

Introduction to Reinforcement Learning

Introduction to Reinforcement Learning (RL)

Key concepts, terminology, and the basic learning loop

1. What Is Reinforcement Learning?

- **Definition:** A learning paradigm where an **agent** interacts with an **environment** to achieve a goal by taking actions that maximize cumulative **reward**.
 - **Contrast with other ML paradigms**
 - **Supervised learning:** learns from labeled input/output pairs.
 - **Unsupervised learning:** discovers structure in unlabeled data.
 - **RL:** learns from **trial-and-error** feedback (rewards) without explicit labels.
-

2. Core Elements of an RL Problem

Element	Symbol	Description
Agent	—	The decision-making entity (e.g., robot, software bot).
Environment	—	Everything the agent interacts with (world, simulator, game).
State	(s_t)	A representation of the environment at time t .
Action	(a_t)	Decision taken by the agent at time t .
Reward	(r_{t+1})	Scalar feedback received after action (a_t) .
Policy	(π)	Mapping from states to actions (deterministic or stochastic).
Value Function	$(V^\pi(s)) /$ $(Q^\pi(s,a))$	Expected return starting from a state (or state-action pair) following policy (π) .
Model (optional)	—	Agent's internal prediction of next state and reward (used in model-based RL).

3. The RL Interaction Loop

1. **Observe** current state (s_t) .
2. **Select** action $(a_t \sim \pi(\cdot|s_t))$.
3. **Execute** action in the environment.

Introduction to Reinforcement Learning

4. **Receive** reward (r_{t+1}) and next state (s_{t+1}).
 5. **Update** policy/value estimates based on the new experience.
 6. **Repeat** from step 1 until a terminal condition is met.
-

4. Key Objectives

- **Return (cumulative reward)**

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

- ($\gamma \in [0,1]$) is the **discount factor** (balances immediate vs. future rewards).

- **Goal:** Find an optimal policy ($\pi^* = \arg\max_{\pi} \mathbb{E}[G_t \mid \pi]$).
-

5. Types of RL Algorithms

Category	Main Idea	Typical Algorithms
Model-free	Learn directly from interaction, no explicit model of the environment.	- Value-based: Q-learning, SARSA - Policy-gradient: REINFORCE, Actor-Critic
Model-based	Build a model ($\hat{P}(s'$	$s,a)$, $\hat{R}(s,a)$) and plan with it.
Off-policy vs. On-policy	Off-policy learns about a policy different from the one used to generate data (e.g., Q-learning). On-policy learns about the behavior policy (e.g., SARSA).	- Off?