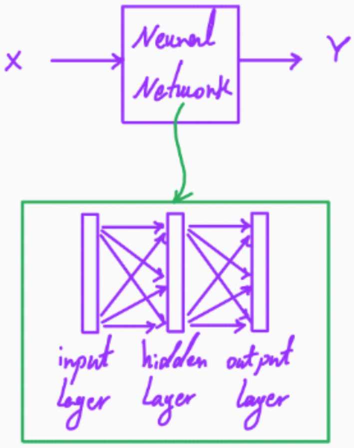
So far, we have been dealing with planning settings where the full environment model (transition function P and reward function R) is known.

- value iteration
- policy iteration
- generalized policy iteration
- linear programming

Now, we move to learning settings where the environment model is not fully known.

- model-based RL $\quad \hat{P} \approx P$, $\hat{R} \approx R$
- model-free RL
  - Monte-Carlo
  - Temporal difference
  - TD(λ)

Function Approximation



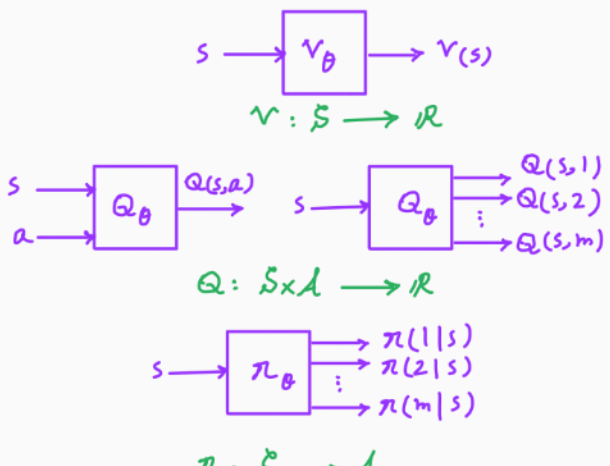
Functions of interest in RL

value functions V, Q / V^*, Q^*

Policy π / π^*

Environment model $\hat{P}$, $\hat{R}$ (in model-based RL)

RL with NN approximators (Deep RL)



Monte Carlo Method

Rather than having the full environment model (transition function P and reward function R), Monte Carlo methods only require experiences, i.e., sample trajectories (sequence of states, actions, rewards).

$\boxed{\text{Model\_free method}}$

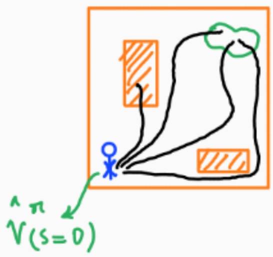
Monte Carlo: Using random sampling to perform a computational task, such as estimation and optimization.

Assumption: The setting is episodic, i.e., the interactions happen in episodes of finite length.

[The updates of MC-based methods happen after each episode
$\longrightarrow$ not fully online]

MC for policy evaluation: Given a stationary policy π,

we want to compute



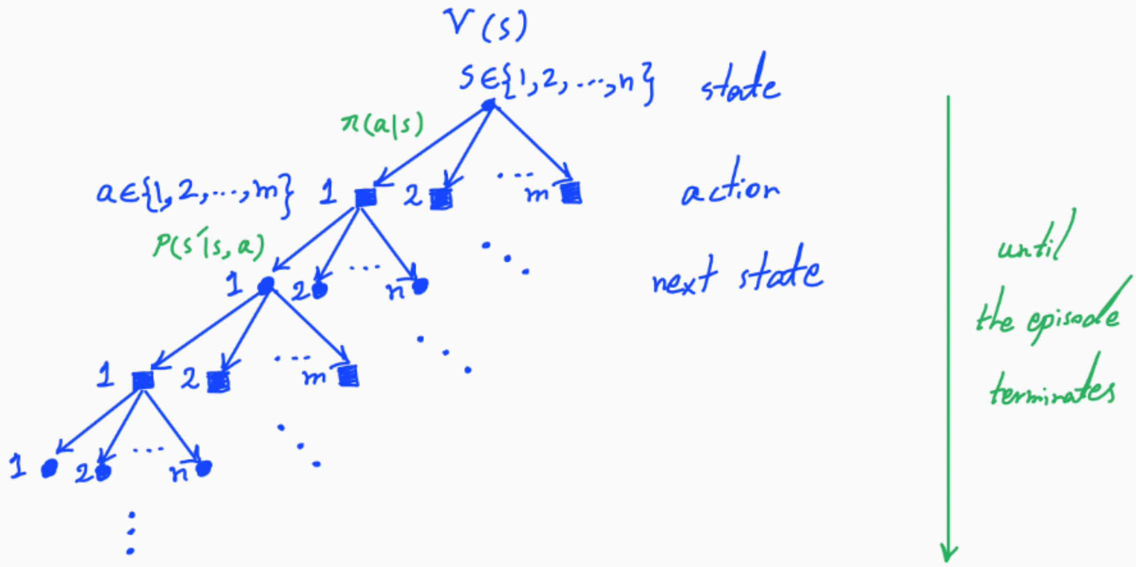
$$V^{\pi}(s) = \mathbb{E}_{\substack{A_t \sim \pi \\ S_{t+1} \sim P(\cdot \mid S_t, A_t)}} \Big[\overbrace{\sum_{t=0}^{T} \gamma^t R(S_t, A_t)}^{\text{return } G} \;\Big|\; S_0 = s \Big]$$

Idea: Estimate the expected return using the empirical mean of the returns of sample trajectories.

$$\boxed{\mathbb{E}_{P_X}[X] \approx \hat{\mathbb{E}}[X] = \frac{1}{N} \sum_{i=1}^{N} x_i}$$

X can be return G

x_i sampled from P_X



Let $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_{|\mathcal{T}|}\}$ denote a set of trajectories sampled from s

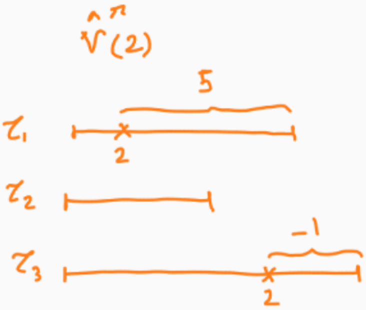
$$\longrightarrow V^{\pi}(s) = \mathbb{E}_{\substack{A_t \sim \pi \\ S_{t+1} \sim P(\cdot \mid S_t, A_t)}} \Big[\sum_{t=0}^{T} \gamma^t R(S_t, A_t) \;\Big|\; S_0 = s \Big] \approx \frac{1}{|\mathcal{T}|} \Big(\sum_{\tau \in \mathcal{T}} \underbrace{\sum_{t=0}^{|\tau_i|} \gamma^t R(s_t^i, a_t^i)}_{\text{return for trajectory } \tau_i} \Big) = \hat{V}^{\pi}(s)$$

$\tau \in \mathcal{T}$: i.i.d. samples

First-visit MC method

Iterative method:

- Initialize $\hat{V}^\pi$
- Initialize $\mathcal{G}(s) = \emptyset$ for all $s \in \mathcal{S}$
- Repeat
  - Generate a sample trajectory τ using π
  - For $s \in \mathcal{S}$ s.t. $s \in \tau$:



$\hat{V}^\pi(2) \approx \frac{5 + (-1)}{2}$

    - t_s = first time s is visited
    - $G = \sum_{t=t_s}^{|\tau|} \gamma^{t-t_s} R(s_t, a_t)$
    - $\mathcal{G}(s) \leftarrow \mathcal{G}(s) \cup \{G\}$
    - $N(s) \leftarrow N(s) + 1$ $\qquad N(s) = |\mathcal{G}(s)|$
    - $\hat{V}^\pi(s) = \frac{1}{N(s)} \sum_{G \in \mathcal{G}(s)} G$

$$\hat{\mu}_{k+1} = \frac{\sum_{i=1}^{k+1} x_i}{k+1} = \frac{\sum_{i=1}^{k} x_i + x_{k+1}}{k+1} = \frac{k\hat{\mu}_k + x_{k+1}}{k+1}$$

$$= \frac{k}{k+1}\hat{\mu}_k + \frac{1}{k+1}x_{k+1} = \hat{\mu}_k + \frac{1}{k+1}(x_{k+1} - \hat{\mu}_k)$$

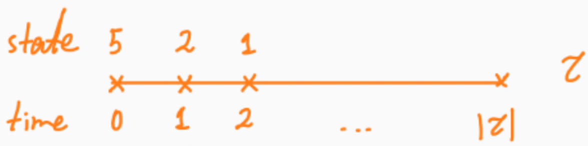
Incremental computation of mean

Note: By Law of Large numbers, $\hat{V}^\pi(s) \xrightarrow[N(s)\to\infty]{} V^\pi(s)$.

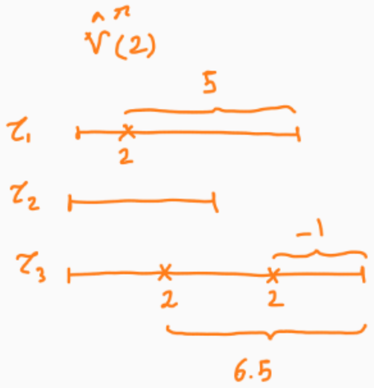
Every-visit MC method

Iterative method:

- Initialize $\hat{v}^{\pi}$
- Initialize $\mathcal{G}(s) = \emptyset$ for all $s \in \mathcal{S}$
- Repeat
    - Generate a sample trajectory τ using π
    - For $s \in \mathcal{S}$ s.t. $s \in \tau$:



        - For every time s is visited:
            - t_s = time s is visited
            - $G = \sum_{t=t_s}^{|\tau|} \gamma^{t-t_s} R(s_t, a_t)$
            - $\mathcal{G}(s) \leftarrow \mathcal{G}(s) \cup \{G\}$
            - $N(s) \leftarrow N(s) + 1$
    - $\hat{V}^{\pi}(s) = \frac{1}{N(s)} \sum_{G \in \mathcal{G}(s)} G$



$\hat{v}^{\pi}(2) \approx \frac{5 + 6.5 + (-1)}{3}$

Note: It can be shown that $\hat{v}^{\pi}(s) \xrightarrow[N(s) \to \infty]{} v^{\pi}(s)$.

MC for policy optimization: We want to find an (approximately) optimal policy π^* through iterative policy evaluation and policy improvement.

$$\begin{tikzcd} \pi \arrow[r, bend left, "\text{evaluation}"] & V^\pi \arrow[l, bend left, "\text{improvement}"] \end{tikzcd}$$

Idea: generalized policy iteration

- Evaluate the policy by estimating the Q function using the empirical mean of the returns of the sample trajectories.
- Improve the policy using the estimated Q function in a greedy manner.

Estimating V or Q?

If V known $\longrightarrow$ $\pi = \underset{a \in A}{\operatorname{argmax}} \Big[\underbrace{R(s,a) + \gamma \, \mathbb{E}_{s' \sim P(\cdot \mid s,a)} [V(s')]}_{Q(s,a)}\Big]$

unknown

If Q known $\longrightarrow$ $\pi = \underset{a \in A}{\operatorname{argmax}} \; Q(s,a)$

$\longrightarrow$ we need to estimate Q instead of V.

MC method with exploring starts

Iterative method:

$$\pi \underset{\text{improvement}}{\overset{\text{evaluation}}{\rightleftarrows}} Q^{\pi}$$

- Initialize $\hat{Q}$
- Initialize π
- Initialize $\mathcal{G}(s,a) = \emptyset$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$

Exploration
- Choose $d \in \underbrace{\Delta(\mathcal{S} \times \mathcal{A})}_{\substack{\text{set of all possible} \\ \text{probability distributions over } \mathcal{S}\times\mathcal{A}}}$ s.t. $d(s,a) > 0$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$
- Repeat
  - Sample $s_0 \in \mathcal{S}$ and $a_0 \in \mathcal{A}$ according to d
  - Generate a sample trajectory τ using π starting from (s_0, a_0)

Policy evaluation
  - For $(s,a) \in \mathcal{S} \times \mathcal{A}$ s.t. $(s,a) \in \tau$:
    - $t_{s,a} =$ first time (s,a) is visited
    - $G = \sum_{t=t_{s,a}}^{|\tau|} \gamma^{t-t_{s,a}} R(s_t, a_t)$
    - $\mathcal{G}(s,a) \leftarrow \mathcal{G}(s,a) \cup \{G\}$ $\qquad \mathcal{G}(s=1, a=3) = \{-5, 7, 1.2, \dots\}$
    - $N(s,a) \leftarrow N(s,a) + 1$ $\qquad N(s,a) = |\mathcal{G}(s,a)|$
    - $\hat{Q}(s,a) = \frac{1}{N(s,a)} \sum_{G \in \mathcal{G}(s,a)} G$

Policy improvement

- For $s \in \mathcal{S}$ s.t. $s \in \tau$:

  - $\pi(s) = \underset{a \in \mathcal{A}}{\arg\max}\ \hat{Q}(s, a)$

Note: In real-world settings, it may not be possible to start from any state-action pairs.

$\longrightarrow$ Can we instead encourage continuous exploration?

Idea:

- Use a soft (stochastic) policy $\pi(a|s)$ such that it chooses all (possible) actions at each state with a non-zero probability, i.e.,

$$\pi(a|s) > 0 \quad \forall s \in \mathcal{S},\ \forall a \in \mathcal{A}.$$

- As the estimates improve over time, reduce exploration.

MC method with ε-greedy exploration

Iterative method:

- Initialize $\hat{Q}$
- Initialize $\pi(a|s)$ s.t. $\pi(a|s) > 0$ for all $s \in \mathcal{S}, a \in \mathcal{A}$
- Initialize $\mathcal{G}(s,a) = \emptyset$ for all $s \in \mathcal{S}, a \in \mathcal{A}$
- Repeat
    - Generate a sample trajectory τ using π

policy evaluation

- For $(s,a) \in \mathcal{S} \times \mathcal{A}$ s.t. $(s,a) \in \tau$:
    - $t_{s,a} =$ first time (s,a) is visited
    - $G = \sum_{t=t_{s,a}}^{|\tau|} \gamma^{t-t_{s,a}} R(s_t, a_t)$
    - $\mathcal{G}(s,a) \leftarrow \mathcal{G}(s,a) \cup \{G\}$
    - $N(s,a) \leftarrow N(s,a) + 1$
    - $\hat{Q}(s,a) = \frac{1}{N(s,a)} \sum_{G \in \mathcal{G}(s,a)} G$

Policy improvement

- For $s \in \mathcal{S}$ s.t. $s \in \tau$:
  - $a^* = \underset{a \in \mathcal{A}}{\arg\max}\ \hat{Q}(s,a)$
  - For $a \in \mathcal{A}$:
    - $\pi(a|s) = \begin{cases} 1 - \varepsilon + \dfrac{\varepsilon}{|\mathcal{A}|} & \text{if } a = a^* \\ \dfrac{\varepsilon}{|\mathcal{A}|} & \text{otherwise} \end{cases}$

ε-greedy

Exploration

Decaying exploration: