Deep Reinforcement Learning

Apr 16th, 2019

Applied Deep Learning

YUN-NUNG (VIVIAN) CHEN

HTTP://ADL.MIULAB.TW

國立臺灣大學
National Taiwan University

Slides credited from Dr. David Silver & Hung-Yi Lee

Outline

Machine Learning

- Supervised Learning v.s. Reinforcement Learning
- Reinforcement Learning v.s. Deep Learning

Introduction to Reinforcement Learning

- Agent and Environment
- Action, State, and Reward

Markov Decision Process

Reinforcement Learning Approach

- Value-Based
- Policy-Based
- Model-Based

Outline

Machine Learning

- Supervised Learning v.s. Reinforcement Learning
- Reinforcement Learning v.s. Deep Learning

Introduction to Reinforcement Learning

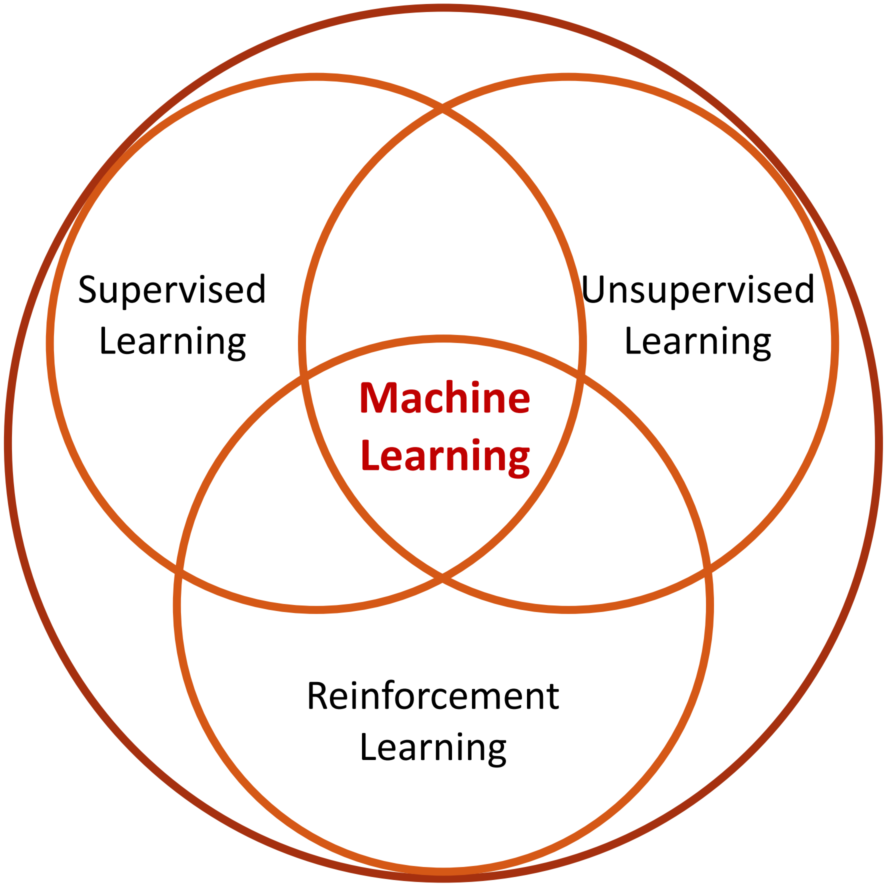
- Agent and Environment
- Action, State, and Reward

Markov Decision Process

Reinforcement Learning Approach

- Value-Based
- Policy-Based
- Model-Based

Machine Learning



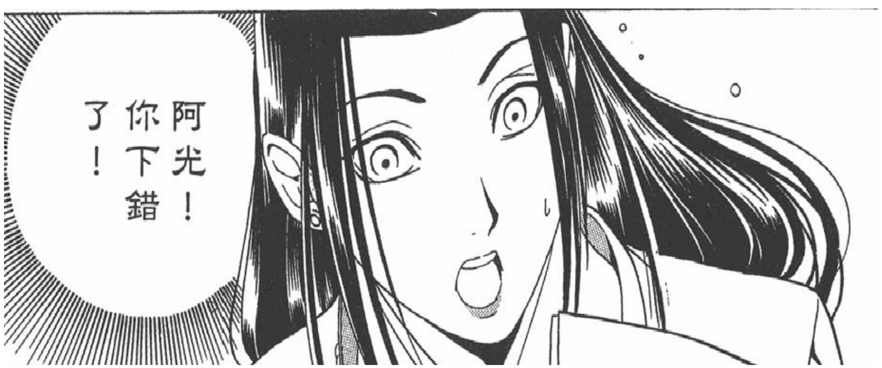
Supervised v.s. Reinforcement

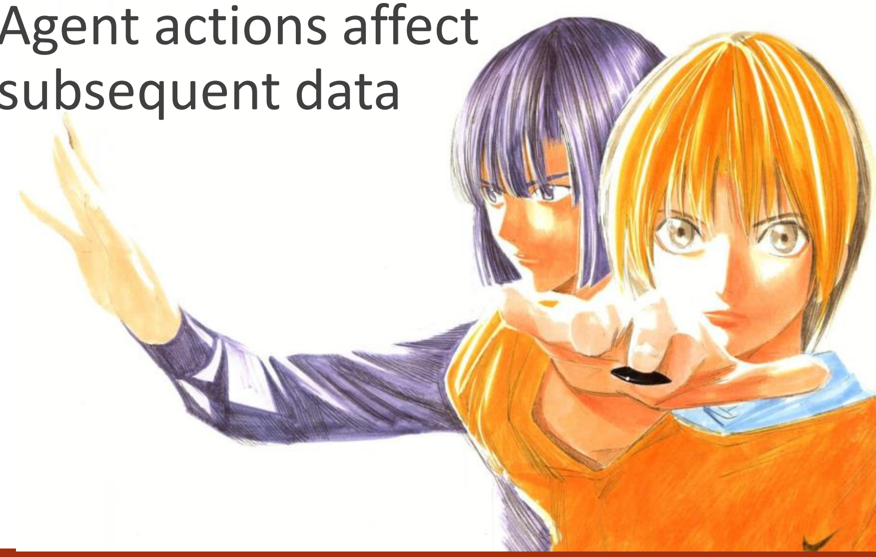
Supervised Learning

- Training based on supervisor/label/annotation
- Feedback is instantaneous
- Time does not matter

Reinforcement Learning

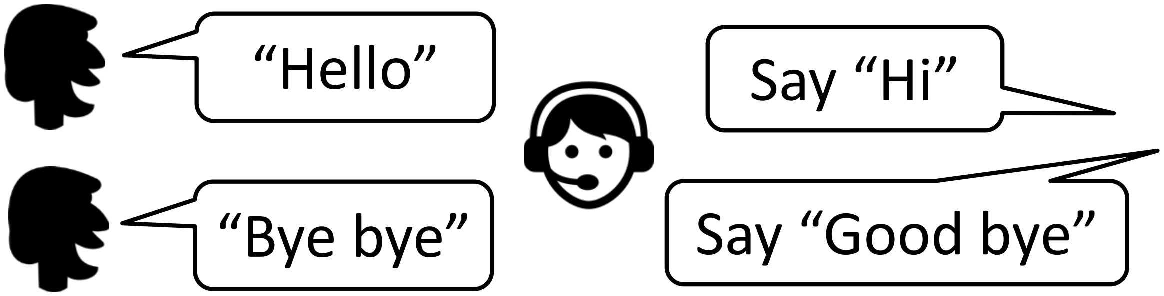
- Training only based on reward signal
- Feedback is delayed
- Time matters
- Agent actions affect subsequent data





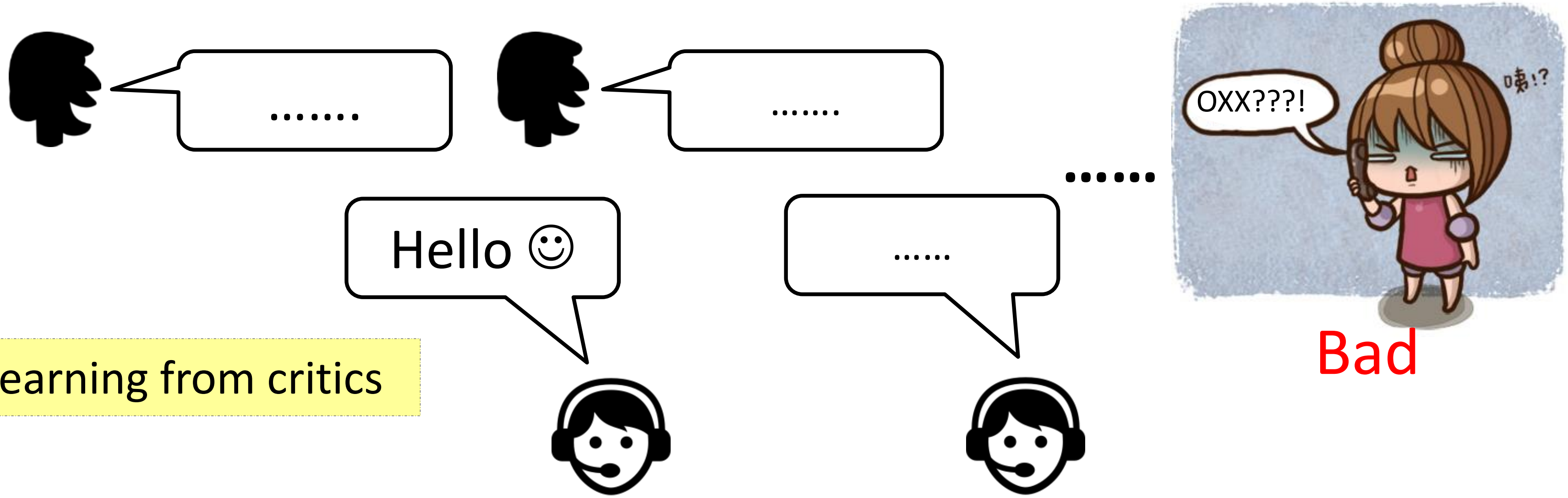
Supervised v.s. Reinforcement

Supervised



Learning from teacher

Reinforcement



Learning from critics

Reinforcement Learning

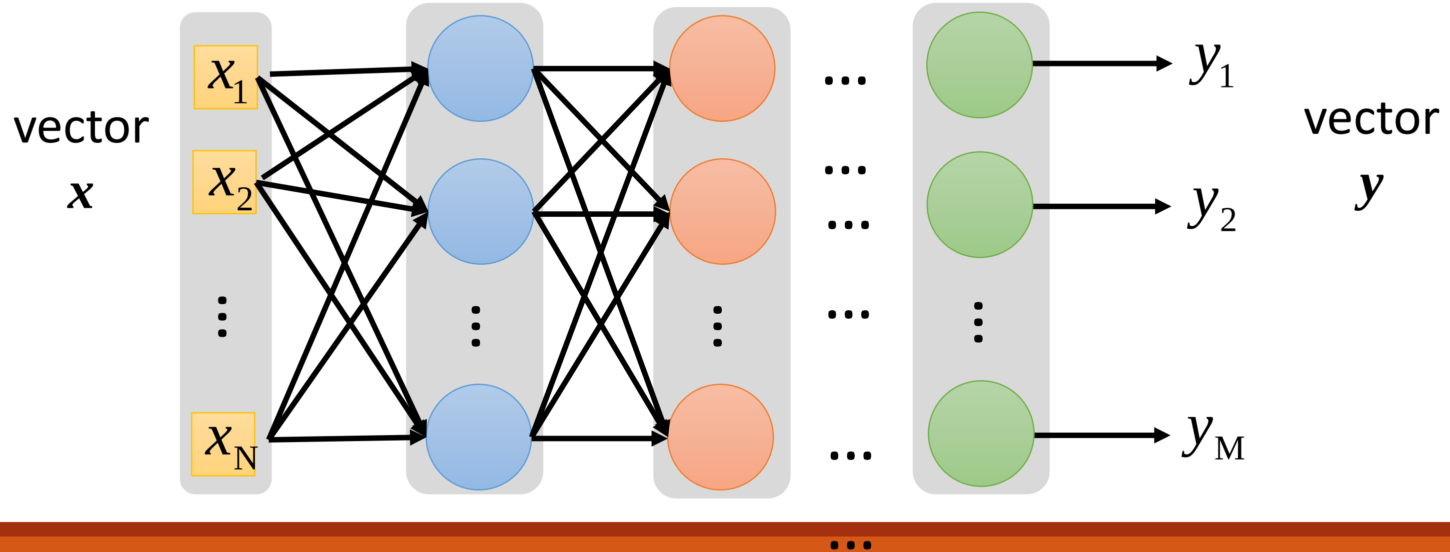
RL is a general purpose framework for **decision making**

- RL is for an *agent* with the capacity to *act*
- Each *action* influences the agent's future *state*
- Success is measured by a scalar *reward* signal
- Goal: *select actions to maximize future reward*

Deep Learning

DL is a general purpose framework for **representation learning**

- Given an *objective*
- Learn *representation* that is required to achieve objective
- Directly from *raw inputs*
- Use minimal domain knowledge



Deep Reinforcement Learning

AI is an agent that can solve human-level task

- RL defines the objective
- DL gives the mechanism
- RL + DL = general intelligence

Deep RL AI Examples

Play games: Atari, poker, Go, …

Explore worlds: 3D worlds, …

Control physical systems: manipulate, …

Interact with users: recommend, optimize, personalize, …

Introduction to RL

Reinforcement Learning

Outline

Machine Learning

- Supervised Learning v.s. Reinforcement Learning
- Reinforcement Learning v.s. Deep Learning

Introduction to Reinforcement Learning

- Agent and Environment
- Action, State, and Reward

Markov Decision Process

Reinforcement Learning Approach

- Value-Based
- Policy-Based
- Model-Based

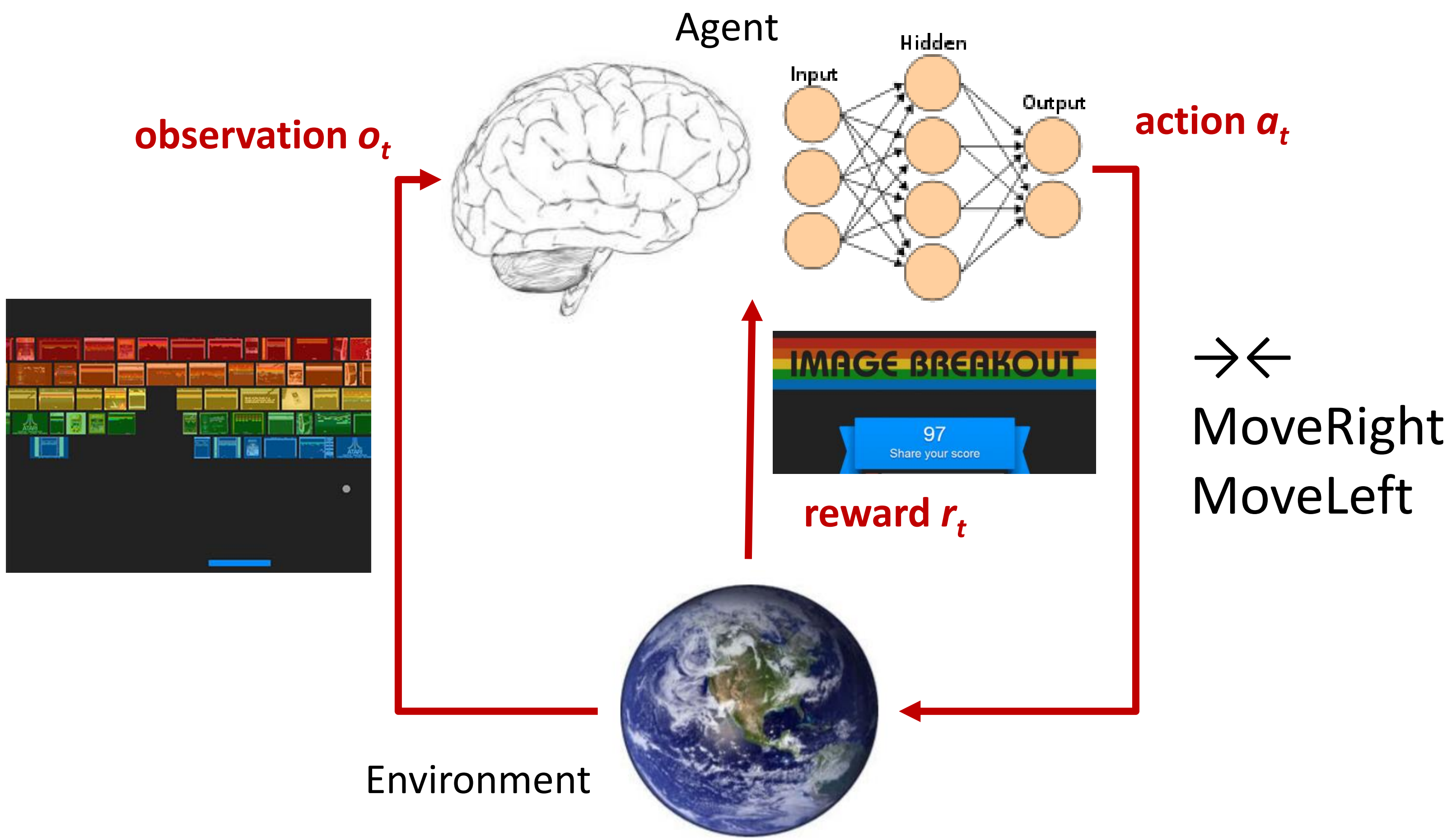
Reinforcement Learning

RL is a general purpose framework for **decision making**

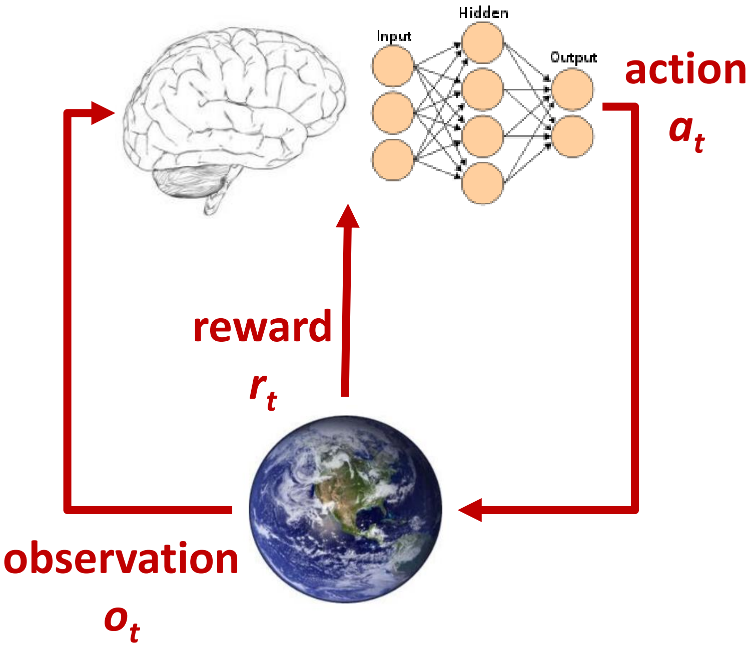
- RL is for an *agent* with the capacity to *act*
- Each *action* influences the agent's future *state*
- Success is measured by a scalar *reward* signal

Big three: action, state, reward

Agent and Environment



Agent and Environment



At time step t

- The agent
  - Executes action a_t
  - Receives observation o_t
  - Receives scalar reward r_t
- The environment
  - Receives action a_t
  - Emits observation o_{t+1}
  - Emits scalar reward r_{t+1}
- t increments at env. step

State

Experience is the sequence of observations, actions, rewards

$$o_1, r_1, a_1, \ldots, a_{t-1}, o_t, r_t$$

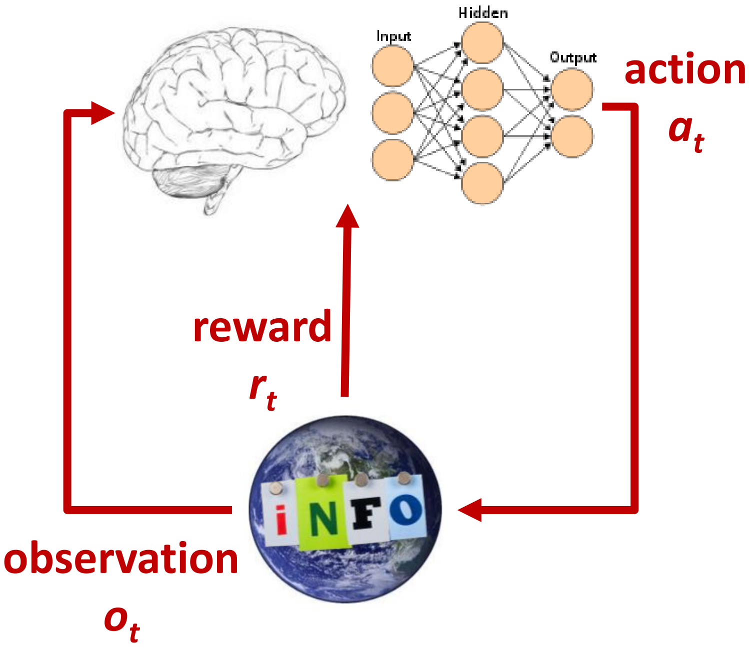
State is the information used to determine what happens next

- what happens depends on the history experience
  - The agent selects actions
  - The environment selects observations/rewards

The state is the function of the history experience

$$s_t = f(o_1, r_1, a_1, \ldots, a_{t-1}, o_t, r_t)$$

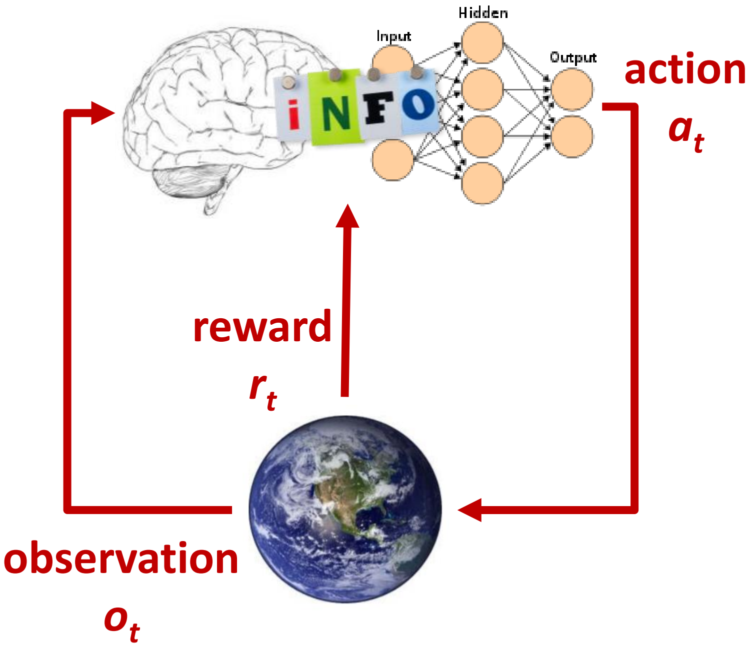
Environment State



The environment state s_t^e is the environment's *private* representation

- whether data the environment uses to pick the next observation/reward
- may not be visible to the agent
- may contain irrelevant information

Agent State



The agent state s_t^a is the agent's *internal* representation

- whether data the agent uses to pick the next action → information used by RL algorithms
- can be any function of experience

Information State

An information state (a.k.a. Markov state) contains all useful information from history

A state is Markov iff $P(s_{t+1} \mid s_t) = P(s_{t+1} \mid s_1, \ldots, s_t)$

The future is independent of the past given the present

$$H_t = \{o_1, r_1, a_1, \ldots, a_{t-1}, o_t, r_t\}$$

$$H_{1:t} \rightarrow s_t \rightarrow H_{t+1:\infty}$$

- Once the state is known, the history may be thrown away
- The state is a sufficient statistics of the future

Fully Observable Environment

Full observability: agent *directly* observes environment state

$$o_t = s_t^a = s_t^e$$

information state = agent state = environment state

This is a Markov decision process (MDP)

Partially Observable Environment

Partial observability: agent *indirectly* observes environment

$$s_t^a \neq s_t^e$$

agent state ≠ environment state

This is partially observable Markov decision process (POMDP)

Agent must construct its own state representation s_t^a

- Complete history: $s_t^a = H_t$
- Beliefs of environment state: $s_t^a = \{P(s_t^e = s^1), ..., P(s_t^e = s^n)\}$
- Hidden state (from RNN): $s_t^a = \sigma(W_s \cdot s_{t-1}^a + W_o \cdot o_t)$

Reward

Reinforcement learning is based on reward hypothesis

A reward r_t is a scalar feedback signal

- Indicates how well agent is doing at step t

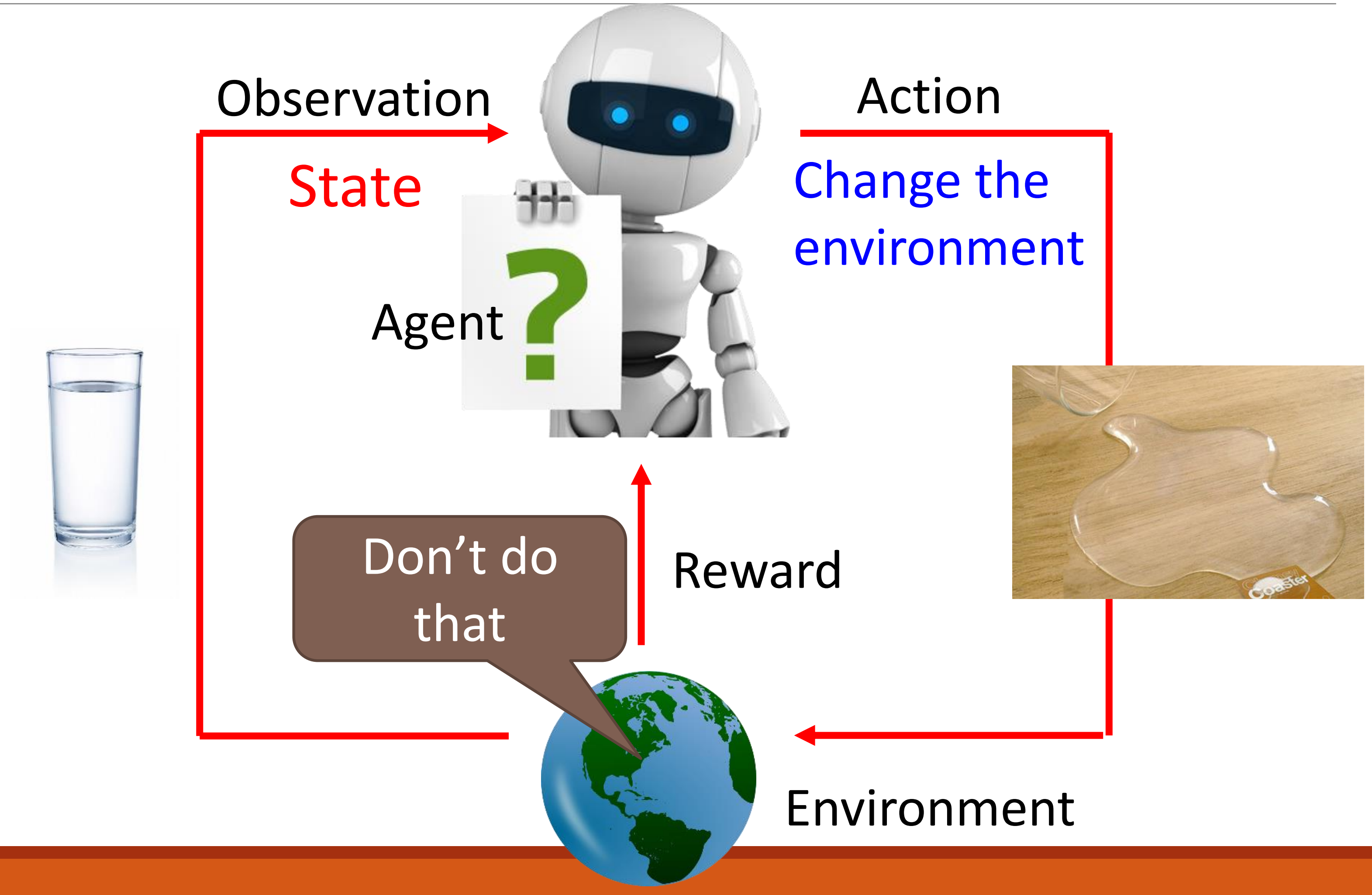
Reward hypothesis: all agent goals can be desired by maximizing expected cumulative reward

Sequential Decision Making

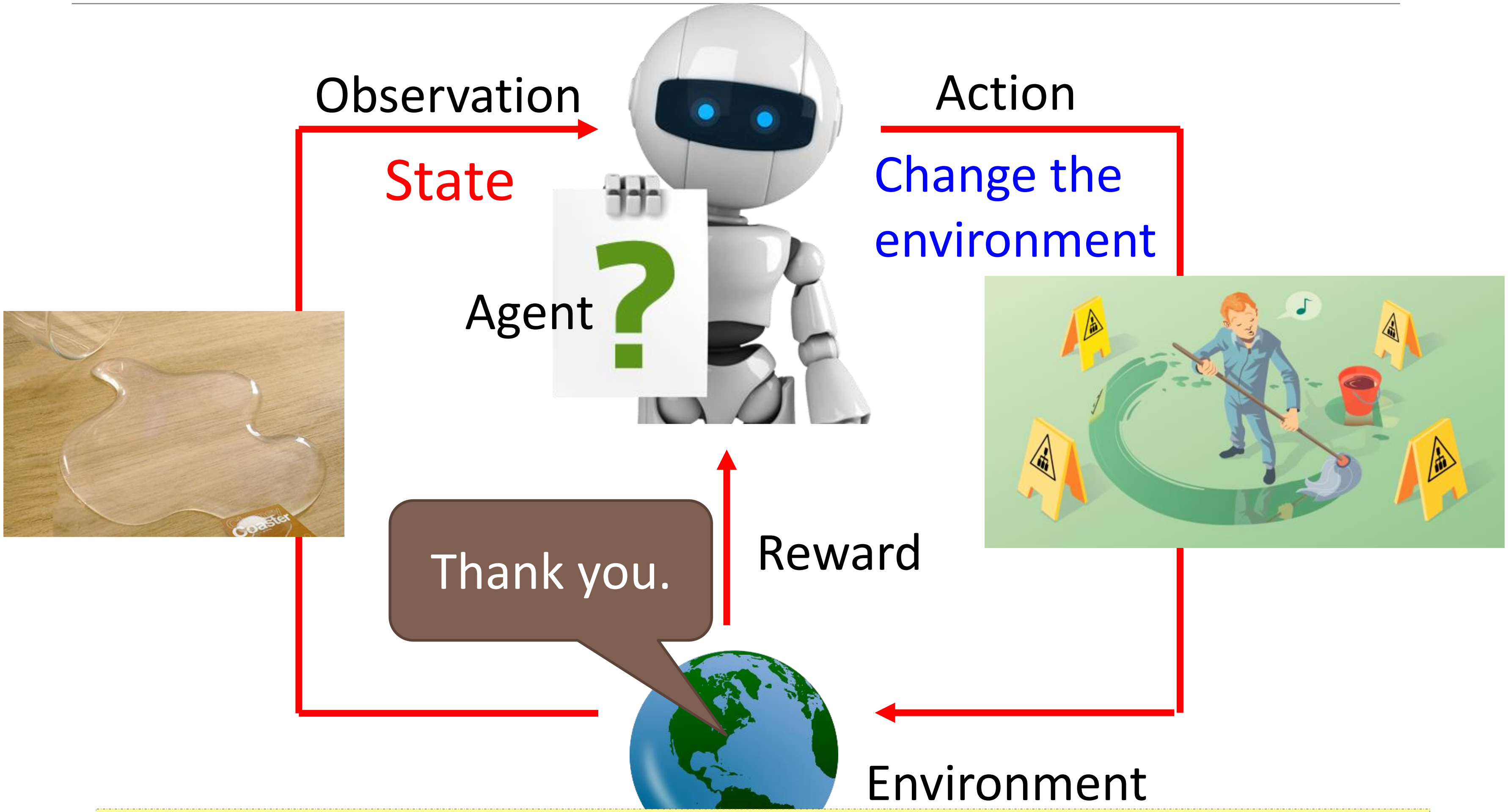
Goal: select actions to maximize total future reward

- Actions may have long-term consequences
- Reward may be delayed
- It may be better to sacrifice immediate reward to gain more long-term reward

Scenario of Reinforcement Learning

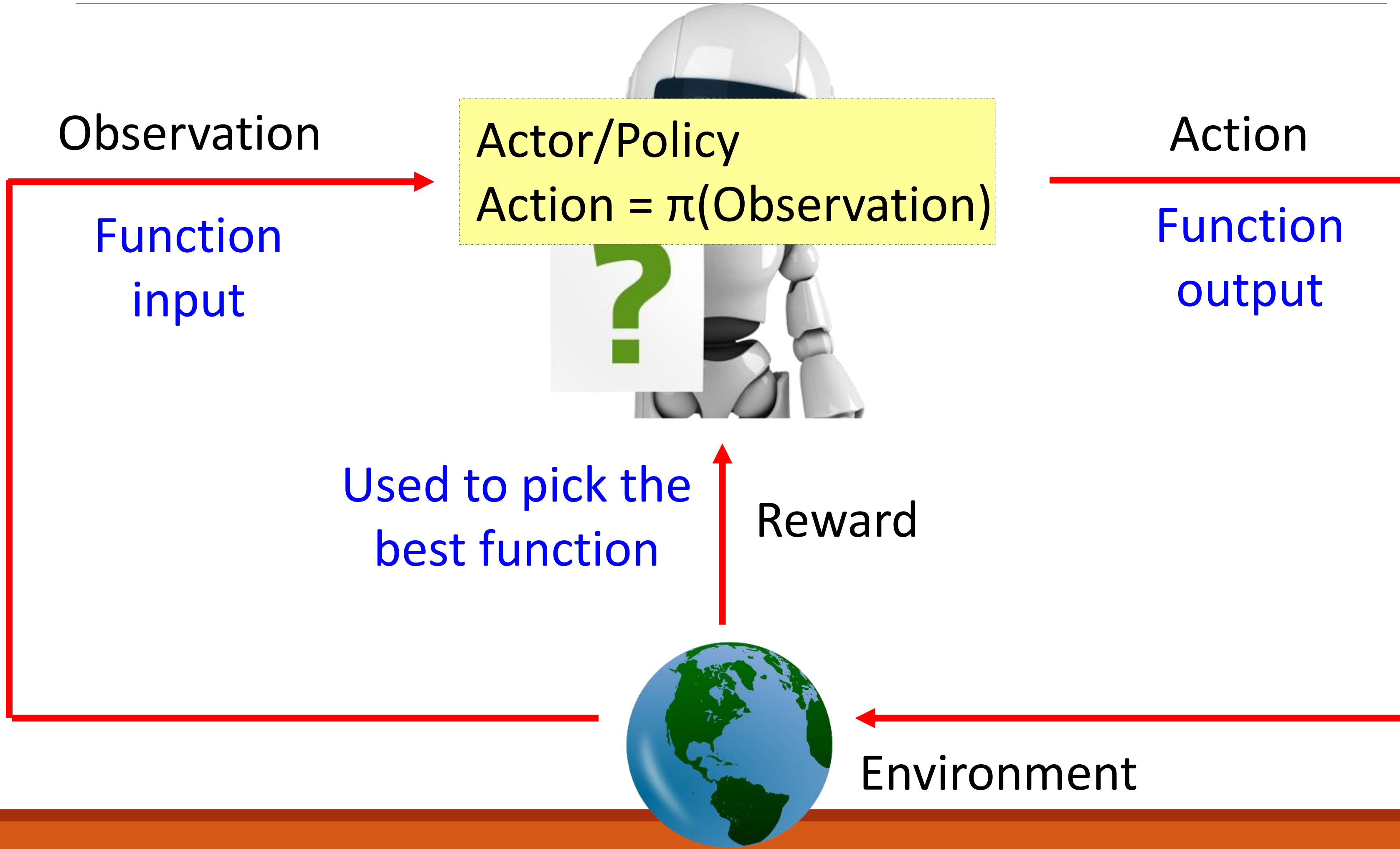


Scenario of Reinforcement Learning

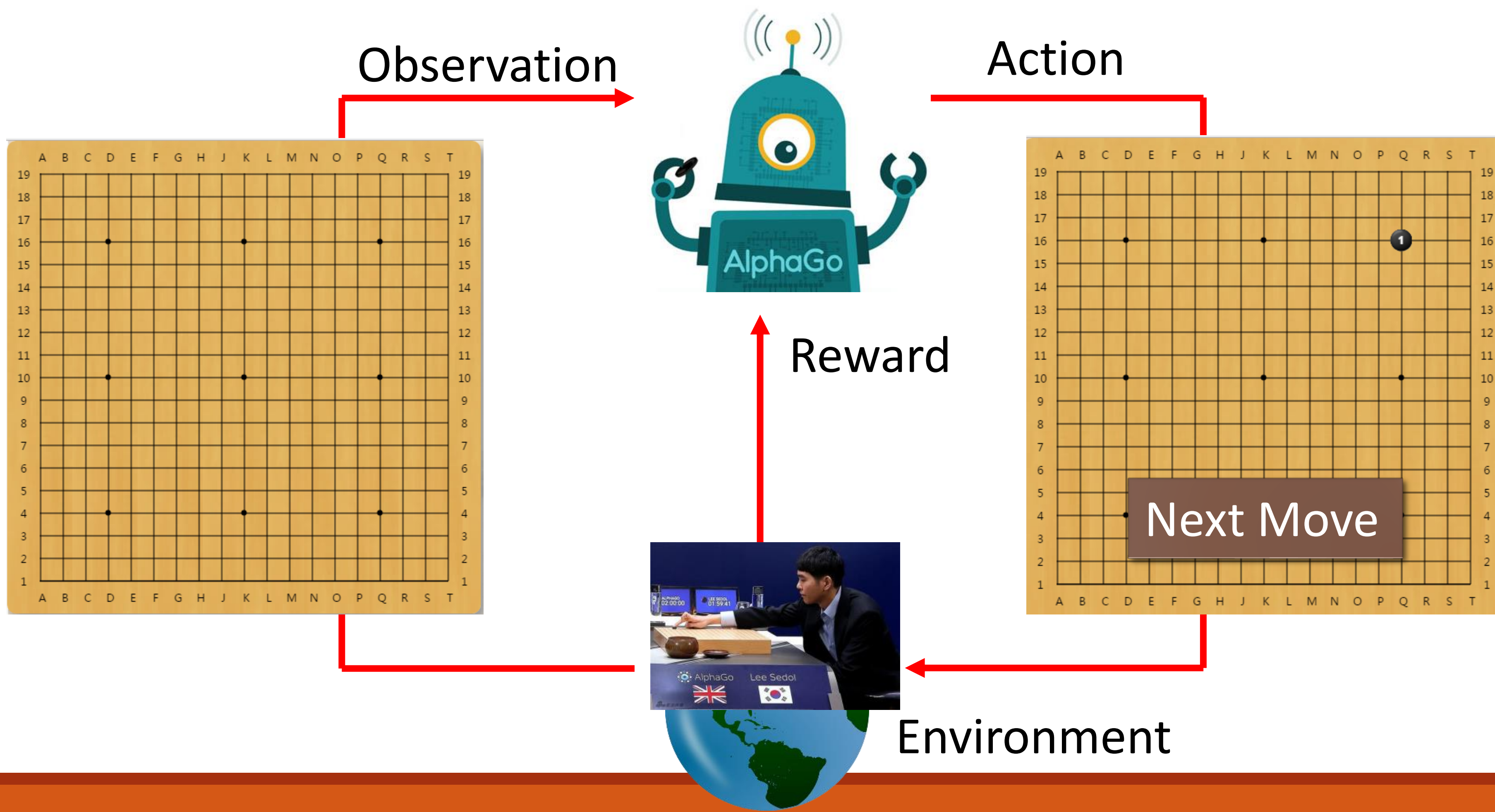


Agent learns to take actions maximizing expected reward.

Machine Learning ≈ Looking for a Function



Learning to Play Go



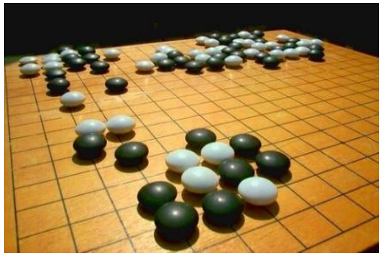
Learning to Play Go

Agent learns to take actions maximizing expected reward.

Learning to Play Go

Supervised **Learning from teacher**

Next move: "5-5"



Next move: "3-3"

Reinforcement Learning **Learning from experience**

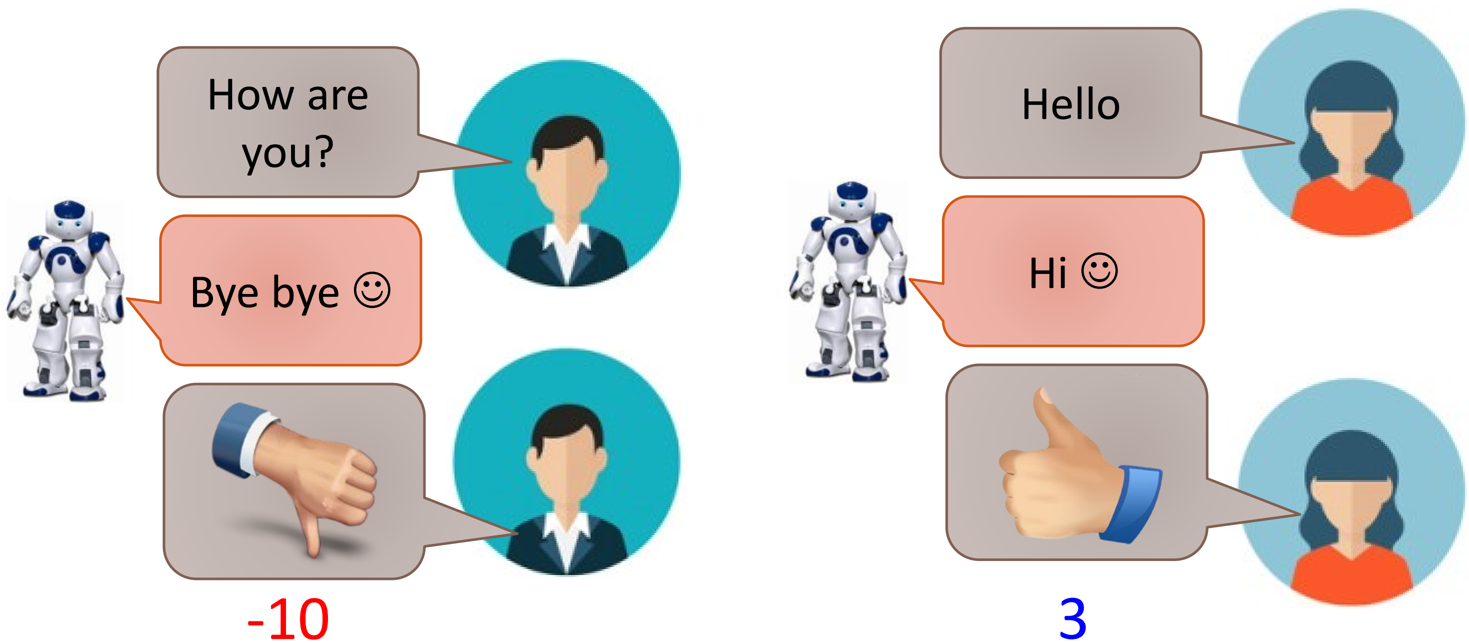
First move → …… many moves …… → Win!

(Two agents play with each other.)

AlphaGo uses supervised learning + reinforcement learning.

Learning a Chatbot

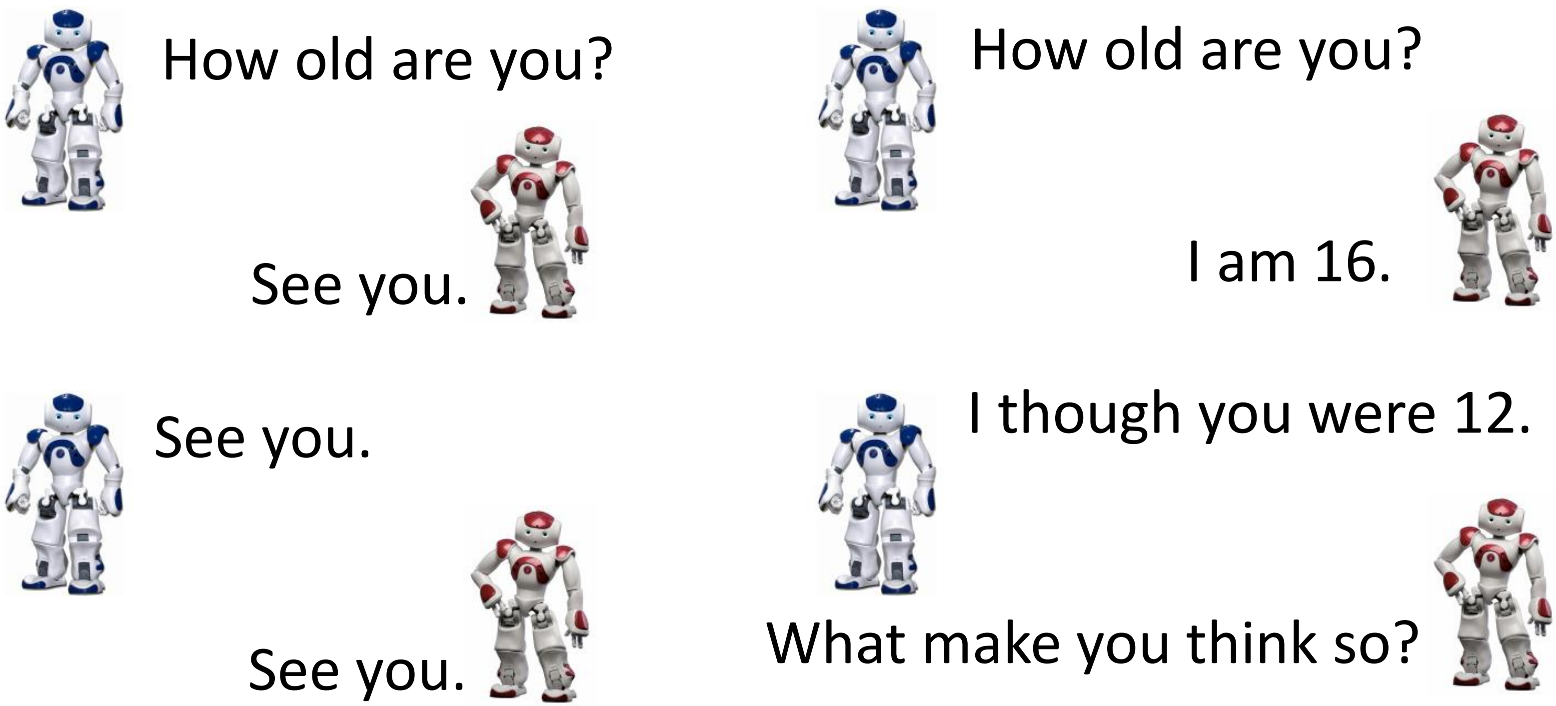
Machine obtains feedback from user



Chatbot learns to maximize the ***expected reward***

Learning a Chatbot

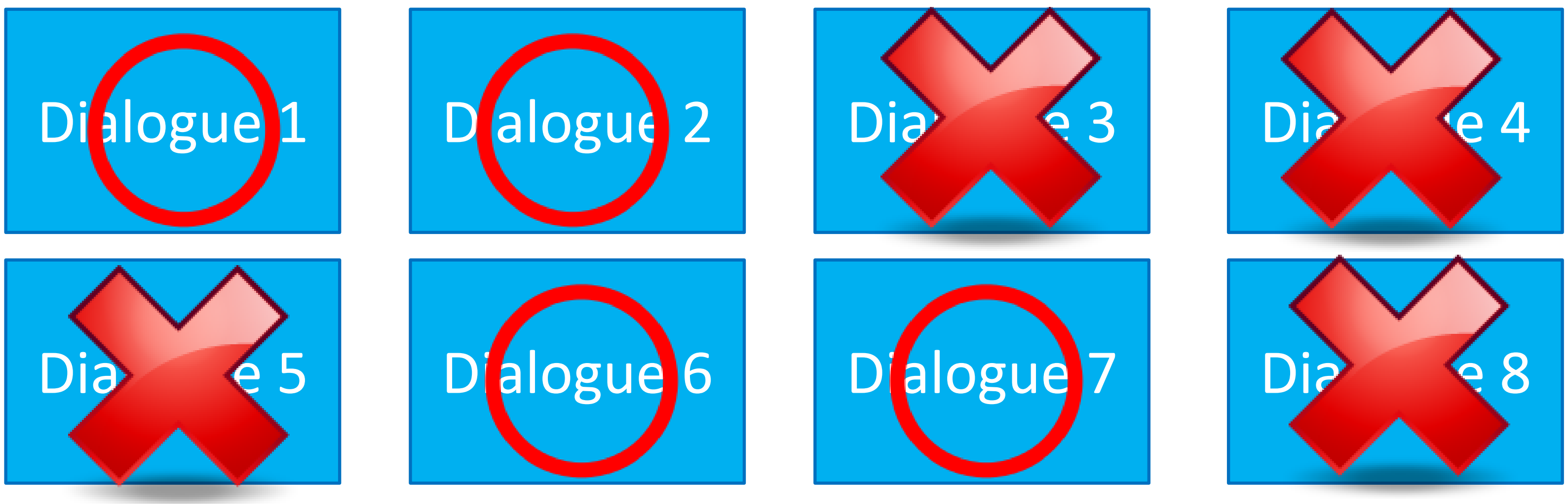
Let two agents talk to each other (sometimes generate good dialogue, sometimes bad)



Learning a chat-bot

By this approach, we can generate a lot of dialogues.

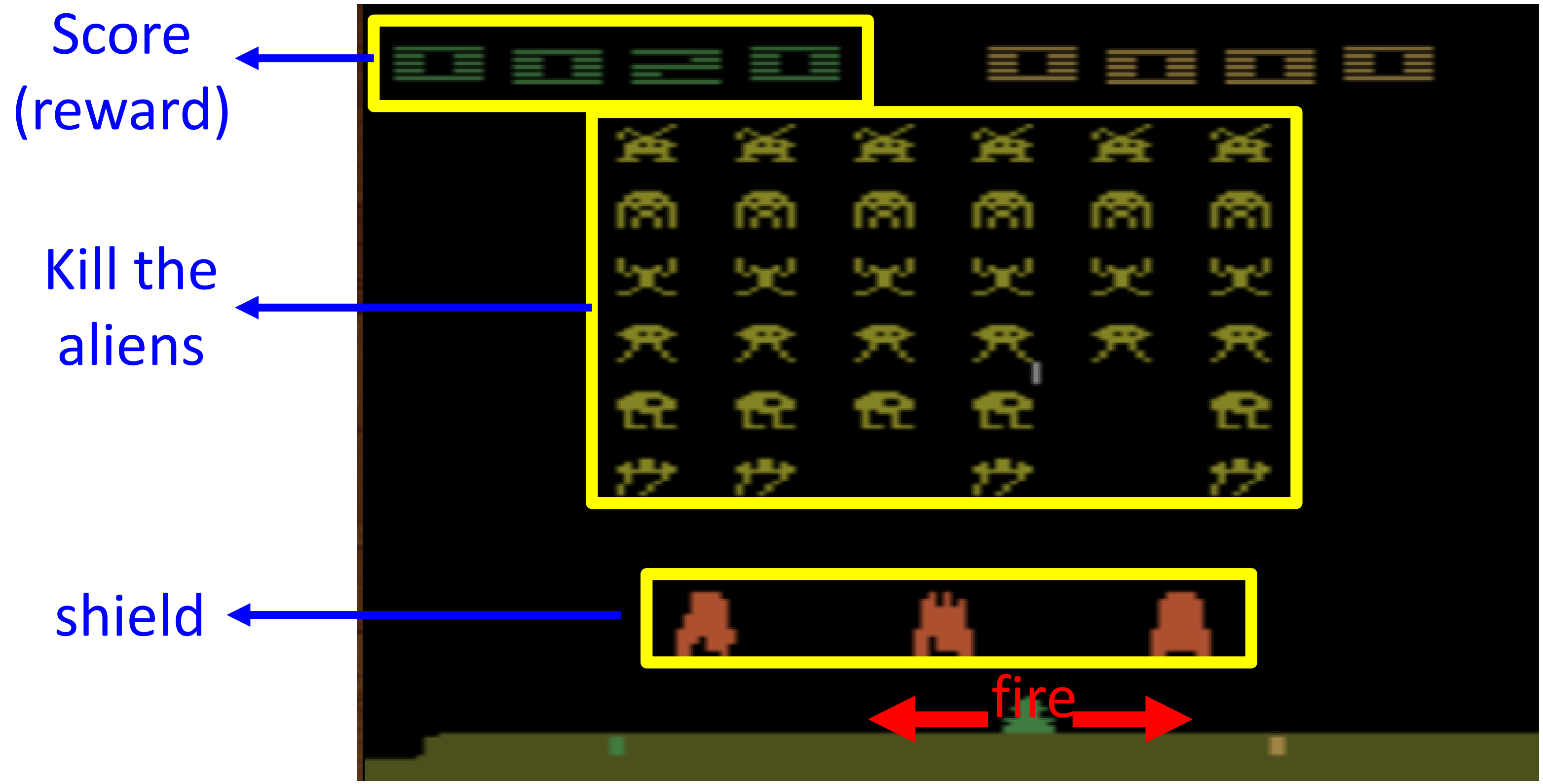
Use pre-defined rules to evaluate the goodness of a dialogue



Machine learns from the evaluation as rewards

Learning to Play Video Game

Space invader: terminate when all aliens are killed, or your spaceship is destroyed

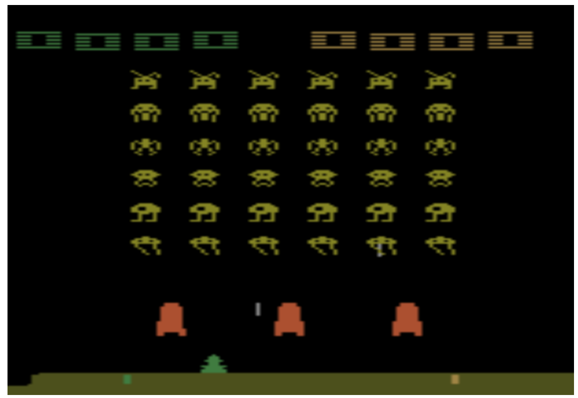


Play yourself: http://www.2600online.com/spaceinvaders.html
How about machine: https://gym.openai.com/evaluations/eval_Eduozx4HRyqgTCVk9ltw

Learning to Play Video Game

Start with observation s_1

Observation s_2



Observation s_3

Obtain reward $r_1 = 0$

Action a_1: "right"

Obtain reward $r_2 = 5$

Action a_2: "fire"

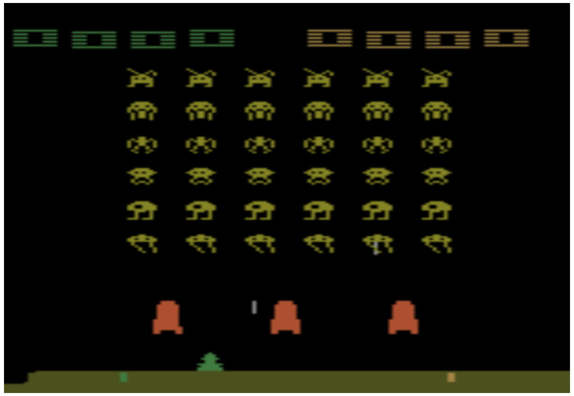
(kill an alien)

Usually there is some randomness in the environment

Learning to Play Video Game

Start with observation s_1

Observation s_2



Observation s_3

After many turns

Game Over (spaceship destroyed)

Action a_T

Obtain reward r_T

This is an ***episode***.

Learn to maximize the expected cumulative reward per episode

More Applications

Flying Helicopter

- https://www.youtube.com/watch?v=0JL04JJjocc

Driving

- https://www.youtube.com/watch?v=0xo1Ldx3L5Q

Robot

- https://www.youtube.com/watch?v=370cT-OAzzM

Google Cuts Its Giant Electricity Bill With DeepMind-Powered AI

- http://www.bloomberg.com/news/articles/2016-07-19/google-cuts-its-giant-electricity-bill-with-deepmind-powered-ai

Text Generation

- https://www.youtube.com/watch?v=pbQ4qe8EwLo

Markov Decision Process

Fully Observable Environment

Outline

Machine Learning

- Supervised Learning v.s. Reinforcement Learning
- Reinforcement Learning v.s. Deep Learning

Introduction to Reinforcement Learning

- Agent and Environment
- Action, State, and Reward

Markov Decision Process

Reinforcement Learning Approach

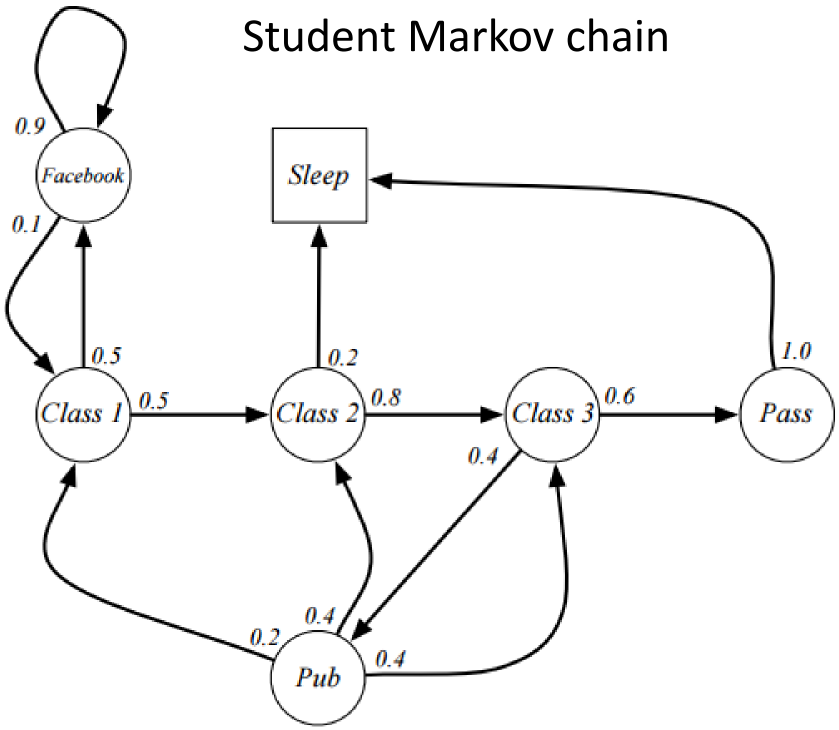
- Value-Based
- Policy-Based
- Model-Based

Markov Process

Markov process is a memoryless random process

- i.e. a sequence of random states S_1, S_2, ... with the Markov property

Student Markov chain



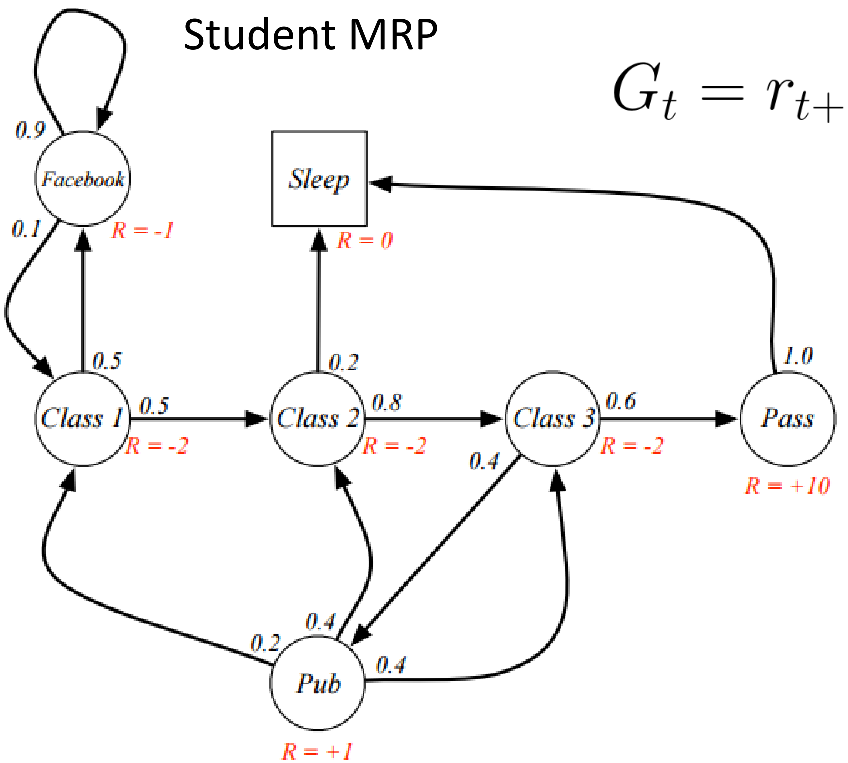
Sample episodes from S_1=C1

- C1 C2 C3 Pass Sleep
- C1 FB FB C1 C2 Sleep
- C1 C2 C3 Pub C2 C3 Pass Sleep
- C1 FB FB C1 C2 C3 Pub
- C1 FB FB FB C1 C2 C3 Pub C2 Sleep

Markov Reward Process (MRP)

Markov reward process is a Markov chain with values

- The return G_t is the total discounted reward from time-step t



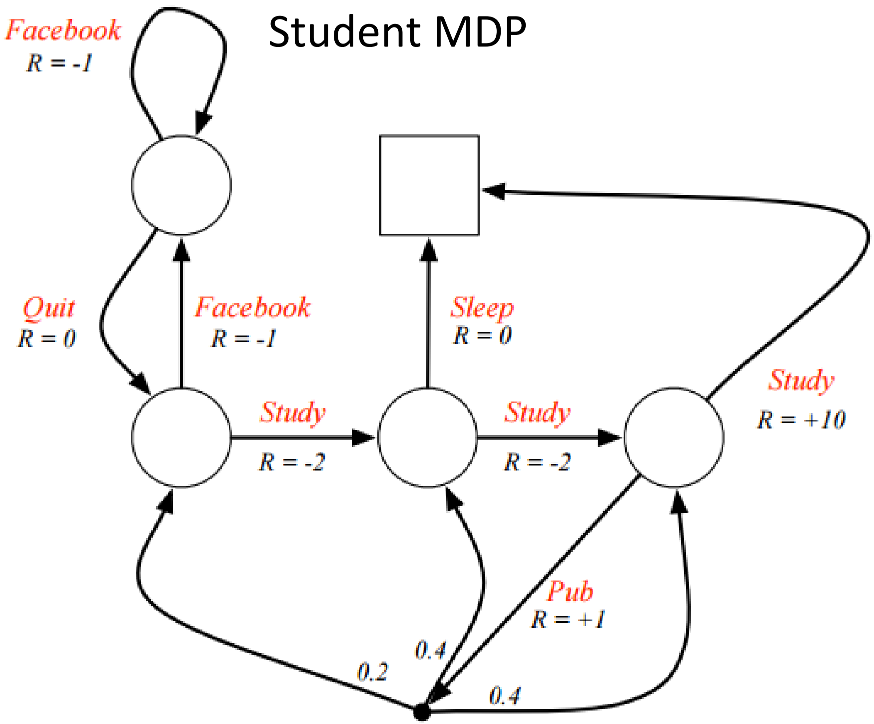
$$G_t = r_{t+1} + \gamma r_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

Markov Decision Process (MDP)

Markov decision process is a MRP with decisions

- It is an environment in which all states are Markov

Student MDP



Markov Decision Process (MDP)

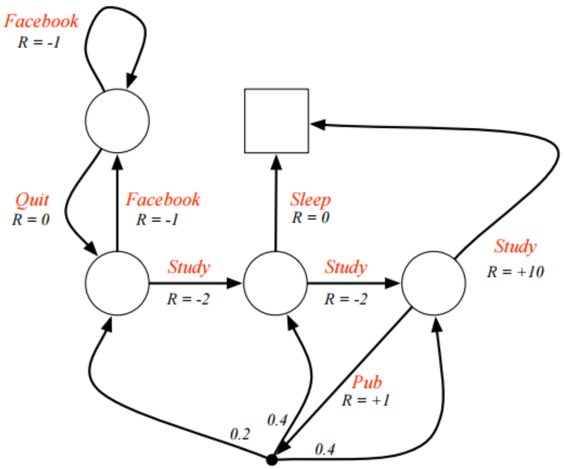
S : finite set of states/observations

A : finite set of actions

P : transition probability

R : immediate reward

γ : discount factor



Goal is to choose policy π at time *t* that maximizes expected overall **return**:

$$\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$$

Reinforcement Learning

Outline

Machine Learning
- Supervised Learning v.s. Reinforcement Learning
- Reinforcement Learning v.s. Deep Learning

Introduction to Reinforcement Learning
- Agent and Environment
- Action, State, and Reward

Markov Decision Process

Reinforcement Learning
- Value-Based
- Policy-Based
- Model-Based

Major Components in an RL Agent

An RL agent may include one or more of these components

- **Value function**: how good is each state and/or action
- **Policy**: agent's behavior function
- **Model**: agent's representation of the environment

Reinforcement Learning Approach

Value-based RL

- Estimate the optimal value function $Q^*(s, a)$

$Q^*(s, a)$ is maximum value achievable under any policy

Policy-based RL

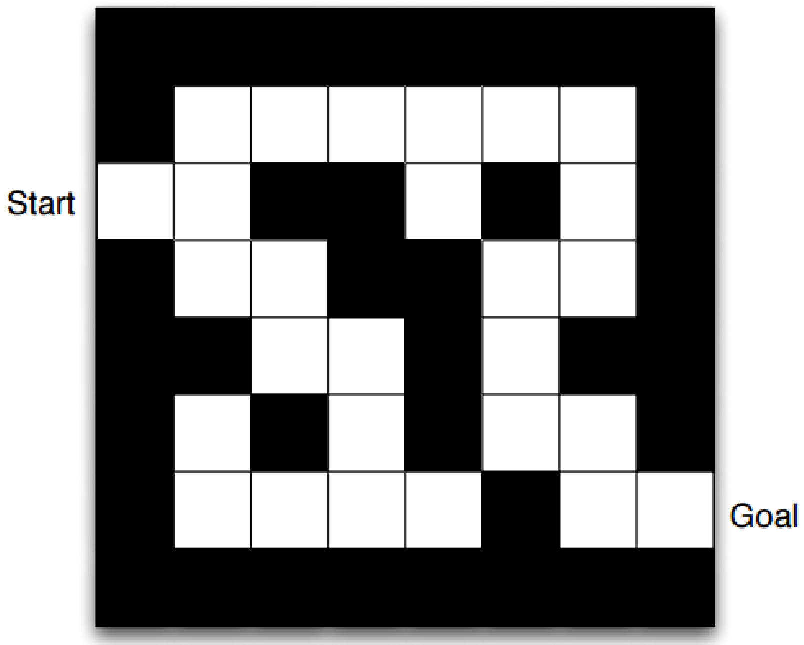
- Search directly for optimal policy π^*

π^* is the policy achieving maximum future reward

Model-based RL

- Build a model of the environment
- Plan (e.g. by lookahead) using model

Maze Example

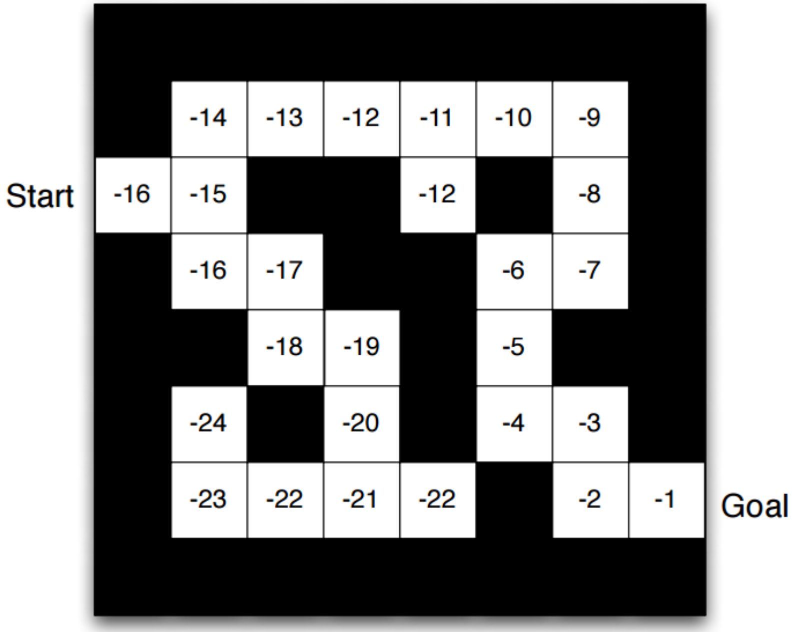


Rewards: -1 per time-step

Actions: N, E, S, W

States: agent's location

Maze Example: Value Function



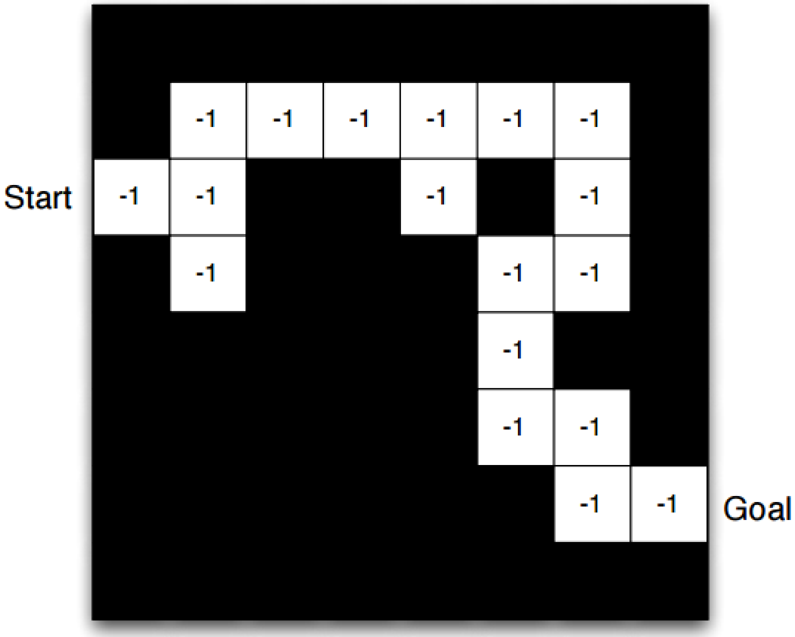
Rewards: -1 per time-step

Actions: N, E, S, W

States: agent's location

Numbers represent value $Q_\pi(s)$ of each state s

Maze Example: Value Function



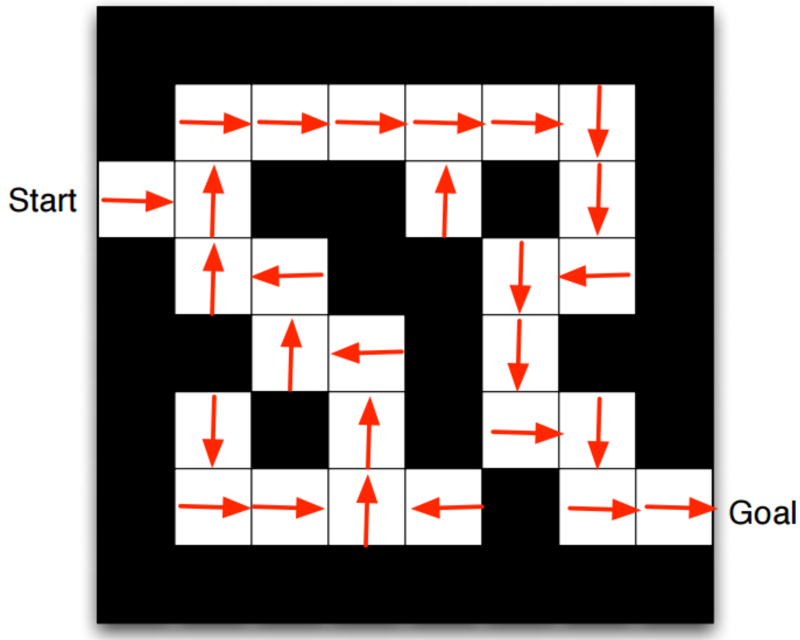
Rewards: -1 per time-step

Actions: N, E, S, W

States: agent's location

Grid layout represents transition model *P*
Numbers represent immediate reward *R* from each state *s* (same for all *a*)

Maze Example: Policy



Rewards: -1 per time-step

Actions: N, E, S, W

States: agent's location

Arrows represent policy $\pi(s)$ for each state s

Categorizing RL Agents

Value-Based
- No Policy (implicit)
- Value Function

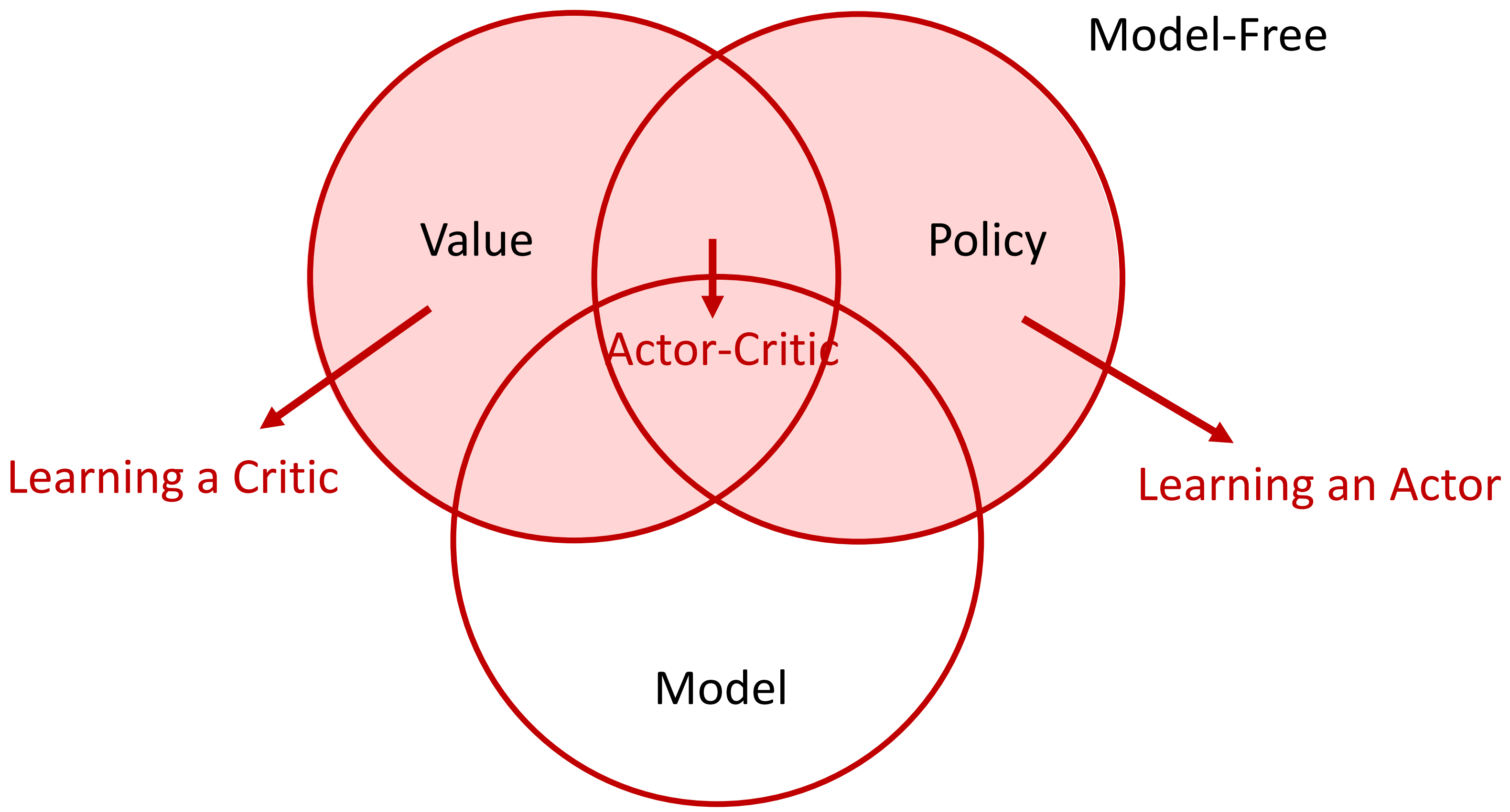
Policy-Based
- Policy
- No Value Function

Actor-Critic
- Policy
- Value Function

Model-Free
- Policy and/or Value Function
- No Model

Model-Based
- Policy and/or Value Function
- Model

RL Agent Taxonomy



Concluding Remarks

RL is a general purpose framework for **decision making** under interactions between *agent* and *environment*

- RL is for an *agent* with the capacity to ***act***
- Each *action* influences the agent's future ***state***
- Success is measured by a scalar ***reward*** signal
- Goal: *select actions to maximize future reward*

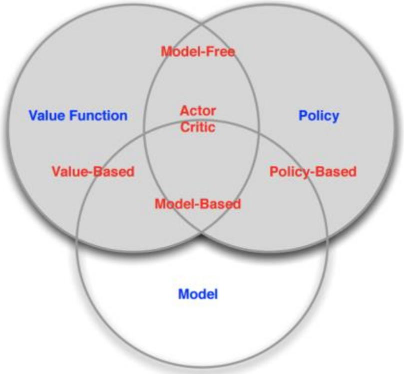
action

state

reward

An RL agent may include one or more of these components

- Value function: how good is each state and/or act
- Policy: agent's behavior function
- Model: agent's representation of the environmen



References

Course materials by David Silver: http://www0.cs.ucl.ac.uk/staff/D.Silver/web/Teaching.html

ICLR 2015 Tutorial: http://www.iclr.cc/lib/exe/fetch.php?media=iclr2015:silver-iclr2015.pdf

ICML 2016 Tutorial: http://icml.cc/2016/tutorials/deep_rl_tutorial.pdf