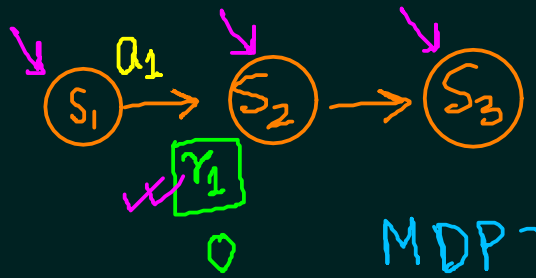
- Supervised Learning: $\langle x_1, y_1 \rangle \dots \langle x_N, y_n \rangle$ → feedback immediate
- Unsupervised Learning: $\langle x_1, * \rangle \dots \langle x_N, * \rangle$ → NO feedback
- Reinforcement Learning
  - Training Data { $\langle x_1, - \rangle \dots \langle W, L, D, +1, -1, 0 \rangle$ feedback delayed } Ex: Chess
    - +1 Win
    - −1 Loss
    - 0 D
  - Sequence of steps → → → ---- → goal (+1 / −1)
    - (policy)

$f: X \to Y$; $h \approx f$

TE for SL: $\langle x_1, y_1 \rangle \dots \langle x_N, y_N \rangle$

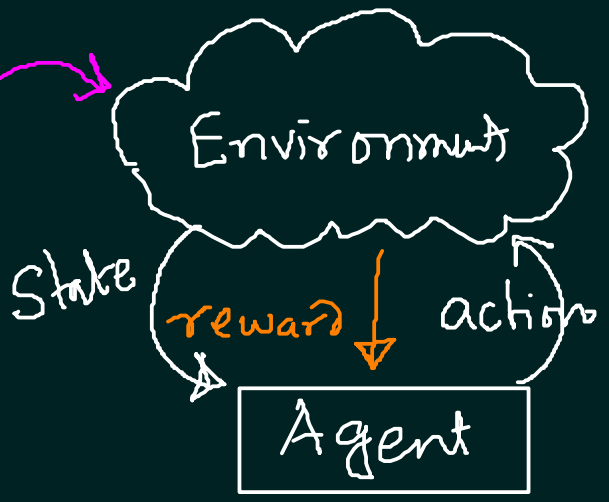
TE for RL: $\langle s_1, a_1 \rangle \dots \langle s_N, a_N \rangle$; r_1 ; r_i ; r_N



MDP

$P(S_t \mid S_{t-1}, a_{t-1})$

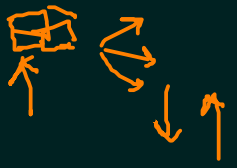
$P(r_t \mid S_{t-1}, a_{t-1})$



Learn a policy $\pi : S \to A$

Challenges: ① outcome of action uncertain

Ex: Backgammon



② Perfect sense of environment — Noise → unknown; Non-det → changing

③ Reward is delayed

④ Reward is stochastic.

⑤ How much do you train?

Task $\pi : S \to A$

Cumulative Reward

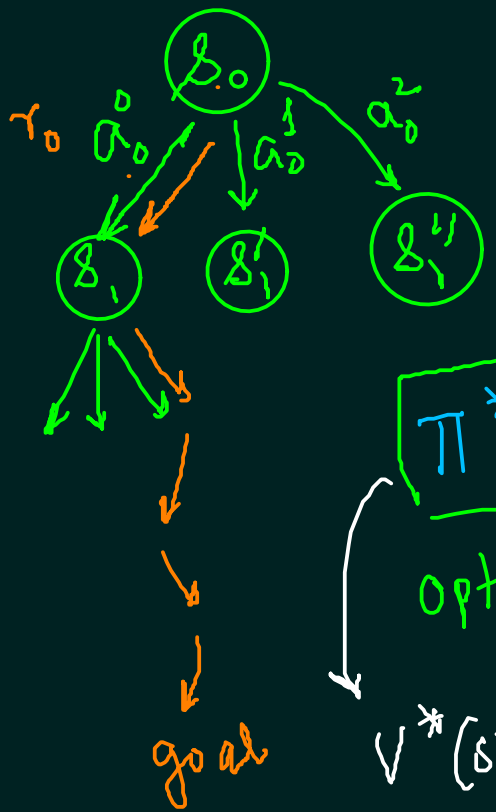
$$V^{\pi}(s_i) = r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots = \sum_{i=0}^{\infty} \gamma^i r_i$$

$0 \le \gamma < 1$

value function

$\pi \to (a_0 \to a_1 \to \cdots \to a_n)$

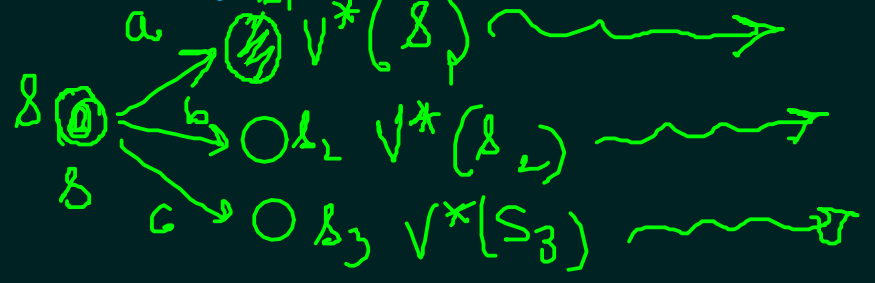
→ How to design rewards?

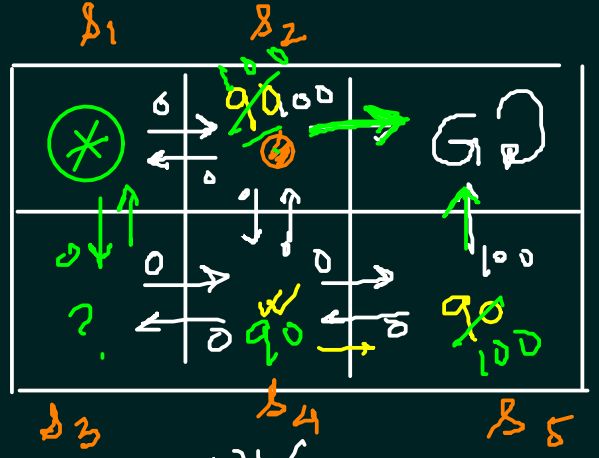


$$\pi^*(s) = \underset{a}{\operatorname{argmax}} \left[r(s,a) + \gamma V^*(\delta(s,a)) \right]$$

optimal policy

$V^*(s)$





$V^*(s)$

$V^*(s_2) = 100$

(Value Iteration
Policy Iteration)

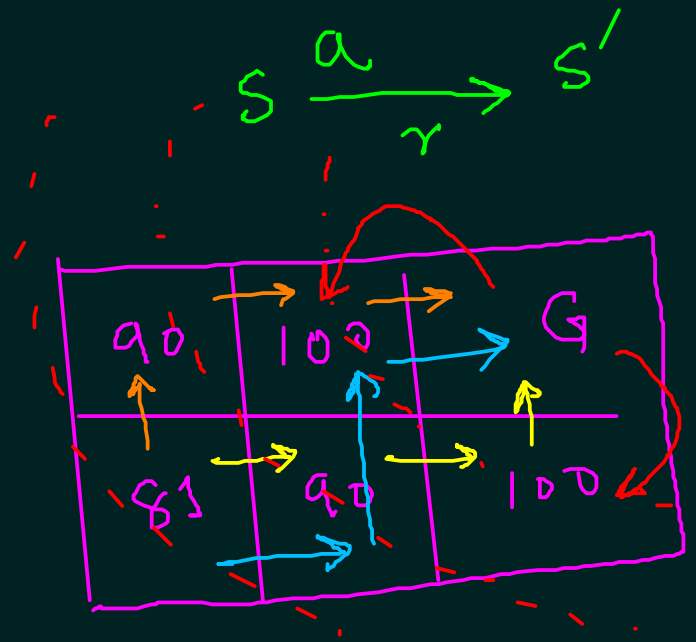
$V^*(s_4) = 0 + 0.9 \times \max\{0, 100, 0\} = 90$

$r(s_2, a)$ $\quad V^*(s_1) = 0 + 0.9 \times 100$

$= 90$

$V^*(s_3) = 0 + 0.9 \times 90$

$= 81$



OR

Alt. Best policy

Learn

$\langle s_1, r_1 \rangle \rightarrow$

$r(s,a)$ } all known

$\delta(s,a)$ }

$V^*(s)$

MDP

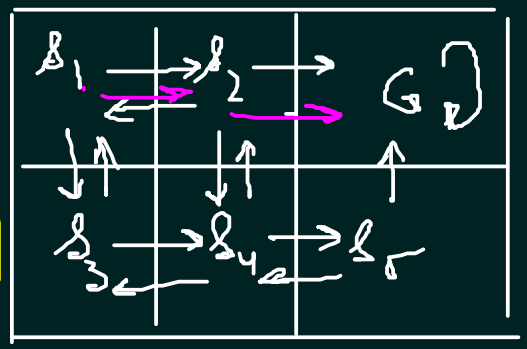
PO MDP

action not clear.

Q-learning

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

Bellman Eqn



$$V(s_1) = r(s_1,a) + \gamma V(s_2) + \gamma^2 V(G)$$

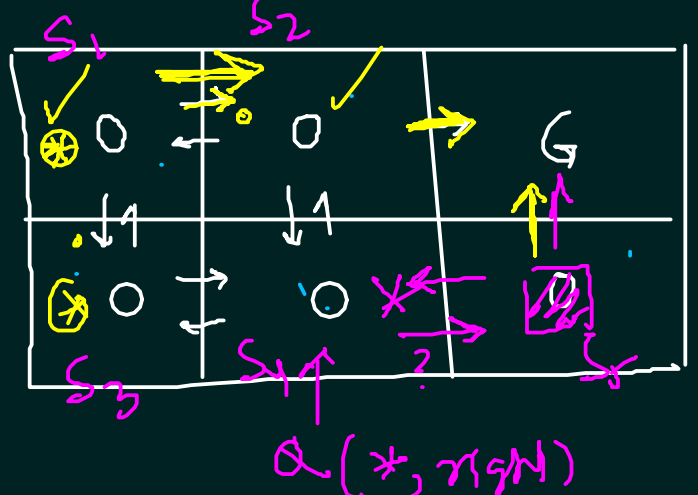
$$V(s_2) = \text{---}$$

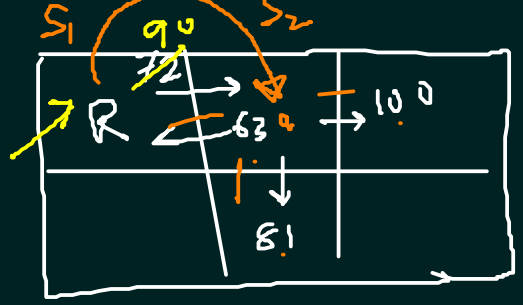
$$\begin{bmatrix} V(s_1) \\ \vdots \\ V(s_n) \end{bmatrix} = \begin{bmatrix} R \\ \vdots \\ R \end{bmatrix} + \gamma \begin{bmatrix} V \\ \vdots \\ V \end{bmatrix}$$

$Q(s,a)$ ← best action at state s

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

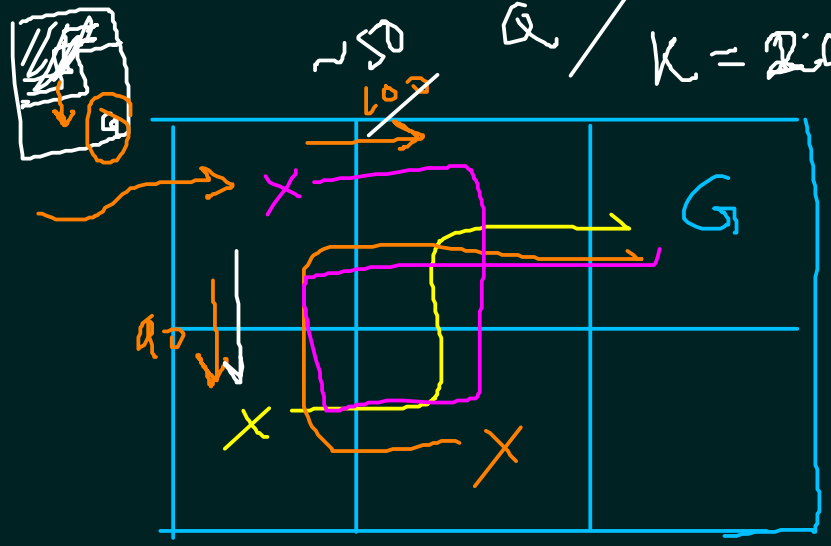
?





$$\hat{Q}(s_1, \text{right}) \leftarrow r + \gamma \max_{a'} \hat{Q}(s_2, a')$$

$$= 0 + 0.9 \times 100 = 90$$



Q / K = 2:01

Convergence for
1. det. Act
2. $|r(s,a)| < c$
3. Infinitely often visit

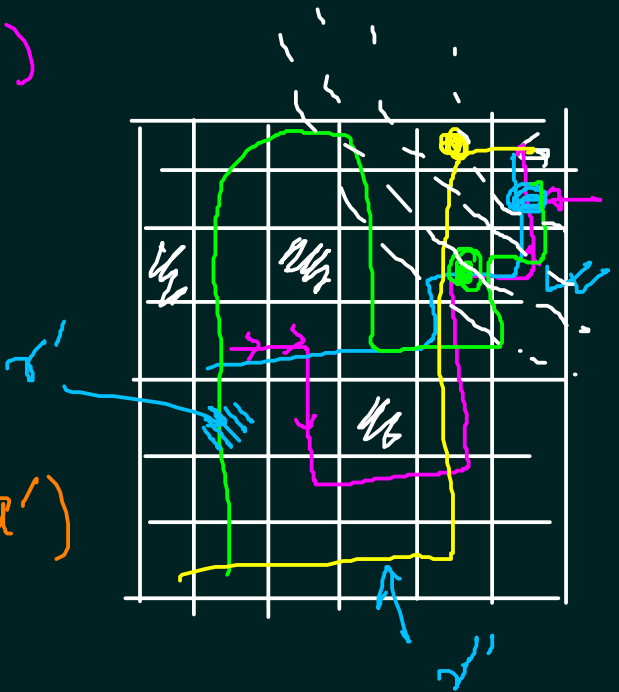
$\hat{Q}(s,a)$ $\forall s,a$ $\rightarrow Q(s,a)$

locally optimal policy π^* NN

Theorem:

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

Dynamic Prog.



$$\hat{Q}(s,a) = r(s,a) + \gamma \max_{a'} (Q[s',a'])$$

$\delta(s,a) = s'$

Exploration vs Exploitation