- **Value function** in MDP
  - **State-Value function :** 현재 state의 가치
    - 현재 state부터 기대되는 Return
  - **Action-Value function :** 현재 action의 가치
    - 현재 Action으로 부터 기대되는 Return
- **Optimal Policy**
  - **state-value function을 최대**로 하는 policy
  - 현재 state부터 기대되는 Return을 최대로 하는 policy

Discount factor

Return $G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+1} + \dots$

$$E[G(t)] = \int G(t)p(t)dx$$

State-value function : 현재 state St부터 모든 action을 다했을 때 기대되는 Return

$$V(S_t) = \int_{a_t:a_\infty} G_t p(a_t, S_{t+1}, a_{t+1}, \dots | S_t) da_t : a_\infty$$

action-value function : St에서 action at를 했을 때부터 기대되는 Return

$$Q(S_t, a_t) = \int_{S_{t+1}:a_\infty} G_t p(S_{t+1}, a_{t+1}, S_{t+2}, a_{t+2}, \dots | S_t, a_t) ds_{t+1} : a_\infty$$

Optimal Policy

$V(S_t)$**를 maximize 하는 Policy** $P(a_t|S_t), P(a_{t+1}|S_{t+1}), P(a_\infty|S_\infty)$

Reinforcement Learning

Bellman equation

- **벨만 방정식**
  - 강화학습에서 자주 등장하는 방정식
  - V와 Q사이의 관계를 방정식으로 표현
    - **V를 Q로 표현**
    - **V를 next V로 표현**
    - Q를 next V로 표현
    - Q를 next Q로 표현
  - **Q-Learning, SARSA의 motive**

Bayesian rule $p(x, y) = p(x|y)p(y)$ $p(x, y|z) = p(x|y, z)p(y|z)$

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+1} + \ldots = R_t + \gamma G_{t+1}$$

action-value function : St에서 action at를 했을 때부터 기대되는 Return

$$Q(S_t, a_t) = \int_{S_{t+1}:a_\infty} G_t p(S_{t+1}, a_{t+1}, S_{t+2}, a_{t+2}, \ldots | S_t, a_t) ds_{t+1} : a_\infty$$

State-value function : 현재 state St부터 모든 action을 다했을 때 기대되는 Return

$$V(S_t) = \int_{a_t:a_\infty} G_t \frac{p(a_t, S_{t+1}, a_{t+1}, \ldots | S_t)}{p(S_{t+1}, a_{t+1}, \ldots | S_t, a_t)p(a_t|S_t)} d_{a_t:a_\infty}$$

$$p(a_{t+1}, \ldots | S_t, a_t, S_{t+1})p(a_t, S_{t+1}|S_t)$$

$$V(S_t) = \int_{a_t} \int_{S_{t+1}:a_\infty} G_t p(S_{t+1}, a_{t+1}, \ldots | S_t, a_t) d_{S_{t+1}:a_\infty} p(a_t|S_t) da_t$$

State-value function
= action-value function의 기댓값

V를 Q로 표현 $$V(S_t) = \int_{a_t} Q(S_t, a_t) p(a_t|S_t) da_t$$

$$V(S_t) = \int_{a_t, S_{t+1}} \int_{a_{t+1}:a_\infty} (R_t + \gamma G_{t+1}) p(a_{t+1}, \ldots | S_{t+1}) d_{a_{t+1}:a_\infty} p(a_t, S_{t+1}|S_t) d_{a_t, S_{t+1}}$$

V를 next V로 표현 $$V(S_t) = \int_{a_t, S_{t+1}} (R_t + \gamma V(S_{t+1})) p(a_t, S_{t+1}|S_t) d_{a_t, S_{t+1}}$$

$$p(S_t + 1|S_t, a_t) p(a_t|S_t)$$

Transition : environment에서 주어짐

Policy : 전체를 maximize하는 policy를 찾아야 함

Reinforcement Learning

Bellman equation

- **벨만 방정식**
  - 강화학습에서 자주 등장하는 방정식
  - V와 Q사이의 관계를 방정식으로 표현
    - V를 Q로 표현
    - V를 next V로 표현
    - **Q를 next V로 표현**
    - **Q를 next Q로 표현**
  - **Q-Learning, SARSA의 motive**

Bayesian rule $p(x, y) = p(x|y)p(y)$ $p(x, y|z) = p(x|y, z)p(y|z)$

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+1} + \ldots = R_t + \gamma G_{t+1}$$

State-value function : 현재 state St부터 모든 action을 다했을 때 기대되는 Return

$$V(S_t) = \int_{a_t:a_\infty} G_t \frac{p(a_t, S_{t+1}, a_{t+1}, \ldots | S_t)}{p(S_{t+1}, a_{t+1}, \ldots | S_t, a_t)p(a_t|S_t)} d_{a_t:a_\infty}$$

$$p(a_{t+1}, \ldots | S_t, a_t, S_{t+1})p(a_t, S_{t+1}|S_t)$$

action-value function : St에서 action at를 했을 때부터 기대되는 Return

$$Q(S_t, a_t) = \int_{S_{t+1}:a_\infty} G_t \frac{p(S_{t+1}, a_{t+1}, S_{t+2}, a_{t+2}, \ldots | S_t, a_t)}{p(a_{t+1}, S_{t+2}, a_{t+2}, \ldots | S_t, a_t, S_{t+1})p(S_t|S_{t+1}, a_t)} ds_{t+1} : a_\infty$$

$$Q(S_t, a_t) = \int_{S_{t+1}} \int_{a_{t+1}:a_\infty} (R_t + \gamma G_{t+1}) p(a_{t+1} : a_\infty | S_{t+1}) d_{a_{t+1}:a_\infty} p(S_{t+1}|S_t, a_t) d_{S_{t+1}}$$

$$Q(S_t, a_t) = \int_{S_{t+1}} (R_t + \gamma V(S_{t+1})) p(S_{t+1}|S_t, a_t) d_{S_{t+1}}$$ Q를 next V로 표현

$$Q(S_t, a_t) = \int_{S_{t+1}:a_\infty} G_t \frac{p(S_{t+1}, a_{t+1}, S_{t+2}, a_{t+2}, \ldots | S_t, a_t)}{p(S_{t+2} : a_\infty | S_t, a_t, S_{t+1}, a_{t+1})p(S_{t+1}, a_{t+1}|S_t, a_t)} ds_{t+1} : a_\infty$$

$$Q(S_t, a_t) = \int_{S_t+1,a_{t+1}} \int_{S_{t+2}:a_\infty} (R_t + \gamma G_{t+1}) p(S_{t+2} : a_\infty | S_{t+1}, a_{t+1}) d_{S_{t+2}:a_\infty} p(S_{t+1}, a_{t+1}|S_t, a_t) d_{S_{t+1},a_{t+1}}$$

$$Q(S_t, a_t) = \int_{S_t+1,a_{t+1}} (R_t + \gamma Q(S_{t+1}, a_{t+1})) p(S_{t+1}, a_{t+1}|S_t, a_t) d_{S_{t+1},a_{t+1}}$$ Q를 next Q로 표현

$$Q(S_t, a_t) = \int_{S_t+1,a_{t+1}} (R_t + \gamma Q(S_{t+1}, a_{t+1})) p(a_{t+1}|S_t, a_t, S_{t+1}) p(S_{t+1}|S_t, a_t) d_{S_{t+1},a_{t+1}}$$

Policy : 전체를 maximize하는 policy를 찾아야 함

Transition : environment에서 주어짐

Reinforcement Learning

Bellman equation

- **벨만 방정식**
  - V와 Q사이의 관계를 방정식으로 표현
    - V를 Q로 표현
    - V를 next V로 표현
    - **Q를 next V로 표현**
    - **Q를 next Q로 표현**

Reinforcement learning

Optimal Policy

- **state-value function을 최대**로 하는 policy
  - **현재**부터 기대되는 Return을 최대화
  - Optimal Q를 구했다고 가정했을때
  - Optimal Q를 maximize하는 action을 선택하는것
    - 디랙 델타함수 - 확률밀도 함수 - greedy

$$\delta(x) = \begin{cases} \infty & (x = 0) \\ 0 & (x \neq 0) \end{cases}$$

넓이가 1

$$\int_{-\infty}^{\infty} \delta(x)\,\mathrm{d}x = \int_{t_0-\epsilon}^{t_0+\epsilon} \delta(t - t_0)\,\mathrm{d}t = 1 \quad (\epsilon > 0)$$

action-value function :
St에서 action at를 했을 때부터 기대되는 Return

$$Q(S_t, a_t) = \int_{S_{t+1}:a_\infty} G_t p(S_{t+1}, a_{t+1}, S_{t+2}, a_{t+2}, \ldots | S_t, a_t) ds_{t+1} : a_\infty$$

Optimal Policy $p^*(a_{t+1}|S_{t+1}), p^*(a_{t+2}|S_{t+2}), \ldots$ 포함되어 있음

$V(S_t)$**를 maximize 하는 Policy** $P(a_t|S_t), P(a_{t+1}|S_{t+1}), P(a_\infty|S_\infty)$

State-value function :
현재 state St부터 모든 action을 다했을 때 기대되는 Return

$$V(S_t) = \int_{a_t:a_\infty} G_t p(a_t, S_{t+1}, a_{t+1}, \ldots | S_t) d_{a_t:a_\infty}$$

Maximize 시켜야함

$$V(S_t) = \int_{a_t} Q^*(S_t, a_t) p(a_t|S_t) da_t$$

Policy

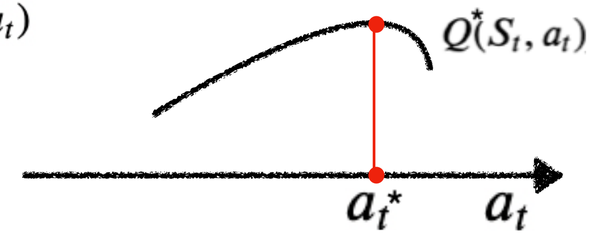
$\underset{p(a_t|S_t)}{argmax}$

$p^*(a_{t+1}|S_{t+1}), p^*(a_{t+2}|S_{t+2}), \ldots$

Optimal을 모두 구했다고 가정

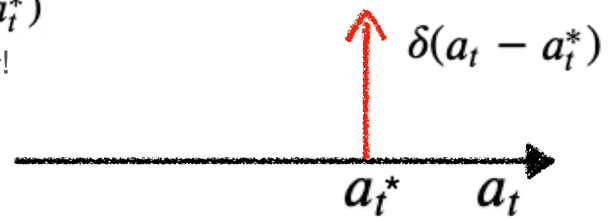
$$a_t^* = \underset{a_t}{argmax} Q^*(S_t, a_t)$$

Greedy action하는 이유
저게 가장 좋은 policy니까



$$p^*(a_t|S_t) = \delta(a_t - a_t^*)$$

Optimal action만 골라라!



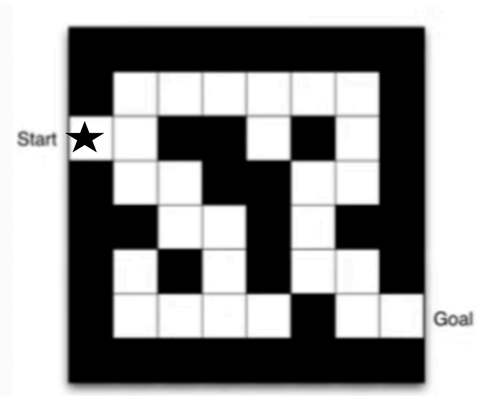
Reinforcement learning
정리

- 강화학습은 순차적 의사결정 문제 -> MDP를 통해 수식화
- Value function
  - State-value function V
  - Action-value function Q
- Bellman equation
  - V와 Q의 관계를 방정식으로 표현
- Optimal Policy
  - delta function - greedy action

Reinforcement learning

Optimal Q - Q*

- Optimal Policy를 어떻게 구하는지 알았는데
- Q*가 있다고 가정했었는데 이건 어떻게 구할까?
  - episode를 진행하면서 Q를 계속해서 update
  - 최종적으로는 Q*를 만들고 greedy action
- 다양한 method
  - MC(Monte Carlo)
  - TD(Temporal Difference)
    - SARSA
  - Q-learning



- Rewards: -1 per time-step
- Actions: N, E, S, W
- States: Agent's location

다음시간 다룰내용

- Q* 를 구하는 방법? - 수학적 의미
  - Monte Carlo
  - Temporal difference (TD)
  - SARSA
  - Q-learning
- Q-learning -> DQN

감사합니다