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

大模型微调-第7章

计算机科学与技术学院

目录

目录

大模型微调

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

- **大模型训练包括“预训练”和“微调”两个关键阶段**

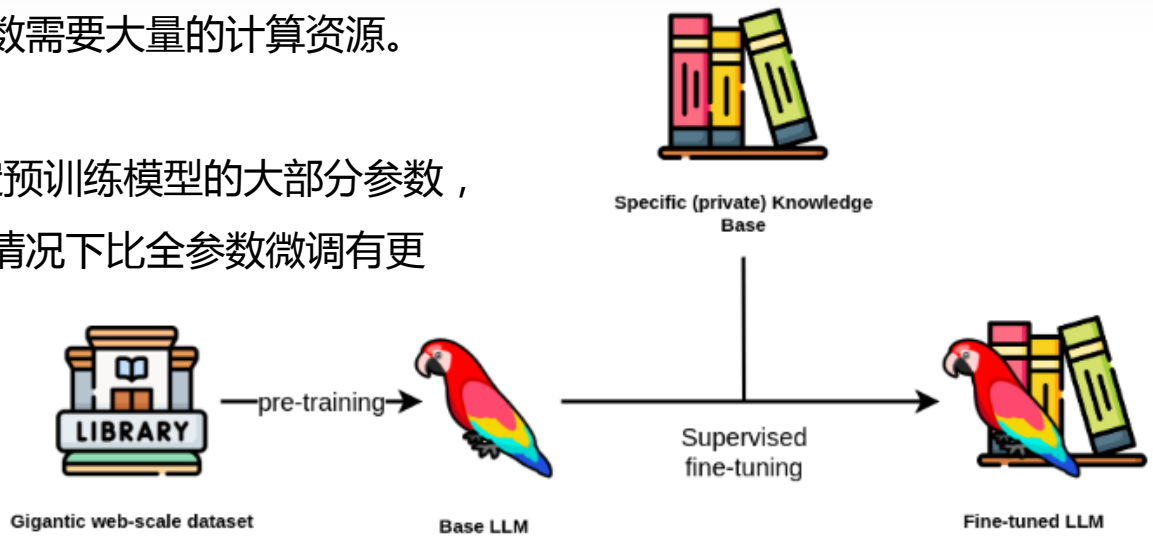
在预训练阶段，大模型通过在大量数据上进行训练学习，已经掌握了丰富的语言规则、知识信息以及视觉模式。然而，在大规模（公开）数据上通过自监督学习训练出来的模型虽然具有较好的“通识”能力（称为基础模型），却往往难以具备“专业认知”能力（称为专有模型/垂直模型）。

大模型的预训练成本非常昂贵，庞大的计算资源和数据让普通用户难以从头开始训练大模型。充分挖掘这些预训练大模型的潜力，针对特定任务的微调不可或缺。大模型微调是将预训练好的大模型参数作为起点，利用少量有标签的数据进一步调整大模型参数，以适应特定的任务，使得大模型不仅仅停留在理解通用知识的层面，更能够针对特定问题提供精准的解决方案。

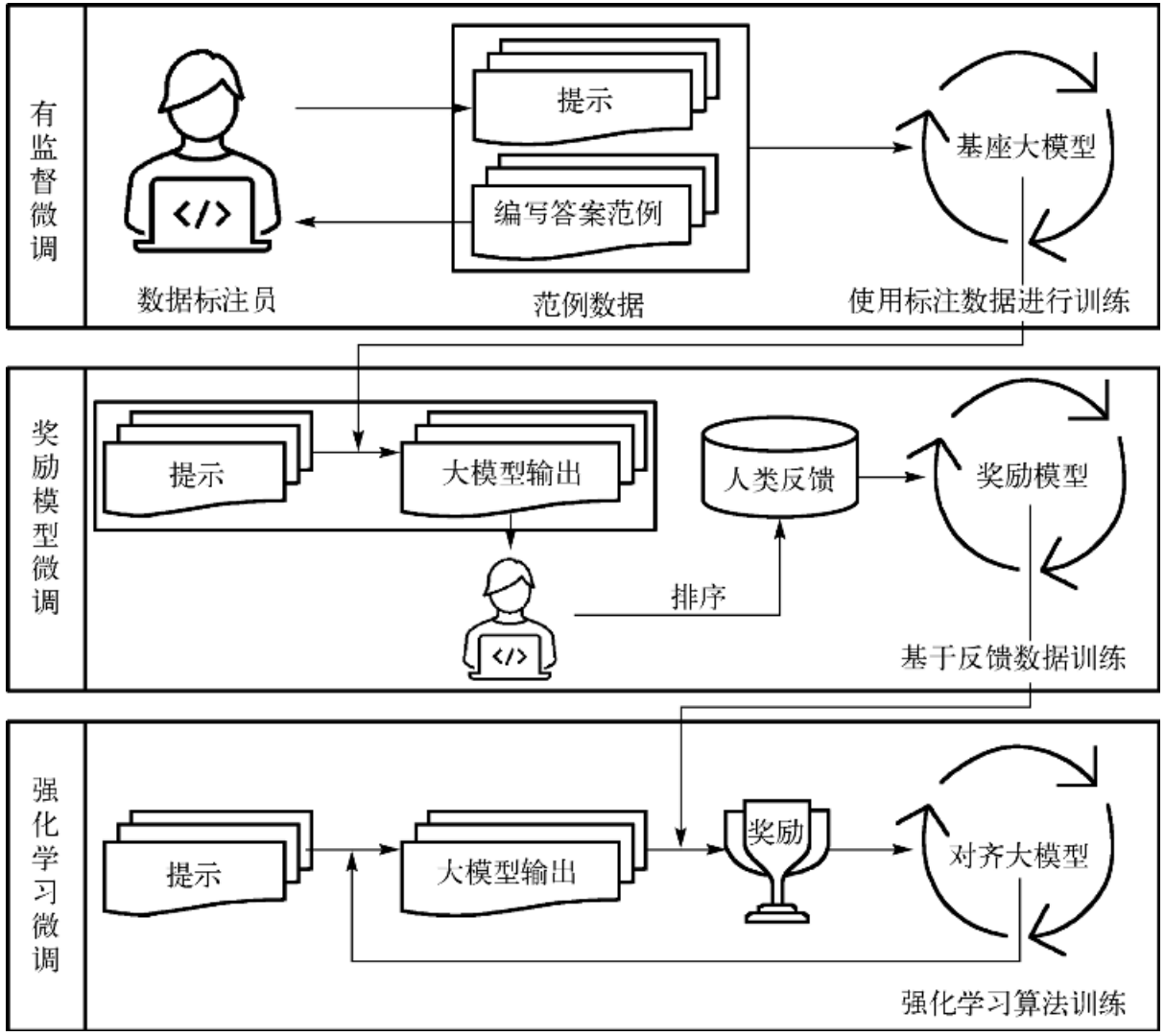
- **有监督微调分为：全参数微调和参数高效微调**

全参数微调指的是在特定任务上对整个预训练模型的所有参数进行更新。这种技术简单直接，可以使模型适应新的任务。但是随着模型参数规模变得越来越大，更新所有参数需要大量的计算资源。同时，当特定任务的数据量不足时，全参数微调容易导致过拟合。

参数高效微调（Parameter-Efficient Fine-Tuning，PEFT）是指固定预训练模型的大部分参数，仅微调少量或额外的模型参数来达到与全参数微调接近的效果，甚至在某些情况下比全参数微调有更好的效果，更好地泛化到域外场景。



大模型微调



- **指令微调**

过少量的、精心设计的指令数据来微调预训练后的大模型，使其具备遵循指令和进行多轮对话的能力，以提高其在处理命令式语言和指令性任务时的性能和适应性。

- **基于人类反馈的强化学习（Reinforcement Learning Human Forward，RLHF）微调：**

以人类的偏好作为奖励信号，通过强化学习与人类反馈相结合的方式，指导模型的学习和优化，从而增强模型对人类意图的理解和满足程度。主要包括：**奖励模型微调**和**强化学习微调**两个阶段。

奖励模型微调阶段通过学习人类对模型输出的评价（如喜好、正确性、逻辑性等）提供一个准确评价模型行为的标准。

强化学习微调阶段则基于奖励模型来指导优化模型的行为。通过这种方式，基于人类反馈的强化学习微调能够有效地将人类的智慧和偏好整合到模型训练过程中，提高模型在特定任务上的性能和可靠性。

目录

参数高效微调-增量式微调

参数高效微调

参数高效微调（PEFT）是在保持模型性能的同时，以最小的计算成本对模型进行微调，以适应特定任务或数据集的技术。

现有的参数高效微调可以大体分为**增量式微调**、**指定式微调**、**重参数化微调**三大类。

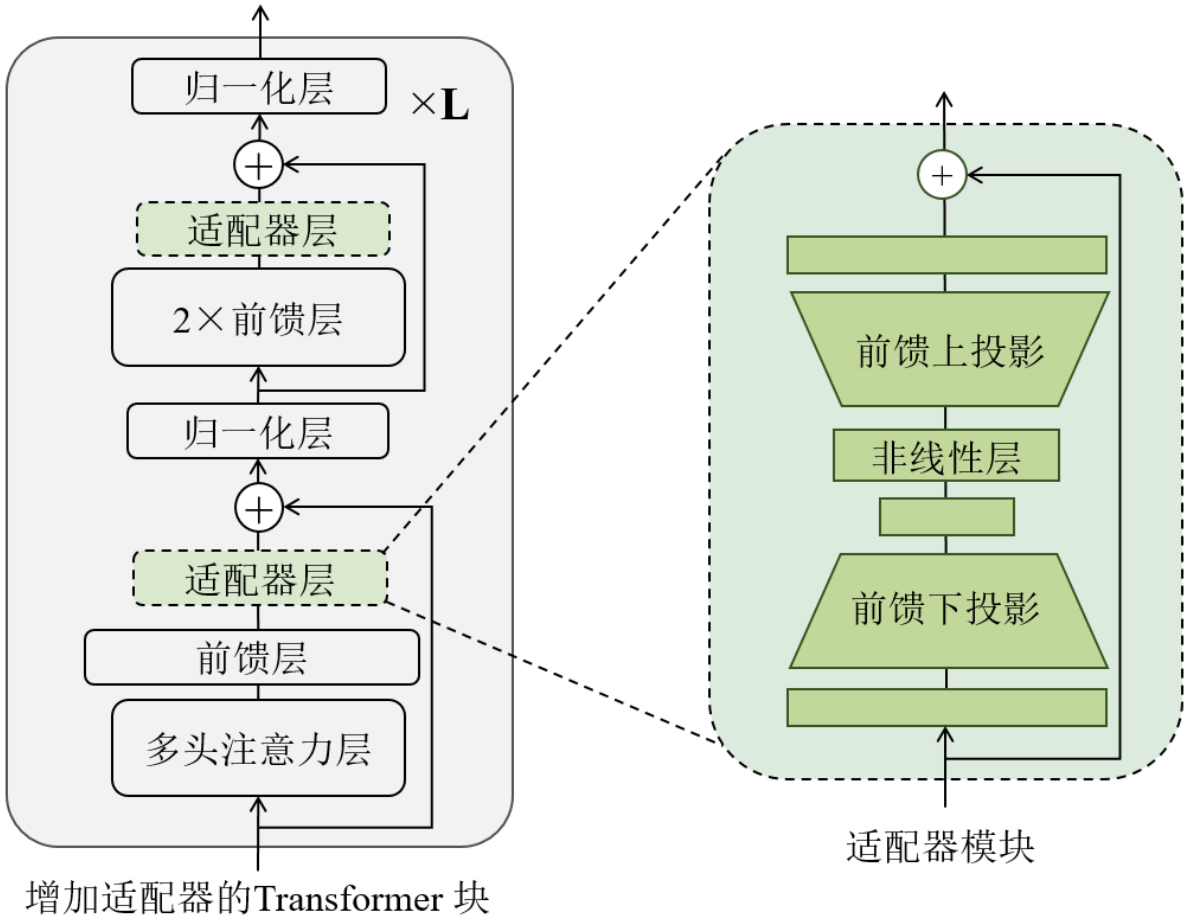
增量式微调

增量式（Addition-based）微调是在预训练模型基础上，仅仅调整少量添加的额外可训练的层或参数， 使模型能够快速地适应新任务或数据集的技术。根据添加的额外参数的位置或方式不同，增量式微调技术可以分为**适配器微调**和**前缀微调**。

适配器微调通常是指在预训练模型的中间层或特定层中插入额外的小型网络模块（适配器），进行特定任务的优化。

前缀微调指的是在模型的输入端添加一个连续的任务特定向量序列（称为前缀），这个向量序列与原始输入一起进入模型，在参数微调时模型能够“关注”这个前缀，从而引导模型生成更符合任务需求的输出。

参数高效微调-增量式微调 -适配器（Adapter）微调



加入适配器后的Transformer层主体架构以及适配器模块结构，微调时处理适配器的参数，其余参数均冻住

适配器微调

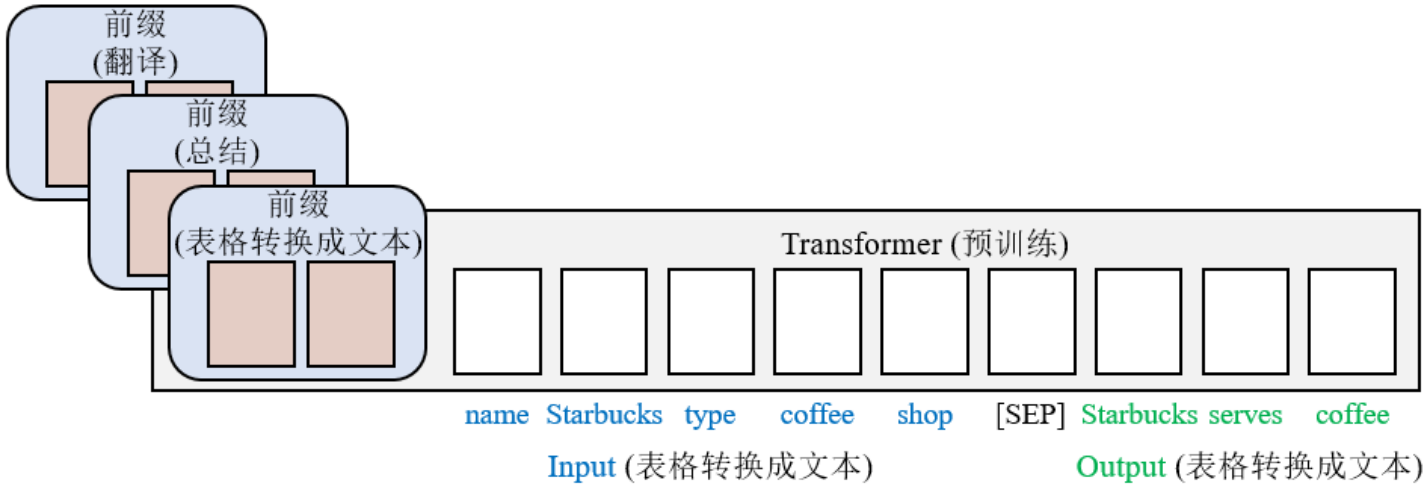
适配器微调（Adapter Tuning）是一种在预训练后的大模型中间层中，插入适配器（小型网络模块）来适应新任务的技术。在微调时将大模型主体冻结，仅训练特定于任务的参数，即适配器参数，减少训练时算力开销。以Transformer架构为例，如左图所示：

图解：

在多头注意力的投影和第二个前馈网络的输出之后分别插入适配器模块。其中，每个适配器模块主要由两个前馈（Feedforward）子层组成，第一个前馈子层以Transformer块的输出作为输入，将原始输入维度（高维特征）投影到（低维特征）。在两个前馈网络中，安插了一个非线性层。在输出阶段，通过第二个前馈子层还原输入维度，映射回原始维度，作为适配器的输出。

同时，通过一个跳跃连接将Adapter的输入重新加到最终的输出中，这样可以保证，即使适配器一开始的参数初始化接近0，适配器也由于跳跃连接的设置而接近于一个恒等映射，从而确保训练的有效性。

参数高效微调-增量式微调-前缀（Prefix）微调



- **前缀微调**

前缀微调（Prefix Tuning）在资源有限、任务多样化的场景下具有显著的优势。它是基于提示词前缀优化的微调技术，其原理是在输入token之前构造一段与任务相关的虚拟令牌作为前缀（Prefix），然后训练的时候只更新前缀的参数，而预训练模型中的其他参数固定不变。以Transformer架构为例，如上图所示：

- **图解：**

图中展示了使用前缀微调技术实现表格转换成文本（Table-to-Text）、总结（Summarization）和翻译（Translation）这三个下游任务。以表格转换成文本任务为例，输入任务是一个线性化的表格“name: Starbucks | type: coffee shop”，输出是一个文本描述“Starbucks serves coffee.”。在输入序列之前添加了一系列连续的特定任务向量表示的前缀参与注意力计算。

前缀微调能够有效地训练上游前缀以指导下游语言模型，实现单个基础模型同时支持多种任务的目标。前缀微调适用于涉及不同用户个性化上下文的任务中。通过为每个用户单独训练的前缀，能够避免数据交叉污染问题，从而更好地满足个性化需求。

参数高效微调-增量式微调-前缀（Prefix）微调

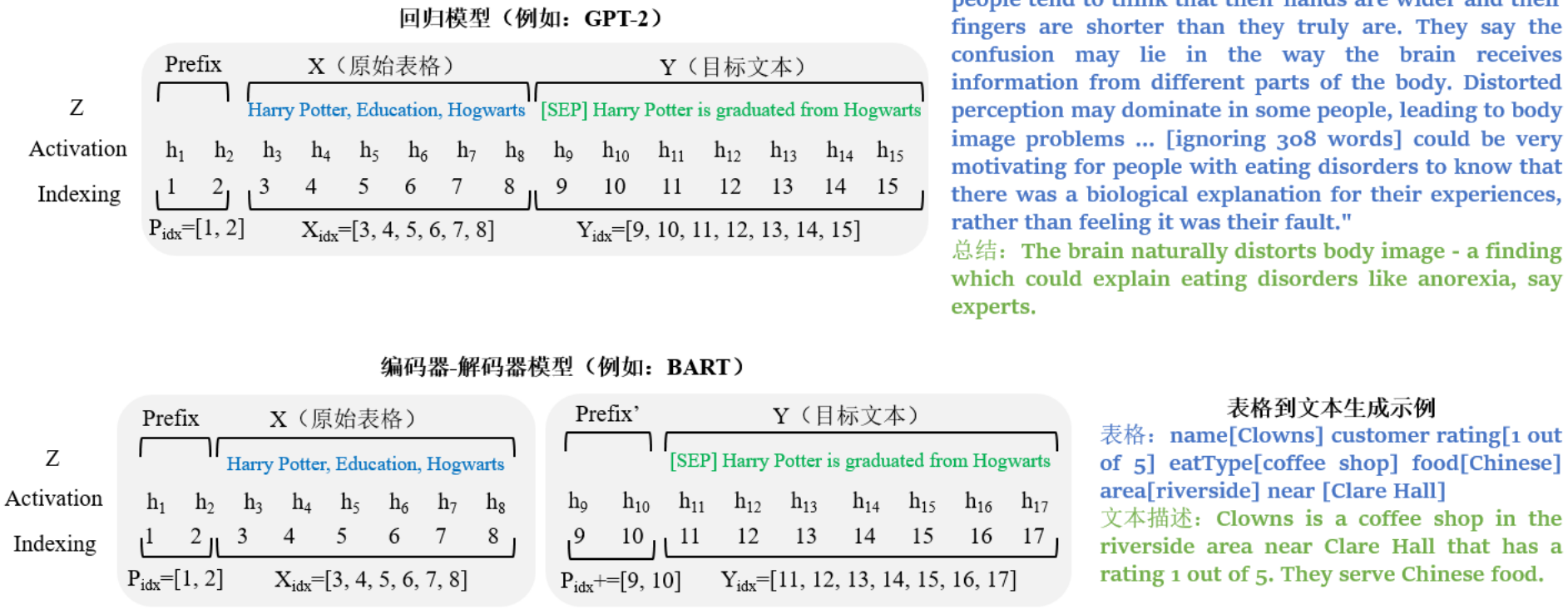
针对不同的模型结构，前缀微调需要构建不同的前缀，如下图所示：

- **回归架构模型：**

在输入之前添加前缀，得到z = [PREFIX; x; y]，合适的上文能够在固定预训练模型的情况下引导生成下文，如GPT-3的上下文学习。

- **编码器-解码器架构模型：**

编码器和解码器都需要增加前缀，得到z = [PREFIX; x; PRE FIX0; y]。编码器端增加前缀用来引导输入部分的编码，解码器端增加前缀用来引导后续 token 的生成。



回归架构模型和编码器-解码器架构模型构造前缀的方式对比示意图

目录

参数高效微调-指定式微调

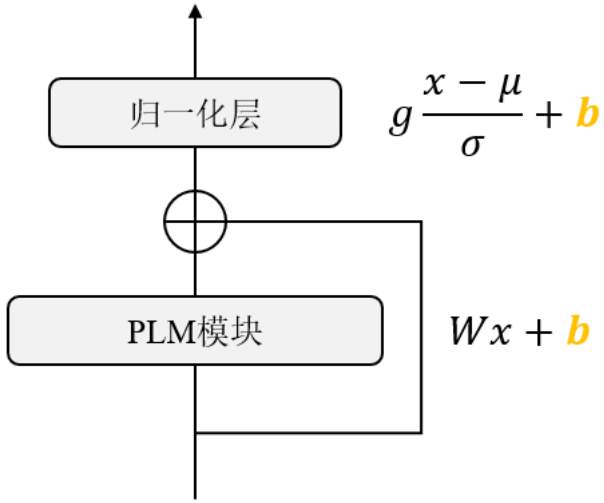
南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

指定式微调

适配器微调和前缀微调通过引入少量额外的可训练参数，实现了高效的参数微调。然而，当模型规模较大时，会导致部署困难及参数修改方式不够灵活等。为了避免引入额外参数带来的复杂性增加问题，可以选取部分参数进行微调，这种方法称为指定式（Specification-based）微调。指定式微调将原始模型中的特定参数设为可训练状态，同时将其他参数保持冻结状态。

代表性方法之一：BitFit（Bias-terms Fine-tuning）

一种更为简单、高效的稀疏微调策略，训练时只更新偏置的参数或者部分偏置参数。对于每个新任务，BitFit仅需存储偏置参数向量（这部分参数数量通常小于参数总量的0.1%）以及特定任务的最后线性分类层。如下图所示，在每个线性或卷积层中，权重矩阵W保持不变，只优化偏置向量b。对于Transformer模型而言，冻结大部分Encoder参数，只更新偏置参数跟特定任务的分类层参数。涉及的偏置参数有注意力模块中计算Query、Key、Value与合并多个注意力结果时涉及的偏置参数、MLP层中的偏置参数、归一化层的偏置参数。



BitFit需要更新的偏置参数示意图

目录

参数高效微调-重参数化微调

□ 重参数化微调（Reparametrization-based）

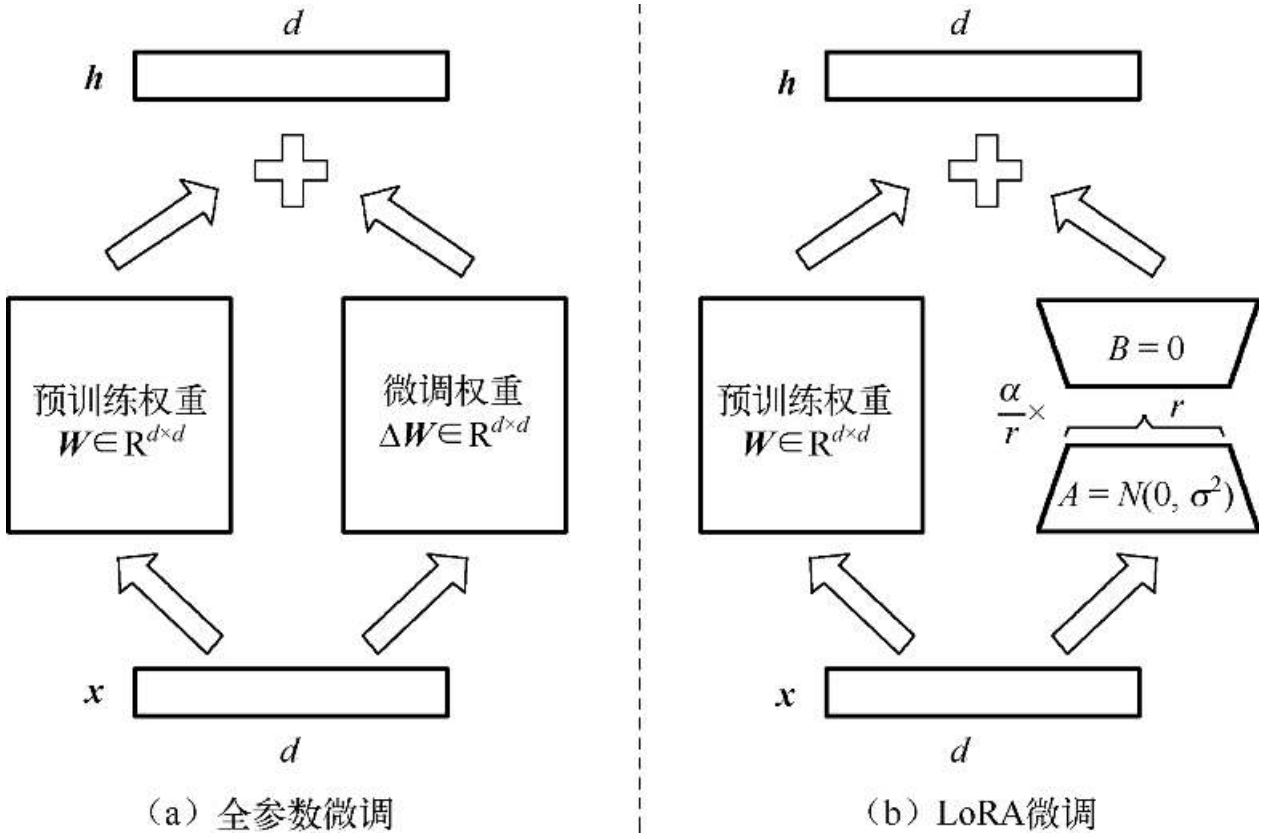
重参数化微调通过转换现有的优化过程，将其重新表达为更有效的参数形式。

在微调任务中，微调权重与初始预训练权重之间的差异经常表现出“低本征秩”的特性。这意味着它们可以被很好地近似为一个低秩矩阵。低秩矩阵具有较少的线性独立列，可以被理解为具有更低“复杂度”的矩阵，并且可以表示为两个较小矩阵的乘积。这一观察引出了一个关键的点，即微调权重与初始预训练权重之间的差异可以表示为两个较小矩阵的乘积。通过更新这两个较小的矩阵，而非整个原始权重矩阵，可以大幅提升计算效率。基于此思想，低秩适配（Low-Rank Adaptation：LoRA）微调方法被提出，并引发了广泛关注。

□ LoRA微调

LoRA微调指通过在预训练模型中引入低秩结构来实现高效的参数微调。其核心思想是通过低秩分解来修改模型的权重矩阵，使其分解为较低维度的因子，从而减少在微调过程中需要更新的参数数量。

参数高效微调-重参数化微调-LoRA



全参数微调与LoRA微调的参数构成示意图

在全参数微调方法下，模型参数可以拆分为两部分，即冻住的预训练权重$W \in R^{d \times d}$与微调过程中产生的权重更新量$\Delta W \in R^{d \times d}$，如图（a）所示。设输入为x，输出为h，则微调后h可以表示为$h = Wx + \Delta Wx$

LoRA微调方法通过对权重更新矩阵应用数学上的低秩分解，将原始的高维权重矩阵表示为两个或多个较小矩阵的乘积，实质上减少了模型参数的数量，如图（b）所示，LoRA微调方法使用两个低秩矩阵A和B近似代替增量更新ΔW，微调后的h改写为：$h = Wx + BAx$

其中，$A \in R^{r \times d}$，$B \in R^{d \times r}$；r被称为“秩”。微调参数数量从$d \times d$降低至$2 \times r \times d$，同时不改变输出数据的维度。初始化时，对A使用高斯初始化，对B使用零初始化，使得训练刚开始时BA的值为零，不会给模型引入额外的噪声。此外，使用超参数α来调整增量权重的值，h可以进一步表示成$h = Wx + \frac{\alpha}{\gamma} BAx$，实际操作中一般取$\alpha \geq r$。

参数高效微调-重参数化微调-LoRA变体

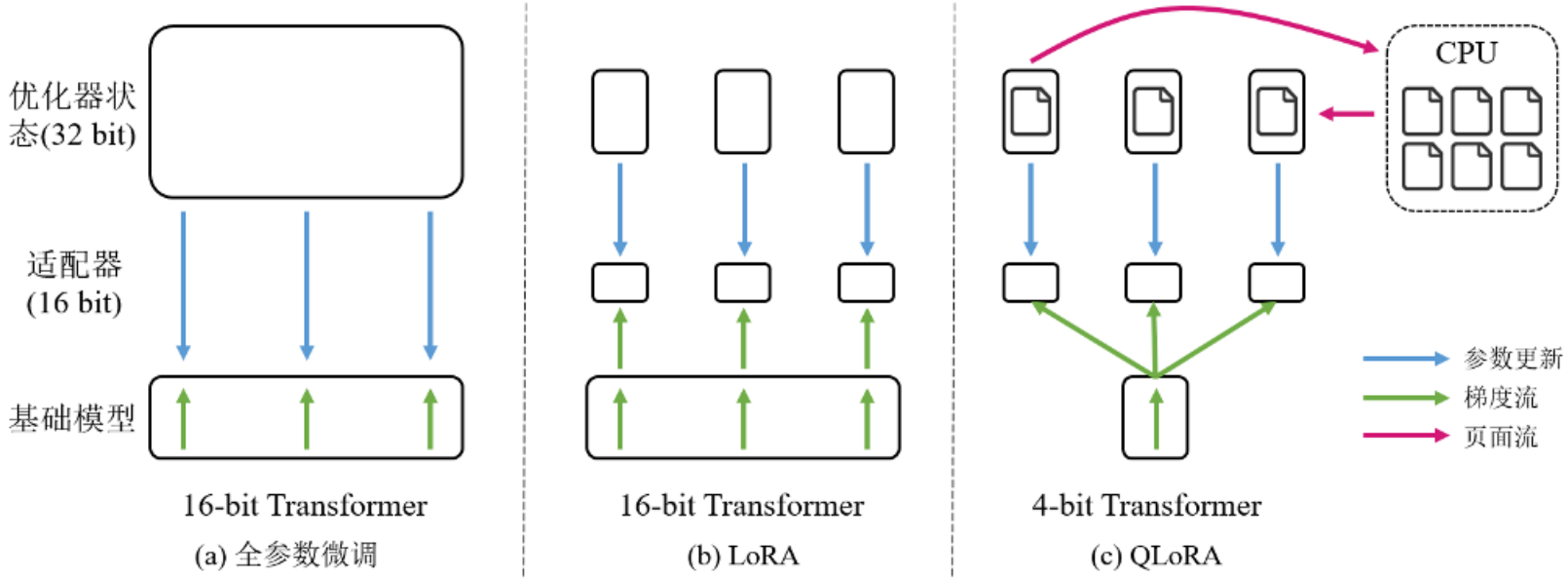
南京航空航天大学 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **自适应预算分配的参数高效微调（Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning：AdaLoRA）**

由于LoRA为所有的低秩矩阵指定了唯一秩的设置，忽视了不同模块、不同层参数在特定任务中的重要性差异，导致大模型的效果存在不稳定性。针对这一问题，自适应预算分配的参数高效微调（AdaLoRA）方法被提出，它在微调过程中根据各权重矩阵对于下游任务的重要性来动态调整秩的大小，以减少可训练参数数量的同时保持或提高性能。

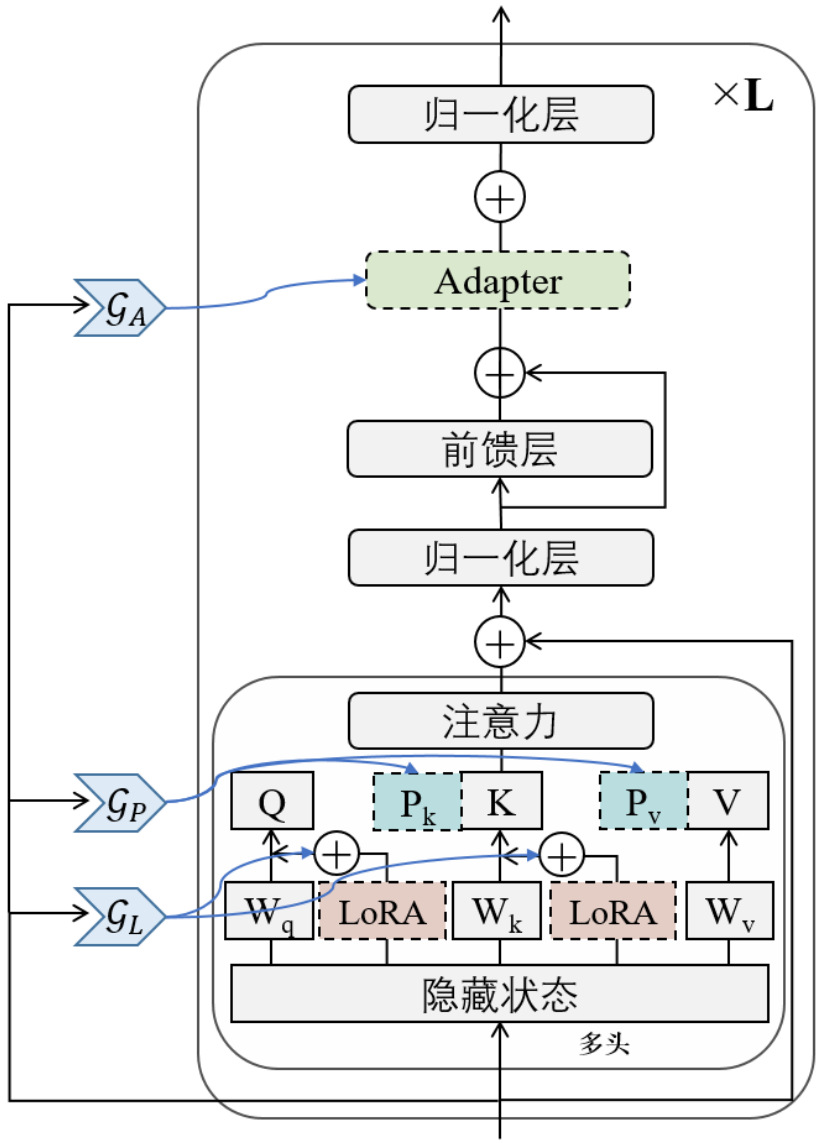
- **量化高效（Efficient Fine-Tuning of Quantized LLMs：QLoRA）微调**

量化高效微调（QLoRA）是大模型微调中一种提升模型在硬件上运行效率的技术。随着大模型参数量的不断增加，如拥有660亿一个参数的超大模型LLaMA，其显存占用高达300GB。在这样的情况下，传统的16bit量化压缩存储微调所需的显存甚至超过了780GB，使得常规的LoRA技术难以应用。面对这一挑战，QLoRA微调基于LoRA微调的逻辑，通过冻结的4bit量化预训练模型来传播梯度到低秩适配器。下图展示了不同于LoRA微调和全参数微调QLoRA的创新之处，即它巧妙地结合了量化技术和适配器方法，以在资源受限的情况下提高模型的可训练性和性能。



目录

参数高效微调-混合微调



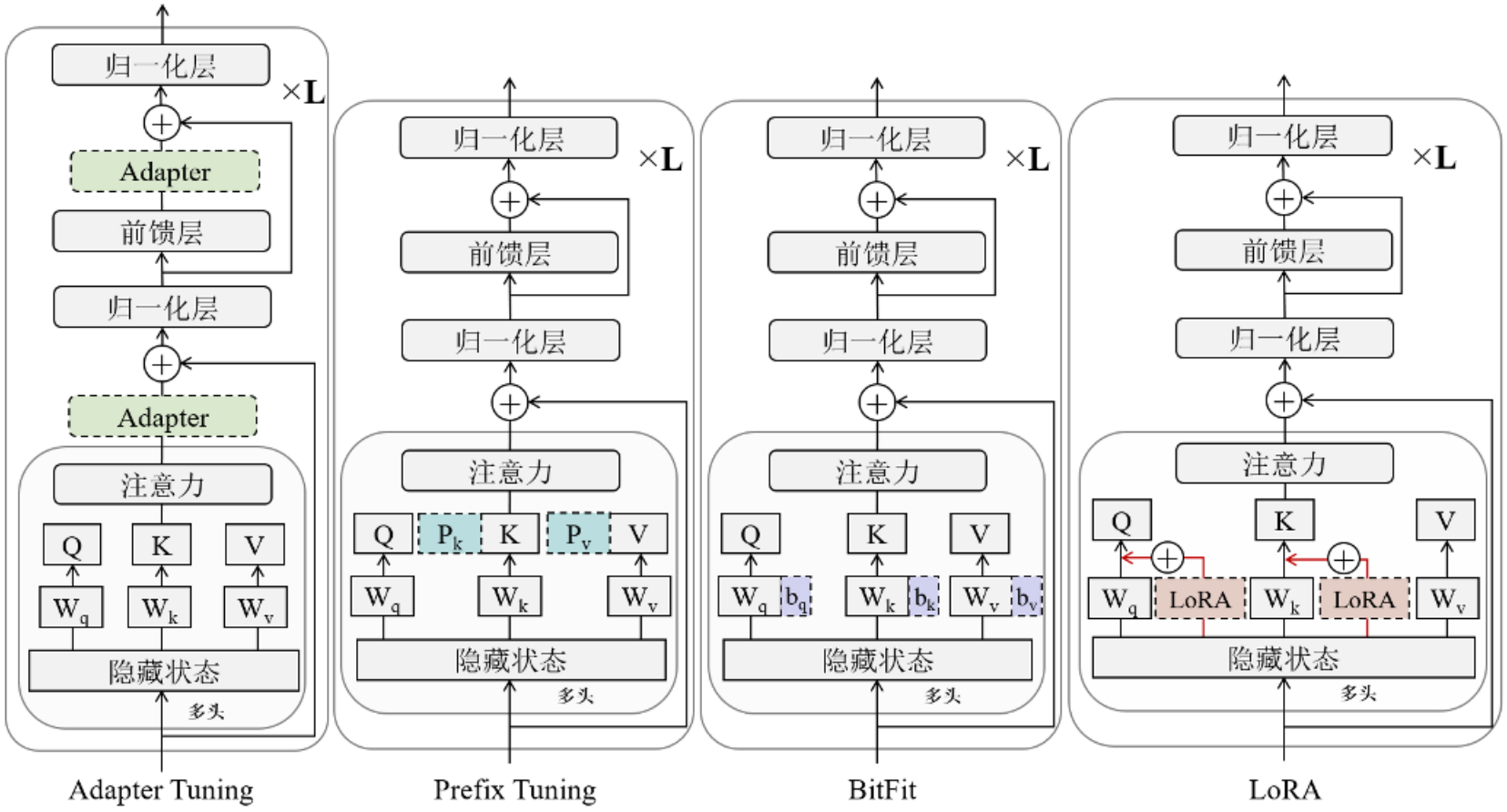
UniPELT 方法示意图

混合微调

不同的参数高效微调方法在应用于同一个任务时可能存在着巨大的性能差异，这给如何选择最合适的微调方法带来了挑战。能否将这些性能优异的方法结合起来，以获得更优的结果呢？面对这一问题，UniPELT（A Unified Framework for Parameter-Efficient Language Model Tuning）提出了一个综合性的微调框架（如左图所示），LoRA微调、前缀微调和适配器微调三种方法整合在一起，通过学习一个门控机制来动态地选择并激活适合当前任务或数据的最佳微调方法。此方法能够在不同任务或数据集上自适应地选择和调整微调方法，从而在保证高效性的同时，实现更优的微调效果。

在UniPELT框架中：LoRA重参数化应用于W_q和W_v注意力矩阵，前缀微调应用于每个Transformer层的Key和Value，并在Transformer块的前馈子层之后添加适配器。对于每个模块，使用线性层来实现门控。通过G_p参数控制前缀微调方法的开关，G_L参数控制LoRA微调方法的开关，G_A参数控制适配器微调方法的开关。如左图所示，图中颜色的部分表示可训练参数包括LoRA矩阵W_A（降维矩阵的参数）和W_B（升维矩阵的参数）、前缀微调参数P_k和P_v、适配器微调参数和门控函数权重。

参数高效微调-小结



不同参数高效微调方法对比示意图

左图展示了4种微调方法在Transformer模块上的应用方式：

适配器微调：设计适配器结构，在模型的适当位置插入适配器，仅微调适配器部分的参数。

前缀微调：在输入序列之前添加一个连续向量，仅微调前缀部分的参数。

BitFit：仅调整模型的偏置参数。

LoRA微调：引入低秩分解的矩阵，新增的矩阵权重可以与原始权重合并。

适配器微调、前缀微调属于增量式微调方法，它们通过引入额外的结构来微调参数；BitFit属于指定式微调方法，专注于调整模型中的部分参数；LoRA微调属于重参数化微调方法，将原始权重重参数化为原始矩阵与新增低秩矩阵的乘积权重之和。

参数高效微调-小结

参数高效微调方法能够有效减少微调所需的计算资源和时间，保持模型的整体性能稳定，不会对整个模型结构做出重大改变，可以在实际应用中帮助研究者更加轻松地优化大模型。参数高效微调方法具体分为增量式微调方法、指定式微调方法、重参数化微调方法以及多方法并用的混合微调方法。下表总结了常用的参数高效微调方法的优缺点及适用场景。在实际应用中，需要根据预训练模型、具体任务和数据集等因素选择合适的微调方法。

名称	优点	缺点	适用场景
适配器微调	较低的计算成本和较好的性能	增加模型层数，导致模型的参数数量和计算量增加，影响模型的效率，延长推理时间。当训练数据不足或者适配器的容量过大时，可能会导致适配器过拟合训练数据，降低模型的泛化能力	适用于处理小数据集
前缀微调	只微调预训练模型的前缀，就能达到与全参数微调相当的性能，减少了计算成本和过拟合的风险	前缀token会占用序列长度，有一定的额外计算开销	适用于各种需要添加特定前缀的自然语言处理任务，如文本分类、情感分析等
BitFit	训练参数数量极小 （约 0.1%）	在大部分任务上的效果差于适配器微调、LoRA微调等方法	适用于处理小规模到中等规模的数据集
LoRA微调	无推理延迟，可以通过可插拔的形式切换到不同的任务，易于实现和部署，简单且效果好	低秩矩阵中的维度和秩的选择对微调效果产生较大影响，需要超参数调优	适用于需要快速收敛且对模型复杂度要求较高的任务，如机器翻译和语音识别等
UniPELT	多种微调方法混合涉及模型的不同部分，使得模型的鲁棒性更好	相比于单个微调方法训练参数数量大，推理更耗时	在低数据场景中相对于单个微调方法提升更显著

目录