南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

高效大模型策略

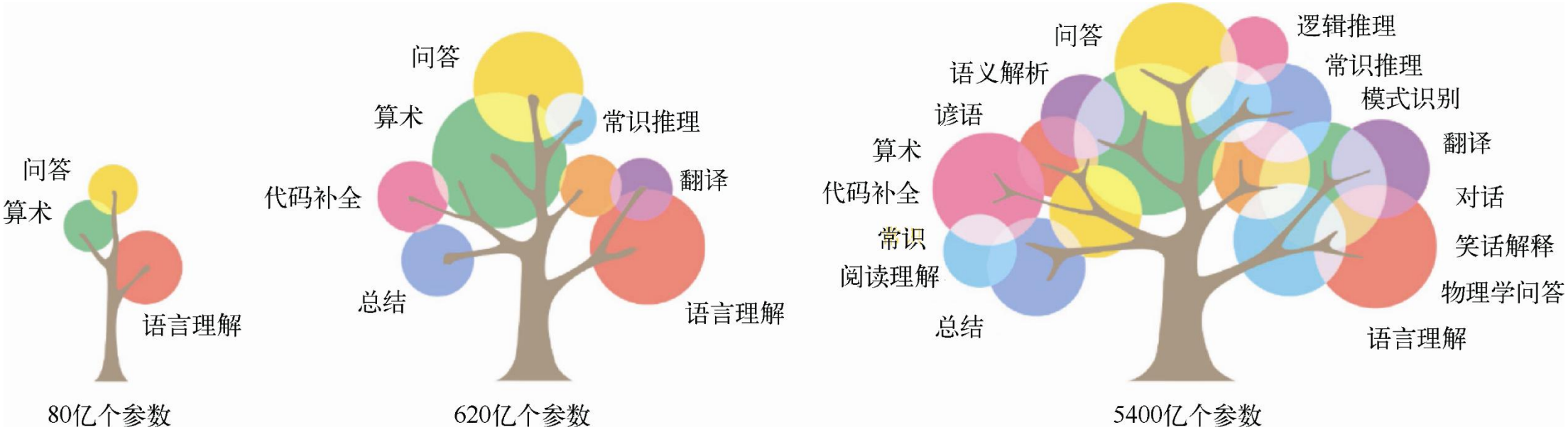
魏明强、宫丽娜
计算机科学与技术学院

目录

目录

大模型效率面临的问题

- 由于更大的参数规模需要更高的计算成本和内存需求，大模型的训练和微调会受到严重限制
- 训练这些模型需要大量的数据和资源，给数据获取、资源分配和模型设计带来挑战，探索不同架构或策略的成本变得过高
- 大规模参数使大模型不适合部署在资源受限的环境中，如边缘设备



随着模型参数规模的增大，大模型不仅提高了现有任务的性能，而且还出现了很多新功能

目录

大模型效率及其评估指标

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- 本章将“大模型效率”定义为大模型产生特定性能时所需的资源，与性能成正相关，与资源成负相关

- 高效大模型策略旨在不影响模型性能的情况下优化计算和内存资源，这些评估指标将是高效大模型策略的重要依据和体现

- **评估大模型效率的关键指标**
  - 参数数量
  - 模型大小
  - 浮点运算次数
  - 推理时间/token生成速度
  - 内存占用
  - 碳排放

大模型效率及其评估指标

评估大模型效率的关键指标

- 参数数量

参数数量是直接影响模型学习能力和复杂性的关键因素。

这些参数包括权重和偏差等参数，在训练或微调阶段是可以学习的。

更大的参数数量通常使模型能够学习到更复杂的数据模式和新功能，但会影响训练和推理计算的时间。

- 模型大小

模型大小定义为存储整个模型所需的磁盘空间，通常以千兆字节（GB）或兆字节（MB）等为单位。

模型大小会受到多个因素的影响，其中最主要的因素是参数数量，其他因素有参数数据类型和特定的体系结构。

模型大小会直接影响存储需求，提前考虑模型大小对在存储受限环境下的部署尤其重要。

- 浮点运算次数

浮点运算次数是指单次前向传播过程中浮点运算（如加减乘除法）的次数（计算量），用于估算大模型的计算复杂度。

较高的浮点运算次数通常意味着模型有着更高的计算要求，在资源有限的环境中部署这种模型将是一个挑战。

系统的并行优化以及不同的架构也都会影响最终的整体计算效率。

大模型效率及其评估指标

评估大模型效率的关键指标

- 推理时间/token生成速度

推理时间也称为延迟，是大模型在推理阶段从输入到生成响应所需的时间，单位通常为毫秒或秒。

推理时间是在实际部署的设备上进行评估的，考虑了特定的硬件和优化条件，提供了现实世界性能的实用衡量标准。

token生成速度是指模型在每秒内可以处理的token数，它能够用来规范推理时间，是反映模型速度和效率的关键性能指标。

- 内存占用

内存占用是指在推理或训练期间加载和运行模型所需的随机存取存储器的内存大小，通常以MB或GB为单位。

内存占用的内容不仅包括模型参数，还包括其他运行时必需数据，如中间变量和数据结构。

较大的内存占用会限制模型的可部署性，尤其是在资源受限的环境中，需要优化技术来降低占用，如模型剪枝或量化。

- 碳排放

碳排放通常以模型从训练到推理的过程中排放的二氧化碳量来衡量，反映了训练和运行该模型对环境的影响。

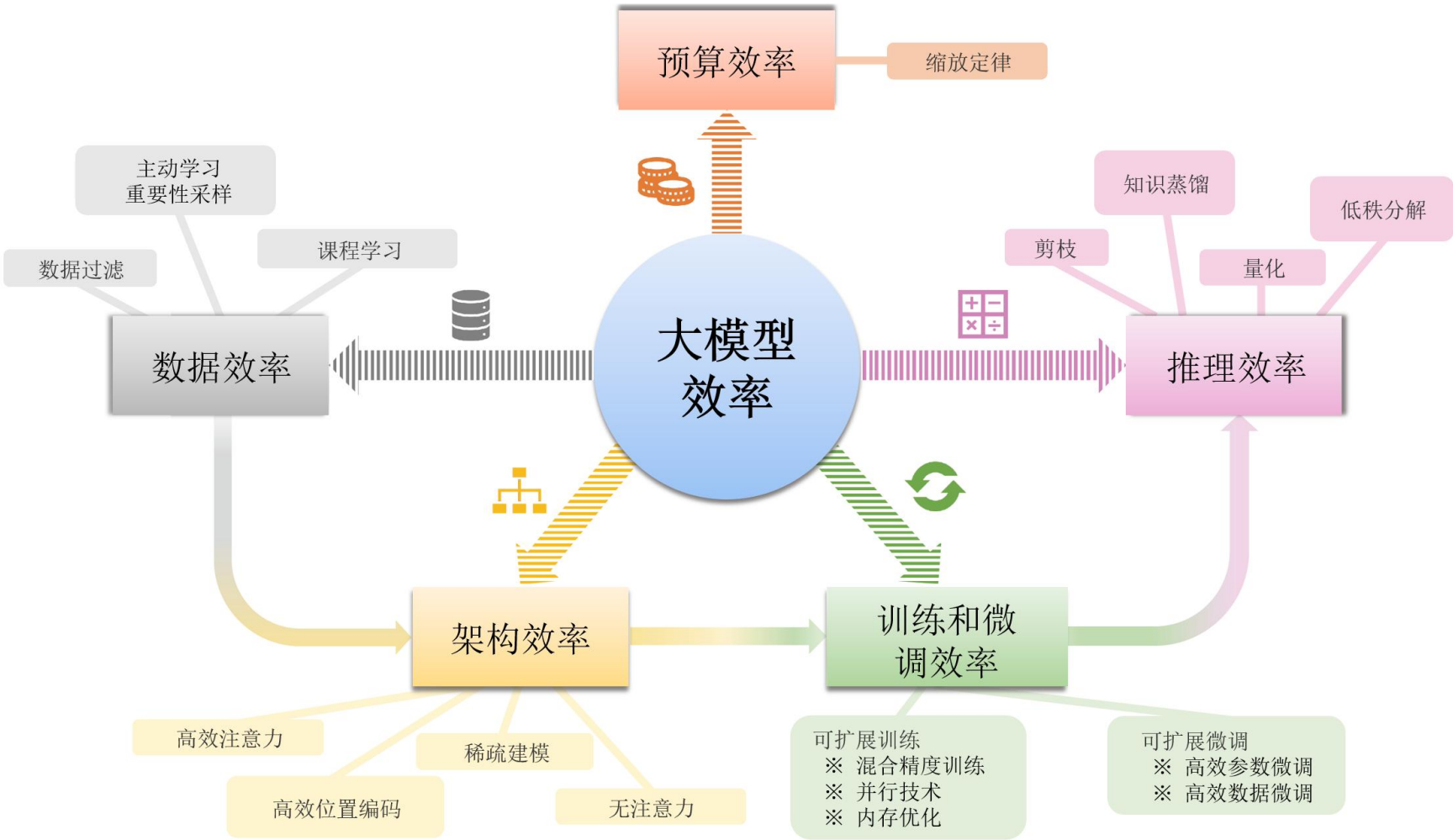
碳排放受到各种因素的影响，包括所用硬件的能源效率、电力来源，以及模型训练和运行的持续时间。

可以通过模型优化、硬件加速和算法改进等方式提高能效，还可以为数据中心（如苹果公司的云上贵州数据中心、腾讯的七星洞数据中心）选择更环保的能源，从而减少碳排放。

目录

高效大模型策略

- **提高大模型效率的关键方向**
  - 预算效率
  - 数据效率
  - 架构效率
  - 训练效率
  - 推理效率
  - 微调效率



目录

预算效率

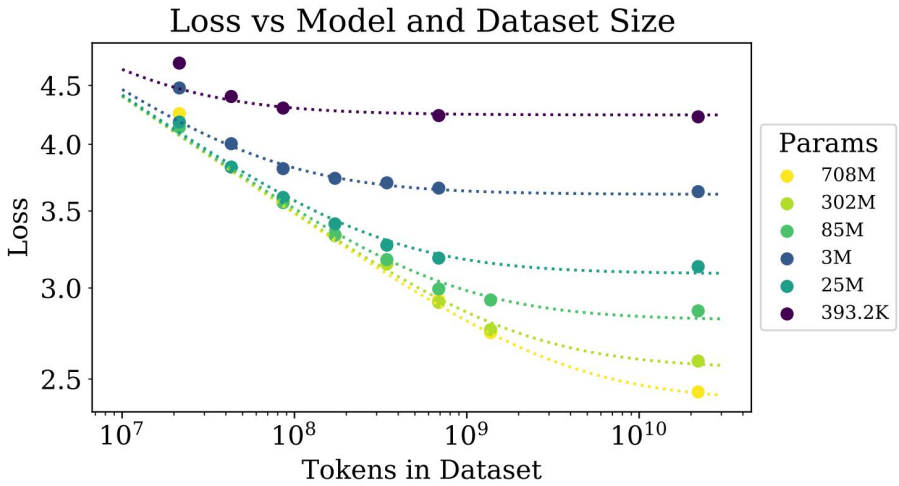
□ 大模型性能受到各种因素的影响，包括训练数据、模型大小、体系结构、计算资源和训练策略等。通过调整这些因素来达到满足预期的最佳性能，能够降低资源的消耗以提高效率。但如果采用试错方法会在调试中浪费大量的资源，而且通常无法找到最高效的设计方案。为此，可以通过提前预测大模型的性能来调整大模型的设计和资源安排。

缩放定律研究了在某些系统中，随着系统规模的增大，某些特定指标或行为会发生可预测的变化。通过缩放定律可以提前预测大模型性能，以便更有效地规划和分配资源，进而提高大模型效率。

- 缩放定律表明，大模型性能（假设目标函数为L）主要取决于三个因素：模型参数的数量N、数据集大小D及训练的计算预算C。

模型的性能会随着模型参数的数量N、数据集大小D和训练的计算预算C的增加而持续增加。当任意两个因素不受瓶颈限制时，模型的性能与第三个因素之间存在幂律关系。

但如果固定模型参数的数量N或数据集大小D中的一个，而增加另一个，模型的性能的提升幅度会因受到惩罚而有所减少。



目录

数据效率

□ **据效率策略从数据利用方面提高大模型效率。大模型对数据的需求是无止境的，但海量的数据给大模型训练不仅延长了训练时间，而且由于耗电大、存储容量大而导致训练成本急剧上升。有效的数据使用方法对大模型的训练和验证都至关重要，能够在降低资源消耗的同时提升模型性能，从而提高大模型效率。**

- 数据过滤

-将训练重点指向信息量更大的样本，较少集中在信息价值较小的样本上。

-通过重复数据消除、数据下采样减少训练集中的冗余，提高数据质量。

- 主动学习/重要性采样

-用较少的标注来训练样本，实现较好或等效的性能。

-根据样本对学习过程的重要性对样本进行优先级排序，仅选择和标注最有用的样本，有策略地减少训练样本总数的目的。

- 课程学习

-通过仔细设计训练数据中样本的反馈顺序来提高模型训练效率的策略。

-先从简单的样本或子任务开始训练，并逐步升级到具有挑战性的任务上。

目录

架构效率

□ Transformer架构中的关键操作注意力机制，通常需要相对于序列长度的二次复杂度来进行计算，因此在处理长输入序列时速度明显较慢。因此，减少注意力操作所需的计算成为提高体系结构效率的直接解决方案，对训练和推理阶段都有效。

- 高效注意力

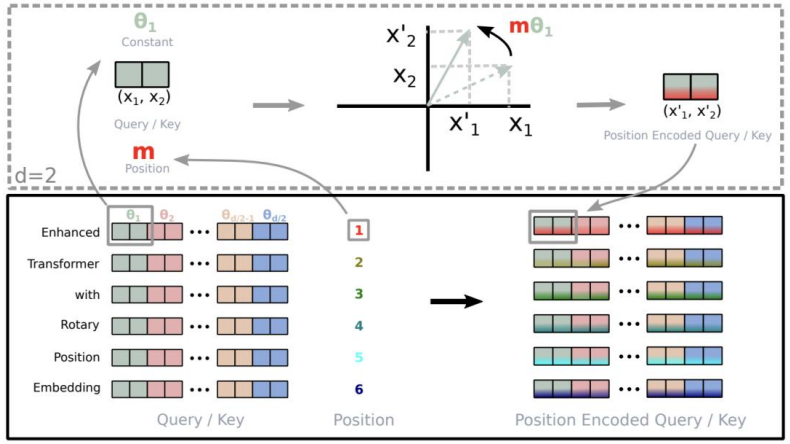
-并非所有token关系都具有相同的重要性，可以识别并仅保留最关键的关系。

-在硬件层面减少硬件之间的数据传输。

- 高效位置编码

-相对位置编码方法利用两个token之间的相对位置，而非单个token的绝对位置。

-旋转位置编码使用旋转矩阵对输入序列进行编码。



旋转位置编码

- 稀疏模型

-计算时只将大模型中用于给定的任务、样本或token的某些部分被激活。

- 无注意力模型

-用其他模块取代注意力机制，在性能上已经能够实现与标准的Transformer相当的效果。

目录

训练效率

□ 大模型数据和模型的规模会直接影响到模型的训练效率。因此，训练效率是决定大模型效率的重要因素，提高训练效率需要解决由大模型数据和模型规模的增加带来的问题。

- 稳定训练策略

-调整超参数如批处理大小、学习率等，实行权重衰减和梯度裁剪等稳定策略，防止梯度消失或爆炸。

- 混合精度训练

-用更低精度的存储格式存储参数，减少内存使用，加速模型内的通信过程

-使用FP32存储权重，使用FP16进行传递和计算

- 并行训练技术

-数据并行将数据集划分在多个加速器上被并行处理。

-模型并行将模型本身划分到多个加速器上。

- 内存优化

-将模型参数、梯度和优化器状态等元素划分到不同GPU上，可以根据需要从其他GPU中检索所需数据。

目录

推理效率

□ **模型巨大的参数数量给在云服务和资源有限设备上的部署带来了挑战，同时使得维持推理的成本很高。因此，加速推理的常见的方法是构建压缩模型，以达到与完整模型相当的性能。这种方法通常可以分为4类：剪枝、知识蒸馏、量化和低秩分解**

- 剪枝

-剪枝技术旨在识别大模型中的冗余并置零，可对单个参数进行非结构化剪枝，也可对结构单元进行结构化剪枝。

- 知识蒸馏

-通过利用一个大模型（教师模型）的知识训练一个小模型（学生模型）。

- 量化

-降低模型参数的数值精度，提高执行速度，降低模型大小。

-量化需要特定的硬件才能体现在低位精度上的优势。

-通常在模型训练完成后对参数进行量化。

- 低秩分解

-大模型权重矩阵存在于包括自注意力层和MLP层及嵌入层在内的线性层中，权重矩阵通常是低秩的表明模型权重中存在冗余，将权重矩阵分解为两个或更多个较小的矩阵以节约参数。

目录

微调效率

□ 在大规模且多样的数据集上训练的大模型已经具有出色的通用问题解决能力。通过有针对性的微调，它们在特定领域或任务中的性能可以得到显著提升。

- 参数高效微调

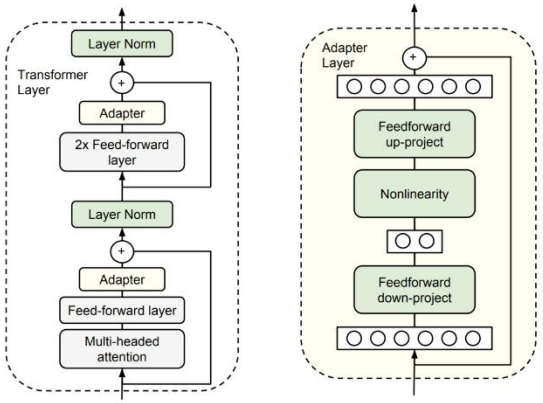
-仅选择预训练参数的一部分进行微调，其余参数保持不变

-用额外的小规模可学习块（适配器）来增强预训练模型，嵌入大模型的模块中。

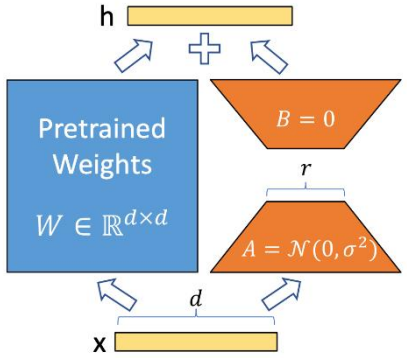
-使用低秩适配器（一组降维升维低秩矩阵）学习更详细的任务特定信息。

- 数据高效调整

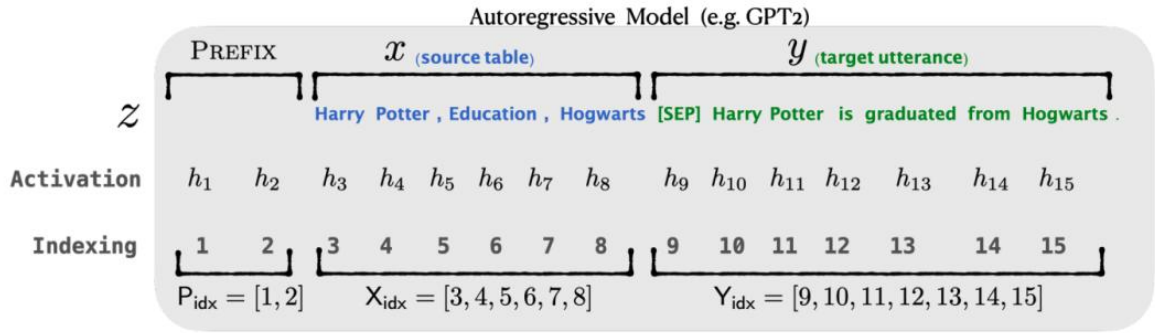
-在输入序列前添加一组有限的提示前缀序列。



适配器



低秩适配器



提示前缀序列

谢谢！
Thanks!