【强化学习】直通PPO算法

TRiddle
阿里巴巴 员工

已关注

收录于 • RL from scratch

226 人赞同了该文章

只要花30分钟，你就能轻松入门ChatGPT的秘密武器RLHF中的核心——PPO算法。

首先我们用简短的篇幅复习一下强化学习的基本概念，后续的算法会基于这些概念进行讲解；接着学习Actor-Critic算法和A2C算法，学完这两个算法之后，我们就能够掌握PPO最本质的思想；最后我们来完成终极目标——学习PPO算法。

强化学习

强化学习是什么呢?

> 强化学习是一种解决控制（或决策）任务的框架，它从**环境**中试错并获得**奖励**（正或负），然后将其视作反馈从而进行学习。

其中，负责决策和试错的智能体被我们称为**agent**。可以简单地类比为监督学习中的机器学习或深度学习模型，是一个可学习的函数。

【强化学习】直通PPO算法

首发于
RL from scratch

强化学习过程包含若干个**episode**，每个episode包含若干**step**。

例如，围棋的一局，超级马里奥游戏中从游戏开始到救出公主的过程，或者语言模型生成一个句子的过程，这些都是一个episode。围棋中某位棋手的一次落子，超级马里奥游戏中玩家的一次操作，或者语言模型生成句子中的一个token，这些都是一个step。

第t个step中，agent与环境交互包含以下步骤（如上图）：

1. agent收到来自环境的**状态**S_t
2. 基于该状态 S_t ，agent采取**动作**A_t
3. 环境进入新状态 S_{t+1}
4. 环境会给agent带来一些奖励 R_t

如何理解状态、动作和奖励呢？

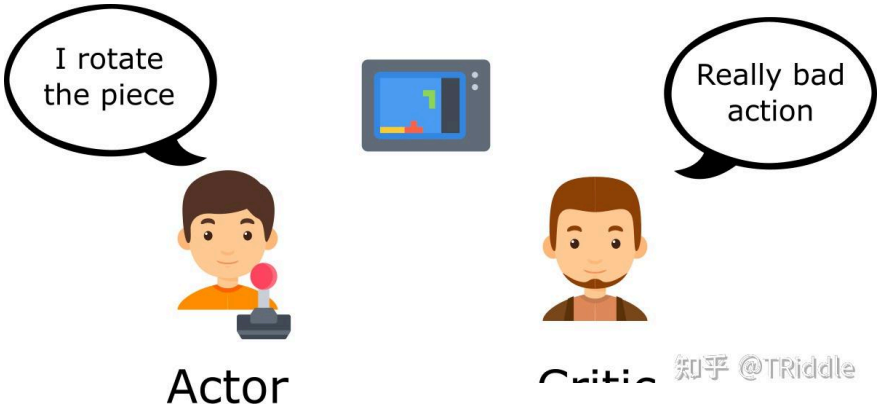
- 如果任务是下围棋，那么状态就是围棋中的局面（每个棋子的摆放位置和落子顺序），动作就是落子，奖励就是最终的输赢
- 如果任务是玩超级马里奥游戏，那么状态就是屏幕上所有元素（马里奥、怪物、管道等）的排列，动作就是按下手柄上的某个键，奖励就是吃到蘑菇或者赢得游戏
- 如果任务是语言模型的一次句子生成，那么状态就是当前已经生成的token，动作就是生成一个token，奖励就是最终人类对这个句子的喜好

我们希望一个episode中所有奖励之和能够越大越好。因此agent的目标是最大化一个episode中所有奖励之和的期望（之所以是期望而不是精确值，是因为采取动作后进入哪个新状态是环境说了算的，具有一定的随机性）。

如何做到呢？agent一遍遍地经历强化学习过程，一边收集数据，一边更新参数。最终就能够达成目标。

Actor-Critic算法

现在我们来学习Actor-Critic算法。在这部分我会花最多的篇幅，因为它是PPO算法的基础。



关于作者

TRiddle
现役搜索 | NLP菜鸡，退役...
阿里巴巴 员工

回答 8
文章 10
关注者 1,470

已关注
发私信

知乎 【强化学习】直通PPO算法

首发于
RL from scratch

演员和评论家就好像是正在玩俄罗斯方块的你和正在看你玩游戏的朋友。你一开始不知道怎么玩，所以随机尝试一些动作。你朋友观察你的行为并提供反馈。你从这些反馈中学习，然后更新策略从而更好地玩游戏。另一方面，你朋友也会更新他提供反馈的方式，以便下次更好地给出反馈。

- 演员是我们最终需要的agent，负责选择动作。其内部有一个概率分布 p(A_t|S_t) ，指导演员在特定的状态 S_t 下选择动作 A_t 。这个概率分布又被称为“**策略**”。
- 评论家是一个辅助模型，负责预估该动作的**收益**，也就是状态 S_t 开始，选择动作 A_t 后，直到结束为止能够获得的奖励之和的期望 Q(S_t, A_t) 。这种收益又被称为“**状态动作价值**[+]”。

问：评论家存在的意义是什么？

答：直观地说，**如果没有评论家，你就无法提前得知当前动作的价值**（必须得等到episode结束才行）。

顺带提一句，演员和评论家都可以用神经网络来建模

- 可以用输入一个向量 S_t ，输出一个概率分布的神经网络来建模 p(A_t|S_t)
- 可以用输入两个向量 S_t, A_t ，输出一个标量的神经网络来建模Q(S_t, A_t)

刚刚提到过，强化学习过程中的一个step要发生4件事。那么演员和评论家在一个step中，在这4件事发生的时候要做什么才能学到合适的参数呢？

1. 演员收到来自环境的状态 S_t
2. 演员生成动作 A_t ，然后评论家估计状态动作价值 Q(S_t, A_t) 。演员用 loss = -\log p(A_t|S_t) Q(S_t, A_t) 来更新参数
3. 环境收到 A_t 之后给出 S_{t+1} ，更新参数后的演员用 S_{t+1} 生成 A_{t+1}
4. 环境给出 R_t ，评论家用 loss = [Q(S_{t+1}, A_{t+1}) + R_t - Q(S_t, A_t)]^2 来更新参数

我们该怎么理解演员的loss呢？

说人话就是对状态 S_t 而言动作A_t的价值越大，演员就越要强化 A_t，否则就要弱化A_t。这有点像巴普洛夫的狗，演员会逐渐对需要强化的动作产生条件反射。

我们来分析一下：

- 当 Q(S_t, A_t) 大于0时：Q(S_t, A_t) 的绝对值越高或者 p(A_t|S_t) 越高，loss也就越低。此时演员必须更新参数来增大p(A_t|S_t)，更新的幅度受 Q_(S_t, A_t) 的影响
- 当 Q(S_t, A_t) 小于0时：Q(S_t, A_t) 的绝对值越高或者 p(A_t|S_t) 越高，loss也就越高。此时演员必须更新参数来减小p(A_t|S_t)，更新的幅度受 Q_(S_t, A_t) 的影响

我们又该怎么理解评论家的loss呢？

说人话就是评论家在得到新的信息后，需要改进自己预估的能力。例如，曾评论过梵高画作的评论家，如果“有幸”能够活到今天，就应该能通过梵高画作在如今的价值明白，自己的的评论能力已经跟不上这个时代了。

我们再来分析一下，评论家预估出的状态动作价值 Q(S_t, A_t) 可以分解为两部分： Q(S_t, A_t) = Q(S_{t+1}, A_{t+1}) + \hat{R}_t 。其中 \hat{R}_t 是第t个step的预估奖励，Q(S_{t+1}, A_{t+1}) 是第t+1个step之后所有step的预估奖励之和。

现在，环境告诉我们第t个step的真实奖励是 R_t，我们用这个奖励替换掉预估奖励之后，这个等式就不成立了，也就是说 Q(S_t, A_t) \ne Q(S_{t+1}, A_{t+1}) + R_t 。所以才需要用不等号两边的数值的差来定义loss。在获得新的信息之后，通过loss更新参数，评论家的认知差就被抹平了。

总之，在强化学习过程中，演员逐渐形成条件反射，评论家的评论越来越准确。到最后我们就可以可以用演员来做决策了。

【强化学习】直通PPO算法

首发于 RL from scratch

接下来我们来学习A2C（Advantage Actor-Critic）算法。它是Actor-Critic算法的改良，只要再加一点小改动就是PPO算法了。

它的思想很简单。假如你和你的朋友都是学生，你平时考试考90分，他平时考试考60分。经过一个月的期末复习，在期末考试中你考了96分，他考了95分。你觉得谁的期末复习策略是成功的?

显然你朋友的期末复习策略是更成功的。虽然你考了更高的分数，但这个分数基于你平时的积累，相当于是正常发挥了。而你朋友却是超常发挥。因此单看期末，他的复习策略更值得他好好强化。

我们再来看A2C算法。在其中，演员不参考评论家预测的收益的大小来更新参数，而是根据实际收益超出评论家预期收益的程度来更新参数。这样比较合理，也训练过程也更加稳定。例如，你平时考90分，期末考96分，超出预期的程度是6；而你朋友平时考60分，期末考95分，超出预期的程度就是35。因此A2C算法也觉得你朋友的期末复习策略更值得强化。

这种“超出预期的程度”在A2C算法中被称为**优势（Advantage）**。优势为正数表示超出预期，否则表示低于预期。下面我们用Adv来表示。

A2C算法的步骤与Actor-Critic算法的差别不大，因此就直接给出了：

1. 演员收到来自环境的状态 S_t，生成动作 A_t
2. 环境收到A_t 之后给出奖励 R_t 和新状态 S_{t+1}
3. 评论家估计状态价值 V(S_t)，V(S_{t+1}) 并计算优势 Adv(S_t, A_t) = V(S_{t+1}) + R_t - V(S_t)
4. 演员用 loss = -\log p(A_t|S_t) \text{Adv}(S_t, A_t) 更新参数
5. 评论家用 loss = [\text{Adv}(S_t, A_t)] ^2 更新参数

其中的V是新东西，它是从状态 S_t 开始，直到结束为止能够获得的奖励之和的期望，也被称为“状态价值函数”。就是说A2C中我们不学Q了，改学V。但其实它们建模的是差不多的东西。

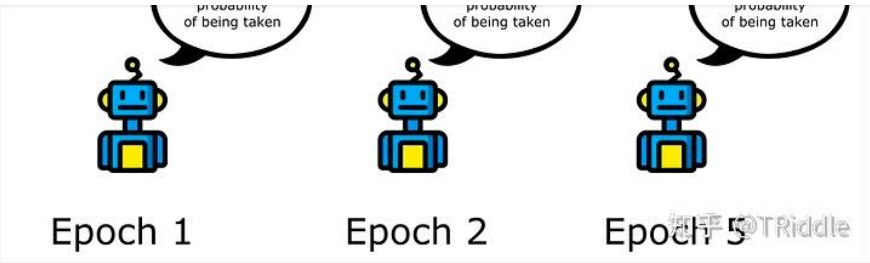
值得一提的是，在A2C算法中，我们先可以收集多个episode的多个step的数据，再一次性做参数更新。

PPO算法

终于，我们要开始学习PPO算法了，简单来说，PPO可以看作是一种特殊的A2C算法。

【强化学习】直通PPO算法

首发于
RL from scratch



PPO的思路是，为了维持训练的稳定性，想让策略 p(A_t|S_t) 的更新幅度不要太大。怎么操作呢？可以找一个东西来限制p。

回顾一下，在A2C中，对于每个状态 S_t 下采取的动作 A_t ，演员的loss是

loss = -\log p(A_t|S_t) \text{Adv}(S_t, A_t)

而在PPO中，演员的loss则是：

loss = -\frac{p(A_t|S_t)}{p'(A_t|S_t)} \text{Adv}(S_t, A_t)

其中p是本次参数更新前的策略，p'是上一次参数更新前的策略（梯度是不会回传到p'的），这个p’ 就是我们找来限制p的东西。

乍一看这个式子有点复杂，但其实有一种简单的理解方式。首先相比A2C的loss少了一个log，因为log是单调函数，所以可以暂时忽略这个变化。其次相比A2C的loss多了一个 \frac{1}{p'(A_t|S_t)} ，你可以把 -p(A_t|S_t) 以外的部分当成学习率（我们不让梯度经过p’ 回传到参数上）。当 \text{Adv}(S_t, A_t) > 0 且 p'(A_t|S_t) 很大时，这个“学习率”就会变得很小。

意思是在 S_t 状态下，如果动作A_t 能给你带来优势，但你预测 A_t 的概率已经很高了的话，为了维持训练的稳定性，就没必要再使劲更新参数了。

\text{Adv}(S_t, A_t) < 0 的情况可以自己分析一下。

现在，我们已经限制了策略 p(A_t|S_t) 的更新幅度，但还缺少一个“熔断机制”。什么意思呢？就是万一策略的更新幅度还是太大了，我们要停止策略的参数更新。

PPO的做法是什么呢？因为 \frac{p(A_t|S_t)}{p'(A_t|S_t)} 衡量了旧策略和现行策略之间差异，所以可以为它设置两个阈值。为了方便描述，我们令 r(A_t, S_t) = \frac{p(A_t|S_t)}{p'(A_t|S_t)}：

- 当Adv大于0时，若r大于1.2，则停止参数更新
- 当Adv小于0时，若r小于0.8，则停止参数更新

用一个式子就能描述这种“熔断机制”：

loss = -\min(r(A_t, S_t) \text{Adv}(S_t, A_t),\ \text{clip}(r(A_t, S_t) , 0.8, 1.2) \text{Adv}(S_t, A_t))

其中clip(r, 0.8, 1.2)表示：当r小于0.8时，clip函数值为0.8，当r大于1.2时，clip函数值为1.2，否则clip函数值为r。

来验证一下新的loss是否实现了“熔断机制”吧：

- Adv大于0：r大于1.2之后，min操作就会取右边的值；此时loss中就只剩常量了，不产生任何梯度；而r无论多小都还是会产生梯度
- Adv小于0：r小于0.8之后，min操作就会取右边的值，此时loss中就只剩常量了，不产生任何梯度；而r无论多大都还是会产生梯度

到此就全部结束了。最后，我们用一

【强化学习】直通PPO算法

首发于 RL from scratch

所属专栏 · 2025-02-07 23:13 更新

RL from scratch

TRiddle 阿里巴巴 员工

3 篇内容 · 431 赞同

订阅

最热内容 · 【强化学习】多臂老虎机的上置信界算法（全网最通俗）

编辑于 2025-03-24 10:24 · 北京

强化学习 (Reinforcement Learning) 深度强化学习 PPO算法

理性发言，友善互动

26 条评论 默认 最新

刘坤

感谢😊😊😊😊

2023-07-19 · 江苏 回复 3

刘坤 ‣ **StevenJokes**

早都放弃了🤣🤣🤣学开发来福包厂了

2024-07-22 · 浙江 回复 1

StevenJokes

学会了，我可以去OpenAI工作了吧😡

2024-07-22 · 浙江 回复 喜欢

暮春之夏

AC算法中的收益Q(St,At)，是状态 St开始，选择动作At后，直到结束为止能够获得的奖励之和的期望 。
A2C算法中的价值V(St,At)，是从状态St开始，直到结束为止能够获得的奖励之和的期望。
可以认为AC算法中的收益Q 和A2C算法中的价值V是一个东西吗?

2023-12-14 · 北京 回复 1

TRiddle 作者

看起来很像但其实不是一个东西。在强化学习中，V(St)叫做状态价值函数，Q(St, At)叫做动作价值函数。它们之间是存在转化关系的，Q(St, At)对求期望消掉At就能得到V(St)

2023-12-19 · 北京 回复 2

cerax

有两个问题请教一下。
ppo中，adv(st,at)小于0的情况下，可以看做学习率是负的，也就是 在at不能带来优势的时候，对于old model认为输出概率高的at，我们应该是反向优化（梯度上升），并且考虑到不偏移的限制，这个负向优化的程度也被p‘缩小，这么理解对吗?
2. a2c算法中评论家估计状态价值 Vst Vst+1并计算adv = Vst+1 + Rt - Vst, 这个Rt是reward model给出的实际观测奖励对吧，但是我们的模型不都是只在最后一步给出奖励吗？也就是Rt = 0，Rn = score。是不是有一个奖励衰减没有介绍，还是我理解的哪里有问题?

2023-09-06 · 湖北 回复 1

TRiddle 作者

对于第一个问题，你的理解是准确的。对于第二个问题，你提到的奖励衰减和获得奖励的频率其实是两个问题；首先奖励衰减确实没介绍；其次有些场景的确是到最后才得到一个来自环境总奖励，例如围棋的输赢；但在另一些场景下，中途也可能会有来自环境奖励，例如在电子游戏中获得经验和装备

2023-09-07 · 北京 回复 1

小鱼儿at青岛

ppo更新幅度那里我觉得从反
是个c=a*b问题，求c关于a的

知乎 【强化学习】直通PPO算法

首发于 RL from scratch

23 小时前 · 山东 回复 喜欢

会打球的赤司
之前看别的都不太理解，这个应该是最通俗易懂的了😚
02-05 · 浙江 回复 喜欢

雾里
而r无论多小都还是会产生梯度
这句话怎么理解？意思是不被剪切的情况下吗
2024-12-02 · 新加坡 回复 喜欢

TRiddle 作者
对的～就是梯度不被剪切的情况
2024-12-24 · 浙江 回复 喜欢

马特兰博
面试要是让我讲讲ppo算法，这哪里讲的明白啊
2024-09-12 · 北京 回复 喜欢

TRiddle 作者
也许可以先简要讲讲，然后反问面试官想具体了解哪些内容～
2024-09-14 · 北京 回复 喜欢

Consolas
你好，我想问一下A2C里，critic的损失是不是写错了呀，应该是Adv(s_t,a_t)^2?
2024-08-01 · 福建 回复 喜欢

TRiddle 作者
确实是，已更正。感谢反馈😚
2024-08-06 · 北京 回复 喜欢

Consolas ‣ **TRiddle**
啊 感谢回复 😚😚
2024-08-06 · 福建 回复 喜欢

Libertax
无敌的😚
2024-08-01 · 浙江 回复 喜欢

TRiddle 作者
😚
2024-08-06 · 北京 回复 喜欢

wzz
深入浅出的神。 能写深入浅出文章的前提是自己要吃透知识点。
2024-04-04 · 浙江 回复 喜欢

TRiddle 作者
谢谢鼓励～谬赞了😚
2024-04-07 · 北京 回复 喜欢

点击查看全部评论 ›

理性发言，友善互动

推荐阅读

【强化学习】PPO算法代码详解

【强化学习】PPO算法

[强化学习-08]--PPO

知乎 【强化学习】直通PPO算法

首发于 RL from scratch

法，由OpenAI在2017年提出。PPO结合了策略梯度方法的优点和信任区域优化（Trust Region...

禾风

于价值的强化学习算法 DQN,基于策略的强化学习算法REINFORCE,基于价值和策略的组合算法Actor-...

快速实现AI算法

r)，计算折扣奖励以及优势函数
buffer_s.append(s)
buffer_a.append(a)...

飞翔