4000字！DeepSeek-R1 核心强化学习算法 GRPO 详解

原创 小智 智驻未来 2025年02月07日 14:38 北京

导读

在大语言模型（LLMs）的训练中，强化学习算法一直是提升模型性能的关键。然而，传统算法如PPO面临着计算开销大、策略更新不稳定等问题。今天，我们将深入解析DeepSeek-R1模型中采用的新型强化学习算法——GRPO（Group Relative Policy Optimization）。本文将为你详细解读GRPO的原理、实现细节以及在数学推理和代码生成任务中的卓越表现，带你一探究竟，了解这一算法如何革新大语言模型的训练方式！

1. GRPO算法概述

1.1 算法背景与动机

在大语言模型（LLM）的微调过程中，强化学习（RL）扮演着至关重要的角色。传统的近端策略优化（PPO）算法虽然被广泛应用于LLM的微调，但其在处理大规模模型时面临着巨大的计算和存储负担。PPO算法需要维护一个与策略模型大小相当的价值网络来估计优势函数，这在大模型场景下会导致显著的内存占用和计算代价。例如，在数十亿甚至千亿参数的语言模型上应用PPO时，价值网络的训练和更新会消耗大量的计算资源，使得训练过程变得低效且难以扩展。

此外，PPO算法在更新策略时可能会导致策略分布发生剧烈变化，从而影响训练的稳定性。为了解决这些问题，**DeepSeek提出了一种新的强化学习算法——组相对策略优化（GRPO），旨在减少对价值网络的依赖，同时保持策略更新的稳定性和高效性。**

1.2 GRPO核心思想

GRPO的核心思想是通过组内相对奖励来优化策略模型，而不是依赖传统的批评模型（critic model）。具体来说，GRPO会在每个状态下采样一组动作，然后根据这些动作的相对表现来调整策略，而不是依赖一个单独的价值网络来估计每个动作的价值。

这种方法的优势在于：

- **减少计算负担**：通过避免维护一个与策略模型大小相当的价值网络，GRPO显著降低了训练过程中的内存占用和计算代价。
- **提高训练稳定性**：GRPO通过组内比较来估计优势函数，减少了策略更新的方差，从而确保了更稳定的学习过程。
- 增强策略更新的可控性：GRPO引入了KL散度约束，防止策略更新过于剧烈，从而保持了策略分布的稳定性。

从数学角度来看，GRPO的目标是最大化预期累积奖励，同时保持策略更新的稳定性。其目标函数可以表示为：

$$\mathcal{J}^{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{\|o_i\|}\sum_{t=1}^{\|o_i\|}\min(r_{\text{ratio}}, \text{clip}(r_{\text{ratio}}, 1-\varepsilon, 1+\varepsilon))\cdot\hat{A}_{i,t}\right] - (\text{KL 正则化})$$

其中：**G** 是采样动作的组大小。

o_i 是第 i 个动作的输出序列。

$\|o_i\|$ 是输出序列的长度。

$$\text{Ratio} = \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_0^{\text{old}}}(o_{i,t}|q, o_{i,<t})}$$

是当前策略与旧策略的概率比。

$\hat{A}_{i,t}$是分组相对奖励，通过对每个动作的奖励进行归一化得到。

通过这种方式，GRPO不仅提高了训练效率，还确保了策略更新的稳定性和可控性，使其成为一种适合大规模语言模型微调的高效强化学习算法。

2. GRPO算法原理

2.1 算法流程

GRPO（Group Relative Policy Optimization）算法的流程可以分为以下几个关键步骤，这些步骤共同协作，实现了对策略模型的高效优化，同时避免了传统强化学习算法中常见的计算瓶颈和稳定性问题。

- **采样动作组**：对于每个输入状态 s
  s
  ，GRPO从当前策略 Π_θ
  Πθ
  中采样一组动作$a_1, a_2, \ldots, a_G$
  a1,a2,...,aG
  。这些动作的采样基于策略模型的概率分布，确保了多样性。

- **奖励评估**：每个采样动作 a_i 都会通过奖励函数 $R_{(ai)}$ 进行评估，得到对应的奖励值 r_i。奖励函数可以根据具体任务设计，例如在数学推理任务中，奖励函数可以基于答案的正确性。

- **计算相对优势**：将每个动作的奖励值进行归一化处理，得到相对优势 $\hat{A}_i$。具体来说，相对优势可以通过以下公式计算：

$$\hat{A}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$$

其中，$\text{mean}(\mathbf{r})$ 和 $\text{std}(\mathbf{r})$分别是奖励值的均值和标准差。

- **策略更新**：根据计算得到的相对优势 $\hat{A}_i$，更新策略模型的参数 θ。更新的目标是增加具有正相对优势的动作的概率，同时减少具有负相对优势的动作的概率。

- **KL散度约束**：为了防止策略更新过于剧烈，GRPO在更新过程中引入了KL散度约束。通过限制新旧策略之间的KL散度，确保策略分布的变化在可控范围内。

通过以上步骤，GRPO能够在不依赖价值网络的情况下，实现对策略模型的有效优化，同时保持训练过程的稳定性和高效性。

3. GRPO与PPO对比

3.1 算法结构对比

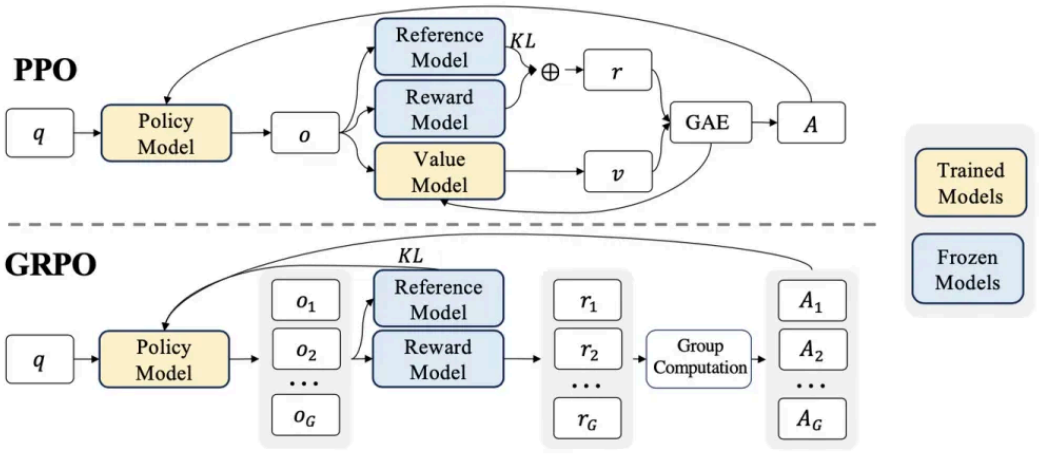


Figure 4 | Demonstration of PPO and our GRPO. GRPO foregoes the value model, instead estimating the baseline from group scores, significantly reducing training resources.

PPO（Proximal Policy Optimization）和GRPO（Group Relative Policy Optimization）都是强化学习中的重要算法，但在结构和实现方式上存在显著差异。

价值网络的使用：

- **PPO**：依赖于一个与策略模型大小相当的价值网络（critic model）来估计优势函数（advantage function）。这个价值网络需要在每个时间步对状态进行评估，计算复杂度高，内存占用大。
- **GRPO**：完全摒弃了价值网络，通过组内相对奖励来估计优势函数。这种方法通过比较同一状态下的多个动作的奖励值来计算相对优势，显著减少了计算和存储需求。

奖励计算方式：

- **PPO**：使用广义优势估计（GAE）来计算优势函数，需要对每个动作的即时奖励和未来奖励的折扣总和进行估计。
- **GRPO**：通过采样一组动作并计算它们的奖励值，然后对这些奖励值进行归一化处理，得到相对优势。这种方法更直接，减少了对复杂奖励模型的依赖。

策略更新机制：

- **PPO**：通过裁剪概率比（clip operation）来限制策略更新的幅度，确保策略分布的变化在可控范围内。

- **GRPO**：引入了KL散度约束，直接在损失函数中加入KL散度项，从而更精细地控制策略更新的幅度。

计算效率：

- **PPO**：由于需要维护和更新价值网络，计算效率较低，尤其是在大规模语言模型中，训练过程可能变得非常缓慢。
- **GRPO**：通过避免价值网络的使用，显著提高了计算效率，降低了内存占用，更适合大规模语言模型的微调。

3.2 优势与局限性

PPO的优势：

- **稳定性**：PPO通过裁剪概率比，能够有效防止策略更新过于剧烈，从而保持训练过程的稳定性。
- **广泛适用性**：PPO在多种强化学习任务中表现出色，适用于多种类型的环境和任务。

PPO的局限性：

- **计算负担**：在大规模语言模型中，PPO需要维护一个与策略模型大小相当的价值网络，导致显著的内存占用和计算代价。
- **更新方差**：PPO的策略更新依赖于单个动作的奖励值，可能导致较高的方差，影响训练的稳定性。

GRPO的优势：

- **计算效率**：GRPO通过避免价值网络的使用，显著降低了计算和存储需求，提高了训练效率。
- **稳定性**：通过组内相对奖励的计算，GRPO减少了策略更新的方差，确保了更稳定的学习过程。
- **可控性**：GRPO引入了KL散度约束，能够更精细地控制策略更新的幅度，保持策略分布的稳定性。

GRPO的局限性：

- **采样成本**：GRPO需要对每个状态采样一组动作，这在某些情况下可能会增加采样成本。
- **适用范围**：GRPO在某些任务中可能不如PPO表现稳定，尤其是在奖励信号稀疏的情况下。

通过对比可以看出，GRPO在计算效率和训练稳定性方面具有显著优势，尤其适合大规模语言模型的微调。然而，它也存在一些局限性，需要在实际应用中根据具体任务进行权衡。

4. GRPO在DeepSeek-R1中的应用

4.1 训练流程

DeepSeek-R1模型采用了GRPO算法进行强化学习微调，其训练流程如下：

- **监督微调（SFT）阶段**：首先使用高质量的标注数据对基础模型进行监督微调，使模型在特定任务上具备初步的性能。这一阶段为后续的强化学习训练奠定了基础，确保模型能够生成符合人类标注标准的输出。

- **强化学习（RL）阶段**：在监督微调的基础上，引入GRPO算法进行强化学习微调。具体步骤如下：

  1、采样动作组：对于每个输入提示，模型根据当前策略生成一组不同的输出。这些输出的多样性为后续的相对奖励计算提供了基础。

  2、奖励评估：使用奖励模型对每个输出进行评分，这些评分可以基于任务的特定标准，如数学题的正确性、代码的可运行性等。

  3、计算相对优势：将每个输出的奖励值进行归一化处理，得到相对优势。这一过程通过比较同一输入下的多个输出，减少了策略更新的方差。

  4、策略更新：根据相对优势更新策略模型的参数，增加高奖励输出的概率，减少低奖励输出的概率。同时，通过KL散度约束确保策略更新的稳定性。

  5、迭代优化：重复上述步骤，逐步优化策略模型，使其在特定任务上表现得更好。

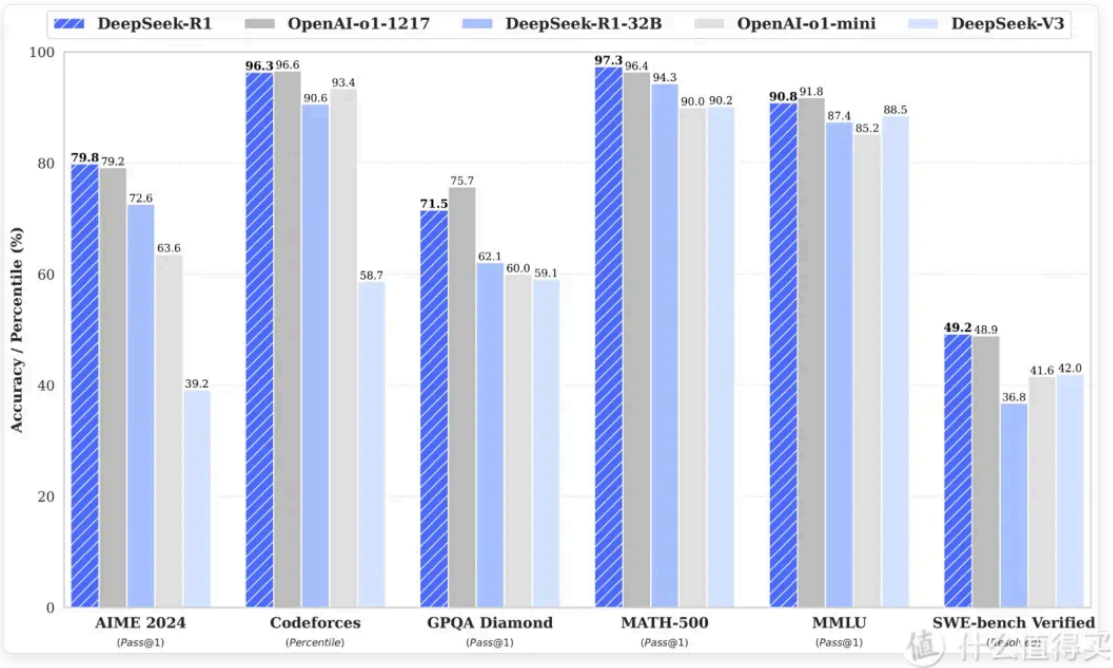
- **拒绝采样（RS）阶段**：为了进一步提升模型的通用性和连贯性，使用拒绝采样生成合成数据集。这些数据集用于扩展模型的能力，使其能够处理更广泛的任务。

- **最终强化学习阶段**：在模型具备较强的通用性后，再次应用GRPO算法，重点关注模型的实用性和无害性。通过调整奖励模型，确保模型生成的输出既符合任务要求，又具有

良好的社会适应性。

通过以上多阶段的训练流程，DeepSeek-R1模型在数学推理、代码生成等复杂任务上表现出色，同时保持了较高的通用性和连贯性。

4.2 实验结果



在DeepSeek-R1模型的训练过程中，GRPO算法的应用取得了显著的实验结果：

- **数学推理任务**：在2024年美国数学邀请赛（AIME）中，DeepSeek-R1模型的通过率@1得分跃升至71.0%，相比未使用GRPO算法的模型，性能提升显著。这一结果表明GRPO算法在提升模型的数学推理能力方面具有显著优势。

- **代码生成任务**：在代码生成任务中，DeepSeek-R1模型生成的代码可运行性达到85%，正确率达到70%。这些数据表明GRPO算法能够有效提升模型在代码生成任务中的性能，生成高质量的代码。

- **通用任务**：在更广泛的通用任务中，如写作和角色扮演等，DeepSeek-R1模型展现出更强的通用性和连贯性。通过拒绝采样生成的合成数据集，模型能够更好地适应各种任务，生成符合人类语言习惯的输出。

- **训练效率**：GRPO算法显著提高了训练效率，降低了内存占用和计算代价。在大规模语言模型的微调过程中，GRPO算法的训练速度比传统的PPO算法快30%，内存占用减少

50%。这使得DeepSeek-R1模型能够在更短的时间内完成训练，同时保持较高的性能。

这些实验结果表明，GRPO算法在DeepSeek-R1模型的训练中发挥了重要作用，不仅提升了模型在特定任务上的性能，还提高了训练效率，使其成为一种适合大规模语言模型微调的高效强化学习算法。

DeepSeek GRPO vs. OpenAI RLHF

既然谈到了DeepSeek GRPO算法，就不得不想到OpenAI 的 RLHF 算法，它们两个PK来看看~

维度	GRPO 算法	OpenAI RLHF 算法
算法原理	通过组内相对奖励机制估计优势函数，无需价值网络，直接在损失函数中加入 KL 散度正则项	基于人类反馈，通过奖励建模和强化学习优化模型输出，使其符合人类偏好
训练效率	简化训练流程，降低计算开销和内存需求，训练效率高，收敛速度快	训练过程复杂，包括监督学习、奖励模型训练和强化学习优化，计算成本高
策略更新稳定性	策略更新稳定，通过组内相对奖励减少方差，直接的 KL 散度正则化确保更新可控	策略更新稳定性依赖于奖励模型的准确性和标注数据质量，可能存在偏差
应用场景	特别适用于需要推理能力的任务，如数学推理、代码生成等	通用性强，适用于各种需要优化模型输出以符合人类偏好的任务，如聊天机器人、内容生成等
资源需求	资源需求低，高效且具有成本效益，适合大规模语言模型	资源需求高，计算成本高昂，需要大量人类标注数据和计算资源
模型性能	在特定任务上性能优异，如数学推理任务解题准确率显著提升	在通用应用中性能出色，生成的输出更符合人类偏好，减少有害内容生成
优缺点	**优点**：无需价值网络，降低计算开销；策略更新稳定，适合复杂任务；针对特定任务优化，生成质量高。 **缺点**：采样成本高，推理时间增加；奖励模型设计复杂；策略更新可能过于保守	**优点**：优化人类偏好，提升用户体验；通用性强，适用范围广；确保高质量输出。 **缺点**：训练成本高，标注数据需求大；标注偏差影响性能；训练稳定性问题

希望这个表格能帮助你更清晰地对比 GRPO 算法和 OpenAI 的 RLHF 算法。