Q-learning и Deep Q-learning

Медведев Алексей Владимирович

МГУ имени М. В. Ломоносова, факультет ВМК, кафедра ММП

Рассмотрим задачу обучения с подкреплением, например обучим компьютер играть в Atari.
Некоторые обозначения:

- a — действие.
- s — состояние.
- π — стратегия ($p(a|s)$).
- r_t — вознаграждение в момент времени t.

ATARI

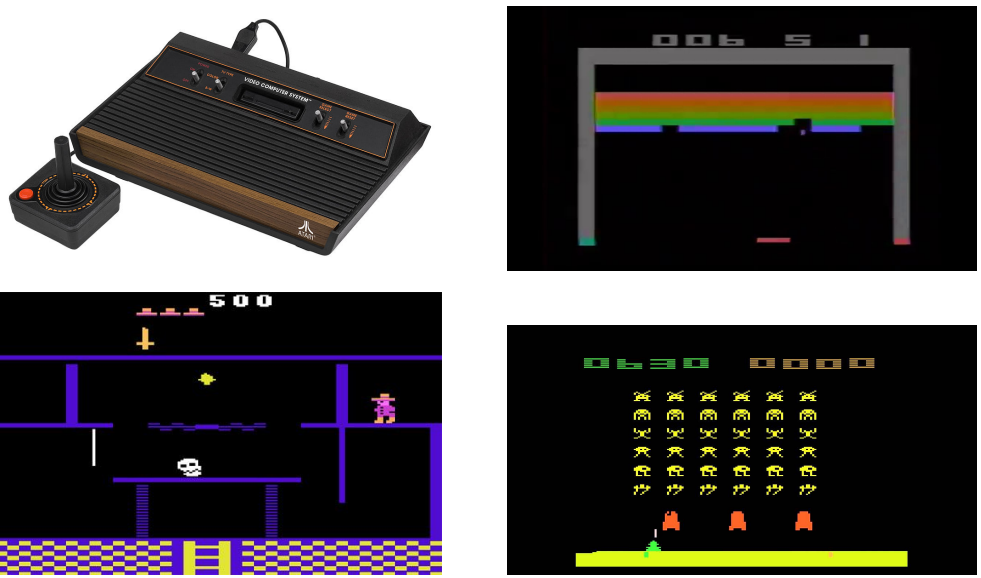


Таблица: Визуализация объекта исследований.

Небольшое введение

Q-learning — это model-free, off-policy алгоритм, который ищет оптимальную стратегию.

- model-free — значит, что алгоритм может искать оптимальную стратегию, ничего не зная о среде, в которой он находится. Или если модель MDP(markov decision process) известна, но слишком огромна (шахматы или го).
- off-policy — значит, что алгоритм получает опыт, действуя по одной стратегии, но обучает другую. Это понятие напрямую связано с Exploitation-Exporation dilemma.

Exploitation-Exporation dilemma

Мы можем действовать жадно и выбирать каждый раз максимально хорошее по какому-то нам известному критерию действие. Но тогда мы можем попасть в локальный оптимум, поэтому с вероятностью ϵ мы выбираем случайное действие. Разумно на начальных этапах выбрать $\epsilon = 1$, т.к агенту ничего неизвестно о среде.

Постановка задачи

Action-value функция:

1. $Q^*(s, a) = \max_{\pi} E\left[r_t + \gamma r_{t+1} + ... | s_t = s, a_t = a, \pi\right]$
2. $Q^*(s, a) = E_{s'}\left[r_t + \gamma \max_{a} Q^*(s', a) | s_t = s, a_t = a\right]$

Как решать?

Итеративно

$$Q_{i+1}(s,a) = E_{s'}\left[r_t + \gamma \max_a Q_i(s',a) | s_t = s, a_t = a\right]$$

Функцию Q можно промоделировать, например нейросетью

$$Q(s,a,\theta) \approx Q^*(s,a);\ y = r_t + \gamma \max_a Q(s',a,\theta^-)$$

И воспользоваться градиентным спуском

$$L(\theta_i) = E_{s,a,r,s'}\left[(y_i - Q(s,a,\theta_i))^2\right]$$
$$\nabla_{\theta_i} L(\theta_i) = E_{s,a,r,s'}\left[(y_i - Q(s,a,\theta_i))\nabla_{\theta_i} Q(s,a,\theta_i)\right]$$

Проблемы градиентного спуска

Проблемы

1. Скоррелированные данные.
2. Небольшие изменения Q функции ведут сильному изменению стратегии.(Например агент сдвинулся в левую часть карты и теперь мы подбираем стратегию только для нее, а затем при возвращении в правую часть должны радикально менять стратегию).
3. Взрыв градиента из-за неотмасштабированного значения наград.

Решения

Решения

1. Запоминать (s, s', a, r_t) в специальный буфер, а затем обучать Q функцию по mini-batch из этого множества.
2. Заморозить параметры target сети на C итераций.
3. Резать градиенты и вознаграждения.

Алгоритм

Algorithm 1: deep Q-learning with experience replay.

Initialize replay memory D to capacity N

Initialize action-value function Q with random weights θ

Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$

For episode $= 1, M$ **do**

  Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequence $\phi_1 = \phi(s_1)$

  For $t = 1$,T **do**

    With probability ε select a random action a_t

    otherwise select $a_t = \text{argmax}_a Q(\phi(s_t), a; \theta)$

    Execute action a_t in emulator and observe reward r_t and image x_{t+1}

    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$

    Store transition $\left(\phi_t, a_t, r_t, \phi_{t+1}\right)$ in D

    Sample random minibatch of transitions $\left(\phi_j, a_j, r_j, \phi_{j+1}\right)$ from D

    Set $y_j = \begin{cases} r_j & \text{if episode terminates at step j+1} \\ r_j + \gamma \max_{a'} \hat{Q}\left(\phi_{j+1}, a'; \theta^{-}\right) & \text{otherwise} \end{cases}$

    Perform a gradient descent step on $\left(y_j - Q\left(\phi_j, a_j; \theta\right)\right)^2$ with respect to the network parameters θ

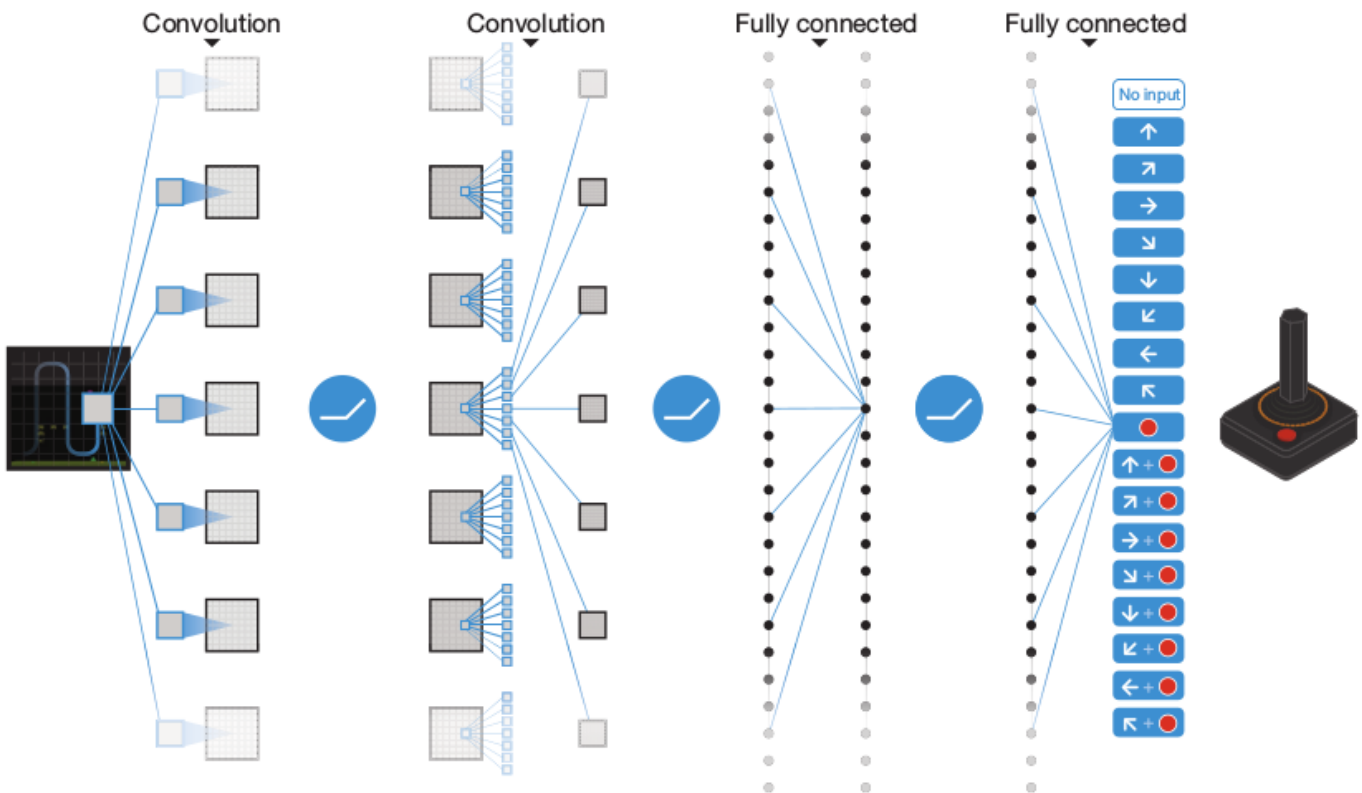
    Every C steps reset $\hat{Q} = Q$

  End For

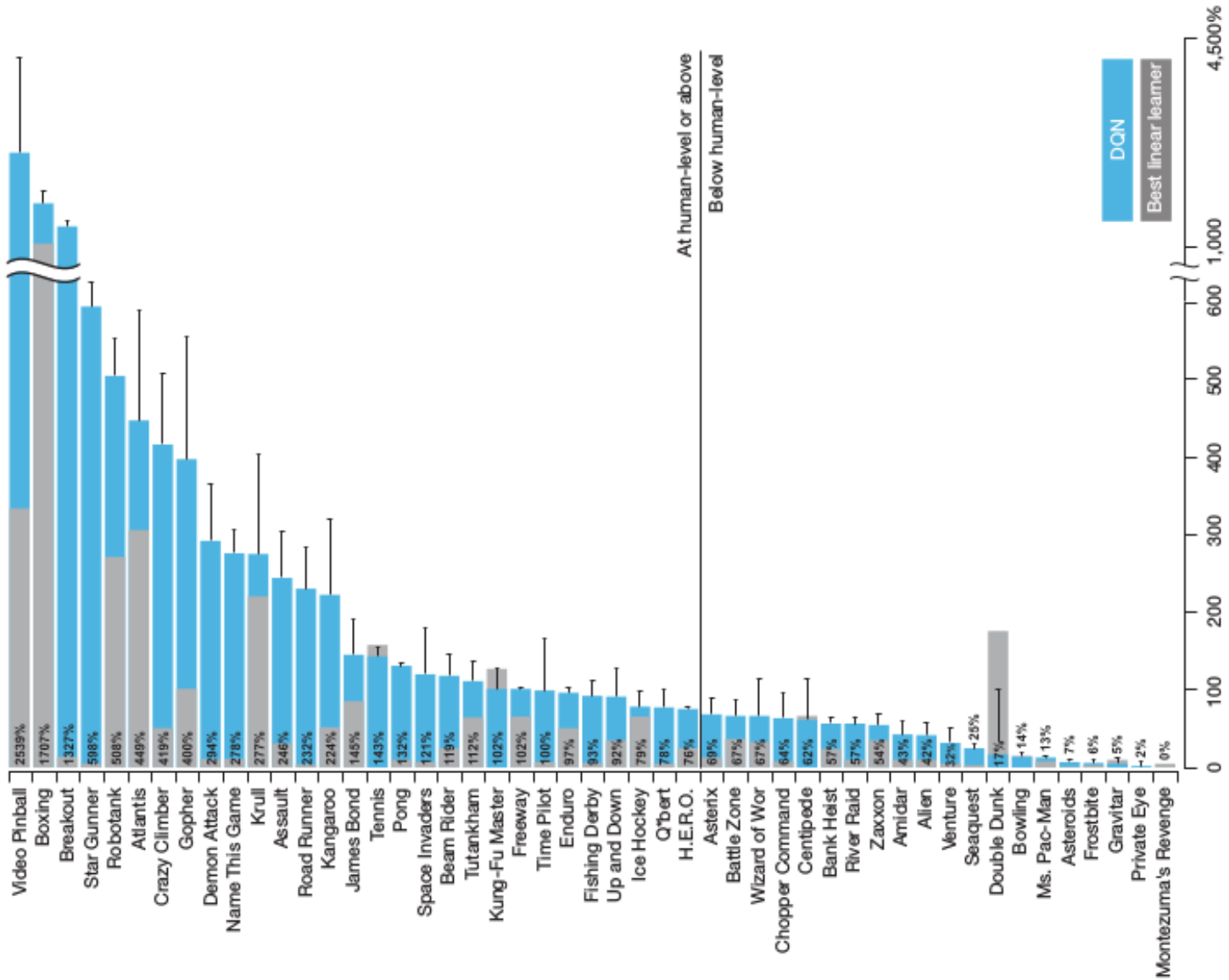
End For

DQN

Q функцию можно представить нейросетью, этот метод получил название DQN(deep Q-network). Сам эксперимент и сеть были придуманы энтузиастами компании DeepMind, которая является подразделением Google.



Итоги


DQN
Best linear learner
At human-level or above
Below human-level
0
100
200
300
400
500
600
1,000
4,500%
Video Pinball 2539%
Boxing 1707%
Breakout 1327%
Star Gunner 598%
Robotank 508%
Atlantis 449%
Crazy Climber 419%
Gopher 400%
Demon Attack 394%
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 26%
Double Dunk 17%
Bowling 14%
Ms. Pac-Man 13%
Asteroids 7%
Frostbite 6%
Gravitar 5%
Private Eye 2%
Montezuma's Revenge 0%

ball

ball

Полезные ссылки

- https://www.youtube.com/watch?v=JICo0nFtnxA
- https://www.cs.toronto.edu/ vmnih/docs/dqn.pdf
- https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf