CPEN 455: Deep Learning

Lecture 12: Deep Reinforcement Learning Part I

Renjie Liao

University of British Columbia

Winter, Term 2, 2024

Outline

- Reinforcement Learning (RL) I
  - **Overview & Applications**
  - RL Formulation & Taxonomy
  - Markov Decision Process (MDP)
  - Bellman Equations
  - Solving RL problems using the Bellman Equations
- Reinforcement Learning (RL) II
  - Model-Free RL: (Deep) Q-learning
  - Model-Free RL: Policy Gradient & Actor-Critic
  - Model-Based RL: Dyna-Q

Reinforcement Learning (RL)

RL is about *learning to make decisions from interaction*!

Reinforcement Learning (RL)

RL is about *learning to make decisions from interaction*!

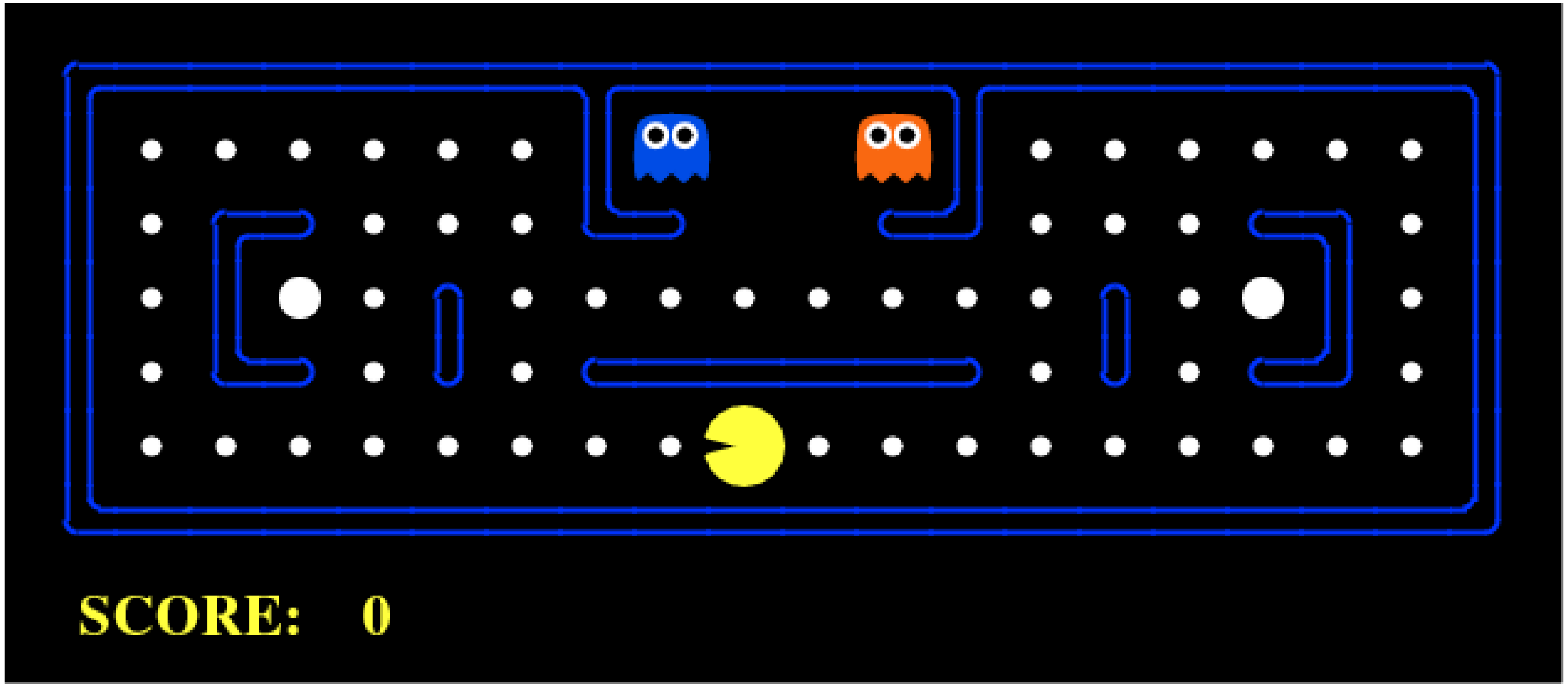
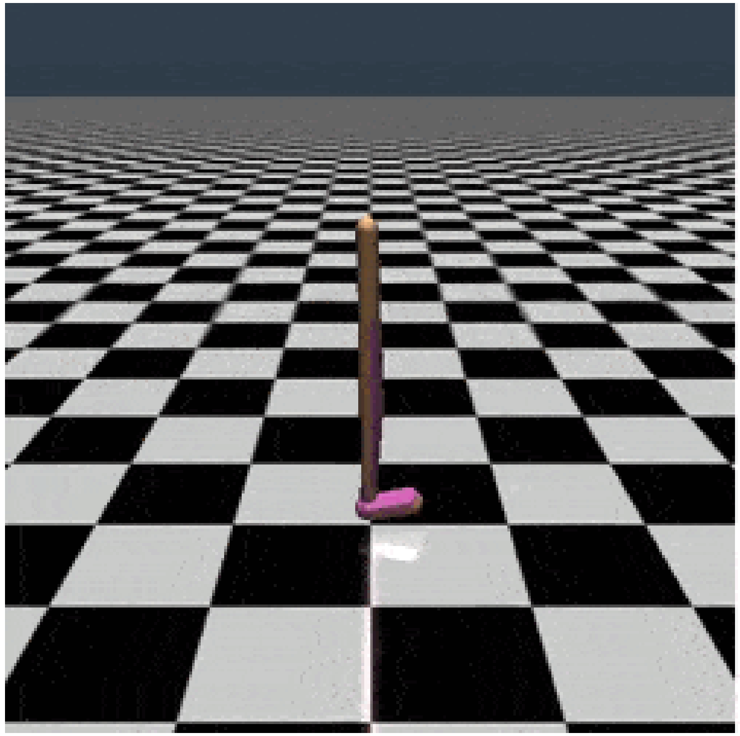


Image Credit: [1]

Reinforcement Learning (RL)

RL is about *learning to make decisions from interaction*!



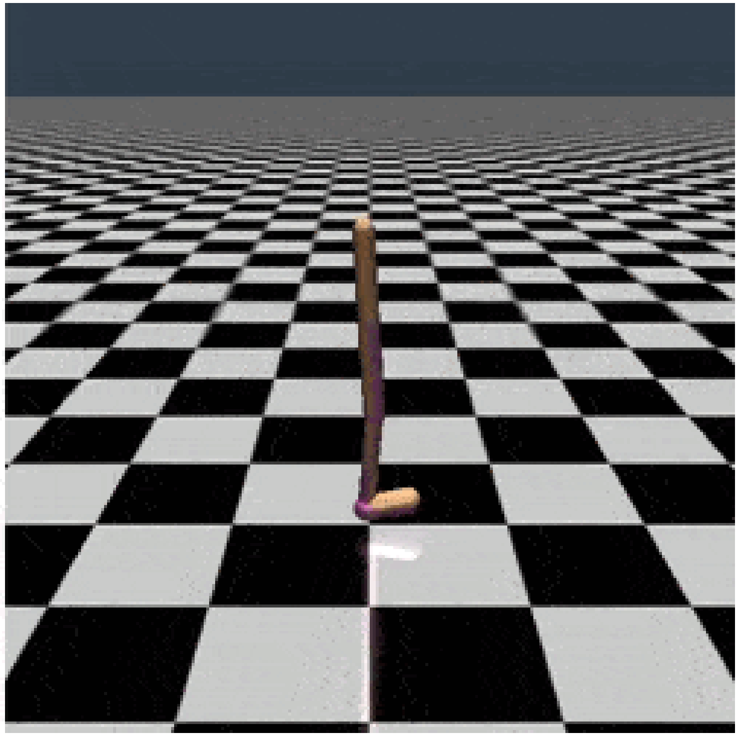


Image Credit: [2]

Reinforcement Learning (RL)

RL is about *learning to make decisions from interaction*!

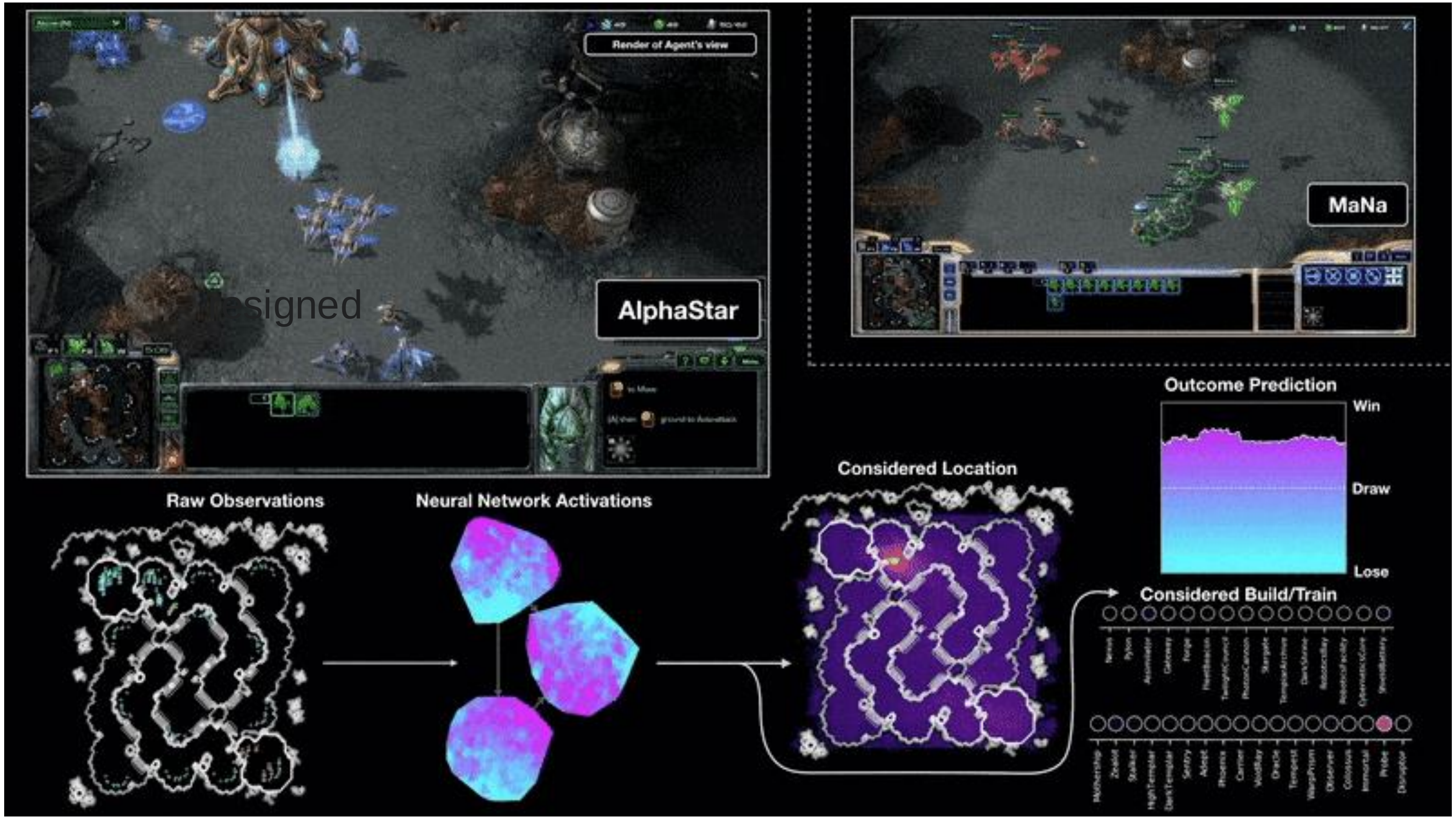
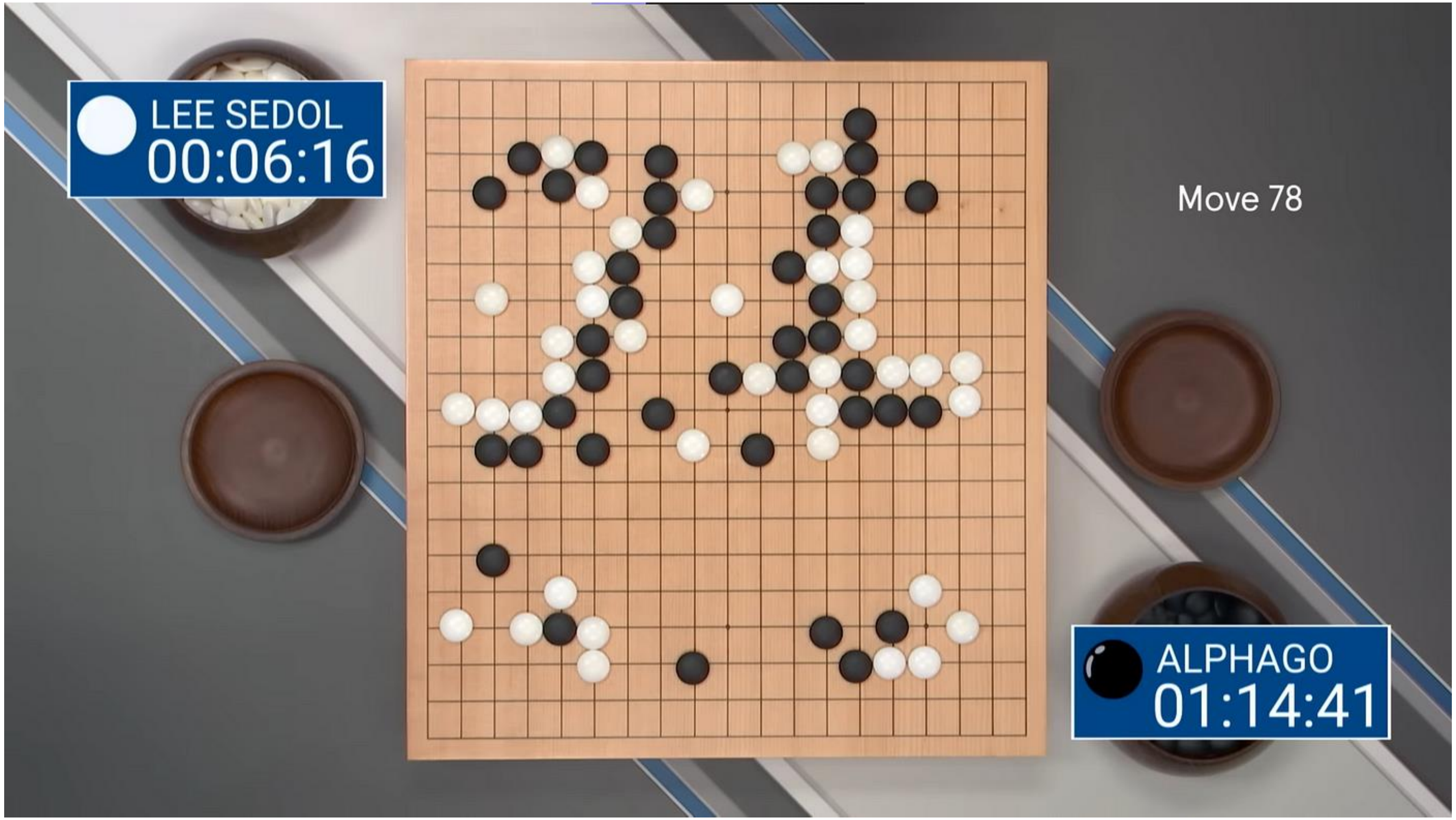


Image Credit: [3]

Reinforcement Learning (RL)

RL is about *learning to make decisions from interaction*!



God's move: AlphaGo thought this move happens with 0.007% probability in human players!

This may be the last time a human go player beats AI!

Image Credit: [4]

Reinforcement Learning (RL)

RL is about *learning to make decisions from interaction*!

Image Credit: [5]

Reinforcement Learning (RL)

RL is about *learning to make decisions from interaction*!

Image Credit: [6]

Reinforcement Learning (RL)

RL is about *learning to make decisions from interaction*!

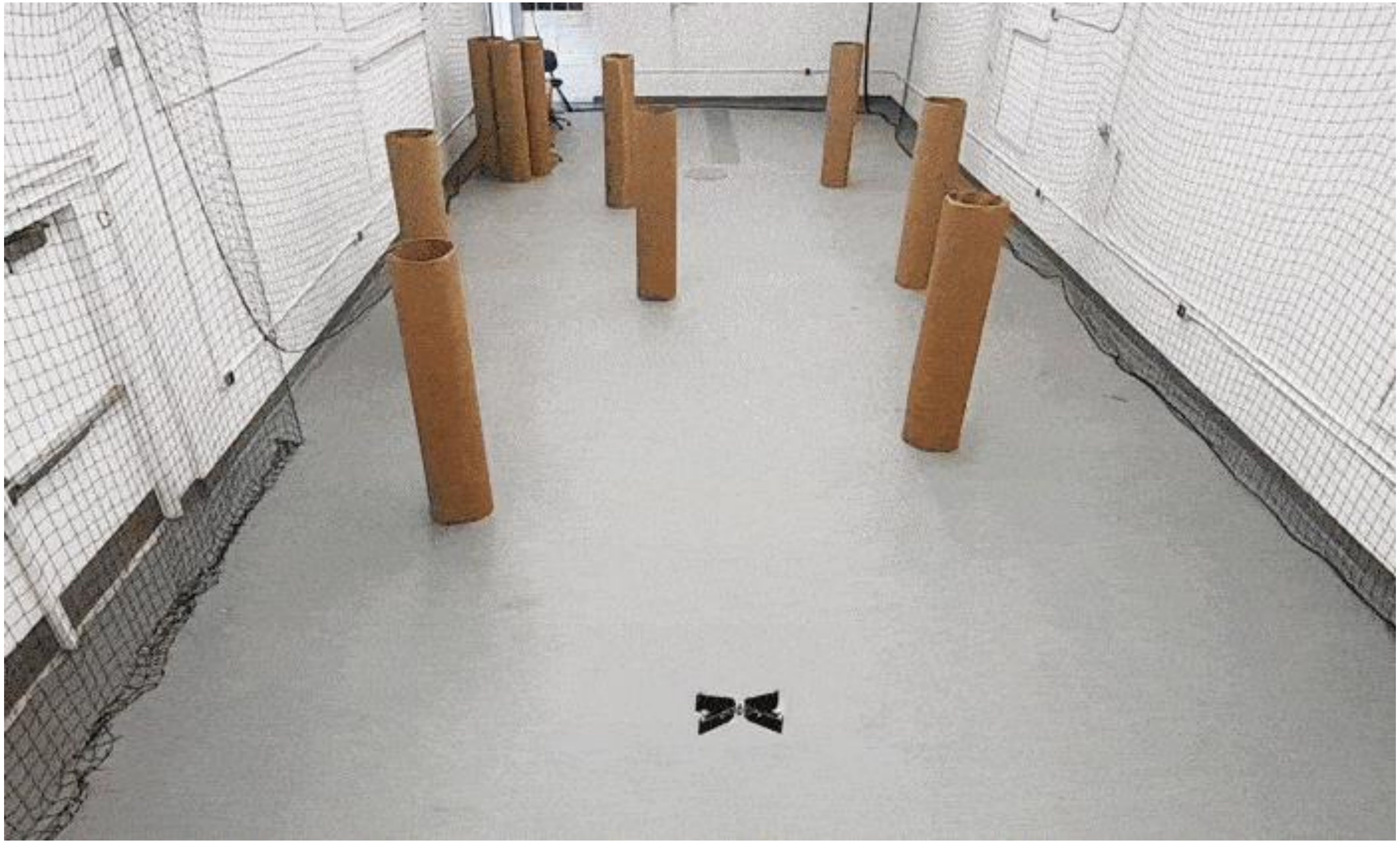


Image Credit: [7]

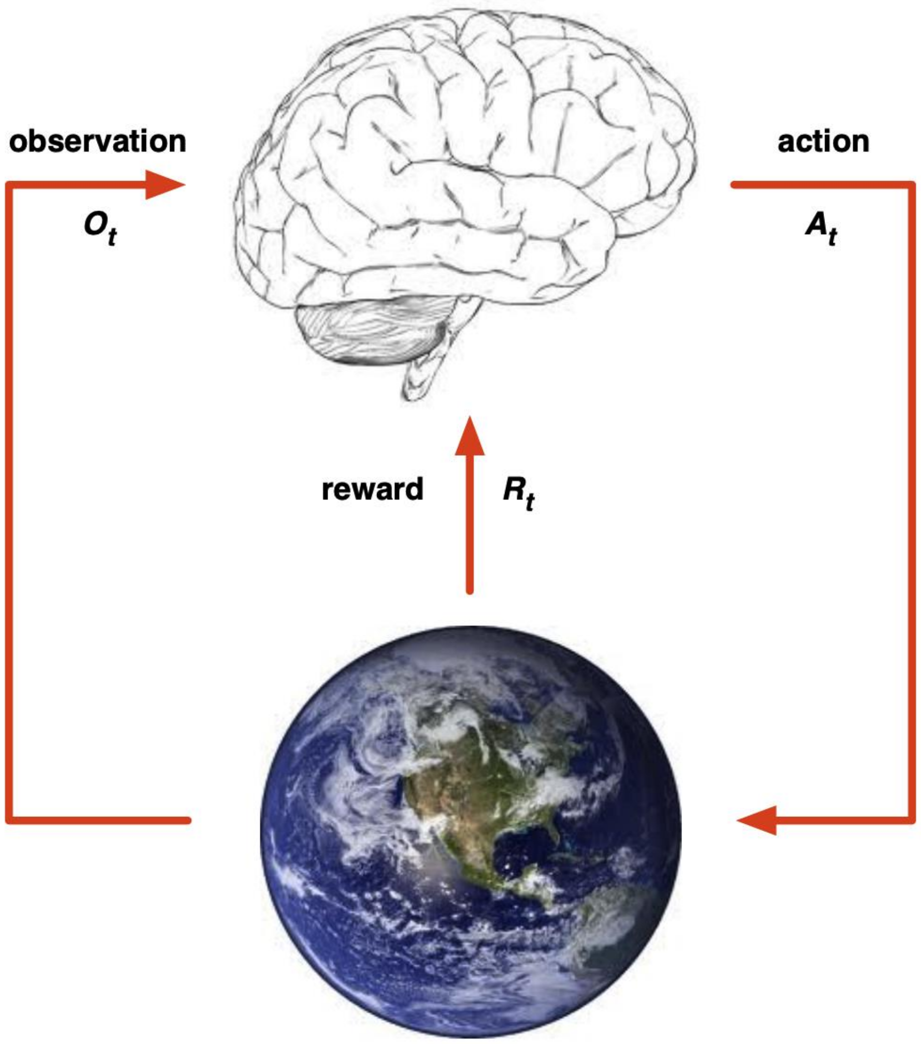
Reinforcement Learning (RL)

Image Credit: [8]

Outline

- Reinforcement Learning (RL) I
  - Overview & Applications
  - **RL Formulation & Taxonomy**
  - Markov Decision Process (MDP)
  - Bellman Equations
  - Solving RL problems using the Bellman Equations
- Reinforcement Learning (RL) II
  - Model-Free RL: (Deep) Q-learning
  - Model-Free RL: Policy Gradient & Actor-Critic
  - Model-Based RL: Dyna-Q

RL Formulation

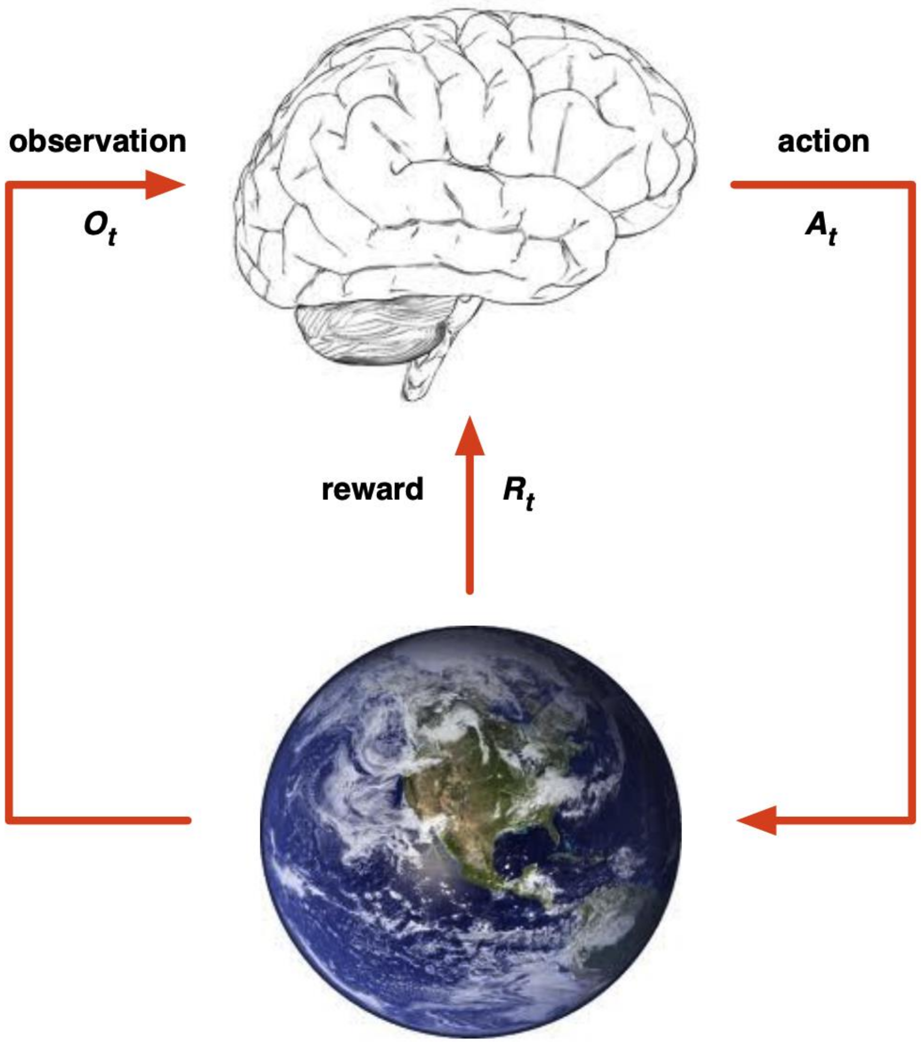


The Interaction Loop

- **Agent**: an intelligent program or a real robot
- **Environment**: the (simulated/real) “world” where the agent interacts
- **Reward**: a scalar feedback signal that defines the goal

Image Credit: [9]

RL Formulation



The Interaction Loop

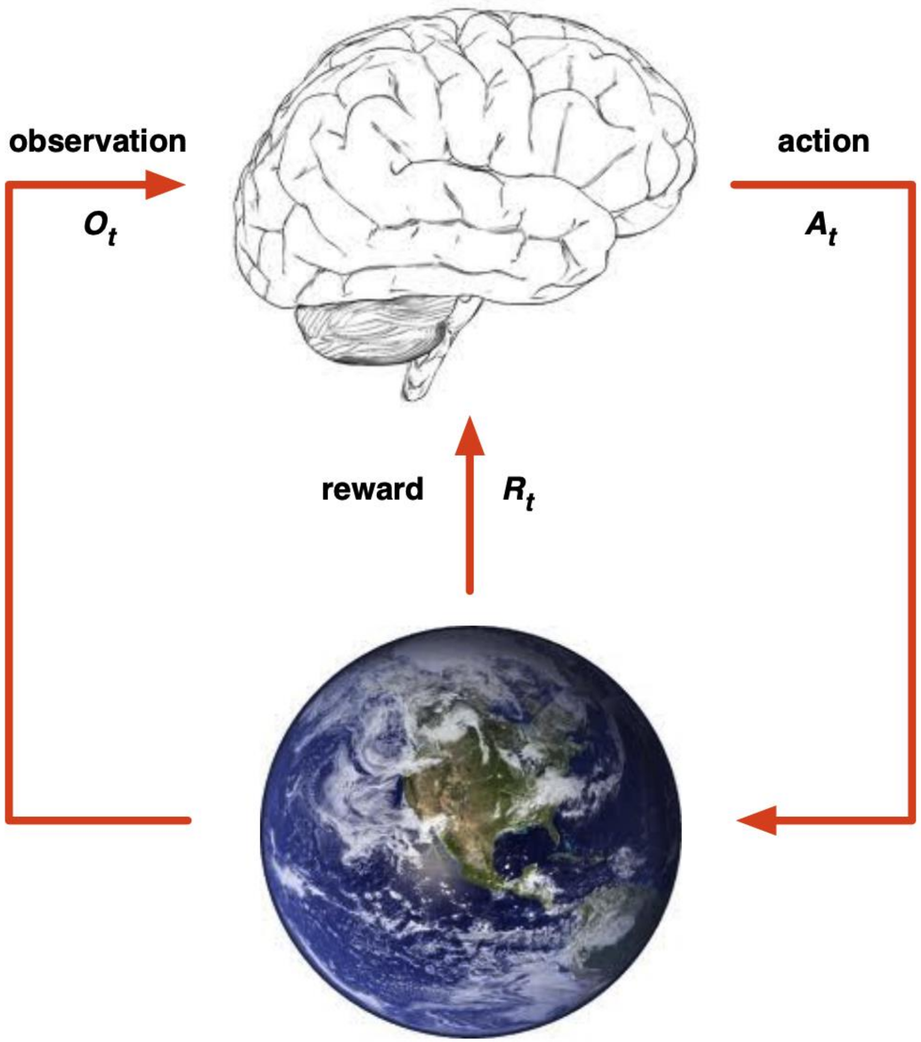
- **Agent**: an intelligent program or a real robot
- **Environment**: the (simulated/real) “world” where the agent interacts
- **Reward**: a scalar feedback signal that defines the goal

At each time step t:

- Agent receives observation O_t and reward R_t, and then executes action A_t
- Environment receives action A_t and emits observation O_{t+1} and reward R_{t+1}

Image Credit: [9]

RL Formulation

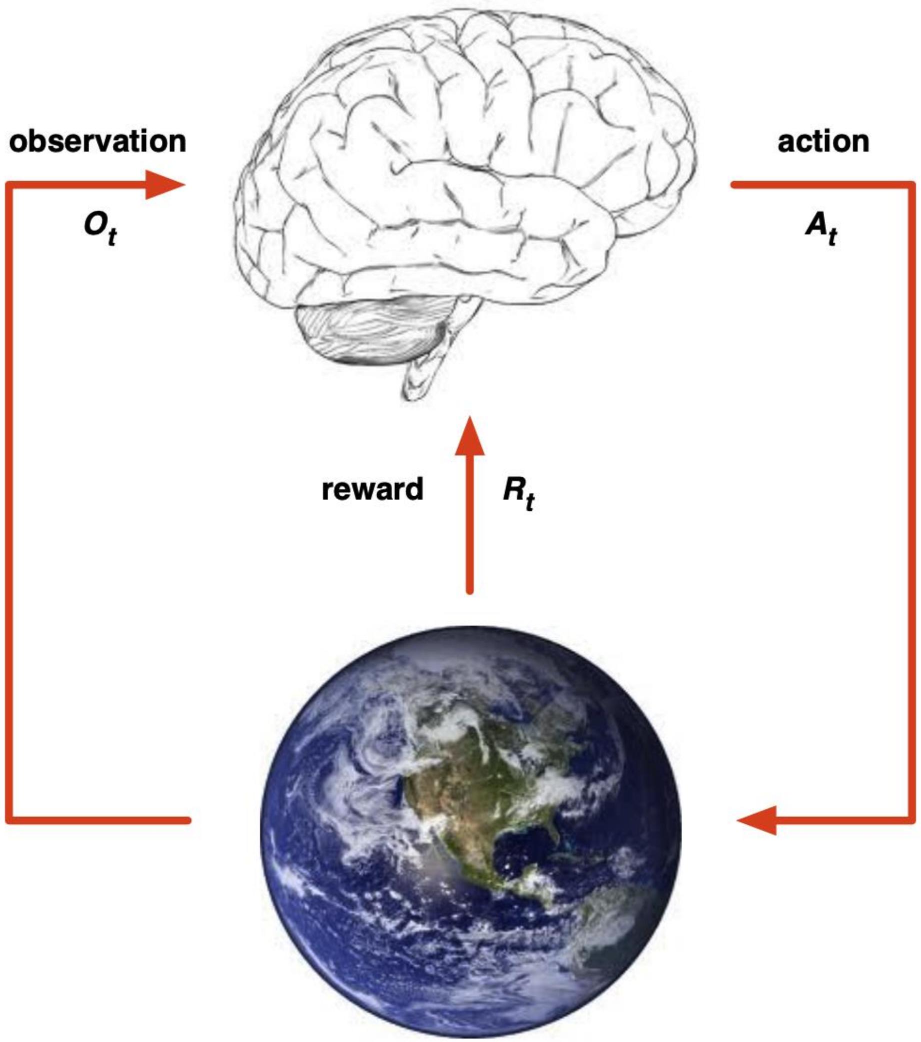


The agent's job is to maximize cumulative reward G_t:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots$$

Image Credit: [9]

RL Formulation



The agent's job is to maximize cumulative reward G_t:

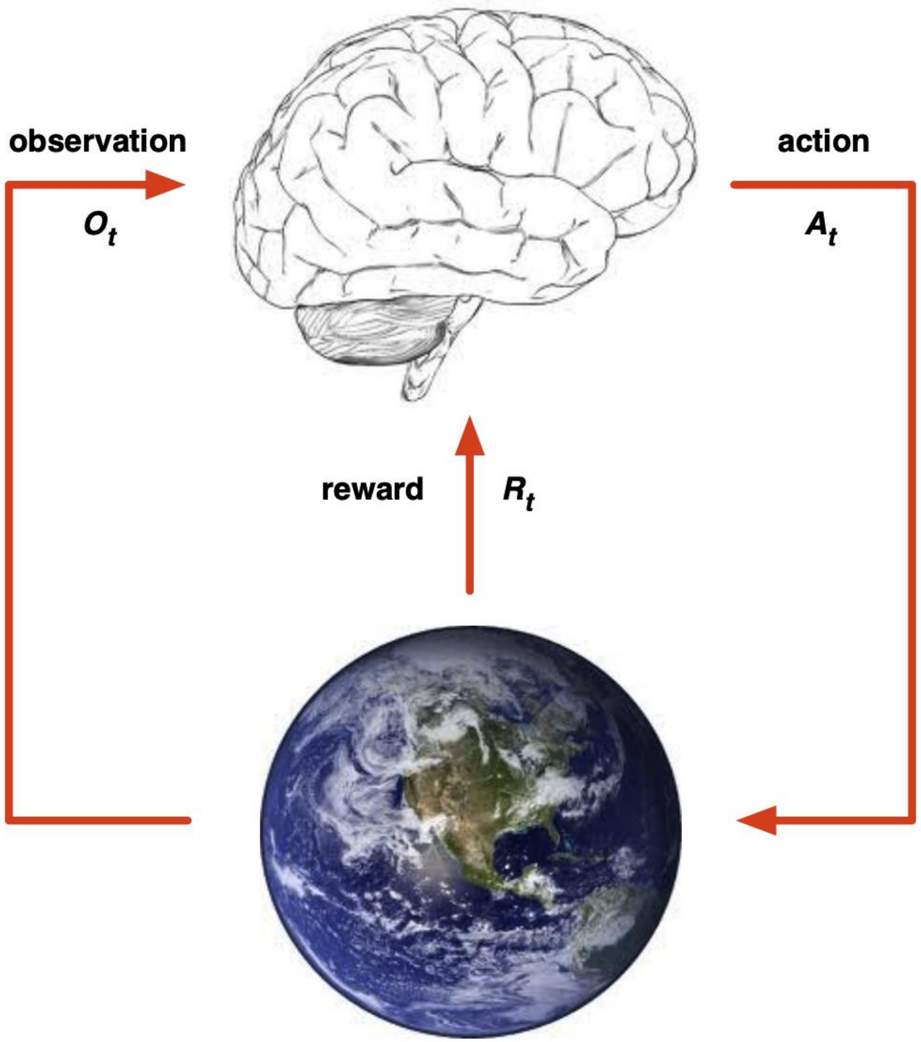
$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots$$

The reward hypothesis [11]:

All of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward).

Image Credit: [9]

RL Formulation



The agent's job is to maximize cumulative reward G_t:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots$$

The reward hypothesis [11]:

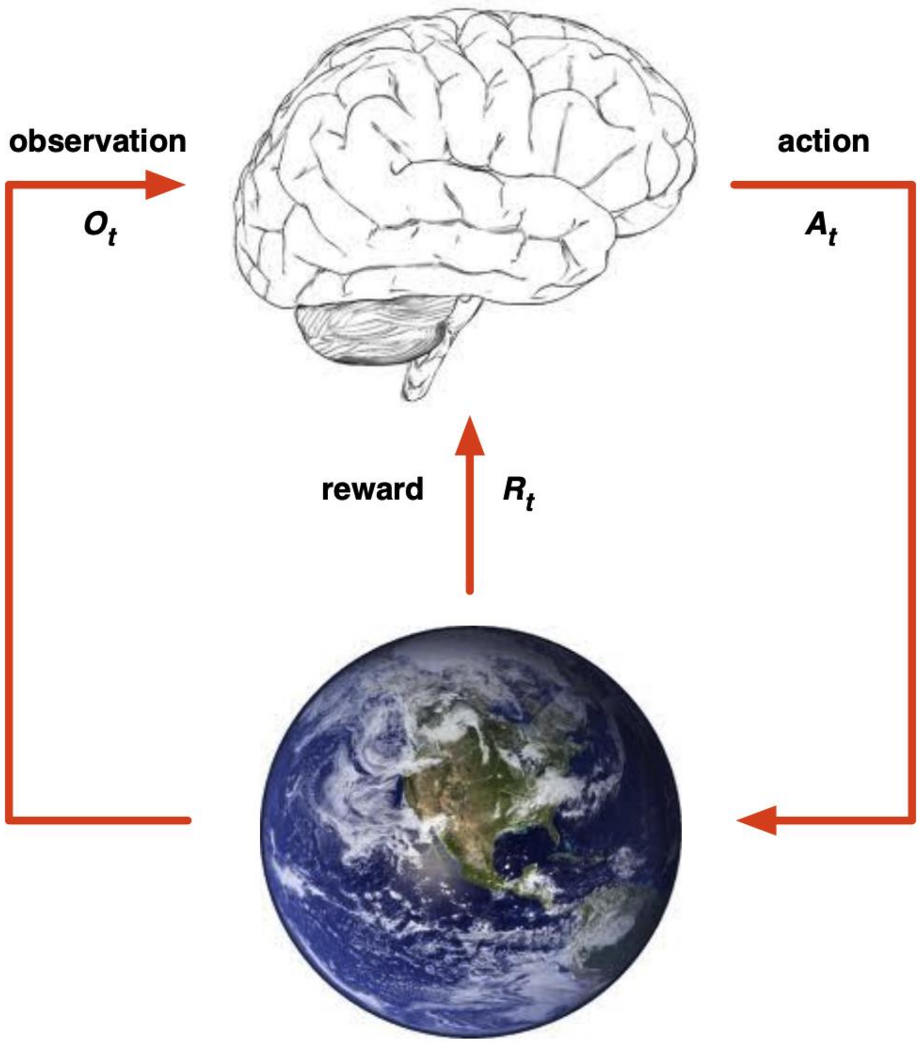
All of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward).

RL differs from supervised/unsupervised learning:

- Supervision is scarce, e.g., *reward* is often a scalar
- Supervision is often sparse, e.g., an agent gets the reward after a sequence of actions
- Sequential data is often non-iid, e.g., an agent's current decision would affect the future data distribution

Image Credit: [9]

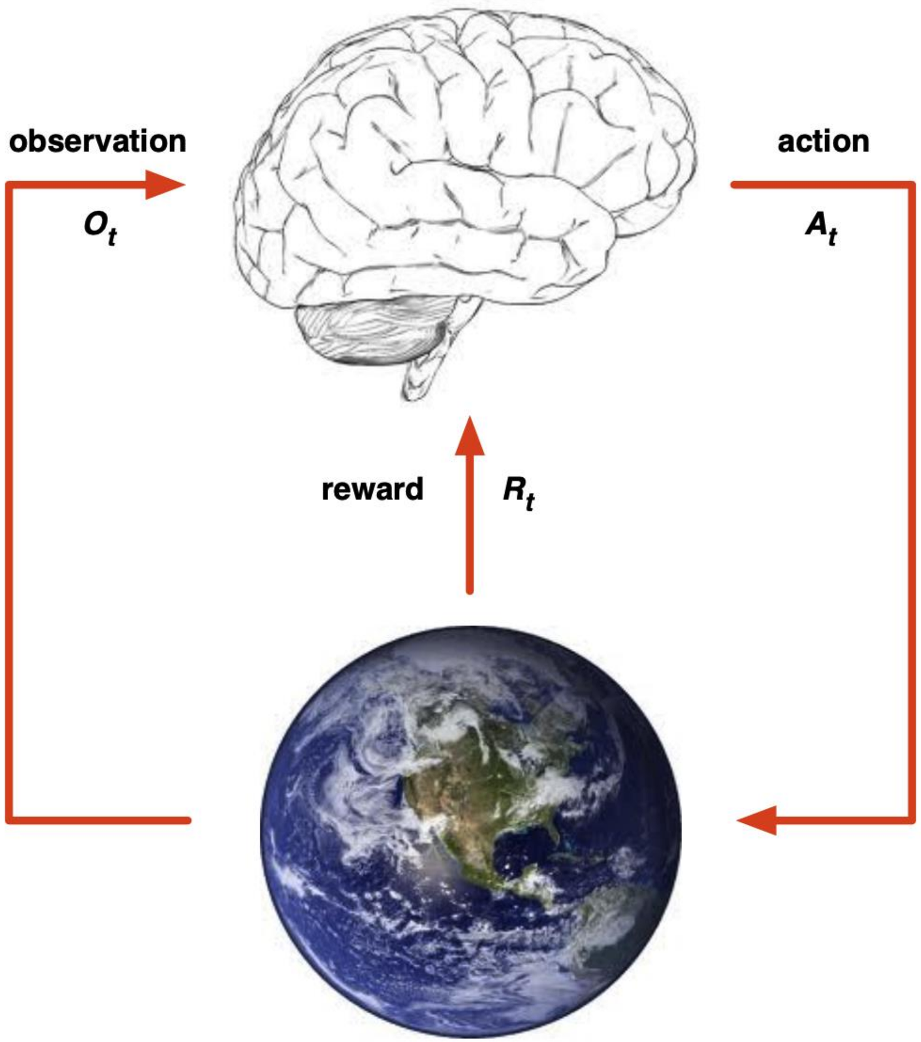
RL Formulation



What are the key components of the agent?

Image Credit: [9]

RL Formulation

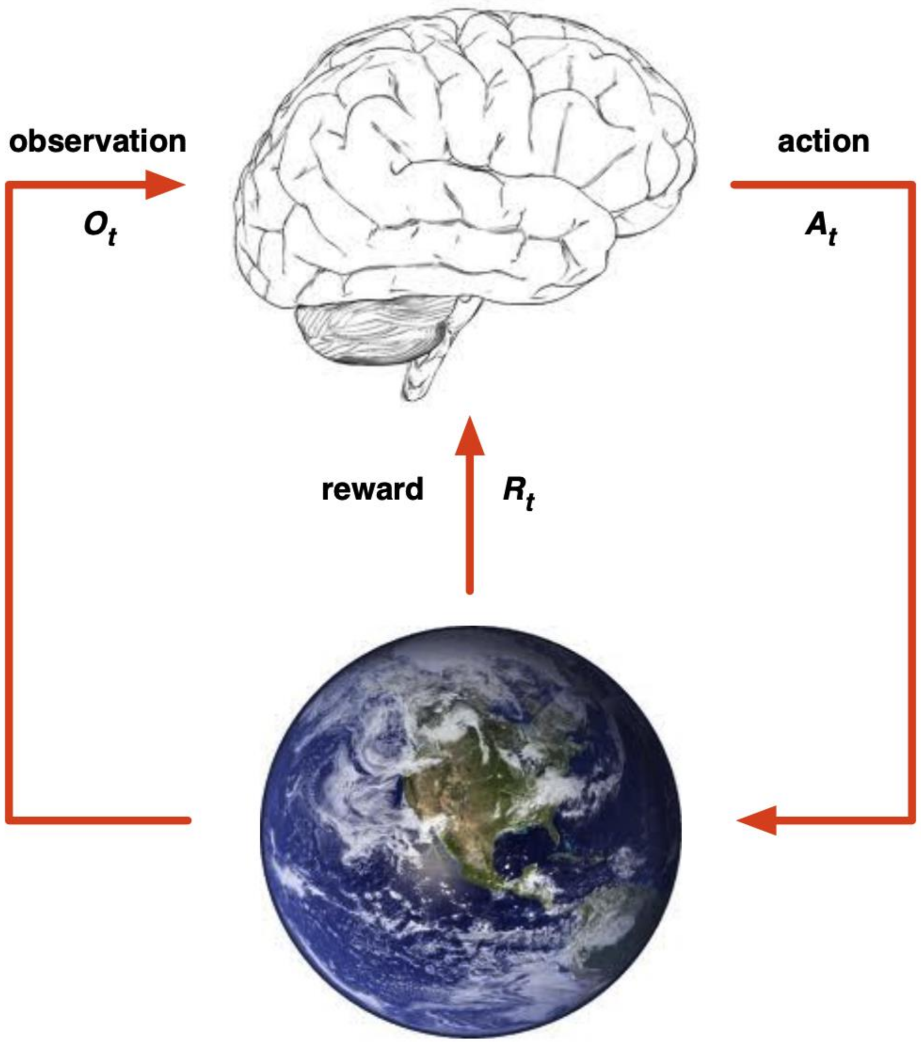


What are the key components of the agent?

Agent contains: Agent State, Policy, Value(?), and Model(?).

Image Credit: [9]

RL Formulation



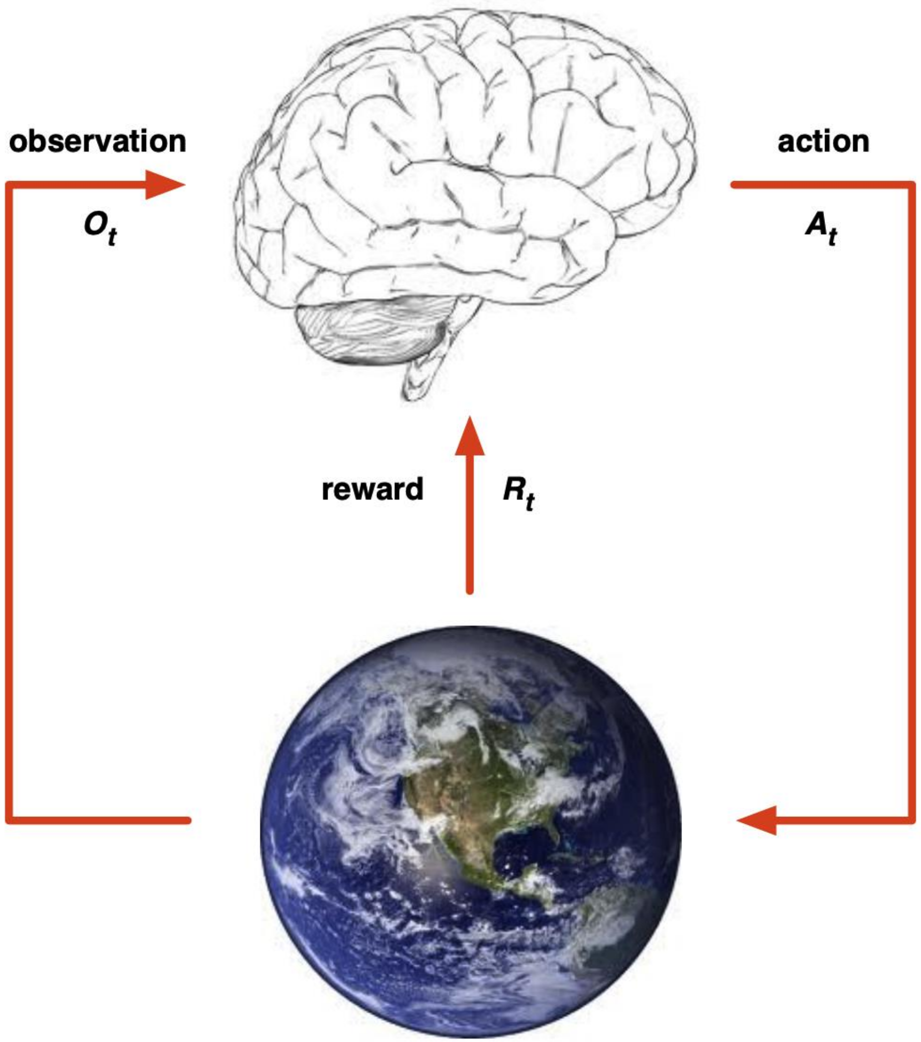
What are the key components of the agent?

Agent contains: **Agent State**, Policy, Value(?), and Model(?).

- The environment's (internal) state, is usually invisible (fully observable vs. partially observable) to the agent. Even if it is visible, it may contain lots of irrelevant information.

Image Credit: [9]

RL Formulation



What are the key components of the agent?

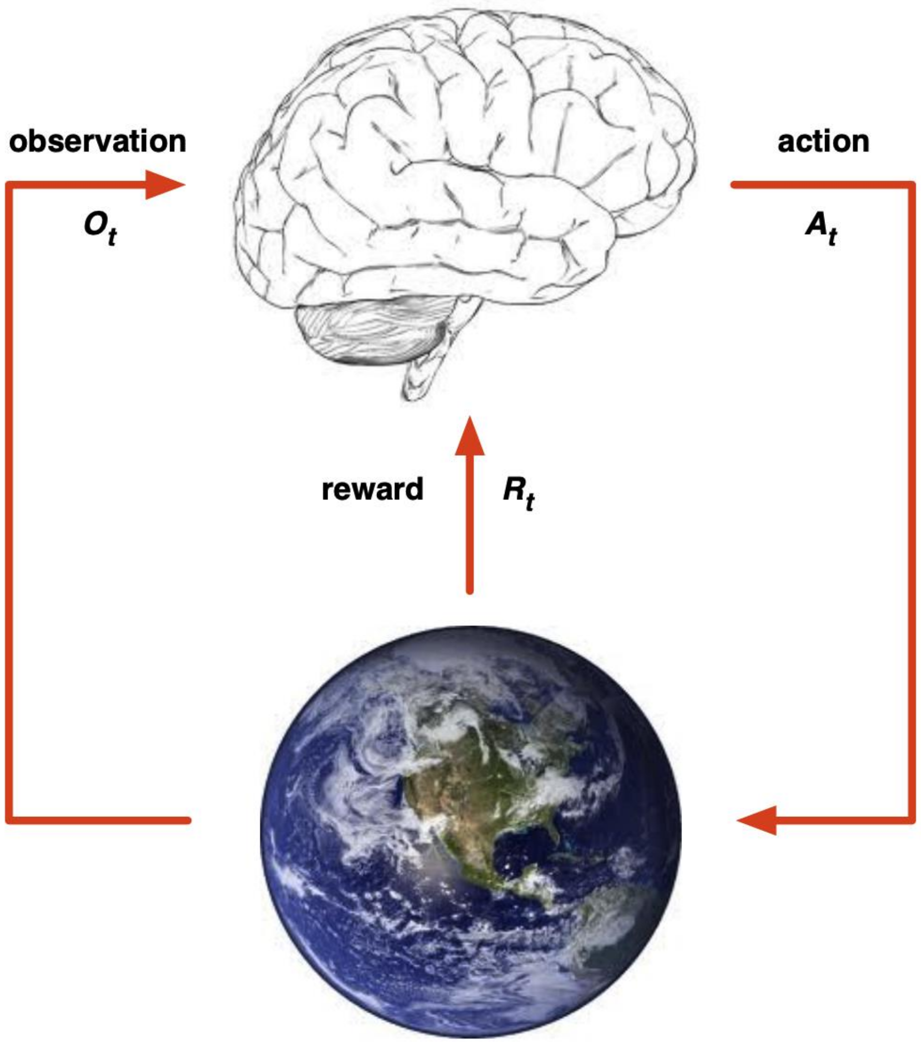
Agent contains: **Agent State**, Policy, Value(?), and Model(?).

- The environment's (internal) state, is usually invisible (fully observable vs. partially observable) to the agent. Even if it is visible, it may contain lots of irrelevant information.
- The **history** is the full sequence of observations, actions, and rewards up to time t:

$$H_t = O_1, A_1, R_1, \dots, O_{t-1}, A_{t-1}, R_{t-1}, O_t, R_t$$

Image Credit: [9]

RL Formulation



What are the key components of the agent?

Agent contains: **Agent State**, Policy, Value(?), and Model(?).

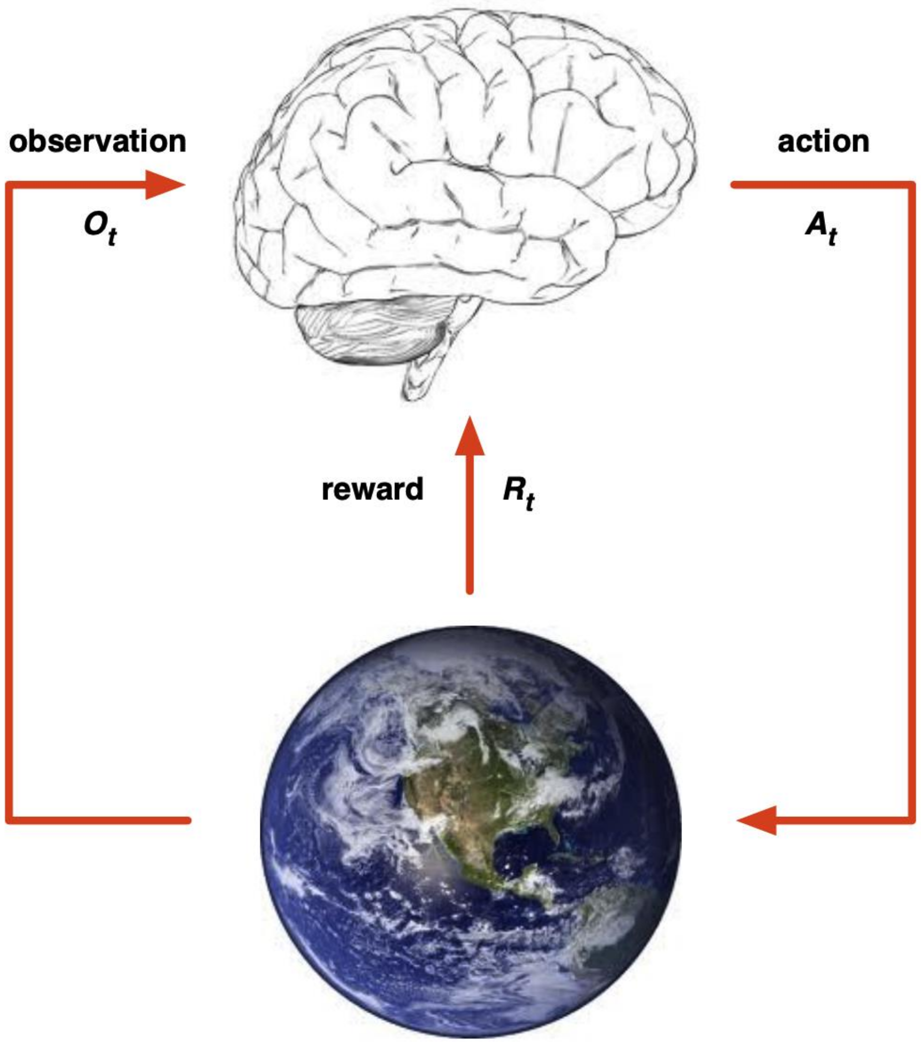
- The environment's (internal) state, is usually invisible (fully observable vs. partially observable) to the agent. Even if it is visible, it may contain lots of irrelevant information.
- The **history** is the full sequence of observations, actions, and rewards up to time t:

$$H_t = O_1, A_1, R_1, \dots, O_{t-1}, A_{t-1}, R_{t-1}, O_t, R_t$$

- The **agent state** is the agent's internal information/representation used to determine the next action. It is a function of the history $S_t = f(H_t)$.

Image Credit: [9]

RL Formulation



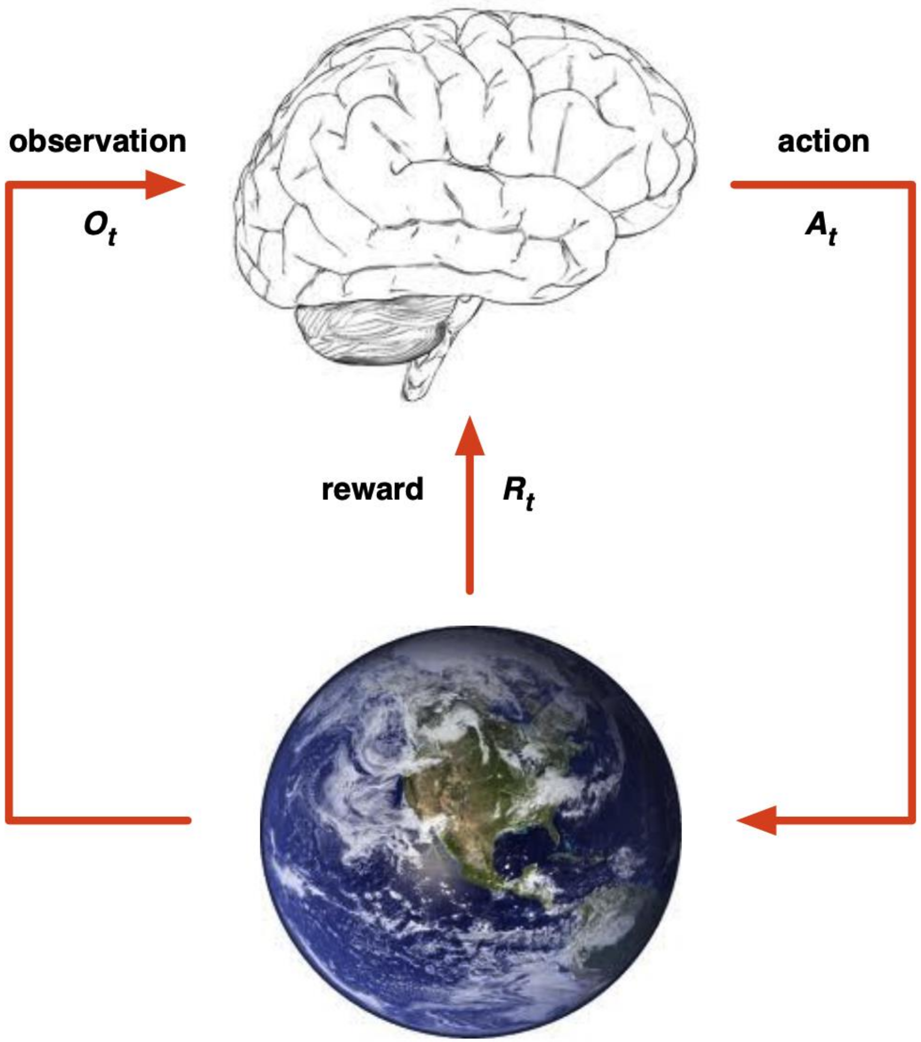
What are the key components of the agent?

Agent contains: Agent State, **Policy**, Value(?), and Model(?).

- A policy is a map from agent state to action that defines the agent's behavior

Image Credit: [9]

RL Formulation



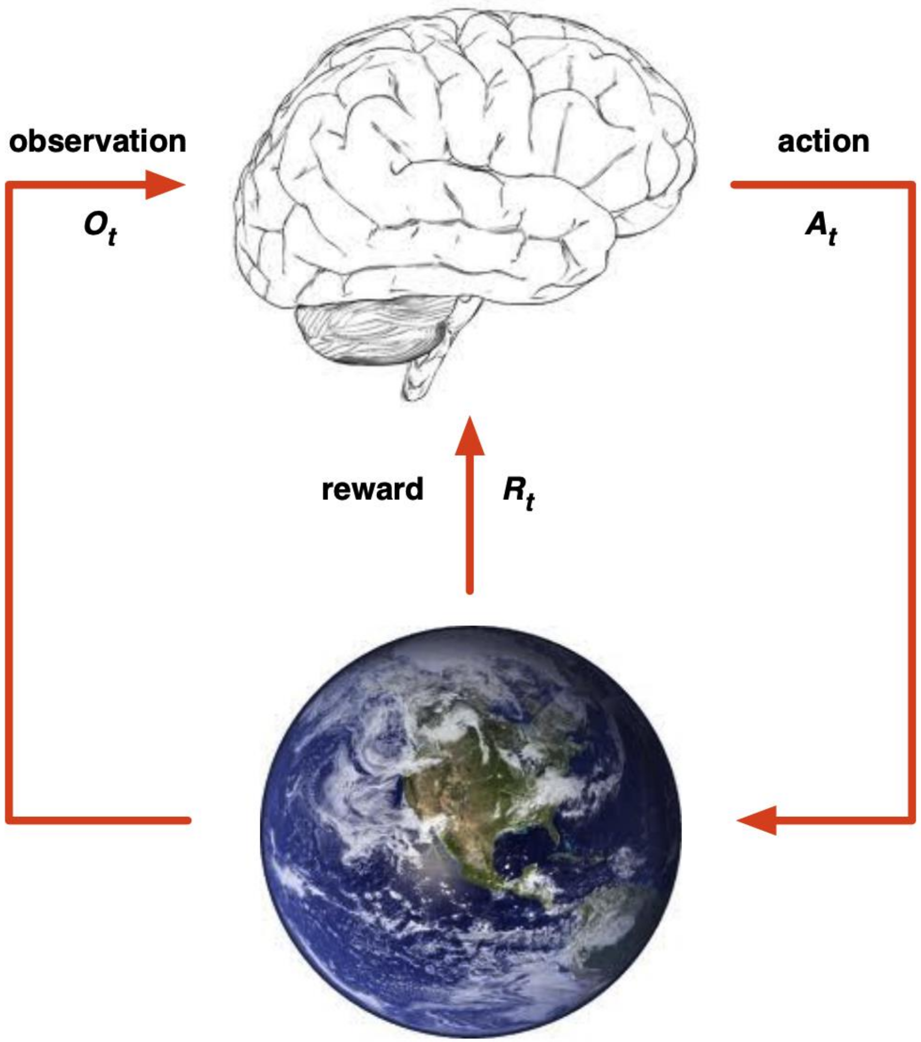
What are the key components of the agent?

Agent contains: Agent State, **Policy**, Value(?), and Model(?).

- A policy is a map from agent state to action that defines the agent's behavior
- It could be deterministic or stochastic. We can conveniently denote it as a probability distribution $\pi(a|s)$.

Image Credit: [9]

RL Formulation



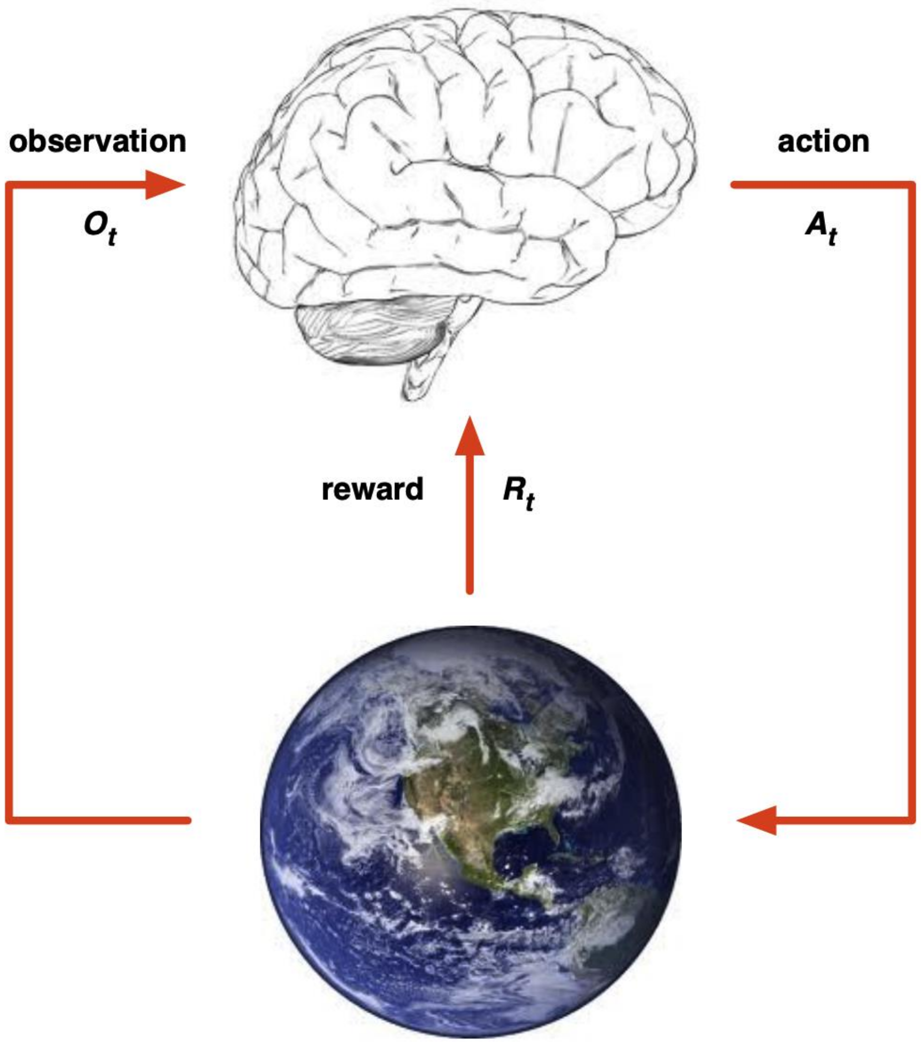
What are the key components of the agent?

Agent contains: Agent State, Policy, **Value**(?), and Model(?).

- A value function is a prediction of future reward

Image Credit: [9]

RL Formulation



What are the key components of the agent?

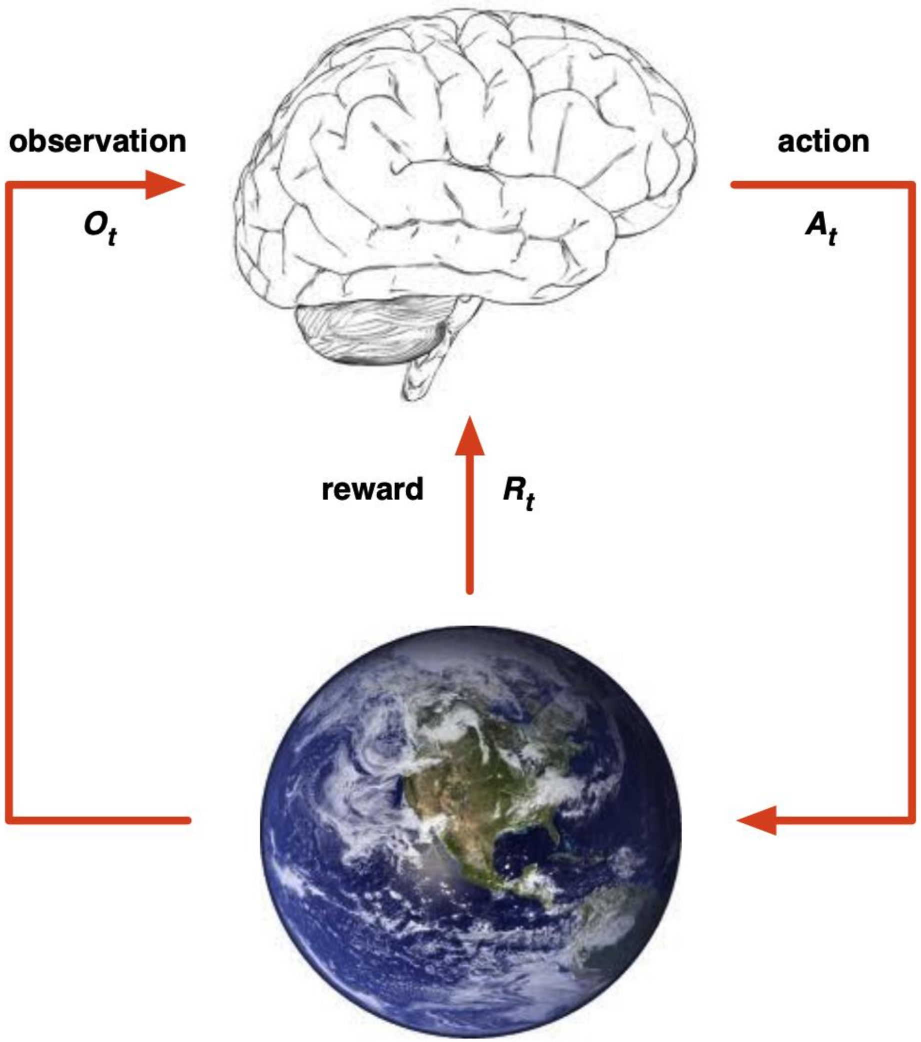
Agent contains: Agent State, Policy, **Value**(?), and Model(?).

- A value function is a prediction of future reward
- It is used to evaluate the goodness of states ($\gamma \in [0,1]$ is a discounting factor):

$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | S_t = s]$$

Image Credit: [9]

RL Formulation



What are the key components of the agent?

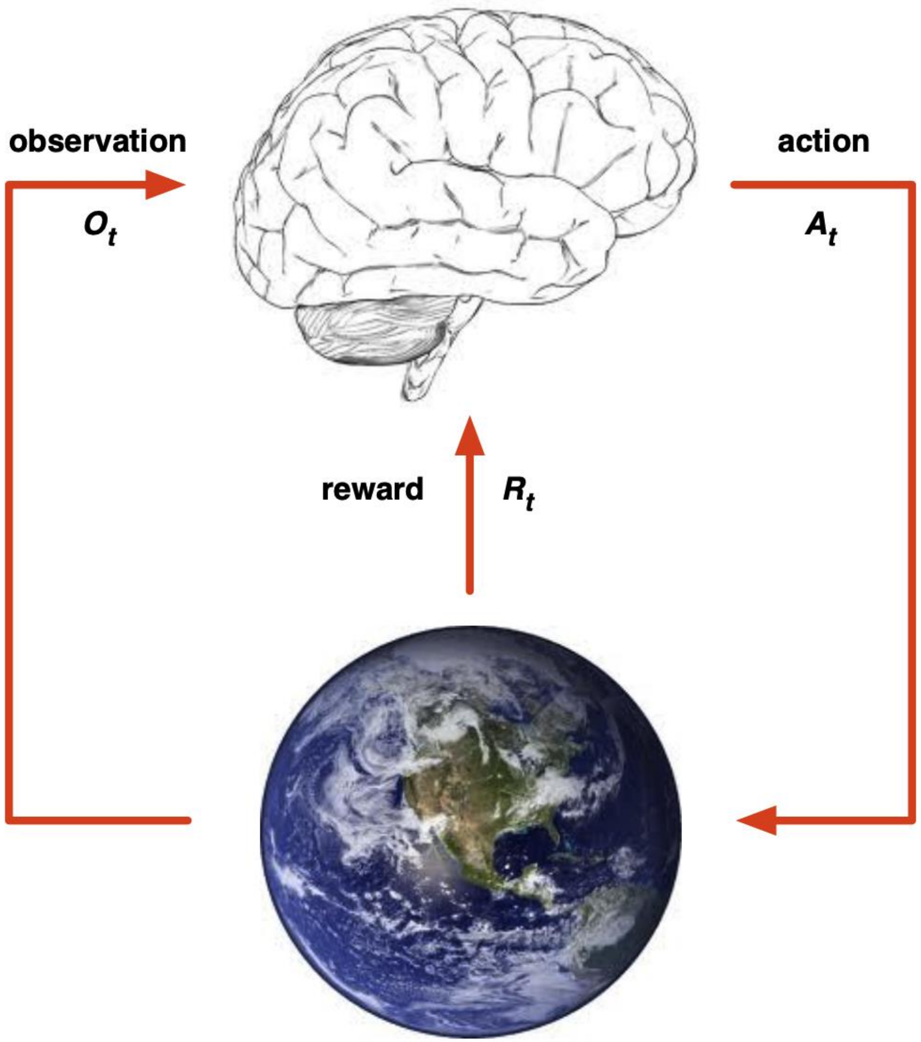
Agent contains: Agent State, Policy, Value(?), and **Model**(?).

- A model predicts what the environment will do next, e.g., $\mathcal{R}$ predicts the next immediate reward

$$\mathcal{R}(s, a) \approx \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$$

Image Credit: [9]

RL Formulation



What are the key components of the agent?

Agent contains: Agent State, Policy, Value(?), and **Model**(?).

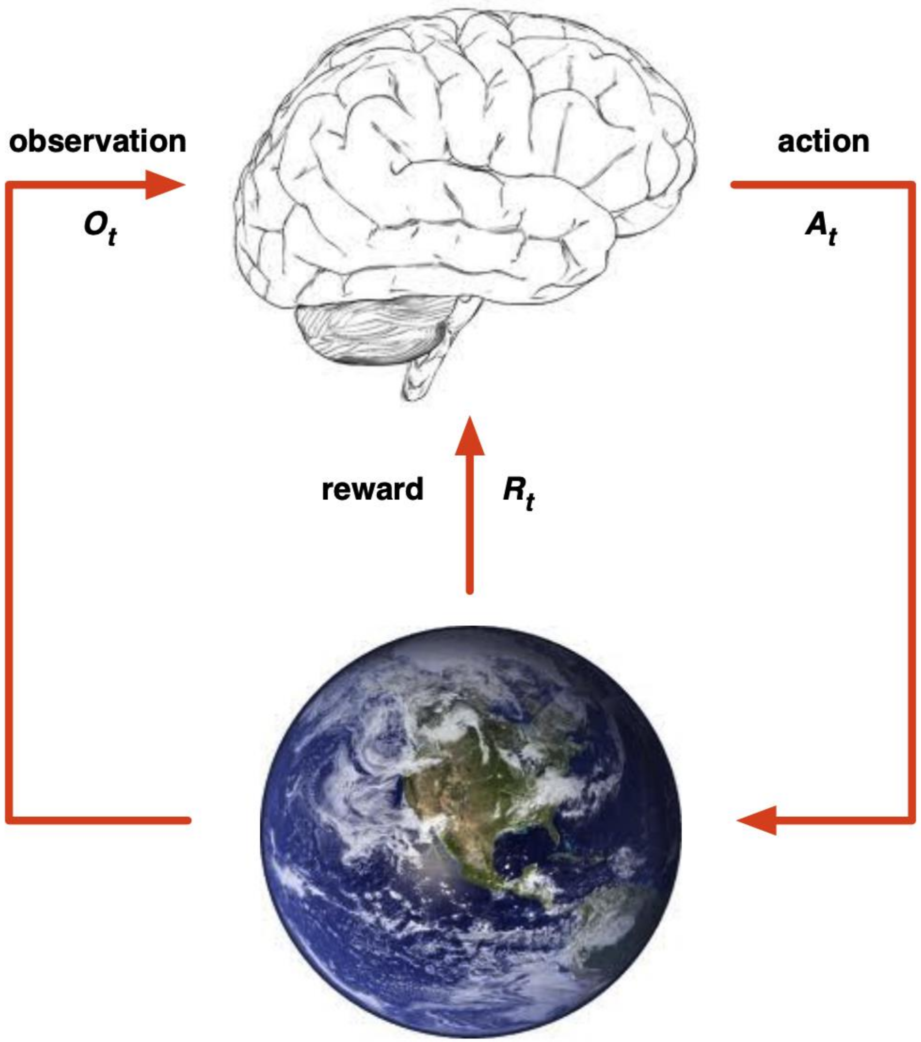
- A model predicts what the environment will do next, e.g., $\mathcal{R}$ predicts the next immediate reward

$$\mathcal{R}(s, a) \approx \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$$

- A model does not immediately give us a good policy - we would still need to plan

Image Credit: [9]

RL Formulation



What are the key components of the agent?

Agent contains: Agent State, Policy, Value(?), and **Model**(?).

- A model predicts what the environment will do next, e.g., $\mathcal{R}$ predicts the next immediate reward

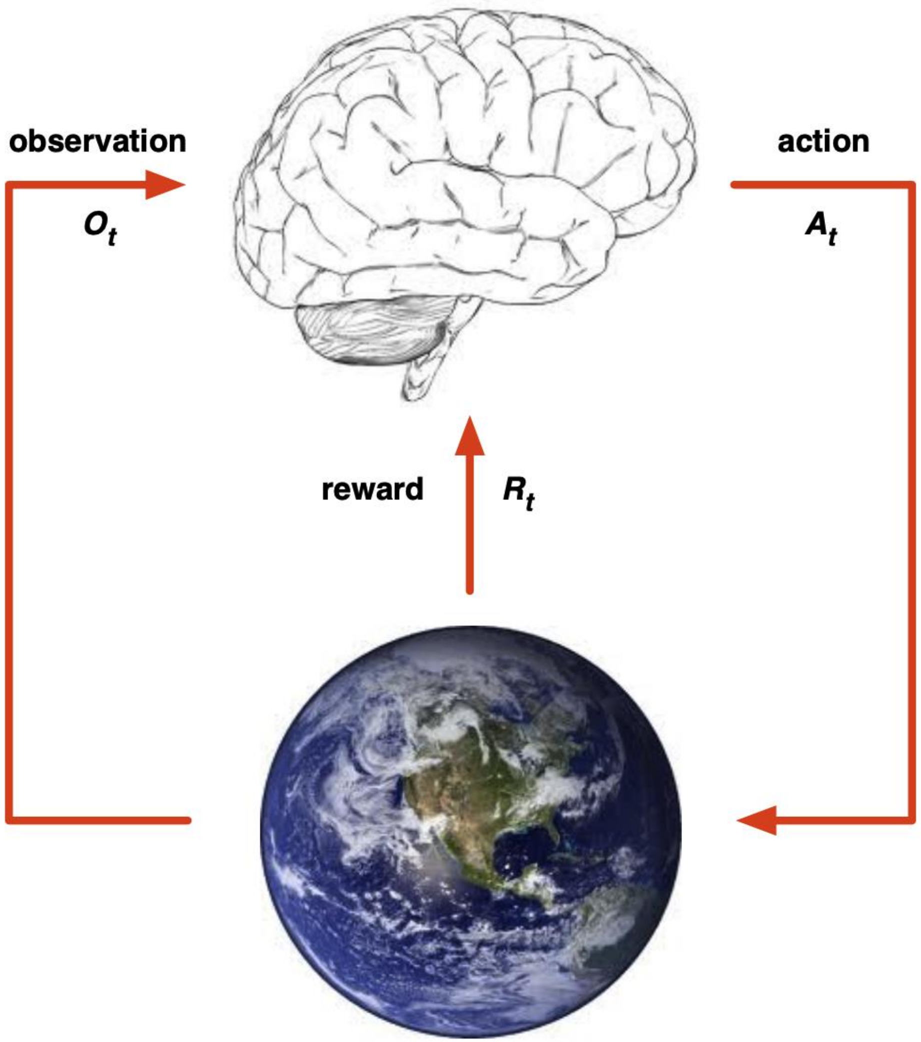
$$\mathcal{R}(s, a) \approx \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$$

- A model does not immediately give us a good policy - we would still need to plan

- We could also use stochastic (generative) models

Image Credit: [9]

RL Formulation



What are the key components of the agent?

Agent contains: Agent State, Policy, Value(?), and Model(?).

Agent Taxonomy:

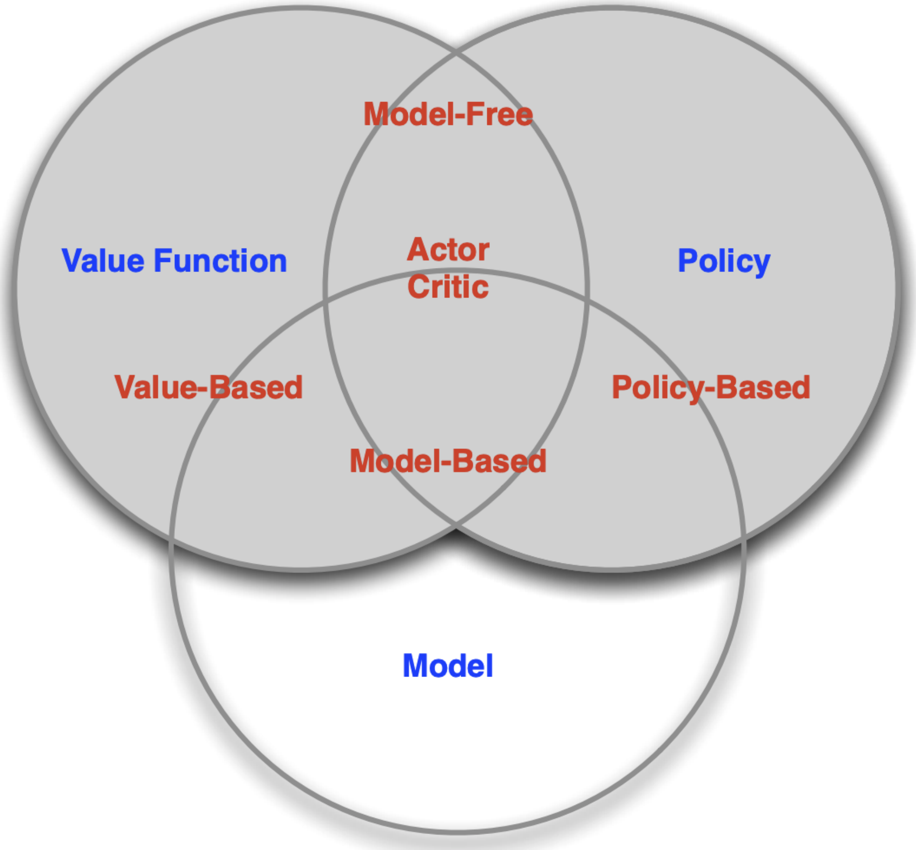


Image Credit: [9]

Learning and Planning

Two fundamental problems in sequential decision making:

1. **Reinforcement Learning**:
    - The environment is initially unknown
    - The agent interacts with the environment
    - The agent improves its policy

Learning and Planning

Two fundamental problems in sequential decision making:

1. **Reinforcement Learning**:
    - The environment is initially unknown
    - The agent interacts with the environment
    - The agent improves its policy
2. **Planning**:
    - A model of the environment is known
    - The agent performs computations with its model (without any external interaction)
    - The agent improves its policy
    - a.k.a. deliberation, reasoning, introspection, pondering, thought, search

RL Taxonomy

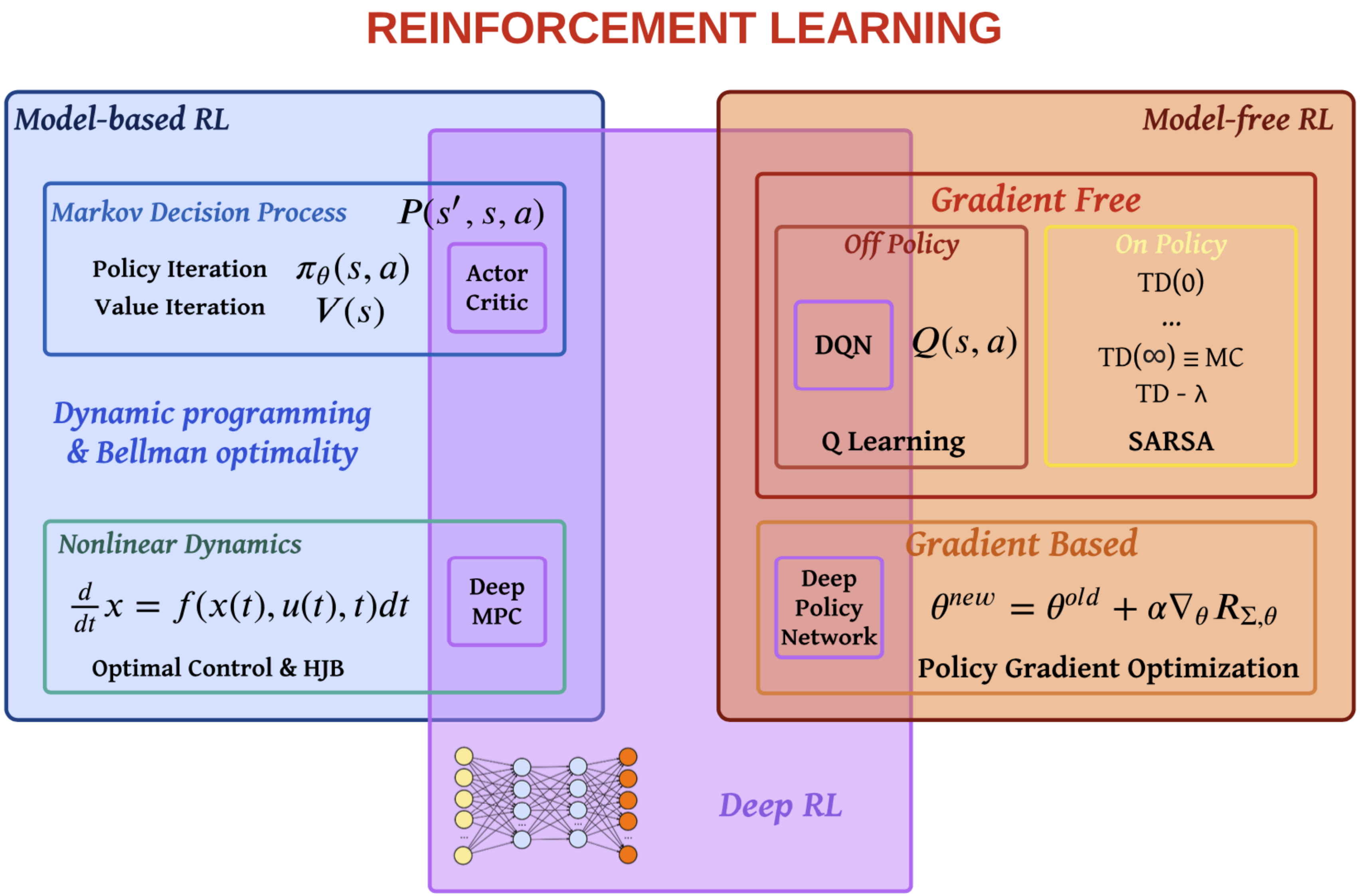


Image Credit: [10]

RL Taxonomy

Category	Subcategory	Examples	Does It Learn a Model?
Model-Free RL	**Value-Based**	Q-Learning, SARSA, DQN, TD-Learning	❌ No
	Policy-Based	REINFORCE, PPO, TRPO, SAC	❌ No
	Actor-Critic	A2C, A3C, DDPG, TD3	❌ No
Model-Based RL	**Explicit Model Learning**	Dyna-Q, AlphaZero, World Models	✅ Yes
	Planning Methods	Monte Carlo Tree Search (MCTS), Model-Predictive Control (MPC)	✅ Yes

Outline

- Reinforcement Learning (RL) I
    - Overview & Applications
    - RL Formulation & Taxonomy
    - **Markov Decision Process (MDP)**
    - Bellman Equations
    - Solving RL problems using the Bellman Equations
- Reinforcement Learning (RL) II
    - Model-Free RL: (Deep) Q-learning
    - Model-Free RL: Policy Gradient & Actor-Critic
    - Model-Based RL: Dyna-Q

Markov Decision Process

Almost all RL problems can be formalized as Markov decision processes (MDPs).

Markov Decision Process

Almost all RL problems can be formalized as Markov decision processes (MDPs).

A Markov decision process (MDP) is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix $\mathcal{P}^a_{ss'} = \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a)$
- $\mathcal{R}$ is a reward function $\mathcal{R}^a_s = \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$
- $\gamma \in [0, 1]$ is a discount factor

Markov Decision Process

Almost all RL problems can be formalized as Markov decision processes (MDPs).

A Markov decision process (MDP) is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

Markov Property: The future is independent of the past given the present!

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix $\quad \mathcal{P}^a_{ss'} = \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a)$
- $\mathcal{R}$ is a reward function $\quad \mathcal{R}^a_s = \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$
- $\gamma \in [0, 1]$ is a discount factor

Markov Decision Process

Almost all RL problems can be formalized as Markov decision processes (MDPs).

A Markov decision process (MDP) is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

Markov Property: The future is independent of the past given the present!

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix $\mathcal{P}^a_{ss'} = \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a)$
- $\mathcal{R}$ is a reward function $\mathcal{R}^a_s = \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$
- $\gamma \in [0, 1]$ is a discount factor

MDP describes an environment where all states are Markov and can be extended to:

- countably infinite states and or action spaces
- continuous state and or action spaces
- continuous time (requires partial differentiable equations)
- partially observable (POMDPs)

Markov Decision Process

Return: the total discounted reward from time t

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

Markov Decision Process

Return: the total discounted reward from time t

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

Why discount? Mathematically convenient, avoid infinite returns, uncertainty about the future……

Markov Decision Process

Return: the total discounted reward from time t

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

Policy: the distribution over actions given states

$$\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$$

Why discount? Mathematically convenient, avoid infinite returns, uncertainty about the future……

Markov Decision Process

Return: the total discounted reward from time t

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

Why discount? Mathematically convenient, avoid infinite returns, uncertainty about the future……

Policy: the distribution over actions given states

$$\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$$

We assume *stationary* policies

Markov Decision Process

Return: the total discounted reward from time t

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

Why discount? Mathematically convenient, avoid infinite returns, uncertainty about the future……

Policy: the distribution over actions given states

$$\pi(a|s) = \mathbb{P}(A_t = a|S_t = s)$$

We assume *stationary* policies

Value (a.k.a., State-Value) function: the expected return starting from state s and then following policy π

$$V_\pi(s) = \mathbb{E}_\pi \left[G_t|S_t = s\right]$$

Markov Decision Process

Return: the total discounted reward from time t

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

Why discount? Mathematically convenient, avoid infinite returns, uncertainty about the future……

Policy: the distribution over actions given states

$$\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$$

We assume *stationary* policies

Value (a.k.a., State-Value) function: the expected return starting from state s and then following policy π

$$V_\pi(s) = \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Optimal value function

$$V_*(s) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Markov Decision Process

Return: the total discounted reward from time t

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

Why discount? Mathematically convenient, avoid infinite returns, uncertainty about the future……

Policy: the distribution over actions given states

$$\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$$

We assume *stationary* policies

Value (a.k.a., State-Value) function: the expected return starting from state s and then following policy π

$$V_\pi(s) = \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Optimal value function

$$V_*(s) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Q (a.k.a. Action-Value) function: the expected return starting from state s, taking an action a, and then following policy π

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$$

Markov Decision Process

Return: the total discounted reward from time t

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

Why discount? Mathematically convenient, avoid infinite returns, uncertainty about the future……

Policy: the distribution over actions given states

$$\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$$

We assume *stationary* policies

Value (a.k.a., State-Value) function: the expected return starting from state s and then following policy π

$$V_\pi(s) = \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Optimal value function

$$V_*(s) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Q (a.k.a. Action-Value) function: the expected return starting from state s, taking an action a, and then following policy π

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$$

Optimal Q function

$$Q_*(s, a) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$$

Markov Decision Process

Return: the total discounted reward from time t

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

Why discount? Mathematically convenient, avoid infinite returns, uncertainty about the future……

Policy: the distribution over actions given states

$$\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$$

We assume *stationary* policies

Value (a.k.a., State-Value) function: the expected return starting from state s and then following policy π

$$V_\pi(s) = \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Optimal value function

$$V_*(s) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Q (a.k.a. Action-Value) function: the expected return starting from state s, taking an action a, and then following policy π

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$$

Optimal Q function

$$Q_*(s, a) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$$

What is the relationship between the value function and Q function?

Markov Decision Process

Return: the total discounted reward from time t

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

Why discount? Mathematically convenient, avoid infinite returns, uncertainty about the future……

Policy: the distribution over actions given states

$$\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$$

We assume *stationary* policies

Value (a.k.a., State-Value) function: the expected return starting from state s and then following policy π

$$V_\pi(s) = \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Optimal value function

$$V_*(s) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Q (a.k.a. Action-Value) function: the expected return starting from state s, taking an action a, and then following policy π

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$$

Optimal Q function

$$Q_*(s, a) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$$

What is the relationship between the value function and Q function?

$$V_\pi(s) = \sum_a Q_\pi(s, a) \pi(a|s)$$

Optimal Policy

Optimal value function $$V_*(s) = \max_{\pi} \mathbb{E}_\pi \left[G_t | S_t = s\right]$$

Optimal Q function $$Q_*(s, a) = \max_{\pi} \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a\right]$$

One can define a partial ordering over policies $$\pi \geq \pi' \iff V_\pi(s) \geq V_{\pi'}(s), \forall s$$

Theorem (Optimal Policies)

For any Markov decision process

- *There exists an optimal policy* π^* *that is better than or equal to all other policies,* $\pi^* \geq \pi, \forall \pi$
  (There can be more than one such optimal policy.)
- *All optimal policies achieve the optimal value function,* $v^{\pi^*}(s) = v^*(s)$
- *All optimal policies achieve the optimal action-value function,* $q^{\pi^*}(s, a) = q^*(s, a)$

Image Credit: [12]

Optimal Policy

Optimal value function $$V_*(s) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Optimal Q function $$Q_*(s,a) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$$

If we know $Q_*(s,a)$, an optimal policy can be found:

$$\pi_*(a|s) = \begin{cases} 1 & a = a_*(s) = \underset{a}{\operatorname{argmax}} \ Q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Optimal Policy

Optimal value function $$V_*(s) = \max_{\pi} \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Optimal Q function $$Q_*(s,a) = \max_{\pi} \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$$

If we know $Q_*(s,a)$, an optimal policy can be found:

$$\pi_*(a|s) = \begin{cases} 1 & a = a_*(s) = \underset{a}{\operatorname{argmax}} \; Q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Why?

Optimal Policy

Optimal value function $$V_*(s) = \max_\pi \mathbb{E}_\pi\left[G_t | S_t = s\right]$$

Optimal Q function $$Q_*(s,a) = \max_\pi \mathbb{E}_\pi\left[G_t | S_t = s, A_t = a\right]$$

If we know $Q_*(s,a)$, an optimal policy can be found:

$$\pi_*(a|s) = \begin{cases} 1 & a = a_*(s) = \underset{a}{\operatorname{argmax}}\; Q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Why?

Because $V_\pi(s) = \sum_a Q_\pi(s,a)\pi(a|s)$ is a convex combination of $Q_\pi(s,a)$, we have

$$V_\pi(s) \leq Q_\pi(s, a_*), \qquad a_* = \underset{a}{\operatorname{argmax}}\; Q_*(s,a)$$

and the equality holds only if the policy is optimal.

Optimal Policy

Optimal value function $$V_*(s) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Optimal Q function $$Q_*(s,a) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$$

If we know $Q_*(s,a)$, an optimal policy can be found:

$$\pi_*(a|s) = \begin{cases} 1 & a = a_*(s) = \underset{a}{\operatorname{argmax}} \ Q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

- There is always a deterministic optimal policy for any MDP.
- There can be multiple actions that maximize $Q_*(s,a)$, we can just pick any of these.

Outline

- Reinforcement Learning (RL) I
  - Overview & Applications
  - RL Formulation & Taxonomy
  - Markov Decision Process (MDP)
  - **Bellman Equations**
  - Solving RL problems using the Bellman Equations
- Reinforcement Learning (RL) II
  - Model-Free RL: (Deep) Q-learning
  - Model-Free RL: Policy Gradient & Actor-Critic
  - Model-Based RL: Dyna-Q

Bellman Equations

Many RL algorithms are based on *Bellman Equations*, which are recursive formulas and have two main variations: *Bellman Expectation Equations* and *Bellman Optimality Equations*.

Bellman Expectation Equations

For Q-function, we have:

$$Q_\pi(s,a) = \mathbb{E}_\pi\left[G_t | S_t = s, A_t = a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\right]$$

Bellman Expectation Equations

For Q-function, we have:

$$
\begin{aligned}
Q_\pi(s,a) &= \mathbb{E}_\pi\left[G_t|S_t=s, A_t=a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t=s, A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \sum_{k=1}^{\infty}\gamma^k R_{t+k+1}|S_t=s, A_t=a\right]
\end{aligned}
$$

Bellman Expectation Equations

For Q-function, we have:

$$
\begin{aligned}
Q_\pi(s,a) &= \mathbb{E}_\pi\left[G_t|S_t=s,A_t=a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t=s,A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1}+\sum_{k=1}^{\infty}\gamma^k R_{t+k+1}|S_t=s,A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1}|S_t=s,A_t=a\right] + \gamma\mathbb{E}_\pi\left[\sum_{k=1}^{\infty}\gamma^{k-1} R_{t+k+1}|S_t=s,A_t=a\right]
\end{aligned}
$$

Bellman Expectation Equations

For Q-function, we have:

$$
\begin{aligned}
Q_\pi(s,a) &= \mathbb{E}_\pi\left[G_t | S_t = s, A_t = a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \sum_{k=1}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} | S_t = s, A_t = a\right] + \gamma \mathbb{E}_\pi\left[\sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k+1} | S_t = s, A_t = a\right] \\
&= \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right] + \\
&\quad \gamma \sum_{S_{t+1}, A_{t+1}, R_{t+1}, \cdots} \left(\sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k+1}\right) \mathbb{P}(S_{t+1}, A_{t+1}, R_{t+1}, \cdots | S_t = s, A_t = a)
\end{aligned}
$$

Bellman Expectation Equations

For Q-function, we have:

$$
\begin{aligned}
Q_\pi(s,a) &= \mathbb{E}_\pi\left[G_t | S_t = s, A_t = a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} | S_t = s, A_t = a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \sum_{k=1}^{\infty}\gamma^k R_{t+k+1} | S_t = s, A_t = a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} | S_t = s, A_t = a\right] + \gamma \mathbb{E}_\pi\left[\sum_{k=1}^{\infty}\gamma^{k-1} R_{t+k+1} | S_t = s, A_t = a\right] \\
&= \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right] + \\
&\quad \gamma \sum_{S_{t+1}, A_{t+1}, R_{t+1}, \cdots}\left(\sum_{k=1}^{\infty}\gamma^{k-1} R_{t+k+1}\right)\mathbb{P}(S_{t+1}, A_{t+1}, R_{t+1}, \cdots | S_t = s, A_t = a)
\end{aligned}
$$

$$\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$$

Bellman Expectation Equations

For Q-function, we have:

$$
\begin{aligned}
Q_\pi(s,a) &= \mathbb{E}_\pi\left[G_t|S_t=s, A_t=a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t=s, A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \sum_{k=1}^{\infty}\gamma^k R_{t+k+1}|S_t=s, A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1}|S_t=s, A_t=a\right] + \gamma\mathbb{E}_\pi\left[\sum_{k=1}^{\infty}\gamma^{k-1} R_{t+k+1}|S_t=s, A_t=a\right] \\
&= \mathbb{E}\left[R_{t+1}|S_t=s, A_t=a\right] + \\
&\quad \gamma \sum_{S_{t+1},A_{t+1},R_{t+1},\cdots}\left(\sum_{k=1}^{\infty}\gamma^{k-1}R_{t+k+1}\right)\mathbb{P}(S_{t+1},A_{t+1},R_{t+1},\cdots|S_t=s,A_t=a) \\
&= \mathcal{R}_s^a + \gamma \sum_{S_{t+1},A_{t+1},R_{t+1},\cdots}\left(\sum_{k=1}^{\infty}\gamma^{k-1}R_{t+k+1}\right)\mathbb{P}(S_{t+2},A_{t+2},R_{t+2},\cdots|S_{t+1}=s',A_{t+1}=a') \\
&\quad \mathbb{P}(A_{t+1}=a'|S_{t+1}=s')\mathbb{P}(S_{t+1}=s'|S_t=s,A_t=a)
\end{aligned}
$$

$$\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1}|S_t=s, A_t=a\right]$$

Bellman Expectation Equations

For Q-function, we have:

$$
\begin{aligned}
Q_\pi(s,a) &= \mathbb{E}_\pi\left[G_t|S_t=s,A_t=a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t=s,A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1}+\sum_{k=1}^{\infty}\gamma^k R_{t+k+1}|S_t=s,A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1}|S_t=s,A_t=a\right] + \gamma\mathbb{E}_\pi\left[\sum_{k=1}^{\infty}\gamma^{k-1} R_{t+k+1}|S_t=s,A_t=a\right] \\
&= \mathbb{E}\left[R_{t+1}|S_t=s,A_t=a\right] + \\
&\quad \gamma \sum_{S_{t+1},A_{t+1},R_{t+1},\cdots}\left(\sum_{k=1}^{\infty}\gamma^{k-1}R_{t+k+1}\right)\mathbb{P}(S_{t+1},A_{t+1},R_{t+1},\cdots|S_t=s,A_t=a) \\
&= \mathcal{R}_s^a + \gamma \sum_{S_{t+1},A_{t+1},R_{t+1},\cdots}\left(\sum_{k=1}^{\infty}\gamma^{k-1}R_{t+k+1}\right)\mathbb{P}(S_{t+2},A_{t+2},R_{t+2},\cdots|S_{t+1}=s',A_{t+1}=a') \\
&\quad \mathbb{P}(A_{t+1}=a'|S_{t+1}=s')\mathbb{P}(S_{t+1}=s'|S_t=s,A_t=a)
\end{aligned}
$$

$$\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1}|S_t=s,A_t=a\right]$$

$$\pi(a|s) = \mathbb{P}(A_t=a|S_t=s)$$

$$\mathcal{P}_{ss'}^a = \mathbb{P}(S_{t+1}=s'|S_t=s,A_t=a)$$

Bellman Expectation Equations

For Q-function, we have:

$$
\begin{aligned}
Q_\pi(s,a) &= \mathbb{E}_\pi\left[G_t|S_t=s,A_t=a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t=s,A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1}+\sum_{k=1}^{\infty}\gamma^k R_{t+k+1}|S_t=s,A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1}|S_t=s,A_t=a\right]+\gamma\mathbb{E}_\pi\left[\sum_{k=1}^{\infty}\gamma^{k-1} R_{t+k+1}|S_t=s,A_t=a\right] \\
&= \mathbb{E}\left[R_{t+1}|S_t=s,A_t=a\right]+ \\
&\quad \gamma\sum_{S_{t+1},A_{t+1},R_{t+1},\cdots}\left(\sum_{k=1}^{\infty}\gamma^{k-1}R_{t+k+1}\right)\mathbb{P}(S_{t+1},A_{t+1},R_{t+1},\cdots|S_t=s,A_t=a) \\
&= \mathcal{R}_s^a+\gamma\sum_{S_{t+1},A_{t+1},R_{t+1},\cdots}\left(\sum_{k=1}^{\infty}\gamma^{k-1}R_{t+k+1}\right)\mathbb{P}(S_{t+2},A_{t+2},R_{t+2},\cdots|S_{t+1}=s',A_{t+1}=a') \\
&\quad \mathbb{P}(A_{t+1}=a'|S_{t+1}=s')\mathbb{P}(S_{t+1}=s'|S_t=s,A_t=a) \\
&= \mathcal{R}_s^a+\gamma\sum_{s',a'}\pi(a'|s')\mathcal{P}_{ss'}^a\mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+2}|S_{t+1}=s',A_{t+1}=a'\right]
\end{aligned}
$$

$$\mathcal{R}_s^a=\mathbb{E}\left[R_{t+1}|S_t=s,A_t=a\right]$$

$$\pi(a|s)=\mathbb{P}(A_t=a|S_t=s)$$

$$\mathcal{P}_{ss'}^a=\mathbb{P}(S_{t+1}=s'|S_t=s,A_t=a)$$

Bellman Expectation Equations

For Q-function, we have:

$$
\begin{aligned}
Q_\pi(s,a) &= \mathbb{E}_\pi\left[G_t|S_t=s,A_t=a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t=s,A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1}+\sum_{k=1}^{\infty}\gamma^k R_{t+k+1}|S_t=s,A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1}|S_t=s,A_t=a\right]+\gamma\mathbb{E}_\pi\left[\sum_{k=1}^{\infty}\gamma^{k-1} R_{t+k+1}|S_t=s,A_t=a\right] \\
&= \mathbb{E}\left[R_{t+1}|S_t=s,A_t=a\right]+ \\
&\quad \gamma\sum_{S_{t+1},A_{t+1},R_{t+1},\cdots}\left(\sum_{k=1}^{\infty}\gamma^{k-1}R_{t+k+1}\right)\mathbb{P}(S_{t+1},A_{t+1},R_{t+1},\cdots|S_t=s,A_t=a) \\
&= \mathcal{R}_s^a+\gamma\sum_{S_{t+1},A_{t+1},R_{t+1},\cdots}\left(\sum_{k=1}^{\infty}\gamma^{k-1}R_{t+k+1}\right)\mathbb{P}(S_{t+2},A_{t+2},R_{t+2},\cdots|S_{t+1}=s',A_{t+1}=a') \\
&\quad \mathbb{P}(A_{t+1}=a'|S_{t+1}=s')\mathbb{P}(S_{t+1}=s'|S_t=s,A_t=a) \\
&= \mathcal{R}_s^a+\gamma\sum_{s',a'}\pi(a'|s')\mathcal{P}_{ss'}^a\mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+2}|S_{t+1}=s',A_{t+1}=a'\right] \\
&= \mathcal{R}_s^a+\gamma\sum_{s',a'}\pi(a'|s')\mathcal{P}_{ss'}^a Q_\pi(s',a')
\end{aligned}
$$

$$\mathcal{R}_s^a=\mathbb{E}\left[R_{t+1}|S_t=s,A_t=a\right]$$

$$\pi(a|s)=\mathbb{P}(A_t=a|S_t=s)$$

$$\mathcal{P}_{ss'}^a=\mathbb{P}(S_{t+1}=s'|S_t=s,A_t=a)$$

Bellman Expectation Equations

For Q-function, we have:

$$
\begin{aligned}
Q_\pi(s,a) &= \mathbb{E}_\pi\left[G_t|S_t=s, A_t=a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t=s, A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \sum_{k=1}^{\infty}\gamma^k R_{t+k+1}|S_t=s, A_t=a\right] \\
&= \mathbb{E}_\pi\left[R_{t+1}|S_t=s, A_t=a\right] + \gamma\mathbb{E}_\pi\left[\sum_{k=1}^{\infty}\gamma^{k-1} R_{t+k+1}|S_t=s, A_t=a\right] \\
&= \mathbb{E}\left[R_{t+1}|S_t=s, A_t=a\right] + \\
&\quad \gamma \sum_{S_{t+1},A_{t+1},R_{t+1},\cdots}\left(\sum_{k=1}^{\infty}\gamma^{k-1}R_{t+k+1}\right)\mathbb{P}(S_{t+1},A_{t+1},R_{t+1},\cdots|S_t=s,A_t=a) \\
&= \mathcal{R}_s^a + \gamma \sum_{S_{t+1},A_{t+1},R_{t+1},\cdots}\left(\sum_{k=1}^{\infty}\gamma^{k-1}R_{t+k+1}\right)\mathbb{P}(S_{t+2},A_{t+2},R_{t+2},\cdots|S_{t+1}=s',A_{t+1}=a') \\
&\quad \mathbb{P}(A_{t+1}=a'|S_{t+1}=s')\mathbb{P}(S_{t+1}=s'|S_t=s,A_t=a) \\
&= \mathcal{R}_s^a + \gamma\sum_{s',a'}\pi(a'|s')\mathcal{P}_{ss'}^a \boxed{\mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+2}|S_{t+1}=s', A_{t+1}=a'\right]} \\
&= \mathcal{R}_s^a + \gamma\sum_{s',a'}\pi(a'|s')\mathcal{P}_{ss'}^a \boxed{Q_\pi(s',a')}
\end{aligned}
$$

$$\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1}|S_t=s, A_t=a\right]$$

$$\pi(a|s) = \mathbb{P}(A_t=a|S_t=s)$$

$$\mathcal{P}_{ss'}^a = \mathbb{P}(S_{t+1}=s'|S_t=s,A_t=a)$$

Due to *time-homogeneous* Markov chains!

Bellman Expectation Equations

Similarly, for value function, we have:

$$
\begin{aligned}
V_\pi(s) &= \mathbb{E}_\pi\left[G_t | S_t = s\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \sum_{k=1}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} | S_t = s\right] + \mathbb{E}_\pi\left[\sum_{k=1}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right] \\
&= \sum_a \mathbb{E}_\pi\left[R_{t+1} | S_t = s, A_t = a\right] \mathbb{P}(A_t = a | S_t = s) \\
&\quad + \sum_{s'} \sum_a \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+2} | S_{t+1} = s'\right] \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a) \mathbb{P}(A_t = a | S_t = s) \\
&= \sum_a \mathcal{R}_s^a \pi(a|s) + \sum_{s'} \sum_a \gamma V_\pi(s') \mathcal{P}_{ss'}^a \pi(a|s) \\
&= \sum_a \pi(a|s)\left[\mathcal{R}_s^a + \gamma \sum_{s'} V_\pi(s') \mathcal{P}_{ss'}^a\right]
\end{aligned}
$$

$$\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$$

$$\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$$

$$\mathcal{P}_{ss'}^a = \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a)$$

Bellman Expectation Equations

In summary, we conclude the **Bellman Expectation Equations** as follows.

Given an MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, for any policy π, the value function and the Q function obey the following expectation equations:

$$V_\pi(s) = \sum_a \pi(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_\pi(s') \mathcal{P}_{ss'}^a \right]$$

$$Q_\pi(s, a) = \mathcal{R}_s^a + \gamma \sum_{s', a'} \pi(a'|s') \mathcal{P}_{ss'}^a Q_\pi(s', a')$$

Note that $\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$

$$\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$$

$$\mathcal{P}_{ss'}^a = \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a)$$

Bellman Optimality Equations

Recall the *optimal value function* is $V_*(s) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s \right]$

Recall the *optimal Q function* is $Q_*(s,a) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$

Recall the *optimal policy* is

$$\pi_*(a|s) = \begin{cases} 1 & a = a_*(s) = \underset{a}{\operatorname{argmax}} \; Q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Bellman Optimality Equations

Recall the *optimal value function* is $V_*(s) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s \right]$

Recall the *optimal Q function* is $Q_*(s,a) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$

Recall the *optimal policy* is

$$\pi_*(a|s) = \begin{cases} 1 & a = a_*(s) = \underset{a}{\operatorname{argmax}} \; Q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Similar to the expectation case, we can solve a recursive formula for the optimality case:

$$\begin{aligned} Q_*(s,a) = \max_\pi Q_\pi(s,a) &= \max_\pi \mathcal{R}_s^a + \gamma \sum_{s',a'} \pi(a'|s') \mathcal{P}_{ss'}^a Q_\pi(s',a') \\ &= \mathcal{R}_s^a + \gamma \sum_{s',a'} \pi_*(a'|s') \mathcal{P}_{ss'}^a Q_{\pi_*}(s',a') \\ &= \mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a Q_*(s', a_*(s')) \\ &= \mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a \max_{a'} Q_*(s',a') \end{aligned}$$

Bellman Optimality Equations

Recall the *optimal value function* is $V_*(s) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s \right]$

Recall the *optimal Q function* is $Q_*(s,a) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$

Recall the *optimal policy* is

$$\pi_*(a|s) = \begin{cases} 1 & a = a_*(s) = \underset{a}{\operatorname{argmax}} \; Q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Similar to the expectation case, we can solve a recursive formula for the optimality case:

$$\begin{aligned}
V_*(s) = \max_\pi V_\pi(s) &= \max_\pi \sum_a \pi(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_\pi(s') \mathcal{P}_{ss'}^a \right] \\
&= \sum_a \pi_*(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_{\pi_*}(s') \mathcal{P}_{ss'}^a \right] \\
&= \mathcal{R}_s^{a_*(s)} + \gamma \sum_{s'} V_*(s') \mathcal{P}_{ss'}^{a_*(s)} \\
&= \max_a \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_*(s') \mathcal{P}_{ss'}^a \right]
\end{aligned}$$

Bellman Optimality Equations

Recall the *optimal value function* is $V_*(s) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s\right]$

Recall the *optimal Q function* is $Q_*(s,a) = \max_\pi \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a\right]$

Recall the *optimal policy* is

$$\pi_*(a|s) = \begin{cases} 1 & a = a_*(s) = \underset{a}{\operatorname{argmax}} \; Q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Similar to the expectation case, we can solve a recursive formula for the optimality case:

$$\begin{aligned} V_*(s) = \max_\pi V_\pi(s) &= \max_\pi \sum_a \pi(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_\pi(s')\mathcal{P}_{ss'}^a\right] \\ &= \sum_a \pi_*(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_{\pi_*}(s')\mathcal{P}_{ss'}^a\right] \\ &= \mathcal{R}_s^{a_*(s)} + \gamma \sum_{s'} V_*(s')\mathcal{P}_{ss'}^{a_*(s)} \\ &\boxed{= \max_a \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_*(s')\mathcal{P}_{ss'}^a\right]} \end{aligned}$$

Since

$$\begin{aligned} Q_*(s,a) &= \mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a \max_{a'} Q_*(s',a') \\ &= \mathcal{R}_s^a + \gamma \sum_{s'} Q_*(s', a_*(s'))\mathcal{P}_{ss'}^{a_*(s)} \\ &= \mathcal{R}_s^a + \gamma \sum_{s'} V_*(s')\mathcal{P}_{ss'}^a \end{aligned}$$

Bellman Optimality Equations

In summary, we conclude the **Bellman Optimality Equations** as follows.

Given an MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, for any policy π, the optimal value function and the optimal Q function obey the following expectation equations:

$$V_*(s) = \max_a \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_*(s') \mathcal{P}_{ss'}^a \right]$$

$$Q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a \max_{a'} Q_*(s', a')$$

Note that $\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1} | S_t = s, A_t = a\right]$

$\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$

$\mathcal{P}_{ss'}^a = \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a)$

Outline

- Reinforcement Learning (RL) I
  - Overview & Applications
  - RL Formulation & Taxonomy
  - Markov Decision Process (MDP)
  - Bellman Equations
  - **Solving RL problems using the Bellman Equations**
- Reinforcement Learning (RL) II
  - Model-Free RL: (Deep) Q-learning
  - Model-Free RL: Policy Gradient & Actor-Critic
  - Model-Based RL: Dyna-Q

Solve Problems in RL via Bellman Equations

Two important problems in RL:

- Prediction (*a.k.a.*, policy evaluation): given a policy, evaluate the future, e.g., what is my expected return under that policy?

- Control: optimize the future, e.g., estimating optimal value or Q functions, to find the best policy.

Solve Problems in RL via Bellman Equations

Two important problems in RL:

- Prediction (*a.k.a.*, policy evaluation): given a policy, evaluate the future, e.g., what is my expected return under that policy?

  Use Bellman Expectation Equations to solve!

- Control: optimize the future, e.g., estimating optimal value or Q functions, to find the best policy.

  Use Bellman Optimality Equations to solve!

Solve Problems in RL via Bellman Equations

Prediction (*a.k.a.*, policy evaluation): given a policy, evaluate the future, e.g., what is my expected return under that policy?

From Bellman Expectation Equations, we know:

$$V_\pi(s) = \sum_a \pi(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_\pi(s') \mathcal{P}_{ss'}^a \right]$$

Solve Problems in RL via Bellman Equations

Prediction (*a.k.a.*, policy evaluation): given a policy, evaluate the future, e.g., what is my expected return under that policy?

From Bellman Expectation Equations, we know:

$$V_\pi(s) = \sum_a \pi(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_\pi(s')\mathcal{P}_{ss'}^a\right]$$

We can use fixed-point iteration:

Algorithm 1 Policy Evaluation Algorithm

1: **Input**: Initialize value function V_0 (*e.g.*, to 0)
2: **Repeat**
3:     For all state s, update $V_{k+1}(s) = \sum_a \pi(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_k(s')\mathcal{P}_{ss'}^a\right]$
4: **Until** $V_{k+1}(s) = V_k(s)$ for all s
5: **Return** Final value function V_π

Solve Problems in RL via Bellman Equations

Prediction (*a.k.a.*, policy evaluation): given a policy, evaluate the future, e.g., what is my expected return under that policy?

From Bellman Expectation Equations, we know:

$$V_\pi(s) = \sum_a \pi(a|s)\left[\mathcal{R}_s^a + \gamma \sum_{s'} V_\pi(s')\mathcal{P}_{ss'}^a\right]$$

We can use fixed-point iteration:

Algorithm 1 Policy Evaluation Algorithm

1: **Input**: Initialize value function V_0 (*e.g.*, to 0)
2: **Repeat**
3: For all state s, update $V_{k+1}(s) = \sum_a \pi(a|s)\,[\mathcal{R}_s^a + \gamma \sum_{s'} V_k(s')\mathcal{P}_{ss'}^a]$
4: **Until** $V_{k+1}(s) = V_k(s)$ for all s
5: **Return** Final value function V_π

Under mild conditions (*e.g.*, $\gamma < 1$), this algorithm converges!

Solve Problems in RL via Bellman Equations

Control: optimize the future, e.g., estimating optimal value or Q functions, to find the best policy.

Since we have already known how to evaluate a policy, we just need to improve it!

Solve Problems in RL via Bellman Equations

Control: optimize the future, e.g., estimating optimal value or Q functions, to find the best policy.

Since we have already known how to evaluate a policy, we just need to improve it!

How to improve the policy? Greedy strategy!

Algorithm 2 Policy Iteration Algorithm

1: **Input**: Initialize value function V_0 $(e.g.$, to $0)$ and policy π_0
2: **Repeat**
3: Evaluate the policy π_k
4: For all state s, $\pi_{k+1}(a|s) = \operatorname*{argmax}_a \; Q_{\pi_k}(s,a)$
5: **Until** $\pi_{k+1}(s) = \pi_k(s)$ for all s
6: **Return** Final policy π_*

Solve Problems in RL via Bellman Equations

Control: optimize the future, e.g., estimating optimal value or Q functions, to find the best policy.

Since we have already known how to evaluate a policy, we just need to improve it!

How to improve the policy? Greedy strategy!

Algorithm 2 Policy Iteration Algorithm

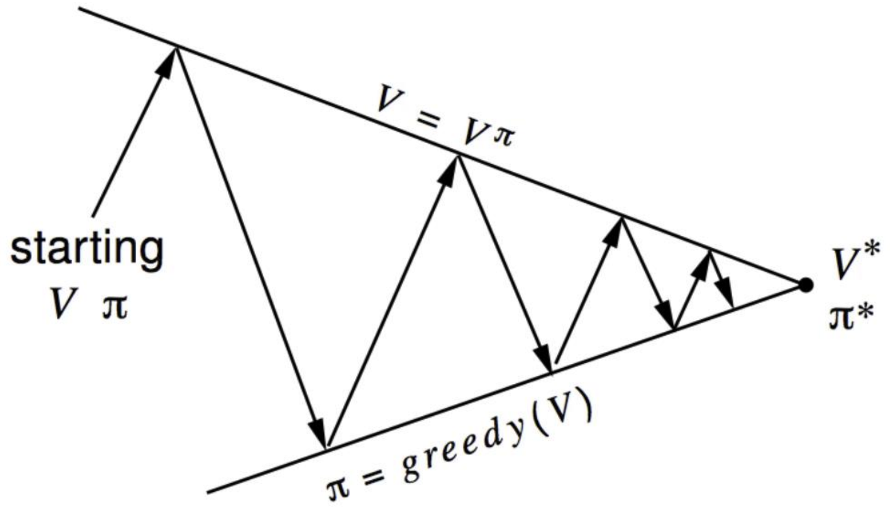
1: **Input**: Initialize value function V_0 (*e.g.*, to 0) and policy π_0
2: **Repeat**
3: Evaluate the policy π_k
4: For all state s, $\pi_{k+1}(a|s) = \underset{a}{\text{argmax}} \; Q_{\pi_k}(s, a)$
5: **Until** $\pi_{k+1}(s) = \pi_k(s)$ for all s
6: **Return** Final policy π_*

One can show that the greedy strategy ensures: $\forall s, V_{\pi_{k+1}}(s) \geq V_{\pi_k}(s)$!

Solve Problems in RL via Bellman Equations

Control: optimize the future, e.g., estimating optimal value or Q functions, to find the best policy.

Since we have already known how to evaluate a policy, we just need to improve it!



Policy evaluation Estimate v_π
Iterative policy evaluation

Policy improvement Generate $\pi' \geq \pi$
Greedy policy improvement

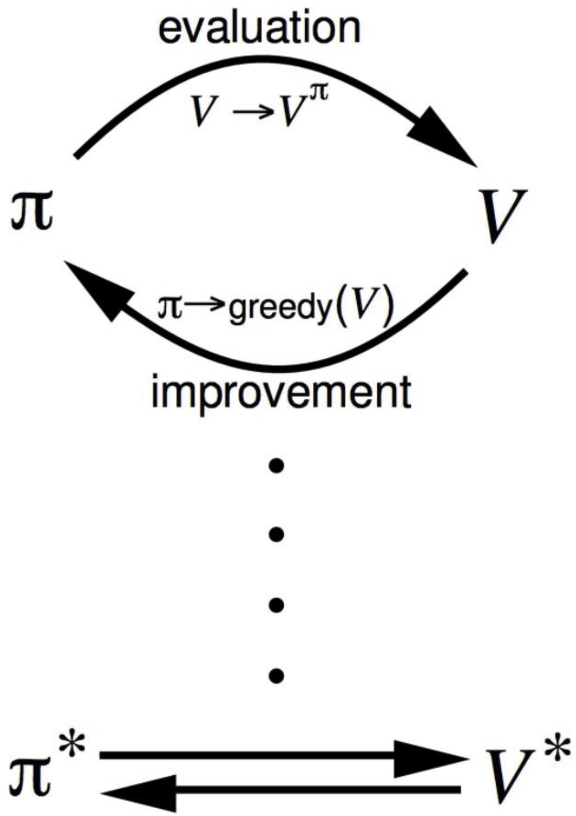


Image Credit: [13]

Solve Problems in RL via Bellman Equations

Control: optimize the future, e.g., estimating optimal value or Q functions, to find the best policy.

We could also leverage the Bellman Optimality Equations!

Algorithm 3 Value Iteration Algorithm

1: **Input**: Initialize value function V_0 (*e.g.*, to 0)
2: **Repeat**
3: For all state s, update $V_*(s) = \max_a \left[\mathcal{R}_s^a + \gamma \sum_{s'} V_*(s')\mathcal{P}_{ss'}^a\right]$
4: **Until** $V_{k+1}(s) = V_k(s)$ for all s
5: **Return** Final value function V_*

Solve Problems in RL via Bellman Equations

Control: optimize the future, e.g., estimating optimal value or Q functions, to find the best policy.

We could also leverage the Bellman Optimality Equations!

Algorithm 3 Value Iteration Algorithm

1: **Input**: Initialize value function V_0 (*e.g.*, to 0)
2: **Repeat**
3: For all state s, update $V_*(s) = \max_a [\mathcal{R}_s^a + \gamma \sum_{s'} V_*(s')\mathcal{P}_{ss'}^a]$
4: **Until** $V_{k+1}(s) = V_k(s)$ for all s
5: **Return** Final value function V_*

This is equivalent to policy iteration with 1-step of policy evaluation!

Solve Problems in RL via Bellman Equations

Control: optimize the future, e.g., estimating optimal value or Q functions, to find the best policy.

We could also leverage the Bellman Optimality Equations!

Algorithm 3 Value Iteration Algorithm

1: **Input**: Initialize value function V_0 (*e.g.*, to 0)
2: **Repeat**
3: For all state s, update $V_*(s) = \max_a [\mathcal{R}_s^a + \gamma \sum_{s'} V_*(s')\mathcal{P}_{ss'}^a]$
4: **Until** $V_{k+1}(s) = V_k(s)$ for all s
5: **Return** Final value function V_*

This is equivalent to policy iteration with 1-step of policy evaluation!

Solve Problems in RL via Bellman Equations

Problem	Bellman Equation	Algorithm
Prediction	Bellman Expectation Equation	Iterative Policy Evaluation
Control	Bellman Expectation Equation + (Greedy) Policy Improvement	Policy Iteration
Control	Bellman Optimality Equation	Value Iteration

Observations:

- Algorithms are based on state-value function $v_\pi(s)$ or $v^*(s) \Rightarrow$ complexity $O(|\mathcal{A}||\mathcal{S}|^2)$ per iteration, for $|\mathcal{A}|$ actions and $|\mathcal{S}|$ states
- Could also apply to action-value function $q_\pi(s,a)$ or $q^*(s,a) \Rightarrow$ complexity $O(|\mathcal{A}|^2|\mathcal{S}|^2)$ per iteration

Image Credit: [12]

References

[1] https://coolinventor.com/wiki/index.php?title=Beginner%27s_Guide_to_Deep_Reinforcement_Learning

[2] https://gym.openai.com/envs/Walker2d-v1/

[3] https://www.deepmind.com/blog/alphastar-mastering-the-real-time-strategy-game-starcraft-ii

[4] https://medium.com/zerone-magazine/the-single-instance-where-man-triumphed-over-ai-the-google-deepmind-challenge-match-1d6af01005a

[5] https://ai.googleblog.com/2021/04/multi-task-robotic-reinforcement.html

[6] https://ai.googleblog.com/2021/01/google-research-looking-back-at-2020.html

[7] https://engineering.princeton.edu/news/2020/11/17/machine-learning-guarantees-robots-performance-unknown-territory

[8] https://awards.acm.org/about/turing-laureates-spotlight

[9] https://www.davidsilver.uk/wp-content/uploads/2020/03/intro_RL.pdf

[10] Murphy, K.P., 2023. Probabilistic machine learning: Advanced topics. MIT Press.

[11] Sutton, R.S. and Barto, A.G., 1998. Reinforcement Learning: An Introduction.

[12] https://github.com/yjavaherian/deepmind-x-ucl-rl/blob/main/slides/All%20Lectures.pdf

[13] https://www.davidsilver.uk/wp-content/uploads/2020/03/DP.pdf

Questions?