02. Reinforcement Learning Basics

Introduction:

- Reinforcement Learning is a conversation between the agent and the environment.
- The environment reveals itself to the agent in the form of “states”. The agent takes “actions” and receive “rewards” for (state-action) combinations.
- The agent doesn’t know the environment. The agent experiences the environment be continuously interacting with it.

Behavior Structures:

- The goal is to create learning algorithms that can approximate the solution of a problem.
- We’re trying to learn a behavior, a way of interacting with the environment that obtains high reward.
- There’re different kinds of behavior:
  - Plan: A fixed sequence of actions, that once you choose it, you stick to executing it. The problem with this behavior is:
    1. During learning you don’t know what kind of plan to execute.
    2. Stochasticity: The environment might not be deterministic.
  - Conditional Plan: This is a plan with if-conditions to change the actions depending on what we experience in the environment. This is different from “dynamic planning”, which encourages the agent to create a new plan if the original predictions were wrong.
  - Stationary Policy/Universal Plan: Mapping from states to actions.
    1. Can handle any kind of stochasticity.
    2. It’s a special version of a Conditional Plan where you had the same if-condition at every possible state.
    3. Very large.
    4. There’s always an optimal stationary policy for any MDP.

Evaluating a Policy:

- We need to ask the question: What do we mean by an “optimal policy”?
- By following a policy from the initial state, we will have a sequence of states, actions, and rewards. There are multiple possible states, actions, and rewards that can happen because the domains are stochastic.
- The goal is to translate each policy to a single number so that we can compare policies and choose the optimal one.

- How to turn a set of sequences of states into a single number:
  - Turn each of the state transitions into actual immediate rewards.
  - Truncate according to horizon: The strings of numbers we get from the first step is infinitely long. If we have a finite horizon, we can cut off the sequence after the finite number of transitions.
  - Summarize each sequence: Turning each of the sequences into a single number (the Return). We do that by adding the rewards according to a discount factor (γ):

$$\sum_{i=1}^{T} \gamma^i r_i$$

  - Summarize over sequences: We do that by averaging the sequences by the likelihood of each sequence (expectation).

Evaluating a Learner:

- A good learner returns a good policy.
- If multiple learners return the same policy, we investigate:
  - Computational Complexity: How much time it takes the learner to come up with the policy.
  - Sample/Experience Complexity: How much data (how many interactions) the learner needs to come up with the policy.
- Space Complexity is not interesting in the Reinforcement Learning domain.