Лекция 5: Оптимизация стратегии, алгоритмы PPO, TRPO, SAC, TD3

Панов А.И.
panov.ai@mipt.ru

Центр когнитивного моделирования МФТИ
Интситут проблем искусственного интеллекта ФИЦ ИУ РАН

2020 – Машинное обучение с подкреплением
Курс для Сбербанка

План лекции

1. Шаг градиентного спуска для вычисления стратегии
2. Алгоритм доверительных областей (TRPO)
3. Алгоритм PPO
4. Алгоритм TD3
5. Алгоритм SAC

Поиск стратегии в RL

- Поиск стратегии: прямая параметризация стратегии:

$$\pi_\theta = \mathbb{P}[a|s;\theta]$$

- Цель – найти такую стратегию π, которая имела бы наивысшее значение функции полезности V^π
- Наиболее распространены градиентные методы поиска стратегии

Базовый алгоритм градиента стратегии

Algorithm 1 Vanilla PG

function VPG
    Инициализируем параметры стратегии θ и базовый уровень B
    for итераций $i = 1, 2, \ldots,$ **do**
        набираем множество траекторий, выполняя текущую стратегию
        for каждого шага t траектории τ^i **do**
            вычисляем отдачу $R_t^i = \sum_{t'=t}^{T-1} r_{t'}$,
            $\hat{A}_t^i = R_t^i - B(s_t)$ - **оценка преимущества**,
        обновляем базовый уровень, минимизируя $\sum_i \sum_t \|B(s_t) - R_t^i\|^2$,
        обновляем оценку градиента стратегии $\hat{g} = \sum_{s_t, a_t} \nabla_\theta \log \pi_\theta(a_t|s_t) \hat{A}_t$,
        обновляем стратегию (параметры), используя оценку градиента $\hat{g}$

Градиент стратегии и скорость обучения

- Методы градиентного спуска обновляют веса на **небольшой** значение в направление градиента
- Размер шага важен в любой задаче, связанной с поиском оптимумов функции
- Обучение с учителем: если шагнуть слишком далеко, следующие обновления могут это исправить
- В обучение с подкреплением:
  - шагнув слишком далеко мы приходим к плохой стратегии,
  - следующая выборка будет собираться собрано в соответствии с плохой стратегией,
  - стратегия определяет сбор данных! (по существу контролируя исследование среды и использование накопленных данных, мы ищем золотую середину между параметрами конкретной стратегии и стохастичностью стратегии)
  - **вероятно, что плохой выбор стратегии не будет исправлен и это приведет к коллапсу в производительности!**

Подбор шага

- Простой подбор размера шага: линейный поиск в направлении градиента:
  - простой, но дорогой метод, наивный: игнорирование факта, в каких случаях линейная аппроксимация хороша и плоха
- Хотелось бы автоматически гарантировать, что новая стратегия не хуже текущей: $V^{\pi'} \geq V^{\pi}$
- Рассмотрим это как подзадачу поиска градиента стратегии

Целевая функция

- Цель: найти параметры стратегии, которые максимизируют функцию полезности:
$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R(a_t, s_t) | \pi_\theta\right],$$
где $s_0 \sim \mu(s_0), a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_a, a_t)$
- Мы можем контролировать выборку по текущей стратегии π_θ
- Но мы хотим предсказать полезность другой стратегии (обучение на основе чужого опыта)
- Выразим ожидаемую отдачу другой стратегии в терминах функции преимущества исходной стратегии
$$J(\tilde{\theta}) = J(\theta) + \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t A^\pi(a_t, s_t)\right] = J(\theta) + \sum_s \mu_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A^\pi(s, a),$$
где $\mu_{\tilde{\pi}}(s)$ определяется как дисконтированная частота состояний s по стратегии $\tilde{\pi}$
- Мы знаем A^π и $\tilde{\pi}$, но не можем вычислить выражение полностью, т.к. не известно $\mu_{\tilde{\pi}}$

Локальная аппроксимация

- Можно ли устранить зависимость от дисконтированного распределения по состояниям по новой стратегии?
- Заменим это распределением по текущей стратегии, введя новую целевую функцию:

$$L_\pi(\tilde{\pi}) = J(\theta) + \sum_s \mu_\pi(s) \sum_a \tilde{\pi}(a|s) A^\pi(s,a)$$

- При этом $L_{\pi_{\theta_0}}(\pi_{\theta_0}) = J(\theta_0)$.
- Градиент новой функции равен градиенту функции полезности стратегии, вычисленной в точке θ_0:

$$\nabla_\theta L_{\pi_{\theta_0}}(\pi_{\theta_0})|_{\theta=\theta_0} = \nabla_\theta J(\theta)|_{\theta=\theta_0}.$$

Консервативная итерация по стратегиям

- Какова полезность новой стратегии, полученной оптимизацией такой суррогатной функцией полезности?
- Рассмотрим смешанную стратегию:

$$\pi_{new}(a|s) = (1-\alpha)\pi_{old}(a|s) + \alpha\pi'(a|s)$$

- В этом случае можно гарантировать следующую нижнюю границу полезности новой стратегии π_{new}:

$$J^{\pi_{new}} \geq L_{\pi_{old}}(\pi_{new}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2,$$

где $\epsilon = \max_s \left|\mathbb{E}_{a\sim\pi'(a|s)}[A^\pi(a,s)]\right|$

- Желательно получить оценку нижней границы полезности для стохастической стратегии в общем виде

Нижняя граница полезности в общем случае

- Напомним, что $L_\pi(\tilde{\pi}) = J(\theta) + \sum_s \mu_\pi(s) \sum_a \tilde{\pi}(a|s) A^\pi(s, a)$

Theorem

Пусть $D_{TV}^{max}(\pi_1, \pi_2) = \max_s D_{TV}^{max}(\pi_1(\cdot|s), \pi_2(\cdot|s))$, *тогда*

$$J^{\pi_{new}} \geq L_{\pi_{old}}(\pi_{new}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}(D_{TV}^{max}(\pi_{old}, \pi_{new}))^2,$$

где $\epsilon = \max_{s,a} |A^\pi(a, s)|$

- $D_{TV}(p, q)^2 \leq D_{KL}(p, q)$ для logprob распределений p и q
- Следствием теоремы является следующее:

$$J^{\pi_{new}} \geq L_{\pi_{old}}(\pi_{new}) - \frac{4\epsilon\gamma}{(1-\gamma)^2} D_{KL}^{max}(\pi_{old}, \pi_{new}),$$

где $D_{KL}^{max}(\pi_1, \pi_2) = \max_s D_{KL}^{max}(\pi_1(\cdot|s), \pi_2(\cdot|s))$

Гарантии улучшения стратегии

- Цель - найти стратегию, которая максимизирует целевую функцию, определяемую нижней границей оценки:

$$M_i(\pi) = L_{\pi_i}(\pi) - \frac{4\epsilon\gamma}{(1-\gamma)^2}(D_{KL}^{max}(\pi_i, \pi))$$

$$J^{\pi_{i+1}} \geq L_{\pi_i}(\pi) - \frac{4\epsilon\gamma}{(1-\gamma)^2}(D_{KL}^{max}(\pi_i, \pi)) = M_i(\pi_{i+1})$$

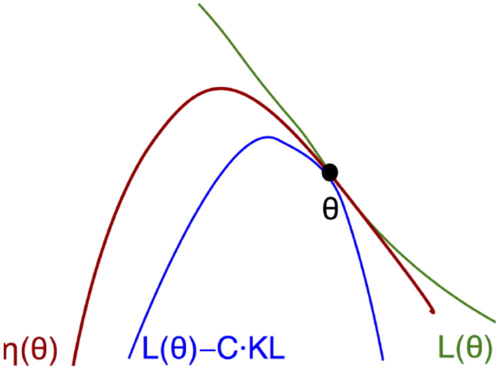
$$J^{\pi_i} = M_i(\pi_i)$$

$$J^{\pi_{i+1}} - J^{\pi_i} \geq M_i(\pi_{i+1}) - M_i(\pi_i)$$

- До тех пор, пока стратегия равна или улучшена по сравнению со старой стратегией по отношению к нижней границе, мы гарантируем монотонное улучшение стратегии!
- Один из вариантов алгоритма maximin.

Гарантии улучшения стратегии

$$J^{\pi_{new}} \geq L_{\pi_{old}}(\pi_{new}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}(D_{KL}^{max}(\pi_{old}, \pi_{new}))$$



Недостатки базового алгоритма градиента стратегии

- Сложно выбрать градиентный шаг
  - входные данные нестационарны из-за изменяющейся стратегии: распределения наблюдений и вознаграждений постоянно меняются,
  - неправильный градиентный шаг приводит к большим проблемам, чем при обучении с учителем:
    - слишком большой шаг $\rightarrow$ плохая стратегия,
    - следующая выборка будет получена по плохой стратегии,
    - ситуация ухудшается - резкое падение производительности.
- Эффективность выборок:
  - только один градиентный шаг на один эпизод (выборку из среды),
  - зависит от масштаба координат.

Оптимизация градиента

- Наша цель - оптимизировать целевую функцию

$$\max_{\theta} L_{\theta_{old}}(\theta_{new}) - \frac{4\epsilon\gamma}{(1-\gamma)^2} D_{KL}^{max}(\theta_{old}, \theta_{new}) = L_{\theta_{old}}(\theta_{new}) - C D_{KL}^{max}(\theta_{old}, \theta_{new})$$

  где C - штрафной коэффициент

- На практике при использовании коэффициента, равного значению, рекомендуемому теорией, градиентные шаги должны быть очень маленькими
- Новая идея: использовать доверительные области (trust regions) для ограничения градиентного шага. Это можно сделать, наложив ограничение на константу при KL дивергенции между новой и старой стратегией:

$$\max_{\theta} L_{\theta_{old}}(\theta)$$

$$\text{при условии } D_{KL}^{s\sim\mu_{\theta_{old}}}(\theta_{old}, \theta) \le \delta$$

- Здесь используется средняя KL вместо максимальной (максимум требует ограничения KL во всех состояниях, что приводит к слишком большому числу ограничений)

Оптимизационная задача

- Запишем оптимизационную задачу:

$$\max_{\theta} L_{\theta_{old}}(\theta)$$

$$\text{при условии } D_{KL}^{s\sim\mu_{\theta_{old}}}(\theta_{old}, \theta) \leq \delta$$

где $L_{\theta_{old}}(\theta) = J(\theta) + \sum_s \mu_{\theta_{old}}(s) \sum_a \pi_\theta(a|s) A_{\theta_{old}}(s, a)$.

- Нам не известны ни точные частоты посещений состояний, ни истинное значение функции преимущества.

Практическая реализация

- Замена частот:

$$\sum_s \mu_{\theta_{old}}(s)[\dots] \to \frac{1}{1-\gamma}\mathbb{E}_{s\sim\mu_{\theta_{old}}}[\dots]$$

- Выборка по значимости:

$$\sum_a \pi_\theta(a|s_n)A_{\theta_{old}}(s_n, a) \to \mathbb{E}_{a\sim q}\left[\frac{\pi_\theta(a|s_n)}{q(a|s_n)}A_{\theta_{old}}(s_n, a)\right]$$

  где q - некоторое выборочное распределение по множеству действий, а s_n - конкретное выбранное состояние.
- Используем выборку по значимости (importance sampling) для оценки всей суммы на основе альтернативного выборочного распределения q (отличного от новой стратегии π_θ).
- Замена функции преимущества:

$$A_{\theta_{old}} \to Q_{\theta_{old}}$$

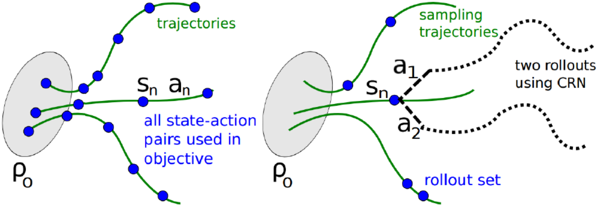
- Введенные замены не меняют решение оптимизационной задачи.

Определение выборочной стратегии

- Оптимизационная задача:

$$\max_{\theta} \mathbb{E}_{s\sim\mu_{\theta_{old}}, a\sim\pi_{\theta_{old}}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} Q_{\theta_{old}}(s,a) \right]$$

$$\text{при условии } \mathbb{E}_{s\sim\mu_{\theta_{old}}} D_{KL}(\pi_{\theta_{old}}(\cdot|s), \pi_\theta(\cdot|s)) \le \delta$$



Trust Region Policy Optimization, Schulman et al, 2015

Алгоритм TRPO

- Решаем оптимизационную задачу для суррогатной целевой функции.
- Используем дивергенцию для ограничения градиентного шага.
- Это приводит к использованию естественного градиентного шага $F^{-1}g$ при условии линейно-квадратичной аппроксимации
- Можем решить задачу для такого шага с использованием сопряженного градиентного метода

$$\max_{\theta} \sum_{n=1}^{N} \left[\frac{\pi_\theta(a_n|s_n)}{\pi_{\theta_{old}}(a_n|s_n)} Q_{\theta_{old}}(s_n, a_n) \right]$$

$$\text{при условии } \overline{KL}(\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)) \leq \delta$$

- Линейно-квадратичная аппроксимация + штраф $\rightarrow$ естественный градиент
- Без ограничений $\rightarrow$ итерация по стратегиям
- Евклидов штраф вместо KL дивергенции $\rightarrow$ базовый градиент стратегии

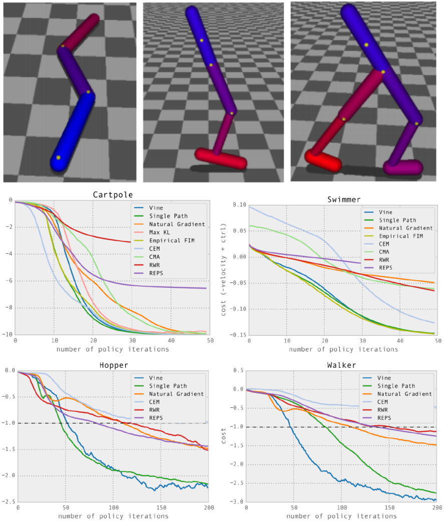
Алгоритм TRPO

Algorithm 2 TRPO

1: **for** итераций $1, 2, 3, \ldots$ **do**
2: запустить стратегию для генерации T временных шагов или N траекторий,
3: оценить функцию преимущества для всех шагов,
4: вычислить градиент стратегии g,
5: использовать сопряженный градиент для вычисления $F^{-1}g$,
6: выполнить линейный поиск на суррогатной функции потерь и KL-ограничением.

Результаты TRPO

Задача управления движением в 2D:



Недостатки TRPO

- Сложно использовать в архитектурах с несколькими выходами (головами), например для стратегии и полезности, т.к. необходимо по-разному взвешивать составляющие метрики
- На практике менее эффективен на задачах с CNN и RNN, например на играх Atari
- Сопряженный градиент (метод второго порядка) делает реализацию сложной

Оптимизация ближайшей стратегии

Вернемся к оптимизационной задаче со штрафом:

$$\max_{\theta} \sum_{n=1}^{N} \left[\frac{\pi_\theta(a_n|s_n)}{\pi_{\theta_{old}}(a_n|s_n)} A_{\theta_{old}}(s_n, a_n) \right] - \beta \overline{KL}_{\pi_{\theta_{old}}}(\pi_\theta(\cdot|s_t))$$

Algorithm 3 PPO

1: **for** итераций $1, 2, 3, \ldots$ **do**
2: запустить стратегию для генерации T временных шагов или N траекторий,
3: оценить функцию преимущества для всех шагов,
4: выполнить стохастический градиентный спуск для функции потерь для нескольких эпох,
5: если KL слишком высока - увеличиваем β, если KL слишком низка - уменьшаем β.

PPO (proximal policy optimization) - примерно та же эффективность, но только с оптимизацией первого порядка (PPO penalty)

Усеченная целевая функция для PPO

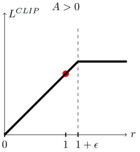
- Суррогатная целевая функция:

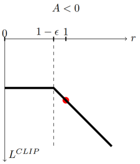
$$L^{CPI}(\theta) = \mathbb{E}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right] = \mathbb{E}_t\left[r_t(\theta)\hat{A}_t\right]$$

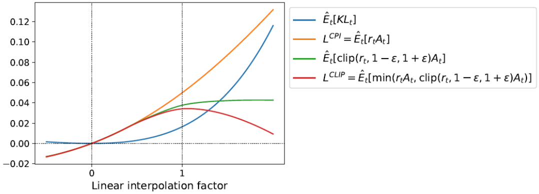
- Определим нижнюю границу с помощью усеченной коэффициентов важности:

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

- Это позволяет определить пессимистическую границу целевой функции, которую можно оптимизировать с помощью стохастического градиентного спуска.







Алгоритм PPO

Algorithm 4 PPO

1: **for** итераций $1, 2, 3, \ldots$ **do**
2: запустить стратегию для генерации T временных шагов или N траекторий,
3: оценить функцию преимущества для всех шагов,
4: выполнить стохастический градиентный спуск для функции потерь $L^{CLIP}(\theta)$ для нескольких эпох,

- Работает немного лучше, чем TRPO на задачах управления, и существенно лучше на играх Atari.
- Совместим с несколькими выходами сетей и с RNN.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. 2017

Развитие DDPG

- Алгоритм DDPG показывает очень хорошие результаты в средах с непрерывным пространством действий, однако очень чувствителен при подборе гиперпараметров
- Часто оказывается, что в процессе обучения DDPG начинается очень сильно переоценивать Q-функцию, а это, в свою очередь, ведет к деградации стратегии
- В алгоритме двойного отложенного глубокого градиента детерменированой стратегии (Twin Delayed DDPG, TD3 - `https://arxiv.org/abs/1802.09477`) предлагается исправить этот недостаток
- Как и DDPG TD3 использует отложенные прецеденты (off-policy)
- Применяется только для окружений с непрерывным множеством действий

Основные особенности TD3

- Усеченное двойное Q-обучение: использование двух Q-функций, вместо одной (twin)
- При вычислении TD-показателя в уравнении Беллмана используется наименьшее из двух Q значений
- Ориентировочные (target) Q-функции обновляются по скользящему среднему
- Обновлении стратегии идет реже, чем обновление Q-функций (например, в два раза реже) (delayed)
- Добавление шума к ориентировочному (target) действию, что уменьшает влияние ошибок в Q-функции на стратегию (smoothing)

Реализация алгоритма TD3

- **Сглаживание ориентировочной стратегий**: TD-ошибка Q-обучения подсчитывается с использованием действий, генерируемых ориентировочной стратегий μ_{θ^-}, с добавлением усеченного шума:

$$a'(s') = \operatorname*{clip}_{a_{\min}, a_{\max}}(\mu_{\theta^-}(s') + \operatorname*{clip}_{-c,c} \epsilon), \ \epsilon \sim \mathcal{N}(0, \sigma)$$

- **Усеченное двойное Q-обучение**: обе Q-функции используют один и тот же TD-показатель:

$$y(r, s', d) = r + \gamma \min_{i=1,2} Q_{w_i^-}(s', a'(s'))$$

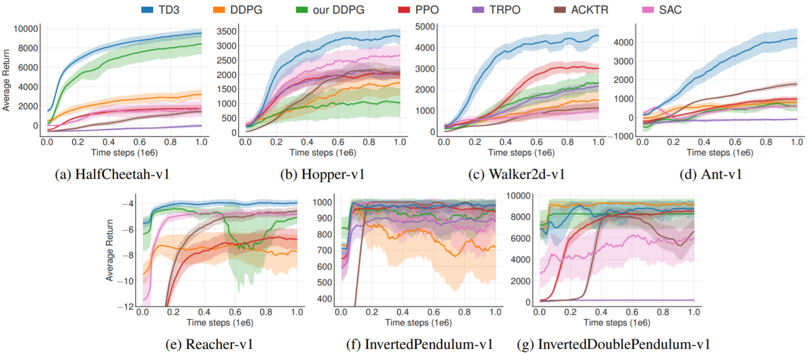
- Используется стандартная квадратичная ошибка:

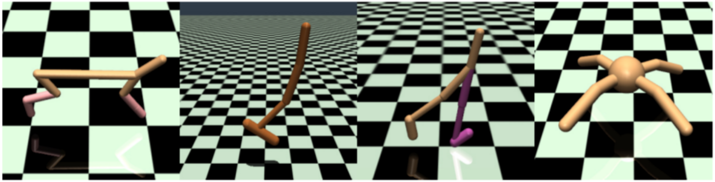
$$L(w_i, \mathcal{D}) = \mathop{\mathbb{E}}_{(s,a,r,s' \sim \mathcal{D})}\left[\left(Q_{w_i}(s, a) - y(r, s')\right)^2\right]$$

- Стратегия обновляется в результате максимизации Q_{w_1}:

$$\max_{\theta} \mathop{\mathbb{E}}_{s \sim \mathcal{D}}[Q_{w_1}(s, \mu_\theta(s))]$$

Результаты TD3





Улучшение исследования среды

Реализация встроенного механизма поддержки исследования среды – алгоритм мягкого актора-критика SAC (Soft actor-critc `https://arxiv.org/abs/1801.01290`):

- оптимизирует стохастическую стратегию с использованием отложенных прецедентов,
- использует идею энтропийной регуляризации,
- неявно реализует идею поддержания исследования среды,
- представляет собой переходный вариант между алгоритмами стохастической оптимизации и DDPG,
- чаще всего применяется в средах с непрерывным пространством действий.

Идея энтропийной регуляризации

- Энтропия - мера неопределенности.
- Формально энтропия H случайной величины x с плотностью вероятности P:
$$H(P) = \mathbb{E}_{x \sim P}[-\log P(x)]$$
- В RL энтропия учитывается как псевдовознаграждение агенту, пропорциональное неопределенности оценки стратегии:
$$\pi = \arg\max_{\pi} \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t, s_{t+1}) + \alpha H(\pi(\cdot|s_t))\right)\right]$$
- В соответствии с этим меняется определение Q-функции и уравнение Беллмана:
$$Q^{\pi}(s, a) = \mathbb{E}_{\substack{\tau \sim \pi \\ s_0 = s \\ a_0 = a}}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) + \alpha \sum_{t=1}^{\infty} \gamma^t H(\pi(\cdot|s_t))\right]$$
$$Q^{\pi}(a, s) = \mathbb{E}_{\substack{s' \sim P \\ a' \sim \pi}}[r(s, a, s') + \gamma(Q^{\pi}(s', a') + \alpha H(\pi(\cdot|s')))]$$

SAC: отличия от TD3

- Мягкий актор-критик одновременно обновляет стратегию π_θ и две Q-функции Q_{w_1} и Q_{w_2}
- Параметр регуляризации α может быть как постоянен, так и менять со временем
- Обновление Q-функций идет по аналогии с TD3 с небольшими отличиями:
  - Q-показатель вычисляется с учетом энтропии,
  - действие в следующем состоянии генерируется не ориентировочной стратегией, в текущей,
  - нет явного сглаживания ориентировочной стратегии, т.к. текущая стратегия и так стохастическая

$$
\begin{aligned}
Q^\pi(s,a) &= \mathop{\mathbb{E}}_{\substack{s'\sim\mathcal{D}\\ \color{red}{a'\sim\pi}}}[r(s,a,s') + \gamma(Q^\pi(s',a') + \alpha H(\pi(\cdot|s')))] = \\
&= \mathop{\mathbb{E}}_{\substack{s'\sim\mathcal{D}\\ \color{red}{a'\sim\pi}}}[r(s,a,s') + \gamma(Q^\pi(s',a') - \alpha \log \pi(\cdot|s'))]
\end{aligned}
$$

Алгоритм SAC

- Обновление -функции идет аналогично TD3 (усечение, среднеквадратичная ошибка, ориентировочные значения):

$$y(r, s') = r + \gamma \min_{i=1,2} Q_{w_i^-}(s', a') - \alpha \log \pi_\theta(a'|s'), \ a' \sim \pi_\theta(\cdot|s')$$

$$L(w_i, \mathcal{D}) = \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{D}} \left[\left(Q_{w_i^-} - y(r, s') \right)^2 \right]$$

- Стохастическая стратегия является результатам репараметризацией – вычисления функции от параметров и гауссова шума:

$$a_\theta(s, \xi) = \tanh(\mu_\theta(s) - \sigma_\theta(s) \odot \xi), \ \xi \sim \mathcal{N}(0, I)$$

- Стратегия обновляется путем максимизации ожидаемой отдачи с учетом энтропийного фактора и шума:

$$\max_\theta \mathop{\mathbb{E}}_{\substack{s\sim\mathcal{D}\\ \xi\sim\mathcal{N}}} \left[\min_{i=1,2} Q_{w_i}(s, a_\theta(s, \xi)) - \alpha \log \pi_\theta(a_\theta(s, \xi)|s) \right]$$

Результаты SAC