Introduction to Reinforcement Learning

Credits: RL Course Google DeepMind

Lars Quaedvlieg
Larsquaedvlieg@outlook.com

October 2020

Contents

Chapter 1

Introduction to Reinforcement Learning

1.0.1 The Reinforcement Learning Problem

What makes reinforcement learning different from other machine learning paradigms?

- There is no supervisor, only a **reward** signal
- Feedback is delayed, not instantaneous
- Time really matters (sequential, non i.i.d data)
- Agent's actions affect the subsequent data it receives

Rewards are scalar feedback signals. Reinforcement Learning is based on the **Reward Hypothesis**, meaning all goals can be described by the maximization of expected cumulative reward. This can be hard, since actions can have long-term consequences and reward can be delayed.

A state is **Markov**, if and only if it holds the **Markov Property**, meaning $\mathbb{P}(S_t + 1|S_t) = \mathbb{P}(S_t + 1|S_1, ..., S_t)$. This means the probability of the future states solely depends on the current state, and not on any previous states. I.e. the history is a sufficient statistic of the future.

Let S_t^a be the state of the agent at any time t and S_t^e be the state of the environment on any time t. If the environment is **fully observable**, then $S_t^a = S_t^e$. This means that the Markov property holds, so formally it is a Markov Decision Process.

However, when the environment is **partially observable**, the agent indirectly observes the environment. Now, $S_t^a \neq S_t^e$. Formally, this is called a partially observable Markov decision process (POMDP). The agent must construct it's own state representation S_t^a. For example:

- Complete history: $S_t^a = H_t$
- Beliefs of environment state: $S_t^a = (\mathbb{P}[S_t^e = s^1], ..., \mathbb{P}[S_t^e = s^n])$
- Recurrent Neural Network: $S_t^a = \sigma(S_{t-1}^a W_s + O_t W_o))$

1.0.2 Components of an RL Agent

An RL agent may include one or more of these components:

- **Policy**: agent's behaviour function
- **Value function**: how good is each state and/or action
- **Model**: agent's representation of the environment

A **policy** describes the agent's behavior. It maps states to actions. You can have deterministic ($a = \pi(s)$) and stochastic policies ($\pi(a|s) = \mathbb{P}(A_t = a|S_t = s)$). Often, π is used to denote a policy.

A **value function** is a prediction of future reward of a given state. You can use it to determine if a state is good or bad. This means you can use it to select actions. It can be computed by $v_\pi(s) = \mathbb{E}_\pi(G_t|S_t = s)$, where G_t is the **return** (or total reward). The return is defined as $G_t = R_1 + \gamma R_2 + \gamma^2 R_3 + ... = \sum_{i=t+1}^{\infty} \gamma^{i-t-1} R_i$ for some $\gamma \in [0, 1)$. This gamma is the **discount factor**, and it influences how much the future impacts return. This is useful, since it is not known if the representation of the environment is perfect. If it is not, it is not good to let the future influence the return as much as more local states. So, it is discounted.

Finally, a **model** predicts what the environment will do next. We let $P_{ss'}^a = \mathbb{P}(S_t + 1 = s'|S_t = s, A_t = a)$ and $R_s^a = \mathbb{P}(R_t + 1|S_t = s, A_t = a)$. P (**Transition model**) is the transition probability to a next state given an action, while R is the probability of obtaining a certain reward when taking an action in some state.

Category	Properties
Value based	No Policy (implicit), Value function
Policy based	Policy, No Value function
Actor Critic	Policy, Value function
Model Free	No Model
Model based	Model

Figure 1.1: Types of RL agents

RL Agents can be categorized into the categories that are listed in 1.1. These can require different approaches that will be discussed later.

There are two fundamental problems in **sequential decision making**.

- **Reinforcement Learning**
  - The environment is initially unknown
  - The agent interacts with the environment
  - The agent improves its policy
- **Planning** (e.g. deliberation, reasoning, introspection, pondering, thought, search)
  - A model of the environment is known
  - The agent performs computations with its model (without any external interaction)
  - The agent improves its policy

It is important for an agent to make a trade-off between exploration and exploitation as well. Depending on the choice in this trade-off, agents will be more or less flexible and may or may not find better actions to perform.

- **Exploration** finds more information about the environment
- **Exploitation** exploits known information to maximize reward

Finally, it is possible to differentiate between prediction and control. **Prediction** is about evaluating the future given a certain policy, while **control** is about finding the best policy to optimize the future.

Chapter 2

The Second Chapter