Введение в обучение с подкреплением

Тема 13: Метод Актор-Критик

Лектор: Кривошеин А.В.

Модификации REINFORCE

Метод REINFORCE непосредственным образом максимизирует ожидаемый доход со старта эпизода путём подбора параметров у модели, которая формирует стратегию агента π_θ:

$$\max_\theta J(\pi_\theta) = \max_\theta \mathbb{E}_{\pi_\theta}[G_0].$$

Рассмотрим ряд модификаций метода REINFORCE. Простейшей полезной модификацией REINFORCE метода является введение **базы** (англ. baseline). Рассмотрим случайную величину вида $b(S_t)$, где b некоторая достаточно хорошая функция.

Полученную формулу для вычисления градиента выше тогда можно привести к виду:

$$\operatorname{grad}_\theta J(\pi_\theta) = \sum_{t=0}^{T-1} \gamma^t \mathbb{E}_{\pi_\theta}[G_t \cdot \operatorname{grad}_\theta \ln \pi_\theta(A_t \mid S_t)] = \sum_{t=0}^{T-1} \gamma^t \mathbb{E}_{\pi_\theta}[(G_t - b(S_t)) \cdot \operatorname{grad}_\theta \ln \pi_\theta(A_t \mid S_t)], \text{ поскольку}$$

$$\sum_{t=0}^{T-1} \gamma^t \mathbb{E}_{\pi_\theta}[b(S_t) \cdot \operatorname{grad}_\theta \ln \pi_\theta(A_t \mid S_t)] = 0.$$

Модификации REINFORCE

На практике для конкретной траектории $\mathrm{grad}_\theta J(\pi_\theta)$ приближается следующей величиной:

$$\mathrm{grad}_\theta J(\pi_\theta) \approx \sum_{t=0}^{T-1} \gamma^t (G_t - b(s_t)) \cdot \mathrm{grad}_\theta \ln \pi_\theta(a_t \mid s_t).$$

Взяв в качестве b функцию тождественно равную нулю, получим исходный метод REINFORCE.

Другой способ выбора базы — это среднее от дохода по траектории

$$b = \frac{1}{T} \sum_{t=0}^{T} \gamma^t G_t = \frac{1}{T} G_0.$$

Эффект использования базы в данном случае заключается в центрировании доходов по траектории относительно нуля. То есть для каждой траектории действия, дающие доходы выше среднего, поощряются, а в ином случае — штрафуются. База также позволяет снизить дисперсию градиентов и ускорить обучение.

Имеет смысл выбирать базу зависящей от состояний. Для некоторых состояний ценности всех действий могут быть высокими и нужна высокая база для отделения действия с большей или меньшей ценностью. Там же, где ценности действий малы, то нужна низкая база.

Подходящей базой в этом смысле является оценка ценности состояний $v_\pi(S_t)$.

Метод Актор-Критик

Итак, для вычисления градиента целевой функции будем использовать формулу градиента в виде

$$\text{grad}_\theta J(\pi_\theta) \approx \sum_{t=0}^{T} \gamma^t (G_t - v_{\pi_\theta}(s_t)) \cdot \text{grad}_\theta \ln \pi_\theta(a_t \mid s_t).$$

Разность $G_t - v_{\pi_\theta}(s_t)$ называют **преимуществом**. Эта величина указывает, насколько фактически выбранное действие является “хорошим” или “плохим” путём сравнения фактически полученного дохода при выборе действия a_t с усреднённым доходом $v_{\pi_\theta}(s_t)$, который может быть получен из текущего состояния при действии по стратегии π_θ.

Полученная формула является основой для алгоритма **Актор-Критик** (англ. Action-Critic). Актором является модель, формирующая стратегии π_θ. А критиком является модель $v_\omega(s_t)$, формирующая оценки V-функции для стратегии π_θ .

Эти две модели будут обучаться одновременно. Агент формирует траекторию взаимодействия со средой по текущей стратегии π_θ. После этого формируем минимизируемую функцию ошибки следующим образом:

$$J(\pi_\theta) = -\sum_{t=0}^{T} \gamma^t (G_t - v_{\pi_\theta}(s_t)) \cdot \ln \pi_\theta(a_t \mid s_t) + \sum_{t=0}^{T} (G_t - v_\omega(s_t))^2.$$

Минимизация этой функции одновременно ведёт и к улучшению стратегии, и к улучшению оценок ценностей состояний.

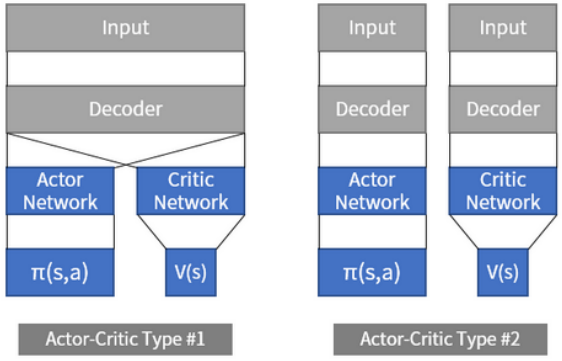
Алгоритм Актор-Критик объединяет подходы, основанные на стратегии и на функциях ценности. Актор настраивает параметризованную стратегию, а Критик настраивает подкрепляющие сигналы (то есть разности $G_t - v_\omega(s_t)$), передающиеся Актору для оценки его действий.

Метод Актор-Критик

Настроенный подкрепляющий сигнал может быть более информативным для обновления стратегии, чем просто доход G_t. Кроме того, такой сигнал имеет меньшую дисперсию, чем набор доходов G_t.

Однако, сам процесс обучения становится сложнее, и пока не будет настроен качественный подкрепляющий сигнал, агенту будет сложно понять, какие действия действительно хорошие.

На практике для формирования значений вероятности $\pi_\theta(\cdot \mid s)$ выбора действий в состоянии s и оценки ценности может использоваться одна и та же модель, то есть одна и та же ИНС. Но на выходе этой сети формируется вектор вероятностей $(\pi_\theta(a_1 \mid s),\ \ldots,\ \pi_\theta(a_N \mid s))$ и значение $v_\theta(s)$.



Второй подход заключается в разделении модели Актора и модели Критика на две различные ИНС.

Метод Актор-Критик

Преимущества общей сети:

1. Настройка стратегии $\pi_\theta(a \mid s)$ и ценностей $v_\omega(s)$ связаны. Общая сеть извлекает из состояния единый набор признаков, которые можно использовать как для обучения стратегии, так и для обучения V-функции.

2. Общая сеть означает меньший набор настраиваемых параметров.

3. Общая сеть повышает эффективность обучения. При использовании двух ИНС Актор будет выдавать плохие стратегии, пока не будет достаточно хорошо обучена V-функция.

Недостаток общей сети:

1. Обучение менее устойчиво, так как общая ИНС обучается по градиентам от двух компонентов функции ошибки. Нормы этих градиентов по разным компонентам могут различаться, что может приводить к неустойчивому обучению.

С этой проблемой можно справится путём добавления веса к одному из слагаемых при формировании функции ошибки. Однако, это ещё один гиперпараметр, который нужно настраивать вручную во время обучения.

Среда с непрерывными действиями

Рассмотрим способ обучения агента в среде с непрерывно меняющимися действиями. Базовая идея в том, чтобы генерировать не значения действий, а распределение вероятностей для выбора значения действия с помощью, например, нормального распределения, параметры которого зависят от текущего состояния и получаются от некоторой модели.

То есть под обозначением $\pi_\theta(a \mid s)$ будем понимать плотность нормального распределения вероятности вида

$$\pi_\theta(a \mid s) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(a-\mu)^2}{2\sigma^2}}, \quad \text{где}$$

$$\mu = \mu(s), \quad \sigma = \sigma(s).$$

Для простоты можно считать, что σ константа, а среднее $\mu = \mu_\theta(s)$ получается как результат работы модели, например, ИНС с параметрами θ. Градиент по параметрам тогда имеет вид:

$$\operatorname{grad}_\theta \ln \pi_\theta(a \mid s) = \frac{a - \mu_\theta(s)}{\sigma^2} \operatorname{grad}_\theta \mu_\theta(s).$$

Регуляризация энтропии

При использовании методов на основе стратегии функция ошибки составлена так, чтобы сдвигать распределение в сторону более вероятного выбора наиболее успешных действий.

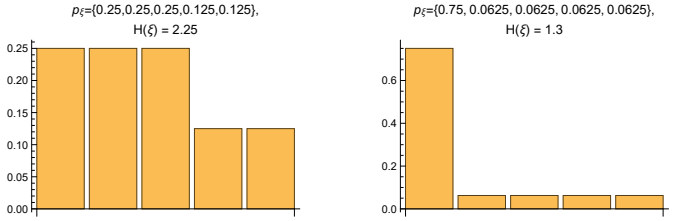
Для сложных сред это может помешать поиску оптимальных действий, поскольку агент может не успеть провести достаточно полное исследование среды. Цель регуляризации энтропии в том, чтобы поощрять агента заниматься исследованием среды.

Понятие **энтропии распределения случайной величины** является количественной мерой неопределённости этого распределения. Для дискретной случайной величины ξ с вероятностным распределением вида $\mathbb{P}(\xi_i = x_i) = p_i$, $i = 1, \ldots, n$ энтропия имеет вид

$$H(\xi) = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i} = -\sum_{i=1}^{n} p_i \log_2 p_i.$$

Чем более равномерным является распределение, тем больше энтропия (больше “неопределённость”).
Чем более вероятно принимается одно из значений случайной величины — тем меньше энтропия (меньше “неопределённость”).

Out[ ]=



Регуляризация энтропии

Модель Актора возвращает вероятностное распределение случайной величины $\pi_\theta(\cdot \mid s)$. Если добавить в функцию ошибки метода Актор-Критик энтропию текущего распределения $H(\pi_\theta(\cdot \mid s))$, то агент будет стремится к более “неопределённым” стратегиям (с более равномерным выбором действий). Функция ошибки будет иметь вид:

$$J(\pi_\theta) = -\sum_{t=0}^{T} \gamma^t (G_t - v_{\pi_\theta}(s_t)) \cdot \ln \pi_\theta(a_t \mid s_t) \; - \beta \, H(\pi_\theta(\cdot \mid s)) + \sum_{t=0}^{T} (G_t - v_{\pi_\theta}(s_t))^2.$$

Параметр $\beta > 0$ контролирует силу влияния значения энтропии на итоговую функцию ошибки. С течением обучения этот параметр можно снижать к нулю.

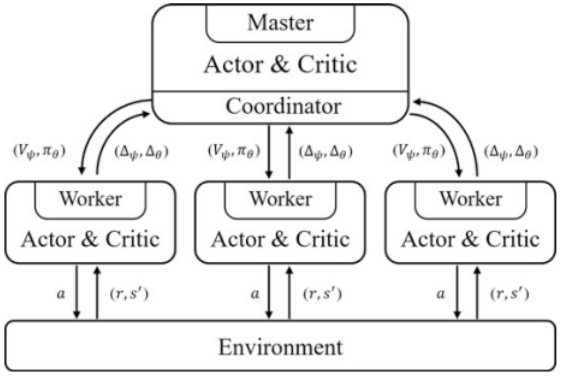
Отметим, что энтропия распределения неотрицательна, а значит величина $-\beta \, H(\pi_\theta(\cdot \mid s))$ отрицательна. Чем более распределение вероятностей $\pi_\theta(\cdot \mid s)$ схоже с равномерным, тем больше модуль этой величины.

ИНС

Метод легко распараллелить. Выделим глобального Актора-Критика. Эта ИНС, которая хранит и обновляет веса. Эту сеть ещё называют **Мастером** (Master).

И выделим ряд **Рабочих** (Workers). Каждый из Рабочих получает копию ИНС Мастера и формирует траекторию взаимодействия со средой. По сформированным траекториям каждый из Рабочих формирует функцию ошибки и находит градиент этой функции по параметрам ИНС.

Эти градиенты отправляются Мастеру через Координатора, который собирает эти градиенты (дожидаясь ответа от каждого Рабочего), Мастер сдвигает свои параметры по полученным градиентам и отправляет новые веса каждому рабочему.



Метод Synchronous A2C

1. Мастер

ИНС π_ω, V_θ, α, набор Рабочих W

Повторять:

($grad_\omega$, $grad_\theta$) = (0,0)

For worker in W:

($grad_\omega$, $grad_\theta$) =($grad_\omega$, $grad_\theta$) + worker(V_θ, π_θ)

$\omega = \omega - \alpha\ grad_\omega$, $\theta = \theta - \alpha\ grad_\theta$.

2. Рабочий

Принимает на вход π_ω, V_θ

Сформировать траекторию τ

Оценить преимущества: $A_t = G_t - V_\theta(S_t)$

Сформировать функцию ошибки:

$$J(\omega,\theta) = -\sum_{t=0}^{T} A_t \cdot \ln \pi_\theta(a_t \mid s_t) + \sum_{t=0}^{T} A_t^2$$

Найти её градиенты по параметрам и вернуть ($grad_\omega$, $grad_\theta$).

A3C

Существует модификация A3C — Asynchronous Advantage Actor-Critic.

В A3C не нужен координатор. Каждый Рабочий напрямую взаимодействует с Мастером.

Мастер не ждёт ответа от всех Рабочих для своего обновления. Как только один из Рабочих сформировал градиенты, Мастер обновляет параметры.

Благодаря этому выше вычислительная эффективность по сравнению с A2C. Однако, так как каждый Рабочий взаимодействует с Мастером независимо друг от друга, то может оказаться так, что применяемый сдвиг параметров производится по градиентам, найденным по устаревшим параметрам.

Тем не менее, на практике такой подход ускоряет обучение.

PPO

Рассмотрим ещё одну идею, развивающую алгоритм Актор-Критик. Одна из проблем методов Актор-Критик связана с тем, что в ходе обучения часто наблюдается резкое падение эффективности действий агента, то есть агент начинает действовать “плохо”. Причём это сложно устранить в дальнейшем обучении, так как агент начинает порождать плохие траектории, которые затем используются в дальнейшей настройке стратегии.

Предложенный в 2017 году алгоритм **оптимизации ближайшей стратегии** (англ. PPO, Proximal Policy Optimization) позволяет решить эту проблему. Суть в том, что мы подменим целевую функцию, которая позволит избежать падения доходов, обеспечив гарантированный монотонный рост этих доходов.

Резкое падение доходов может происходить из-за следующих причин. В ходе оптимизации целевой функции в виде ожидаемого дохода происходит поиск по пространству стратегий. Однако, это делается не напрямую, а через модель, то есть на самом деле мы перебираем параметры модели и тем самым перебираем стратегии.

Проблема в том, что пространства стратегий и пространство параметров могут быть весьма различны, в частности, расстояния между наборами параметров и расстояния между стратегиями могут вести себя различным сложным образом. Например, близкие наборы параметров могут приводить к стратегиям с большой разницей в ожидаемом доходе.

PPO

В силу описанных выше причин сложно подобрать оптимальный размера шага обучения α, так как малый шаг ведёт к медленному обучению и возможности застрять в локальном минимуме, а большой шаг может привести к тому, что в пространстве стратегий мы выйдем из зоны “хороших” стратегий.

Основная идея в том, чтобы подменить максимизацию ожидаемого дохода на максимизацию функции $J^{\text{clip}}(\theta) \approx J(\pi_\theta) - J(\pi_{\theta_{\text{old}}})$.

Максимизируя $J^{\text{clip}}(\theta)$, мы пытаемся получить лучшую стратегию π_θ недалеко от текущей стратегии.

Функция $J^{\text{clip}}(\theta)$ определяется в виде

$$J^{\text{clip}}(\theta) = \sum_{t=0}^{T-1} \gamma^t \mathbb{E}\left[\min\begin{pmatrix} \rho_t(\theta)\,\text{Adv}_t, \\ \text{clip}\,(\rho_t(\theta),\, 1-\varepsilon,\, 1+\varepsilon)\,\text{Adv}_t \end{pmatrix}\right],$$

где $\rho_t(\theta) = \dfrac{\pi_\theta(A_t \mid S_t)}{\pi_{\theta_{\text{old}}}(A_t \mid S_t)}$, $\text{Adv}_t = G_t - v_{\pi_{\theta_{\text{old}}}}(S_t)$, Adv_t называют преимуществом.

Здесь $\pi_{\theta_{\text{old}}}$ это стратегия, по которой генерируется траектория, она фиксирована.

Цель в том, чтобы найти $\theta^{\text{new}} = \text{argmax}\, J^{\text{clip}}(\theta)$, тем самым получим новую стратегию $\pi_{\theta^{\text{new}}}$.

PPO

$$J^{\text{clip}}(\theta) = \sum_{t=0}^{T-1} \gamma^t \mathbb{E}\Big[\min\Big(\begin{matrix}\rho_t(\theta)\,\text{Adv}_t, \\ \text{clip}\,(\rho_t(\theta),\, 1-\varepsilon,\, 1+\varepsilon)\,\text{Adv}_t\end{matrix}\Big)\Big], \qquad \text{где } \rho_t(\theta) = \frac{\pi_\theta(A_t \mid S_t)}{\pi_{\theta_{\text{old}}}(A_t \mid S_t)},$$

Величина $\rho_t(\theta)$ при $\theta = \theta_{\text{old}}$ будет равна 1. Для других параметров θ величина $\rho_t(\theta)$ будет больше или меньше 1. Цель в том, чтобы эта величина не удалялась бы слишком от 1, то есть отношение вероятностей между новой и старой вероятностью лежало бы в интервале $(1-\varepsilon,\ 1+\varepsilon)$.

Если преимущество Adv_t положительное, то при максимизации $J^{\text{clip}}(\theta)$ мы увеличиваем вероятность этого действия. Если отрицательное — то уменьшаем его вероятность.

Если в ходе максимизации величина $\rho_t(\theta)$ становится слишком удалена от 1 в большую сторону с положительным значением Adv_t, то мы будем использовать срезанное значение.

Аналогично, если величина $\rho_t(\theta)$ становится сильно меньше 1 при отрицательном Adv_t, то тоже будет использовано срезанное значение.

Такая функция ошибки предотвращает сдвиги параметров, которые сильно изменяют распределение вероятностей выбора действий.

PPO

Общая схема алгоритма PPO.

1. Инициализировать модель Актора и Критика.
2. Для каждой эпохи обучения:
    3. Сгенерировать несколько траекторий по текущей стратегии
    4. Найти доходы с каждого момента времени и до конца эпизода для каждой траектории
    5. Вычислить преимущества для каждого шага
    6. Обновить параметры Актора на основе указанной выше целевой функции $J^{\text{clip}}(\theta)$
    7. Обновить параметры Критика, минимизируя средне-квадратичную ошибку между доходами и оценками критика.

Algorithm 1 PPO-Clip

Input: initial policy parameters θ_0, initial value function parameters ϕ_0
for $k = 0, 1, 2, \ldots$ **do**
  Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
  Compute rewards-to-go $\hat{R}_t$.
  Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.
  Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t),\ \ g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right),$$

  typically via stochastic gradient ascent with Adam.
  Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

  typically via some gradient descent algorithm.
end for