第 1 卷◆第 3 期◆版本 1.0◆2023 年
文章类型：论文|刊号（ISSN）：2972-4236(P) / 2972-4244(O)

大语言模型中的上下文面板调节与奖励模型

文木源
GPT DESK PTE LTD
DOI:10.12238/acair.v1i3.6550

[摘　要] 直接偏好优化(DPO)旨在符合人类偏好,同时减少强化学习的复杂性。传统方法如人类反馈强化学习(RLHF)首先匹配奖励模型与提示和偏好,然后使用强化学习(RL)来找到最大化奖励的策略。相比之下,DPO通过直接优化策略来满足偏好,无需显式奖励函数或强化学习,简化了过程。DPO是微调语言模型以保持与人类反馈一致的更直接、更有效的方法。此外,OpenAI提到他们通过模仿人类评分来训练模型,以帮助改善RLHF。下一步是将模型拟合到含有丰富“条件”的数据集上,例如训练模型生成包含记忆、条件、目标、计划、未来任务的面板,并使用这个面板进行训练。这些条件将“创意写作任务”转变为“分配材料”的任务,减少了创意写作中的熵。条件强化学习微调(C-RLFT)使得大语言模型能够理解和生成类人文本、适应新信息和个性化响应,同时保持相关性和连贯性。未来的改进工作包括使用RLHF或RLAIF改善条件面板、数据集和模型之间的迭代、使模型与现实世界需求保持一致,以及基于0阶优化构建新的基础模型。这些方向旨在使大语言模型更高效、符合人类偏好,并能在各种环境中运行,包括边缘计算设备。
[关键词] 直接偏好优化；人类反馈强化学习；条件面板；创意写作熵降低；C-RLFT训练；边缘计算
中图分类号：TU205　文献标识码：A

Contextual Panel Conditioning and Reward Models in Large Language Models

Muyuan Wen
GPT DESK PTE LTD

[Abstract] Direct preference optimization (DPO) aims to match human preferences while reducing the complexity of reinforcement learning. Traditional methods such as reinforcement learning with human feedback (RLHF) first match reward models with cues and preferences, and then use reinforcement learning (RL) to find policies that maximize rewards. In contrast, DPO simplifies the process by directly optimizing the policy to satisfy preferences without explicit reward functions or RL processes. DPO is a more direct and potentially more efficient way to fine-tune a language model to remain consistent with human feedback. Additionally, OpenAI mentioned that they trained the model by imitating human ratings to help improve RLHF. The next step is to fit the model to a data set containing rich "conditions".For example, the training model generates a panel containing memories, conditions, goals, plans, and future tasks, and uses this panel for training. These conditions transform the "creative writing task" into the task of "distributing materials", reducing entropy in creative writing. Conditional reinforcement learning fine-tuning(C-RLFT)enables large language models to understand and generate human-like text, adapt to new information, and personalize responses while maintaining relevance and coherence. Future improvements include improving conditional panels using RLHF or RLAIF, iteration between datasets and models, aligning models with real-world needs, and building new base models based on 0-order optimization. These directions aim to make large language models more efficient, consistent with human preferences, and able to run in a variety of environments, including edge computing devices.
[Key words] Direct Preference Optimization; Human Feedback Reinforcement Learning; Conditional Panel; Creative Writing Entropy Reduction; C-RLFT Training; Edge Computing

在创意写作等复杂任务中,使用详细的上下文面板调节大语言模型(LLM)能显著地降低熵。通过使用详细的上下文(例如人物和地点描述以及作者的计划和记忆)将高熵创意写作任务转换为更加结构化的“指定材料的写作”任务,模型可以以高精度和接近零损失的方式执行。OpenAI使用了类似的技术,采用人工评分来根据人类反馈进行强化学习(RLHF)。该研究还结合了RecurrentGPT的方法,使用LLM从原始文本输入生成初始条件面板。训练包括两个部分：具有全局信息条件面板的任务以及学习从单个任务生成这些面板。实验表明,在没有进一步强化学习的情况下,模型在创意写作任务的评估数据集上表现良好,实现了低损失和高准确率。这与在纯文本上训练的模型形成鲜明对比,后者的表现并不那么合格。如果需要随机性或创造性,可以通过人类偏好或其他受控过程有意引入熵。这种平衡的方法可以维持大语言模型的生成能力,同时显著降低其产出的不可预测性。

1 缩放法则

缩放定律的主要目标是找出计算资源、数据集大小和模型参数之间的关系。它有助于解决以下问题：模型应该有多大,或者大语言模型需要多少数据,或者大语言模型应该有多少参数,等等。为了便于大家形象地了解这一问题,我们来看下面的图1至图4及附在图上的说明：

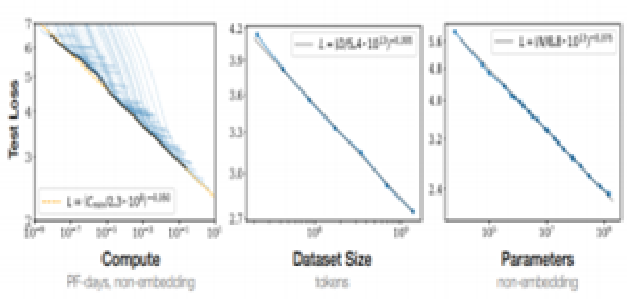

图1随着我们增加模型大小、数据集大小和训练所用的计算量,语言建模的性能平滑地提高。要达到最佳的性能,这三个因素必须同时按比例放大。当不受其他两个因素的限制时,实验性能与每个单独的因素呈幂律关系。

注：以上图示翻译自《模型的缩放定律》https://arxiv.org/pdf/2001.08361.pdf

1.1第一张图(计算)：显示了计算资源(以PF天衡量,代表PetaFlop天,计算能力的单位)和测试损失之间的关系。各种线代表不同的模型配置或实验,突出显示的虚线作为参考。如该图所示,随着计算资源的增加,测试损失呈幂律下降,这表明更多的计算能力可以显著提高模型性能。

1.2第二张图(数据集大小)：这张图说明了数据集大小的增加(以标记为单位)如何与测试损失的减少相关。标有$L=(D/5.4*10^{13})^{-0.095}$(1)的线性线表示特定的缩放法则,其中“D”代表数据集大小。这一趋势表明,在遵循幂律关系的基础上,更大的数据集往往会带来更好的模型性能。

1.3第三张图(参数)：这张图显示了参数数量(非嵌入)对测试损失的影响。我们再次观察到参数数量和测试损失之间的幂律关系,参数越多,测试损失就越低,这意味着模型性能得到提高。总之,随着模型大小、数据集大小和用于训练的计算量的增加,语言建模性能也会提高。然而,为了获得最佳性能,这些因素必须一起扩大。如果这三个因素中的任何一个没有按比例增加,它都可能成为瓶颈,限制扩展其他因素所带来的好处。最重要的是,语言建模的性能与这三个关键因素存在幂律关系,而平衡的缩放对于实现最佳结果至关重要。

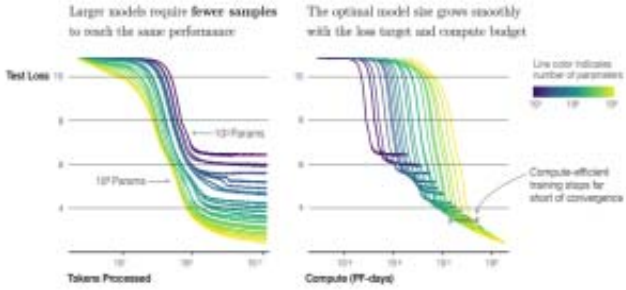

图2我们展示了一系列语言模型的训练过程,模型的大小从103到109个参数(不包括嵌入)不等。

注：以上图示翻译自《模型的缩放定律》https://arxiv.org/pdf/2001.08361.pdf

左图显示了处理的令牌数量(模型训练数据量的度量)和测试损失之间的关系。由图可见,随着参数数量的增加(由曲线向右移动表示),模型需要处理更少的令牌来实现更低的测试损失。这表明较大的模型样本效率更高,可以用更少的数据达到更好的性能。

右图显示出针对所使用的计算资源的测试损失(以PF天为单位)。由图可知,随着计算资源的增加,测试损失会减少。值得注意的是,对于每条曲线(模型大小),都有一个收益递减点,其中额外的计算不会显著降低测试损失。这意味着对于给定的损失目标和计算预算存在最佳模型大小。

以上是不同大小的语言模型的训练运行的效果,较大模型的效率更突出,因为需要更少的数据样本即可达到相同的性能水平。既然存在与损失目标和可用计算资源相关的最佳模型大小,那么,如果计算预算有限,模型并不总是越大越好。语言模型的效率和性能受到其参数数量和可用计算资源的影响,但为了获得最佳结果,需要保持平衡。

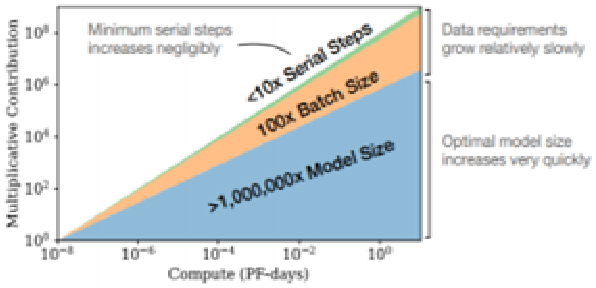

图3随着可用的计算量增加,我们可以选择如何分配计算量,用于训练更大的模型、使用更大的批次和进行更多的训练步骤。我们以计算量增加10亿倍为例进行说明。对于最优的计算效率训练,大部分的增加应该用于增加模型大小。为了避免重复使用数据,只需要相对较小的数据增加。在数据的增加中,大部分可以用于通过更大的批次大小来增加并行性,只需要非常小的串行训练时间的增加。

注：以上图示翻译自《模型的缩放定律》https://arxiv.org/pdf/2001.08361.pdf

随着计算资源的增加,模型的最佳大小似乎会迅速增加。这表明,凭借更强大的计算能力,可以更有效地训练更大的模型。

这张双对数图展示了计算资源(以PF天为单位)与“乘法贡献”之间的关系,而“乘法贡献”指的是模型的综合效益,即尺寸、批次大小和串行步骤数对模型训练过程的效率或有效性的影响。

随着计算资源的增加,模型的最佳大小似乎会迅速增加。这表明,凭借更强大的计算能力,可以更有效地训练更大的模型。

增加批次大小(在训练的每一步中一起处理的样本数量)的影响会大幅增长,但它的增长并不如模型大小那么急剧。这表明较大的批次大小有助于提高效率,但会出现收益递减的情况。

此外,随着计算量的增加,串行步骤(顺序处理步骤)的数量略有增加。这表明随着计算能力的增强,串行处理的需求只会略有增长。

最后,数据需求的增长相对缓慢,这意味着所需的数据量不需要像计算量或模型大小一样快速增长来保持模型的效率。

当我们有更多的计算资源时,我们可以选择如何分配:是训练更大的模型,使用更大的批次大小,还是训练更多的步骤？对于更高效的训练来说,大部分资源的增加应该用于更大的模型和更大的批次大小,而不是简单地更多数据或更长的训练时间。上面图示中说明了计算量增长十亿倍的策略,强调相对较小的数据增长就足以避免过于频繁地重用数据(可能导致过拟合),并且数据的增长应该主要用于增强计算能力。此外,我们应该通过更大的批次大小而不是更长的训练来实现并行性。

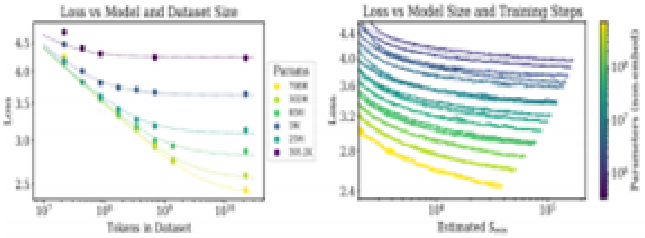

图4左图:根据方程(1.5),提前停止的测试损失L(N,D)随着数据集大小D和模型大小N的变化而呈可预测的变化。右图：经过一个初始的瞬态阶段,所有模型大小N的学习曲线都可以用方程(1.6)来拟合,该方程以Smin为参数,Smin是在大批量大小下训练时的步数(详见第5.1节)。

注:以上图示翻译自《模型的缩放定律》https://arxiv.org/pdf/2001.08361.pdf

左图显示了与数据集中令牌数量相关的损失,不同的曲线代表不同参数大小的模型(范围从393.2K到708M参数)。随着数据集中标记数量的增加,所有模型的损失都会减少。然而,较大的模型(具有更多参数)在相同的数据集大小下往往具有较低的损失,这表明它们在利用数据方面更有效。

右图则显示了与估计Smin相比的损失,这可能代表达到特定性能阈值所需的最小训练步骤数。右侧的色标表示模型中参数的数量。随着训练步骤数的增加,所有大小的模型的损失都会减少。参数数量较多的模型比较小的模型更快地达到较低的损失水平。

由图示可见,数据集的大小和模型的大小(就参数而言)都是模型性能的重要预测因素(通过损失来衡量)。此外,损失和这些变量之间的关系可以通过特定的方程来描述,表明随着数据量或训练步骤数量的增加,损失如何减少存在着可预测的模式。这些真知灼见来自OpenAI的一篇著名论文,它提出了一些构建大语言模型的重要见解,涉及如何获得最佳大语言模型、如何通过计算资源预算和数据集规模来估计模型大小和损失。这些规则即使在现在也非常有用,并吸引了许多后续研究,例如DeepMind的Chinchilla。

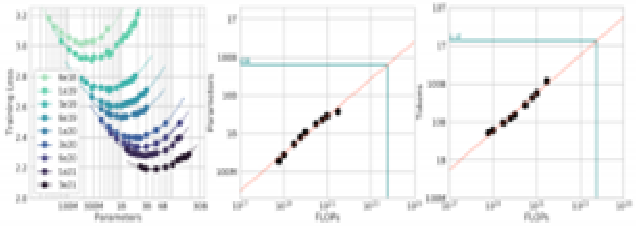

图5IsoFLOP曲线。对于不同的模型大小,我们选择训练的token数量,使得最终的FLOPs是一个常数。余弦周期长度被设定为与目标FLOP计数相匹配。我们发现损失中有一个明显的低谷,意味着对于给定的FLOP预算,有一个最优的模型可以训练(左图)。利用这些低谷的位置,我们预测了更大模型的最优模型大小和token数量(中图和右图)。用绿色,我们展示了用Gopher的计算预算训练的最优模型的参数和token的估计值。

注：该图翻译自《训练计算最优大型语言模型》https://arxiv.org/pdf/2203.15556.pdf

这是来自DeepMind的一篇论文,他们在这篇论文中提出了Chinchilla。这篇论文对缩放定律也做出了很大贡献。著名的开源大语言模型LLaMa系列跟踪了这项研究,并得到了相当积极的反馈。所有基准测试都有效,我们似乎找到了模型大小、数据大小和计算成本的最佳值。他们大致符合以下公式：

isoFLOPs=iso-FLOPs=sameFLOPs(2)

2 缩放法则并不能解决所有问题

为什么缩放法则并不能解决所有问题？缩放法则专注于训练事物,其目标是单一损失,训练集和验证集是静态的。这些事

情并不意味着缩放法则失效了，但它们使问题变得简单，有时甚至令人无法接受。

在现实世界的需求中，我们需要了解部署成本，并且我们的任务不仅仅是预测下一个单词，我们需要它来执行任务，并且任务在训练数据集中的频率可能较低。我们需要进行对齐，并且我们不知道对齐对模型大小/基础训练样本/基础训练步骤上的不同基础模型有何影响。我们还需要在数据大小和数据质量之间进行权衡，以某种方式，我们可以努力提高数据集质量，或者有意地制造更多数据，我们需要选择目标。此外，一旦我们的模型部署完毕，现实世界的数据量就会增加，我们可以在上面做人类反馈强化学习之类的事情。我们也会进一步了解到这些指标在不同规模的模型上有何不同，通过对齐可以解决多少问题。

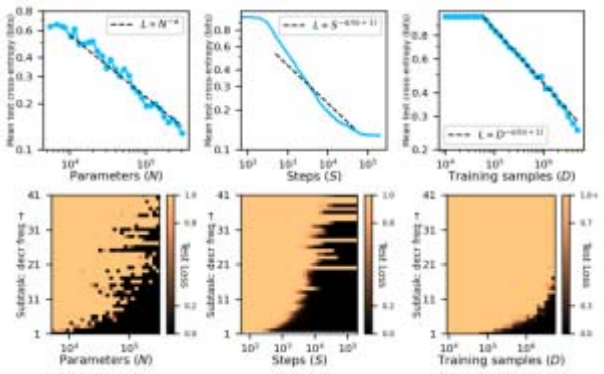

图6当在多任务稀疏奇偶性数据集上训练时，神经网络的损失关于参数N、训练时间S和训练样本D(对于多轮训练)呈现幂律缩放。这里α=0.4，我们绘制了线性∝Nα，∝Sα/(α+1)，∝Dα/(α+1)。下图：神经缩放按子任务分解。单个子任务的缩放行为表现出突现性，即子任务在特定的规模以上突然被学习。平均测试损失的幂律神经缩放平均了网络性能的大量定性变化(按子任务分解)，随着损失在越来越多的子任务上趋于零，这些子任务在频率上呈幂律分布。

注：该图翻译自《神经缩放的量化模型》https://arxiv.org/pdf/2303.13506.pdf

当模型在低频任务上表现不佳但与缩放定律保持良好一致时，它凸显了模型对齐阶段需要解决的挑战。出现此问题的原因是，基础模型通常是在某些任务代表性不足的数据集上进行训练的。因此，模型在这些领域的熟练程度是有限的。

一种可能的解决方案是调整基本模型以平衡任务频率。这意味着有意修改训练数据集，以确保更平等地表示不太频繁的任务。这种方法可以帮助模型在更广泛的任务中形成更平衡的理解和表现。

另一种方法是在基础训练阶段除了纯文本之外还训练模型的少量任务。少样本学习涉及针对每项任务使用少量示例来训练模型，教其从有限的数据中进行概括。这种方法可能会增强模型在训练数据中没有大量体现的任务上表现良好的能力，因为它学会了充分利用有限的信息。

这两种策略都旨在通过更多样化的任务和学习场景来丰富其训练，从而减轻基础模型的局限性，提高其整体性能和适应性。

3 结语

缩放法则是一种有效的AI训练方法，但它也有其局限性和挑战。在实际应用中，我们需要考虑多种因素，如部署成本、任务多样性、数据质量和数量、人类反馈等。为了提高模型的泛化能力和适应性，我们需要探索不同的数据处理和训练策略，如调整任务频率、少样本学习等。这些策略可以帮助我们在不同规模的模型上实现更好的对齐效果，从而提升AI的性能和价值。

[参考文献]

[1]EduardoG Altmann,Giampaolo Cristadoro,and Mirko DegliEsposti. On the origin of long range correlations in texts. Proceedings of the National Academy of Sciences,109(29):11582−11587,2012.25.

[2]Gehman,S.Gururangan,M.Sap,Y.Choi,and N.A.Smith. RealToxicityPrompts:Evaluatingneural toxic degeneration in language models. In Findings of the Association for Computational Linguistics:EMNLP 2020,pages3356−3369,Online,Nov.2020.Association for Computational Linguistics.doi:10.18653/v1/2020.findings−emnlp.301.URL https://aclanthology.org/2.

[3]Jared Kaplan, Sam McCandlish, Tom Henighan,Tom B Brown,Benjamin Chess, Rewon Child,Scott Gray, Alec Radford,Jeffrey Wu,and Dario Amodei.“Scaling laws for neural language models”.In:arXiv preprint arXiv:2001.08361 (2020).020.findings−emnlp.301.