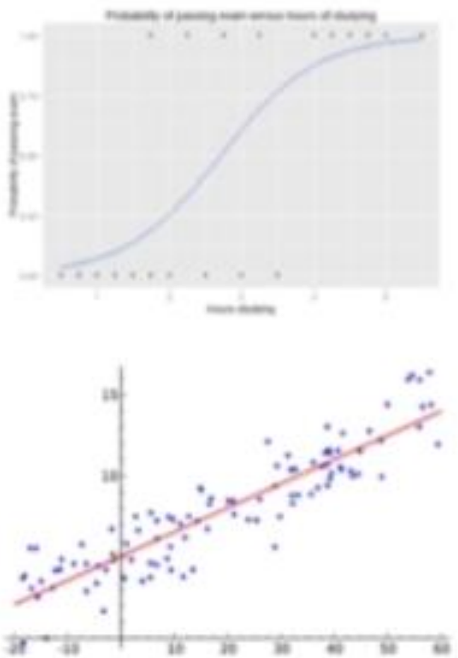
1강. Introduction

Contents

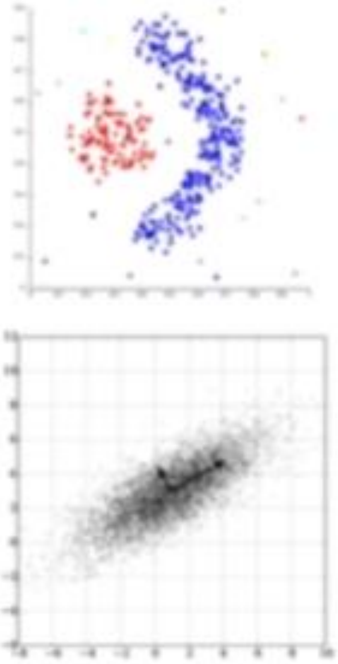
- RL Overview
    - What is Reinforcement Learning ?
    - How RL is different from other machine learning ?
      (w.r.t why Reinforcement Learning ?)
    - What types of RL algorithms ?
- Course Intro.
- MDP (w/ Code example)

Three paradigms of Machine Learning

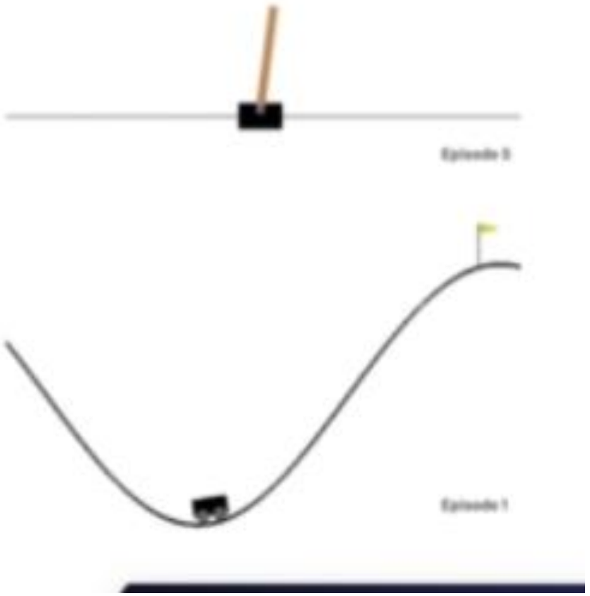
Supervised Learning



Unsupervised Learning



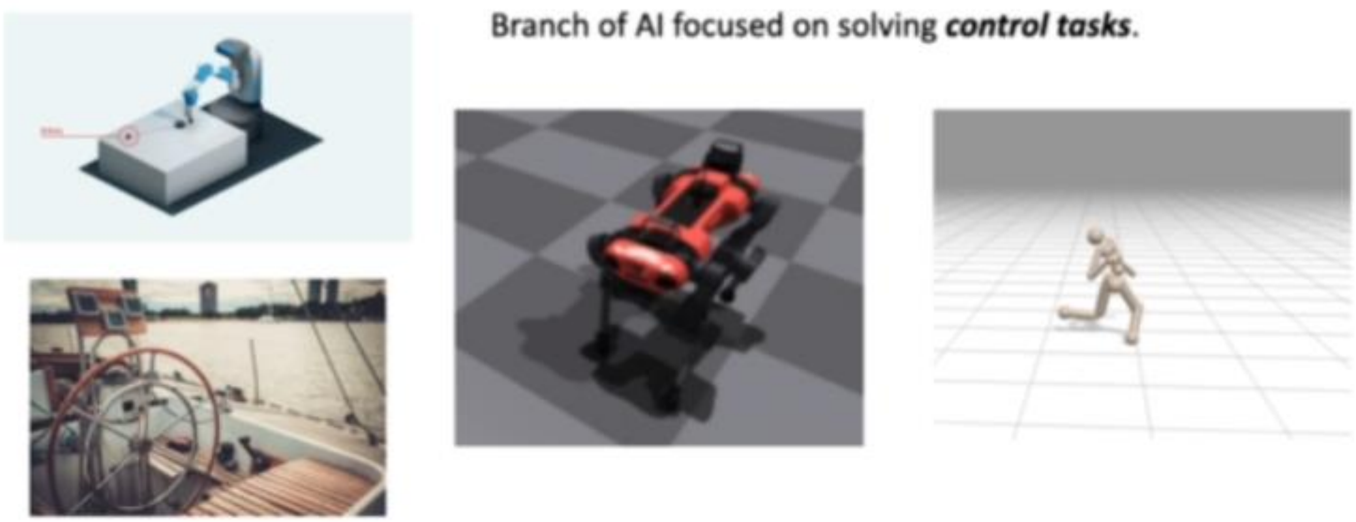
Reinforcement Learning



What is RL ?

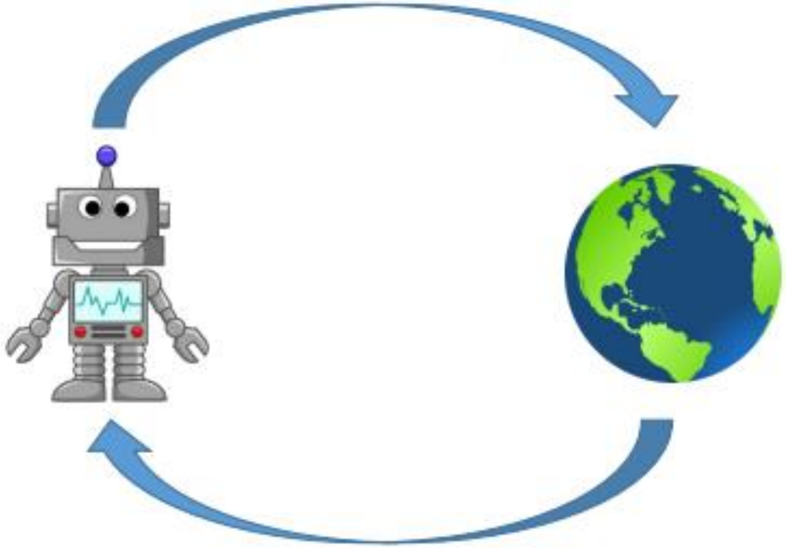
- Mathematical formalism for learning-based decision-making

- Approach for learning decision-making and control from experience



What is RL ?

decisions (actions)



<u>consequences</u>
observations (states)
rewards

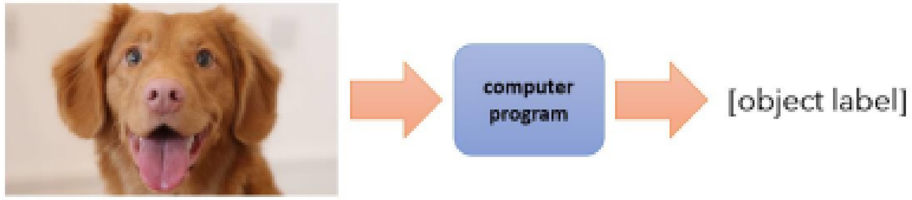
Actions: muscle contractions
Observations: sight, smell
Rewards: food

Actions: motor current or torque
Observations: camera images
Rewards: task success measure (e.g., running speed)

Actions: what to purchase
Observations: inventory levels
Rewards: profit

Supervised vs. Reinforcement Learning

supervised learning



input: $\mathbf{x}$

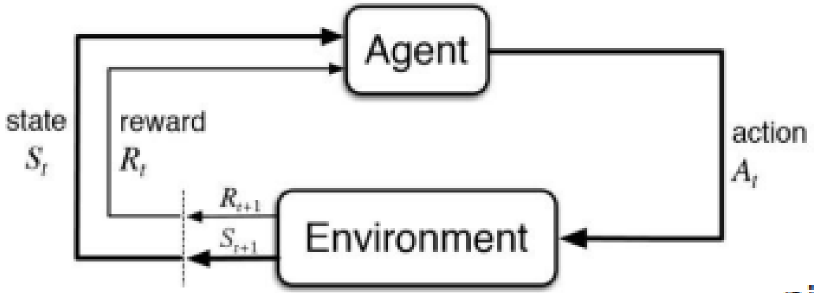
output: $\mathbf{y}$

data: $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}$

goal: $f_\theta(\mathbf{x}_i) \approx \mathbf{y}_i$

someone gives this to you

reinforcement learning



pick your own actions

input: $\mathbf{s}_t$ *at each time step*

output: $\mathbf{a}_t$ *at each time step*

data: $(\mathbf{s}_1, \mathbf{a}_1, r_1, \ldots, \mathbf{s}_T, \mathbf{a}_T, r_T)$

goal: learn $\pi_\theta : \mathbf{s}_t \rightarrow \mathbf{a}_t$

to maximize $\sum_t r_t$

Supervised vs. Reinforcement Learning

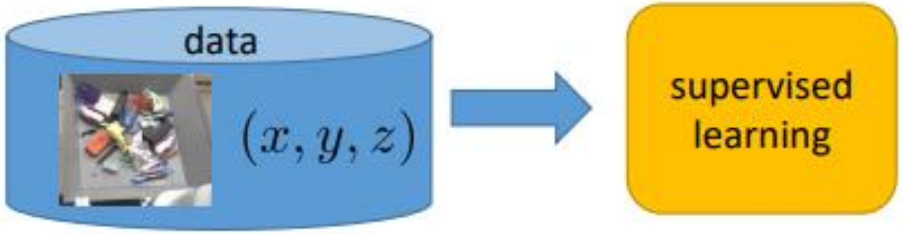
Standard (supervised) machine learning:

given $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$

learn to predict y from $\mathbf{x}$ $\qquad f(\mathbf{x}) \approx y$

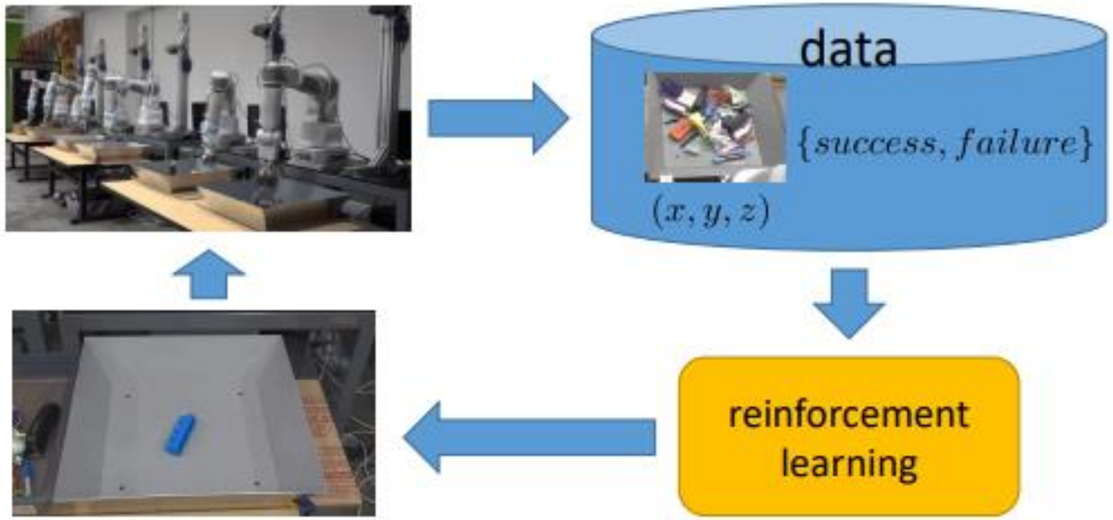
Usually assumes:

- i.i.d. data
- known ground truth outputs in training

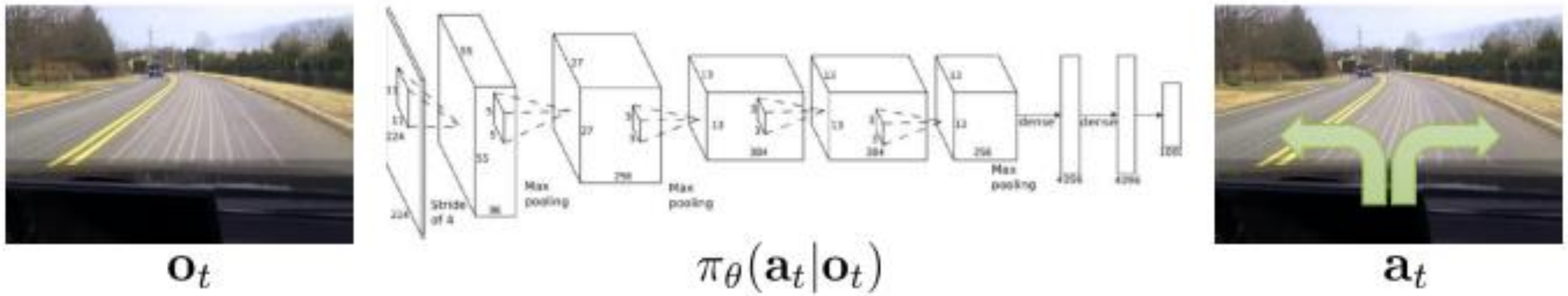


Reinforcement learning:

- Data is **not** i.i.d.: previous outputs influence future inputs!
- Ground truth answer is not known, only know if we succeeded or failed
  - more generally, we know the reward

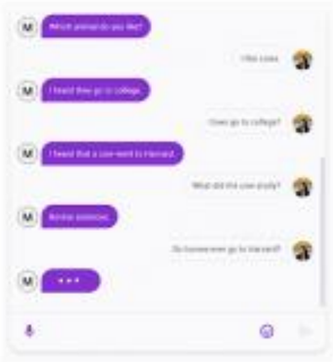


Supervised vs. Reinforcement Learning

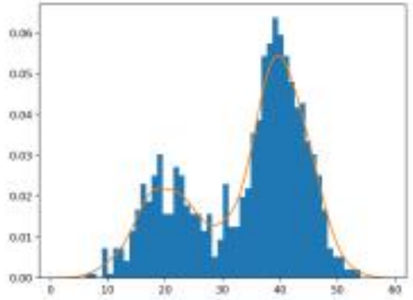


behavioral cloning

Data-Driven AI vs. RL



$p_\theta(\mathbf{x})$ $\quad$ $p_\theta(\mathbf{y}|\mathbf{x})$



Data-Driven AI vs. RL

Impressive because no person had thought of it!

“Move 37” in Lee Sedol AlphaGo match: reinforcement learning “discovers” a move that surprises everyone

Impressive because it looks like something a person might draw!

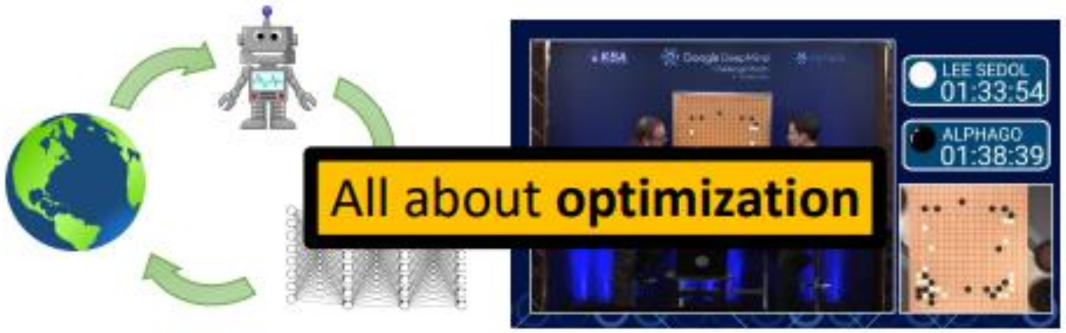
Data-Driven AI vs. RL

Data-Driven AI

\+ learns about the real world from data

\- doesn't try to do **better** than the data

Reinforcement Learning



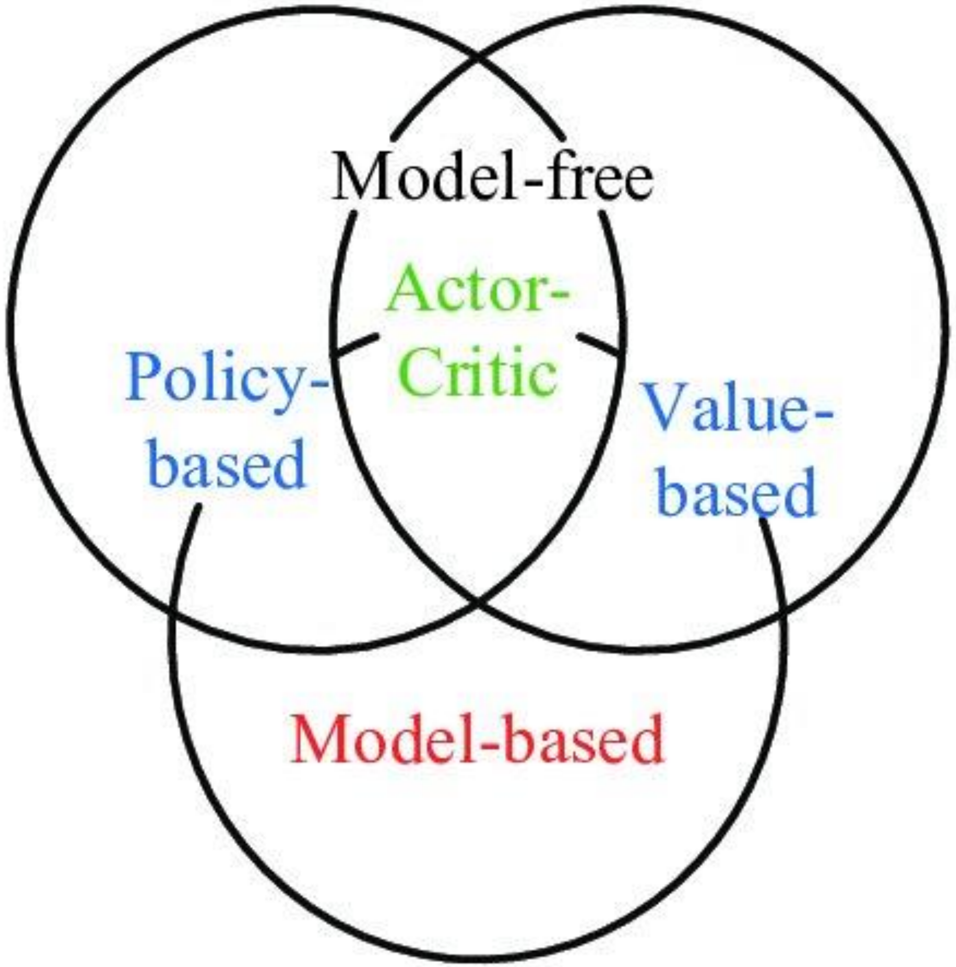
\+ optimizes a goal with emergent behavior

\- but need to figure out how to use at scale!

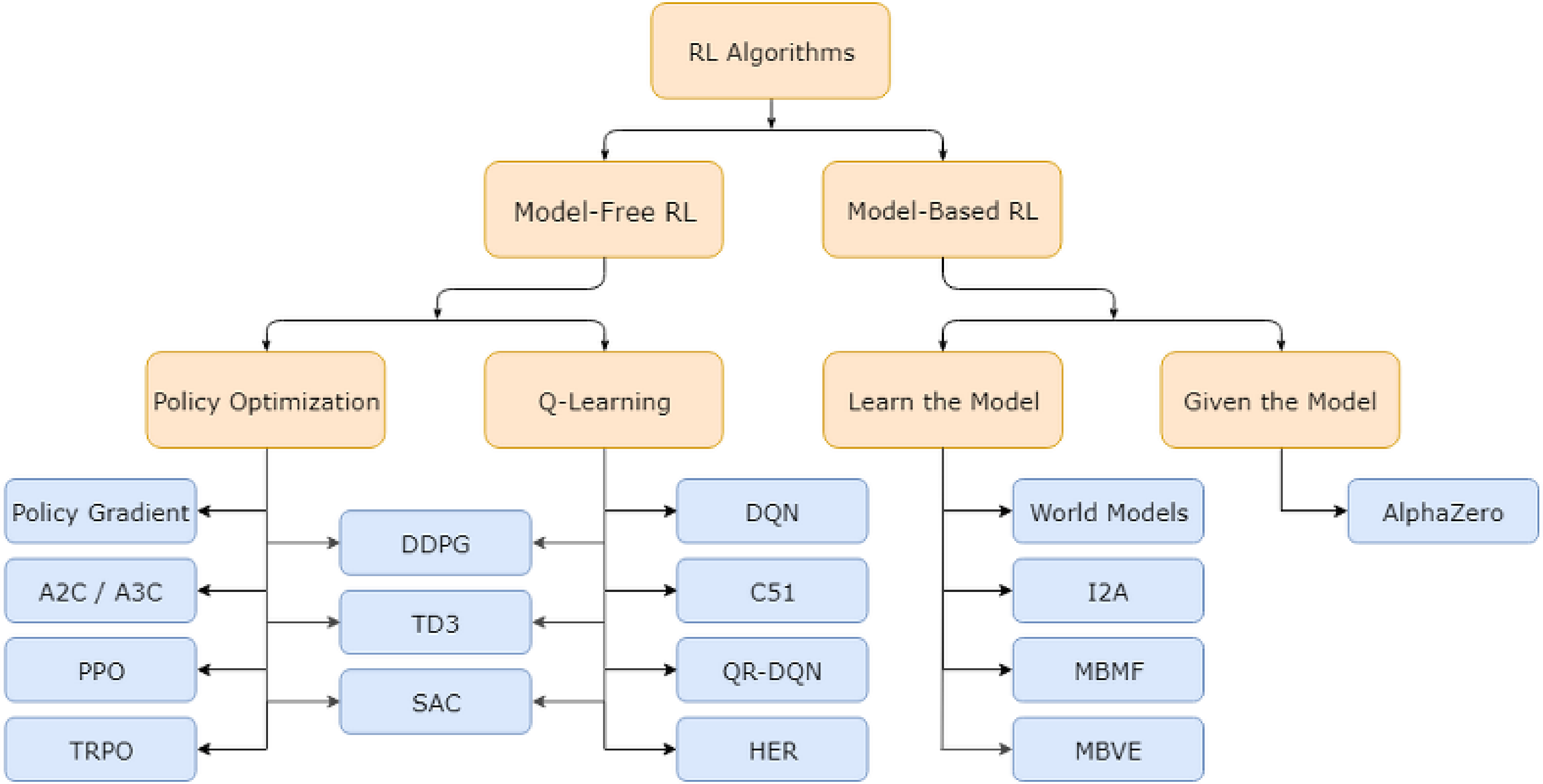
Data without **optimization** doesn't allow us to solve new problems in new ways

RL can discover new solutions

RL Algorithms Classification



RL Algorithms Classification

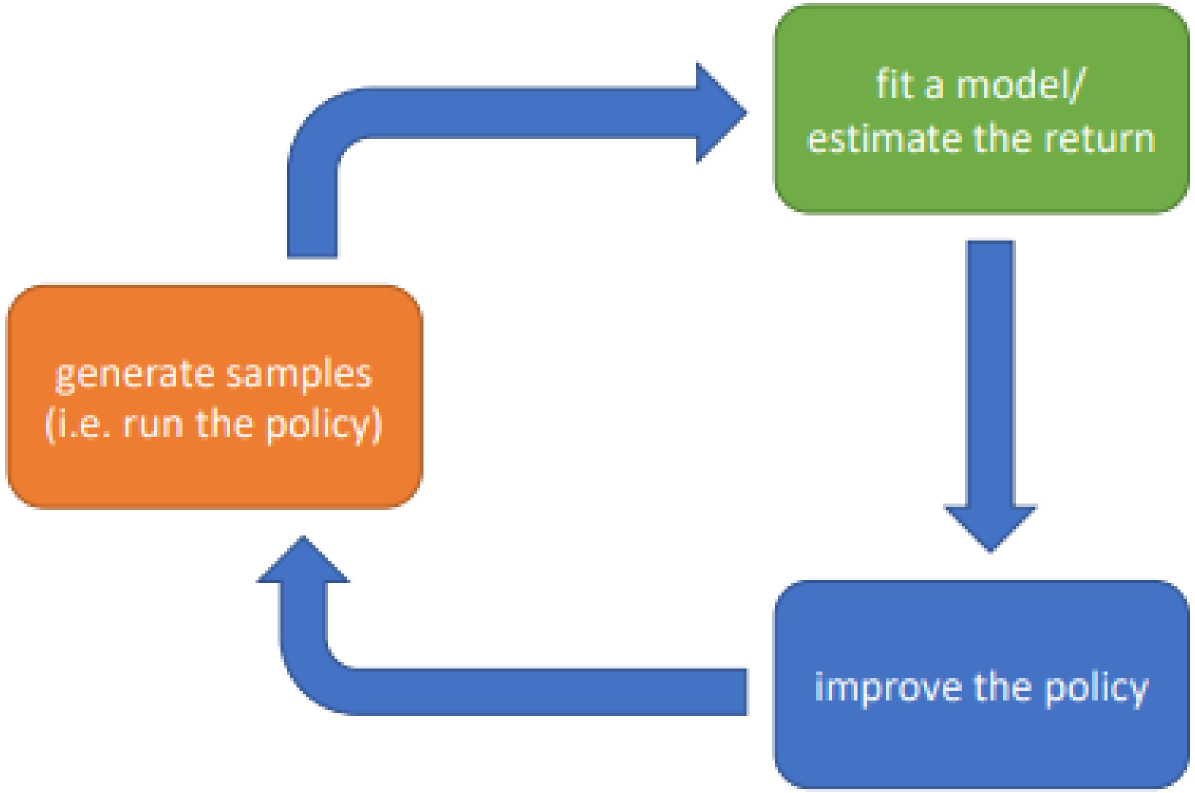


Types of algorithms

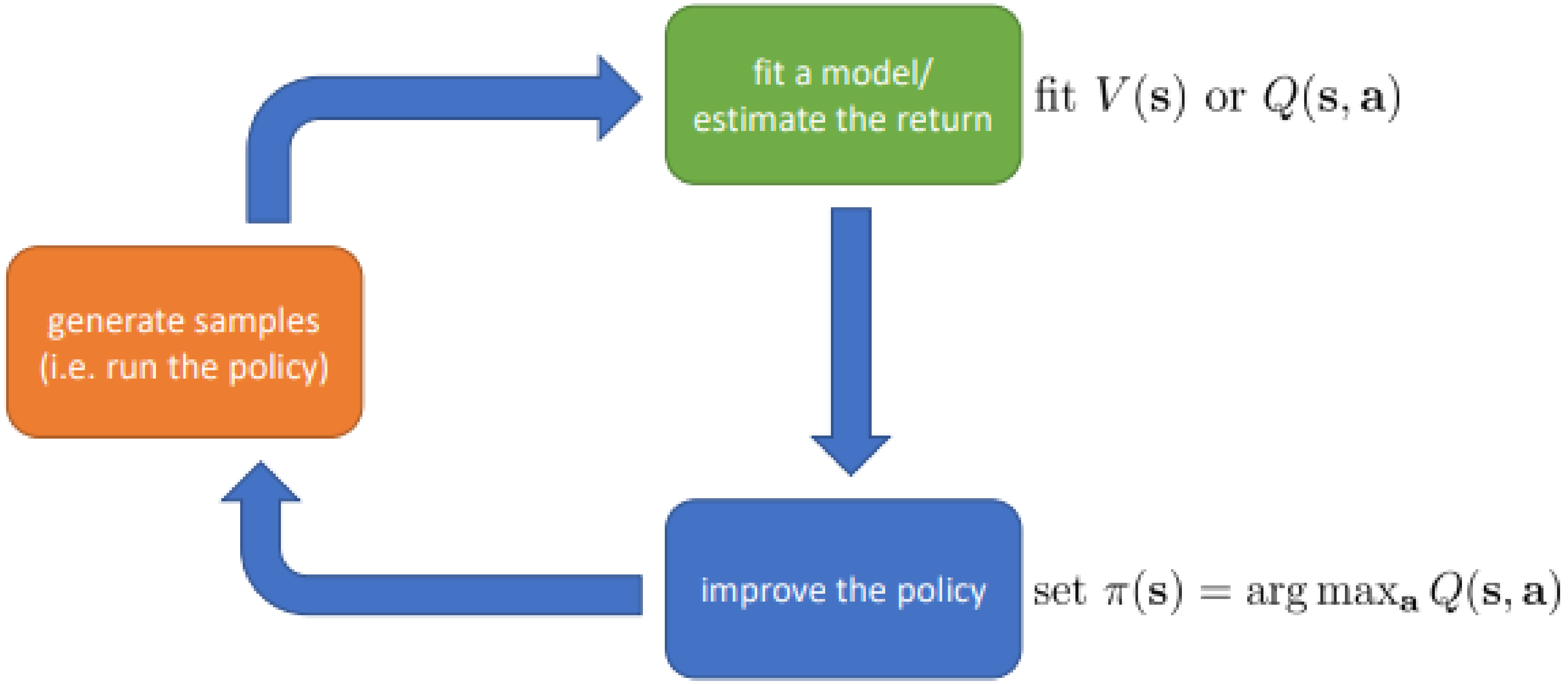
$$\theta^\star = \arg\max_\theta E_{\tau \sim p_\theta(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

- Policy gradients: directly differentiate the above objective
- Value-based: estimate value function or Q-function of the optimal policy (no explicit policy)
- Actor-critic: estimate value function or Q-function of the current policy, use it to improve policy
- Model-based RL: estimate the transition model, and then…
  - Use it for planning (no explicit policy)
  - Use it to improve a policy
  - Something else

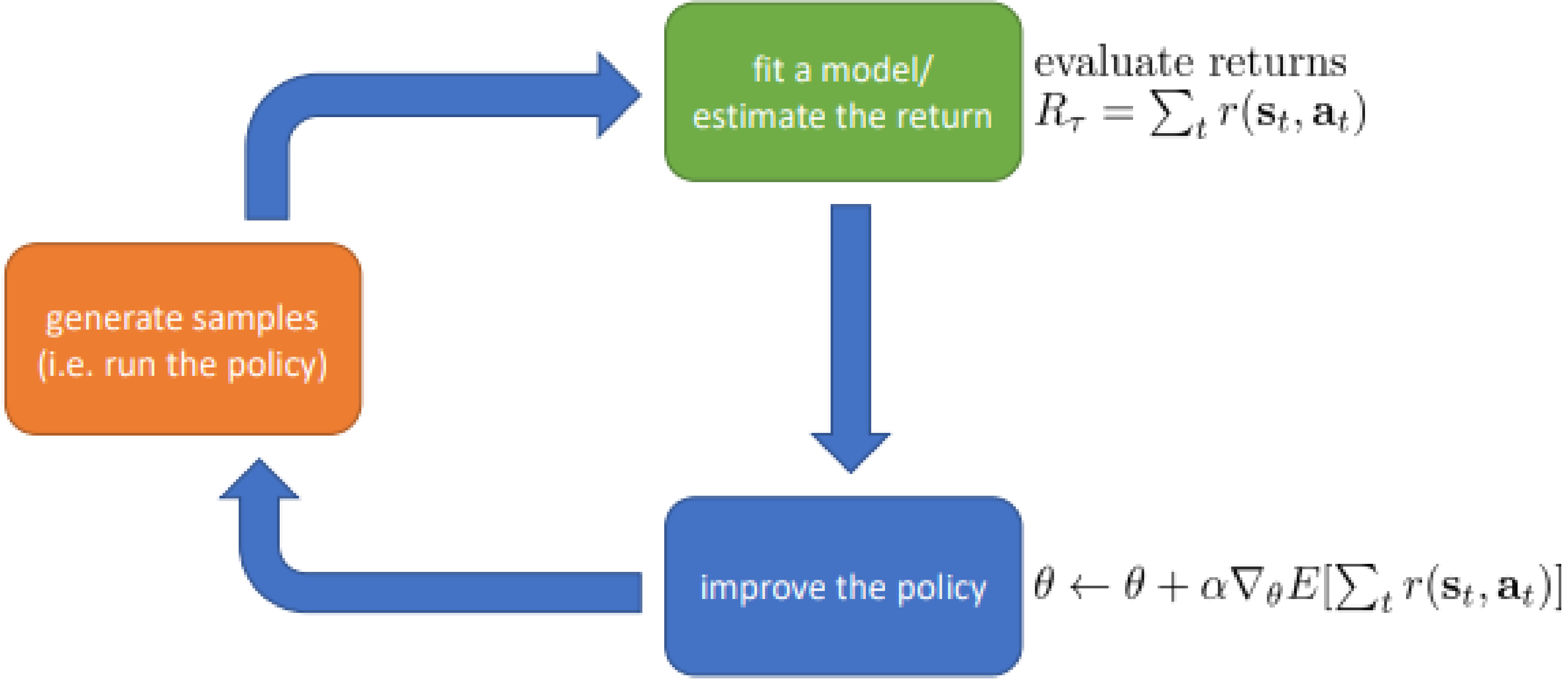
The anatomy of RL algorithm



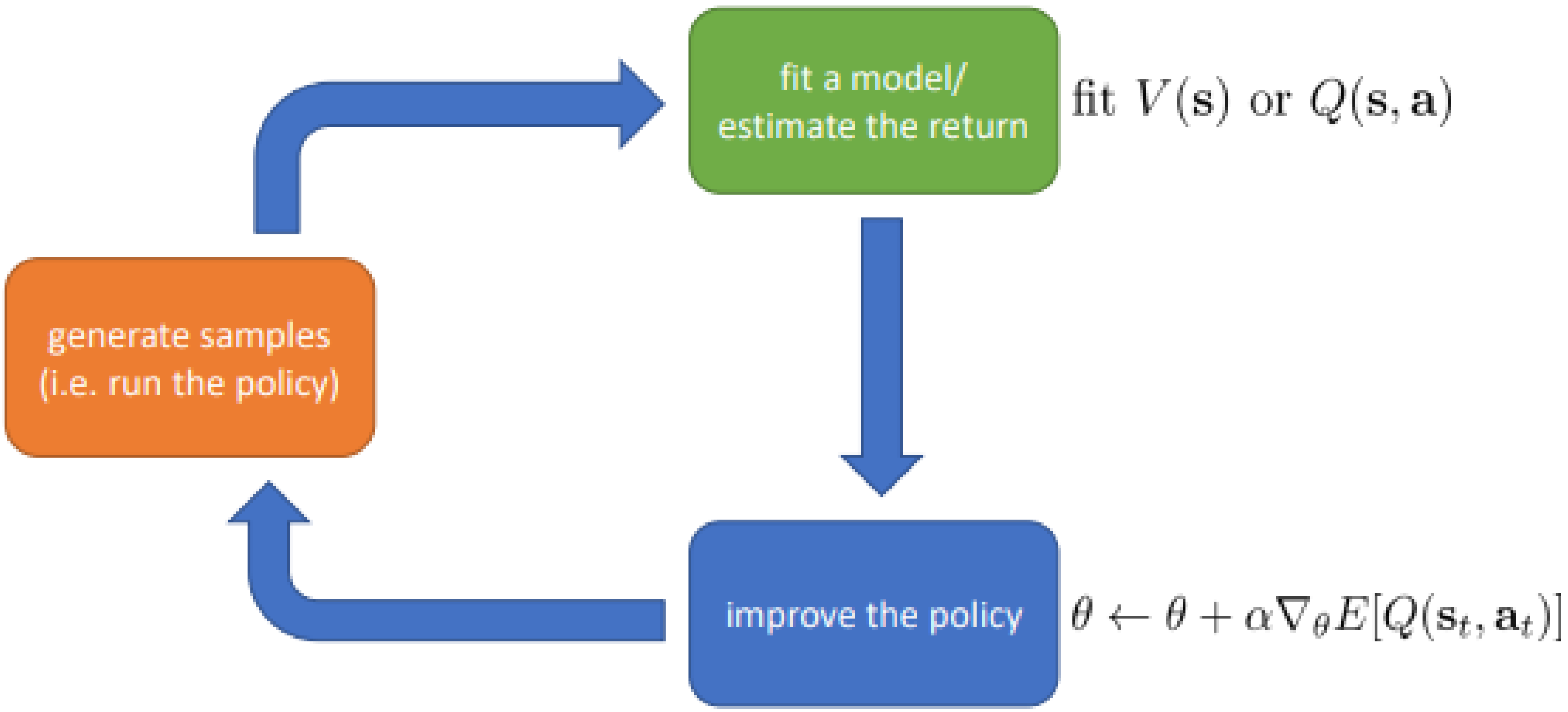
Value-based algorithms



Direct policy gradients

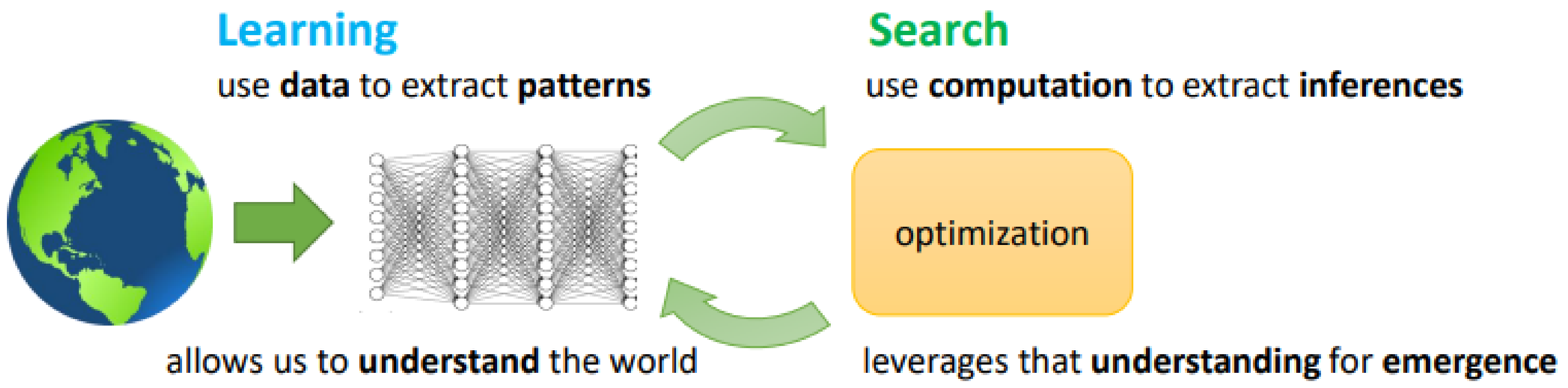


Actor-critic: value functions + policy gradients



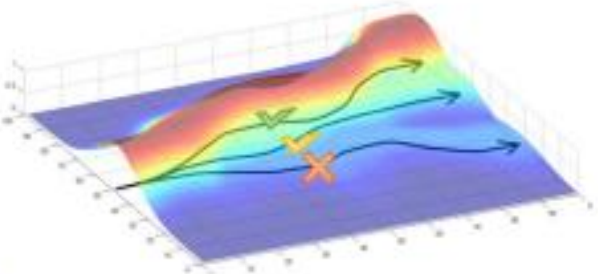
Deep Reinforcement Learning

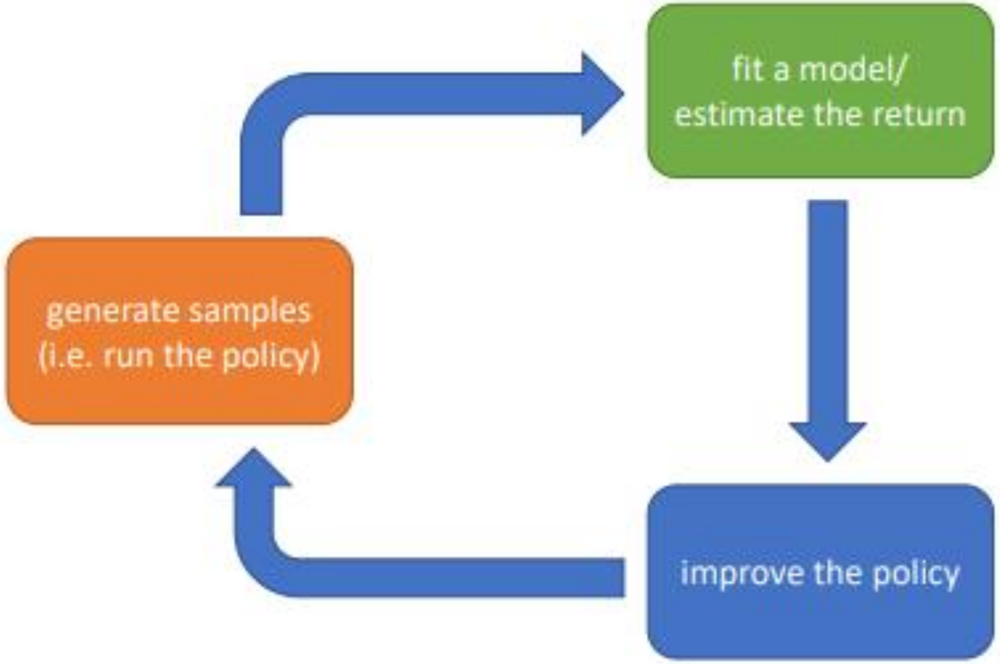
- Deep = scalable learning from large, complex datasets
- Reinforcement learning = optimization



Deep Reinforcement Learning

A simple example





$$J(\theta) = E_\pi\left[\sum_t r_t\right] \approx \frac{1}{N}\sum_{i=1}^{N}\sum_t r_t^i$$

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

Prerequisites

Calculus

$$\nabla L(\theta) = \left[\frac{\partial L(\theta)}{\partial w_1}, \dots, \frac{\partial L(\theta)}{\partial w_n}\right]$$

Linear algebra

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{pmatrix}$$

Python

python

Machine Learning

PyTorch

Schedule

주차	수업내용	교재범위 및 과제물	비고
1	강화학습 Overview		
2	강화학습 기본: Dynamic Programming	실습	
3	강화학습 기본: Monte-Carlo Methods	실습	
4	강화학습 기본: Temporal Difference Learning	실습	
5	가치기반 강화학습: Q-learning, Deep Q-learning	실습	
6	심층강화학습 실습환경 소개: NN, Pytorch	실습	
7	가치기반 강화학습: DQN	실습	
8	중간시험		
9	가치기반 강화학습: Deep Q-learning (Continuous)	실습	
10	정책기반 강화학습: Policy Gradient	실습	
11	정책기반 강화학습: Actor Critic		
12	정책기반 강화학습: DDPG	실습, 팀과제	
13	정책기반 강화학습: SAC	실습, 팀과제	
14	정책기반 강화학습: HER	실습, 팀과제	
15	기말발표		

Markov Decision Process (MDP)

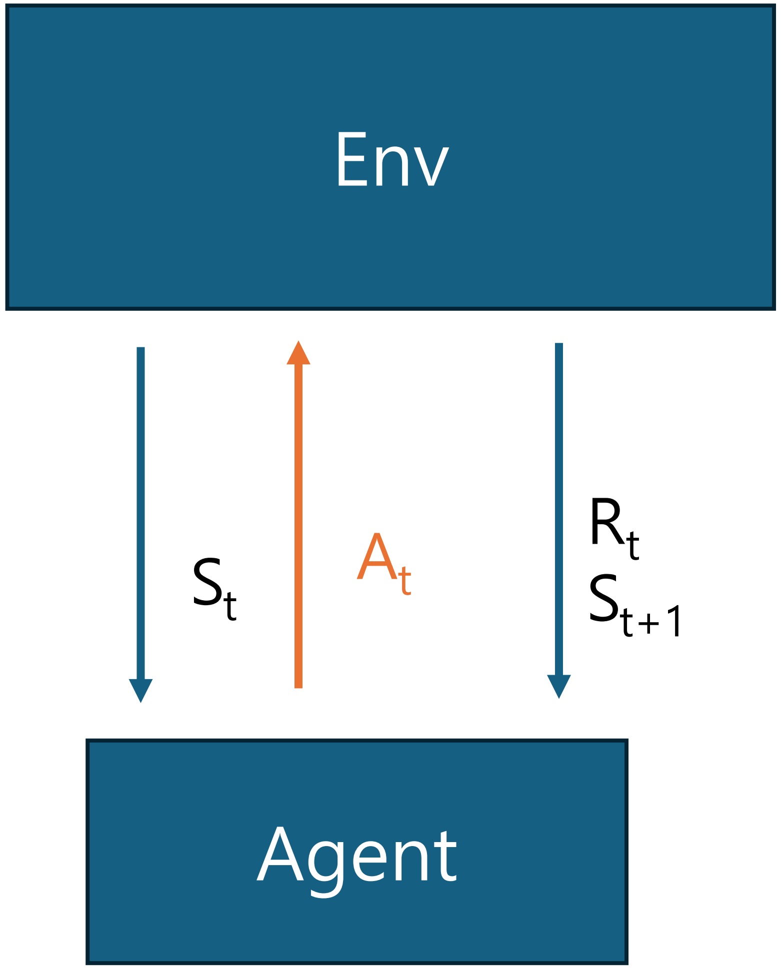
Markov Decision Process(MDP)

- Discrete-time stochastic control process
  - Time moves forward in finite intervals: t {1,2,3,4}
  - Future states depend only partially on the actions taken
  - It is based on decision making to reach the target state

- Formalism to define a control task problem




Agent interaction with Env.



(S,A,R,P)

- Set of possible **states** of the task

$\mathbf{o}_t$ – observation

$\mathbf{s}_t$ – state

- Set of **actions** that can be taken in each of the states
- Set of **rewards** for each (s, a) pair
- Probabilities of passing from one state to another when taking each possible action (Transition **Probabilities**)

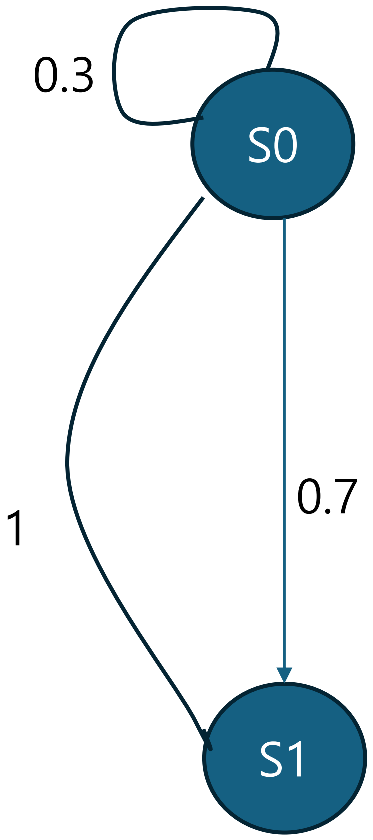
Markov Property

- The process has no memory:

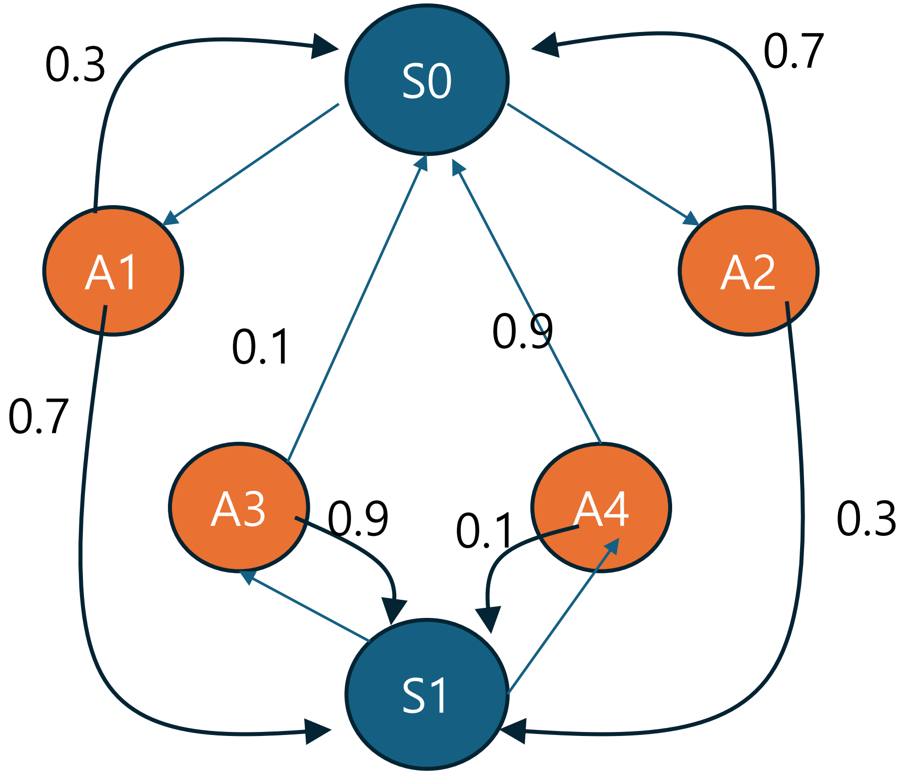
$$P[S_{t+1} \mid S_t = s_t] = P[S_{t+1} \mid S_t = s_t, S_{t-1}=s_{t-1}, \ldots, S_0 = s_0]$$

- The next state depends only on the current state, not on the previous ones
- If a process meets this property, it is known as Markov process (would it be valid in the real world ?)

Markov chain vs. MDP



Differences are actions and rewards



Example: Markov Process(MP)

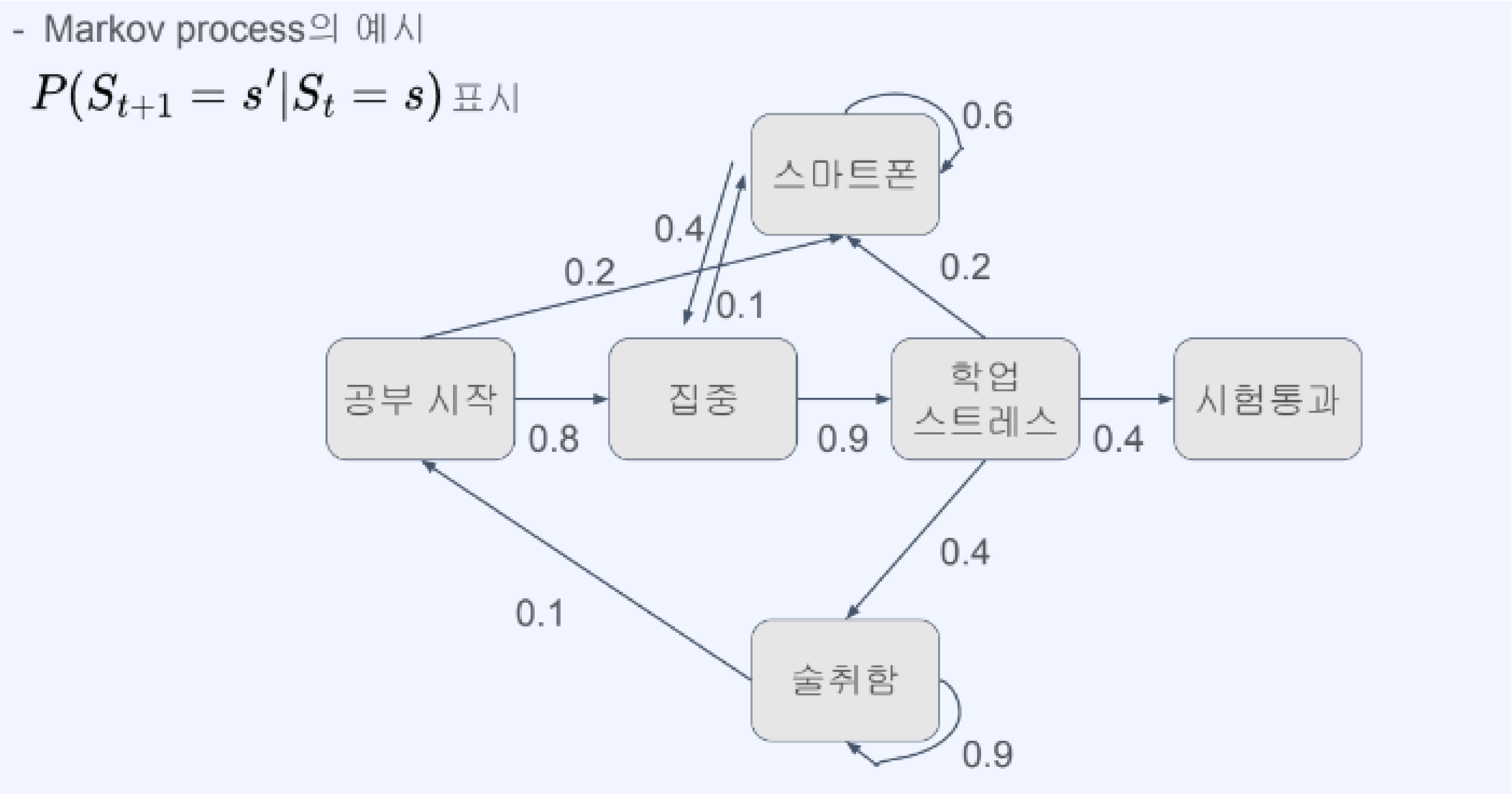


그림 출처: Fastcampus 권태환

Example: Markov Reward Process(MRP)

- Markov reward process의 예시

$P(S_{t+1} = s' | S_t = s)$ 표시

$R(s, s')$ 표시

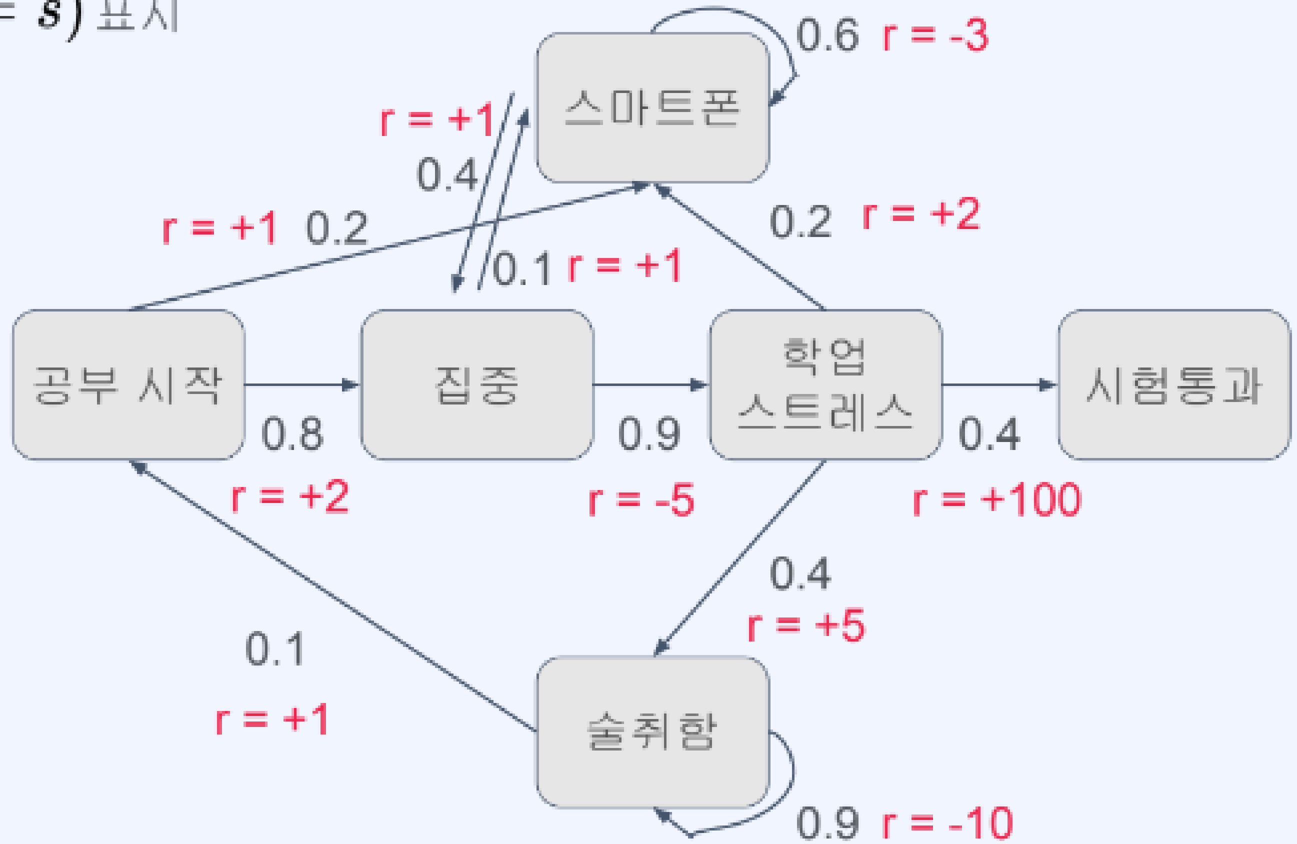


그림 출처: Fastcampus 권태환

Example: Markov Decision Process(MDP)

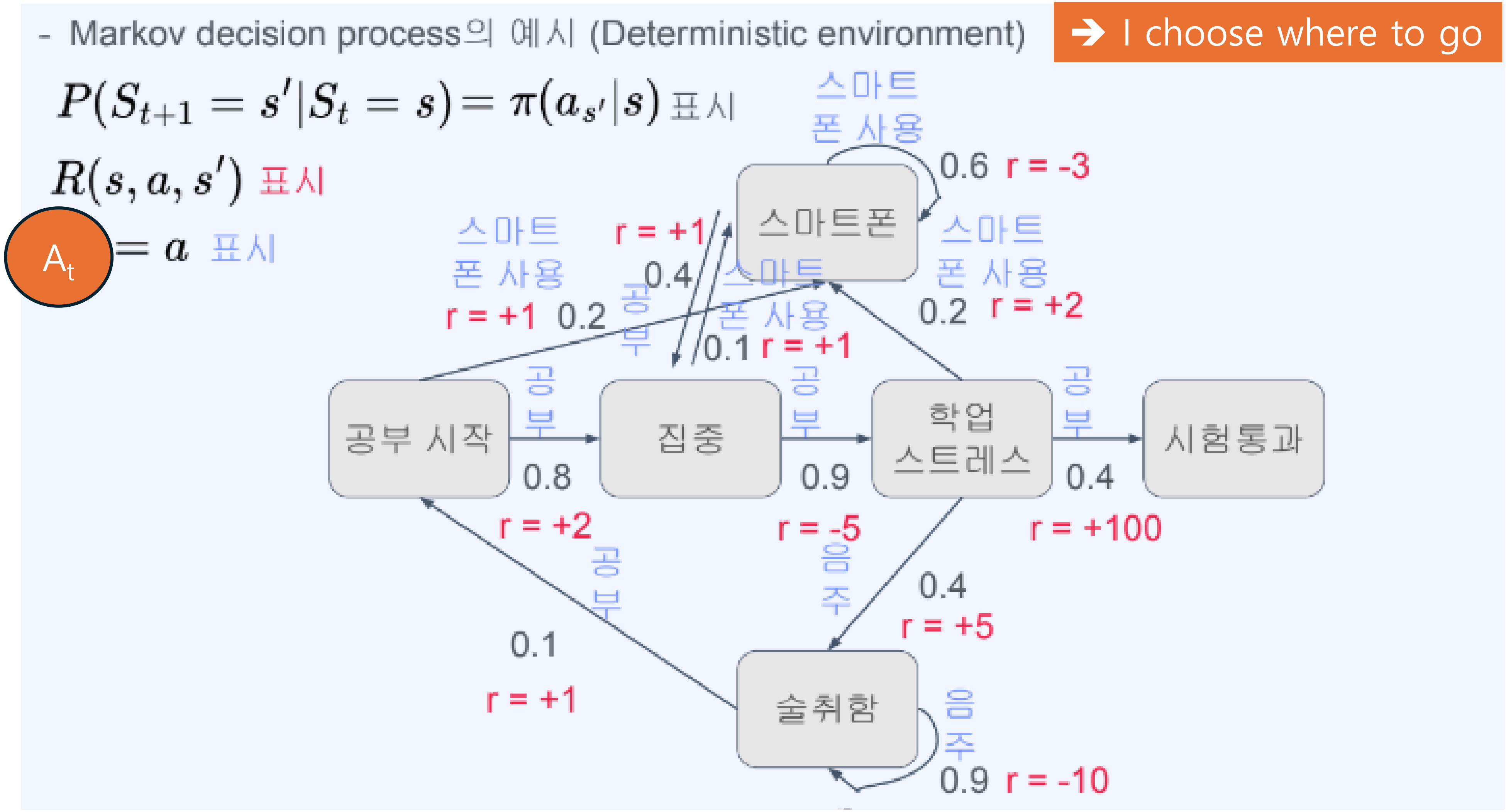
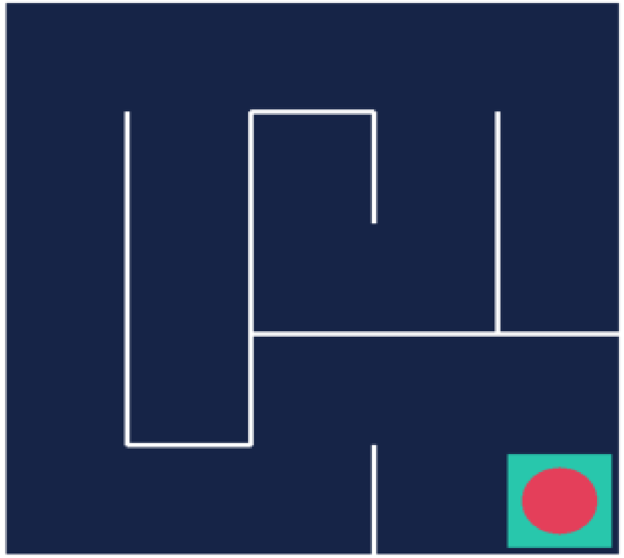


그림 출처: Fastcampus 권태환

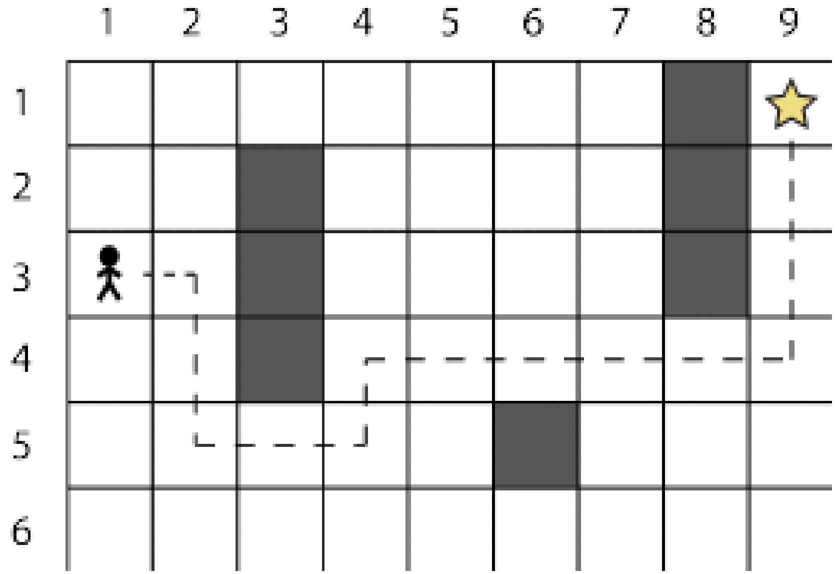
Types of MDP

- Finite vs. Infinite
- Episodic vs Continuing



Trajectory and episode

- Trajectory:
  - Elements that are generated when the agent moves from one state to another
  - {tau} = $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, S_3$



Reward vs Return

- The goals of the task are represented by the rewards(Rt)
  - We want to maximize the sum of rewards
- A short-term reward can worsen long-term results
  - We want to maximize the long-term sum of rewards

- Reward: R_t
- Return: $G_t = R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T$
  - We want to maximize the episode's return

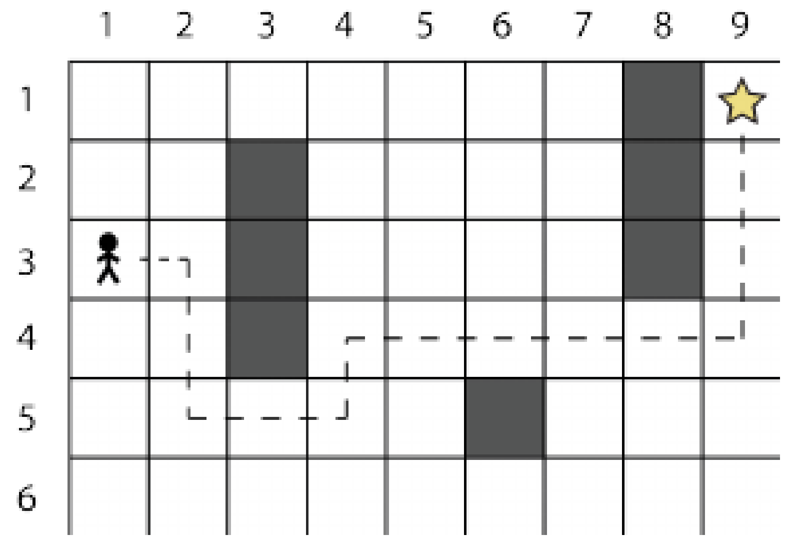
Discount factor

- We will multiply future rewards by a discount factor
  - $G_0 = R_1 + \gamma^* R_2 + \gamma^2 {}^* R_2 + \gamma^3 R_3 + \ldots + \gamma^{T-t-1} R_T$, $\gamma \in [0,1]$
    - $\begin{cases} \text{If } \gamma = 0\text{, all future rewards will be 0} \\ \text{If } \gamma = 1\text{, all future rewards will not be discounted} \end{cases}$
  - γ measures how far into the future the agent has to look when planning its actions
- We want to maximize the long-term sum of discounted rewards

Discount factor

- The agent has no incentive to go to the goal through the shortest route

$$G_0 = R_1 + R_2 + R_3 + \ldots$$



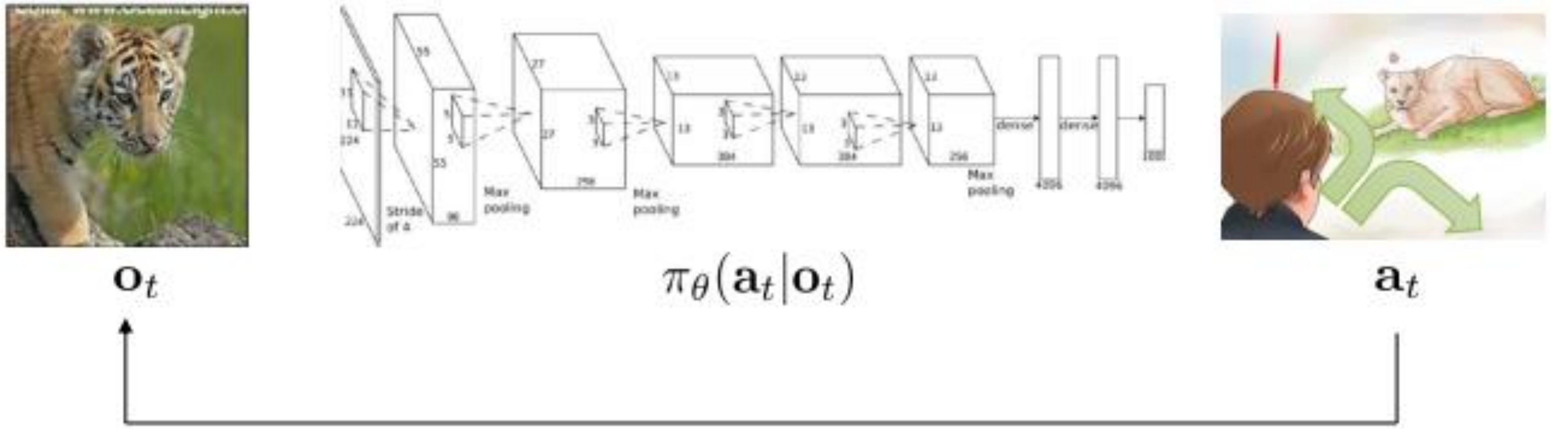
Policy π(s)

- Function that decides what action to take in a particular state

$$\pi\text{: S} \rightarrow \text{A}$$

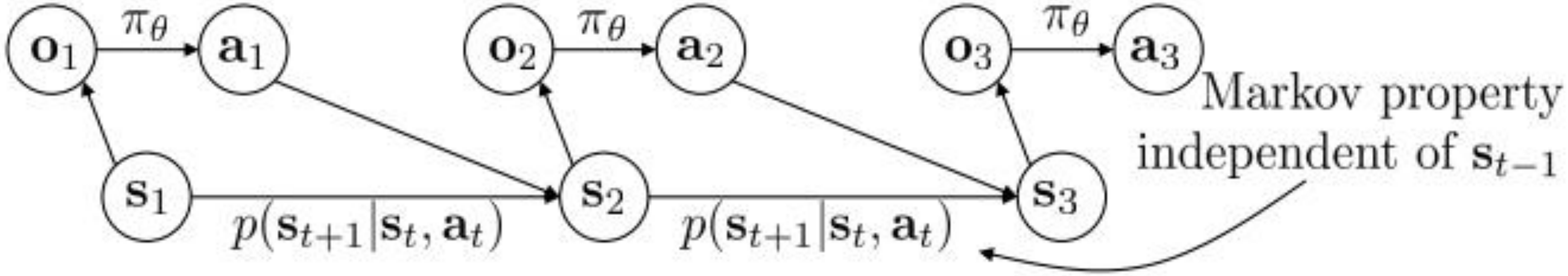
- Probability of taking an action a in state s: $\pi(a|s)$
  - Stochastic: $\pi(a|s) = [p(a_1), p(a_2), \ldots, p(a_n)]$
    e.g.. $\pi(a|s) = [0.3, 0.2, 0.5]$
- Action a taken in state s: $\pi(s)$
  - Deterministic: $\pi(s) \rightarrow a$
    e.g., $\pi(s) = a1$
- We must find the optimal policy π^*

Terminology(Overall)



$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)



MDP Code Example

Setup Code Env.

- Git download: https://git-scm.com/downloads
- Git clone https://github.com/parkjin-nim/rl_lecture.git
- Install Anaconda
- conda create -n <your env. name> -f environment.yml
- Open Jupyter notebook
- Open MDP_introduction.ipynb