

Deep Q-Learning in a nutshell

Bar Hilleli

barhilleli5@gmail.com

Oct 15, 2017



Markov Decision Process - MDP

Definition

The MDP framework consists of four elements: $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$

- \mathcal{S} is a discrete set of states
- \mathcal{A} is a discrete set of actions
- \mathcal{R} is a reward model $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is a transition probability matrix
$$\mathcal{P}_{ss'}^a = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

Policy

Definition

A (deterministic) **policy** is a mapping from a given world state s , to a desired action a :

$$\pi : \mathcal{S} \rightarrow \mathcal{A}$$

Objective

Given an MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ we wish to find a policy so as to **maximize** the **long term sum of the discounted immediate rewards**

$$\sum_{t=0}^{\infty} \gamma^t r_t$$

where γ is called the *discount factor* in range $[0, 1]$.

State-Value Function

Definition

The **state-value function** for policy π , denoted by $V^\pi(s, a)$, is the **expected return when starting in state s and following policy π thereafter**, and is formally defined as,

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right]$$

Action-Value Function

Definition

Similarly, The **action-value function** for policy π , denoted by $Q^\pi(s, a)$, is the **expected reward of taking action a in state s and following policy π thereafter**, and is formally defined as,

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right]$$

Optimal Policy

Considering finite MDPs, an **optimal policy**, π^* , is better than all other policies in the sense that its expected return is greater than or equal to that of all other policies for all states. Meaning,

$$V^{\pi^*}(s) \geq V^{\pi}(s)$$

for all $s \in \mathcal{S}, \pi \in \Pi$.

Optimal State-Value Function

Definition

The value functions of π^* is called **optimal state-value function**, denoted by \mathbf{V}^* and defined as,

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

for all $s \in \mathcal{S}$.

Optimal Action-Value Function

Definition

Similarly the **optimal action-value function**, denoted Q^* , is defined as

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$$

for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$.

Q^* in terms of V^*

$Q^*(s, a)$ gives the expected return taking action a in state s and thereafter following π^* , therefore we can write,

$$Q^*(s, a) = \mathbb{E}_{s_{t+1}} [r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a]$$

Bellman Optimality Equation

Since the value of a state s under π^* must be equal to the expected return of the best action from that state we have that,

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a)$$

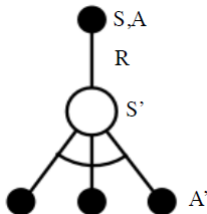
Therefore, we can write the **Bellman optimality equation** for Q^* ,

$$Q^*(s, a) = \mathbb{E}_{s_{t+1}} \left[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \mid s_t = s, a_t = a \right]$$

Q-Learning

- The **Q-learning** algorithm aims to find the optimal action-value function.
- The **Q-learning update** is:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right)$$



Approximating $Q^*(s, a)$ using a Neural Network

- Dealing with a very large state space of images, we **approximate** the optimal action-value function using a **deep Q-network (DQN)** with **parameters θ** :

$$Q_{\theta}(s, a) \approx Q^*(s, a).$$

- A deep Q-network is a neural network that for a given state outputs a vector of action values.

