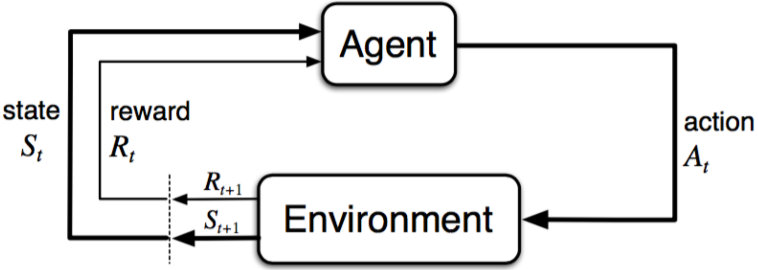
Обучение с подкреплением, Q-learning и Atari DQN

Конобеев Михаил

Задача обучения с подкреплением



Раздел машинного обучения, рассматривающий задачу выбора оптимальных действий, при взаимодействии со средой.

Виды сред:

- конечная (эпизодическая): $G_t = R_{t+1} + R_{t+2} + \dots + R_T$
- бесконечная: $G_t = R_{t+1} + R_{t+2} + \dots$

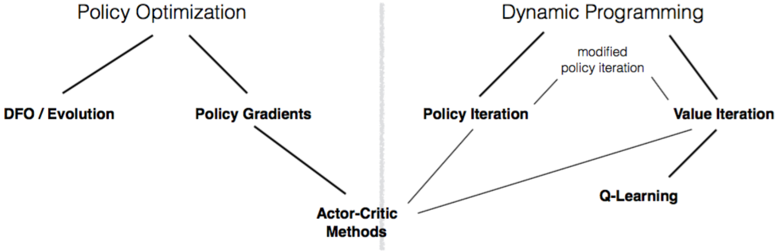
Дисконтирование:

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

Зачем нужно дисконтирование?

- неопределенность в будущем может быть не представима полностью
- люди и животные показывают предпочтение к получению награды как можно раньше
- иногда избежать дисконтирования ($\gamma = 1$), например, при конечной среде

Подходы


Policy Optimization
Dynamic Programming
modified policy iteration
DFO / Evolution
Policy Gradients
Policy Iteration
Value Iteration
Q-Learning
Actor-Critic Methods

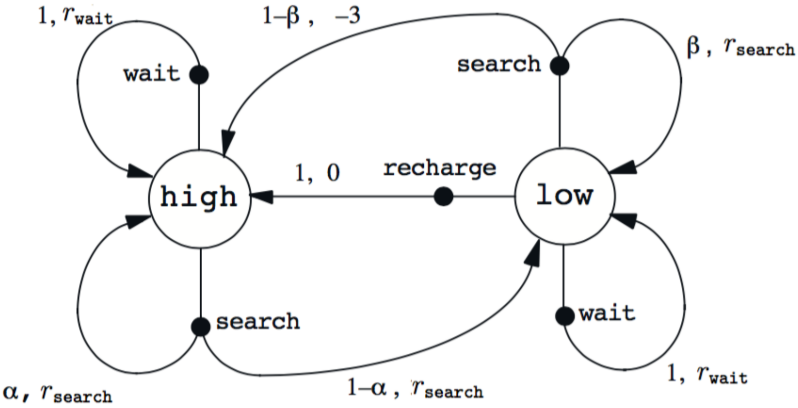
Математическая постановка

Определение

Марковский процесс принятия решения – кортеж $\langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$:

- $\mathcal{S}$ – конечное множество состояний
- $\mathcal{A}$ – конечное множество действий
- p – распределение переходов между состояниями: $p(s'|s,a) = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$
- r – функция награды: $r(s,a,s') = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s']$
- γ – коэффициент дисконтирования.

Пример MDP


1, r_{wait}
wait
1–β , –3
search
β , r_{search}
high
1, 0
recharge
low
search
wait
α , r_{search}
1–α , r_{search}
1, r_{wait}

Определение

Стратегия π – распределение по действиям для заданного состояния: $\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s]$.

Определение

Функция ценности состояния $v(s)$ для заданного MDP равна ожидаемой кумулятивной награде, начиная в состоянии s: $v_\pi(s) = \mathbb{E}_{\pi,\mathcal{E}}[G_t \mid S_t = s]$.

Определение

Функция ценности действия $q(s,a)$ для заданного MDP равна ожидаемой кумулятивной награде, начиная с состояния s, при первом действии равном a: $q_\pi(s,a) = \mathbb{E}_{\pi,\mathcal{E}}[G_t \mid S_t = s, A_t = a]$.

Определим частичное упорядочивание стратегий: $\pi \geq \pi'$, если $v_\pi(s) \geq v_{\pi'}(s),\ \forall s$.

Теорема

Для любого MDP существует оптимальная стратегия $\pi^* \geq \pi\ \forall \pi$. Для π^* верно:

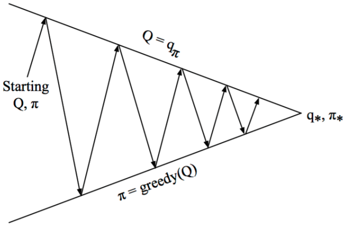
- $v_{\pi^*}(s) = \max\limits_\pi v_\pi(s),\ \forall s$
- $q_{\pi^*}(s, a) = \max\limits_\pi q_\pi(s, a),\ \forall s, a.$

$$v_{\pi^*}(s) = \sum_a \pi^*(a|s) q_{\pi^*}(s, a) \leq \max_a q_{\pi^*}(s, a),$$

но $v_{\pi^*} \geq v_\pi(s), \forall s, \forall \pi$. Следовательно, $v_{\pi*}(s) = \max_a q_{\pi^*}(s, a)$, то есть существует оптимальная детерминистическая стратегия.

Generalized Policy Iteration

Идея: оценивать, какую кумулятивную награду приносит каждое действие в текущей стратегии, после чего производить её обновление жадным образом.



Почему обычно не используется v-функция? Чтобы жадно производить улучшение необходимо знать $p(s'|s,a)$ и $r(s,a,s')$ для всех переходов.

Уравнения Беллмана

Рекурсивно используем информацию о ценности каждого состояния или пары состояние/действие:

$$
\begin{aligned}
q_\pi(s,a) &= \mathbb{E}_{\pi,\mathcal{E}}\left[G_t \mid S_t = s, A_t = a\right] \\
&= \mathbb{E}_{\pi,\mathcal{E}}\left[\sum_{k=0}^{\infty}\gamma^k R_{t+1+k} \mid S_t = s, A_t = a\right] \\
&= \mathbb{E}_{\pi,\mathcal{E}}\left[R_{t+1} + \gamma\sum_{k=0}^{\infty}\gamma^k R_{t+2+k} \mid S_t = s, A_t = a\right] \\
&= \sum_{s'} p(s'|s,a)\left[r(s,a,s') + \gamma\mathbb{E}_{\pi,\mathcal{E}}\left[\, G_{t+1} \mid S_{t+1} = s'\,\right]\right] \\
&= \mathbb{E}_{\pi,\mathcal{E}}\left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right]
\end{aligned}
$$

Уравнение оптимальности Беллмана

$$
\begin{aligned}
q_{\pi^*}(s, a) &= \mathbb{E}\left[R_{t+1} + \gamma \max_a q_{\pi^*}(S_{t+1}, a) \mid S_t = s, A_t = a\right] \\
&= \sum_{s'} p(s'|s, a)\left[r(s, a, s') + \gamma \max_{a'} q_{\pi^*}(s', a')\right]
\end{aligned}
$$

Аналогично можно получить уравнение Беллмана и уравнение оптимальности Беллмана для v-функции.

SARSA

- Оцениваем ценность пары состояние-действие лишь одной итерацией, полученной из уравнения Беллмана:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right]$$

- Обновление стратегии должно позволять исследовать новые пары состояние-действие (например, используется ε-жадная стратегия)
- On-policy – то есть происходит оценивание текущей стратегии
- Сходится к оптимальной стратегии для любого MDP, если построение новой стратегии в пределе сходятся к жадной
- При линейной аппроксимации блуждает около оптимальной стратегии

SARSA

Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Choose A from S using policy derived from Q (e.g., ε-greedy)
  Repeat (for each step of episode):
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)
    $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

Figure 6.9: Sarsa: An on-policy TD control algorithm.

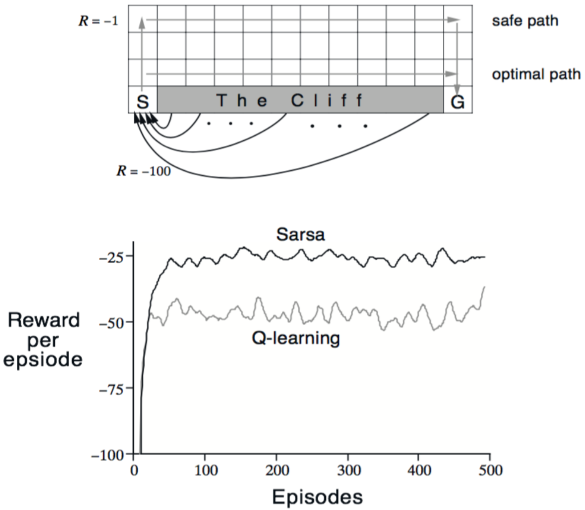
Q-learning

- Возможно ли сразу оценивать q_{π^*}?
- $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_t, a) - Q(S_t, A_t))$
- Не оценивает текущую стратегию, оценивает сразу q_{π^*} (off-policy)
- При обучении может получать кумулятивные награды меньше, чем SARSA
- При наивной линейной аппроксимации может расходится

Q-learning

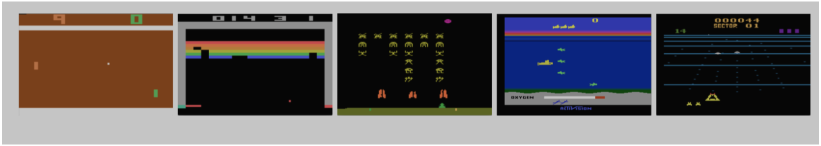
Initialize $Q(s, a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize S
    Repeat (for each step of episode):
        Choose A from S using policy derived from Q (e.g., ε-greedy)
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
        $S \leftarrow S'$;
    until S is terminal

Пример

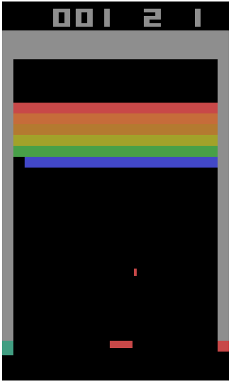

R = −1
safe path
optimal path
S
The Cliff
G
R = −100
Sarsa
Q-learning
Reward per epsiode
−25
−50
−75
−100
0
100
200
300
400
500
Episodes

Atari

- 49 разнообразных игр,
- на вход подается изображение и награда
- один и тот же алгоритм, одинаковые гиперпараметры



Не является MDP:



Простое преобразование: $s_t = (x_t, x_{t+1}, x_{t+2}, x_{t+3})$, где x_i – i-й кадр в эпизоде. Для каждого кадра из кортежа выбирается оптимальное действие, на основании s_{t-1}.

Аппроксимация функций

- Знаем для чего нужно обучаться:
    - SARSA: $Q : S_t, A_t \mapsto R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$
    - Q-Learning: $Q : S_t, A_t \mapsto R_{t+1} + \gamma \max_a Q(S_{t+1}, a)$
- Generalized Policy Iteration: меняется текущая стратегия
- Целевые переменные могут меняться при каждом обновлении весов
- Опыт взаимодействия увеличивается с каждой отметкой времени
- Невозможно обучаться онлайн (изображения не i.i.d.)
- Многие состояния не будут возникать при обучении, нужна высокая обобщающая способность

Experience Replay

Сохраним большое число взаимодействий агента со средой:
$\mathcal{D} = \{(s_1, a_1, r_1, s_2), \ldots, (s_n, a_n, r_n, s_{n+1})\}$.
Вынуждены использовать off-policy метод:

$$y_j = \begin{cases} r_j, & \text{if } s_{j+1} \text{ is terminal} \\ r_j + max_{a'} Q(s_{j+1}, a', \theta^{(i)}), & \text{otherwise} \end{cases}$$

На каждой итерации обучение происходит по случайному подмножеству (mini-batch).

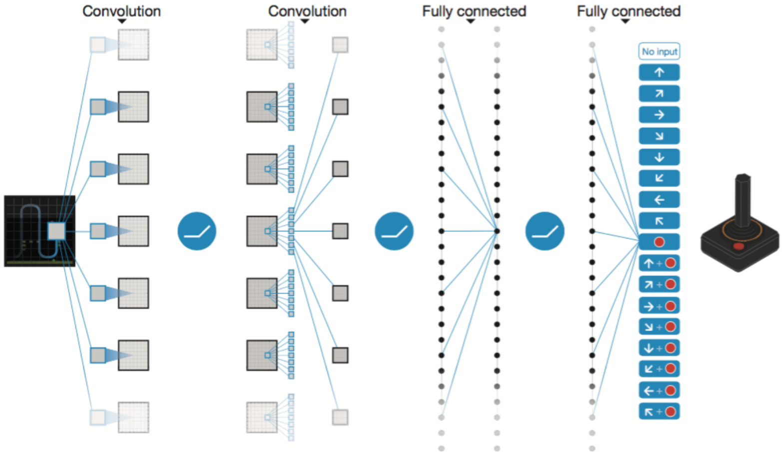
$$\theta^{(i+1)} = \theta^{(i)} - \alpha \nabla_\theta \sum_j \left(y_j - Q(s_j, a_j, \theta^{(i)}) \right)^2$$

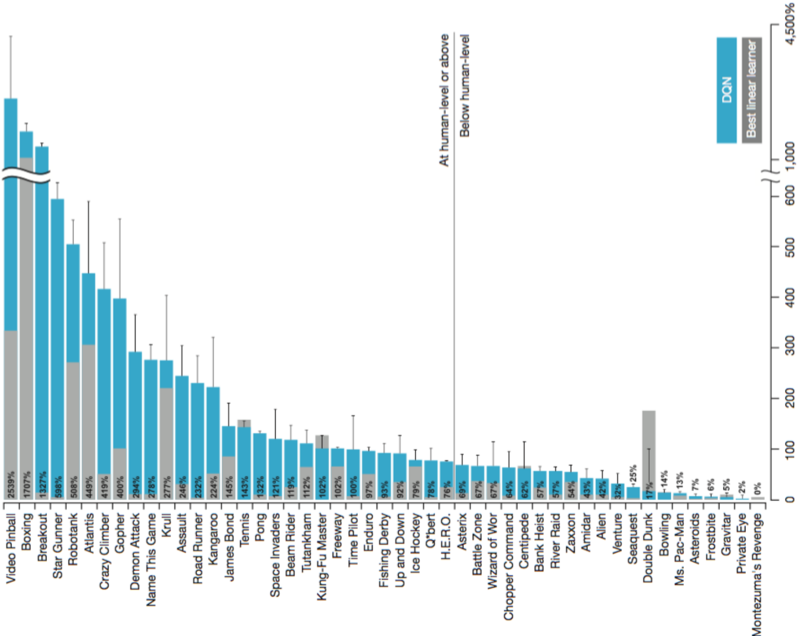
Fixed Q-target

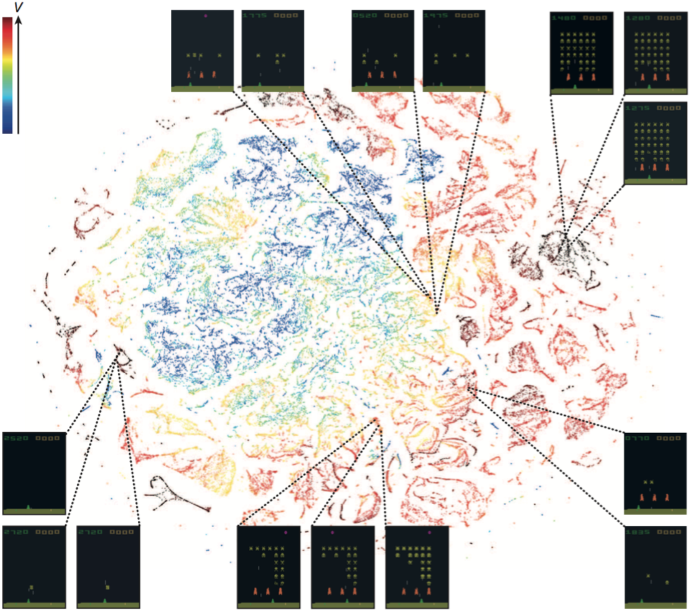
Проблема нестационарности целевых значений остается. Пусть

$$y_j = \begin{cases} r_j, & \text{if } s_{j+1} \text{ is terminal} \\ r_j + max_{a'} \hat{Q}(s_{j+1}, a', \theta^-), & \text{otherwise} \end{cases}$$

Через некоторое постоянное число итераций происходит обновление весов: $\theta^- = \theta^{(i)}$


Convolution
Convolution
Fully connected
Fully connected
No input


DQN
Best linear learner
At human-level or above
Below human-level
4,500%
1,000
600
500
400
300
200
100
0
Video Pinball 2539%
Boxing 1707%
Breakout 1327%
Star Gunner 598%
Robotank 508%
Atlantis 449%
Crazy Climber 419%
Gopher 400%
Demon Attack 294%
Name This Game 278%
Krull 277%
Assault 246%
Road Runner 232%
Kangaroo 224%
James Bond 145%
Tennis 143%
Pong 132%
Space Invaders 121%
Beam Rider 119%
Tutankham 112%
Kung-Fu Master 102%
Freeway 102%
Time Pilot 100%
Enduro 97%
Fishing Derby 93%
Up and Down 92%
Ice Hockey 79%
Q*bert 78%
H.E.R.O. 76%
Asterix 69%
Battle Zone 67%
Wizard of Wor 67%
Chopper Command 64%
Centipede 62%
Bank Heist 57%
River Raid 57%
Zaxxon 54%
Amidar 43%
Alien 42%
Venture 32%
Seaquest 25%
Double Dunk 17%
Bowling 14%
Ms. Pac-Man 13%
Asteroids 7%
Frostbite 6%
Gravitar 5%
Private Eye 2%
Montezuma's Revenge 0%


V

Источники

- Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. MIT Press, 1998.
- Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* (2015).
- David Silver. *Reinforcement Learning Course*. 2015. URL: http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html.