A.I. Wiki

A Beginner’s Guide to Important Topics in AI, Machine Learning, and Deep Learning.

Search

Reinforcement Learning Definitions

Reinforcement learning employs a number of terms of art to express the concepts you must learn in order to understand reinforcement learning as a whole. As in many disciplines, the terms of reinforcement learning interlock; that is, they are used to define each other. The more of the them you learn, the better you will understand the rest. So what is gateway experience or definition that will help you understand the jargon of this field? The gateway is an analogy.

The most basic terms of reinforcement learning echo those of the individual human life. A useful metaphor for a reinforcement learning agent is simply a human being moving through the world. Reinforcement learning happens to codify the structure of a human life in mathematical statements, and as you sink deeper into RL, you will add a layer of mathematical terms to those that are drawn from the basic analogy. And indeed, understanding RL agents may give you new ways to think about how humans make decisions.

Several concepts distinguish reinforcement learning from other types of machine learning and optimization, including the ideas of agents, environments, states, actions and rewards. The glossary on this page will define and illustrate those terms and others.

Below, capital letters tend to denote sets of things, and lower-case letters denote a specific instance of that thing; e.g. `A` is all possible actions, while `a` is a specific action contained in the set.

- Agent: An **agent** takes actions; for example, a drone making a delivery, or Super Mario navigating a video game. The algorithm is the agent. In life, the agent is you.
- Action (A): `A` is the set of all possible moves the agent can make. An **action** is almost self-explanatory, but it should be noted that agents usually choose from a list of discrete, possible actions. In video games, the list might include running right or left, jumping high or low, crouching or standing still. In the stock markets, the list might include buying, selling or holding any one of an array of securities and their derivatives. When handling aerial drones, alternatives would include many different velocities and accelerations in 3D space.
- Discount factor: The **discount factor** is multiplied by future rewards as discovered by the agent in order to dampen thse rewards’ effect on the agent’s choice of action. Why? It is designed to make future rewards worth less than immediate rewards; i.e. it enforces a kind of short-term hedonism in the agent. Often expressed with the lower-case Greek letter gamma: γ. If γ is .8, and there’s a reward of 10 points after 3 time steps, the present value of that reward is 0.8^3 x 10 . A discount factor of 1 would make future rewards worth just

as much as immediate rewards. We're fighting against delayed gratification (https://en.wikipedia.org/wiki/Stanford_marshmallow_experiment) here.

- Environment: The world through which the agent moves, and which responds to the agent. The environment takes the agent's current state and action as input, and returns as output the agent's reward and its next state. If you are the agent, the environment could be the laws of physics and the rules of society that process your actions and determine the consequences of them.
- State (S): A **state** is a concrete and immediate situation in which the agent finds itself; i.e. a specific place and moment, an instantaneous configuration that puts the agent in relation to other significant things such as tools, obstacles, enemies or prizes. It can the current situation returned by the environment, or any future situation. Were you ever in the wrong place at the wrong time? That's a state.
- Reward (R): A **reward** is the feedback by which we measure the success or failure of an agent's actions in a given state. For example, in a video game, when Mario touches a coin, he wins points. From any given state, an agent sends output in the form of actions to the environment, and the environment returns the agent's new state (which resulted from acting on the previous state) as well as rewards, if there are any. Rewards can be immediate or delayed. They effectively evaluate the agent's action.
- Policy (π): The **policy** is the strategy that the agent employs to determine the next action based on the current state. It maps states to actions, the actions that promise the highest reward.
- Value (V): The expected long-term return with discount, as opposed to the short-term reward R. $V\pi(s)$ is defined as the expected long-term return of the current state under policy π. We discount rewards, or lower their estimated value, the further into the future they occur. See *discount factor*. And remember Keynes: "In the long run, we are all dead." That's why you discount future rewards. It is useful to distinguish
- Q-value or action-value (Q): **Q-value** is similar to Value, except that it takes an extra parameter, the current action a. $Q\pi(s, a)$ refers to the long-term return of an action taking action a under policy π from the current state s. Q maps state-action pairs to rewards. Note the difference between Q and policy.
- Trajectory: A sequence of states and actions that influence those states. From the Latin "to throw across." The life of an agent is but a ball tossed high and arching through space-time unmoored, much like humans in the modern world.
- Key distinctions: Reward is an immediate signal that is received in a given state, while value is the sum of all rewards you might anticipate from that state. Value is a long-term expectation, while reward is an immediate pleasure. Value is eating spinach salad for dinner in anticipation of a long and healthy life; reward is eating cocaine for dinner and to hell with it. They differ in their time horizons. So you can have states where value and reward diverge: you might receive a low, immediate reward (spinach) even as you move to position with great potential for long-term value; or you might receive a high immediate reward (cocaine) that leads to diminishing prospects over time. This is why the value function, rather than immediate rewards, is what reinforcement learning seeks to predict and control.

Share Tweet

Chris Nicholson

Chris Nicholson is the CEO of Pathmind. He previously led communications and recruiting at the Sequoia-backed robo-advisor, FutureAdvisor, which was acquired by BlackRock. In a prior life, Chris spent a decade reporting on tech and finance for The New York Times, Businessweek and Bloomberg, among others.