

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



RL Foundations for Agents

From SFT to policy gradients

Carnegie Mellon University
Language Technologies Institute

Daniel Fried
11-768: AI Agents

1

智能体的强化学习基础

从监督微调（SFT）到策略梯度

Daniel Fried

11-768: AI 智能体

智能体的强化学习基础

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Running example: Number Search

A hidden integer lies between 1 and 16. The agent has four guesses. After an incorrect guess, the environment replies **higher** or **lower**. Reward is 1 only when it finds the number.

Hidden number for this task instance: 11

Four sample trajectories:

τ_1	8	higher	12	lower	10	higher	11	correct	R = 1
τ_2	4	higher	8	higher	12	lower	11	correct	R = 1
τ_3	16	lower	8	higher	9	higher	10	timeout	R = 0
τ_4	1	higher	2	higher	3	higher	4	timeout	R = 0

τ_1 and τ_2 different actions · same successful outcome

2

运行示例：数字搜索

隐藏整数位于 1 到 16 之间。智能体有四次猜测机会。猜错后，环境会回答 “更高” 或 “更低”。只有找到该数字时奖励才为 1。

此任务实例的隐藏数字：11。

四条采样轨迹：

τ_1 : 8 → 更高 → 12 → 更低 → 10 → 更高 → 11 → 正确, R=1。

τ_2 : 4 → 更高 → 8 → 更高 → 12 → 更低 → 11 → 正确, R=1。

τ_3 : 16 → 更低 → 8 → 更高 → 9 → 更高 → 10 → 超时, R=0。

τ_4 : 1 → 更高 → 2 → 更高 → 3 → 更高 → 4 → 超时, R=0。

τ_1 和 τ_2 的动作不同，但成功结果相同。

运行示例：数字搜索

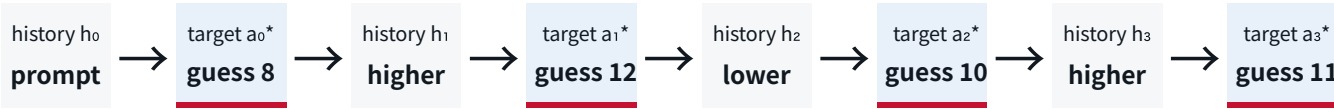
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

SFT on demonstrations

SFT trains the agent to maximize the probability of all target actions in a demonstration



$$\mathcal{L}_{\text{SFT}} = - \sum_t \log \pi_{\theta}(a_t^* \mid h_{1:t})$$

Target actions may come from a person, a stronger model, or (today) a successful trajectory from this agent

3

从示范进行监督微调

SFT 训练智能体，让它最大化示范中所有目标动作的概率。

提示词 → 猜 8 → 更高 → 猜 12 → 更低 → 猜 10 → 更高 → 猜 11。

历史 h₀ / 目标 a₀*; 历史 h₁ / 目标 a₁*; 历史 h₂ / 目标 a₂*; 历史 h₃ / 目标 a₃*。

$$\mathcal{L}_{\text{SFT}} = - \sum_t \log \pi_{\theta}(a_t^* \mid h_{1:t})$$

目标动作可以来自人类、更强的模型，或者（本讲中）来自这个智能体自身的一条成功轨迹。

从示范进行监督微调

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Task mismatch

SFT maximizes the probability of demonstrated actions. We would like to maximize the probability that the agent carries out the task. (these are not quite the same!)

DEMONSTRATION								
τ_1	8	higher	12	lower	10	higher	11	correct
ANOTHER SUCCESS								
τ_2	4	higher	8	higher	12	lower	11	correct

4

- There are multiple ways to carry out a task.
- Some actions are worse than others (e.g. should never guess "13" after "12 -> lower").

任务目标不匹配

SFT 最大化示范动作的概率。我们真正想最大化的，是智能体完成任务的概率。（二者并不完全相同！）

示范 τ_1 : 8 → 更高 → 12 → 更低 → 10 → 更高 → 11 → 正确。

另一条成功轨迹 τ_2 : 4 → 更高 → 8 → 更高 → 12 → 更低 → 11 → 正确。

同一个任务有多种完成方式。

有些动作比其他动作更差（例如，在 “12 → 更低” 之后绝不应该猜 “13” ）。

任务目标不匹配

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Data mismatch

We'd like to be able to learn from sub-optimal data, too

OTHER ALTERNATIVES	DEMONSTRATION								
	τ_1	8	higher	12	lower	10	higher	11	correct
	τ_2	4	higher	8	higher	12	lower	11	correct
	τ_3	16	lower	8	higher	9	higher	10	timeout
	τ_4	1	higher	2	higher	3	higher	4	timeout

5

数据不匹配

我们也希望能够从次优数据中学习。

示范 τ_1 : 8 $\rightarrow$ 更高 $\rightarrow$ 12 $\rightarrow$ 更低 $\rightarrow$ 10 $\rightarrow$ 更高 $\rightarrow$ 11 $\rightarrow$ 正确。

其他方案 τ_2 : 4 $\rightarrow$ 更高 $\rightarrow$ 8 $\rightarrow$ 更高 $\rightarrow$ 12 $\rightarrow$ 更低 $\rightarrow$ 11 $\rightarrow$ 正确。

τ_3 : 16 $\rightarrow$ 更低 $\rightarrow$ 8 $\rightarrow$ 更高 $\rightarrow$ 9 $\rightarrow$ 更高 $\rightarrow$ 10 $\rightarrow$ 超时。

τ_4 : 1 $\rightarrow$ 更高 $\rightarrow$ 2 $\rightarrow$ 更高 $\rightarrow$ 3 $\rightarrow$ 更高 $\rightarrow$ 4 $\rightarrow$ 超时。

数据不匹配

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Exposure bias

The agent was not exposed to its own mistakes at training time, and might not be prepared to handle mistakes when generating

TRAINING DEMONSTRATION	8	higher	12	lower	10	higher	11	correct
	8	higher	2	?				

6

暴露偏差

训练时，智能体没有经历自己犯下的错误，因此在生成时可能没有准备好应对错误。

训练示范：8 → 更高 → 12 → 更低 → 10 → 更高 → 11 → 正确。

生成过程：8 → 更高 → 2 → ？

暴露偏差

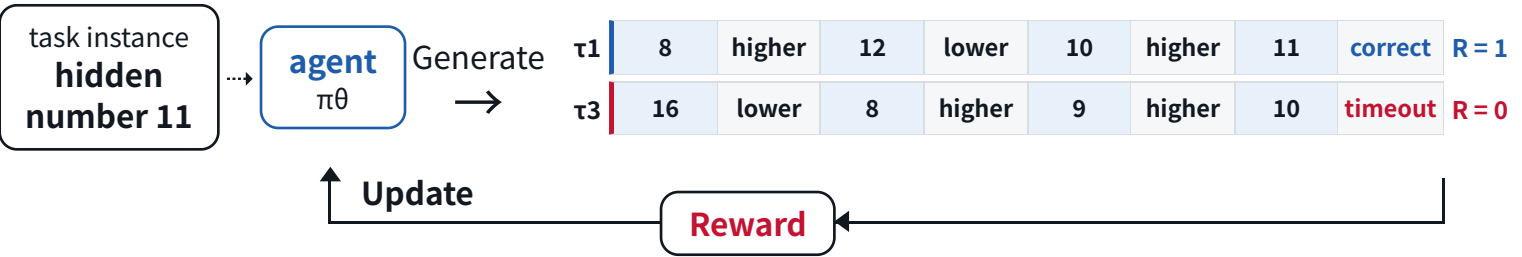
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Reinforcement learning

Generate actions/trajectories from the agent, evaluate their **reward**, and adjust the parameters of the agent to maximize expected reward.



Task mismatch	We train the agent to maximize expected reward.
Data mismatch	The agent can learn from all trajectories (seek high reward; avoid low reward)
Exposure bias	The agent generates trajectories in training (on-policy), so it learns to recover from errors

Adapted from Sean Welleck · RL Fundamentals ↗

7

强化学习

从智能体生成动作 / 轨迹，评估其奖励，并调整智能体参数以最大化期望奖励。

任务实例：隐藏数字 11 → 智能体策略 π_θ → 生成轨迹。

τ_1 : 8 → 更高 → 12 → 更低 → 10 → 更高 → 11 → 正确, R=1。

τ_3 : 16 → 更低 → 8 → 更高 → 9 → 更高 → 10 → 超时, R=0。

根据奖励更新策略。

任务目标不匹配：训练智能体最大化期望奖励。

数据不匹配：智能体可以从所有轨迹中学习（追求高奖励，避免低奖励）。

暴露偏差：训练期间由智能体生成轨迹（on-policy），因此它能学会从错误中恢复。

改编自 Sean Welleck · RL Fundamentals ↗

强化学习

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

交互构成轨迹

Interactions as trajectories

8

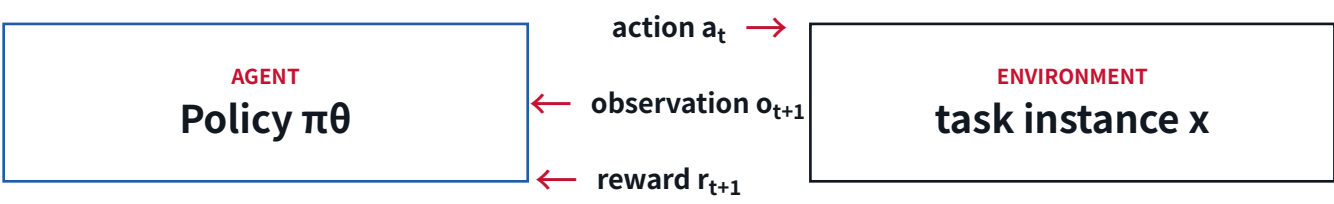
交互构成轨迹

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

The agent–environment loop



	Task instance x	Observation o_{t+1}	Action a_t	Reward r_{t+1}
Text generation	the prompt	tokens so far	next token	score from a judge model
Web agent	the task and website	rendered page	click, type, scroll	number of sub-tasks completed
Number Search	number (hidden)	higher, lower, or correct	the next guess	1 on correct; otherwise 0

9

智能体—环境循环

智能体：策略 π_θ ；接收观测 o_{t+1} 和奖励 r_{t+1} ；任务实例 x 。

任务实例 x | 观测 o_{t+1} | 动作 a_t | 奖励 r_{t+1}

文本生成 | 提示词 | 到目前为止的词元 | 下一个词元 | 评判模型给出的分数

网页智能体 | 任务与网站 | 渲染后的页面 | 点击、输入、滚动 | 已完成的子任务数量

数字搜索 | 数字（隐藏） | 更高、更低或正确 | 下一次猜测 | 猜对时为 1，否则为 0

智能体—环境循环

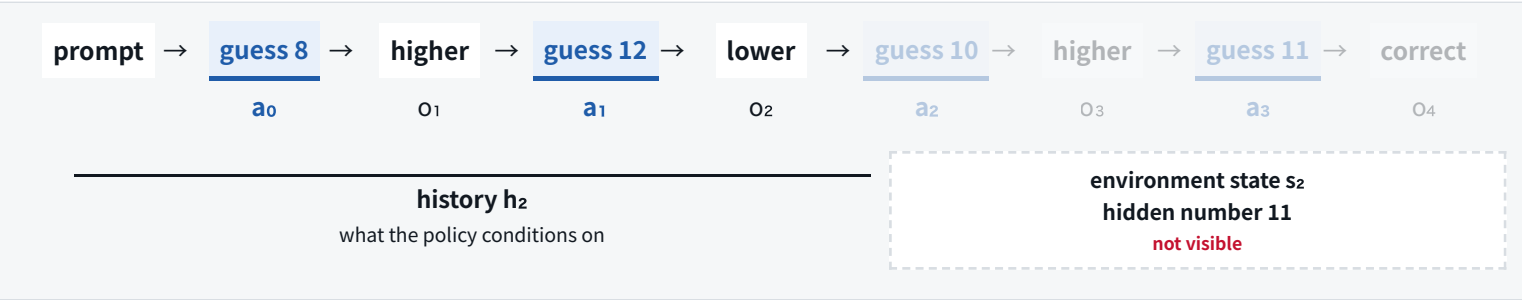
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Trajectories

The agent conditions only on past actions and observations (not hidden environment state).



$$h_t = (o_0, a_0, o_1, \dots, o_t) \quad \pi_\theta(a_t \mid h_t)$$

10

轨迹

智能体只根据过去的动作与观测作出判断（不使用环境隐藏状态）。

提示词 → 猜 8 → 更高 → 猜 12 → 更低 → 猜 10 → 更高 → 猜 11 → 正确。

a_0 、 O_1 、 a_1 、 O_2 、 a_2 、 O_3 、 a_3 、 O_4 ；环境状态 s_2 中的隐藏数字为 11。

策略所依据的历史 $h_t = (o_0, a_0, o_1, \dots, o_t)$ ； $\pi_\theta(a_t \mid h_t)$ 。

策略的条件信息；不可见。

轨迹

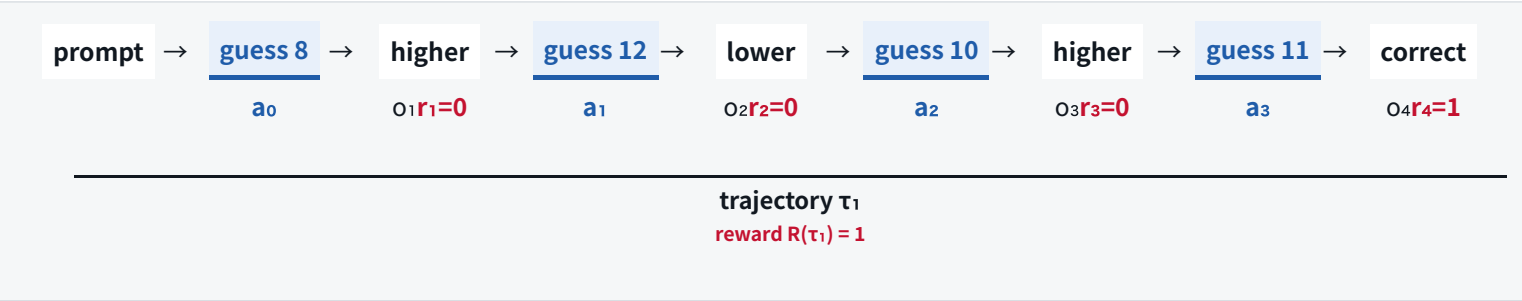
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Trajectories

Trajectories can also include reward at each step.



$$\tau = (o_0, a_0, r_1, o_1, \dots, a_{T-1}, r_T, o_T) \quad R(\tau) = \sum_{t=1}^T r_t$$

In number search, 1 for "correct" and 0 otherwise.

11

轨迹

轨迹还可以包含每一步的奖励。

提示词 → 猜 8 → 更高 → 猜 12 → 更低 → 猜 10 → 更高 → 猜 11 → 正确。

每步动作 a_0 、 a_1 、 a_2 、 a_3 对应观测 / 奖励： o_1 、 $r_1=0$ ； o_2 、 $r_2=0$ ； o_3 、 $r_3=0$ ； o_4 、 $r_4=1$ 。

轨迹 τ_1 ；奖励 $R(\tau_1)=1$ 。

$\tau = (o_0, a_0, r_1, o_1, \dots, a_{T-1}, r_T, o_T)$ ； $R(\tau)=\sum_{t=1}^T r_t$ 。

在数字搜索中，猜对时奖励为 1，其他时候为 0。

轨迹还可以包含逐步奖励

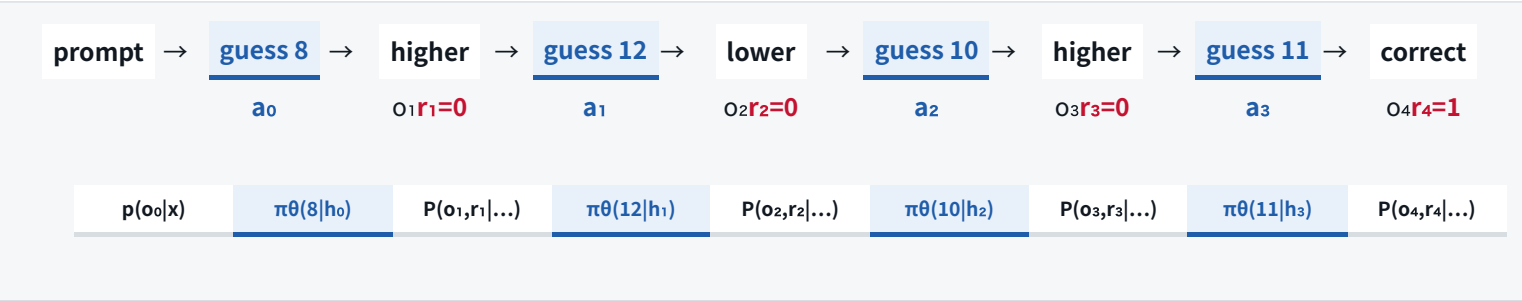
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Trajectories

Trajectory probabilities depend on the policy **and** the environment.



$$p_{\theta}(\tau \mid x) = \prod_t \pi_{\theta}(a_t \mid h_t) (o_{t+1}, r_{t+1} \mid h_t, a_t, x)$$

We don't know P, but we can sample from it by interacting with the environment.

12

轨迹概率取决于策略和环境。

轨迹：提示词 → 猜 8 → 更高 → 猜 12 → 更低 → 猜 10 → 更高 → 猜 11 → 正确。

各步概率：p(o₀|x); π_θ(8|h₀); P(o₁,r₁|...); π_θ(12|h₁); P(o₂,r₂|...); π_θ(10|h₂) ; P(o₃,r₃|...); π_θ(11|h₃); P(o₄,r₄|...)。

$$p_{\theta}(\tau \mid x) = \prod_t \pi_{\theta}(a_t \mid h_t) P(o_{t+1}, r_{t+1} \mid h_t, a_t, x)$$

我们不知道 P，但可以通过与环境交互来采样。

轨迹概率

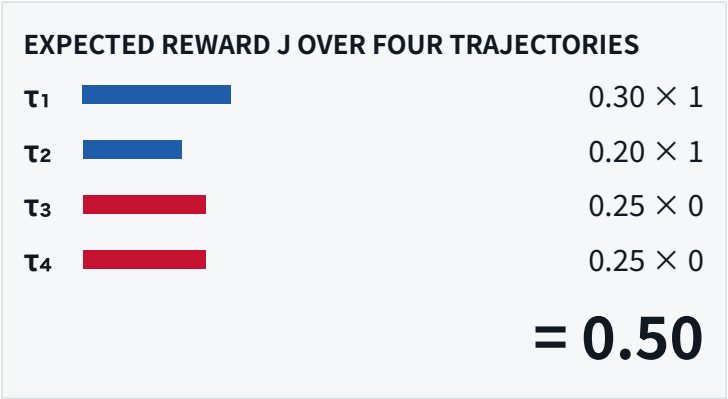
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

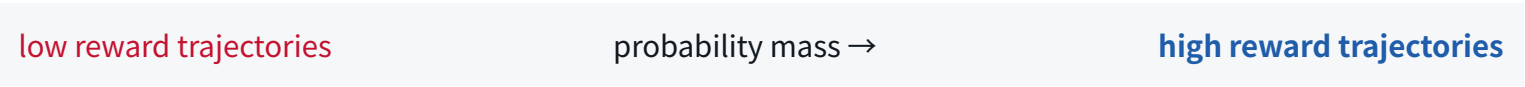
ZH / 中文译文 · 第 1/1 部分

Objective: maximize expected reward

$$J(\theta) = \mathbb{E}_{\tau \sim p_{\theta}(\cdot|x)}[R(\tau)]$$



Training adjusts θ to maximize expected reward (i.e. the agent has a higher probability of high reward trajectories)



13

目标：最大化期望奖励

$$J(\theta) = \mathbb{E}_{\tau \sim p_{\theta}(\cdot|x)}[R(\tau)]。$$

四条轨迹的期望奖励 J: τ_1 : 0.30×1 ; τ_2 : 0.20×1 ; τ_3 : 0.25×0 ; τ_4 : 0.25×0 ; 合计 = 0.50。

训练会调整 θ 以最大化期望奖励（也就是让智能体更可能生成高奖励轨迹）。

低奖励轨迹 ← 概率质量 → 高奖励轨迹。

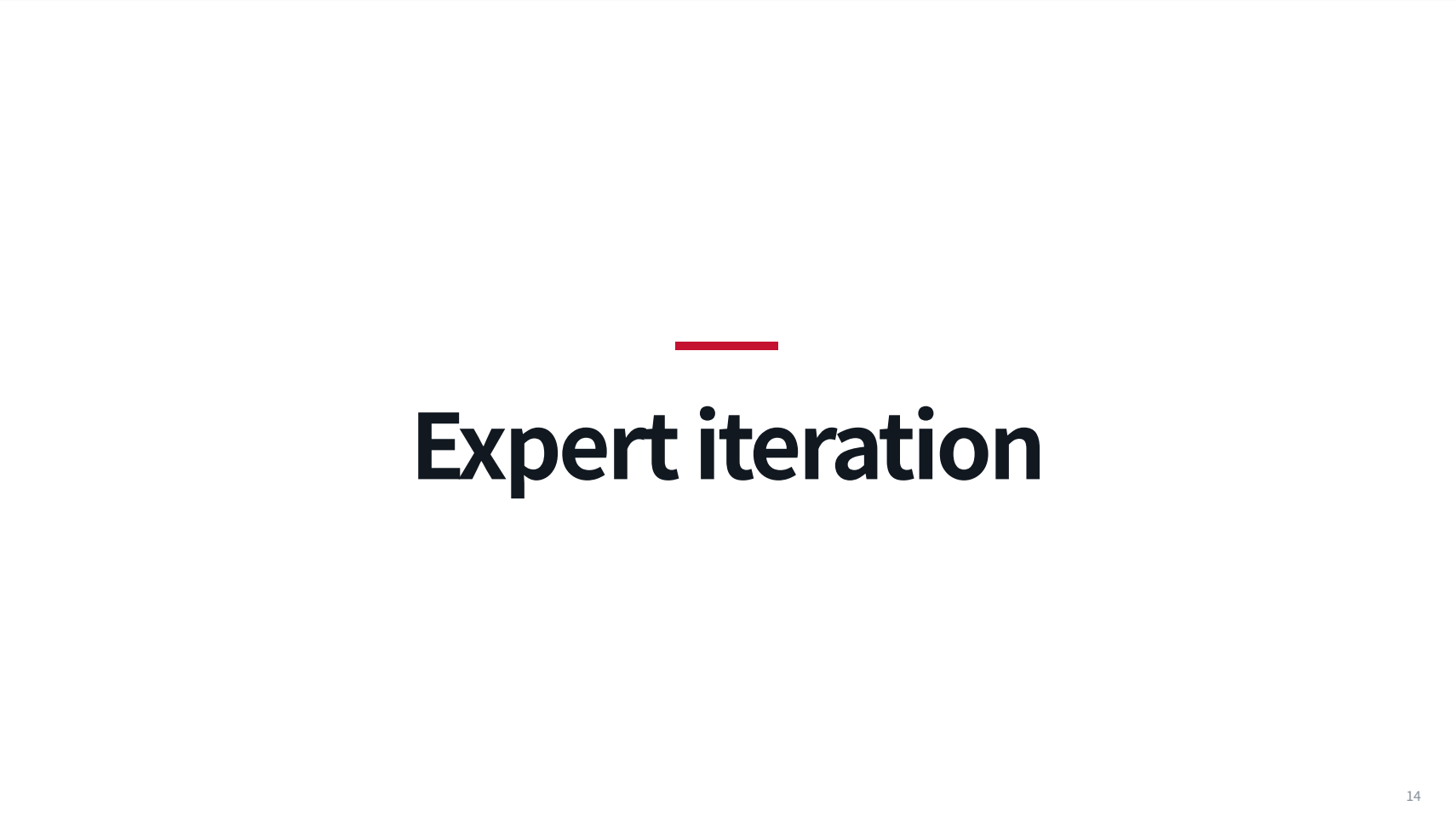
目标：最大化期望奖励

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

专家迭代



专家迭代

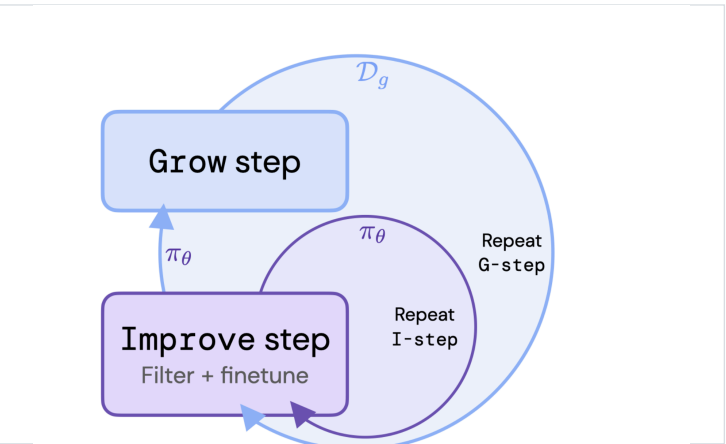
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Reinforced self-training

ReST generates data from the policy, and then updates the policy using successful trajectories.



Gulcehre et al. · ReST, Fig. 1 ↗ Singh et al. · ReST-EM ↗

15

强化式自训练

ReST 从策略中生成数据，然后使用成功轨迹更新策略。

- 1 增长：从当前策略采样轨迹并计算奖励。
- 2 改进：使用奖励过滤数据；扩充现有数据集；训练策略。
- 3 重复：使用更新后的策略。

Gulcehre 等 · ReST, 图 1 ↗ · Singh 等 · ReST-EM ↗

强化式自训练

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Reinforced self-training

The loss is the standard SFT loss, on trajectories with high reward



$$\mathcal{L}_{\text{ReST}} = - \sum_{\tau: R(\tau)=1} \sum_t \log \pi_{\theta}(a_t \mid h_t)$$

Gulcehre et al. · ReST ↗ Singh et al. · ReST-EM ↗

16

强化式自训练

在高奖励轨迹上使用标准 SFT 损失。

τ_1 : 8 → 更高 → 12 → 更低 → 10 → 更高 → 11 → 正确; R=1, 保留。

τ_2 : 4 → 更高 → 8 → 更高 → 12 → 更低 → 11 → 正确; R=1, 保留。

τ_3 : 16 → 更低 → 8 → 更高 → 9 → 更高 → 10 → 超时; R=0, 丢弃。

τ_4 : 1 → 更高 → 2 → 更高 → 3 → 更高 → 4 → 超时; R=0, 丢弃。

采样轨迹 → 计算奖励 → 保留 τ_1 、 τ_2 → SFT。

$$\mathcal{L}_{\text{ReST}} = - \sum_{\tau: R(\tau)=1} \sum_t \log \pi_{\theta}(a_t \mid h_t)$$

Gulcehre 等 · ReST ↗ · Singh 等 · ReST-EM ↗

强化式自训练：筛选高奖励轨迹

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

从奖励到梯度

策略梯度

From rewards to gradients

Policy gradients

17

从奖励到梯度

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

How to maximize expected reward?

We can't use backpropagation directly: sampling and the environment are non-differentiable



Sampled action

Changing θ changes the action probabilities, but actions are discrete and sampled.

Task environment

The world is not (usually!) differentiable. How do you take the derivative of a unit test w.r.t. code?

$$J(\theta) = \sum_{\tau} p_{\theta}(\tau \mid x) R(\tau)$$

$p_{\theta}(\tau|x)$ changes with θ

$R(\tau)$ is observed after the rollout

General solution: policy gradient (REINFORCE, Williams 1992).

18

如何最大化期望奖励？

我们不能直接使用反向传播：采样和环境都不可微。

参数 $\theta \rightarrow$ 策略 $\pi_{\theta}(a|h) \rightarrow$ 采样动作 $a \rightarrow$ 环境中的观测 o 、奖励 r 。

采样动作：改变 θ 会改变动作概率，但动作是离散采样的。

任务环境：世界（通常）不可微。怎样对单元测试关于代码求导？

$J(\theta) = \sum_{\tau} p_{\theta}(\tau \mid x) R(\tau)$ ； $p_{\theta}(\tau|x)$ 随 θ 变化； $R(\tau)$ 在轨迹运行后观测到。

通用解法：策略梯度（REINFORCE, Williams, 1992）。

如何最大化期望奖励？

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Policy gradient

Maximize expected reward under the policy.	$J(\theta) = \mathbb{E}_{\tau \sim p_{\theta}(\cdot x)} [R(\tau)] = \sum_{\tau} p_{\theta}(\tau x) R(\tau)$
Take the gradient: only the probability depends on θ .	$\nabla_{\theta} J = \sum_{\tau} \nabla_{\theta} p_{\theta}(\tau x) R(\tau)$
Multiply and divide by trajectory probability.	$= \sum_{\tau} p_{\theta}(\tau x) R(\tau) \frac{\nabla_{\theta} p_{\theta}(\tau x)}{p_{\theta}(\tau x)}$
Use $\nabla p / p = \nabla \log p$.	$= \sum_{\tau} p_{\theta}(\tau x) R(\tau) \nabla_{\theta} \log p_{\theta}(\tau x)$
This is an expectation we can estimate from samples.	$= \mathbb{E}_{\tau \sim p_{\theta}(\cdot x)} [R(\tau) \nabla_{\theta} \log p_{\theta}(\tau x)]$

REINFORCE (Williams, 1992) ↗

19

策略梯度

最大化策略下的期望奖励： $J(\theta) = \mathbb{E}_{\{\tau \sim p_{\theta}(\cdot|x)\}} [R(\tau)] = \sum_{\tau} p_{\theta}(\tau|x) R(\tau)$ 。

求梯度： $\nabla_{\theta} J = \sum_{\tau} \nabla_{\theta} p_{\theta}(\tau|x) R(\tau)$ 。

只有概率依赖 θ 。

用轨迹概率同时乘、除： $\nabla_{\theta} J = \sum_{\tau} p_{\theta}(\tau|x) R(\tau) \cdot [\nabla_{\theta} p_{\theta}(\tau|x) / p_{\theta}(\tau|x)]$ 。

使用 $\nabla p / p = \nabla \log p$ 。

$\nabla_{\theta} J = \sum_{\tau} p_{\theta}(\tau|x) R(\tau) \nabla_{\theta} \log p_{\theta}(\tau|x)$ 。

这是一个可以通过样本估计的期望： $\nabla_{\theta} J = \mathbb{E}_{\{\tau \sim p_{\theta}(\cdot|x)\}} [R(\tau) \nabla_{\theta} \log p_{\theta}(\tau|x)]$ 。

REINFORCE (Williams, 1992) ↗

策略梯度推导

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Policy gradient

$$\nabla_{\theta} J = \mathbb{E}_{\tau \sim p_{\theta}(\cdot | x)} \left[R(\tau) \nabla_{\theta} \log p_{\theta}(\tau | x) \right]$$

Replace the expectation by trajectories we sample.

$$\tau \sim p_{\theta}(\cdot | x), \quad \widehat{\nabla_{\theta} J} = R(\tau) \nabla_{\theta} \log p_{\theta}(\tau | x) = R(\tau) \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t | h_t)$$

$\tau \sim p_{\theta}(\cdot x)$ Current-policy trajectories and histories obtain by sampling: policy and environment	$R(\tau)$ Task-level evaluation obtain from the environment	$\nabla_{\theta} \log \pi_{\theta}(a_t h_t)$ Changes probabilities of sampled actions same mechanics as in SFT
--	---	--

Williams · REINFORCE, 1992 ↗

20

策略梯度

$\nabla_{\theta} J = E_{\{\tau \sim p_{\theta}(\cdot | x)\}} [R(\tau) \nabla_{\theta} \log p_{\theta}(\tau | x)]。$

用采样轨迹替代期望。

$\tau \sim p_{\theta}(\cdot | x), \nabla_{\theta} J = R(\tau) \nabla_{\theta} \log p_{\theta}(\tau | x) = R(\tau) \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t | h_t)。$

$\tau \sim p_{\theta}(\cdot | x)$ ：当前策略轨迹与历史，通过策略和环境采样获得。

$R(\tau)$ ：任务级评估，由环境提供。

$\nabla_{\theta} \log \pi_{\theta}(a_t | h_t)$ ：改变采样动作的概率；机制与 SFT 相同。

Williams · REINFORCE, 1992 ↗

策略梯度：用采样轨迹估计

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Trajectory log probability

$p(o_0|x) \cdot \pi_\theta(a_0|h_0) \cdot P(o_1,r_1|h_0,a_0,x) \cdot \pi_\theta(a_1|h_1) \cdot P(o_2,r_2|h_1,a_1,x)$

$$\begin{aligned} \log p_\theta(\tau \mid x) &= \sum_t \log P(o_t, r_t \mid h_t, a_t, x) + \sum_t \log \pi_\theta(a_t \mid h_t) \\ &= \text{const} + \sum_t \log \pi_\theta(a_t \mid h_t) \\ \nabla_\theta \log p_\theta(\tau \mid x) &= \sum_t \nabla_\theta \log \pi_\theta(a_t \mid h_t) \end{aligned}$$

Same computation as SFT: sum token log probabilities. Here the tokens were sampled by the policy.

$a_t = (u_{t,1}, \dots, u_{t,m_t}), \quad \nabla_\theta \log \pi_\theta(a_t \mid h_t) = \sum_{k=1}^{m_t} \nabla_\theta \log \pi_\theta(u_{t,k} \mid h_t, u_{t,<k})$

t indexes actions; *k* indexes tokens within action *a_t*.

We can just multiply by R(τ) to get ∇θJ.

21

$p(o_0|x) \cdot \pi_\theta(a_0|h_0) \cdot P(o_1,r_1|h_0,a_0,x) \cdot \pi_\theta(a_1|h_1) \cdot P(o_2,r_2|h_1,a_1,x)$

$\log p_\theta(\tau|x) = \sum_t \log P(o_t,r_t|h_t,a_t,x) + \sum_t \log \pi_\theta(a_t|h_t)$

$= \text{const} + \sum_t \log \pi_\theta(a_t|h_t)$

$\nabla_\theta \log p_\theta(\tau|x) = \sum_t \nabla_\theta \log \pi_\theta(a_t|h_t)$

与 SFT 中的计算相同：对词元对数概率求和。在这里，词元由策略采样得到。

$a_t = (u_{\{t,1\}}, ..., u_{\{t,m_t\}}); \nabla_\theta \log \pi_\theta(a_t|h_t) = \sum_{\{k=1\}^{\{m_t\}}} \nabla_\theta \log \pi_\theta(u_{\{t,k\}}|h_t,u_{\{t,<k\}})$ 。

t 表示动作； k 表示动作 a_t 内的词元。

只需乘以 R(τ)，即可得到 ∇_θJ。

轨迹的对数概率

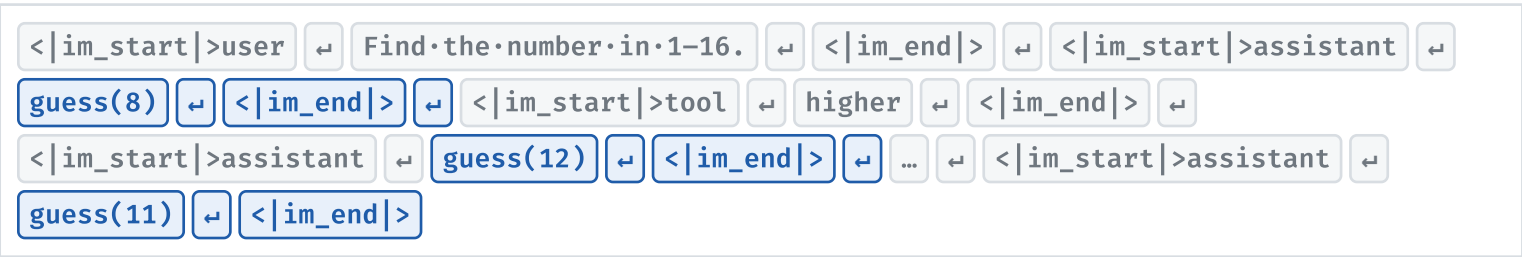
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Computing the policy gradient

Effectively, we just do masking and rescaling of the standard SFT loss computation.



“.” = space, “↵” = newline · colored = sampled by the policy, gray = context only

$$\mathcal{L}_{PG} = -R(\tau) \sum_{t:m_t=1} \log \pi_{\theta}(u_t \mid u_{<t})$$

m_t = 1 tokens the policy sampled, including the end-of-turn token	m_t = 0 prompt and observation tokens, which the policy did not choose	R(τ) one scalar, shared by every generated token in the trajectory
--	--	--

22

计算策略梯度

实际上，我们只需对标准 SFT 损失计算应用掩码并重新缩放。

<|im_start|>user ↵ Find the number in 1-16. ↵ <|im_end|> ↵
<|im_start|>assistant ↵ guess(8) ↵ <|im_end|> ↵ <|im_start|>tool ↵ higher ↵ <|im_end|> ↵
↵ <|im_end|> ↵ <|im_start|>assistant ↵ guess(12) ↵ <|im_end|> ↵ ... ↵
<|im_start|>assistant ↵ guess(11) ↵ <|im_end|>

“.” 表示空格，“↵” 表示换行 · 彩色表示由策略采样，灰色表示仅作上下文。

L_PG = -R(τ) ∑_{t:m_t=1} log π_θ(u_t | u_{<t})。

m_t=1：策略采样的词元，包括轮次结束词元。

m_t=0：提示词与观测词元，策略并未选择它们。

R(τ)：一个标量，由轨迹中的每个生成词元共享。

计算策略梯度

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Policy-gradient training loop

After an update, the new policy collects the next batch.



$$\mathcal{L}_{PG} = - \sum_i R(\tau_i) \sum_t \log \pi_{\theta}(a_{i,t} \mid h_{i,t})$$

23

策略梯度训练循环

每次更新后，新策略会收集下一批数据。

- 1 收集：采样任务实例；让智能体在这些任务上采样。
- 2 评估：计算每条轨迹的奖励。
- 3 形成损失：用奖励加权动作的对数概率。
- 4 更新：反向传播、改变 θ ，然后再次收集数据。

$\mathcal{L}_{PG} = -\sum_i R(\tau_i) \sum_t \log \pi_{\theta}(a_{i,t} | h_{i,t})。$



策略梯度训练循环

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

With binary rewards, this looks a lot like ReST

REINFORCE

$$-\sum_i R(\tau_i) \sum_t \log \pi_{\theta}(a_{i,t} \mid h_{i,t})$$

=

SFT loss on the successful trajectories

$$-\sum_{i:R(\tau_i)=1} \sum_t \log \pi_{\theta}(a_{i,t} \mid h_{i,t})$$

(there are some differences, e.g. ReST does batched updates and aggregates datasets)

[Williams · REINFORCE ↗](#) [Singh et al. · ReST-EM ↗](#)

24

二元奖励下，这与 ReST 很相似

REINFORCE: $-\sum_i R(\tau_i) \sum_t \log \pi_{\theta}(a_{i,t} | h_{i,t})$ 。

= 对成功轨迹使用的 SFT 损失: $-\sum_{i:R(\tau_i)=1} \sum_t \log \pi_{\theta}(a_{i,t} | h_{i,t})$ 。

(两者仍有一些差异，例如 ReST 使用批量更新并汇总数据集。)

Williams · REINFORCE ↗ · Singh 等 · ReST-EM ↗

二元奖励下，这与 ReST 很相似

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



超越期望

基线与优势

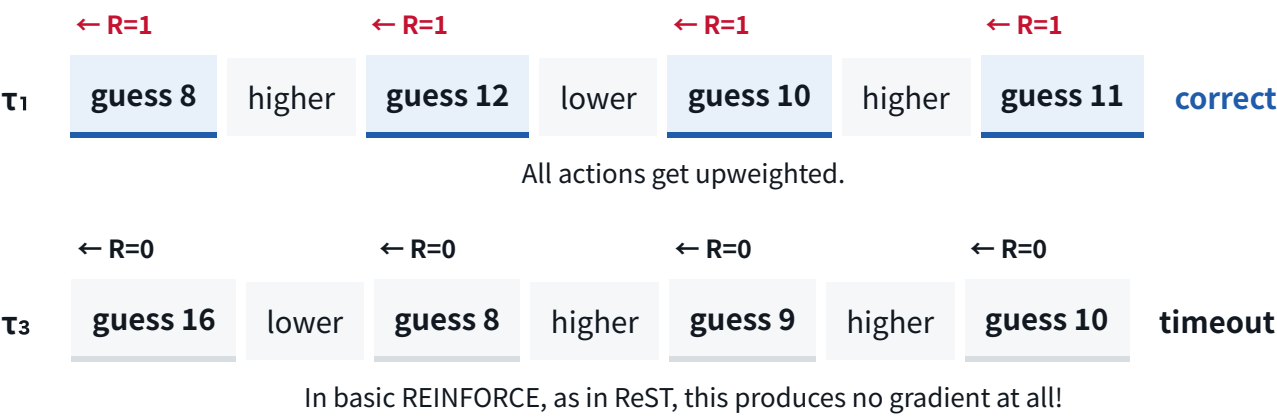
超越期望

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Rewards should be relative



The update should depend on how well the policy usually does on this example.

26

奖励应当是相对的

成功轨迹 τ_1 : 猜 8 $\rightarrow$ 更高 $\rightarrow$ 猜 12 $\rightarrow$ 更低 $\rightarrow$ 猜 10 $\rightarrow$ 更高 $\rightarrow$ 猜 11 $\rightarrow$ 正确; $R=1$ 。

所有动作的权重都会提高。

失败轨迹 τ_3 : 猜 16 $\rightarrow$ 更低 $\rightarrow$ 猜 8 $\rightarrow$ 更高 $\rightarrow$ 猜 9 $\rightarrow$ 更高 $\rightarrow$ 猜 10 $\rightarrow$ 超时; $R=0$ 。

在基本 REINFORCE 中, 与 ReST 一样, 这样做完全不会产生梯度!

更新应取决于策略在这个例子上的通常表现。

奖励应当是相对的

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

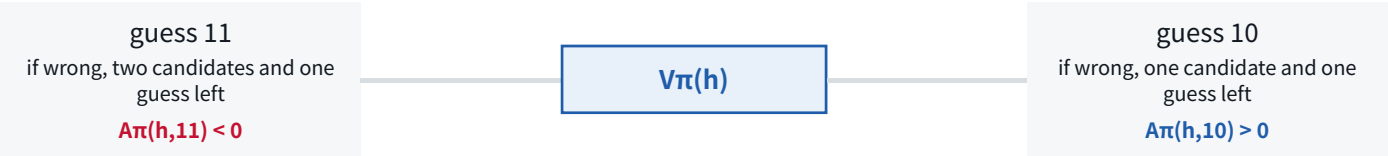
ZH / 中文译文 · 第 1/1 部分

Advantages

Advantage: how much better is this action, than the policy's average?

$$A^\pi(h, a) = \underbrace{Q^\pi(h, a)}_{\text{expected reward after } a} - \underbrace{V^\pi(h)}_{\text{expected reward from } h}$$

History h: guess 8 / higher · guess 12 / lower — the number is 9, 10 or 11, with two guesses left.



Neither Q^π nor V^π is known, so we estimate advantage from the sampled reward.

$$\hat{A}_t = \underbrace{R(\tau)}_{\text{trajectory reward}} - \underbrace{b(h_t)}_{\text{an estimate of } V^\pi(h_t)}$$

27

优势：这个动作比策略平均水平好多少？

$A^\pi(h,a) = Q^\pi(h,a) - V^\pi(h)$ 。

$Q^\pi(h,a)$ ：采取动作 a 后的期望奖励； $V^\pi(h)$ ：从历史 h 出发的期望奖励。

历史 h：猜 8 / 更高 · 猜 12 / 更低——还剩两次猜测，数字是 9、10 或 11。

猜 11：猜错时还剩两个候选数和一次猜测； $A^\pi(h,11) < 0$ 。

猜 10：猜错时还剩一个候选数和一次猜测； $A^\pi(h,10) > 0$ 。

Q^π 与 V^π 都是未知的，因此我们用采样奖励来估计优势。

$\hat{A}^t = R(\tau) - b(h_t)$ ； $R(\tau)$ 为轨迹奖励； $b(h_t)$ 是 $V^\pi(h_t)$ 的估计值。

优势

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Policy gradients with advantage estimates

Incorporate advantages by recentering the reward.

REINFORCE: one trajectory reward weights every action.

$$\widehat{\nabla_{\theta} J} = R(\tau) \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t \mid h_t)$$

Subtract a baseline — the weight is now $\hat{A}_t$.

$$\widehat{\nabla_{\theta} J} = \sum_t \underbrace{(R(\tau) - b(h_t))}_{\hat{A}_t} \nabla_{\theta} \log \pi_{\theta}(a_t \mid h_t)$$

Both lines are different ways to estimate the same $\nabla_{\theta} J$.

$\hat{A}_t = R(\tau) - b(h_t)$. Different choices for b produce different algorithms

a constant upcoming bandit example	a running mean of rewards adapts as the policy improves	$V\phi(h_t)$ a trained value model: actor-critic, PPO	the group mean no extra model: GRPO (next section)
--	---	--	--

28

用优势估计进行策略梯度

通过将奖励重新居中来纳入优势。

REINFORCE: $\nabla_{\theta} J = R(\tau) \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t|h_t)$ 。奖励为每个动作赋予权重。

减去基线: $\nabla_{\theta} J = \sum_t (R(\tau) - b(h_t)) \nabla_{\theta} \log \pi_{\theta}(a_t|h_t)$ 。现在的权重是 $\hat{A}^t$ 。

两行都是估计同一个 $\nabla_{\theta} J$ 的不同方式。

$\hat{A}^t = R(\tau) - b(h_t)$ 。b 的不同选择会产生不同算法。

常数：用于接下来的多臂赌博机示例。

奖励的滑动平均：随着策略提升而适应。

$V_{\phi}(h_t)$ ：训练得到的价值模型，即 actor-critic、PPO。

组均值：无需额外模型，即下一节的 GRPO。

用优势估计进行策略梯度

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Baselines do not bias the gradient estimator

$$\widehat{\nabla_{\theta} J} = \sum_t (R(\tau) - b(h_t)) \nabla_{\theta} \log \pi_{\theta}(a_t | h_t)$$

$$\mathbb{E}[\widehat{\nabla_{\theta} J}] = \nabla_{\theta} J - \mathbb{E}\left[\sum_t b(h_t) \nabla_{\theta} \log \pi_{\theta}\right]$$

so this is an unbiased estimate of J's gradient if the expectation is 0

$$\mathbb{E}_{a \sim \pi_{\theta}(\cdot|h)}[b(h) \nabla_{\theta} \log \pi_{\theta}(a | h)]$$
$$= b(h) \sum_a \pi_{\theta}(a | h) \nabla_{\theta} \log \pi_{\theta}(a | h)$$
$$= b(h) \sum_a \nabla_{\theta} \pi_{\theta}(a | h) \quad \textcolor{red}{\pi \nabla \log \pi = \nabla \pi}$$
$$= b(h) \nabla_{\theta} \sum_a \pi_{\theta}(a | h) = b(h) \nabla_{\theta} 1 = \mathbf{0}$$

So $E[\nabla_{\theta} J] = \nabla_{\theta} J$ for any baseline that conditions only on the history

29

基线不会使梯度估计产生偏差

$\nabla_{\theta} J = \sum_t (R(\tau) - b(h_t)) \nabla_{\theta} \log \pi_{\theta}(a_t | h_t)$ 。

$E[\nabla_{\theta} J] = \nabla_{\theta} J - E[\sum_t b(h_t) \nabla_{\theta} \log \pi_{\theta}]$ ；因此，若后一项的期望为 0，该估计就是 J 的无偏梯度估计。

$E_{\{a \sim \pi_{\theta}(\cdot|h)\}}[b(h) \nabla_{\theta} \log \pi_{\theta}(a|h)] = b(h) \sum_a \pi_{\theta}(a|h) \nabla_{\theta} \log \pi_{\theta}(a|h)$ 。
 $= b(h) \sum_a \nabla_{\theta} \pi_{\theta}(a|h)$ 。

$\pi \nabla \log \pi = \nabla \pi$ 。
 $= b(h) \nabla_{\theta} \sum_a \pi_{\theta}(a|h) = b(h) \nabla_{\theta} 1 = 0$ 。

因此，对于任何只依赖历史的基线， $E[\nabla_{\theta} J] = \nabla_{\theta} J$ 。

基线不会使梯度估计产生偏差

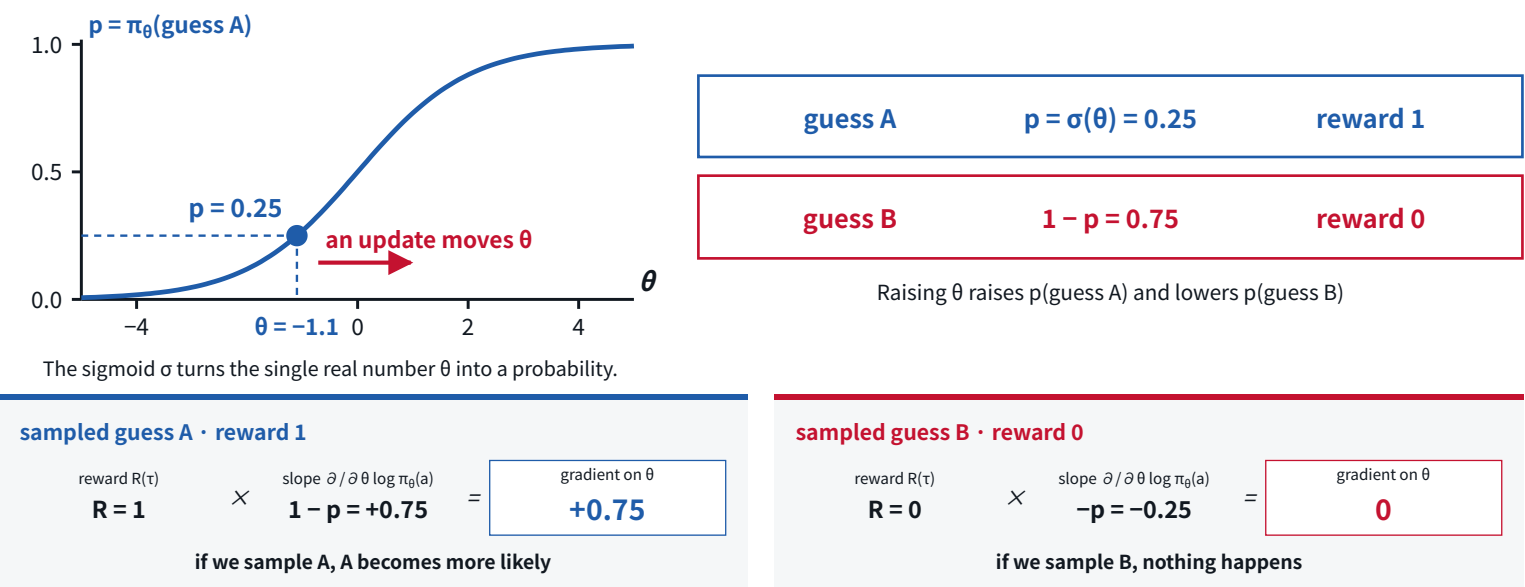
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Simple example: two-armed bandit

With one policy parameter and two actions, we can compute gradients exactly.



This gradient is correct, but a baseline/advantage can make it more useful.

30

简单示例：双臂赌博机

只有一个策略参数和两个动作时，可以精确计算梯度。

$p = \pi_\theta(\text{猜 A})$ ；图示 sigmoid σ 将实数 θ 转换为概率； $p = \sigma(\theta) = 0.25$ 。提高 θ 会提高 $p(\text{猜 A})$ ，降低 $p(\text{猜 B})$ 。图中坐标：纵轴 p 刻度 0.0、0.5、1.0；横轴 θ 刻度 -4、0、2、4；当前点 $\theta = -1.1$ 、 $p = 0.25$ 。

猜 A：概率 $p = 0.25$ ，奖励 1；猜 B：概率 $1 - p = 0.75$ ，奖励 0。

图示 sigmoid 曲线上的一次更新会移动 θ 。

采样到猜 A · 奖励 1： $R = 1$ ； $\partial / \partial \theta \log \pi_\theta(a)$ 的斜率为 $1 - p = +0.75$ ；梯度为 $+0.75$ 。采样 A 后，A 会变得更可能。

采样到猜 B · 奖励 0： $R = 0$ ； $\partial / \partial \theta \log \pi_\theta(a)$ 的斜率为 $-p = -0.25$ ；梯度为 0。采样 B 时没有变化。

这个梯度是正确的，但基线 / 优势可以让它更有用。

简单示例：双臂赌博机

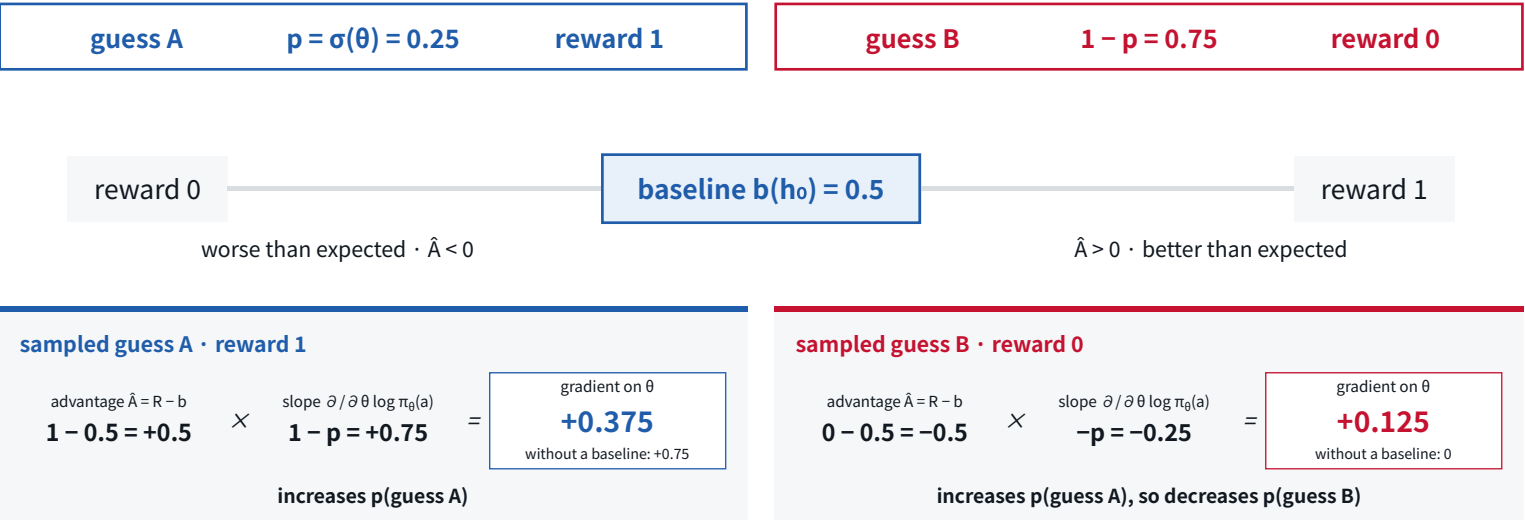
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Simple example: Adding a baseline

Subtracting a baseline says whether a reward was better or worse than expected.



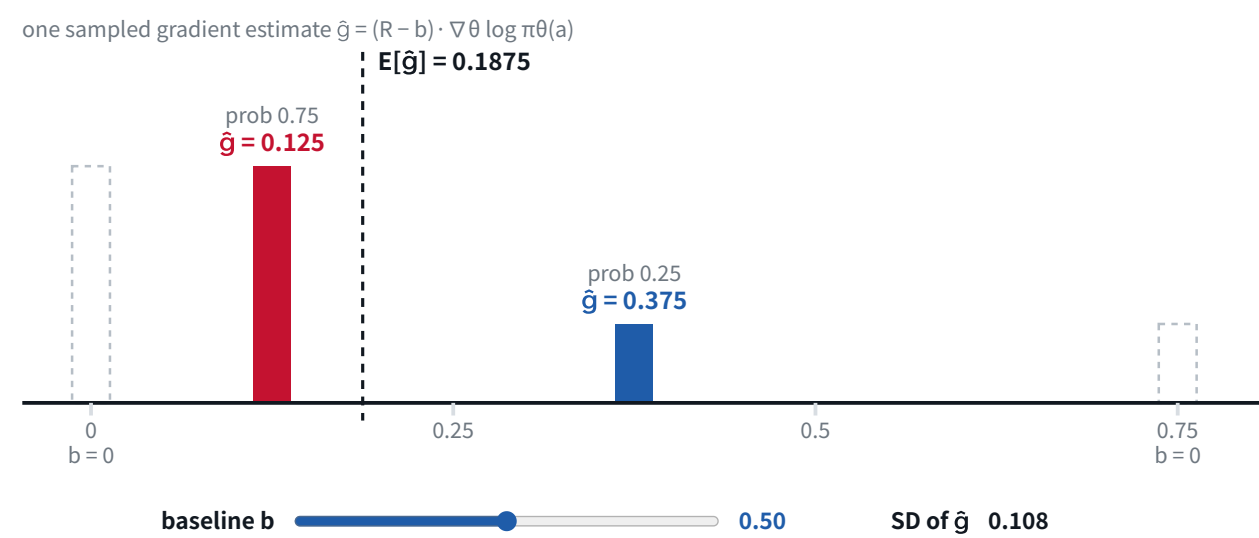
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Simple example: varying the baseline

Changing the baseline changes the variance of the gradient estimate, while keeping the expectation the same



32

改变基线会改变梯度估计的方差，同时保持期望相同。

一次采样的梯度估计 $\hat{g} = (R - b) \cdot \nabla_{\theta} \log \pi_{\theta}(a)$; $E[\hat{g}] = 0.1875$ 。

概率 0.75: $\hat{g} = 0.125$; 概率 0.25: $\hat{g} = 0.375$ 。

基线 b: 0、0.25、0.50、0.75; 标准差 $SD(\hat{g}) = 0.108$ 。

简单示例：改变基线

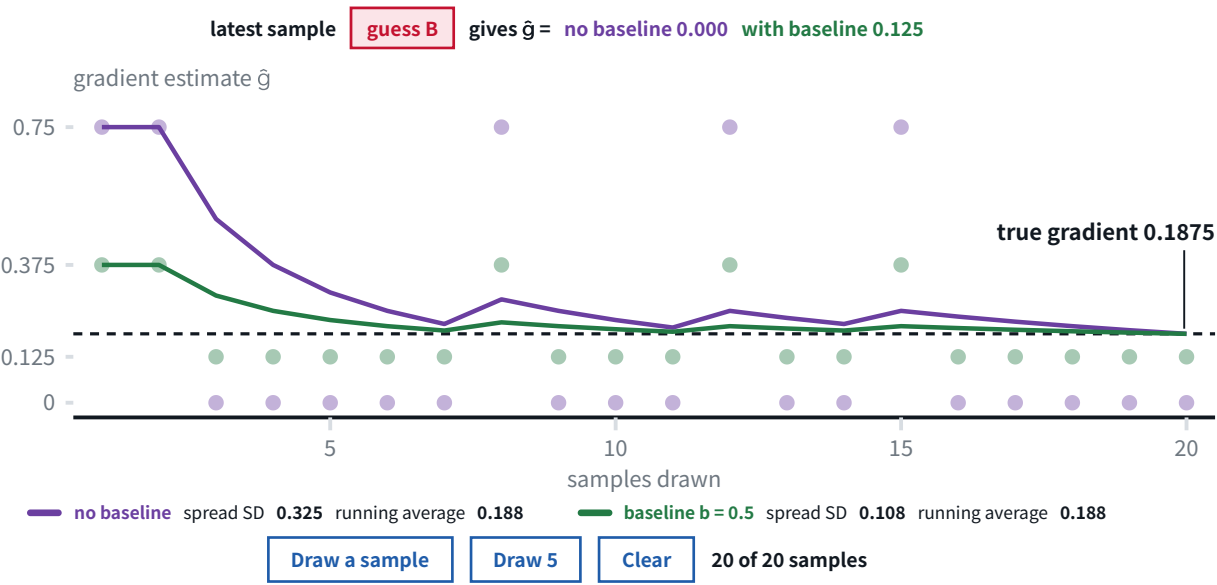
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Simple example: noise in the gradient estimate

Computed a running estimate of the gradient by repeated sampling, with no baseline and with baseline 0.5.



Both averages converge to the same value, but the baseline has lower variance.

33

重复采样，在无基线和基线为 0.5 的情况下计算梯度的滑动估计。

最新样本：猜 B；无基线时 $\hat{g}=0.000$ ，使用基线时 $\hat{g}=0.125$ 。

纵轴为梯度估计 $\hat{g}$ ，图中标记 0、0.125、0.375、0.75；横轴为已采样数 5、10、15、20；真实梯度 0.1875。

无基线：分布标准差 0.325；滑动平均 0.188。基线 $b=0.5$ ：分布标准差 0.108；滑动平均 0.188。

交互按钮：抽取一个样本；抽取 5 个样本；清空；当前为 20 个样本中的 20 个。

两种平均值都会收敛到相同数值，但基线的方差更低。

简单示例：梯度估计中的噪声

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Baselines and credit assignment

What are the advantages of baselines (so far)?

Helps

Marks a trajectory as better or worse than expected.
A good baseline reduces gradient variance.

Does not solve

All actions have the same advantage $\hat{A}$.
Estimating $V\pi(h)$ may require a separate value model.

$\hat{A} = +0.5$

guess 8

guess 12

guess 10

guess 11

34

基线目前有哪些优势？

有帮助：标记一条轨迹比预期更好还是更差；良好的基线能降低梯度方差。

无法解决：所有动作都有相同优势 $\hat{A}$ ；估计 $V^{\pi}(h)$ 可能需要单独的价值模型。

序列示例： $\hat{A}=+0.5$ ；猜 8 → 猜 12 → 猜 10 → 猜 11。

基线与信用分配

第 9 讲 · 智能体的强化学习基础

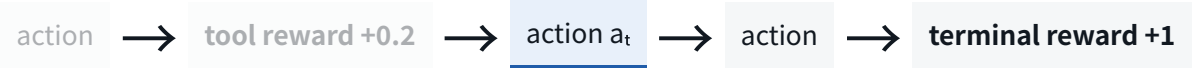
EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

What if we have intermediate rewards?

If the environment gives rewards before the final step, use reward-to-go to reduce variance.

$$R_t = \sum_{t'=t+1}^T r_{t'}$$



ignored included in R_t

- An action is weighted only by the rewards that come after it.
- Earlier rewards do not depend on a_t, so dropping them cuts variance without adding bias.
- Number Search pays out once, at the end: R_t = R(τ) for every action.

35

如果环境在最后一步之前就给出奖励，可以使用后续回报来降低方差。

$R_t = \sum_{t'=t+1}^T r_{t'}$ 。

动作 → 工具奖励 +0.2 → 动作 a_t → 动作 → 终止奖励 +1。

仅使用动作之后的奖励来给动作加权。

更早的奖励不依赖 a_t，因此去掉它们可以降低方差，同时不引入偏差。

数字搜索只在最后给奖励：对于每个动作，R_t=R(τ)。

如果有中间奖励呢？

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

比较轨迹

组相对优势估计

Comparing trajectories

Group-relative advantage estimates

36

比较轨迹

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Group-relative advantage estimates

Advantages: how much better is this trajectory than expected? We can compute this by sampling the policy multiple times. No separate model needed!

									reward R _j	centered R _j −R̄	divide by σ _R	advantage Â _j
τ ₁	8	higher	12	lower	10	higher	11	correct	1	+0.5	÷0.5	+1
τ ₂	4	higher	8	higher	12	lower	11	correct	1	+0.5	÷0.5	+1
τ ₃	16	lower	8	higher	9	higher	10	timeout	0	−0.5	÷0.5	−1
τ ₄	1	higher	2	higher	3	higher	4	timeout	0	−0.5	÷0.5	−1
from the group									R̄ = 0.5	σ _R = 0.5		

$$b(h_t) = \bar{R} = \frac{1}{G} \sum_{j=1}^G R(\tau_j) \quad \text{for every } h_t \text{ in the group}$$
$$\hat{A}_j = \frac{R(\tau_j) - \bar{R}}{\sigma_R}, \quad \bar{R} = b(h_t), \quad \sigma_R = \text{std}\{R(\tau_1), \dots, R(\tau_G)\}$$

37

优势：这条轨迹比预期好多少？可以通过多次采样策略来计算，不需要额外的模型！

τ₁: 猜 8 → 更高 → 猜 12 → 更低 → 猜 10 → 更高 → 猜 11 → 正确; R_j=1, 居中后 +0.5, 除以 0.5 后优势 +1。

τ₂: 猜 4 → 更高 → 猜 8 → 更高 → 猜 12 → 更低 → 猜 11 → 正确; R_j=1, 居中后 +0.5, 除以 0.5 后优势 +1。

τ₃: 猜 16 → 更低 → 猜 8 → 更高 → 猜 9 → 更高 → 猜 10 → 超时; R_j=0, 居中后 -0.5, 除以 0.5 后优势 -1。

τ₄: 猜 1 → 更高 → 猜 2 → 更高 → 猜 3 → 更高 → 猜 4 → 超时; R_j=0, 居中后 -0.5, 除以 0.5 后优势 -1。

本组均值 R=0.5; 组内每个 h_t 都使用该均值。

b(h_t)=R=(1/G)Σ_{{j=1}^G} R(τ_j); Â_j=(R(τ_j)-R)/σ_R; σ_R=std{R(τ₁),...,R(τ_G)}。

组相对优势估计

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

The GRPO update

Batches contain multiple task instances, each with its own group.

task instance	rewards of the G rollouts				group mean	advantage estimates $\hat{A}_{i,j}$			
x_1 · hidden 11	1	1	0	0	$\bar{R}_1 = 0.50$	+1	+1	-1	-1
x_2 · hidden 3	1	0	0	0	$\bar{R}_2 = 0.25$	+1.73	-0.58	-0.58	-0.58
x_3 · hidden 7	1	1	1	0	$\bar{R}_3 = 0.75$	+0.58	+0.58	+0.58	-1.73

policy-gradient loss · i indexes the batch

$$-\sum_i R(\tau_i) \sum_t \log \pi_{\theta}(a_{i,t} \mid h_{i,t})$$

Each task instance has its own R_i and σ_i .

Shao et al. · DeepSeekMath, § 4.1.2 ↗

GRPO loss · i batch, j group

$$-\sum_i \frac{1}{G} \sum_{j=1}^G \hat{A}_{ij} \sum_t \log \pi_{\theta}(a_{ij,t} \mid h_{ij,t})$$

Each task instance is normalized individually.

GRPO supplies group-relative weights to the same policy-gradient term.

Written for current-policy trajectories, without the paper's ratio, clipping, or KL terms.

38

批次包含多个任务实例，每个实例都有自己的组。

任务实例 x_1 (隐藏数字 11)：奖励 1、1、0、0；组均值 $R_1=0.50$ ；优势 +1、+1、-1、-1。

任务实例 x_2 (隐藏数字 3)：奖励 1、0、0、0；组均值 $R_2=0.25$ ；优势 +1.73、-0.58、-0.58、-0.58。

任务实例 x_3 (隐藏数字 7)：奖励 1、1、1、0；组均值 $R_3=0.75$ ；优势 +0.58、+0.58、+0.58、-1.73。

策略梯度损失：i 表示批次索引， $-\sum_i R(\tau_i) \sum_t \log \pi_{\theta}(a_{i,t} \mid h_{i,t})$ 。

GRPO 损失：i 表示批次，j 表示组， $-(1/G) \sum_i \sum_{j=1}^G \hat{A}^{ij} \sum_t \log \pi_{\theta}(a_{ij,t} \mid h_{ij,t})$ 。

每个任务实例有自己的 R_i 和 σ_i ；每个任务实例单独归一化。GRPO 将组相对权重用于相同的策略梯度项。

Shao 等 · DeepSeekMath, 第 4.1.2 节 ↗

此处写的是当前策略轨迹的形式，未包括论文中的比率、裁剪或 KL 项。

GRPO 更新

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

DrGRPO’s two changes

DrGRPO removes the group-standard-deviation and response-length divisions.

	GRPO	DrGRPO	Why it changes
Advantage estimate	$\frac{R_j - \bar{R}}{\sigma_R}$	$R_j - \bar{R}$	σ_R is measured inside each group, so it differs from one task instance to the next. Instances the policy almost always passes or almost always fails have a small σ_R , so they count for more.
Response aggregation	$\frac{1}{ y_j } \sum_t (\cdot)$	$\frac{1}{C} \sum_t (\cdot)$	$ y_j $ differs from one response to the next. A long incorrect response is penalized less per token than a short one, so the policy drifts toward longer wrong answers.

C is a global constant, the same for every response; the paper uses the generation budget.

Rewards stay fixed; the relative weights of task instances and responses change.

Liu et al. · DrGRPO, Fig. 1 and § 3.2 ↗

39

DrGRPO 移除了组标准差除法和响应长度除法。

GRPO 与 DrGRPO 比较：优势估计；响应聚合；变化原因。

优势估计：GRPO $(R_j - \bar{R}) / \sigma_R$ ；DrGRPO $R_j - \bar{R}$ 。 σ_R 在每个组内测量，因此随任务实例而异。策略几乎总能通过或几乎总会失败的任务，其 σ_R 较小，因此会获得更大权重。

响应聚合：GRPO $(1/|y_j|) \sum_t (\cdot)$ ；DrGRPO $(1/C) \sum_t (\cdot)$ 。 $|y_j|$ 随响应而异。长的错误响应每个词元受到的惩罚比短响应小，因此策略会逐渐偏向更长的错误答案。

C 是全局常数，对每个响应相同；论文使用生成预算。

奖励保持不变；改变的是任务实例与响应的相对权重。

Liu 等 · DrGRPO，图 1 与第 3.2 节 ↗

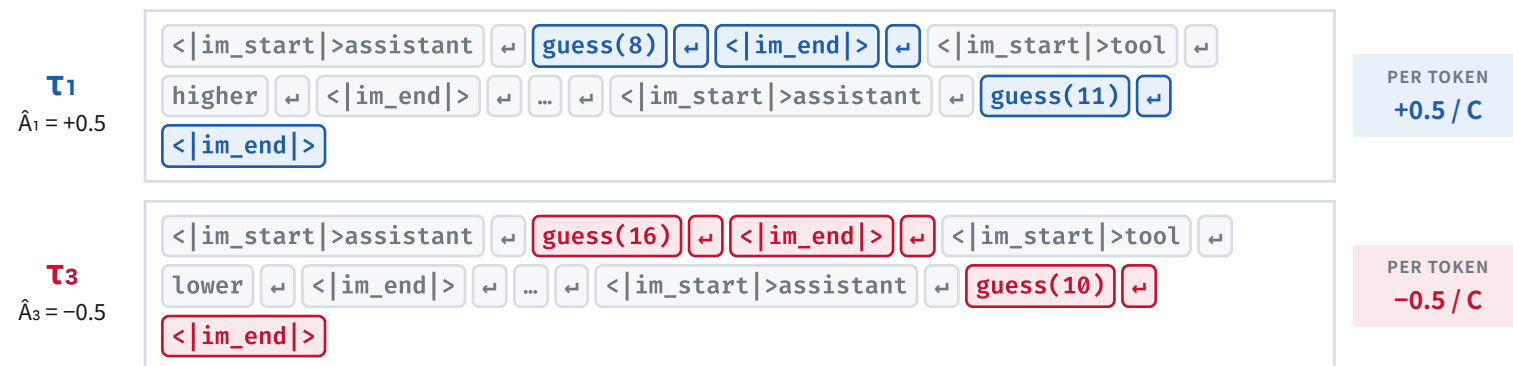
DrGRPO 的两项改动

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

Token coefficients in DrGRPO

For one task instance, every generated token in trajectory j shares that trajectory's advantage estimate.



$$\mathcal{L}_{\text{DrGRPO}} = -\frac{1}{G} \sum_{j=1}^G \frac{1}{C} \sum_{t: m_{j,t}=1} \hat{A}_j \log \pi_{\theta}(u_{j,t} \mid u_{j,<t})$$

$$\hat{A}_j = R(\tau_j) - R, \text{ with no division by } \sigma R.$$

Simplified: the paper's objective also has a ratio and clipping.

Liu et al. · DrGRPO, § 3.2 ↗

Shao et al. · DeepSeekMath, Eq. 3 ↗

40

ZH / 中文译文 · 第 1/1 部分

对于一个任务实例，轨迹 j 中的每个生成词元都共享该轨迹的优势估计。

轨迹 τ_1 : $\langle \text{lim_start} \rangle \text{assistant} \downarrow \text{guess}(8) \downarrow \langle \text{lim_end} \rangle \downarrow \langle \text{lim_start} \rangle \text{tool}$
 $\downarrow \text{higher} \downarrow \langle \text{lim_end} \rangle \downarrow \dots \downarrow \langle \text{lim_start} \rangle \text{assistant} \downarrow \text{guess}(11) \downarrow$
 $\langle \text{lim_end} \rangle$; $\hat{A}_1 = +0.5$; 每个词元 $+0.5 / C$ 。

轨迹 τ_3 : `<|im_start|>assistant` $\hookleftarrow$ `guess(16)` $\hookleftarrow$ `<|im_end|>` $\hookleftarrow$ `<|im_start|>tool`
 $\hookleftarrow$ `lower` $\hookleftarrow$ `<|im_end|>` $\hookleftarrow$... $\hookleftarrow$ `<|im_start|>assistant` $\hookleftarrow$ `guess(10)` $\hookleftarrow$
`<|im_end|>`; $\hat{A}_3 = -0.5$; 每个词元 $-0.5 / C$ 。

$$L_{\text{DrGRPO}} = -(1/G) \sum_{j=1}^G (1/C) \sum_{t: m_{j,t}=1} \hat{A}_j \log \pi_{\theta}(u_{j,t} | u_{j, < t}).$$
$$\hat{A}_j = R(\tau_j) - R, \text{ 不除以 } \sigma_R.$$

简化形式：论文目标还包含比率和裁剪。

Liu 等 · DrGRPO, 第 3.2 节 ↗ · Shao 等 · DeepSeekMath, 公式 3 ↗

DrGRPO 中的词元系数

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Equal rewards and zero advantage estimates

When every reward in a group matches, the group is degenerate and contributes no gradient.

ALL TRAJECTORIES FAIL

[0, 0, 0, 0]

→

[0, 0, 0, 0]

possibly too hard for the current policy

ALL TRAJECTORIES SUCCEED

[1, 1, 1, 1]

→

[0, 0, 0, 0]

possibly too easy for the current policy

What could we change if every sampled trajectory fails?

Choose different task instances, broaden exploration, add demonstrations, or supply a more informative reward.

41

当一个组里的所有奖励都相同，该组就退化，不会产生梯度。

所有轨迹均失败：[0, 0, 0, 0] → [0, 0, 0, 0]；当前策略可能认为任务太难。

所有轨迹均成功：[1, 1, 1, 1] → [0, 0, 0, 0]；当前策略可能认为任务太简单。

如果采样的每条轨迹都失败，我们可以如何调整？

选择不同任务实例、拓宽探索范围、加入示范，或提供信息量更大的奖励。

奖励相等导致优势估计为零

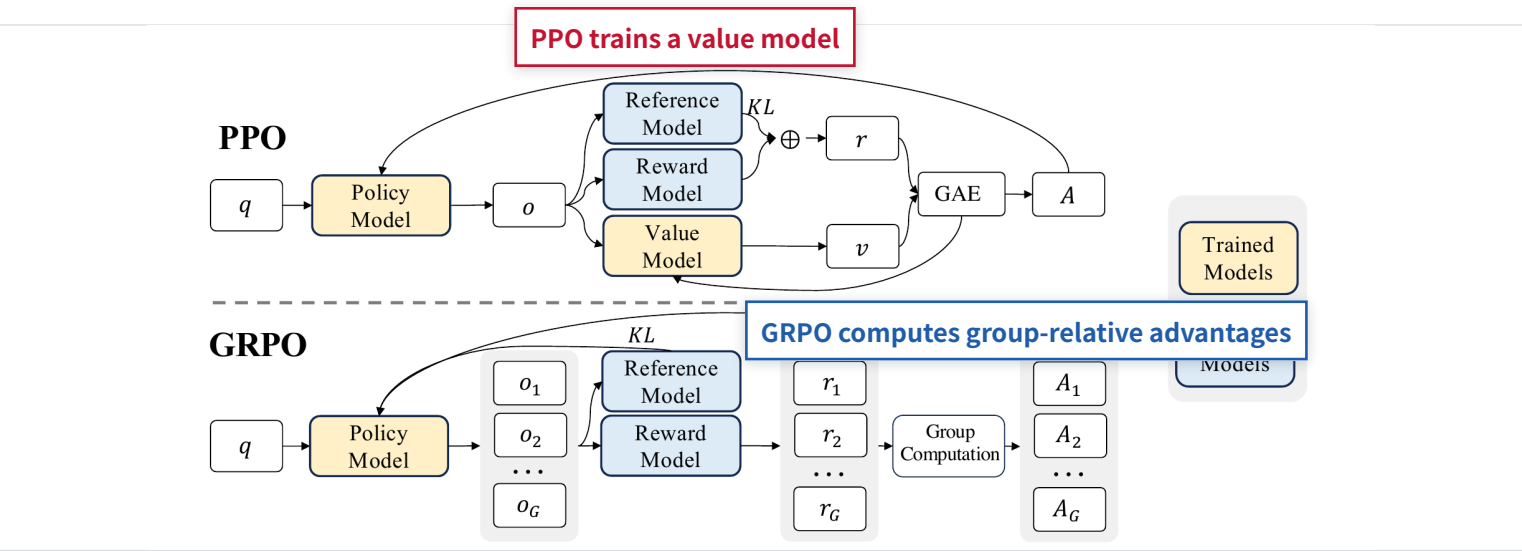
第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

GRPO vs PPO

GRPO was proposed as a more efficient alternative to PPO, which uses a separate model to estimate the advantage baseline.



Shao et al. · DeepSeekMath, Fig. 4 and § 4.1.2 ↗

The advanced RL lecture covers ratios, clipping, and reference-model terms.

42

GRPO 被提出作为 PPO 的一种效率更高的替代方法；PPO 使用单独的模型估计优势基线。

PPO 训练价值模型。

GRPO 计算组相对优势。

Shao 等 · DeepSeekMath, 图 4 与第 4.1.2 节 ↗

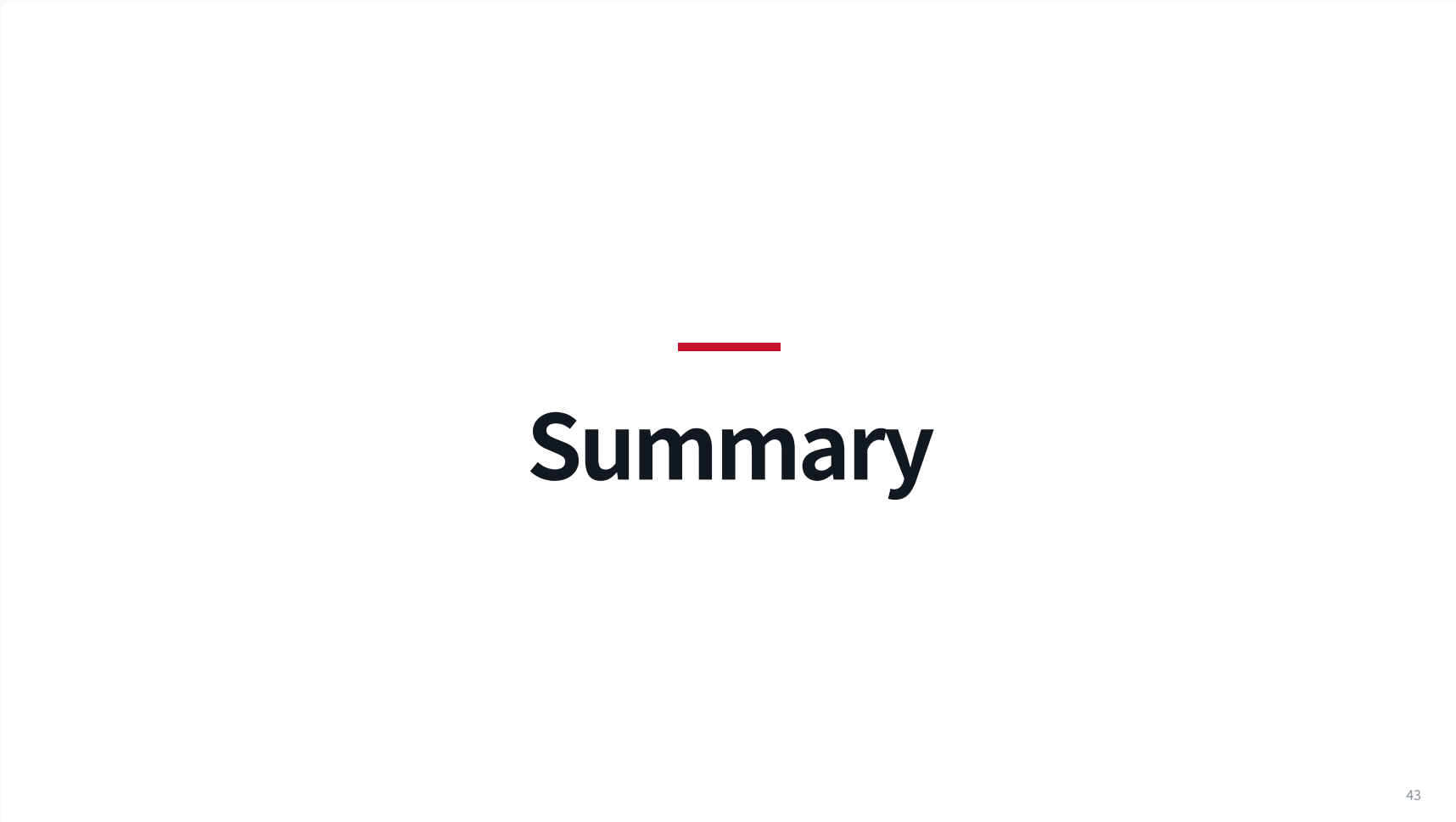
高级强化学习课程会介绍比率、裁剪和参考模型项。

GRPO 与 PPO

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



总结

总结

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Comparing the methods we’ve seen so far

Every method today weights the actions the agent took. They (mainly) differ in the weight.

$$\widehat{\nabla_{\theta} J} = \sum_t w_t \nabla_{\theta} \log \pi_{\theta}(a_t \mid h_t)$$

ReST-EM	$w_t = 1$ on a successful trajectory, 0 on the rest (the SFT loss on what was kept)
REINFORCE	$w_t = R(\tau)$, the reward of the trajectory the action came from
GRPO/DrGRPO	$w_t = \hat{A}_j$, the reward relative to the other rollouts on the same task instance

τ_1	8	higher	12	lower	10	higher	11	correct	$R = 1$
τ_2	4	higher	8	higher	12	lower	11	correct	$R = 1$
τ_3	16	lower	8	higher	9	higher	10	timeout	$R = 0$
τ_4	1	higher	2	higher	3	higher	4	timeout	$R = 0$

44

今天的每种方法都会对智能体采取的动作加权。它们主要区别在于权重。

$$\nabla_{\theta} J = \sum_t w_t \nabla_{\theta} \log \pi_{\theta}(a_t \mid h_t)。$$

ReST-EM：成功轨迹上 $w_t=1$ ，其余轨迹上为 0（对保留下来的数据计算 SFT 损失）。

REINFORCE： $w_t=R(\tau)$ ，即该动作所属轨迹的奖励。

GRPO / DrGRPO： $w_t=\hat{A}_j$ ，即相对于同一任务实例上其他轨迹采样的奖励。

τ_1 ：猜 8 → 更高 → 猜 12 → 更低 → 猜 10 → 更高 → 猜 11 → 正确， $R=1$ 。

τ_2 ：猜 4 → 更高 → 猜 8 → 更高 → 猜 12 → 更低 → 猜 11 → 正确， $R=1$ 。

τ_3 ：猜 16 → 更低 → 猜 8 → 更高 → 猜 9 → 更高 → 猜 10 → 超时， $R=0$ 。

τ_4 ：猜 1 → 更高 → 猜 2 → 更高 → 猜 3 → 更高 → 猜 4 → 超时， $R=0$ 。

比较目前为止介绍的方法

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Takeaways

Sampling trajectories from the policy and weighting them by reward addresses the three problems we started with.

Task mismatch

The objective is expected reward, so any trajectory that succeeds is reinforced.

Data mismatch

The agent can also learn from sub-optimal trajectories, based on their reward.

Exposure bias

The trajectories come from the current policy, so the agent is trained to recover from its own mistakes.

45

从策略中采样轨迹，并根据奖励赋予权重，可以解决我们一开始提出的三个问题。

任务目标不匹配：目标是期望奖励，因此任何成功轨迹都会得到强化。

数据不匹配：智能体也能根据奖励从次优轨迹中学习。

暴露偏差：轨迹来自当前策略，因此训练智能体从自己的错误中恢复。

要点

第 9 讲 · 智能体的强化学习基础

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



Questions?

Upcoming: async RL / off-policy data, importance weighting

46

问题？

后续内容：异步 RL / 离线策略数据、重要性加权

问答