Deep Reinforcement Learning Notebook

Han Cheol Moon
School of Computer Science and Engineering
Nanyang Technological University
Singapore
hancheol001@edu.ntu.edu.sg

February 28, 2023

Contents

Chapter 1

Introduction

Reinforcement learning (RL) is learning what to do so as to maximize a numerical reward signal. The learner is not told which actions to take, but instaed must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards.

Important features of RL:

- Trial-and-error search
- Delayed reward
- RL uses training information that *evaluates* the actions taken rather than *instructs* by giving correct actions.
  - Evaluative feedback, not about best or worst.

1.1 Markov Chain

- Reachable: $i \to j$
- Communicate: $i \leftrightarrow j$
- Irreducible: $i \leftrightarrow j, \forall i, j$
- Absorbing state: If the only possible transition is to itself. This is also a terminal state.
- Transient state: A state s is called a transient state, if there is another state s', that is reachable from s, but not vice versa.
- Recurrent state: A state that is not transient.
- Periodic: A state is periodic if all of the paths leaving s come back after some multiple steps ($k > 1$).
  - Recurrent state is aperiodic if $k = 1$.
- Ergodicity if a Markov chain follows:

  - Irredicible
  - Recurrent
  - Aperiodic

1.2 Multi-Armed Bandits

Bandit problems are stateless. Each arm has a fixed distribution of rewards. It does not depend on which arms were pulled previously. The goal is to explore the reward distributions of all arms and then keep pulling the best one.

Markov decision processes are a temporal extension of bandit problems: pulling an arm influences the future rewards. Technically, there is a state that changes by pulling an arm. The reward distributions depend on that state.

You can view bandit problems as Markov decision processes where all states are terminal. In that case, all decision sequences have a length of 1 and subsequent pulls don't influence each other.

- When the lever of a slot machine is pulled it gives a random reward coming from a probability distribution specific to that machine.
- Although the machines look identical, their reward probability distributions are different.
- In each turn, gamblers need to decide whether to play the machine that has given the highest average reward so far, or to try another machine.

In our k-armed bandit problem, each of the k actions has an expected or mean reward given that that action is selected: let us call this the *value* of that action. We denote the action selected on time step t as A_t, and the corresponding reward as R_t. The value then of an arbitrary action a, denoted $q_*(a)$, is the expected reward given that a is selected:

$$q_*(a) = \mathbb{E}[R_t|A_t = a].$$

Since we do not know which action is the best, we have to estimate the value of actions a at time step t, $Q_t(a)$.

1.2.1 Action-value Methods

One natural way to estimate this is by averaging the rewards actually received: $Q_t(a)$ is sum of rewards when a taken prior t over number of times a taken prior to t:

$$Q_t(a) = \frac{\sum_{t=1}^{t-1} R_i \cdot \mathbb{1}_{A_i=a}}{\sum_{t=1}^{t-1} \mathbb{1}_{A_i=a}}$$

Then, the simplest action selection rule is to select one of the actions with the highest estimated value, which is a greedy action selection method represented as follows:

$$A_t = \underset{a}{\operatorname{argmax}}\, Q_t(a).$$

This approach always exploits current knowledge to maximize immediate reward; it spends no time at all sampling apparently inferior actions to see if they might really be better. A simple alternative is to behave greedily most of the time, but every once in a while, instead select randomly from among all the actions. This near greedy action selection rule is called ϵ-greedy method. In the limit as the number of steps increases, every action will be sampled an infinite number of times, which ensures all the $Q_t(a)$ converge to $q_*(a)$.

Chapter 2

Markov Decision Process

The general framework of MDPs (representing environments as MDPs) allows us to model virtually any complex sequential decision-making problem under uncertainty in a way that RL agents can interact with and learn to solve solely through experience.

Components of RL:

- An agent
- A policy
- A reward signal: what is good in an immediate sense.
- A value function: what is good in the long run.
- A model of the environment: This allows inferences to be made about how the environment will behave.

Definition 1 (Markov Property) *A state S_t is **Markov** if and only if*

$$P[S_{t+1}|S_t, A_t] = P[S_{t+1}|S_t, A_t, S_{t-1}, A_{t-1}, ...]$$

Definition 2 (Transition Function)

$$p(s', r|s, a) = P(S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a)$$

- The way the environment changes as a response to actions is referred to as the state-transition probabilities, or more simply, the transition function, and is denoted by $T(s, a, s')$.
- $\sum_{r\in\mathcal{R}} \sum_{s'\in\mathcal{S}} p(s', r|s, a) = 1, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}(s)$
- $p(s'|s, a) = P(S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a) = \sum_{r\in\mathcal{R}} p(s', r|s, a)$

Definition 3 (Reward Function)

$$\begin{aligned} r(s, a) &= \mathbb{E}[R_t|S_{t-1} = s, A_{t-1} = a] \\ &= \sum_{r\in\mathcal{R}} r \sum_{s'\in\mathcal{S}} p(s', r|s, a) \end{aligned}$$

- The expected reward function is defined as a function that takes in a state-action pair.
- It is the expectation of reward at time step t, given the state-action pair in the previous time step.
- It can also be defined as a function that takes a full transition tuple s, a, s'.

$$\begin{aligned} r(s, a, s') &= \mathbb{E}[R_t | S_{t-1} = s, A_{t-1} = a, S_t = s] \\ &= \sum_{r \in \mathcal{R}} \frac{p(s', r|s, a)}{p(s'|s, a)} \end{aligned}$$

- **Reward Hypothesis**: That all of what we mean by goals and purposes can be well thought of as athe maximization of the expected value of the cumulative sum of a received scalar signal (called reward).

Definition 4 (Discount Factor, γ)

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-1} R_t$$

- The sum of all rewards obtained during the course of an episode is referred to as the return, G_t.
- Episodic tasks: $G_t = R_{t+1} + R_{t+2} + \cdots + R_T$.
  - $G_t = R_{t+1} + \gamma G_{t+1}$
- Continuing tasks: $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$.
  - $\gamma = 0$: myopic evaluation
  - $\gamma = 1$: far-sighted evaluation
  - Uncertainty about the future that may not be fully observed
  - Mathematically convenient to discount rewards.
  - Avoid infinite returns in cyclic Markov processes.

2.1 Policies and Value Functions

2.1.1 The State-Value Functions

Almost all reinforcement learning algorithms involve estimating *value functions*-functions of states (or of state-action pairs) that estimate how good it is for the agent to be in a given state (or how good it is to perform a given action in a given state). The notion of "how good" here is defined in terms of future rewards that can be expected, or, to be precise, in terms of expected return. Of course, the rewards the agent can expect to receive in the futhre depend on what actions it will take. Accordingly, value functions are defined with respect to particular ways of action, called policies.

Formally, a policy is a mapping from states to probablities of selecting each possible action. It can be defined

$$\pi(a|s)$$

Definition 5 (The State-Value Function, V) *The state value function $v(s)$ for policy π of an Markov Reward Process is the expected return starting from state s*

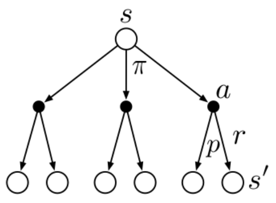
$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} r^k R_{t+1+k}\middle|S_t = s\right], \quad \forall s \in \mathcal{S}$$

- The value of a state s is the expection over policy π.
- Policies are universal plans, which provides all possible plans for all states.
  - Plans are not enough in stochastic environments.
  - Policy can be stochastic or deterministic.
  - A policy is a function that prescribes actions to take for a given non-terminal state.
- If we are given a policy and the MDP, we should be able to calculate the expected return starting from every single state.

- *Bellman equation*

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma v(s_{t+1})|S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')] \end{aligned}$$

- Starting from state s, the root node at the top, the agent could take any of some set of actions – three are shown in the diagram – based on its policy π. From each of these, the environment could respond with one of several next states, s' (two are shown in the figure), along with a reward, r, depending on its dynamics given by the function p. The Bellman equation averages over all the possibilities, weighting each by its probability of occuring. **It states that the value of the start state must equal (discounted) the value of the expected next state, plus the reward expected along the way**.



- The derivation details are given in Appendix A.1

2.1.2 The Action-Value Function

- Another critical question that we often need to ask is not merely about the value of a state but the value of taking action a at a state s.
- Which action is better under each policy?
- The action-value function, also known as Q-function or $Q^\pi(s,a)$, captures precisely this.

  – The expected return if the agent follows policy π after taking action a in state s.

Definition 6 (The Action-Value Function, Q) *The action-value function $q_\pi(s,a)$ for policy π is the expected return starting from state s, tacking action a under policy π*

$$q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\middle|S_t = s, A_t = a\right]$$

- The Bellman equation for action values is given by

$$\begin{aligned} q_\pi(s,a) &= \mathbb{E}_\pi[G_t|S_t = s, A_t = a] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s, A_t = a] \\ &= \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')] \end{aligned}$$

- Notice that we do not weight over actions because we are interested only in a specific action.
- The state-value function can be expressed by using the action-value function as follows:
- The derivation is given in Appendix A.1

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] \\ &= \sum_{g_t} g_t \cdot P(G_t|S_t = s) \\ &= \sum_{g_t} g_t \sum_{a} P(G_t|S_t = s, A_t = a)P(A_t = a|S_t = s) \\ &= \sum_{a}\sum_{g_t} g_t P(G_t|S_t = s, A_t = a)P(A_t = a|S_t = s) \\ &= \sum_{a} q_\pi(s,a)\pi(a|s) \end{aligned}$$

Note that the expectation is parameerized π as written in $\mathbb{E}_\pi$. We can also prove it by the *Law of Total Expectation*,

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] \\ &= \mathbb{E}[\mathbb{E}_\pi[G_t|S_t = s, A_t = a]] \\ &= \sum_{a} \mathbb{E}_\pi[G_t|S_t = s, A_t = a]P(A_t = a|S_t = s) \\ &= \sum_{a} q_\pi(s,a)\pi(a|s) \end{aligned}$$

An intuitive explantation of this derivation is that the expectation depends on an action $a \sim \pi(a|s)$. We want to estimate the expected total return by the sampled action (this is because the total return is the function of a, implicitly). Then we need to introduce an action variable a and its probability in the expression as the second line of the equation.

2.1.3 The Action-Advantage Function

Definition 7 (The Action-Advantage Function, A)

$$a_\pi(s,a) = q_\pi(s,a) - v_\pi(s)$$

- The advantage function describes how much better it is to take action a instead of following policy π. In other words, the advantage of choosing action a over the default action.

2.2 Optimality

Solving a reinforcement learning task means, roughly, finding a policy that achieves a lot of reward over the long run. For finite MDPs, we can precisely define an optimal policy in the following way. Value functions define a partial ordering over policies. A policy π is defined to be better than or equal to a policy π' if its expected return is greater than or equal to that of π' for all states. In other words, $\pi \geq \pi'$ if and only if $v_\pi(s) \geq v_{\pi'}(s)$ for all $s \in \mathcal{S}$. There is always at least one policy that is better than or equal to all other policies. This is an *optimal policy.* The optimal state-value function, denoted v_* can be defined as

Definition 8 (Optimal State-Value Function) *The optimal state-value function $v_*(s)$ is the maximum value over all policies*

$$v_*(s) = \max_\pi v_\pi(s), \quad \forall s \in \mathcal{S}.$$

- The optimal state-value function can be obtained as follows:

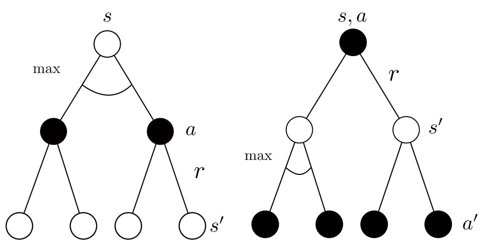
$$\begin{aligned} v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\ &= \max_a \mathbb{E}_{\pi_*}[G_t | S_t = s, A_t = a] \\ &= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] \\ &= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a] \\ &= \max_a \sum_{r,s'} p(s', r | s, a) \Big[r + \gamma v_*(s') \Big] \end{aligned}$$

Definition 9 (Optimal Action-Value Function) *The optimal action-value function $q_*(s,a)$ is the maximum value over all policies*

$$q_*(s,a) = \max_\pi q_\pi(s,a), \quad \forall s \in \mathcal{S} \textit{ and } a \in \mathcal{A}.$$

- The optimal action-value function can be obtained as follows:

$$\begin{aligned} q_*(s,a) &= \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S + t = s, A_t = a] \\ &= \sum_{s',r} p(s', r | s, a)[r + \gamma \max_a q_*(s', a')] \end{aligned}$$



- The optimal value function specifies the best possible performance in the MDP.
- The MDP is solved when we know the optimal value function

Theorem 1 (Optimal Policy Theorem)

$$\pi \geq \pi' \quad \textit{if} \quad v_\pi(s) \geq v_{\pi'}(s), \forall s$$

For any Markov Decision Process:

- *There exists an optimal policy π_* that is better than or equal to all other policies, $\pi_* \geq \pi, \forall \pi$*
- *All optimal policies achieve the optimal value function, $v_{\pi_*}(s) = v_*(s)$*
- *All optimal policies achieve the optimal action-value function, $q_{\pi_*}(s,a) = q_*(s,a)$*

An optimal policy can be found by maximizing over $q_*(s,a)$,

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \operatorname{argmax}_{a \in \mathcal{A}} q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

- There is always a deterministic optimal policy for any MDP
- If we know $q_*(s,a)$, we immediately have the optimal policy
  - Q-learning: learns Q values first
  - Policy gradient: learns optimal policy without learning Q values

Bibliography

[1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[2] Rachit Singh. . `https://rachitsingh.com/elbo_surgery/`, 2017. Online; accessed 29 January 2014.

[3] Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567, 2018.

Appendix

A.1 Bellman Equation

Bellman equation can be derived as follows:

$$
\begin{aligned}
v_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1}|S_t = s] + \mathbb{E}_\pi[\gamma G_{t+1}|S_t = s], \quad \text{By Linearity of Expectation.} \\
&= \sum_{r_{t+1}} r_{t+1} P(R_{t+1}|S_t = s) + \mathbb{E}_\pi[\gamma G_{t+1}|S_t = s] \\
&= \sum_{r_{t+1}} r_{t+1} \sum_a P(R_{t+1}|S_t = s, A_t = a) P(A_t = a|S_t = s) + \mathbb{E}_\pi[\gamma G_{t+1}|S_t = s] \\
&= \sum_a \sum_{r_{t+1}} r_{t+1} \sum_{s'} P(R_{t+1}, S_{t+1} = s'|S_t = s, A_t = a) P(A_t = a|S_t = s) + \mathbb{E}_\pi[\gamma G_{t+1}|S_t = s] \\
&= \sum_a \sum_r r \sum_{s'} P(s', r|s, a)\pi(a|s) + \mathbb{E}_\pi[\gamma G_{t+1}|S_t = s] \\
&= \sum_a \sum_{s'} \sum_r r P(s', r|s, a)\pi(a|s) + \mathbb{E}_\pi[\gamma G_{t+1}|S_t = s] \\
&= \sum_a \sum_{s'} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} \pi(a|s) + \gamma \mathbb{E}_\pi[G_{t+1}|S_t = s] \\
&= \sum_a \sum_{s'} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} \pi(a|s) + \gamma \sum_a \mathbb{E}_\pi[G_{t+1}|S_t = s, A_t = a] P(A_t|S_t) \\
&= \sum_a \pi(a|s) \left[\sum_{s'} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} + \gamma \mathbb{E}_\pi[G_{t+1}|S_t = s, A_t = a] \right] \\
&= \sum_a \pi(a|s) \left[\sum_{s'} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} + \gamma \sum_{g_{t+1}} g_{t+1} P(G_{t+1}|S_t = s, A_t = a) \right] \\
&= \sum_a \pi(a|s) \left[\sum_{s'} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} + \gamma \sum_{g_{t+1}} g_{t+1} \frac{P(G_{t+1}, S_t = s, A_t = a)}{P(S_t = s, A_t = a)} \right] \\
&= \sum_a \pi(a|s) \left[\sum_{s'} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} + \gamma \sum_{g_{t+1}} g_{t+1} \frac{\sum_{s'} P(G_{t+1}, S_t = s, S_{t+1} = s', A_t = a)}{P(S_t = s, A_t = a)} \right] \\
&= \sum_a \pi(a|s) \left[\sum_{s'} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} + \gamma \sum_{g_{t+1}} g_{t+1} \frac{\sum_{s'} P(G_{t+1}|s, s', a) P(s, s', a)}{P(s, a)} \right] \\
&= \sum_a \pi(a|s) \left[\sum_{s'} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} + \gamma \sum_{g_{t+1}} g_{t+1} \sum_{s'} P(G_{t+1}|s, s', a) P(s'|s, a) \right] \\
&= \sum_a \pi(a|s) \left[\sum_{s'} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} + \gamma \sum_{s'} P(s'|s, a) \sum_{g_{t+1}} g_{t+1} P(G_{t+1}|s') \right] \quad \text{by Markov Property} \\
&= \sum_a \pi(a|s) \left[\sum_{s'} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} + \gamma \sum_{s'} \mathcal{P}^a_{ss'} v_\pi(s') \right] \\
&= \sum_a \pi(a|s) \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma v_\pi(s') \right] \\
&= \sum_a \pi(a|s) \sum_{r,s'} p(s', r|s, a) \Big[r + \gamma v_\pi(s') \Big]
\end{aligned}
$$

Or simply,

$$\begin{aligned} v_\pi(s) &= \sum_a \pi(a|s) q(s,a) \\ &= \sum_a \pi(a|s) \sum_{r,s'} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big] \end{aligned}$$

Reference

Similarly,

$$\begin{aligned} q_\pi(s,a) &= \mathbb{E}_\pi[G_t|S_t = s, A_t = a] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s, A_t = a] \\ &= \mathbb{E}_\pi[R_{t+1}|S_t = s, A_t = a] + \gamma \mathbb{E}_\pi[G_{t+1}|S_t = s, A_t = a] \\ &= \sum_r r p(r|s,a) + \gamma \mathbb{E}_\pi[G_{t+1}|S_t = s, A_t = a] \\ &= \sum_r r \sum_{s'} p(s',r|s,a) + \gamma \mathbb{E}_\pi[G_{t+1}|S_t = s, A_t = a] \\ &= \sum_{s',r} r p(s',r|s,a) + \gamma \mathbb{E}[\mathbb{E}_\pi[G_{t+1}|S_t = s, A_t = a, R_{t+1}, S_{t+1}]] \quad \text{By Law of Total Expectation.} \\ &= \sum_{s',r} r p(s',r|s,a) + \gamma \sum_{s',r} \mathbb{E}_\pi[G_{t+1}|S_t = s, A_t = a, R_{t+1} = r, S_{t+1} = s'] p(s',r|s,a) \\ &= \sum_{s',r} p(s',r|s,a)[r + \gamma \mathbb{E}_\pi[G_{t+1}|S_t = s, A_t = a, R_{t+1} = r, S_{t+1} = s'] \\ &= \sum_{s',r} p(s',r|s,a)[r + \gamma \mathbb{E}_\pi[G_{t+1}|S_{t+1} = s'] \quad \text{By Markov Property.} \\ &= \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')] \end{aligned}$$