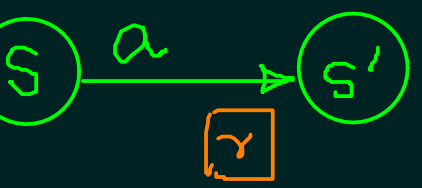
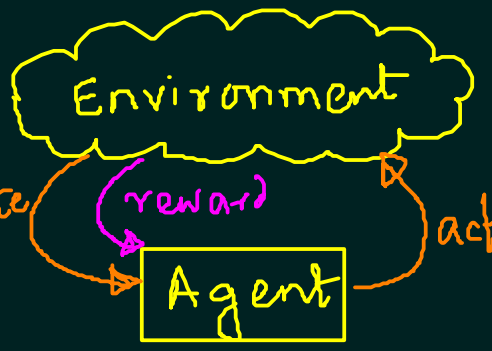
▷ Reinforcement Learning :

SUMMARY





↳ **Task**: Learn an optimal policy $\pi : S \to A$

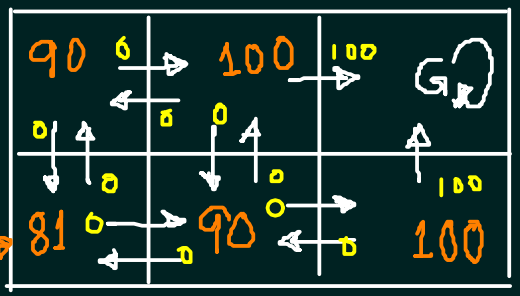
$$\delta(s,a) = s' \qquad r(s,a)$$ } MDP

$$\pi : s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_4} s_4 \to \cdots$$

$$V^{\pi}(s_1) = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots = r_1 + \gamma V^{\pi}(\delta(s_1, a_1))$$ ← value function

$$\pi^*(s) = \underset{a}{\operatorname{argmax}} \left[r(s,a) + \gamma V^*(\delta(s,a)) \right]$$ ← Bellman Equation

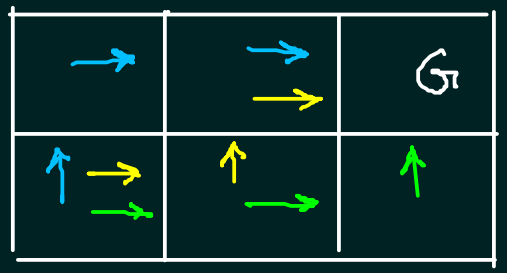
↑ immediate reward ; ↑ discount ; ↑ future reward



Value function compute

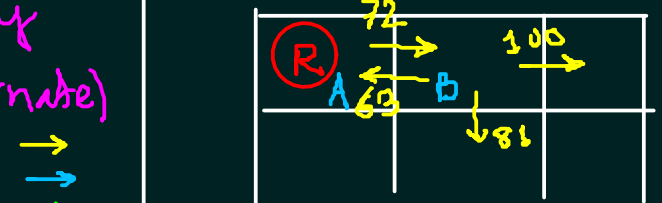
$V^*(s) = 0 + 0.9 \times 90 = 81$

Inferrible to precompute in BFS manner



Policy (alternate)

③



▷ **Q - Learning** :

$$\pi^*(s) = \underset{a}{\operatorname{argmax}}\, Q(s,a)$$

$$V^*(s) = \max_a Q(s,a)$$

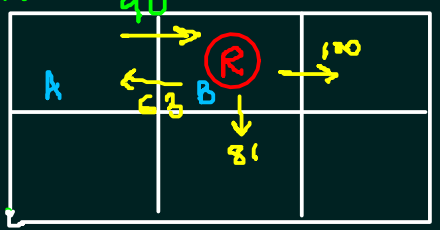
$$Q(s,a) \leftarrow r(s,a) + \gamma V^*(\delta(s,a))$$
$$= r(s,a) + \gamma \max_{a'} Q(\delta(s,a), a')$$

$$\Rightarrow \hat{Q}(s,a) = r(s,a) + \gamma \hat{Q}(s',a') \quad , \quad s' = \delta(s,a)$$

R →

$\hat{Q}(A, \rightarrow) = 0 + 0.9 \max\{63, 81, 100\}$

update

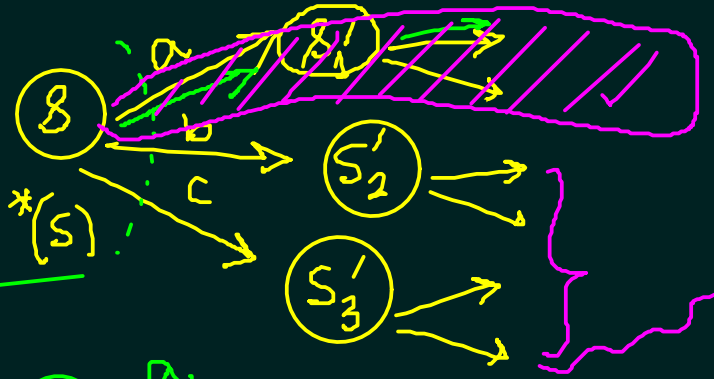


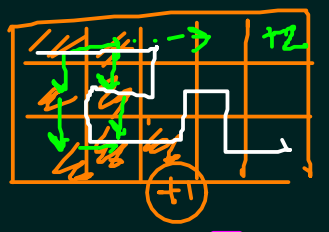
Exploration vs. Exploitation:

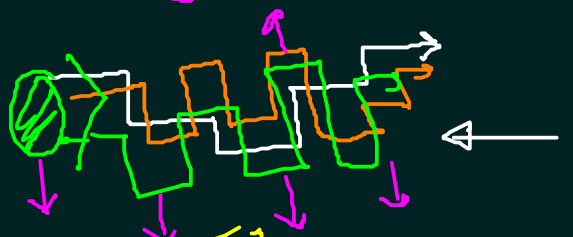
↳ Get stuck local maxima

Solution: $P(a|s) = \frac{1}{Z} e^{\hat{Q}(s,a)/T}$ → $\pi^*(s)$: max

↳ alternative explore of a'

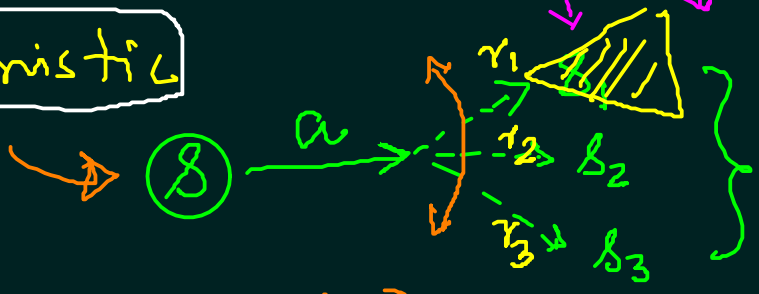






— Actions are not-deterministic

$P(s'|a,s)$ → value



$$V^{\pi}(s_t) = E\left[\sum_{i=0}^{\infty} \gamma^i r_{t+1}\right]$$ ← expected sum of cumulative rewards

$$Q(s,a) = E\left[r(s,a) + \gamma V^*(\delta(s,a))\right]$$

$$= E[r(s,a)] + \gamma E[V^*(\delta(s,a))]$$

$$= E[r(s,a)] + \gamma \sum_{s'} P(s'|s,a) \cdot V^*(s')$$

$P(s'|s,a)$ — expected Q function update rule

$$\hat{Q}(s,a) \leftarrow E[r(s,a)] + \gamma \sum_{s'} P(s'|s,a) \max_{a'} \hat{Q}(s',a')$$

$P(s' \mid s,a) = ?$ $\quad \frac{\#\ s' \leftarrow s,a}{\#(s,a)}$ $\quad s' \sim P_{sa}$ MLE

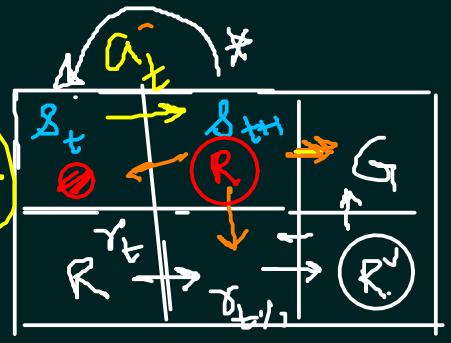
$s_1^{(1)} \xrightarrow{a_1^{(1)}} s_2^{(1)} \xrightarrow{a_2^{(1)}} s_3^{(1)} \rightarrow \cdots$ $\quad r_1^{(1)}, r_2^{(1)}$

$s_1^{(2)} \xrightarrow{a_1^{(2)}} s_2^{(2)} \xrightarrow{a_2} \cdots$ $\quad r_1^{(2)}, r_2$

Training Example

▷ Ramifications :

$\hat{Q}^{(1)}(s_t,a_t) \leftarrow r_t + \gamma \max_a \hat{Q}(s_{t+1},a)$



$\hat{Q}^{(2)}(s_t,a_t) \leftarrow r_t + \gamma r_{t+1} + \gamma^2 \max_a \hat{Q}(s_{t+2},a)$ ← 2-step

$\hat{Q}^{(n)}(s_t,a_t) \leftarrow r_t + \gamma r_{t+1} + \cdots + \gamma^{n-1} r_{t+n-1} + \gamma^n \max_a \hat{Q}(s_{t+n},a)$

↖ n-steps

$0 \le \lambda \le 1$ ←

(Sutton)

$\hat{Q}^{\lambda}(s_t,a_t) = (1-\lambda)\left[\hat{Q}^{(1)}(s_t,a_t) + \underbrace{\lambda}_{w_2}(1-\lambda)\hat{Q}^{(2)}(s_t,a_t) + \underbrace{\lambda^2}_{w_3}(1-\lambda)^2 \hat{Q}^{(3)}(s_t,a_t) + \cdots\right]$ ← new

w_i

$\Rightarrow Q^{\lambda}(s_t,a_t) = r_t + \gamma\left[(1-\lambda)\max_a \hat{Q}(s_t,a_t) + \lambda Q^{\lambda}(s_{t+n},a_{t+1})\right]$

Temporal Difference RL

Monte-Carlo Policy Evaluation:

(1) $s_1^{(1)} \xrightarrow{a_0} s_2^{(1)} \xrightarrow{r_1} s_3^{(1)} \rightarrow \cdots$

(2) $s_1^{(2)} \xrightarrow{b_0} s_3^{(2)} \xrightarrow{b_1} s_4 \rightarrow \cdots$

(k) ⋮

$\pi^*(s) \rightarrow$ policy

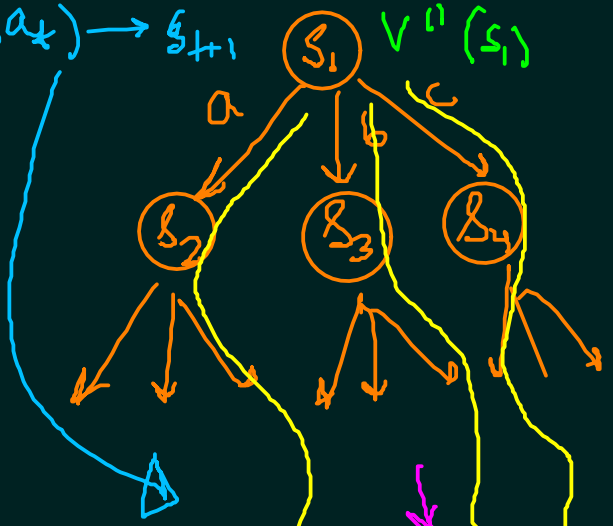
$\delta(s_t, a_t) \rightarrow s_{t+1}$

$V^{\pi}(s_1)$

$\pi(s_1) = V_1$

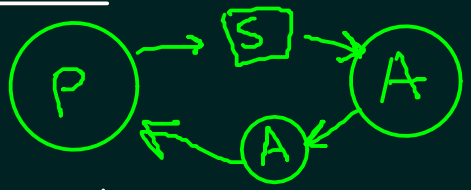
$V^{\pi} = V_2$

$V^{\pi}(s_1) = \Sigma$ avg of V_i's



$s_{t+1} = s_t + \beta a_t$

Control:



(x, y, θ) ← state

$(\dot{x}, \dot{y}, \dot{\theta})$ ← action

(x, y, z)

$\phi, \psi, \theta)$ state

$(\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\psi}, \dot{\theta})$ action

$\mathbb{R}^n$ [Curse of Dimensionality]

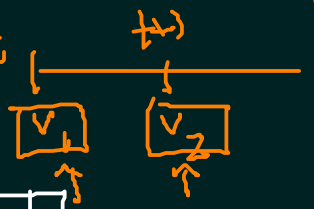
$n \rightarrow$ (k values) $\rightarrow$ k^n

Reward Hacking
Reward Shaping
Safety Shield

$\mathcal{S}_2$

$\mathcal{S}_1$

Q
$e^{\overline{T}_P}$



sampling

discretize

$\mathcal{S}_1 \rightarrow \uparrow$

$\mathcal{S}_2 \rightarrow \downarrow$