An Expert's Guide to Reinforcement Learning:
From Core Concepts to Advanced Algorithms

A Comprehensive Exposition

By Taha Majlesi

July 17, 2025

Contents

Part I

The Foundations of Reinforcement Learning

Introduction to Part I

This initial part of the report establishes the conceptual and mathematical groundwork for Reinforcement Learning (RL). It is designed to provide a robust understanding of the fundamental principles that govern this unique field of machine learning. Our goal here is to answer the essential questions of ”what” RL is and ”why” it is important before proceeding to the ”how” of its various algorithmic implementations in later parts. The sections that follow will define the core problem of sequential decision-making, formalize it using the precise language of Markov Decision Processes (MDPs), and detail the objectives that a Reinforcement Learning agent seeks to optimize.

0.1 The Reinforcement Learning Problem

Reinforcement Learning (RL) emerges as a distinct and powerful paradigm within machine learning, singularly focused on the challenge of **goal-directed learning from interaction**. It provides a computational framework for an *agent* to learn optimal behaviors through a process of trial and error, navigating complex and uncertain *environments* to achieve specified objectives. This section introduces the fundamental problem that RL aims to solve, contrasting it with other machine learning methodologies to highlight its unique characteristics and challenges.

0.1.1 Motivation: Sequential Decision-Making Under Uncertainty

At its heart, Reinforcement Learning addresses the problem of how an intelligent agent ought to take actions in an environment to maximize some notion of cumulative reward. This problem is characterized by **sequential decision-making under uncertainty**, where the agent must make a series of choices over time. A critical feature of this problem is that the full consequences of an action may not be immediately apparent; instead, feedback is often **delayed**. An action taken now might have positive or negative repercussions far into the future.

An Illustrative Example: The Robotic Arm

To make this concrete, consider the task of a robotic manipulator learning to pick up an object from a bin.

A Traditional Approach (Supervised Learning): A classical machine learning approach would be to use supervised learning. This would require a human to create a vast, labeled dataset of successful and unsuccessful grasps. Each data point would need to specify the precise parameters of a grip (e.g., joint angles, gripper position, force) and its outcome (a label of ”success” or ”failure”). Collecting such an exhaustive and perfectly labeled dataset is often impractical, time-consuming, and prohibitively expensive. Furthermore, if the object's shape or position changes, the entire dataset might become obsolete.

The Reinforcement Learning Approach: RL offers a more flexible and autonomous path. The agent (the robotic arm's control program) learns the task through direct interaction with its environment.

1. **Action:** It attempts to grip the object at various locations, specified by coordinates like (x, y, z).
2. **Observation & Reward:** After each attempt, it receives a simple feedback signal—a **reward**. For instance, it might receive a reward of $+1$ for a successful grasp and a reward of -1 for a failure.
3. **Learning:** Through a process of trial and error, the agent learns to associate its actions (choosing grip coordinates) with outcomes (rewards). It gradually refines its strategy, or **policy**, to increase the frequency of successful grasps.

This example illuminates the core tenets of the RL problem:

- **The optimal strategy is unknown:** The best policy is not known in advance and must be discovered by the agent.

- **Learning occurs through a sequence of actions:** The agent's behavior is not evaluated based on a single decision, but on a sequence of decisions over time. A successful grasp is the result of a whole sequence of correct adjustments.

- **Trial-and-error is the learning mechanism:** The agent learns by exploring different actions and observing the outcomes, a form of active search.

- **Feedback is often delayed and evaluative:** The reward signal indicates the overall quality of a sequence of actions, not the correctness of any single one. It tells the agent *what* it achieved (a good outcome), but not *how* it should have acted differently.

- **Data is generated actively and is non-IID:** The agent generates its own data through exploration. This data is dynamic, and its statistical properties change as the agent learns. This is a stark contrast to the static, independent, and identically distributed (IID) datasets used in supervised learning.

0.1.2 Contrasting RL with Supervised and Unsupervised Learning

To fully appreciate the uniqueness of Reinforcement Learning, it is essential to contrast it with the other two primary paradigms of machine learning. Their differences lie in their fundamental objectives, data requirements, and feedback mechanisms.

- **Supervised Learning:** This paradigm operates on the principle of learning from examples with known answers, much like a student learning with a teacher. It requires a static, pre-collected dataset where each input x is paired with a correct output label y. The goal is to learn a mapping function f such that $f(x)$ accurately predicts y for new, unseen inputs. The feedback is **direct and instructive**: the model's prediction is compared against the ground-truth label, and the resulting error is used to adjust the model's parameters.

- **Unsupervised Learning:** This paradigm deals with unlabeled data. Its objective is not to predict a specific output but to discover hidden structures, patterns, or natural groupings within the data itself. There is no explicit feedback signal or "correct" answer. The model learns by identifying similarities or anomalies in the data, making it ideal for tasks like customer segmentation or dimensionality reduction.

- **Reinforcement Learning:** RL occupies a third, distinct space. It is not about prediction from static data but about learning a **strategy for decision-making** in a dynamic environment. The agent does not begin with a predefined dataset; it generates its own data through interaction. The feedback mechanism is a scalar **reward signal** from the environment, which is **evaluative, not instructive**. It indicates how good an action was but does not specify which action would have been better. This process of learning from sparse, often delayed, rewards to achieve a long-term goal is the defining characteristic of RL.

The Active Data Generation Problem

A core differentiator of RL is its active data generation process. In supervised and unsupervised learning, the algorithm is a passive recipient of a static dataset. In RL, the agent is both a consumer and a producer of its own data. The agent's current strategy (its policy) directly influences the distribution of future states and rewards it will encounter. This creates a challenging non-stationary learning problem where the agent must continuously adapt its behavior based on the feedback from an environment it is simultaneously influencing. The well-known **exploration-exploitation dilemma**—the trade-off between trying new things to find better rewards (exploration) and sticking with what is known to be good (exploitation)—is a direct and unavoidable consequence of this active, closed-loop learning process.

Table 1: Comparison of Machine Learning Paradigms

Criterion	**Supervised Learning**	**Unsupervised Learning**	**Reinforcemen**
Definition	Learns a mapping from labeled data with a "teacher" providing correct answers.	Discovers hidden patterns and structures in unlabeled data without guidance.	An agent learns sequential decis trial-and-error i an environment cumulative rew
Type of Data	Labeled data, (input, output) pairs.	Unlabeled data, only inputs are provided.	No predefined c generates its ow data (state, act through interac
Goal/Objective	To learn a function that accurately predicts the output for new inputs.	To find inherent groupings or patterns in the data.	To learn an opt maximizes the reward over tim
Feedback	Direct, instructive feedback via error between prediction and ground-truth label.	No explicit feedback signal; learning is based on data structure.	Evaluative feed reward signal, sparse and dela

0.1.3 The Agent-Environment Interaction Loop: An Anatomy of RL

The process of Reinforcement Learning is formally described through the **agent-environment interaction loop**, a framework that illustrates the continuous cycle of sensing, acting, and learning. This loop consists of two primary components:

- **The Agent:** The learner and decision-maker. It is the entity that perceives the environment and executes actions. In our robotics example, the agent is the computer program controlling the manipulator. It embodies the policy and the learning algorithm.

- **The Environment:** Everything outside the agent with which it interacts. It comprises the world in which the agent operates, responds to the agent's actions, and presents new situations to the agent. For the robot, the environment includes the bin, the object, the camera, and the physical laws governing their interactions.

The interaction unfolds over a sequence of discrete time steps, $t = 0, 1, 2, \ldots$. The cycle proceeds as follows:

https://www.researchgate.net/profile/Fabio-Bonsignorio/publication/323217939/

Figure 1: The fundamental agent-environment interaction loop in Reinforcement Learning. The agent and environment continuously interact, each influencing the other in a cycle of action and observation.

1. **Observation of State:** At each time step t, the agent receives an observation of the environment's state, denoted as $S_t \in \mathcal{S}$, where $\mathcal{S}$ is the set of all possible states. The state contains all the information the agent needs to make a decision.

2. **Action Selection:** Based on the state S_t, the agent selects an action, $A_t \in \mathcal{A}(S_t)$, from the set of available actions in that state. This decision is dictated by the agent's current policy, π.

3. **Environment Response:** As a consequence of the agent's action A_t, the environment transitions to a new state, S_{t+1}, and provides a scalar reward, $R_{t+1} \in \mathbb{R}$, to the agent. This reward tells the agent about the immediate outcome of its action.

This continuous loop generates a sequence of states, actions, and rewards, known as a **trajectory** or history:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, \ldots$$

The agent's sole objective is to learn a behavior strategy, called a **policy (π)**, which maps states to actions, in order to maximize the total amount of reward it receives over the long run. The challenge lies in learning from this stream of experience to find a policy that is effective not just for immediate rewards, but for the cumulative, long-term outcome.