ТЕХНОСФЕРА

Лекция 15
Reinforcement learning pt.2

Храбров Кузьма

13 мая 2017 г.

План лекции

Напоминания (MDP, policy and q-function)

Q-learning

DQN

Markov decision process

Definition

MRP это кортеж (S, A, R, P, γ), где

- S - состояния (дискретное пространтсво)
- A - действия (дискретное пространтсво)
- R - функция rewards, $R_s^a = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]$
- P - матрица переходов (transition matrix)
  $P_{ss'}^a = Pr(S_{t+1} = s'|S_t = s, A_t = a)$
- γ - discount factor

Definition (Policy)

$\pi(a|s) = Pr(A_t = a|S_t = s)$ - стратегия, т.е. то как мы выбираем действия оказавшись в состоянии s.

Definition (Value function)

$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$ - ценность состояния.

Definition (Q-function)

$q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$ - ценность действия в состоянии s.

Bellman Equations

Definition (Policy)

$\pi(a|s) = Pr(A_t = a|S_t = s)$ - стратегия, т.е. то как мы выбираем действия оказавшись в состоянии s.

Definition (Value function)

$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$ - ценность состояния.

Definition (Q-function)

$q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$ - ценность действия в состоянии s.

Bellman equations

Bellman equation для value-function

$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s \right]$$

Bellman equation для q-function

$$q_\pi(s, a) = \mathbb{E}_\pi \left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a \right]$$

Оптимальные value-functions

Оптимальная value-function - максимальное принимаемое значение

$$Q^*(s, a) = \max_\pi Q^\pi(s, a) = Q^{\pi^*}(s, a)$$

Как только мы знаем Q^* мы можем выбрать оптимальную стратегию

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

Оптимальное значение - максимум по всем принимаемым решениям:

$$Q^*(s,a) = r_{t+1} + \gamma \max_{a_{t+1}} r_{t+2} + \gamma^2 \max_{a_{t+2}} r_{t+3} + \dots$$
$$= r_{t+1} + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})$$

Соответствующее уравнение Беллмана:

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^*(s',a') \mid s,a\right]$$

Q-learning

Будем решать уравнение Беллмана:

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^*(s',a') \mid s,a\right]$$

методом конечных приращений, то есть будем повторять:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a'))$$

где α - learning rate. Обычно берут $\alpha = 0.9$.
Упражнение: покажите, что таким образом минимизируется квадрат разности.

Подходы к обучению с подкреплением

1. Value-based RL
   Оцениваем оптимальную Q-функцию $Q^*(s, a)$.
   Максимальное значение принимаемое при любой стратегии.
2. Police-based RL
   Ищем оптимальную стратегию π^*. Стратегия обеспечивающая максимальное будущее вознаграждание.
3. Model-based RL
   Строим и использем модель внешней среды.

Deep Q-learning

В каждом подходе к RL можно применить нейронные сети.
Разберем поднобно Value-based случай.
Будем аппроксимировать $Q(s,a) = Q(s,a,w)$ с помощью нейронной сети с весами w. Тогда для решения соответствующего уравнения Беллмана можно минимизировать функцию потерь MSE

$$l = \left(r + \gamma \max_a Q(s', a', w) - Q(s,a,w)\right)^2$$

Проблемы:

1. Корреляции между входами
2. Нестационарные целевые переменные

DQN-2

Чтобы убрать корреляции в данных: используем опыт агента (Replay memory)

s_1, a_1, r_2, s_2
s_2, a_2, r_3, s_3
s_3, a_3, r_4, s_4
$\dots$
$s_t, a_t, r_{t+1}, s_{t+1}$

$\rightarrow$ s, a, r, s'

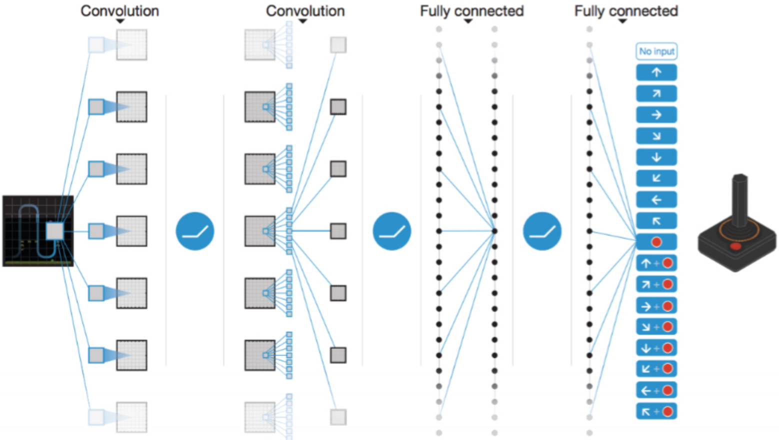
$s_t, a_t, r_{t+1}, s_{t+1}$ $\rightarrow$

Сэмплируем опыт из данных и обновляем

$$l = \left(r + \gamma \max_a Q(s', a', w^-) - Q(s, a, w)\right)^2$$

Причем w^- оставляем фиксированными, чтобы убрать нестационарность.

ATARI



Convolution
Convolution
Fully connected
Fully connected
No input

ATARI=2

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    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 $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
  end for
end for

ATARI-improvements

1. Double DQN
2. Prioritised replay
3. Duelling network

Вопросы