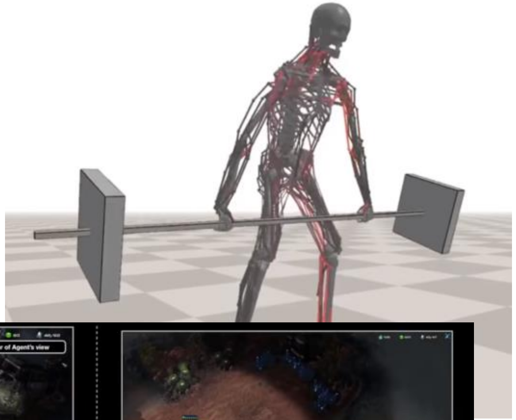
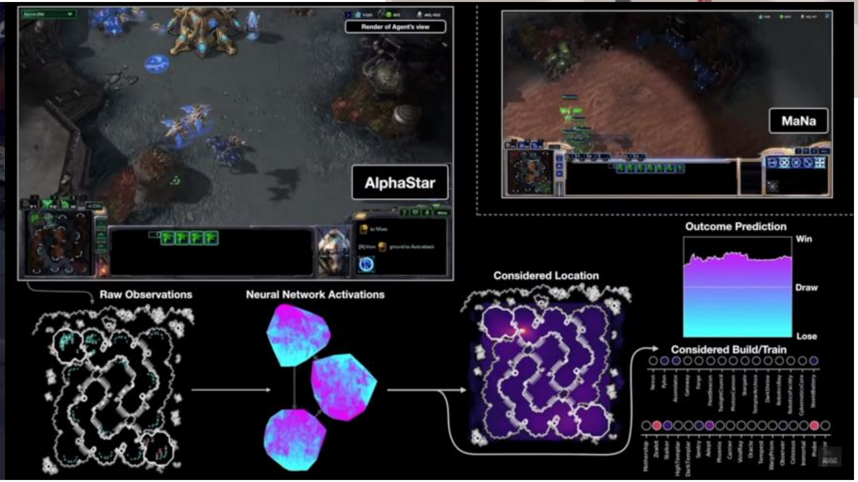
Обучение с подкреплением

On-policy Deep RL

Напоминание: ограничения

1) ~~Должны знать T и R~~
2) ~~Дискретное пространство состояний~~
3) Дискретное пространство действий





Напоминание: Алгоритм Actor-Critic

Актор - нейронная сеть, определяющая политику. Важно, чтобы можно было посчитать логарифм вероятности распределения действий, совершаемых политикой.

Критик - нейронная сеть, используемая для приближения value-функции.

Алгоритм:

1. Инициализировать критика и актора
2. В течении **T** итераций:
    1. Получить **N** эпизодов из среды
    2. Посчитать приближение **R** суммарной награды или advantage
    1. Обновить сеть актора с помощью градиента:
$$R(s_i, a_i)\nabla_\theta \log \pi_\theta(a_i|s_i)$$
    1. Обновить сеть критика

Напоминание: Алгоритм Actor-Critic

Для определения суммарной дисконтированной награды можно использовать разные функции:

1. One-step: $R_t = r_t + \gamma V(s_{t+1})$
1. N-step: $R_{t:t+n-1} = \sum_{i=t}^{t+n-1} \gamma^i r_i + \gamma^n V(s_{t+n})$
1. Lambda-return: $R_t^\lambda = (1-\lambda)[\sum_{i=1}^{T} \lambda^{i-1} R_{t:t+i}] + \lambda^T R_{t:t+T}$

Напоминание: проблемы policy gradient

REINFORCE:

1) Оценка Монте-Карло требует большого количества сэмплов для сходимости
2) Высокая чувствительность к learning rate

Actor-Critic:

1) Высокая чувствительность к learning rate
2) Оценка с помощью приближенной value-функции может иметь высокую ошибку. Это частично исправляется с помощью lambda-return

Чувствительность к Learning Rate?

На самом деле даже маленькое значение learning rate не гарантирует сходимость.

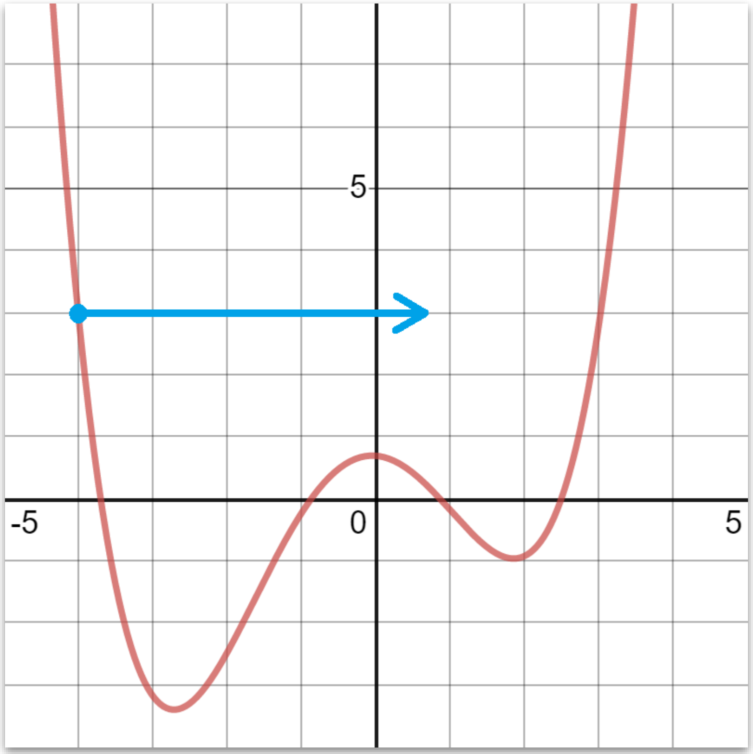
Проблема в том, что мы обновляем веса с помощью $R(s_i, a_i)\nabla_\theta \log \pi_\theta(a_i|s_i)$

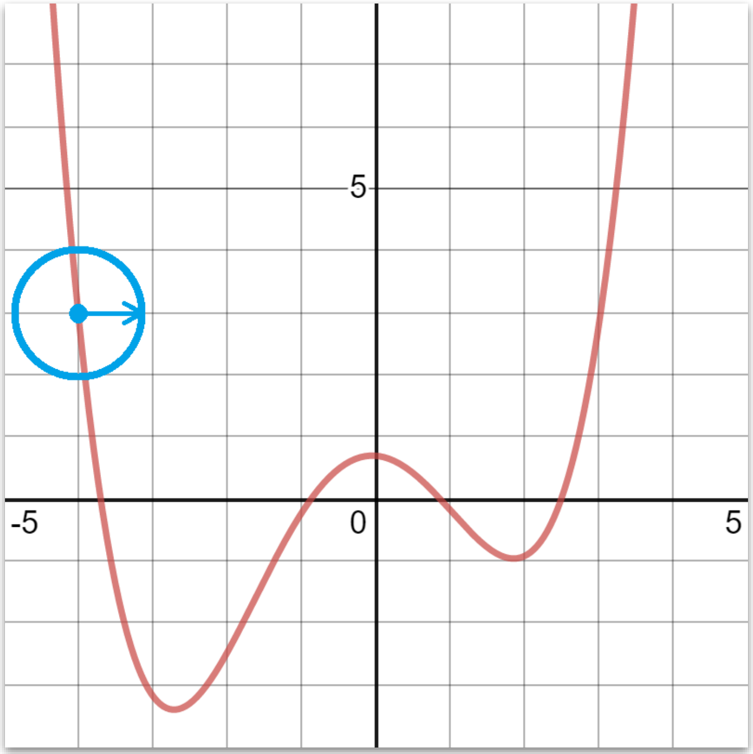
Пример 1: Награда равна +1 за правильное действие и -1 за неправильное действие

Пример 2: Награда равна +100 за правильное действие и -100 за неправильное

Пример 3: Награда равна +1001 за правильное действие и +999 за неправильное

Идея: ограничим изменение политики





Формулировка задачи

Раньше мы просто максимизировали мат. ожидание суммарной награды:

$$\mathbb{E}_{\tau|\pi_\theta} R(\tau) \to \max$$

Теперь будем решать задачу с ограничением возможного изменения политики:

$$\mathbb{E}_{\tau|\pi_\theta} R(\tau) \to \max$$
$$\text{s.t. } D_{KL}(\theta||\theta_{\text{old}}) \leq \delta$$

Где $D_{KL}(\theta||\theta_{\text{old}}) = \mathbb{E}_{s|\pi_{\theta_{old}}} D_{KL}(\pi_\theta(.|s)||\pi_{\theta_{old}}(.|s)))$

Напоминание: Дивергенция Кульбака-Лейблера отражает то, насколько далеки два распределения

$$D_{KL}(p||q) = \mathbb{E}_{x\sim p} \log \frac{p(x)}{q(x)}$$

Считаем мат.ожидание награды

В алгоритме A2C мы использовали Log Derivative Trick и Monte Carlo Sampling чтобы приблизить градиент мат. ожидания advantage:

$$\mathbb{E}_{s,a\sim\pi_\theta} A_{\pi_\theta}(s,a)$$

Однако поскольку теперь есть ограничение изменения политики, то мы можем использовать Importance Sampling:

$$\mathbb{E}_{s,a\sim\pi_\theta} A_{\pi_\theta}(s,a) \approx \mathbb{E}_{s,a\sim\pi_{\theta_{old}}} \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\pi_{\theta_{old}}}(s,a)$$

Это позволит нам использовать данные из предыдущей версии политики.

Считаем мат.ожидание награды

Это называется **surrogate advantage:**

$$\mathbb{E}_{s,a\sim\pi_{\theta_{old}}}\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}A_{\pi_{\theta_{old}}}(s,a)$$

Есть две причины его использовать:

1. Если политика поменялась не сильно, то можно сказать, что он примерно равен оценке advantage текущей политики

1. Максимум такой функции всегда неотрицательный. Это значит, что максимизируя такую функцию, мы не ухудшим старую политику.

   Замечание: advantage не положителен только для оптимальной политики. Для неоптимальной он может быть положительным.

NPG и TRPO

В итоге получаем такую задачу оптимизации:

$$\mathbb{E}_{s,a\sim\pi_{\theta_{old}}} \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\pi_{\theta_{old}}}(s,a) \to \max$$

$$\text{s.t. } D_{KL}(\theta||\theta_{old}) \le \delta$$

Разложим в ряд Тейлора до первого ненулевого члена:

Градиент матожидания advantage

$$g^T(\theta - \theta_{old}) \to \max$$

$$\text{s.t. } 0.5 \cdot (\theta - \theta_{old})^T H (\theta - \theta_{old}) \le \delta$$

Гессиан KL-дивергенции

NPG и TRPO

В итоге имеем:

$$g^T(\theta - \theta_{old}) \to \max$$
$$\text{s.t. } 0.5 \cdot (\theta - \theta_{old})^T H(\theta - \theta_{old}) \leq \delta$$

Решаем с помощью перехода к двойственной задаче:

$$\theta = \theta_{old} + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g$$

Поскольку считать обратную матрицу больно, решаем $Hx = g$ по **x** с помощью метода сопряженных градиентов

NPG и TRPO

Natural policy gradient:

$$\theta = \theta_{old} + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g$$

Trust Region Policy Optimization:

$$\theta = \theta_{old} + \alpha^i \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g$$

Где ***i*** - это наименьшее целое неотрицательное число такое, что соблюдается условие на KLD и advantage не отрицателен.

PPO with penalty

В TRPO была такая задача оптимизации:

$$\mathbb{E}_{s,a\sim\pi_{\theta_{old}}}\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}A_{\pi_{\theta_{old}}}(s,a) \to \max$$
$$\text{s.t. } D_{KL}(\theta||\theta_{old}) \le \delta$$

Вместо того, чтобы пытаться решить честно, сделаем простую регуляризацию:

$$L_{\text{PPO-penalty}} = \mathbb{E}_{s,a\sim\pi_{\theta_{old}}}\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}A_{\pi_{\theta_{old}}}(s,a) - \beta\cdot D_{KL}(\pi_\theta(.|s)||\pi_{\theta_{old}}(.|s))$$

PPO with penalty

Функция потерь:

$$L_{\text{PPO-penalty}} = \mathbb{E}_{s,a\sim\pi_{\theta_{old}}} \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\pi_{\theta_{old}}}(s,a) - \beta \cdot D_{KL}(\pi_\theta(.|s)||\pi_{\theta_{old}}(.|s))$$

Что такое $\boldsymbol{\beta}$?

Пусть у нас задано желаемое значение KLD d_{target}, а также значение, полученное в результате обновления политики

Тогда $\boldsymbol{\beta}$ будем менять следующим образом:

$$\begin{cases} \beta_{t+1} = \beta_t \cdot a & \text{if } d < d_{target}/b \\ \beta_{t+1} = \beta_t / a & \text{if } d > d_{target} \cdot b \\ \beta_{t+1} = \beta_t \end{cases}$$

PPO with clipping

В TRPO была такая задача оптимизации:

$$\mathbb{E}_{s,a\sim\pi_{\theta_{old}}}\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}A_{\pi_{\theta_{old}}}(s,a) \to \max$$

$$\text{s.t. } D_{KL}(\theta||\theta_{old}) \leq \delta$$

Еще более простой подход. Будем клипать отношение вероятностей:

$$L_{\text{PPO-clip}} = \mathbb{E}_{s,a\sim\pi_{\theta_{old}}} \min[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}A_{\pi_{\theta_{old}}}(s,a), \text{clip}(\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}, 1-\epsilon, 1+\epsilon)A_{\pi_{\theta_{old}}}(s,a)]$$

PPO with clipping

Функция потерь:

$$L_{\text{PPO-clip}} = \mathbb{E}_{s,a\sim\pi_{\theta_{old}}} \min[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\pi_{\theta_{old}}}(s,a), \text{clip}(\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}, 1-\epsilon, 1+\epsilon) A_{\pi_{\theta_{old}}}(s,a)]$$

Давайте запишем иначе:

$$L_{\text{PPO-clip}} = \mathbb{E}_{s,a\sim\pi_{\theta_{old}}} \begin{cases} \min(\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}, 1+\epsilon) A_{\pi_{\theta_{old}}}(s,a) & A_{\pi_{\theta_{old}}}(s,a) \geq 0 \\ \max(\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}, 1-\epsilon) A_{\pi_{\theta_{old}}}(s,a) & A_{\pi_{\theta_{old}}}(s,a) < 0 \end{cases}$$

Proximal Policy Optimization

Алгоритм:

1. В течении **T** итераций:
   1. Получить **n** эпизодов в соответствии с текущей политикой
   2. Посчитать для каждого эпизода оценку advantage
   3. В течении **K** итераций:
      1. Засэмплить батч **B** из эпизодов, полученных в 1.1
      2. Обновить актора в соответствии с $L_{\text{PPO-penalty}}$ или $L_{\text{PPO-clip}}$
      3. Обновить критика с помощью MSE, L1 или Smooth L1

Как можно улучшить обучение:

1. Нормализовать advantage при обучении actor’a
2. Применять clipping градиентов
3. Прекращать цикл 1.3 если KLD между старой и новой политиками стала больше заданного значения

Proximal Policy Optimization

Замечания:

1. Политика все еще стохастическая

1. Можем легально использовать данные от старой политики благодаря Importance Sampling и ограничению изменения политики

1. Для оценки advantage можно использовать те же подходы, что и для награды в алгоритме actor-critic. Стандартным считается использование lambda-returns

1. Для улучшения исследования среды агентом чаще всего используют энтропию распределения действий в качестве регуляризации