

2
0
2
3
2
2
7

LLaMA：开放高效的基础语言模型

Hugo Touvron*, Thibaut Lavril*, Gautier Izacard*, Xavier Martinet Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Naman Goyal Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin Edouard Grave*, Guillaume Lample*...

元人工智能

摘要

我们介绍了 LLaMA，这是一个参数从 7B 到 65B 不等的基础语言模型集合。我们在数万亿个词库上训练我们的模型，并证明完全可以使用公开可用的数据集来训练最先进的模型，而无需求助于无法访问的专有数据集。特别是，LLaMA-13B 在大多数基准测试中都优于 GPT-3 (175B)，而 LLaMA-65B 与最好的模型 Chinchilla-70B 和 PaLM-540B 相比也具有竞争力。我们向研究界发布了所有模型¹。

如果一个较小的模型训练时间较长，其推理性能最终会更低。例如，尽管霍夫曼等人（2022 年）建议在 200B 标记上训练一个 10B 模型，但我们发现，即使在 1T 标记之后，7B 模型的性能也会继续提高。

这项工作的重点是训练一系列语言模型，通过训练比通常情况下更多的词块，在各种推理预算下达到最佳性能。由此产生的模型被称为 LLaMA，参数范围从 7B 到 65B 不等，与现有的最佳 LLM

相比，性能极具竞争力。例如，LLaMA-13B 在大多数基准测试中都优于 GPT-3，尽管其体积小了 10 倍。我们相信，由于该模型可以在单个 GPU 上运行，它将有助于实现 LLM 访问和研究的民主化。在更高端的规模上，我们的 65B 参数模型也能与 Chinchilla 或 PaLM-540B 等最好的大型 LLM 模型相媲美。

1 引言

在大量文本语料库中训练的大型语言模型 (LLMs) 已经证明，它们有能力根据文本指令或少量示例完成新任务 (Brown 等人，2020 年)。当模型扩展到足够大的规模时，这些“少数几个例子”的特性首次出现 (Kaplan 等人，2020 年)，由此产生了一系列侧重于进一步扩展这些模型的工作 (Chowdhery 等人，2022 年；Rae 等人，2021 年)。这些工作基于参数越多性能越好的假设。然而，霍夫曼等人 (2022 年) 的最新研究表明，在给定的计算预算下，最佳性能不是由最大的模型实现的，而是由在更多数据上训练的较小模型实现的。霍夫曼等人 (2022 年) 提出的缩放定律的目的是确定如何在特定的训练计算预算下最佳地缩放数据集和模型大小。然而，这一目标忽略了推理预算，而推理预算在大规模服务于语言模型时变得至关重要。在这种情况下，给定一个目标性能水平，首选的模型不是训练速度最快的，而是推理速度最快的。通讯 (htouvron, thibautlav, gizacard, egrave, glample)@meta.com
¹<https://github.com/facebookresearch/llama>

与 Chinchilla、PaLM 或 GPT-3

不同的是，我们只使用公开可用的数据，这使得我们的工作符合开源原则，而大多数现有模型所依赖的数据要么不是公开可用的，要么没有记录（例如“书籍 - 2TB”或“社交媒体对话”）。也有一些例外，特别是 OPT (Zhang 等人，2022 年)、GPT-NeoX (Black 等人，2022 年)、BLOOM (Scao 等人，2022 年) 和 GLM (Zeng 等人，2022 年)，但都无法与 PaLM-62B 或 Chinchilla 竞争。在本文的其余部分，我们将概述我们对变压器架构 (Vaswani 等人，2017 年) 所做的修改，以及我们的训练方法。然后，我们报告了我们的模型的性能，并在一组标准基准上与其他 LLM 进行了比较。最后，我们利用负责任人工智能社区的一些最新基准，揭示了我们的模型中存在的一些偏差和毒性。

2 方法

我们的训练方法与之前工作 (Brown 等人, 2020 年; Chowdhery 等人, 2022 年) 中描述的方法类似，并受到钦奇拉缩放定律 (Hoffmann 等人, 2022 年) 的启发。我们使用标准优化器在大量文本数据上训练大型转换器。

2.1 预训练数据

我们的训练数据集由表 1

所列的多个数据源混合而成，涵盖了各种不同的操作。在大多数情况下，我们重复使用已被用于训练其他 LLM 的数据源，但仅限于使用公开可用的数据，并与开源兼容。这就产生了以下混合数据及其在训练集中所占的比例：

英语 CommonCrawl [67%]。我们使用 CCNet 管道 (Wenzek 等人, 2020 年) 预处理了从 2017 年到 2020 年的五次 CommonCrawl

转储。该流程在行级对数据进行重复，使用 fastText 线性分类器进行语言识别，删除非英语页面，并使用 n 克语言模型过滤低质量内容。此外，我们还训练了一个线性模型，将维基百科中用作参考的页面与随机抽样的页面进行分类，并丢弃未被归类为参考的页面。

C4 [15%]。在探索性实验中，我们发现使用不同的预处理 Com- monCrawl

数据集可以提高性能。因此，我们在数据中加入了公开的 C4 数据集 (Raffel 等人, 2020 年)：与 CCNet 的主要区别在于质量过滤，它主要依赖于标点符号的存在或网页中单词和句子的数量等特征。GitHub 的预处理也包含重复数据删除和语言识别步骤以与 CCNet

Apache、BSD 和 MIT 许可发布的项目。此外，我们还根据行的长度或字母数字字符的比例使用启发式过滤低质量文件，并使用带掩码的快照删除垃圾标题之后的数据。最后，我们在文件级别对重叠数据集的权重，以进行精确匹配。维基百科 [4.5%]。我们添加了 2022 年 8 月至 8 月期间的维基百科转储，涵盖 20

采样道具年代 磁盘大小

	采样比例	年代	磁盘大小
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

表 1：预训练数据。表

1：用于预训练的数据混合物，我们列出了每个子集的采样比例、在 1.4T 词组上训练时在子集上执行的历时数以及磁盘大小。在 1T 标记上的预训练运行具有相同的采样比例。

使用拉丁字母或西里尔字母的语言：bg、ca、c s、da、de、en、es、fr、hr、hu、it、nl、pl、pt、ro、ru、sl、sr、sv、uk。我们在处理数据时会删除超链接、注释和其他格式化模板。

古腾堡和 Books3

[4.5%]。我们的训练数据集包括两个图书语料库：古腾堡项目 (Gutenberg Project)，其中包含公共领域的图书；以及 TheP- ile 的 Books3 部分 (Gao 等人, 2020 年)，这是一个用于训练大型语言模型的公开数据集。我们在图书层面进行重复数据删除，删除内容重叠率超过 90% 的图书。

ArXiv [2.5%]。我们处理 arXiv Latex

文件，为数据集添加科学数据。按照 Lewkowycz 等人 (2022 年) 的做法，我们删除了第一节之前的所有内容以及参考书目。我们还删除了 .tex 文件中的注释，并对用户编写的定义和宏进行了内联扩展，以提高论文间的一致性。

Stack Exchange [2%]。我们收录了 Stack Exchange 的转储数据，这是一个提供高质量问题和答案的网站，涵盖了从计算机科学到化学等各种领域。我们保留了 28 个最大网站的数据，重新移动了文本中的 HTML 标记，并按得分（从高到低）对答案进行了排序。

标记化器。我们使用字节对编码 (BPE) 算法 (Sennrich 等人, 2015 年) 对数据进行标记，使用的是 Sentence- Piece 的实现 (Kudo 和 Richardson, 2018 年)。值得注意的是，我们将所有数字拆分为单个数字，并回退到字节来分解未知的 UTF-8 字符。