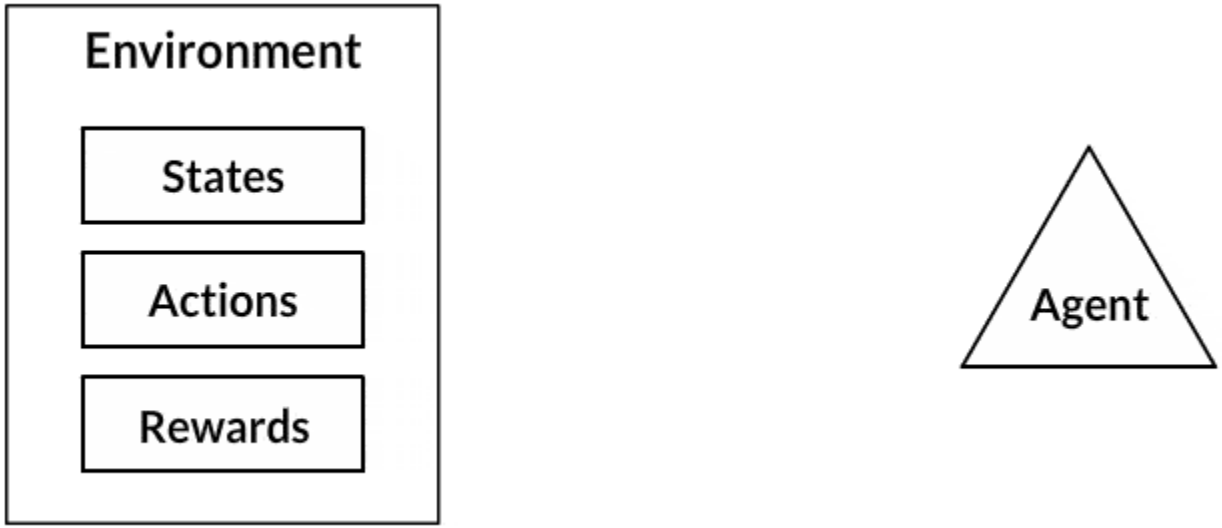
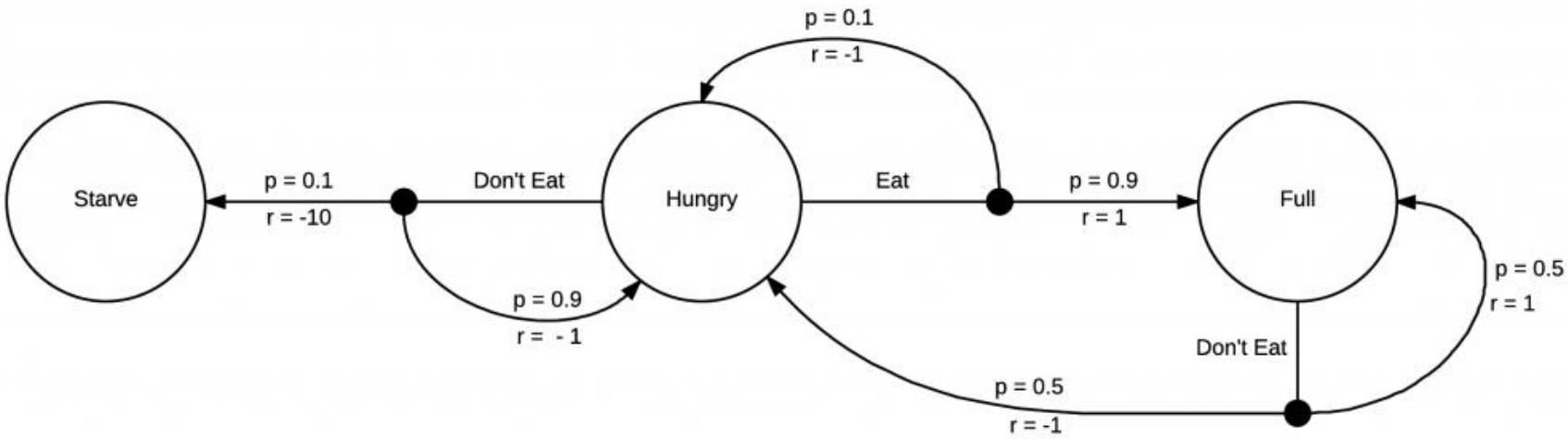
Q-learning

Студент 172 группы, Федоров Павел

Обучение с подкреплением



Стратегия



$$\pi(hungry, E) = 0.5, \pi(hungry, \overline{E}) = 0.5, \pi(full, \overline{E}) = 1.0$$

Функции значений

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t \mid s_t = s] = \mathbb{E}_{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s]$$

Функция значения состояния

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}[G_t \mid s_t = s, a_t = a] = \mathbb{E}_{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a]$$

Функция значения действия

Уравнения Беллмана

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} \mathcal{P}^{a}_{ss'} \left[\mathcal{R}^{a}_{ss'} + \gamma V^{\pi}(s') \right]$$

$$Q^{\pi}(s,a) = \sum_{s'} \mathcal{P}^{a}_{ss'} \left[\mathcal{R}^{a}_{ss'} + \gamma \sum_{a'} \pi(s',a') Q^{\pi}(s',a') \right]$$

Суть: Ценность начальной точки — это награда, которую ожидаем получить от пребывания в ней, плюс ценность того, где будем дальше.

Уравнение оптимальности Беллмана

$$Q^*(s,a) = \sum_{s',r} P(s',r|s,a)[r + \gamma max_{a'} Q^*(s',a')]$$

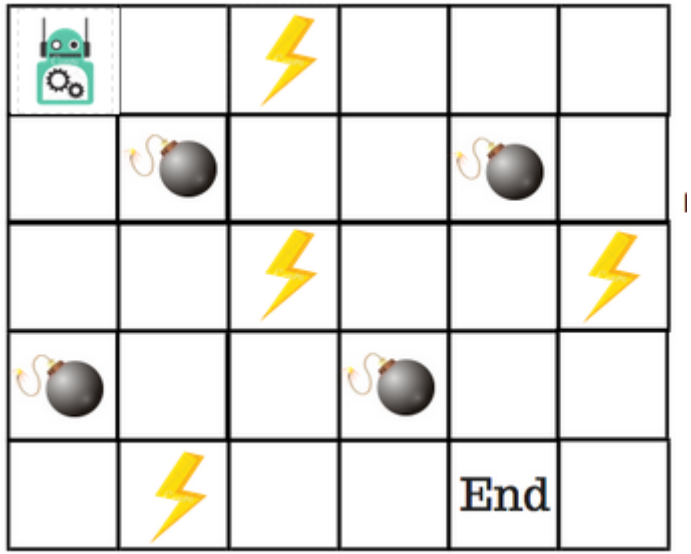
Описывает ожидаемый выход для текущего действия a в положении s и после этого придерживаемся оптимальной стратегии

Q-learning метод

Метод основан на введении функции$Q(s, a)$, отражающей ценность каждого возможного действия a агента для текущего состояния s, в котором сейчас находится симуляция.

$$Q^{new}(s_t, a_t) \leftarrow (1 - \alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}},$$

Таблица состояний значений Q



Actions :	↑	→	↓	←
Start	0	0	0	0
Nothing / Blank	0	0	0	0
Power	0	0	0	0
Mines	0	0	0	0
END	0	0	0	0

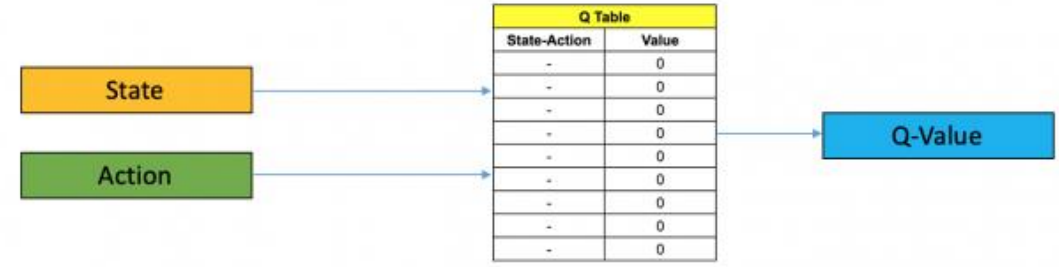
Алгоритм

- Шаг 1: инициализируем Q-таблицу, заполняя ее нулями, а для Q-значений задаем произвольные константы.
- Шаг 2: теперь пусть агент реагирует на окружающую среду и пробует разные действия. Для каждого изменения состояния выбираем одно из всех действий, возможных в данном состоянии (S).
- Шаг 3: Переходим к следующему состоянию (S’) по результатам предыдущего действия
- Шаг 4: Для всех возможных действий из состояния (S’) выбираем одно с наивысшим Q-значением.
- Шаг 5: Обновляем значения Q-таблицы в соответствии с вышеприведенным уравнением.
- Шаг 6: Превращаем следующее состояние в текущее.
- Шаг 7: Если целевое состояние достигнуто – завершаем процесс, а затем повторяем.

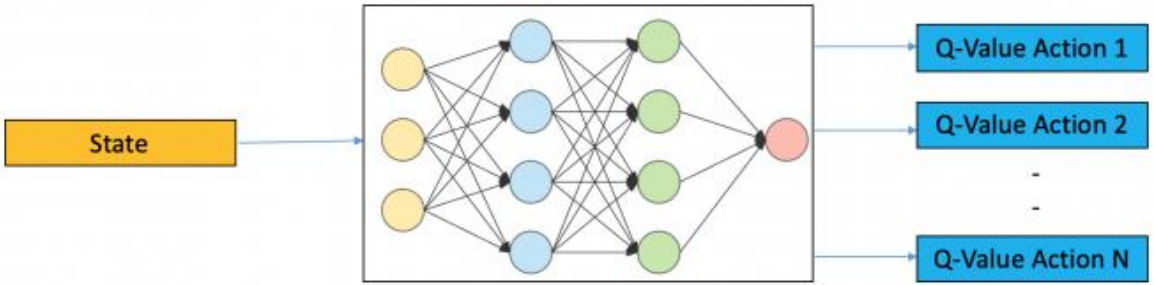
On-policy и off-policy

SARSA	Q-обучение
Обучение стратегии, наилучшей среди ε-жадных.	Обучение оптимальной стратегии
Высокая скорость обучения, т.к. принимаются во внимание “исследовательские” шаги (с вероятностью ε)	Высока вероятность попадания в локальный минимум.

Глубокие Q-learning сети

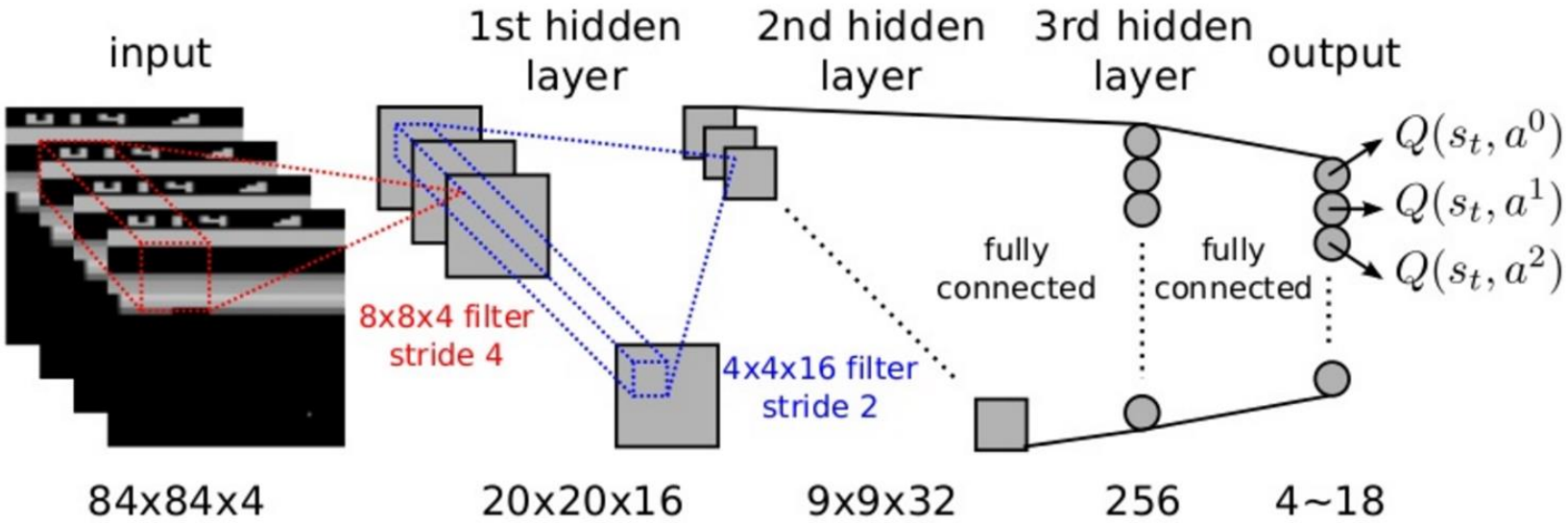


Q Learning



Deep Q Learning

Архитектура DQN



Experience replay

1. Собрать и сохранить сэмплы в буфере с текущей стратегией
2. Сэмплировать батчи опыта $e_t = (s_t, a_t, r_t, s_{t+1})$ из буфера
3. Использовать их для обновления сети
4. Повторить 1-3

$$L(\theta_i) = \mathbb{E}_{s,a\sim p(\cdot)}[(y_i - Q(s, a; \theta_i))^2]$$

где,

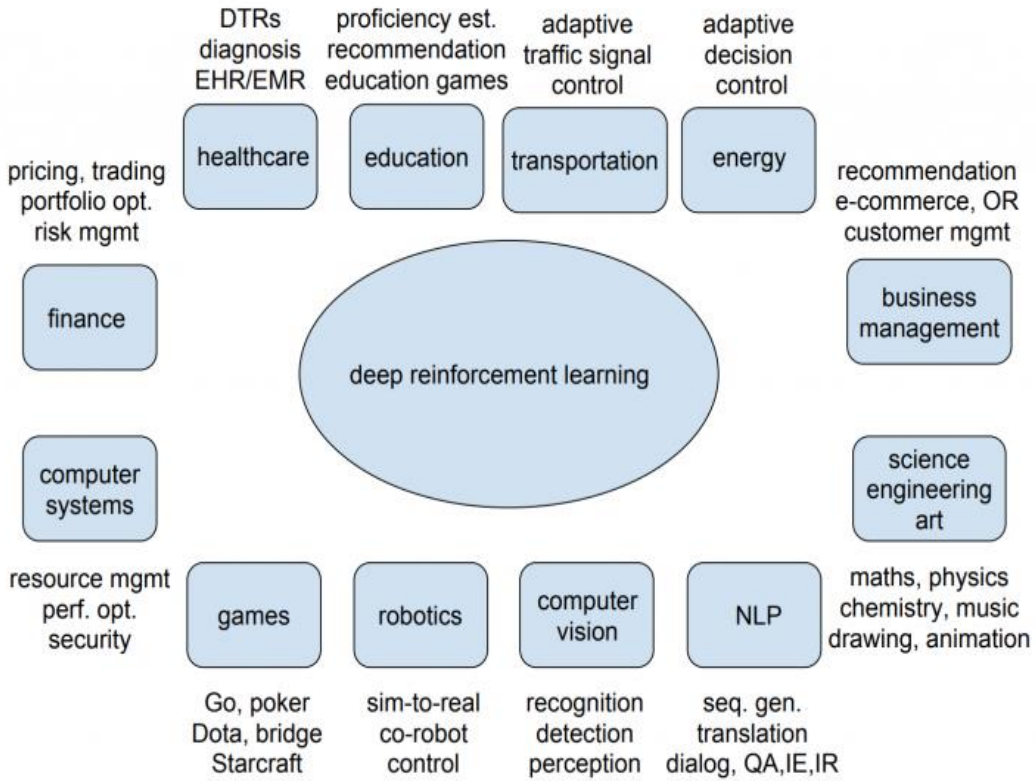
$$y_i = \begin{cases} R_T & \text{for terminal state } s_T \\ R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') & \text{for non-terminal state } s_t \end{cases}$$

Experience Replay

Достоинства:
Более эффективное использование предыдущего опыта благодаря многократному обучению. Когда получение реального опыта обходится дорого, можно использовать уже накопленный. Обновления сети являются инкрементными и не сходятся быстро, поэтому полезно многократное прохождение с одними и теми же данными.

Недостатки:
Труднее использовать многошаговые алгоритмы обучения, которые можно настроить, чтобы получить лучшие кривые обучения, балансируя между смещением (из-за начальной загрузки) и дисперсией (из-за задержек и случайности в долгосрочных результатах).

Область применений



Вопросы

1. Записать уравнение Беллмана (с пояснениями). В чем его смысл и как оно связано с q-learning?

2. Для чего нужен experience replay? Его плюсы и недостатки

3. Описать архитектуру deep q-network

Список литературы

1. https://xaviergeerinck.com/bellman-equations
2. https://arxiv.org/pdf/1312.5602.pdf
3. https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/
4. https://blog.floydhub.com/an-introduction-to-q-learning-reinforcement-learning/
5. http://www.incompleteideas.net/book/RLbook2018trimmed.pdf
6. https://towardsdatascience.com/dqn-part-1-vanilla-deep-q-networks-6eb4a00febfb