Foundations

Key Concepts: Rewards, States, Actions, Policies

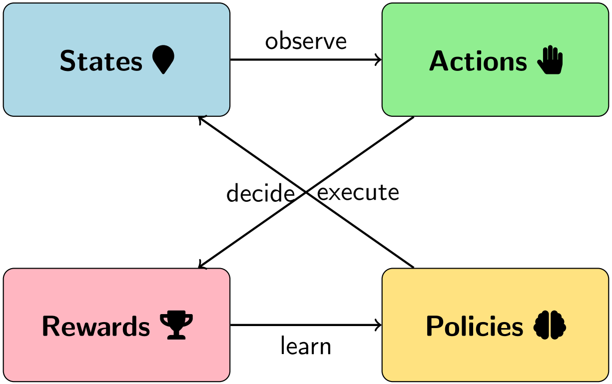
Sarwan Ali

Department of Computer Science
Georgia State University

Reinforcement Learning Foundations

Today's Learning Journey

The Four Pillars of Reinforcement Learning



These four concepts form the foundation of every reinforcement learning problem

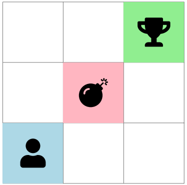
States: The World's Description

What is a State?

- A complete description of the environment at a given time
- Contains all relevant information for decision making
- Denoted as $s \in \mathcal{S}$ (state space)

Key Properties:

- **Markov Property**: Future depends only on current state
- **Observable**: Agent can perceive the state
- **Discrete or Continuous**: Finite vs infinite state spaces

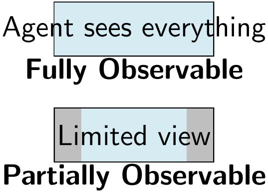


Grid World States

Types of States

Fully Observable

- Agent sees complete environment state
- s_t = environment state
- Example: Chess, Tic-tac-toe



Partially Observable

- Agent has limited view
- o_t = observation $\neq$ state
- Must maintain belief state
- Example: Poker, autonomous driving

Mathematical Representation

State space: $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ (discrete) or $\mathcal{S} \subseteq \mathbb{R}^n$ (continuous)

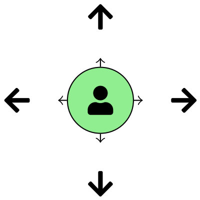
Actions: What the Agent Can Do

What is an Action?

- Choices available to the agent
- Way to influence the environment
- Denoted as $a \in \mathcal{A}$ (action space)

Types of Action Spaces:

- **Discrete**: Finite set of actions
- **Continuous**: Real-valued actions
- **State-dependent**: $\mathcal{A}(s)$ varies by state



Discrete Actions

Examples

- **Game**: {Up, Down, Left, Right, Shoot}
- **Trading**: {Buy, Sell, Hold}
- **Robot**: Joint angles $\mathbb{R}^7$ (continuous)

Action Space Characteristics

Discrete Actions: $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$. **Examples:** Atari games, Board games, Menu selection

Finite choices

Continuous Actions: $\mathcal{A} \subseteq \mathbb{R}^n$. **Examples:** Robot control, Vehicle steering, Portfolio weights

Infinite possibilities

Mathematical Formulation

Action taken at time t: $a_t \in \mathcal{A}(s_t)$ where $\mathcal{A}(s_t)$ is the set of valid actions in state s_t

Rewards: The Learning Signal

What is a Reward?

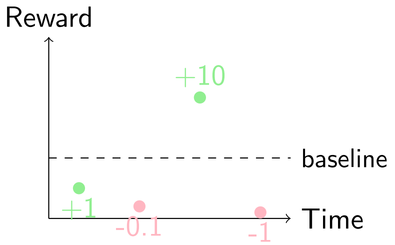
- Scalar feedback signal
- Indicates how good an action was
- Denoted as $r_t \in \mathbb{R}$
- **The only way agent learns!**

Reward Function:

$$R(s, a, s') = \mathbb{E}[r_{t+1} | s_t = s, a_t = a, s_{t+1} = s']$$

Key Principles:

- Immediate vs Delayed rewards
- Sparse vs Dense rewards
- Positive, negative, or zero



Reward Signal Over Time

Reward Hypothesis

"All goals can be described by the maximization of expected cumulative reward"

Types of Rewards

Dense Rewards: Frequent feedback, Easy to learn, May lead to myopic behavior

Shaped Rewards: Engineered guidance, Balance of both, Risk of reward hacking

Sparse Rewards: Infrequent feedback, Hard to learn, More realistic

Examples

- **Dense**: Game score every frame
- **Sparse**: Win/lose at end of game
- **Shaped**: Distance to goal + win bonus

Policies: The Agent's Strategy

What is a Policy?

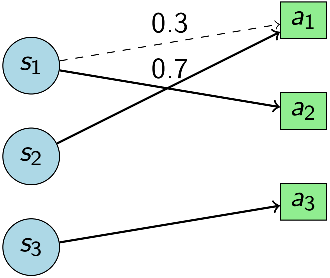
- Agent's behavior function
- Maps states to actions
- Denoted as $\pi : \mathcal{S} \to \mathcal{A}$
- **The brain of the agent**

Mathematical Definition:

$$\pi(a|s) = P(a_t = a|s_t = s)$$

Types:

- **Deterministic**: $a = \pi(s)$
- **Stochastic**: $\pi(a|s)$ (probability distribution)

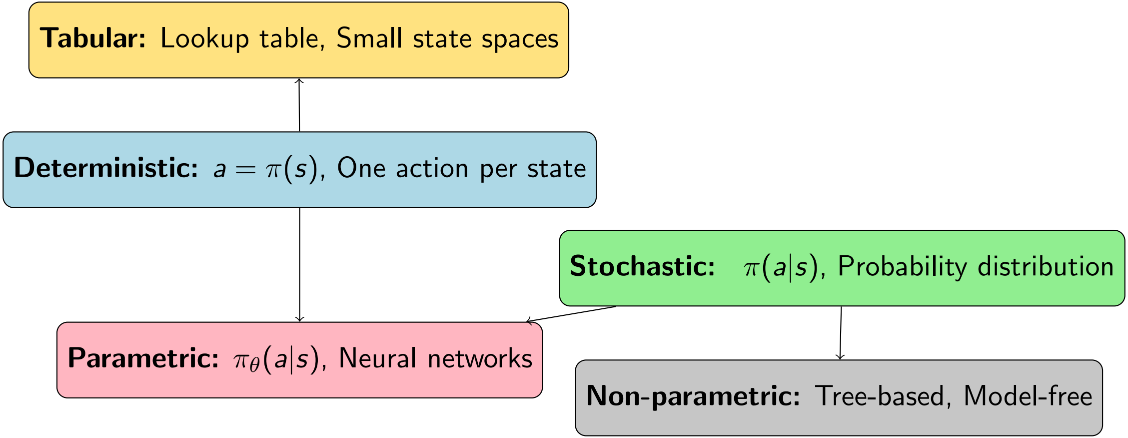


Stochastic Policy

Goal

Find optimal policy π^* that maximizes expected cumulative reward

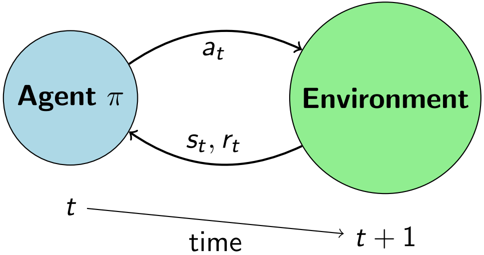
Policy Types and Representations



Policy Optimization

Learning involves finding π^* through exploration and exploitation

The RL Loop: How Components Interact



Agent observes s_t

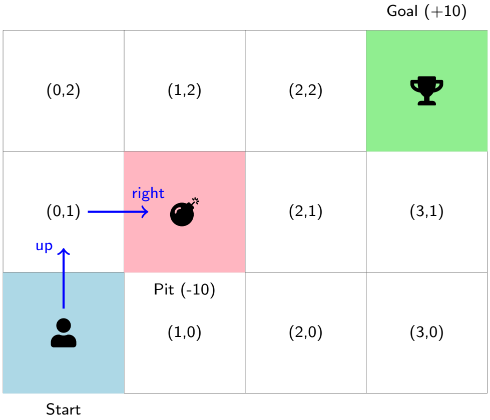
Policy selects a_t

Environment returns r_t, s_{t+1}

The Sequential Decision Process:

1. Agent observes state s_t
2. Policy π selects action a_t
3. Environment provides reward r_t and next state s_{t+1}
4. Agent updates policy and repeats

Example: Grid World Navigation



States: Grid positions
Actions: {up, down, left, right}
Rewards: Goal: +10, Pit: -10, Step: -1
Policy: Navigate to goal safely

Key Relationships

State $\rightarrow$ Action

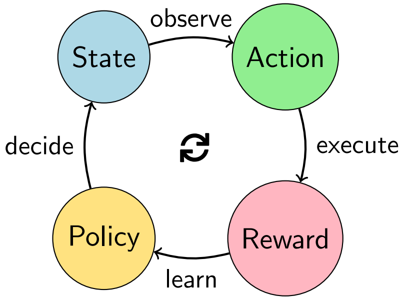
- Policy determines action selection
- $a_t \sim \pi(\cdot|s_t)$

Action $\rightarrow$ Reward

- Environment provides feedback
- $r_t = R(s_t, a_t, s_{t+1})$

Reward $\rightarrow$ Policy

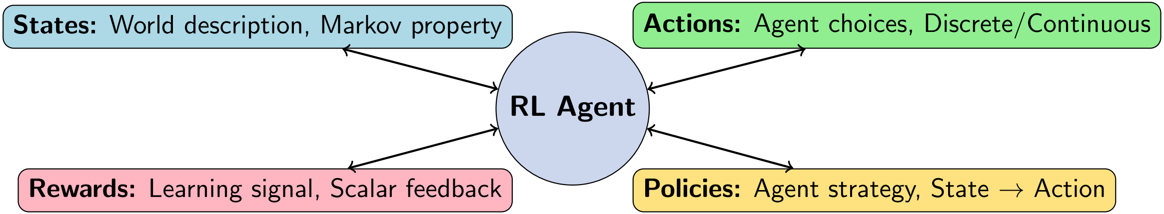
- Learning signal for improvement
- π updated to maximize rewards



Central Equation

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t | \pi\right]$$

Key Takeaways



Essential Points

- These four components define every RL problem
- Agent learns optimal policy through trial and error
- Goal: Maximize expected cumulative reward
- Foundation for all RL algorithms we'll study

Next Steps

What's Coming Next?

Examples of RL applications (games, robotics, recommendation systems)

Preparation:

- Review probability theory
- Think about how to formalize the concepts learned today
- Consider: How do we measure policy quality?

These foundations will be the building blocks for everything that follows!