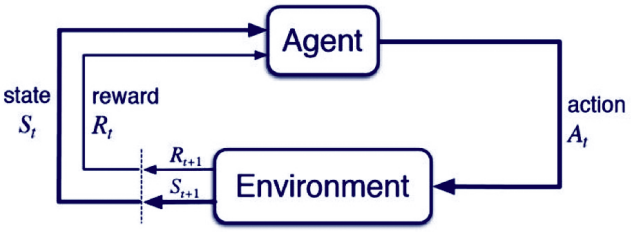
Deep Reinforcement Learning (Sp25)

Instructor: Dr. Mohammad Hossein Rohban

Summary of Lecture 1: Introduction to RL

Summarized By: Arash Alikhani

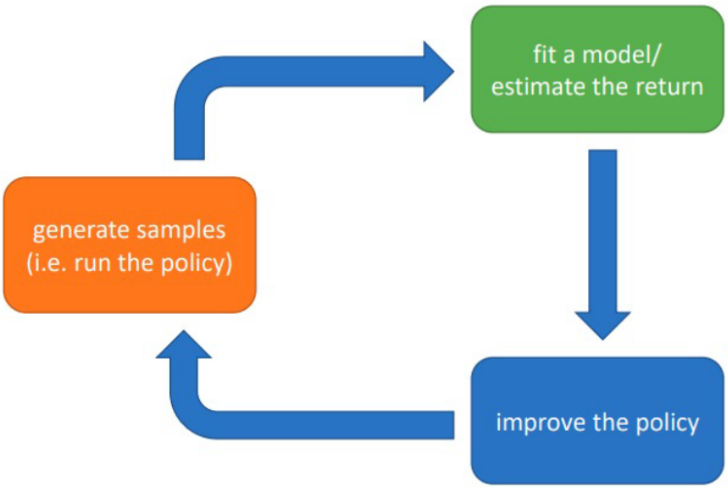
- In **SL**, the ground truth labels are known beforehand, and models are trained on static, independent, and identically distributed (iid) data to minimize prediction errors.
- In contrast, **RL** typically operates without prior knowledge of the best action or policy, requiring an agent to interact with an environment through a sequence of actions to discover optimal strategies.
- **RL** involves trial-and-error search, where feedback in the form of rewards may be delayed, making learning more complex.
- Unlike SL, **RL** deals with dynamic, non-iid data due to the agent's exploration of the environment and the evolving nature of the decision-making process.
- The **RL process** is a loop that generates a sequence of states, actions, rewards, and next states, collectively referred to as experiences. Reinforcement learning is learning what to do—how to map situations to actions—so as to maximize a numerical reward signal.



pick your own actions

input: $\mathbf{s}_t$ *at each time step*
output: $\mathbf{a}_t$ *at each time step*
data: $(\mathbf{s}_1, \mathbf{a}_1, r_1, \ldots, \mathbf{s}_T, \mathbf{a}_T, r_T)$
goal: learn $\pi_\theta : \mathbf{s}_t \rightarrow \mathbf{a}_t$
to maximize $\sum_t r_t$

- **Policy:** The agent's brain that dictates what action to take given a state. An optimal policy maximizes the expected return and is learned through training.
- **Anatomy of Reinforcement Learning:** The agent interacts with the environment to gather experience, estimates returns or fits a model to assess performance, and updates the policy to maximize long-term rewards.



- In this loop, generating samples can be costly when using real-world systems like robots due to real-time constraints. Model learning ($s_{t+1} \approx f_\phi(s_t, a_t)$) is computationally expensive due to the need for complex function approximations. In contrast, fitting returns and policy updates (e.g., gradient-based backpropagation) are generally faster and less resource-intensive.

- **State Value (V) / Value Function:** Represents the expected cumulative reward starting from a given state and following a specific policy. It evaluates the long-term value of being in a particular state.
- **State-Action Value (Q) / Q-Function:** Represents the expected cumulative reward starting from a given state, taking a specific action, and then following a specific policy. It evaluates the long-term value of taking a particular action in a given state.
- **Value-Based RL:** Focuses on learning value functions such as $Q(s, a)$ to evaluate the quality of actions. The policy is derived implicitly by selecting actions that maximize the value function.
- **Direct Policy Gradient:** Directly optimizes the policy by maximizing the expected return using gradients of the policy parameters.
- **Actor-Critic RL:** Combines value-based and policy-based methods by having an actor (policy) and a critic (value function). The critic evaluates the policy's actions and guides the actor, balancing stability and exploration efficiency.
- **Model-Based RL:** Learns a model of the environment's dynamics to predict future states and rewards. This enables planning and efficient policy learning but may suffer from inaccuracies if the model is poorly estimated.
- **Reward Hypothesis:** Assumes that all goals of an agent can be described as the maximization of a cumulative reward function.
- **Imitation Learning:** Involves training an agent by directly copying or mimicking expert demonstrations without explicitly using a reward signal.
- **Inverse RL:** Seeks to infer the hidden reward function that an expert is optimizing based on their observed behavior.
- **AI Planning:** Uses a known model of the environment to develop a sequence of actions to achieve specific goals through internal computations (deliberation, reasoning, introspection, pondering, thought, search) without external interaction, refining its policy solely based on this model.
- **Reinforcement Learning:** Combines aspects of both planning and learning. The environment is initially unknown, and the agent gathers experiences through interaction. If a model is learned, the agent can perform internal computations similar to planning, while still relying on real-world feedback to update its policy.
- Comparison of AI Planning, Supervised Learning (SL), Unsupervised Learning (UL), Reinforcement Learning (RL), and Imitation Learning (IL) in terms of optimization, learning from experience, generalization, delayed consequences, and exploration:

	AI Planning	SL	UL	RL	IL
Optimization	X			X	X
Learns from experience		X	X	X	X
Generalization	X	X	X	X	X
Delayed Consequences	X			X	X
Exploration				X	