Введение в Reinforcement Learning

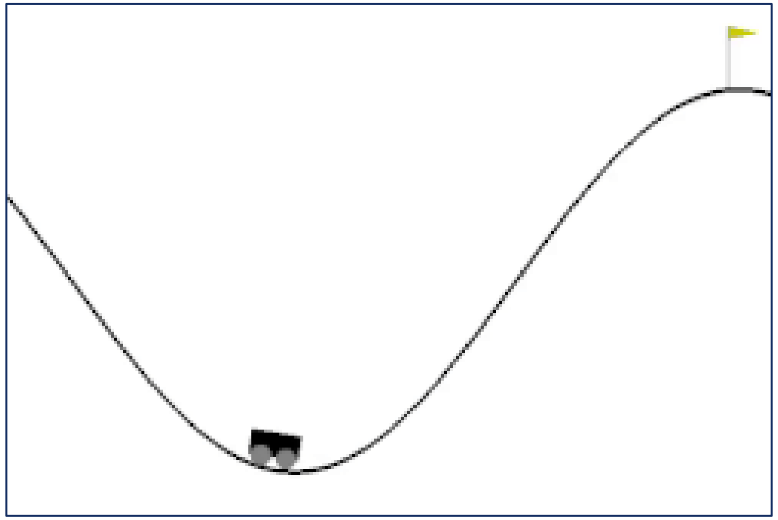
Виктор Кантор

План

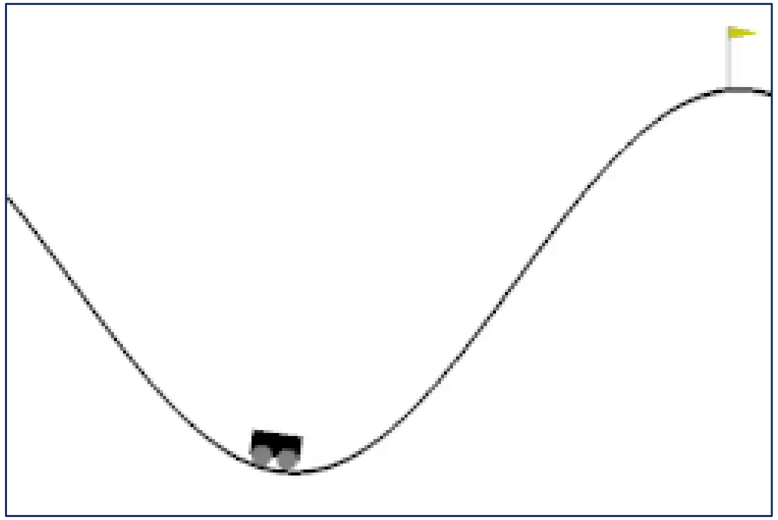
1. Примеры задач
2. Базовые идеи RL
3. SARSA & Q-learning
4. Policy Gradient и идеи его улучшения

1. Примеры задач

Mountain car

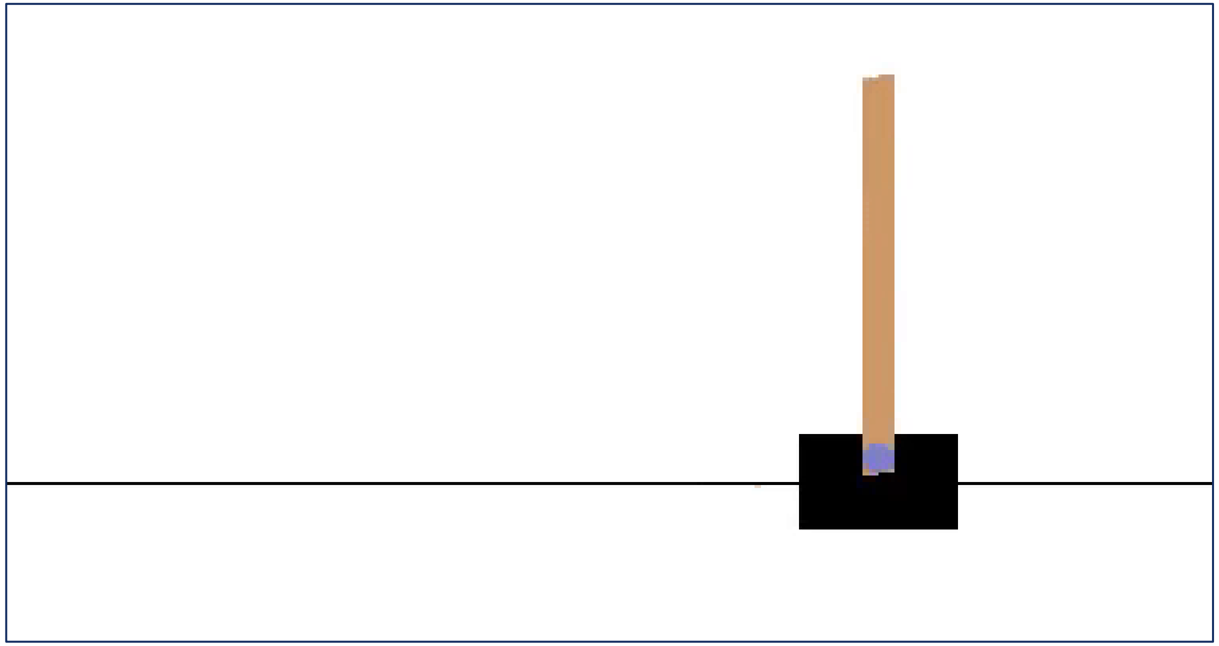


В начале обучения

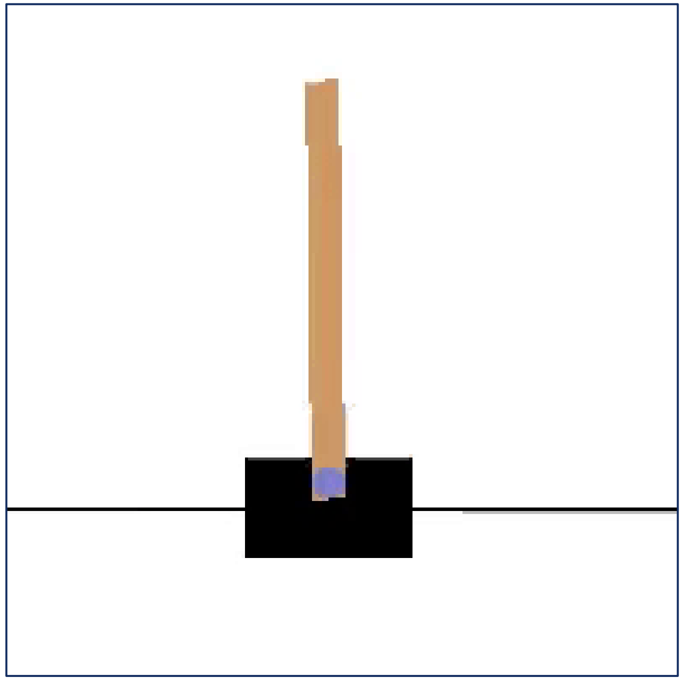


В конце обучения

Cartpole (Перевернутый маятник)



Частично решенная задача

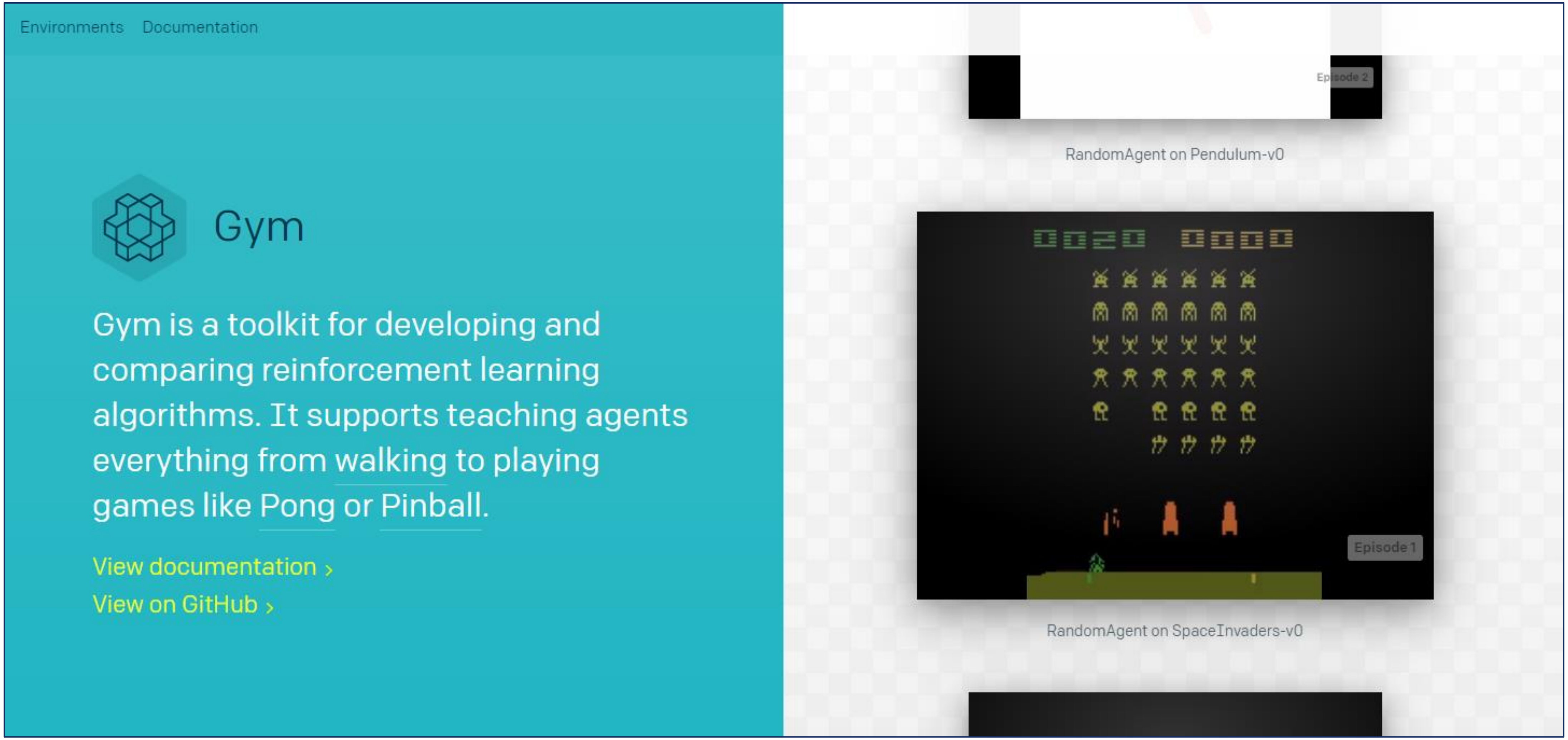


Итоговый алгоритм

Cartpole (Перевернутый маятник)

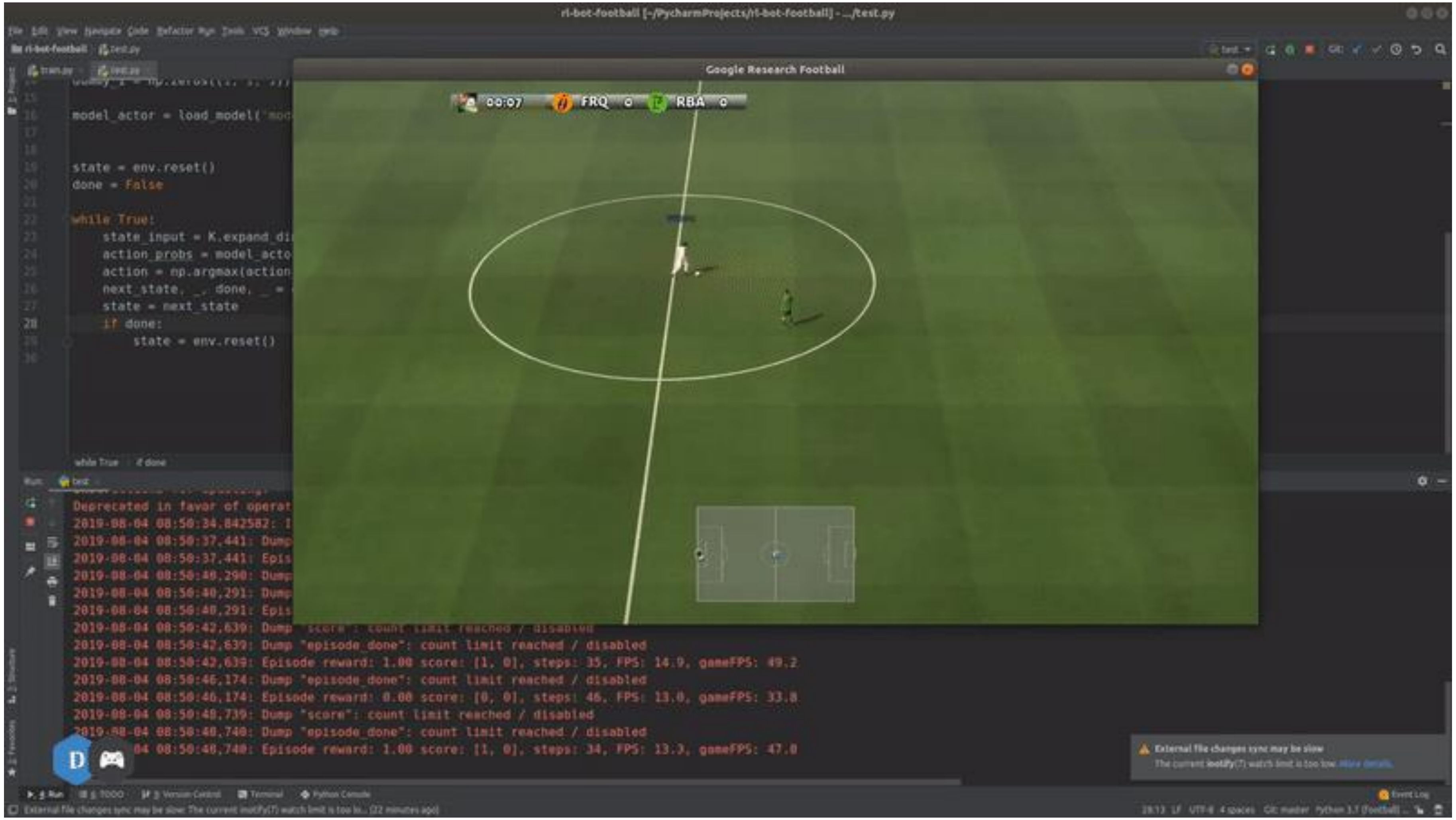
Источник: www.youtube.com/watch?v=XiigTGKZfks

Тестовая среда: Open AI Gym



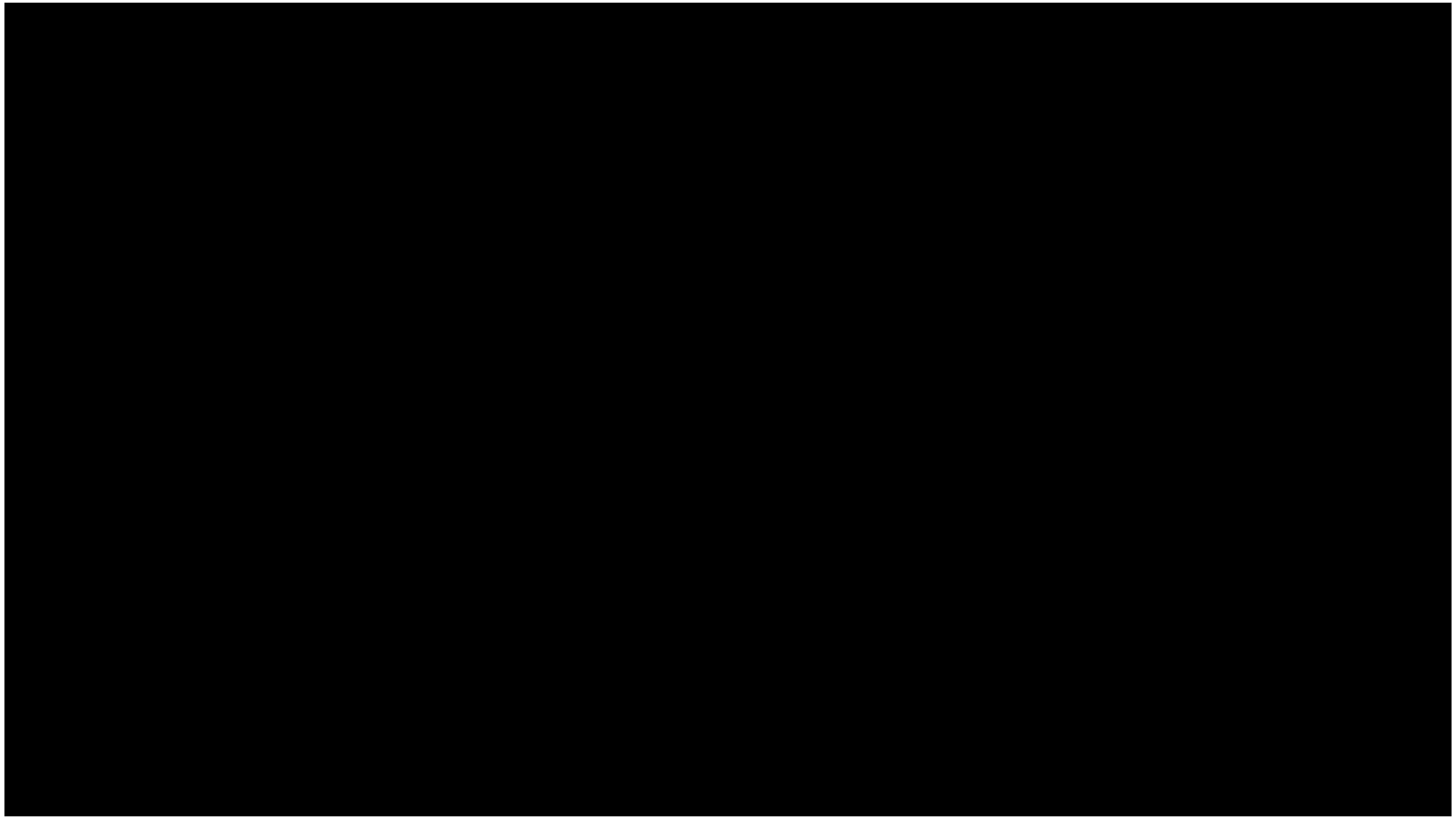
https://gym.openai.com/

Google Research Football



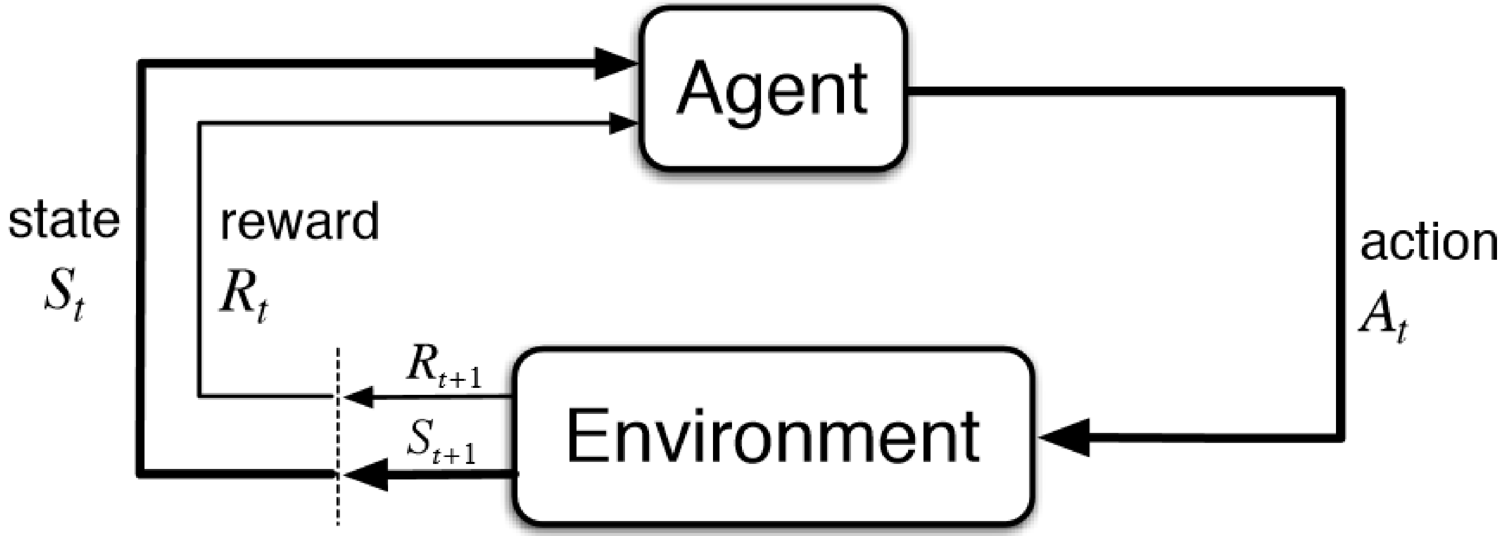
https://github.com/google-research/football https://ai.googleblog.com/2019/06/introducing-google-research-football.html

Google Research Football



https://github.com/google-research/football https://ai.googleblog.com/2019/06/introducing-google-research-football.html

Взаимодействие агента со средой



2. Базовые идеи RL

Markov Decision Process (MDP)

$$(S, A, \{P_{sa}\}, \gamma, R)$$

- S – множество состояний
- A – множество действий
- $\{P_{sa}\}$ - распределение вероятностей перехода в новое состояние: $P_{sa}(s')$ – вероятность перехода из s в s' после действия a (моделирует поведение среды)
- R – фидбек / награда от среды: $R(s)$ «баллов» получает агент попав в состояние s, в более общем виде - $R(s, a)$ (может влиять не только состояние, но и действие)
- γ – коэффициент дисконтирования награды

Стратегия (policy) и value function

$$\pi : S \rightarrow A$$

$$V^{\pi}(s) = \boldsymbol{E}\left[\sum_{k=0}^{+\infty} \gamma^{k} R(s_k) \,|s_0 = s, \pi\right]$$

Стратегия (policy) и value function

$$\pi : S \to A$$

$$V^{\pi}(s) = \boldsymbol{E}\left[\sum_{k=0}^{+\infty} \gamma^k R(s_k) \,|s_0 = s, \pi\right]$$

Вопрос к любителям строгости обозначений: что «не так» с условием в этой вероятности?

Уравнение Беллмана для V^{π}

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s') \quad (1)$$

Уравнение Беллмана для V^{π}

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s')$$

Уравнение Беллмана для V^{π}

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s')$$

Справедливость уравнения очевидна, а польза? Чем оно нам может быть полезно?

Уравнение Беллмана для V^{π}

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s')$$

Справедливость уравнения очевидна, а польза? Чем оно нам может быть полезно?

Имея политику $\pi(s)$ и уже оценив $\{P_{sa}\}$ можем записать $|S|$ линейных уравнений относительно $V^{\pi}(s)$, решив которые получим значения value function

Как оценить $\{P_{sa}\}$ и $R(s)$

Простой ответ: у вас есть среда, поиграйте с ней ☺

1. Инициализируем π
2. Повторяем:
   a) Выполнить π в MDP некоторое количество раз
   b) Обновить $R(s)$ и $P_{sa}(s') = \frac{\#(a,s) \to s' + 1}{\#(a,s) + |S|}$
   c) Обновить V и π

Optimal value function

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

$$\boxed{V^*(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V^*(s')} \quad (2)$$

Optimal policy

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

$$\boxed{\pi^*(s) = \operatorname*{argmax}_{a \in A} \sum_{s' \in S} P_{sa}(s') V^*(s')}$$

3

Value iteration

1. Для всех s инициализируем $V(s) := 0$
2. Повторяем до сходимости для каждого s:

$$V^{\pi}(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V^{*}(s')$$

2

Value iteration

1. Для всех s инициализируем $V(s) := 0$
2. Повторяем до сходимости для каждого s:

$$V^{\pi}(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V^{*}(s') \quad \leftarrow 2$$

Вопрос: как и когда получим стратегию?

Policy iteration

1. Инициализируем π
2. Повторяем:

   a) $V := V^{\pi}$ 1

   b) $\pi(s) = \underset{a \in A}{\operatorname{argmax}} \sum_{s' \in S} P_{sa}(s') V(s')$ 3

Policy iteration

1. Инициализируем π
2. Повторяем:

   a) $V := V^{\pi}$ 1

   b) $\pi(s) = \underset{a \in A}{\operatorname{argmax}} \sum_{s' \in S} P_{sa}(s') V(s')$ 3

Вопрос: как вы думаете, когда используют policy iteration, а когда value iteration?

Непрерывное пространство состояний

- С конечным набором состояний теперь справимся
- Что делать, если у нас непрерывное пространство состояний, например координаты?

Непрерывное пространство состояний

- С конечным набором состояний теперь справимся
- Что делать, если у нас непрерывное пространство состояний, например координаты?

Вариант 1. Дискретизация пространства состояний (нарежем координаты на клетки)

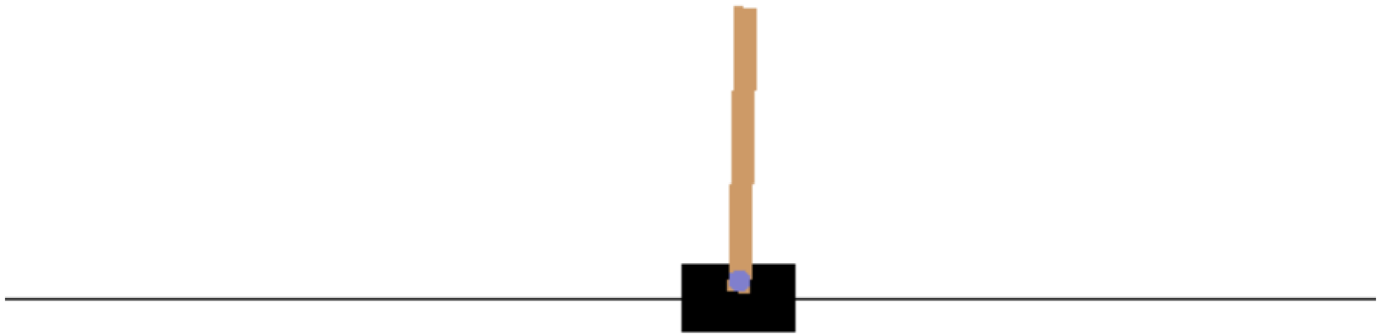
Непрерывное пространство состояний

- С конечным набором состояний теперь справимся
- Что делать, если у нас непрерывное пространство состояний, например координаты?

Вариант 1. Дискретизация пространства состояний (нарежем координаты на клетки)

Вариант 2. Прикручивать модели, которые будут прогнозировать s_{t+1} по s_t и a_t (не обязательно ML, можно физические или иные пригодные для моделирования среды). Также понадобится аппроксимировать моделью value function.

Пример задачи: перевернутый маятник



3. On-policy & off-policy: SARSA & Q-learning

Q-function (state-action value)

$$
\begin{aligned}
Q^{\pi}(s,a) &= \mathrm{E}_{\pi}\left(\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} \;\middle|\; s_t = s,\ a_t = a\right) = \\
&= \mathrm{E}_{\pi}\left(r_{t+1} + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) \;\middle|\; s_t = s,\ a_t = a\right)
\end{aligned}
$$

SARSA

1: инициализация стратегии $\pi_1(a|s)$ и состояния среды s_1
2: **для всех** $t = 1, \dots T, \dots$
3: агент выбирает действие $a_t \sim \pi_t(a|s_t)$:
$a_t = \arg\max_a Q(s_t, a)$ — жадная стратегия
(но возможны и другие: ε-жадная, по Гиббсу, $\dots$)
4: среда генерирует $r_{t+1} \sim p(r|a_t, s_t)$ и $s_{t+1} \sim p(s|a_t, s_t)$;
5: агент разыгрывает ещё один шаг: $a' \sim \pi_t(a|s_{t+1})$;
6: $Q(s_t, a_t) := Q(s_t, a_t) + \alpha_t\big(r_{t+1} + \gamma Q(s_{t+1}, a') - Q(s_t, a_t)\big)$;

http://www.machinelearning.ru/wiki/images/archive/3/35/20121120170456%21Voron-ML-RL-slides.pdf

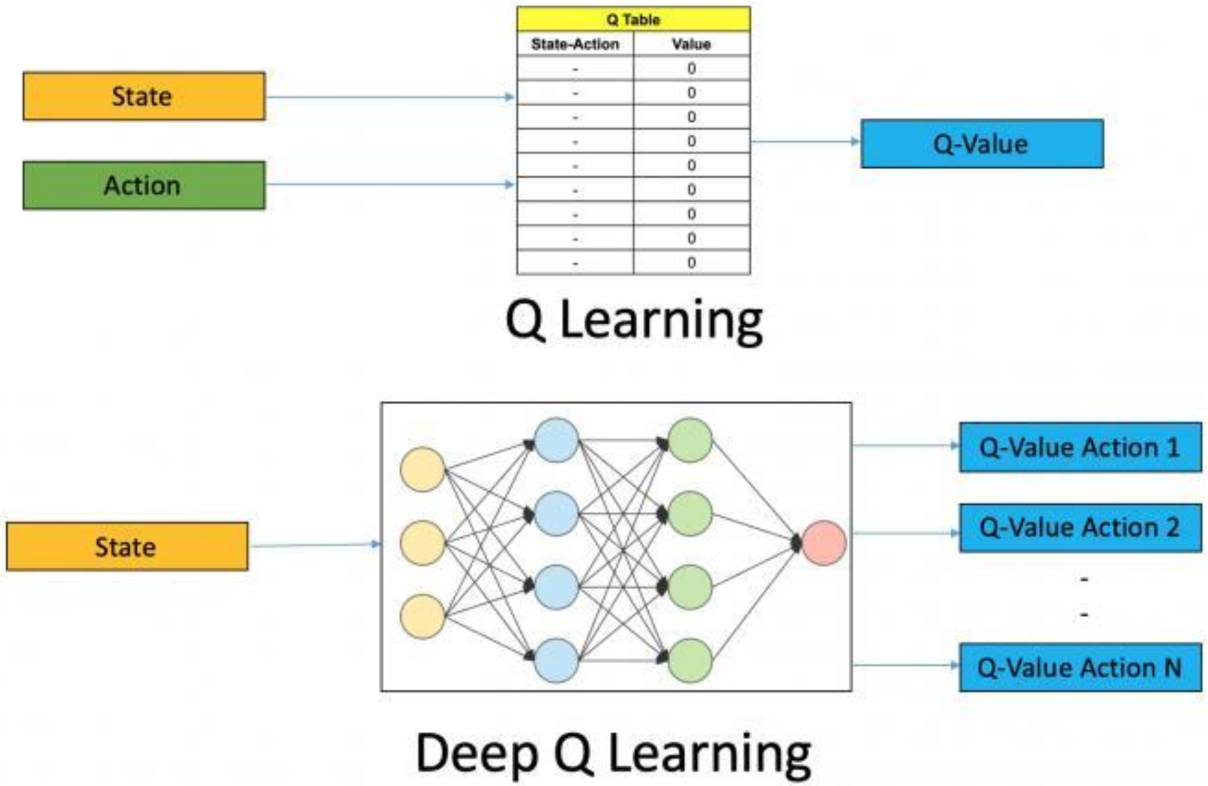
Q-learning

$$Q^*(s, a) = \mathrm{E}\Big(r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a')\ \Big|\ s_t = s,\ a_t = a\Big)$$

Делаем то же, что в SARSA, но:

$$Q(s_t, a_t) := Q(s_t, a_t) + \alpha_t\Big(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\Big)$$

Deep Q-learning



Как добавить модель в обучение

Вместо шага с обновлением state-action value function, появляется шаг с обновлением таргета модели и обновлением параметров модели:

$$y(s') = R_{sa}(s') + \gamma \max_{a'} Q_k(s', a')$$

$$\theta_{k+1} = \theta_k - \alpha \nabla_\theta \boldsymbol{E}_{s' \sim P_{sa}(s')}[(Q_k(s', a') - y(s'))^2]|_{\theta=\theta_k}$$

4. Policy Gradient и идеи его улучшения

Policy Gradient

- Рассмотрим случай стохастической стратегии: вместо детерминированной функции $\pi(s)$ будем обучать распределение $\pi_\theta(a|s)$ в виде модели (например, нейросети) с параметрами θ
- Попробуем выписать градиент оптимизируемого функционала по параметрам стратегии и обновлять их по направлению этого градиента

Policy Gradient

- Рассмотрим случай стохастической стратегии: вместо детерминированной функции $\pi(s)$ будем обучать распределение $\pi_\theta(a|s)$ в виде модели (например, нейросети) с параметрами θ
- Попробуем выписать градиент оптимизируемого функционала по параметрам стратегии и обновлять их по направлению этого градиента

Вопрос: почему по направлению градиента, а не против?

Policy Gradient

- Инициализируем стратегию и насэмплируем T траекторий из MDP
- Изучим оптимизируемый функционал:

$$\eta(\theta) \triangleq \mathrm{E}\left[\sum_{t=0}^{T-1} \gamma^t R(s_t, a_t)\right]$$

$$f(\tau) = \textstyle\sum_{t=0}^{T-1} \gamma^t R(s_t, a_t) \qquad\qquad \eta(\theta) = \mathrm{E}_{\tau\sim P_\theta}\left[f(\tau)\right]$$

Policy Gradient

$$\begin{aligned}\nabla_\theta \mathrm{E}_{\tau \sim P_\theta}\left[f(\tau)\right] &= \nabla_\theta \int P_\theta(\tau) f(\tau) d\tau \\ &= \int \nabla_\theta (P_\theta(\tau) f(\tau)) d\tau \\ &= \int (\nabla_\theta P_\theta(\tau)) f(\tau) d\tau\end{aligned}$$

Policy Gradient

$$\begin{aligned}
\nabla_\theta \mathrm{E}_{\tau \sim P_\theta}\left[f(\tau)\right] &= \nabla_\theta \int P_\theta(\tau) f(\tau) d\tau \\
&= \int \nabla_\theta (P_\theta(\tau) f(\tau)) d\tau \\
&= \int (\nabla_\theta P_\theta(\tau)) f(\tau) d\tau \\
&= \int P_\theta(\tau) (\nabla_\theta \log P_\theta(\tau)) f(\tau) d\tau
\end{aligned}$$

Policy Gradient

$$\begin{aligned}
\boxed{\nabla_\theta \mathrm{E}_{\tau \sim P_\theta}\left[f(\tau)\right]} &= \nabla_\theta \int P_\theta(\tau) f(\tau) d\tau \\
&= \int \nabla_\theta (P_\theta(\tau) f(\tau)) d\tau \\
&= \int (\nabla_\theta P_\theta(\tau)) f(\tau) d\tau \\
&= \int P_\theta(\tau) (\nabla_\theta \log P_\theta(\tau)) f(\tau) d\tau \\
&\boxed{= \mathrm{E}_{\tau \sim P_\theta}\left[(\nabla_\theta \log P_\theta(\tau)) f(\tau)\right]}
\end{aligned}$$

Policy Gradient

$$\boxed{\nabla_\theta \mathrm{E}_{\tau\sim P_\theta}\left[f(\tau)\right]} = \nabla_\theta \int P_\theta(\tau) f(\tau) d\tau$$

$$= \int \nabla_\theta (P_\theta(\tau) f(\tau)) d\tau$$

$$= \int (\nabla_\theta P_\theta(\tau)) f(\tau) d\tau$$

$$= \int P_\theta(\tau) (\nabla_\theta \log P_\theta(\tau)) f(\tau) d\tau$$

$$\boxed{= \mathrm{E}_{\tau\sim P_\theta}\left[(\nabla_\theta \log P_\theta(\tau)) f(\tau)\right]}$$

Вопрос: то, что это красиво, итак понятно, но зачем?

Policy Gradient

$$\boxed{\nabla_\theta \mathrm{E}_{\tau \sim P_\theta}\left[f(\tau)\right]} = \nabla_\theta \int P_\theta(\tau) f(\tau) d\tau$$

$$= \int \nabla_\theta (P_\theta(\tau) f(\tau)) d\tau$$

$$= \int (\nabla_\theta P_\theta(\tau)) f(\tau) d\tau$$

$$= \int P_\theta(\tau)(\nabla_\theta \log P_\theta(\tau)) f(\tau) d\tau$$

$$\boxed{= \mathrm{E}_{\tau \sim P_\theta}\left[(\nabla_\theta \log P_\theta(\tau)) f(\tau)\right]}$$

Вопрос: то, что это красиво, итак понятно, но зачем?

Ответ: чтобы не брать градиенты матожидания f, если не можем

Как это считать

$$\nabla_\theta \mathrm{E}_{\tau \sim P_\theta}\left[f(\tau)\right] = \mathrm{E}_{\tau \sim P_\theta}\left[(\nabla_\theta \log P_\theta(\tau)) f(\tau)\right]$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} (\nabla_\theta \log P_\theta(\tau^{(i)})) f(\tau^{(i)})$$

Как это считать

$$\begin{aligned}\nabla_\theta \mathrm{E}_{\tau\sim P_\theta}\left[f(\tau)\right] &= \mathrm{E}_{\tau\sim P_\theta}\left[(\nabla_\theta \log P_\theta(\tau))f(\tau)\right] \\ &\approx \frac{1}{n}\sum_{i=1}^{n}(\nabla_\theta \log P_\theta(\tau^{(i)}))f(\tau^{(i)})\end{aligned}$$

$$P_\theta(\tau) = \mu(s_0)\pi_\theta(a_0|s_0)P_{s_0a_0}(s_1)\pi_\theta(a_1|s_1)P_{s_1a_1}(s_2)\cdots P_{s_{T-1}a_{T-1}}(s_T)$$

Как это считать

$$\nabla_\theta \mathrm{E}_{\tau\sim P_\theta}\left[f(\tau)\right] = \mathrm{E}_{\tau\sim P_\theta}\left[(\nabla_\theta \log P_\theta(\tau))f(\tau)\right]$$
$$\approx \frac{1}{n}\sum_{i=1}^{n}(\nabla_\theta \log P_\theta(\tau^{(i)}))f(\tau^{(i)})$$

$$P_\theta(\tau) = \mu(s_0)\pi_\theta(a_0|s_0)P_{s_0a_0}(s_1)\pi_\theta(a_1|s_1)P_{s_1a_1}(s_2)\cdots P_{s_{T-1}a_{T-1}}(s_T)$$

$$\log P_\theta(\tau) = \log\mu(s_0) + \log\pi_\theta(a_0|s_0) + \log P_{s_0a_0}(s_1) + \log\pi_\theta(a_1|s_1)$$
$$+ \log P_{s_1a_1}(s_2) + \cdots + \log P_{s_{T-1}a_{T-1}}(s_T)$$

Как это считать

$$\nabla_\theta \mathrm{E}_{\tau\sim P_\theta}\left[f(\tau)\right] = \mathrm{E}_{\tau\sim P_\theta}\left[(\nabla_\theta \log P_\theta(\tau))f(\tau)\right]$$
$$\approx \frac{1}{n}\sum_{i=1}^{n}(\nabla_\theta \log P_\theta(\tau^{(i)}))f(\tau^{(i)})$$

$$P_\theta(\tau) = \mu(s_0)\pi_\theta(a_0|s_0)P_{s_0a_0}(s_1)\pi_\theta(a_1|s_1)P_{s_1a_1}(s_2)\cdots P_{s_{T-1}a_{T-1}}(s_T)$$

$$\log P_\theta(\tau) = \log\mu(s_0) + \log\pi_\theta(a_0|s_0) + \log P_{s_0a_0}(s_1) + \log\pi_\theta(a_1|s_1)$$
$$+ \log P_{s_1a_1}(s_2) + \cdots + \log P_{s_{T-1}a_{T-1}}(s_T)$$

$$\nabla_\theta \log P_\theta(\tau) = \nabla_\theta \log\pi_\theta(a_0|s_0) + \nabla_\theta \log\pi_\theta(a_1|s_1) + \cdots + \nabla_\theta \log\pi_\theta(a_{T-1}|s_{T-1})$$

Итоговый вид градиента по параметрам стратегии

$$
\begin{aligned}
\nabla_\theta \eta(\theta) = \nabla_\theta \mathrm{E}_{\tau \sim P_\theta}\left[f(\tau)\right] &= \mathrm{E}_{\tau \sim P_\theta}\left[\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)\right) \cdot f(\tau)\right] \\
&= \mathrm{E}_{\tau \sim P_\theta}\left[\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)\right) \cdot \left(\sum_{t=0}^{T-1} \gamma^t R(s_t, a_t)\right)\right]
\end{aligned}
$$

В результате можем применять метод, умея сэмплировать из $\{P_{sa}\}$ и умея получать награду по конкретному состоянию и действию $R(s,a)$ - больше нам ничего не требуется

Идеи в основе Proximal Policy Optimization

- Градиент из PG: $\hat{g} = \hat{\mathbb{E}}_t \left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$

- PG Loss: $L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$

Подход Actor-Critic

- Одна модель (актер) оценивает $\pi_\theta(a|s)$ (это было в PG)
- Другая модель (критик) оценивает value function либо advantage (этого не было в PG)

Идеи в основе Proximal Policy Optimization

- Градиент из PG: $\hat{g} = \hat{\mathbb{E}}_t \left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$

- PG Loss: $L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$

- TRPO (Trusted Region Policy Optimization):

$$\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t \right]$$

$$\text{subject to} \quad \hat{\mathbb{E}}_t [\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \leq \delta.$$

Идея 1: перенос ограничения на KL в штраф

$$\underset{\theta}{\text{maximize}}\, \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t - \beta \, \text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)] \right]$$

- Коэффициент β перед регуляризатором можно менять динамически
- Применяют правило: если KL меньше/больше «целевого значения KL» (параметр алгоритма), то умножаем/делим β на 2

Идея 2: clipping

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right]$$

$$\Downarrow$$

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

Proximal Policy Optimization

$$L^{KLPEN}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t - \beta\, \mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]\right]$$

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

$$(V_\theta(s_t) - V_t^{\text{targ}})^2$$

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\left[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\right]$$

Энтропия (для exploration)

Proximal Policy Optimization

Algorithm 1 PPO, Actor-Critic Style

for iteration$=1, 2, \ldots$ **do**
  for actor$=1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{\text{old}} \leftarrow \theta$
end for

PPO Paper: https://arxiv.org/pdf/1707.06347.pdf

План

1. Примеры задач

2. Базовые идеи RL

3. SARSA & Q-learning

4. Policy Gradient и идеи его улучшения

Резюме

1. RL рассматривает задачу обучения агента максимизировать награду при взаимодействии со средой
2. Наиболее популярные подходы – Deep Q-Learning и модификации Policy Gradients (PG)
3. DQN – value based и основан на уравнении Беллмана для Q-функции, PG – policy based и основан на оптимизации награды градиентным подъемом по стратегии
4. Развитие идеи PG привело к появлению Actor-Critic методов и добавлению регуляризации в оптимизируемом функционале (TRPO, PPO)

Что еще можно изучить

- Заметки курса ML Andrew Ng по Reinforcement learning
- Статьи по Q-learning и Policy Gradients
- Туториалы с применением различных стратегий для задач из Open AI Gym
- Статьи с более сложными методами:
  - https://arxiv.org/pdf/1707.06347.pdf (PPO)
  - https://proceedings.neurips.cc/paper/2017/file/facf9f743b083008a894eee7baa16469-Paper.pdf