Deep Reinforcement Learning Notebook

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

February 5, 2023

Contents

Chapter 1

Introduction

1.1 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.

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)

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

- 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 $T(s, a, s')$.
- $\sum_{s' \in S} p(s'|s, a) = 1, \forall s \in S, \forall a \in A(s)$

Definition 3 (Reward Function)

$$r(s, a) = \mathbb{E}[R_t|S_{t-1} = s, A_{t-1} = a]$$

- The 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'.

$$r(s, a, s') = \mathbb{E}[R_t|S_{t-1} = s, A_{t-1} = a, S_t = s]$$

- $R_t \in \mathcal{R} \in \mathbb{R}$

Definition 4 (Discount Factor, γ)

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

- $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$
- $G_t = R_{t+1} + \gamma G_{t+1}$
- $\gamma = 0$: 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.

1.1.1 The State-Value Function

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

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

- 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.

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

- *Bellman equation*

1.1.2 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 $q_\pi(s,a)$ is the expected return starting from state s, tacking action a under policy π*

$$q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$$

- 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.

1.1.3 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.

1.1.4 Optimality

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

$$v_*(s) = \max_\pi v_\pi(s)$$

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

$$v_*(s) = \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma v_*(s')]$$

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

$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

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

$$q_*(s,a) = \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma \max_a q_*(s',a')]$$

- 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 $q_*(s,a)$,

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

- 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:

$$
\begin{aligned}
v_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s\right] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...|S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + ...)|S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1}|S_t = s] + \gamma \mathbb{E}_\pi[G_{t+1}|S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1}|S_t = s] + \gamma \mathbb{E}_\pi\left[\mathbb{E}_\pi[G_{t+1}|S_{t+1} = s'] \middle| S_t = s_t\right] \\
&= \mathbb{E}_\pi[R_{t+1}|S_t = s] + \gamma \mathbb{E}_\pi\left[v(s_{t+1}) \middle| S_t = s_t\right] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma v(s_{t+1})|S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')]
\end{aligned}
$$

The expectation here describes what we expect the return to be if we continue from state s following policy π. The expectation can be written explicitly by summing over all possible actions and all possible returned states. The next two equations can help us make the next step.

Reference