Введение в обучение с подкреплением

Тема 14: Алгоритмы DDPG, TD3, SAC

Лектор: Кривошеин А.В.

Фреймворк stable_baseline3

Реализованные алгоритмы:

Name	Box	Discrete	MultiDiscrete	MultiBinary	Multi Processing
ARS [1]	✔	✔	✘	✘	✔
A2C	✔	✔	✔	✔	✔
CrossQ [1]	✔	✘	✘	✘	✔
DDPG	✔	✘	✘	✘	✔
DQN	✘	✔	✘	✘	✔
HER	✔	✔	✘	✘	✔
PPO	✔	✔	✔	✔	✔
QR-DQN [1]	✘	✔	✘	✘	✔
RecurrentPPO [1]	✔	✔	✔	✔	✔
SAC	✔	✘	✘	✘	✔
TD3	✔	✘	✘	✘	✔
TQC [1]	✔	✘	✘	✘	✔
TRPO [1]	✔	✔	✔	✔	✔
Maskable PPO [1]	✘	✔	✔	✔	✔

О новых методах

Метод DQN является методом обучения с разделённой стратегией (**обучение по отложенному опыту**).
Однако, это метод имеет ряд особенностей и недостатков.

1. Метод не всегда позволяет добиться сходимости.
2. DQN — это не прямой подход к достижению цели агента, он обучается через оптимальную *Q*-функцию.
3. DQN подход работает только для сред с **дискретным** набором действий.

Методы на основе градиента стратегии (REINFORCE, A2C, PPO):

1. непосредственным образом занимаются максимизацией дохода, путём изменения вероятностей выбора действий, то есть обучаются хорошей стратегии действий.

2.работают как в средах с дискретным набором действий, так и с **непрерывным** пространством действий.

3. являются методами обучения по актуальному опыту (**прошлый опыт нельзя использовать** при обучении).

Ниже рассмотрим три метода обучения агента, которые позволяют совместить подходы с обучением по отложенному опыту и непосредственным обучением стратегии.

Метод DDPG

Рассмотрим метод **DDPG (англ. Deep Deterministic Policy Gradients).**

DDPG является развитием DQN, позволяющим использовать DQN подход в средах с непрерывным пространством действий. С другой стороны, DDPG является развитием методов на основе градиента стратегии, добавляя им возможность использовать отложенный опыт для обучения.

Суть DQN: строится приближение оптимальной Q-функции, причём Q-функция реализована как ИНС, принимающая на вход состояния s и формирующая вектор значений Q-функции $\left(\hat{q}(s, a_1; \theta), \ ..., \hat{q}(s, a_N; \theta)\right)$ на выходе, где N — это число возможных действий в состоянии s.

DQN является методом обучения по отложенному опыту. Для обновления параметров требуется сделать один шаг по траектории, получить значения (s, a, r, s') и сдвинуть параметры для минимизации функции ошибки

$$J(\theta) = \frac{1}{2}\left(r + \gamma \max_{a'} \hat{q}(s', a'; \theta^-) - \hat{q}(s, a; \theta)\right)^2.$$

Набор параметров θ^- является “замороженной” на некоторое время копией параметров θ. Эти параметры θ^- обновляются копированием параметров θ раз в некоторое число итераций.

Формула обновления по методу DQN имеет вид:

$$\theta \leftarrow \theta + \alpha \left(r + \gamma \max_{a'} \hat{q}(s', a'; \theta^-) - \hat{q}(s, a; \theta)\right) \mathrm{grad}_\theta \, \hat{q}(s, a; \theta).$$

При этом можно использовать буфер памяти для хранения и использования прошлого опыта взаимодействия.

Метод DDPG

При DQN агент формирует траектории в среде по ε-жадной стратегии относительно текущей Q-функции. Обучается же агент сразу целевой оптимальной стратегии π^*. Хорошо обученная Q-функция близка к оптимальной Q-функции. Связь между оптимальной стратегией и оптимальной Q-функцией имеет вид:

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

В этом смысле целевое значение для обновления параметров сети имеет вид

$$r + \gamma \max_{a'} \hat{q}(s', a'; \theta^-) = r + \gamma \hat{q}(s', \boldsymbol{\pi^*(s')}; \theta^-).$$

Для метода DQN надо уметь **быстро** считать целевые значения, так как обучение проводится часто.

В случае дискретного пространства действий это так:

вычисляется выход ИНС и выбирается максимальное значение.

В случае **непрерывного пространства действий** поиск значения $\max_{a'} \hat{q}(s', a'; \theta^-)$ значительно усложняется.

1. Нет возможности найти $\hat{q}(s', a'; \theta^-)$ для всех возможных действий a', так как их континуум.

2. Нет возможности легко получить жадную стратегию по Q-функции: $\pi^*(s) = \operatorname*{argmax}_a \hat{q}(s, a; \theta)$. Для поиска максимума требуется формально решать некоторую оптимизационную задачу для каждого состояния, что вычислительно затратно.

Метод DDPG

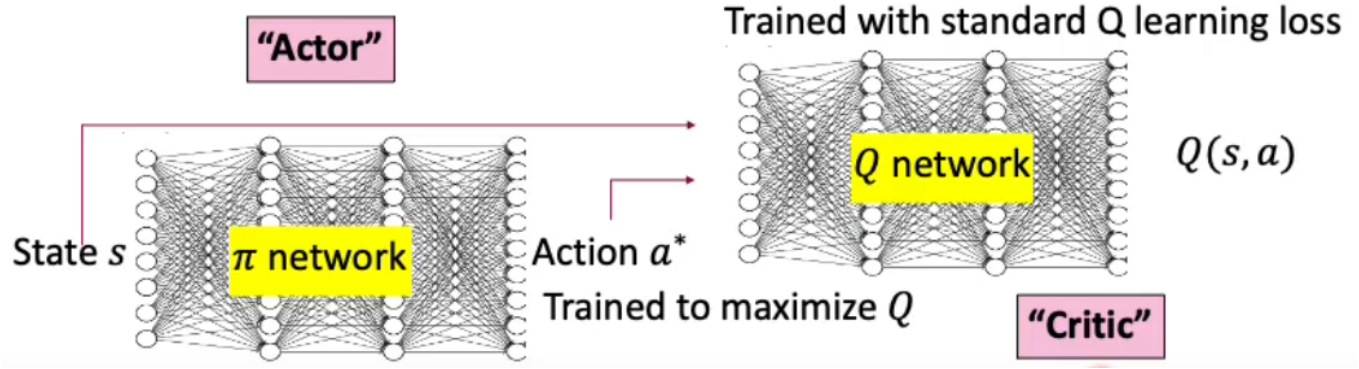
Идея DDPG (2016): обучать отдельную ИНС со своим набором параметров ω, которая по текущему состоянию s на входе, будет выдавать результат решения оптимизационной задачи:

$\pi_\omega(s) = \underset{a}{\operatorname{argmax}}\, \hat{q}(s, a; \theta)$. **Задача этой ИНС**: формировать жадную относительно Q – функции стратегию.

Таким образом, есть две ИНС, как в методе Актор-Критик. Однако, теперь механизм работы этой пары немного иной.

Актор — это модель для стратегии π_ω, которая приближает жадную стратегию относительно текущей Q-функции, $a = \pi_\omega(s)$.

Критик — это модель для Q-функции, которая приближает оптимальную Q-функцию, на вход сети подаётся пара состояние-действие, результатом является число $\hat{q}(s, a; \theta)$.



Метод DDPG

При обучении методом DQN опыт можно генерировать по любой стратегии, в том числе Актор может генерировать эти стратегии. Если Актор генерирует стратегии, то это должны быть исследовательские стратегии. Чтобы это было так, надо в состоянии s совершать зашумлённое действие $\pi_\omega(s) + \xi$, где $\xi \sim \mathcal{N}(0, \sigma)$.

Шаг обучения: чтобы обучить сеть Критика, то есть сеть $\hat{q}(s, a; \theta)$, используется DQN подход и цель обновления формируется по четвёрке (s, a, r, s') с помощью сети Актора

$r + \gamma\ \hat{q}(s', \pi_\omega(s'); \theta^-)$

Формула обновления параметров для Критика по (s, a, r, s') в итоге имеет вид

$\theta \leftarrow \theta + \alpha\left(r + \gamma\, \hat{q}(s', \pi_\omega(s'); \theta^-) - \hat{q}(s, a; \theta)\right) \mathrm{grad}_\theta\, \hat{q}(s, a; \theta).$

“Замороженные” параметры θ^- обновляются по параметрам θ с помощью **soft update** (также polyak averaging или exponential averaging)

$\theta^- \leftarrow \tau\, \theta + (1 - \tau)\, \theta^-$ с некоторым коэффициентом τ.

Метод DDPG

Для обучения сети Актора π_ω решается задача максимизации Q-функции:

$J^Q(\omega) = \max_\omega \mathbb{E}[Q(S_t, \pi_\omega(S_t))]$.

В нашем случае максимизируется приближение Q – функции в текущей точке на траектории

$J^Q(\omega) \approx \max_\omega \hat{q}(s, \pi_\omega(s); \theta)$.

Формула обновления параметров будет иметь вид:

$\omega \leftarrow \omega + \alpha \operatorname{grad}_\omega J^Q(\omega)$, где $\operatorname{grad}_\omega J^Q(\omega) \approx \operatorname{grad}_a \hat{q}(s, a; \theta)|_{a=\pi_\omega(s)} \cdot \operatorname{grad}_\omega \pi_\omega(s)$.

Для сети Актора π_ω можно также использовать два набора параметров ω и ω^-.

Параметры ω будут использоваться для формирования траектории и ИНС будет обучаться.

Параметры ω^- будут использоваться при формировании цели обновления для Q-функции, то есть цель обновления имеет вид

$r + \gamma\ \hat{q}(s', \pi_{\omega^-}(s'); \theta^-)$.

Параметры ω^- обновляются по параметрам ω с помощью **soft update**.

Шаг обновления параметров этих двух ИНС можно делать после каждого шага по траектории. Но более эффективно использовать буфер памяти, извлекая оттуда пакет обучающих примеров.

Метод DDPG

1. Инициализировать

γ, λ, ИНС $\hat{q}(s, a; \theta)$, $\pi_\omega(s)$, $\theta^- = \theta$, $\omega^- = \omega$, буфер памяти D

2. Q-learning

Повторять для каждого эпизода:

Выбрать начальное состояние s

Повторять:

Выбрать $a = \pi_\omega(s)+\xi$, $\xi \sim \mathcal{N}(0, \sigma)$

Наблюдать r, s' и сохранить в буфер (s,a,r,s')

Выбрать батч из буфера памяти D

(s,a,r,s')~D

$a' = \pi_{\omega^-}(s')$

$\theta \leftarrow \theta + \alpha\left(r + \gamma\, \hat{q}(s', a'; \theta^-) - \hat{q}(s, a; \theta)\right) \mathrm{grad}_\theta\, \hat{q}(s, a; \theta)$

$\omega \leftarrow \omega + \alpha\, \mathrm{grad}_a\, \hat{q}(s, a; \theta)\big|_{a=\pi_\omega(s)} \cdot \mathrm{grad}_\omega\, \pi_\omega(s)$

Обновить θ^-, ω^-

$s := s'$

Если s заключительное состояние, то выйти из цикла.

С течением времени можно снижать шум.

Метод TD3

Метод DDPG часто может достигать отличной производительности, но он требует тонкой настройки гиперпараметров.

Одной из проблем для DDPG является то, что обучаемая Q-функция может сильно завышать значения Q-функции, что приводит к изменениям стратегии, которая использует эти завышенные значения в Q-функции.

Модификацией DDPG является метод **TD3** или **Twin Delayed DDPG**.

Алгоритм TD3 решает указанную выше проблему, вводя три приёма:

1. Алгоритм TD3 обучает двух Критиков вместо одного и для формирования целевого значения используется меньшее из двух значений от критиков.
2. Уменьшенная скорость обновления стратегии (например, одно обновление стратегии на два обновления Q-функции).
3. TD3 добавляет шум к действию, которое выбирается для формирования цели обновления. Например,

$a'(s') = \text{clip}(\pi_\omega(s') + \text{clip}(\nu, -c, c), a_{\text{Low}}, a_{\text{High}}), \quad \nu \sim \mathcal{N}(0, \sigma)$,

если действие должно быть в интервале $[a_{\text{Low}}, a_{\text{High}}]$.

Метод TD3

Веса Критиков будем обозначать θ_1, θ_2 и для формирования целевых значений используются веса θ_1^-, θ_2^-. Веса Актора — это ω и для целевых значений используются веса ω^-.

При обучении сетей Критиков формируется единое целевое значение для обновления в виде

$$r + \gamma \min_{i=1,2} \hat{q}(s', a'(s'); \theta_i^-), \quad \text{где } a'(s') = \text{clip}(\pi_{\omega^-}(s') + \text{clip}(\nu, -c, c), a_{\text{Low}}, a_{\text{High}}), \quad \nu \sim \mathcal{N}(0, \sigma),$$

и обе сети сдвигают свои значения в сторону этой цели минимизируя разность:

$$J(\theta_i) = \left(r + \gamma \min_{i=1,2} \hat{q}(s', a'(s'); \theta_i^-) - \hat{q}(s, a; \theta_i)\right)^2, \quad i = 1, 2.$$

При обучении Актора всё также требуется максимизация

$$J^Q(\omega) \approx \max_{\omega} \hat{q}(s, \pi_\omega(s); \theta_1).$$

Параметры ω^-, θ_1^-, θ_2^- обновляются по параметрам ω, θ_1, θ_2 с помощью soft update.
Указанные улучшения значительно повышают эффективность обучения.

Метод TD3 всё также является методом обучения по отложенному опыту и используется для сред с непрерывным пространством действий.

Метод TD3

1. Инициализировать

γ, λ, ИНС $\hat{q}(s, a; \theta_i)$, $\pi_\omega(s)$, $\theta_i^- = \theta_i$, i=1,2, $\omega^- = \omega$, буфер памяти D, N — насколько часто обновлять стратегию

2. Q-learning

Повторять для каждого эпизода:

Выбрать начальное состояние s

Повторять:

Выбрать $a = \pi_\omega(s)+\xi$, $\xi \sim \mathcal{N}(0, \sigma)$

Наблюдать r, s' и сохранить в буфер (s,a,r,s')

Выбрать батч из буфера памяти D

$(s,a,r,s')\sim D$

$a' = \text{clip}\left(\pi_{\omega^-}(s')\right) + \text{clip}(v, -c, c), a_{\text{Low}}, a_{\text{High}}), \quad v \sim \mathcal{N}(0, \sigma)$

$\theta_i \leftarrow \theta_i + \alpha\left(r + \gamma \min_{i=1,2} \hat{q}(s', a'(s'); \theta_i^-) - \hat{q}(s, a; \theta_i)\right) \text{grad}_{\theta_i} \hat{q}(s, a; \theta_i)$

Для каждого N-го эпизода

$\omega \leftarrow \omega + \alpha \, \text{grad}_a \hat{q}(s, a; \theta_1)\big|_{a=\pi_\omega(s)} \cdot \text{grad}_\omega \, \pi_\omega(s)$

Обновить θ_i^-, ω^-

$s := s'$

Если s заключительное состояние, то выйти из цикла.

Метод SAC

Обсудим метод **Soft Actor-Critic**. Слово “soft” значит, что в формулах применяется регуляризации энтропии.

Для пояснения базовой идеи обсудим подход к формированию основных понятий **RL с регуляризацией энтропии.**

Энтропия случайной величины с плотностью распределения P равна

$$H(P) = \mathop{\mathbb{E}}_{x\sim P}[-\ln P(x)].$$

RL с регуляризацией энтропии означает, что агент получает дополнительное вознаграждение за каждый шаг, пропорциональный величине энтропии стохастической стратегии агента в этот шаг.

То есть цель агента в построении стратегии

$$\pi^* = \operatorname*{argmax}_{\pi} \mathbb{E}_{\pi}\Big[\sum_{t=0}^{T}\gamma^t\big(R_{t+1} + \alpha\, H(\pi(\cdot\,|\,S_t))\big)\Big] = \operatorname*{argmax}_{\pi} \mathbb{E}_{\pi}\Big[G_0 + \alpha\sum_{t=0}^{T}\gamma^t\, H(\pi(\cdot\,|\,S_t))\Big].$$

Регуляризация энтропии нужна затем, чтобы стимулировать агента совершать исследования.

Функция ценности состояний также включает энтропию:

$$v_\pi(s) = \mathbb{E}_{\pi}\Big[\sum_{t=0}^{T}\gamma^t\big(R_{t+1} + \alpha\, H(\pi(\cdot\,|\,S_t))\big)\,\Big|\,S_0 = s\Big]$$

Аналогичным образом меняется функция ценности пар состояние-действие:

$$q_\pi(s, a) = \mathbb{E}_{\pi}\Big[R_1 + \sum_{t=1}^{T}\gamma^t\big(R_{t+1} + \alpha\, H(\pi(\cdot\,|\,S_t))\big)\,\Big|\,S_0 = s, A_0 = a\Big]$$

Метод SAC

Можно выписать уравнение Беллмана для изменённой Q-функции:

$$q_\pi(s, a) = \mathop{\mathbb{E}}_{\substack{r'\, s'\sim\text{Model} \\ a'\sim\pi}} \Big[r + \gamma \big(q_\pi(s', a') + \alpha\, H(\pi(\cdot \mid s')) \big) \Big| S_0 = s, A_0 = a \Big] =$$

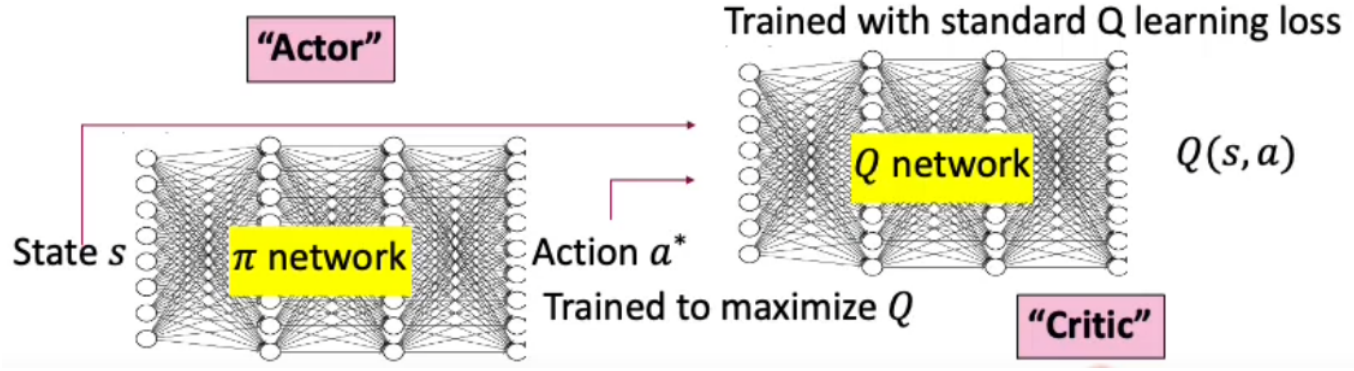
$$\mathop{\mathbb{E}}_{\substack{r'\, s'\sim\text{Model} \\ a'\sim\pi}} \Big[r + \gamma \big(q_\pi(s', a') - \alpha \ln(\pi(a' \mid s')) \big) \Big| S_0 = s, A_0 = a \Big].$$

Для четвёрки значений (s, a, r, s') оценку математического ожидания можно получить в виде

$$Q(s, a) \approx r + \gamma \big(Q(s', a') - \alpha \ln \pi(a' \mid s') \big), \text{ где } a' \sim \pi(\cdot \mid s')$$

Далее, по аналогии с методом Q-learning можно сформулировать, например, метод soft Q-learning.

Метод SAC является по сути добавлением в метод TD3 регуляризации энтропии.



Ранее, в методах A2C мы добавляли слагаемое с энтропией в функцию ошибки для поощрения исследования.

Теперь регуляризация энтропии введена в формулы для МППР. Классические алгоритмы и soft алгоритмы могут сходиться к разным решениям.

Метод SAC

Как и в TD3, метод SAC использует два Критика. **Единое значение цели обновления** формируется с помощью замороженных на время весов в виде

$$r + \gamma\left(\min_{i=1,2} \hat{q}(s', a'; \theta_i^-) - \alpha \ln \pi_\omega(a' \mid s')\right), \quad \text{где } a' \sim \pi_\omega(\cdot \mid s')$$

и обе сети сдвигают свои значения в сторону этой цели минимизируя разность:

$$J(\theta_i) = \left(r + \gamma\left(\min_{i=1,2} \hat{q}(s', a'; \theta_i^-) - \alpha \ln \pi_\omega(a' \mid s')\right) - \hat{q}(s, a; \theta_i)\right)^2, \quad i = 1, 2.$$

Актор использует стохастическую стратегию для выбора действий π_ω. Она же используется для формирования цели обновления. Задача Актора схожа с задачей Актора для TD3, то есть Актор должен выдавать такое распределение вероятностей выбора действий $\pi_\omega(\cdot \mid s)$, которое максимизирует значение soft Q-функции, то есть

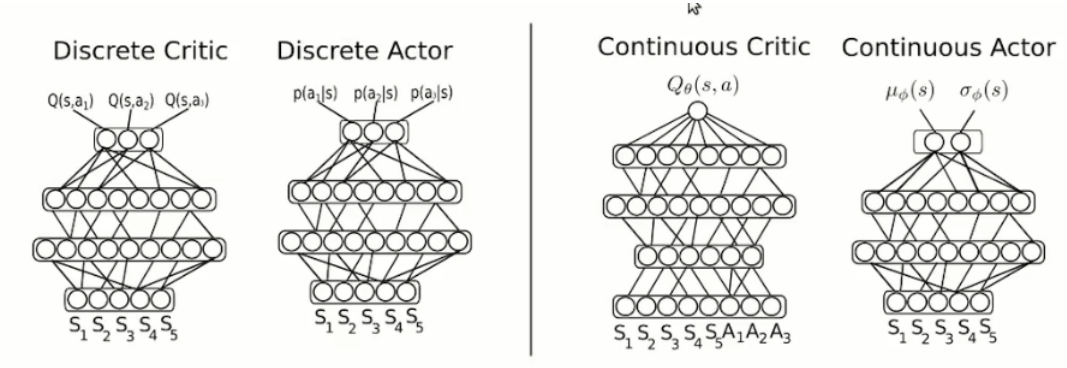
$$J^Q(\omega) = \max_\omega \mathbb{E}_{a \sim \pi_\omega(\cdot|s)}\left[Q(s, a) + \alpha H(\pi_\omega(\cdot \mid s))\right]$$

Метод SAC

Проблема 1: в TD3 Актор возвращал сами действия, здесь же требуется возвращать и распределение вероятностей выбора действий, так как нужно искать энтропию этого распределения.

$J^Q(\omega) = \max_\omega \mathbb{E}_{a\sim\pi_\omega(\cdot|s)}\left[Q(s,a) + \alpha H(\pi_\omega(\cdot|s))\right].$

В дискретном случае Актор может возвращать вероятности выбора действий, а Критик вектор значений Q-функции для различных действий.



В непрерывном же случае, Критик возвращает значение Q-функции для пары (s,a) на входе. А Актор $\pi_\omega(s)$ будет возвращать характеристики вероятностного распределения выбора действий. Для SAC используется “сдавленная” гауссиана (squashed Gaussian)

$a = \tanh(\xi)$, где $\xi \sim \mathcal{N}(\mu_\omega, \sigma_\omega)$, где $\mu_\omega, \sigma_\omega$ результат работы Актора.

Использование сдавленной гауссианы удобно ещё в том смысле, что часто действия лежат в некотором интервале, а гауссиана может выдавать формально значения по всей прямой.

Метод SAC

Проблема 2. Как искать

$$\max_{\omega} \mathbb{E}_{a\sim\pi_\omega(\cdot|s)}[Q(s, a)]$$

В TD3 действие было результатом работы ИНС Актора, и можно было брать градиент от $Q(s, a)$ как от сложной функции.
В SAC ИНС Актора возвращает параметры вероятностного распределения и здесь на ясно, как искать

$$\mathrm{grad}_\omega \mathbb{E}_{a\sim\pi_\omega(\cdot|s)}[Q(s, a)].$$

В SAC используется ре-параметризация распределения. А именно, действия выбираются по формуле

$$a = \tanh(\mu_\omega + \eta\,\sigma_\omega), \quad \text{где } \eta\sim\mathcal{N}(0, 1).$$

Тогда

$$\mathbb{E}_{a\sim\pi_\omega(\cdot|s)}[Q(s, a)] = \mathbb{E}_{\eta\sim\mathcal{N}(0,1)}[Q(s, \tanh(\mu_\omega + \eta\,\sigma_\omega))].$$

Теперь можно искать: $\mathrm{grad}_\omega\mathbb{E}_{\eta\sim\mathcal{N}(0,1)}[Q(s, \tanh(\mu_\omega + \eta\,\sigma_\omega))] = \mathbb{E}_{\eta\sim\mathcal{N}(0,1)}[\,\mathrm{grad}_\omega\, Q(s, \tanh(\mu_\omega + \eta\,\sigma_\omega))]$

Метод SAC

Существует также модификация, которая позволяет автоматически подбирать коэффициент энтропии α в $J^Q(\omega) = \max_\omega \mathbb{E}_{a\sim\pi_\omega(\cdot|s)}\big[Q(s,a) + \alpha H(\pi_\omega(\cdot\,|s))\big]$.

Метод SAC часто показывает одни из лучших результатов для различных задач.