Q-learning

(*и разные трюки)

Подготовил: Грузицкий Андрей
БПМИ211

Вспомним

*дальше будет меньше формул, не пугайтесь

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$
$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}.$$

$$v_\pi(s) \doteq \mathbb{E}_\pi \left[G_t \,|\, S_t = s \right]$$

$$q_\pi(s,a) = \mathbb{E}_\pi \left[G_t \,|\, S_t = s, A_t = a \right] = \sum_{r,\, s'} p(r, s' \,|\, s, a) \left[r + \gamma v_\pi(s') \right]$$

Q*(s,a) - сейчас совершили действие a, а потом выбираем действия с **наибольшим** q

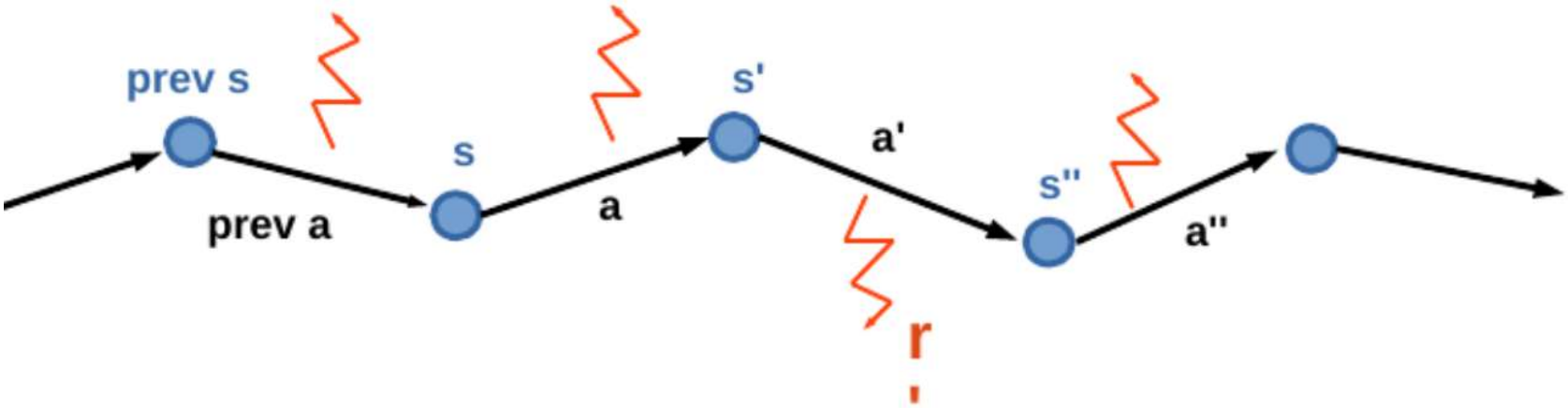
Model-free reinforcement learning

Теперь у нас нет $P(s',r|s,a)$!!!

Model based: можем планировать наперёд

Model free: только пробовать

Учимся на траекториях

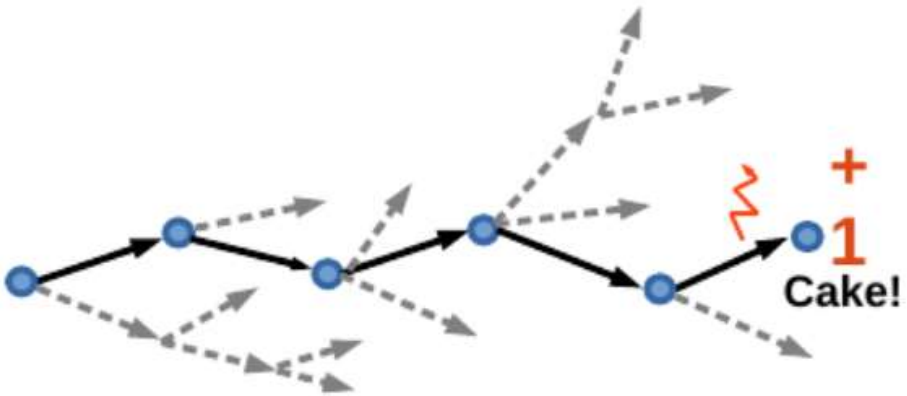


Теперь V(s) - бесполезно, т.к. не знаем P(s’|s, a)

Учим Q(s, a) - берём argmax по a => победа

Monte-carlo

- Averages Q over sampled paths



Требует очень много сессий!

Temporal Difference

- Uses recurrent formula for Q

Меньше сэмплов. Обучаемся даже не доиграв одной игры.

Temporal difference

Можем обновлять значения Q итеративно:

$$Q(s_t, a_t) \leftarrow \underset{r_t, s_{t+1}}{E}\ r_t + \gamma \cdot max_{a'} Q(s_{t+1}, a')$$

Temporal difference

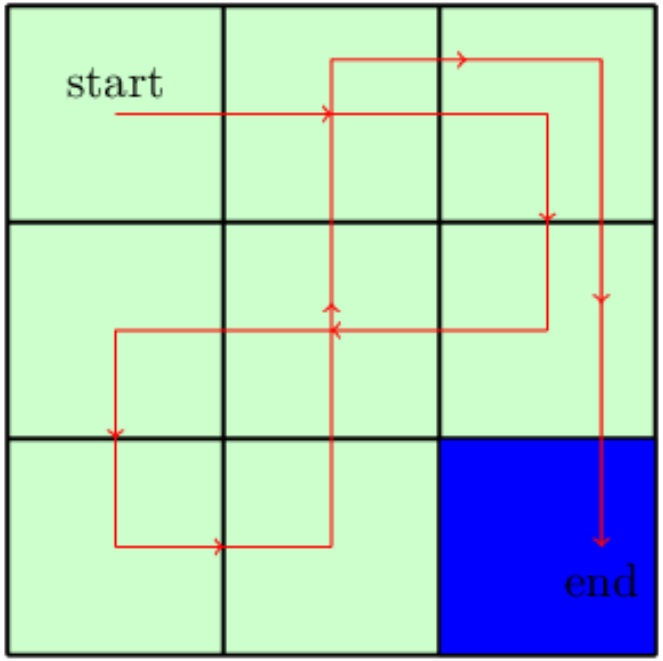
- Приближаем сэмплами:

$$\underset{r_t, s_{t+1}}{E}\ r_t + \gamma \cdot max_{a'} Q(s_{t+1}, a') \approx \frac{1}{N} \sum_i r_i + \gamma \cdot max_{a'} Q(s_i^{next}, a')$$

- Пользуемся скользящим средним. На одном шаге используем только один объект!!!

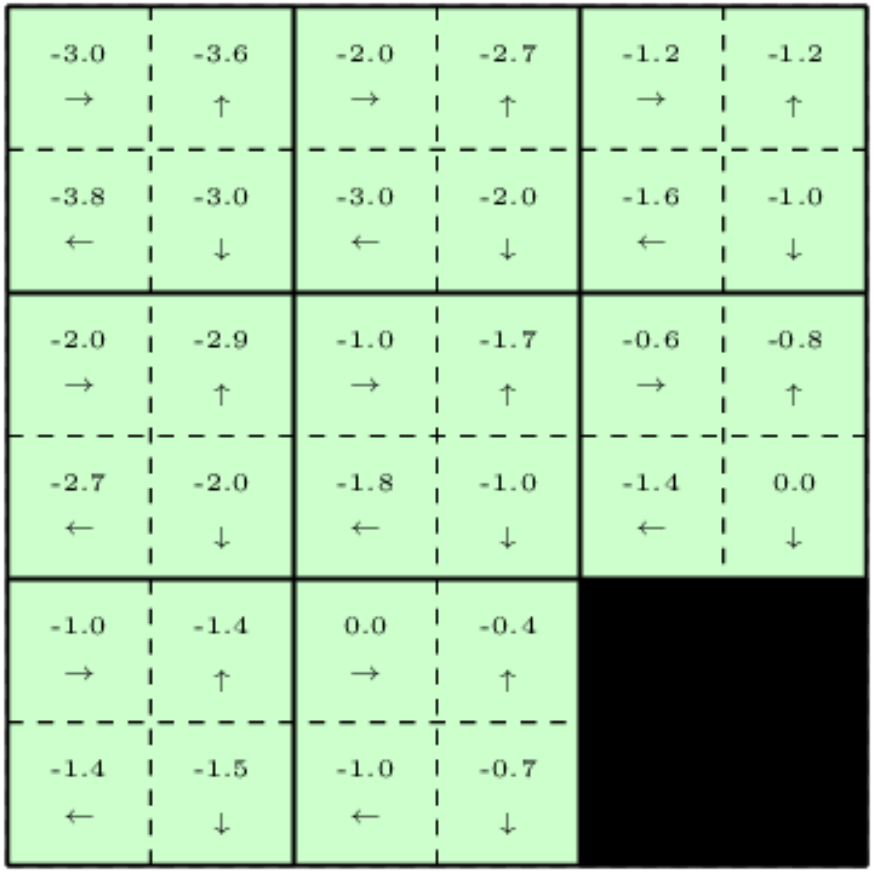
$$Q(s_t, a_t) \leftarrow \alpha \cdot (r_t + \gamma \cdot max_{a'} Q(s_{t+1}, a')) + (1 - \alpha) Q(s_t, a_t)$$

Q-learning

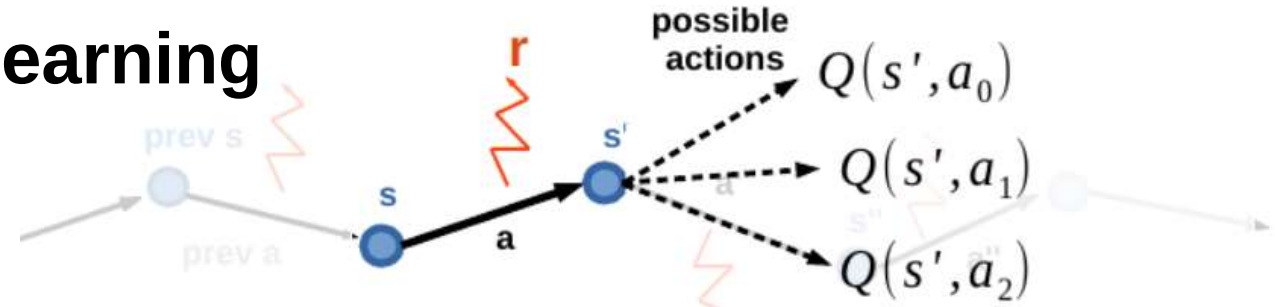


-0.1 →	0 ↑	-0.1 →	0 ↑	0 →	0 ↑
0 ←	0 ↓	0 ←	0 ↓	0 ←	-0.1 ↓
0 →	0 ↑	0 →	-0.1 ↑	0 →	0 ↑
0 ←	-0.1 ↓	-0.1 ←	0 ↓	-0.1 ←	0 ↓
-0.1 →	0 ↑	0 →	-0.1 ↑		
0 ←	0 ↓	0 ←	0 ↓		

Q-learning


-3.0 → -3.6 ↑ -2.0 → -2.7 ↑ -1.2 → -1.2 ↑
-3.8 ← -3.0 ↓ -3.0 ← -2.0 ↓ -1.6 ← -1.0 ↓
-2.0 → -2.9 ↑ -1.0 → -1.7 ↑ -0.6 → -0.8 ↑
-2.7 ← -2.0 ↓ -1.8 ← -1.0 ↓ -1.4 ← 0.0 ↓
-1.0 → -1.4 ↑ 0.0 → -0.4 ↑
-1.4 ← -1.5 ↓ -1.0 ← -0.7 ↓

Q-learning



Инициализируем Q(s, a) нулями

- Loop:

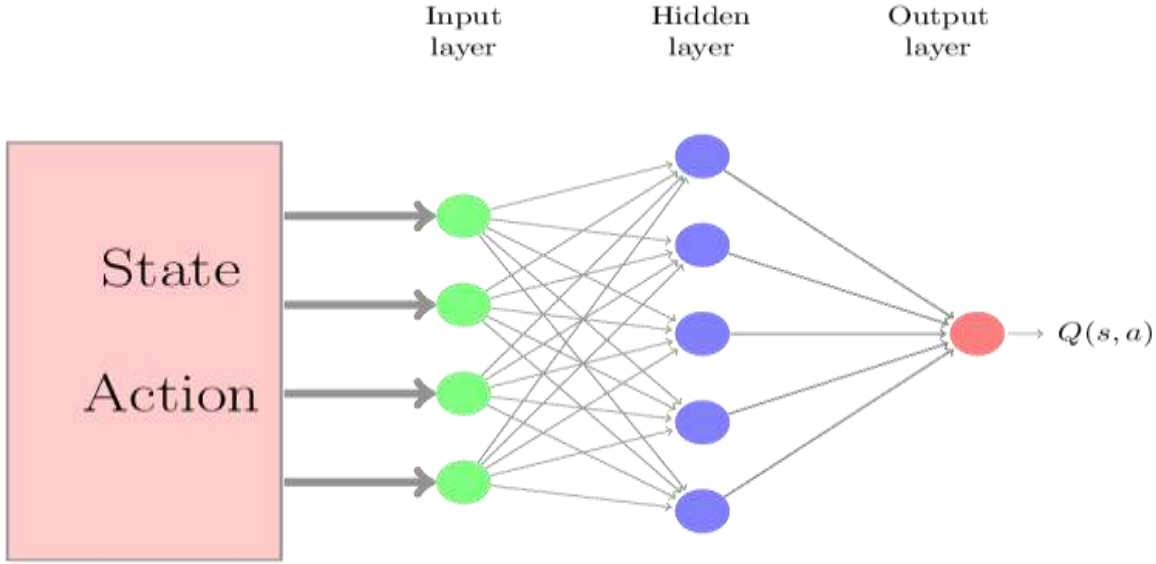
- Из среды получаем значения <s, a, r, s'>

- Считаем $\hat{Q}(s,a) = r(s,a) + \gamma \max Q(s', a_i)$

- Обновляем $Q(s,a) \leftarrow \alpha \cdot \hat{Q}(s,a) + (1-\alpha) Q(s,a)$

Непрерывный случай (таблица со значениями Q не спасёт)

DQN:



$$w = \operatorname{argmin}\left(f^w(s,a) - \left(r + \max_b(f^w(s',b))\right)\right)^2$$

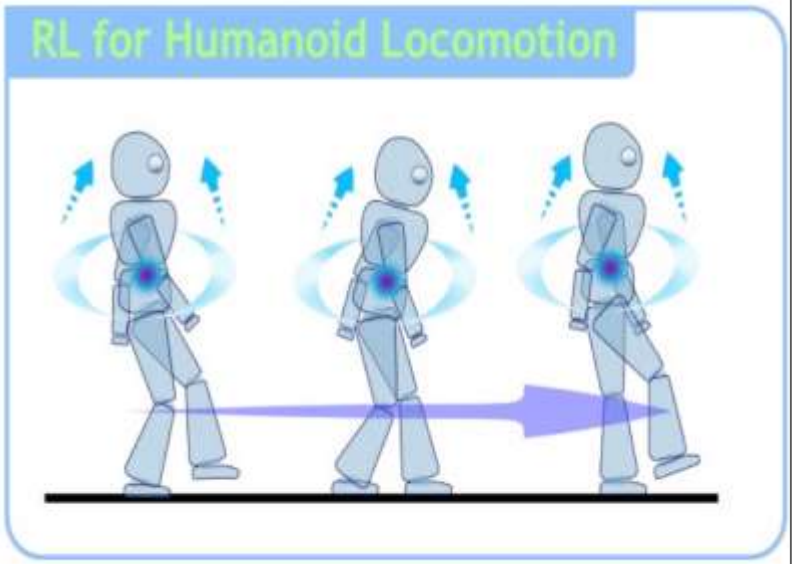
Непрерывный случай

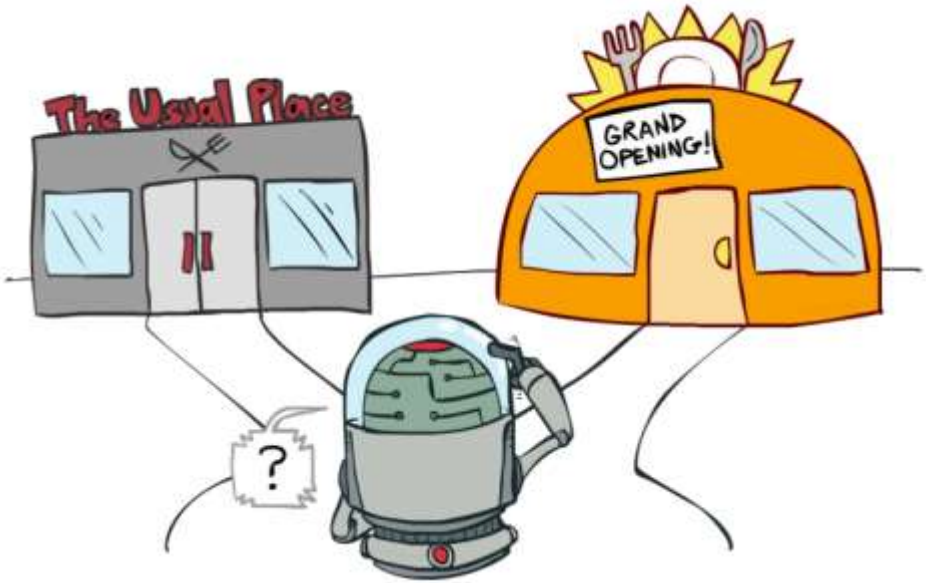
Можем использовать MSE:

$$w = \operatorname{argmin}\left(f^w(s,a) - \left(r + \max_b(f^w(s',b)\right)\right)^2$$

$$L=\left(Q\left(s_t,a_t\right)-\left[r+\gamma\cdot max_{a'}Q\left(s_{t+1},a'\right)\right]\right)^2$$

Exploration Vs Exploitation:





Exploration Vs Exploitation

- ε-greedy

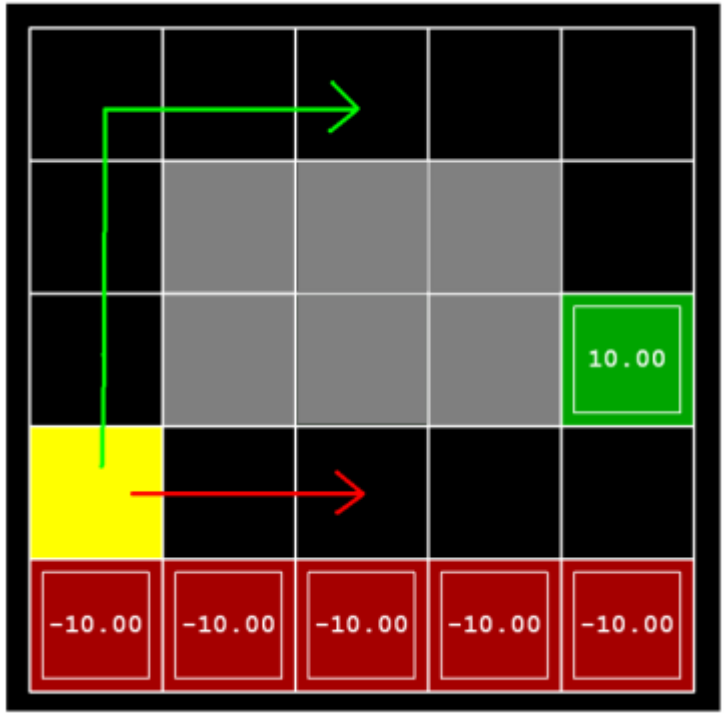
  С вероятностью ε выбираем случайное действие, иначе берём оптимальное

- Softmax

  Выбираем действие пропорционально значению softmax Q-value нормированного на тау

$$\pi(a|s) = softmax\left(\frac{Q(s,a)}{\tau}\right)$$

Cliff problem:



- Q-learning

$$\gamma = 0.99 \quad \epsilon = 0.1$$

- не хотим падать в лаву!!!

Cliff problem:

$$Q(s_t, a_t) \leftarrow \alpha \cdot \hat{Q}(s_t, a_t) + (1-\alpha) Q(s_t, a_t)$$

"better Q(s,a)"

Q-learning

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

SARSA

$$\hat{Q}(s,a) = r(s,a) + \gamma \cdot \mathop{E}_{a' \sim \pi(a'|s')} Q(s', a')$$

On-policy Vs Off-policy

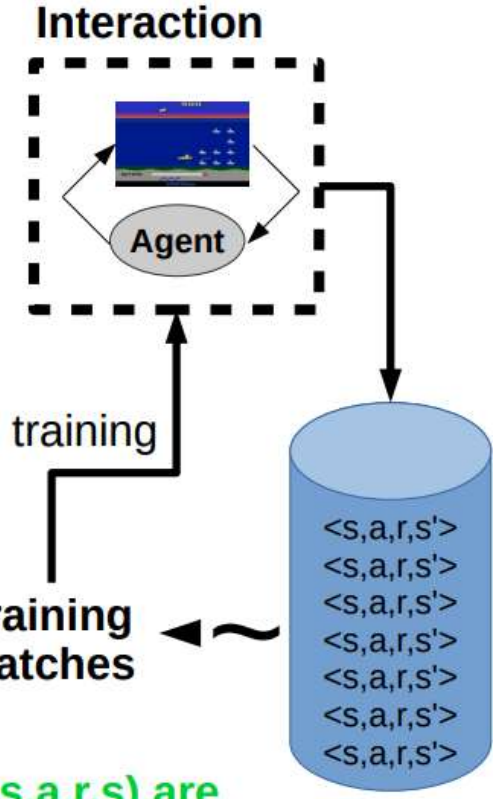
</chapter>

Experience replay

Idea: store several past interactions <*s,a,r,s'*>
Train on random subsamples

Training curriculum:
- play 1 step and record it
- pick N random transitions to train

Profit: you don't need to re-visit same (s,a) many times to learn it.



Only works with off-policy algorithms!

Old (s,a,r,s) are from older/weaker version of policy!