01 欢迎来到这场大模型竞赛

我们正迎来大模型井喷的时代，深度学习模型的创新和突破层出不穷。随着GPT、Stable Diffusion、Sora等模型的问世，大模型已经在文本、图片和视频生成领域展示了其强大的能力。读者可能会好奇，训练这些庞大模型的过程究竟是什么样的，需要多少资源和时间？未来，普通人是否也有机会训练出属于自己的私有大模型呢？

在尝试训练一个大模型时，我们通常会遇到两个主要挑战：

- 这个模型能否在现有硬件环境中运行？
- 需要多长时间才能完成一个数据集的训练？

这两个问题的核心也正是大模型的规模定律（Scaling Law[1]）中提到的对模型表现有重大影响的两个要素：模型规模和数据规模。

1 https://arxiv.org/pdf/2001.08361

1.1 模型规模带来的挑战

我们首先从模型规模说起。很多读者刚入门深度学习时，可能是从ResNet、Google-Inception等经典模型开始学习的。然而，工业界现有模型的规模与这些入门级模型之间存在数量级的差距。短短几年间，“大模型”的代表已经从BERT-Large的0.3B（3亿）参数量迅速发展到GPT-2的1.5B，甚至GPT-3的175B。

我们以GPT-3为例简单估算其模型参数和优化器需要占用的显存大小。假设模型使用单精度浮点数存储[1]，每个参数占用4字节，模型参数需要占用700GB的显存；而Adam优化器的显存占用是参数的两倍，也就是说至少需要2100GB以上的显存才能容纳GPT-3模型和优化器。这甚至还未包括训练过程中动态分配的显存。

目前单张NVIDIA GPU最大的显存容量为80GB（如A100、H100等），而仅GPT-3模型和优化器所需的2100GB显存就已远远超过单卡的显存极限。这意味着，为了运行GPT-3，除了进行显存优化外，还必须采用分布式系统，让多个GPU节点共同承担庞大的显存需求。因此，显存成为模型训练的硬性门槛。在实际应用中我们通常会先优化显存占用，再优化速度。

事实上，现行工业界最大模型的规模早已超越了GPT-3，使得显存优化和分布式训练技术的重要性愈发突出。以语言模型为例，从图1-1中可以看出，近年来其模型规模呈现数量级的增长，过去五年间已经翻了近百倍。

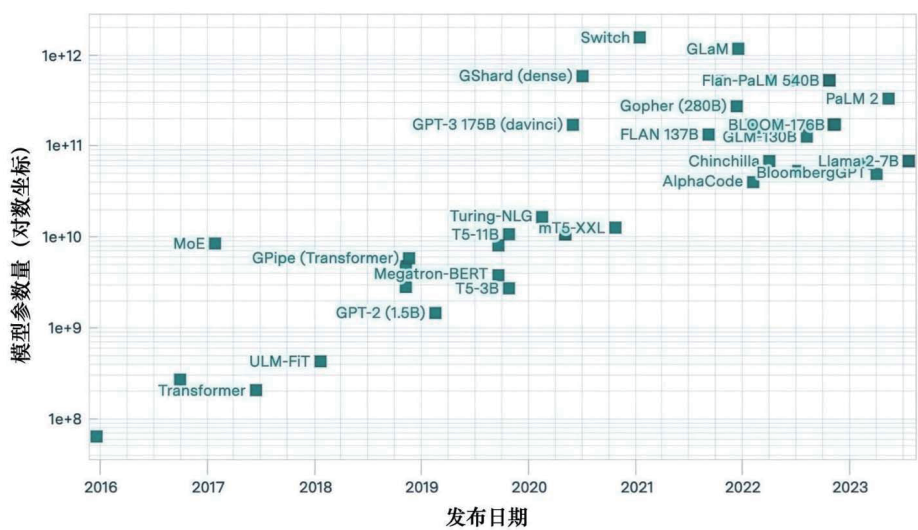

图1-1　以大模型为例展示模型参数规模的增长趋势。本图基于Epoch AI Database[2]于2024年5月的数据绘制，选取了语言模型领域引用次数超过100的部分模型。

1　该假设仅作为示例，实际训练中可能使用低精度的数据类型。

2　https://epochai.org/data/epochdb

1.2 数据规模带来的挑战

如果说模型规模的增长带来的是不断增长的显存占用，那么数据规模的增长带来的则是越来越长的训练时间。不同于入门级的MNIST、COCO、ImageNet这些最多百万样本量的数据集，工业界现行的数据集如Laion-5B、Common Crawl等已经达到10B甚至100B规模了。表1-1以图像领域为例展示了数据集样本数量快速增长的过程。读者可能并不理解100B数据意味着什么，那么不妨通过训练时间来建立一些直观的认识。按照估算[1]，GPT-3如果在100B个token[2]的Common Crawl上训练，如果只用一张V100计算卡训练，需要355年才能跑完全部的数据集——也就是要从康熙年代开始训练，才能赶上今年发布。为了在合理的时间内完成训练，我们不仅需要进行单卡性能的极致优化，还需要借助分布式训练系统进行并行处理，以加速模型的训练过程。

表1-1　图像领域常用数据集的数据规模

数据集	MNIST	Coco 2017	ImageNet2012	Laion-400M	Laion-5B
样本数量	~60K 图片	~118K 图片	~12M 图片	~400M 图片文本对	~5B 图片文本对

尽管当前的数据集规模已经相当庞大，但其增长速度依然惊人。今天的100B大数据集，可能在几年后就会变成中等规模的数据集。正如图1-2所示，近年来深度学习模型训练使用的数据规模呈现指数级增长态势。

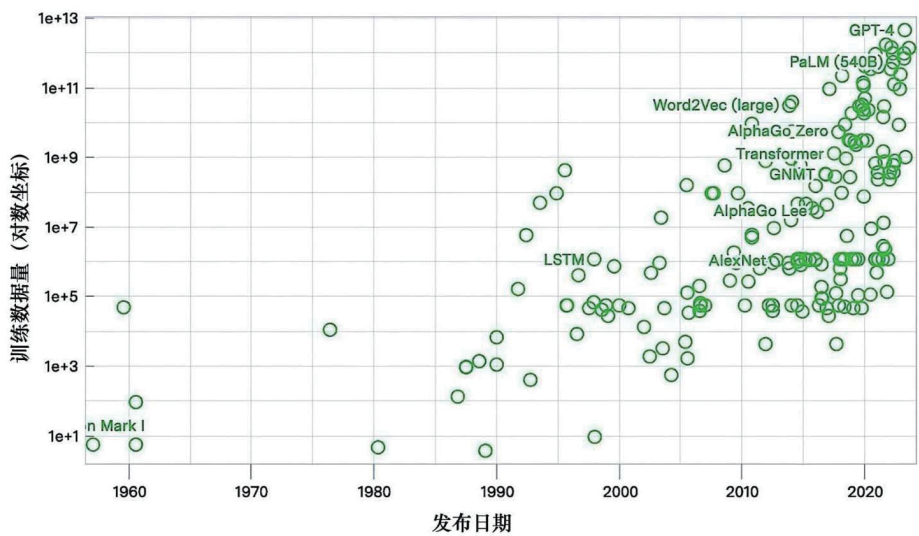

图1-2　训练用数据量的增长趋势。本图基于Epoch AI Database[3]于2024年5月的数据绘制，展示了所有领域模型训练使用的数据量并启用了"Show outstanding systems"标注。

1　https://lambdalabs.com/blog/demystifying-gpt-3

2　模型处理数据的基本单位

3　https://epochai.org/data/epochdb

模型规模与数据规模的双重增长，最终都会反映到训练模型所需的成本和训练所需时间上。以Mosaic ML在2022年底发布的GPT系列模型的训练成本估算为例[1]，如图1-3所示，随着模型参数量的增加，训练时间和成本呈指数级增长。这种增长速度凸显了大规模模型训练在资源和时间上的巨大挑战。

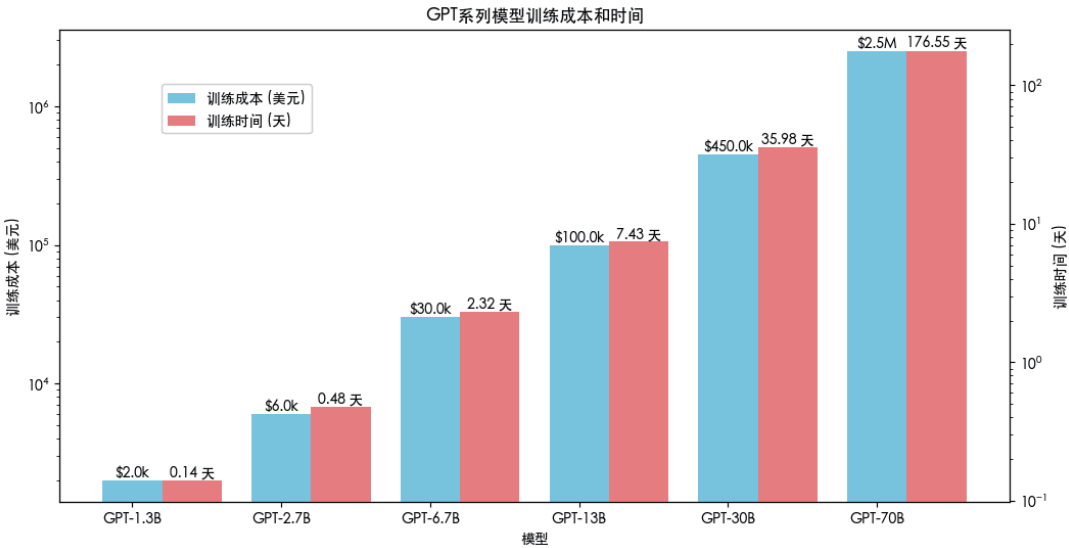

图1-3　以GPT系列模型为例，展示训练成本和时间随着模型增长的趋势，数据来源于MosaicML的博客。

1.3 模型规模与数据增长的应对方法

模型规模和数据规模的增长最直观的影响就反应在训练的成本上，因为我们需要更多的机器且需要训练更长的时间，这也使得显存优化和性能优化显得尤为重要。显存优化可以降低模型的训练门槛，用更少的GPU实现相同规模的训练；而性能优化则可以用更短的时间完成模型的训练，这也变相降低了训练的成本。

然而显存优化和性能优化并不是简单的工作。模型的完整训练过程包括若干不同的训练阶段，必须搞清楚不同阶段的算力需求和特点，才能进行针对性的优化。

先来看一下完整的模型训练过程都包含哪些阶段。整体来说，对于最常见的训练过程，每一轮训练循环都包括5个串行的阶段，分别是**数据加载、数据预处理、前向传播、反向传播、梯度更新**。在此基础上，如果有多张GPU卡甚至多台训练机器可供使用，

1　https://www.databricks.com/blog/gpt-3-quality-for-500k

还可以把每个GPU看作一个节点，进一步将计算任务分散到不同节点形成分布式训练系统。分布式训练也因此会多出一个额外的阶段，也就是**节点通信**，如图1-4所示。

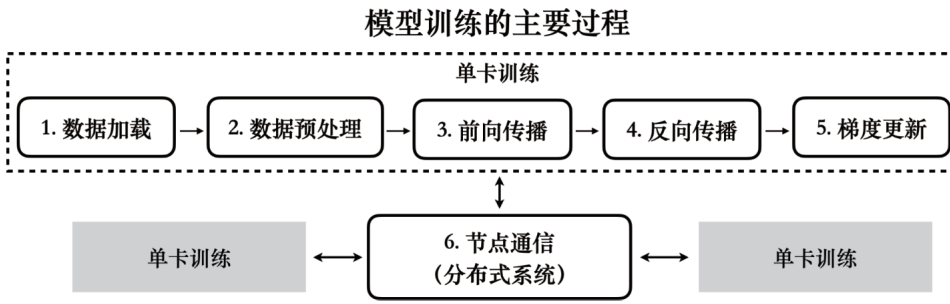

图1-4　模型训练流程示意图

讨论一下，这些训练阶段各自的优化重点是什么。首先，**数据加载**是指将训练数据从硬盘读取到内存的过程。为了避免训练程序停下来等待硬盘读取数据，可以通过将数据加载与模型计算任务重叠来进行优化，例如使用预加载技术。这部分内容将在第5章：数据的加载和处理中详细讨论。

接下来是**数据预处理**，即在CPU上对加载到内存中的数据进行简单处理，以满足模型对输入数据的要求。为了避免数据预处理成为训练的瓶颈，可以使用离线预处理技术或优化CPU预处理代码的效率。这部分内容将在第5章：数据的加载和处理以及第6章：单卡性能优化中讨论。

前向传播是模型训练的前向计算过程，以计算损失函数（Loss）为终点；**反向传播**则是梯度计算过程；**参数更新**则是根据梯度方向对模型参数进行更新。这三个阶段的计算主要依赖于GPU设备。GPU设备的峰值浮点运算能力（peak FLOPS）和显存容量是单卡训练规模和速度的主要限制因素，也是优化的重点。因此，在第6章和第7章中，将分别介绍一些通用的单卡性能优化和显存优化技巧。

然而，单卡的计算能力和显存容量存在较大限制。如果需要进一步提升模型规模或加快训练速度，可以将计算和显存分配到多张GPU上，通过**节点通信**来协同完成更大规模的训练。在第8章中，将详细讨论不同的分布式训练策略及其对节点通信的优化方法。

除了常规的性能优化和显存优化技术外，我们特别准备了第9章：高级性能优化技术。这一章将深入探讨一系列“高投入、高风险、高回报”的优化方法。这些高级技巧有望显著提升GPU的计算效率，但其原理较为复杂、调试相对耗时，因此更适合对训练的性能优化有较高要求的读者。

在第10章：GPT-2优化全流程中，将结合实战，将本书介绍的大部分性能和显存优化技巧串联起来。通过实际案例探索不同优化技巧的应用方法和实际效果。