Reinforcement Learning (RL) - A Simplified Course

Table of Contents

Introduction to Reinforcement Learning (RL)

What is RL?

Reinforcement Learning (RL) is a branch of machine learning where an agent learns by interacting with an environment to achieve a goal. The agent receives feedback in the form of **rewards**, guiding it to take the best actions over time.

Key Concepts:

- **Agent:** The decision-maker (e.g., robot, AI system).
- **Action:** The choices available to the agent.
- **Environment:** The world in which the agent operates.
- **State:** The situation or condition of the agent in the environment at a given time.
- **Reward:** Feedback received for an action, guiding the agent toward optimal behavior.

How RL Works: The Learning Process

RL is based on a **trial-and-error** approach, where the agent learns from experiences to maximize cumulative rewards. The process involves:

1. The agent observes the **state** of the environment.
2. It selects an **action** based on a policy (strategy).
3. The environment responds with a **new state** and a **reward**.
4. The agent updates its policy to maximize future rewards.
5. The cycle continues until the agent learns an optimal behavior.

Environment Interaction Loop

- The agent interacts with the environment.
- It takes actions based on its current state.
- The environment provides rewards and updates the state.
- The agent refines its policy to improve future decisions.

RL Agent Components

1. Policy

Defines the agent's behavior at any given time. It is a mapping of perceived environmental states to actions.

- **Deterministic Policy:** Always chooses the same action for a given state.
- **Stochastic Policy:** Chooses an action based on probability distribution.

2. Value Function

Estimates the long-term reward of being in a particular state or taking a specific action.

- **State-Value Function (V):** The expected cumulative reward from a state.
- **Action-Value Function (Q):** The expected cumulative reward from taking a specific action in a given state.

3. Model of the Environment

A model predicts how the environment will respond to actions. Some RL algorithms use models for planning, while others (like model-free RL) learn directly from experience.

Reward Hypothesis in RL

Rewards define what the agent should achieve. The goal is to **maximize expected cumulative rewards** over time.

Reward Calculation:

- **Immediate Reward:** Instant feedback after an action.
- **Discounted Future Rewards:** Future rewards are discounted using a factor **gamma (γ)**.

Reward Function Formula:

Where:

- **G_t** = Cumulative reward at time step t.
- **R_t** = Reward at time step t.
- **γ (gamma)** = Discount factor ($0 \leq \gamma \leq 1$), determining the importance of future rewards.

Example: Rabbit in a Maze

Scenario:

- A rabbit (agent) wants to eat carrots (reward) while avoiding an electric shock.
- Rewards decrease near dangerous zones.
- A high **gamma (γ)** encourages long-term planning, while a low **gamma** prioritizes immediate rewards.

Challenges in Implementing RL

1. **High Computational Costs**
    - RL requires exploring multiple actions and states, leading to high time and resource consumption.
2. **Domain Specificity**
    - Policies trained in one environment may not generalize well to others.
3. **Designing Reward Functions**
    - Poorly designed rewards can lead to unintended or suboptimal behaviors.

Summary

- **Reinforcement Learning** is about learning through **trial and error** using **rewards**.
- **The agent** interacts with the **environment** to learn an optimal **policy**.
- **Key components** include **policy, value function, and environment model**.
- **Rewards drive learning**, and the goal is to **maximize cumulative rewards**.
- **Challenges** include computational demands, domain dependency, and reward design.

Next Steps:

- Explore **popular RL algorithms** like Q-learning, Deep Q-Networks (DQN), and Policy

Gradient Methods.

- Implement simple RL experiments using Python and OpenAI Gym.
- Learn how RL is applied in **robotics, finance, and gaming**.

Reward Hypothesis

- A reward R_t is a scalar feedback signal that shows how well an agent performs at step t.
- It maximizes the expected cumulative reward for best performance.
- Each time step t for cumulative reward is given by:

$$G_t = R_{t+1} + R_{t+2} + \ldots = \sum_{k=0}^{T} R_{t+k+1}$$

- Discounted cumulative expected rewards:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \text{ where } \gamma \in [0, 1) = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \ldots$$

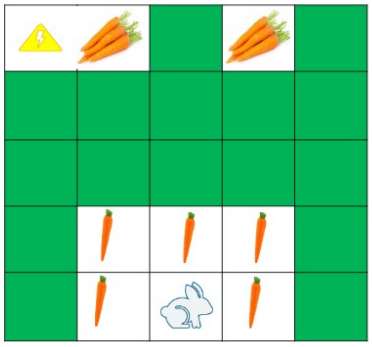
- Each reward will be discounted by gamma to the exponent of the time step.

Reward Hypothesis: Example

Agent: Rabbit

Goal: To eat maximum carrots before getting a shock

- The closer the agent is to the electric shock, the more dangerous it is.
- The rewards will be discounted near the electric shock even if it is bigger (more carrots).
- To discount the rewards, define a discount rate called gamma (between 0 and 1).
- The larger the gamma, the smaller is the discount (agent cares more about the long-term reward).
- The smaller the gamma, the bigger the discount (agent cares more about the short-term reward, the nearest carrots).



Maze example demonstrating rewards

RL Agent Components

Policy	▪ Defines the agent's way of behaving at a particular period of time ▪ Mapping of perceived states of the environment to actions to be taken in those states
Value function	▪ Specifies the long-term positives of the model ▪ Value of a state — the total amount of reward an agent can anticipate to gather over the future, beginning from that state
Model of the environment	▪ Allows conclusions to be made about how the environment will act ▪ Might predict the resultant next state and next reward

Environment, Agent and Information State

Environment state	▪ Environment's state S_t^e: Environment's internal representation ▪ Information used by the environment to select the next observation/reward
Agent state	▪ Agent's state S_t^a: Agents, internal representation ▪ Information used by the agent to select the next action
Information state	▪ Holds all useful information from the history ▪ Given the present, future not being dependent on the past