Action affects world state

- choose classes
  - action : take a class
  - rewards : enjoyment
  - state : satisfy more prereqs

  e.g. a class may be good to take later,
  bad to take now

- robotics
  - action : motor
  - reward : complete a task
  - state : position of robot

  state transitions are noisy/random

- video games

Markov Decision Process (MDP)

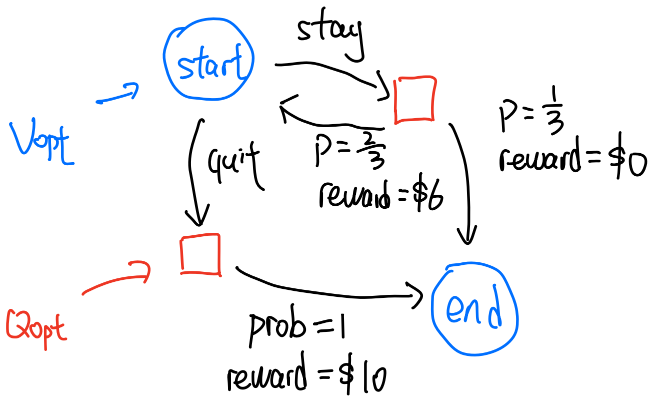
Formal description of a world with state actions, rewards, etc.

At each time :

- player can stay or quit
- if quit : get \$10 and game ends
- if stay :

  prob 1/3, get \$0 and end

  prob 2/3, get \$6 and continuous



Formal MDP:

- set of states (e.g. possible positions of robots)
- starting state : S_{start}
- Actions (s): possible actions at states s
- $T(s, a, s')$: prob of going from state s to s' after taking action a.

  (e.g. $T(\text{start}, \text{stay}, \text{end}) = \frac{1}{3}$
- $R(s, a, s')$: reward of going from s to s' taking action a.
- IsEnd (s): is this an end state

Give MDP, what is optimal agent behavior?

policy: A strategy that agent can use

$$\pi(\underbrace{s}_{\text{current state}}) \rightarrow \underbrace{\text{action} \in \text{Action}(s)}_{\text{chosen action}}$$

The value $V_\pi(s)$ for policy π is expected discounted sum of rewards starting from state s, applying policy π.

Discounting: future rewards are less valuable than current rewards

- at any timestep you could die

we introduce a $\underbrace{\text{discount factor}}_{\text{prob of survival at each timestep}}$ $\gamma \in [0, 1]$

If we get a sequence of rewards $r_1, r_2, \dots$

Discounted sum of rewards $= r_1 + \gamma r_2 + \gamma^2 r_3 + \dots$

The optimal value $V_{OPT}(s)$ is maximum possible value at state s for any policy.

V_{opt} is characterized by recursive formulas:

$$V_{opt}(s) = \begin{cases} 0 & \text{if } \text{IsEnd}(s) \\ \max_{a \in \text{Actions}(s)} \underbrace{Q_{opt}(s,a)}_{\text{expected optimal value after taking action } a \text{ in state } s} & \text{else} \end{cases}$$

$$Q_{opt}(s,a) = \sum_{s'} \underbrace{T(s,a,s')}_{\text{prob of transion to } s'} [\underbrace{R(s,a,s')}_{\text{reward now}} + \underbrace{\gamma V_{opt}(s')}_{\text{discounted future rewards at } s'}]$$

optimal policy: $\pi^*(s) = \arg\max_{a \in \text{Action}(s)} Q_{opt}(s,a)$

if we can estimate $Q_{opt}(s,a)$ for all s, a, we can find π^*.

RL:

- believe the world is a MDP
- don't know $T(s,a,s')$ and $R(s,a,s')$
- has to try many actions in many states

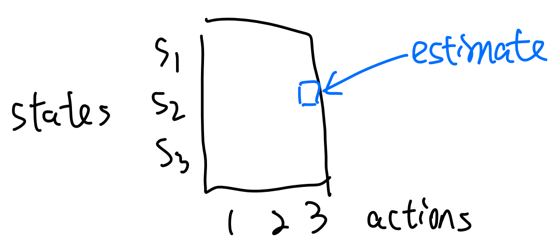
For episode = 1, 2, 3 ...

$s_1 \leftarrow s_{start}$

for $t = 1, 2, \ldots$

- agent chooses action $a_t = \underbrace{\pi_{act}}_{\text{policy we act with during learning}}(s_t)$
- agent receives:
  - reward r_t
  - new state s_{t+1}
- Update agent's parameters

Q-Learning: directly learn $Q_{opt}(s,a)$



$\hat{Q}(s,a)$

$$Q_{opt}(s,a) = \sum_{s'} T(s,a,s') \cdot \left[R(s,a,s') + \gamma V(s') \right]$$

$$V_{opt}(s') = \begin{cases} 0 & \text{if IsEnd}(s') \\ \max\limits_{a \in \text{Action}(s')} Q_{opt}(s',a) \end{cases}$$

We have data

$$\underbrace{s_1, a_1, r_1, s_2}_{1 \text{ example}}, a_2, r_2, \ldots$$

$$s_2, a_2, r_2, \ldots \quad \text{(2 example)}$$

Every time we see (s, a, r, s'):

"nudge" $\hat{Q}(s,a)$ based on this observation

$$\underbrace{\hat{Q}(s,a)}_{\text{estimate of } Q_{opt}(s,a)} \leftarrow (1-\underbrace{\eta}_{\text{learning rate e.g. } 0.1}) \hat{Q}(s,a) + \eta \, \underbrace{(\underbrace{r}_{\text{observed reward}} + \gamma \hat{V}(\underbrace{s'}_{\text{observed next state}}))}_{\text{one sample for } Q_{opt}(s,a)}$$

where $\hat{V}(s') = \begin{cases} 0 & \text{if IsEnd}(s') \\ \max\limits_{a \in \text{Actions}(s')} \hat{Q}(s',a) \end{cases}$

What π_{act} to act with?

$$\underbrace{\pi(s) = \arg\max_{a \in \text{Actions}(s)} \hat{Q}(s,a)}_{\text{pure exploitation strategy}}$$

solution: ε - greedy

policy during learning $\pi_{Act} = \begin{cases} \text{with prob } 1-\varepsilon, \text{ do } \arg\max_a \hat{Q}(s,a) \\ \text{with prob } \varepsilon, \text{ do random action} \in \text{Actions}(s) \end{cases}$

- Training : Q-Learning with $\varepsilon = 0.1$

  balance exploration v.s. exploitation

- Testing : act with $\varepsilon = 0$