

1. About Reinforcement Learning

1-1. Target of Reinforcement Learning

1-2. Characteristic of Reinforcement Learning

2. The Reinforcement Learning Problem

2-1. Rewards

2-2. Environments

2-3. States

2-3-1. History and State

2-3-2. Environment State

2-3-3. (Agent) State

2-3-4. Information state(=Markov state)

2-3-5. Fully or Partially Observable Environments (MDP / POMDP)

3. Inside an Reinforcement Learning Agent

3-1. Policy

3-2. State Value Function

3-3. Model

3-4. Maze Example

3-5. Categorizing Reinforcement Learning Agents

4. Problems within Reinforcement Learning

4-1. Learning and Planning

4-2. Exploration and Exploitation

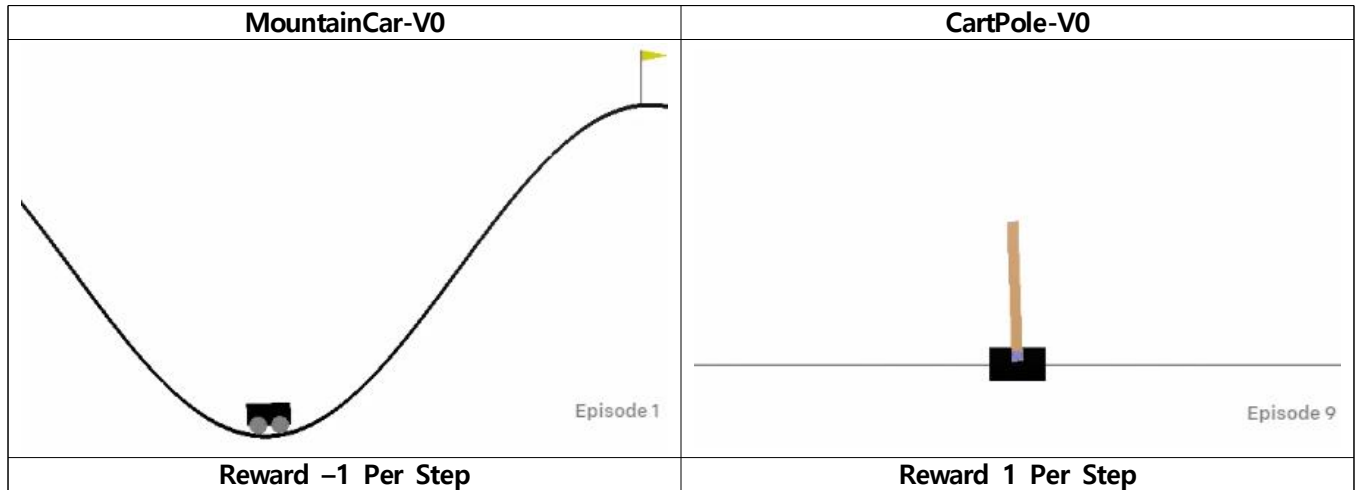
4-3. Prediction and Control

5. Summary of Reinforcement Learning Flow

1. About Reinforcement Learning

1-1. Target of Reinforcement Learning

#. 일반적인 Reinforcement Learning의 목표는 주어진 환경 내에서 Reward의 총합을 최대화하는 것이다. 예를 들어 아래와 같은 환경에서는 Action마다 Reward를 받기 때문에 Episode내에서 Action을 하면서 받은 Reward의 합이 최대화 되도록 Action Sequence를 생성한다.



- a. 환경을 직접 구현하는 경우 위의 예시와 같이 환경이 요구하는 것이 무엇인지에 따라 Reward 설계를 다르게 해야 한다. 즉 전자의 경우에는 '목표 도달'을 통한 Reward 총합의 극대화이면, 후자의 경우에는 단지 'Reward 총합의 극대화'가 목표이다.

1-2. Characteristic of Reinforcement Learning

#. Reinforcement Learning의 특징을 통해 기존 기계학습과 다른 점을 알아보자.

- a. Supervisor 없이 Scalar 형식인 Reward만 존재한다. 이 Reward 또한 어떻게 해야 얻는지 Trial and Error를 통해 Agent가 알 수 있다.
- b. Reward가 즉각 나타나지 않을 수 있다. 때문에 어떤 Action이 해당 Reward를 야기했는지 모를 수 있다.
- c. 강화학습은 순차행동결정 문제이므로 시간에 따른 Action의 순서가 중요하다.
- d. Agent의 Action에 따라 받는 Data가 달라진다. (Agent의 Action에 따라 Env.가 Agent에게 Return하는 Data가 존재함.)

2. The Reinforcement Learning Problem

2-1. Rewards

- a. R_t 로 표현되고 Time step t 에서 Agent가 얼마나 잘하고 있는지를 표현하는 Scalar 지표이다.
- b. Reward를 여러 값들의 가중치 합으로 하나의 Scalar로도 표현할 수 있다. (Vector표기 불가)
- c. 강화학습의 목적은 순차행동결정문제 과정 속에서 축적된 Long-Term Reward의 합을 극대화하는 Optimal Policy를 도출하는 것이다.
- d. Long-Term을 고려해야하기에 눈앞의 Reward가 없거나 작을 수도 있다.

2-2. Environments

- a. Agent 외부에 있는 것들이 모두 Environment이다.
- b. Agent와 상호작용을 하며 Time Step마다 Agent에게 Action을 받고 Reward, Observation(=Next_State)을 준다.

2-3. States

2-3-1. History and State (Trajectory and Transition)

- a. History는 $H_t = S_1, R_1, A_1, \dots, A_{t-1}, S_t, R_t$ 로 표현되고 Env.와 Agent 상호작용의 모든 기록이다.
- b. Observation¹⁾ O_t 는 State S_t 의 구성요소이다. 즉 $S_t = \{O_0, O_1, \dots, O_t\}$ 으로 표시한다.
- c. Trajectory는 $\{(S_0, A_0, R_0, Done), \dots, (S_t, A_t, R_t, Done)\}$ 이며, Transition은 $(S_t, A_t, R_t, Done)$ 을 의미한다.
- d. Agent는 History를 보고 다음 Action을 결정한다.
- e. $S_t = f(H_t)$ 가 State이며 History를 가공하여 Agent, Env.가 무엇을 할지 근거하는 수치이다. 즉 Agent는 State를 이용하여 Action을 판단하고, Env.는 State와 Action을 이용하여 Reward, Observations(=Next_State)를 계산하고 Agent에게 Return 한다.

2-3-2. Environment State

- a. S_t^e 로 표현되며 다음 State에서 Observations, Reward 계산을 위한 재료값이다.
- b. 예를 들어 게임 화면이 전환되기 위한 계산 같은 것이다. 따라서 Agent에겐 보이지도 않고 보이더라도 불필요한 정보가 섞여 있다.

서강대학교 머신러닝 연구실

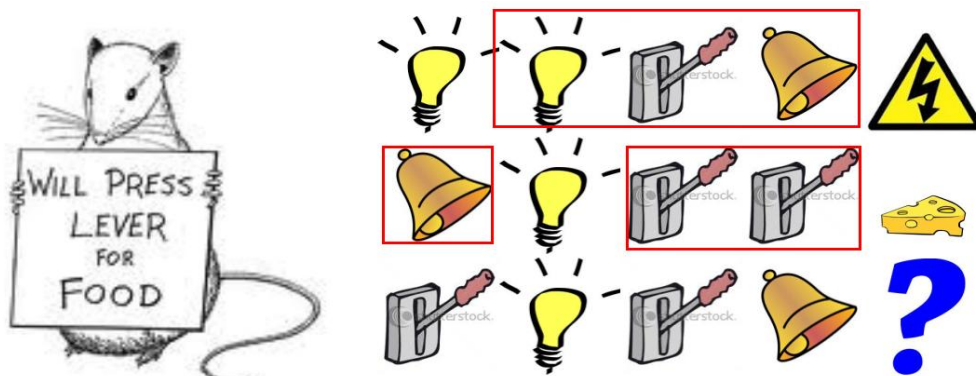
1) Image를 Observation으로 설정하는 경우 Agent의 전 움직임과 위치를 반드시 고려해야 한다.

2-3-3. (Agent) State

- S_t^a 로 표현되며 다음 Action을 선택하기 위해 필요한 정보다.
- Agent State는 자율적으로 선택해서 사용할 수 있다. 예를 들어 주식 분석을 하는데 주가, 거래량, 재무제표 중 원하는 것만 조합해서 State로 사용할 수 있다.

2-3-4. Information state (Markov state)

- State가 Markov 하다는 것은 $P[S_{t+1}|S_t] = P[S_{t+1}|S_1, \dots, S_t]$ 와 필요충분조건이다. 즉 다음 State로 넘어가는 확률을 구할 때, 현재 State만 있으면 된다는 것이다.
- 다른 방식으로 표현하면, '현재가 주어졌으면, 과거는 버려도 된다.' 의 의미이다.
- State를 어떻게 정의하느냐에 따라 Markov State가 성립할 수도 아닐 수도 있지만, 강화학습은 Markov State만 다룬다.
- Environment State는 결국 Markov State이다.
- Rat Example : State 정의의 다양성에 따라 예측 또한 달라진다.



2-3-5. Fully or Partially Observable Environments (MDP / POMDP)

- Agent가 Env.의 State를 모두 파악할 수 있는지에 따라 달라진다.
- MDP : $S_t^a = S_t^e = O_t$ 이 성립하며 Agent는 Env. State를 완전히 알고 있는 상황이다.
- POMDP : $S_t^a \neq S_t^e$ 이며, Agent는 Env. State를 부분적으로만 관찰가능한 상황이다. 이 경우 Agent는 자신의 State를 정의해야 하는데 아래와 같이 다양하게 정의할 수 있다.

Complete History	$S_t^a = H_t$
Beliefs of Environment state	$S_t^a = (P[S_t^e = s^1], \dots, P[S_t^e = s^n])$
RNN	$S_t^a = \sigma(S_{t-1}^a W_s + O_t W_o)$

3. Inside an Reinforcement Learning Agent

#. Agent의 구성요소는 Policy(π), (State) Value Function(v), Model(M)이 있는데 다 있을 수도, 하나만 있을 수도 있다.

3-1. Policy(π)

a. Agent가 어떤 State에 있을 때, 행동을 규정해주는 함수이다. 즉 State와 Action을 Mapping해준다.

b. 종류

Deterministic	action을 반환한다. $a = \pi(s)$
Stochastic	각 action별 확률 값을 반환한다. $\pi(a s) = P[A_t = a S_t = s]$

3-2. State Value Function

a. 어떤 State가 얼마나 좋은지 Agent가 평가하는 지표이며 해당 Episode가 끝날 때까지, State Value Function의 Parameter인 Policy를 따랐을 때, 받을 수 있는 Cumulated Discounted Rewards의 기댓값으로 표현된다. 즉 주어진 Policy를 따라 Episode를 충분히 많이 Sampling(직접 경험)하면 G_t 가 각각 나오는데, 이들의 평균값이다.

$$v_{\pi}(s) = E_{\pi}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s]$$

b. Parameter인 Policy가 Stochastic한 경우 각 State에서 모든 Action에 대한 Rewards를 고려해주어야 하지만 Deterministic이면 Action 하나만 고려해준다. 그렇다고 해서 더 쉬워지는 것은 아니다. 왜냐하면 State Transition Probability를 고려해야 하기 때문이다.

3-3. Model

a. Model은 Agent가 예측한 환경을 의미하며, Model은 Reward와 State Transition Probability로 구성되어 있다. 따라서 Agent가 이 두 가지를 Env.와 Interaction하면서 예측하는 것이다.

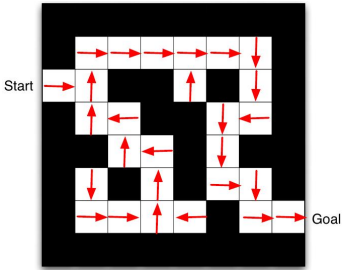
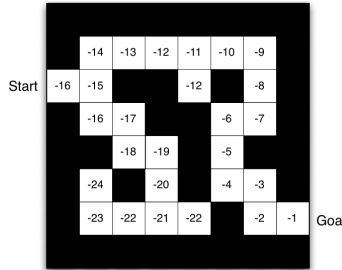
$$P_{ss'}^a = P[S_{t+1} = s' | S_t = s, A_t = a]$$

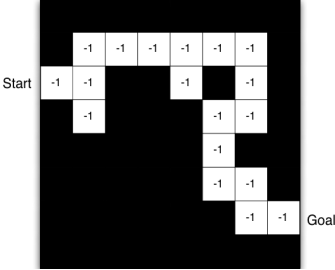
$$R_s^a = E[R_{t+1} | S_t = s, A_t = a]$$

b. Model은 Agent가 최적의 Policy를 찾기 위해 Env.와 Interaction하며 쓰일 수도(Model-Based), 쓰이지 않을 수도(Model-Free) 있다.

3-4. Maze Example

a. 학습이 완료되어 Optimal한 경우이고 State Transition Probability = 1이다.

Policy : <ul style="list-style-type: none"> State를 넣었을 때, 어떤 Action ? 	State Value Function : <ul style="list-style-type: none"> Policy를 따랐을 때, State Value가 어떤 값?
	

Model : <ul style="list-style-type: none"> Agent가 Env에 대한 예측한 것이므로 실제 Env와 완벽하게 일치하지 않을 수 있다. Reward = -1, State Transition²⁾ Prob. = 1 로 Agent가 Model을 학습한 결과이다. 	
---	--

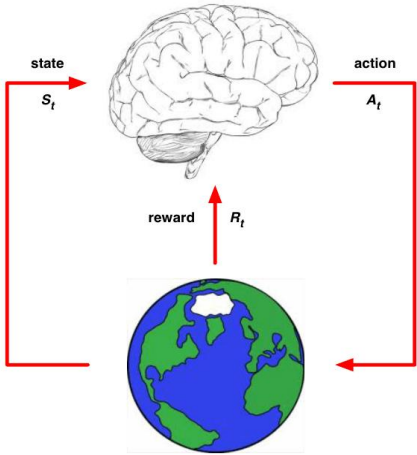
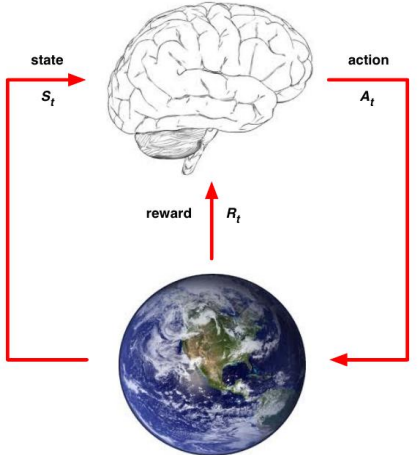
3-5. Categorizing Reinforcement Learning Agents

3-5-1. Policy와 Value function 유무 (Agent의 학습대상의 관점)

- a. Value Based : Value Function만 학습한다. 왜냐하면 Value Function만 업데이트 완료 후에 높은 값만 취해서 가면 되기 때문이다. Value Function에는 State Value Function, Action Value Function 두 가지가 있다.
- b. Policy Based : Policy만 학습한다.
- c. Actor-Critic : Policy와 Value Function 둘 다 학습한다.

3-5-2. Model 유무 (Agent의 학습방법의 관점)

- a. Model-Free : Agent가 Model을 생성하지 않고 Policy만 혹은 Value Function만 혹은 둘 다 받으면서 학습해가는 것이다.
- b. Model-Based : Agent가 내부적으로 Model을 생성한다.

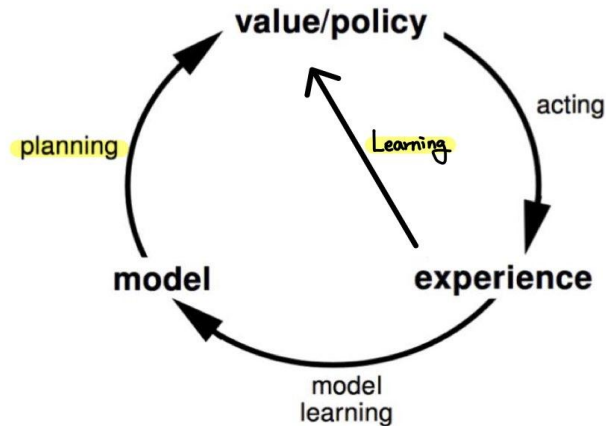
Model-Based	Model-Free
<ul style="list-style-type: none"> Agent가 Env.와의 상호작용을 통해 Experiences(=Transitions³⁾)를 쌓고, 이를 이용하여 Model을 배우고 이 Model을 이용하여 Prediction과 Control을 한다. (Planning) 주로 Dynamic Programming을 사용하여 푼다. 	<ul style="list-style-type: none"> Agent가 Env.와의 상호작용을 통해 Experiences(=Transitions)를 쌓고, 이를 이용하여 바로 Prediction과 Control을 한다.(Learning, Direct RL)
	

3) $transition = \langle s_t, a_t, r, s_{t+1} \rangle$

4. Problems within Reinforcement Learning

4-1. Learning and Planning

#. Model-Free 또는 Model-Based에서의 학습



- a. Learning : Model-Free인 상황에서 학습하는 것을 의미하며, 구체적으로는 환경과 상호작용하면서 Policy를 업데이트 해나가는 학습 방식이다. Direct Reinforcement Learning이라고 한다.
- b. Planning : Model-Based인 상황에서 학습하는 것을 의미하며, Known MDP(=<Reward, State Transition Prob.>) 즉 Model을 input으로 받고 Policy를 Produce 또는 Improve하는 계산과정을 Planning 이라 한다.
- c. Prediction and Control : Learning과 Planning은 모두 Value Function을 계산하는 Prediction 또는 이를 바탕으로 Optimal Policy를 계산하는 Control의 방법론을 적용하여 풀 수 있으며 강화학습의 핵심은 Prediction과 Control을 해결하는 것이다.

4-2. Exploration and Exploitation

#. Action 선택 방법

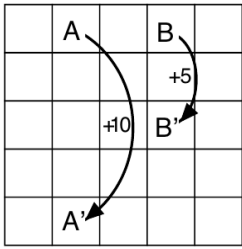
- a. Exploration : 시도하지 않은 action을 하여 정보를 모은다.
- b. Exploitation : 모은 정보를 바탕으로 최선의 action만 선택한다.
- c. 둘 사이엔 Trade-Off가 존재하며, 둘 사이에 어떻게 조절하고, 언제까지 할 것인지도 중요한 관심이다.

4-3. Prediction and Control

#. Agent가 Env.에서 무엇을 학습하려고 하는지에 대해

- Prediction : Prediction은 주어진 Policy를 Parameter로 하여, (Action) State Value를 그 Policy에 따라 Update(= Estimation, 계산, 학습)하는 것을 의미하는데 크게 Monte-Carlo와 Temporal Difference 2가지 방법이 있다.
- Control : a에서 Estimated (Action) State Value를 이용하여 최적의 Policy를 찾는 방법을 의미한다.
- Example

Prediction



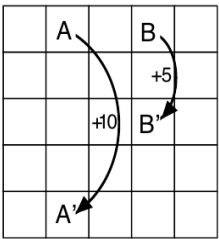
(a)

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

(b)

- Policy는 Uniform Random Policy로 주어짐.
- (b)는 주어진 Uniform Random Policy를 이용하여 State Value function을 구한 값.

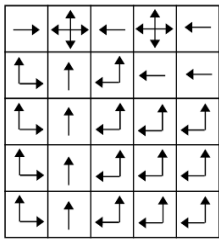
Control



a) gridworld

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

b) v_*



c) π_*

- State에서 어떻게 움직여야 하는지를 구한 것.
- (a)에서의 화살표는 순간이동을 의미함.
- (c)는 주어진 State Value Function (b)을 이용하여 Greedy 방법으로 구한 Optimal Policy이다.

5. Summary of Reinforcement Learning Flow

#. 강화학습의 핵심은 Prediction, Control을 구하는 것이 핵심이다. 7단원 이전까지는 각 단원에서 제시한 상황에서 Prediction, Control 문제를 푸는 방법을 배워왔다. 강화학습의 큰 흐름은 Model-Based, Model-Free로 크게 분류할 수 있는데 전자는 Model을 Agent가 경험을 통해 학습하고 Planning을 하여 Prediction과 Control을 푸는 것⁴⁾이고, 후자는 Model을 배우지 않고 직접 경험을 통해 Learning을 하여 Prediction과 Control을 푸는 것이다.

각 단원의 흐름을 정리하면 아래의 표와 같다.

1단원			• 「Introduction to Reinforcement Learning」
Model-Based	Small Scaled	2단원	• 「Markov Decision Process」 • MDP가 무엇인가?
		3단원	• 「Planning by Dynamic Programming」 • Dynamic Programming을 통해 MDP를 푸는 법
Model-Free	Small Scaled (Tabular Method)	4단원	• 「Model-Free Prediction」 • MDP푸는 법 = Prediction (Policy Evaluation)
		5단원	• 「Model-Free Control」 • MDP푸는 법 = Control (Policy Improvement)
	Large Scaled (Function Approximation)	6단원	• 「Value Function Approximation」 • MDP푸는 법 = Prediction (Policy Evaluation)
		7단원	• 「Policy Gradient」 • MDP푸는 법 = Control (Policy Improvement)

a. Estimated (Action) State Value를 표현하는데 있어서 Tabular Method와 Function Approximation 2가지 방법이 있다.