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

大模型微调-第7章

魏明强、宫丽娜

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**

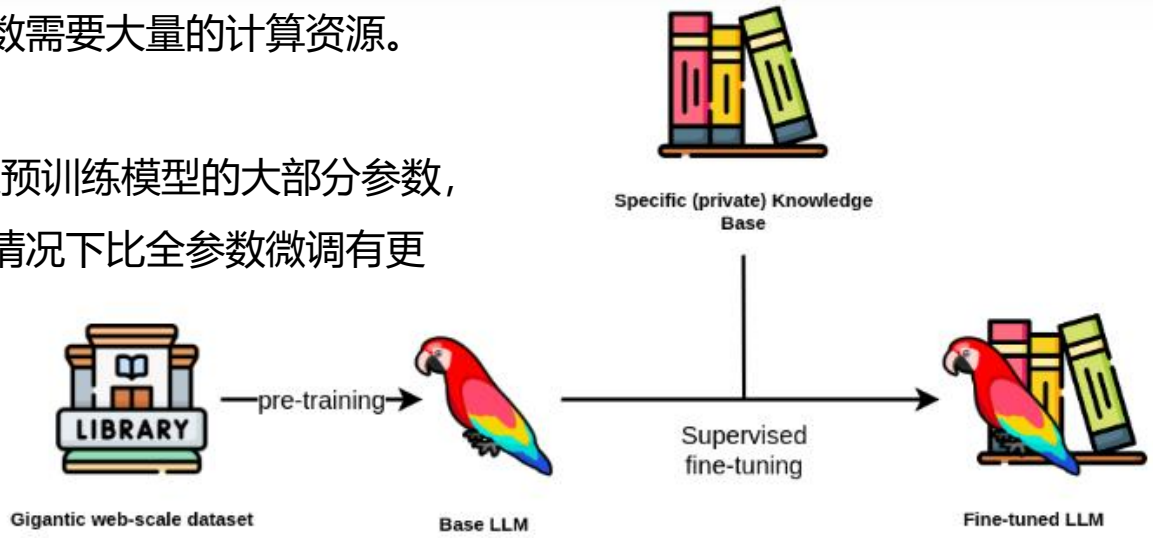
在预训练阶段，大模型通过在大量数据上进行训练学习，已经掌握了丰富的语言规则、知识信息以及视觉模式。然而，在大规模（公开）数据上通过自监督学习训练出来的模型虽然具有较好的“通识”能力（称为基础模型），却往往难以具备“专业认知”能力（称为专有模型/垂直模型）。

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。

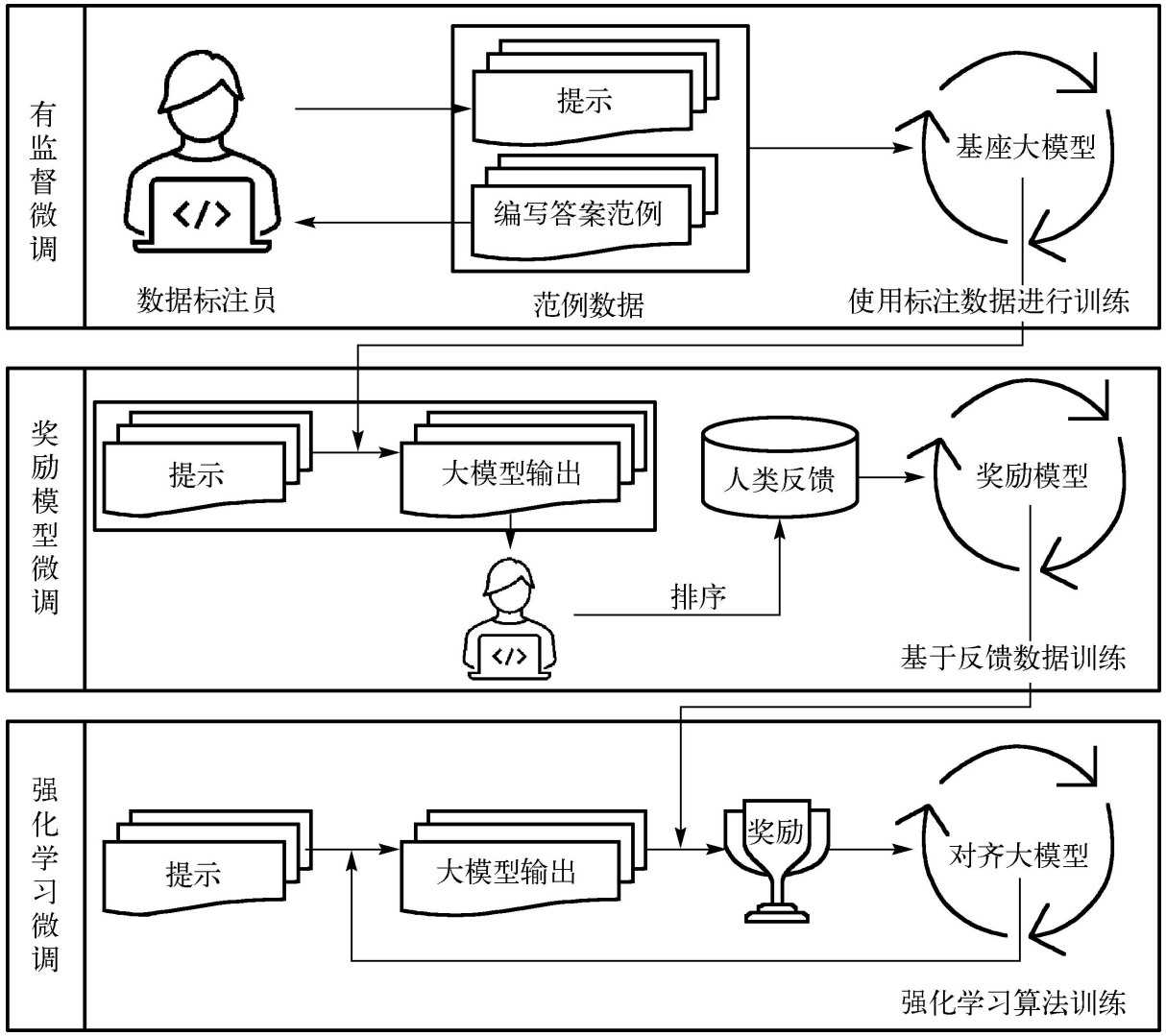
- **有监督微调分为：全参数微调和参数高效微调**

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。

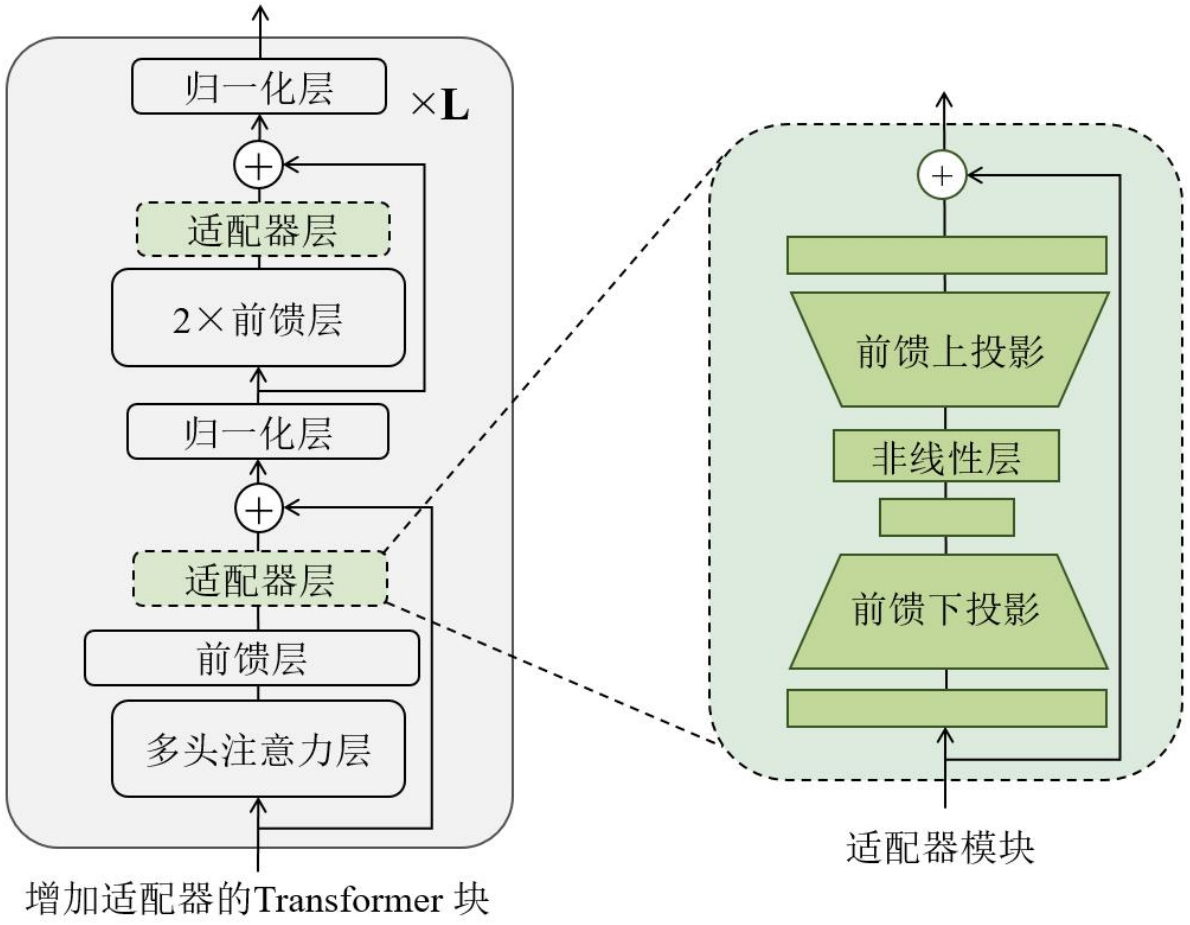
增量式微调

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

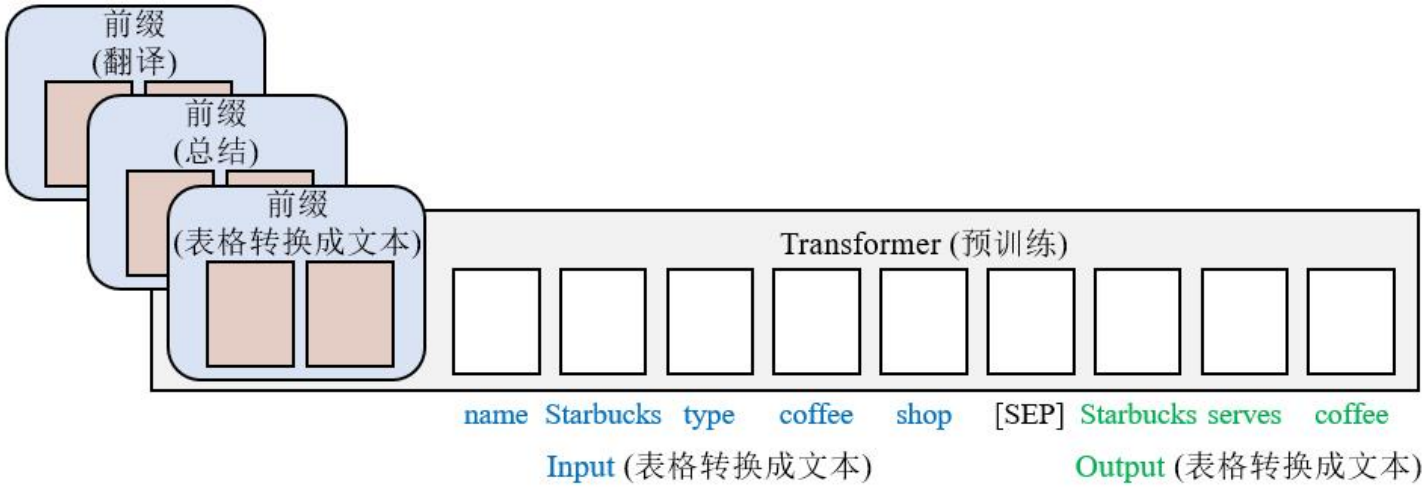
适配器微调（Adapter Tuning）是一种在预训练后的大模型中间层中，插入适配器（小型网络模块）来适应新任务的技术。在微调时将大模型主体冻结，仅训练特定于任务的参数，即适配器参数，减少训练时算力开销。以Transformer架构为例，如左图所示：

□ 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

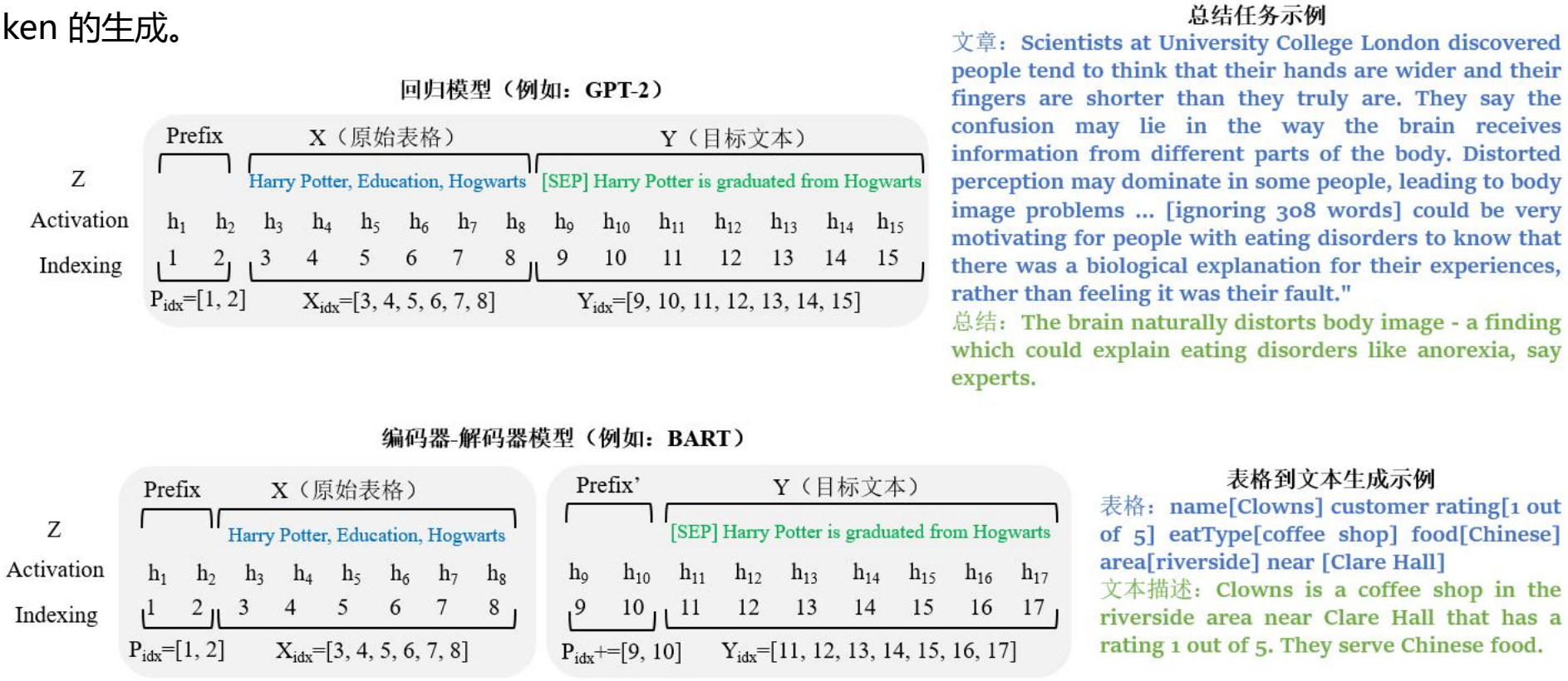
针对不同的模型结构，前缀微调需要构建不同的前缀，如下图所示：

- **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

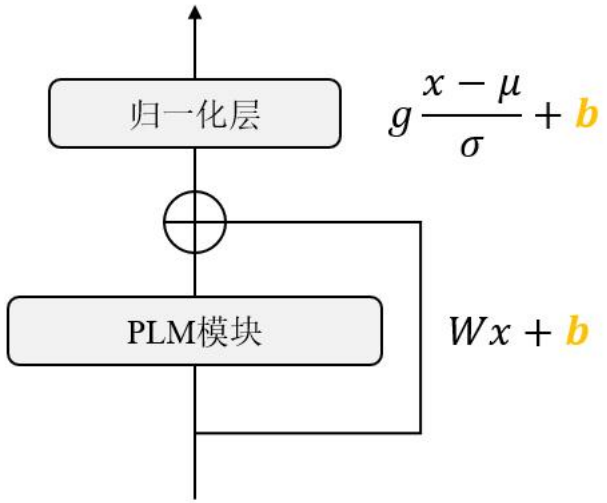
南京航空航天大学 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

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

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

□ 重参数化微调（Reparametrization-based）

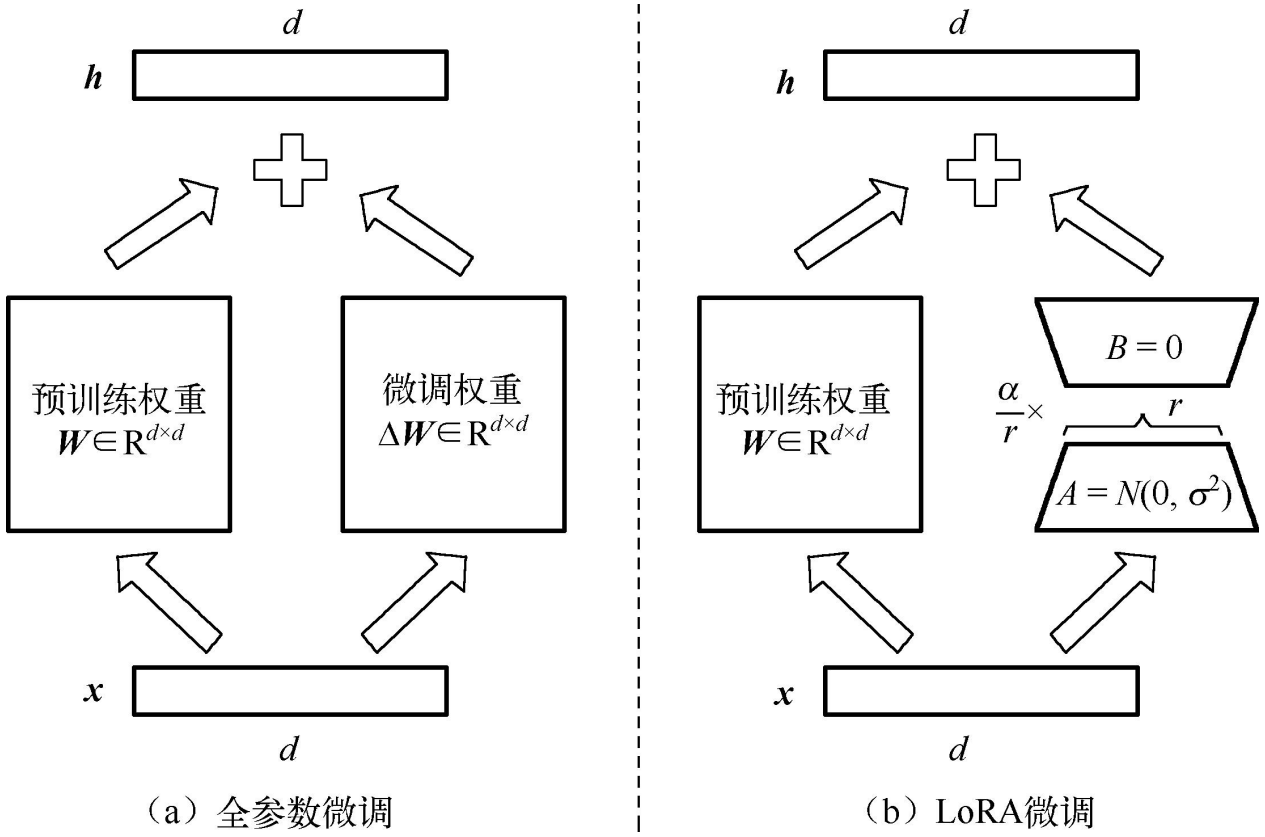
重参数化微调通过转换现有的优化过程，将其重新表达为更有效的参数形式。

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 + ΔWx

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 × d降低至2 × r × 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α ≥ r。

参数高效微调-重参数化微调-LoRA变体

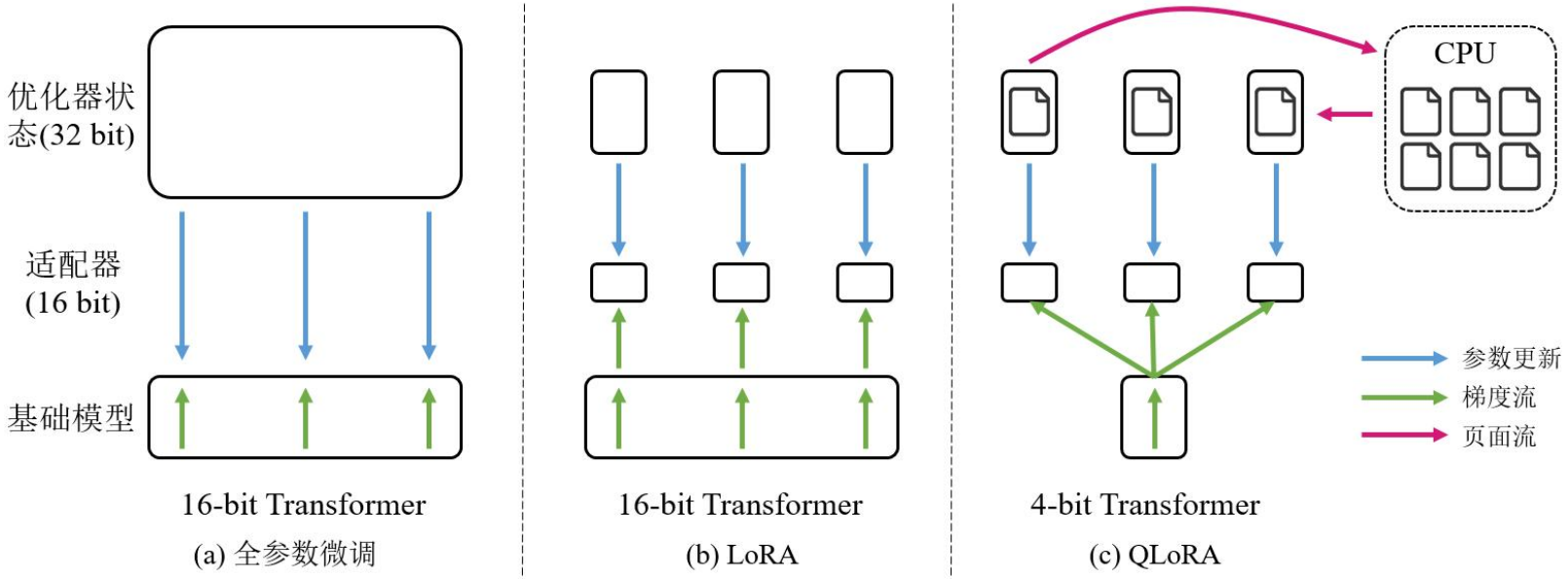
南京航空航天大学 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **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。

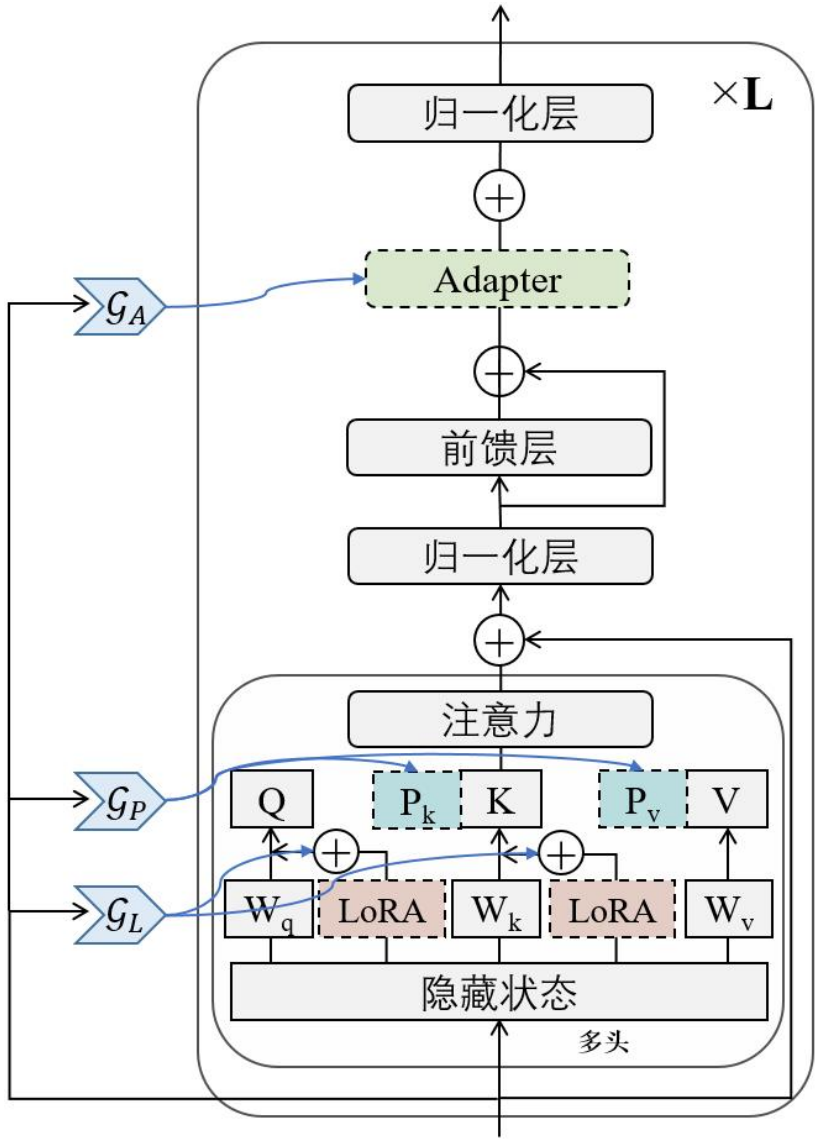
- **量化高效（Efficient Fine-Tuning of Quantized LLMs：QLoRA）微调**

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



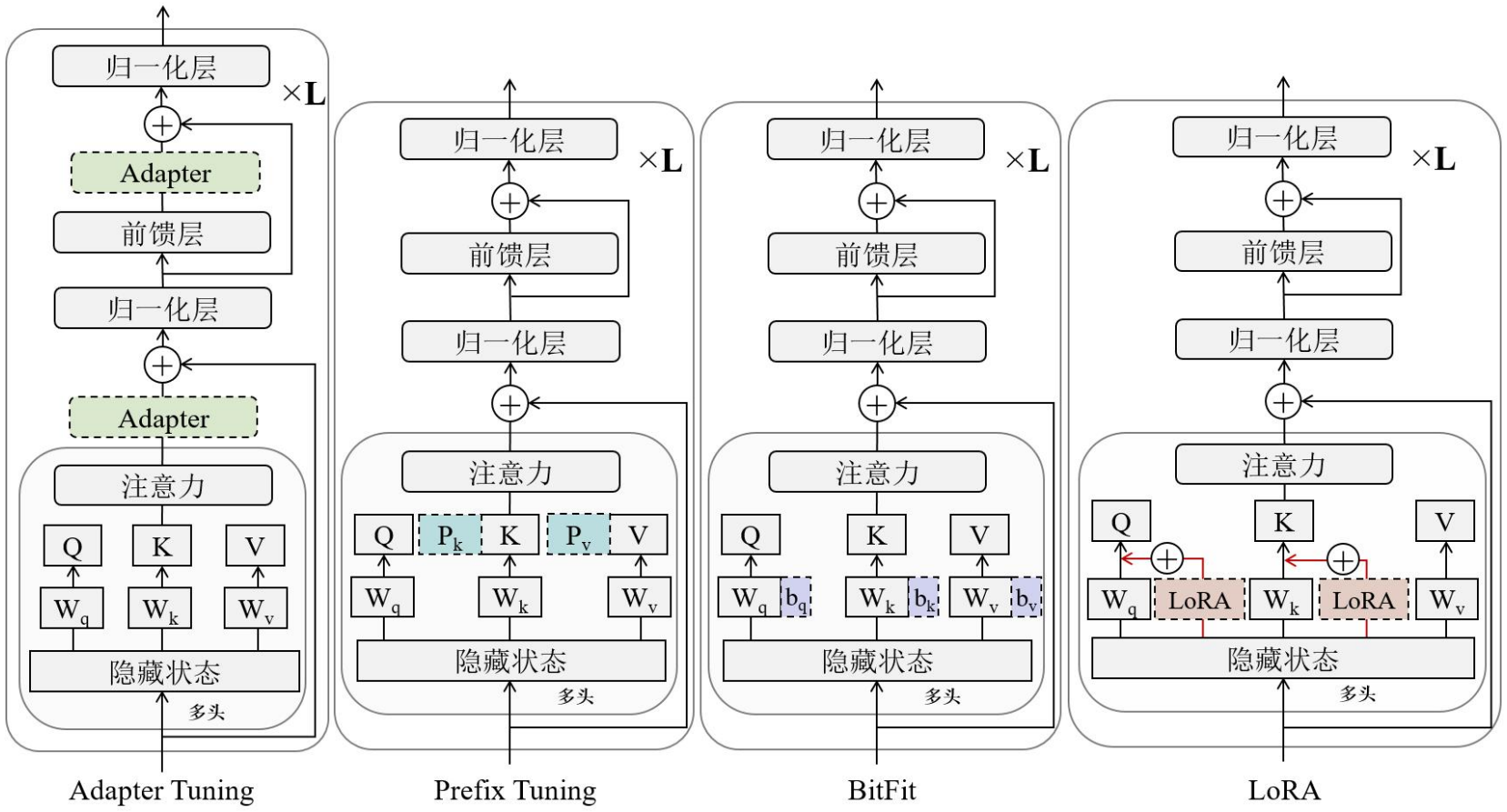
UniPELT 方法示意图

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

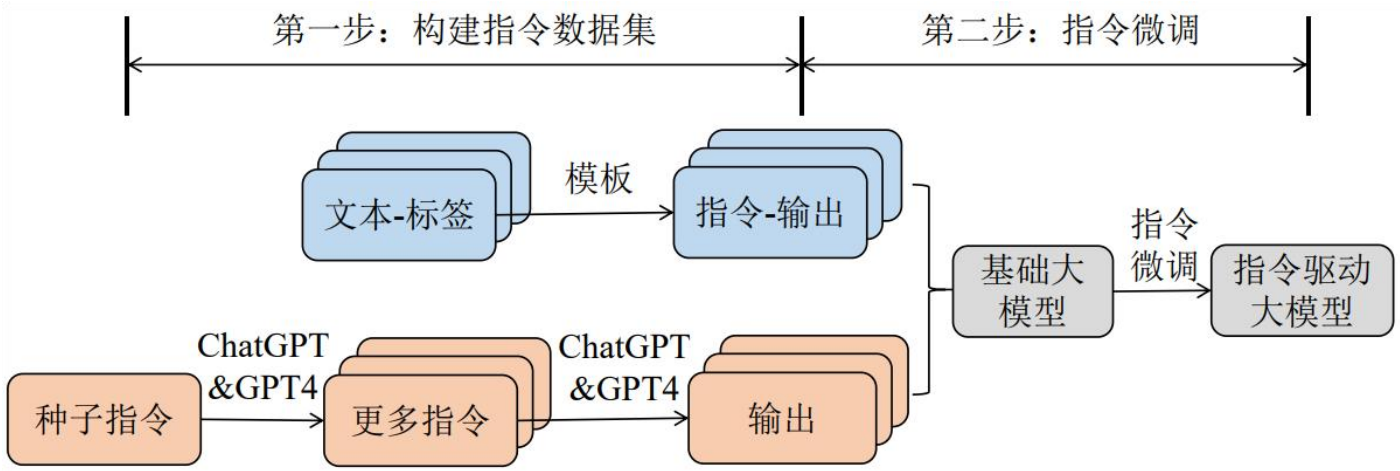
目录

指令微调

指令微调（Instruction Tuning）

模型在训练阶段存在一个关键问题，即训练目标和用户目标之间的不匹配问题。例如， 大模型通常在大型语料库上，通过最小化上下文词预测误差进行训练，而用户希望模型有效且安全地遵循他们的指令。为了解决这个问题，研究人员提出了指令微调技术，使大模型与人的任务指导或示例进行交互，根据输入和任务要求进行相应调整，从而生成更准确、更合理的回答或输出。

指令微调利用<指令，输出>数据集，以监督的方式进一步训练大模型，弥合大模型的预测目标与用户让大模型遵循人类指令的目标之间的差距，让大模型更好地适应特定应用场景或任务，提高输出的质量和准确度。这里，指令代表人类提供给大模型的指令，即指定任务的自然语言文本序列，如“写一篇关于某某主题的发言稿” “为游客出一份某某景点的旅游攻略”等；输出代表遵循指令的期望输出。也就是说，指令微调其实是一种特殊的有监督微调技术，特殊之处在于其数据集的结构，即由人类指令和期望输出组成的配对，这种结构使得指令微调专注于让模型理解和遵循人类指令。指令微调主要包含**构建指令数据集**和**指令微调**两个关键步骤，如下图所示：



指令微调的通用架构

指令微调-指令数据集构建

- **指令数据集两种构建方法：**

来自带注释的自然语言数据集的数据集成（Data Integration），即从带注释的自然语言数据集，使用模板（Template）技术将文本标签对（Text-label Pairs）转换为<指令，输出>对（Instruction-Output Pairs）。例如，Flan和 P3数据集就是通过数据集成策略构建的。

利用大模型基于指令生成输出，例如，可以使用GPT-3.5-Turbo或 GPT-4等大模型收集输出。此方法包含两个步骤：（1）通过人工收集的方式得到指令，或者先手写少量指令然后用大模型来扩充指令；（2）将收集到的指令输入大模型中以获得输出。InstructWild和Self-Instruct等数据集就是通过这种技术构建的。另外，对于多回合会话指令微调数据集，可以让大模型扮演不同的角色（如用户、AI助手）来生成会话格式的消息。

目前，根据上述两种方法构建的指令数据集一般可以分为三类：

①泛化到未见任务：这类数据集通常包含多样化的任务，每个任务都有专门的指令和数据样例。模型在这类数据集上训练后，可以泛化到未见过的新任务上。

②在单轮中遵循用户指令：这类数据集包含指令及其对应的响应，用于训练模型单轮回复用户指令。训练后，模型可以理解指令并做出回复。

③像人类一样提供帮助：这类数据集包含多轮闲聊对话。训练后，模型可以进行多轮交互，像人类一样提供帮助。

总体来说，第一类数据集侧重任务泛化能力，第二类数据集侧重单轮指令理解能力，第三类侧重连续多轮对话能力。研究人员可以根据所需的模型能力选择不同类型的数据集进行指令调优。

目录

指令微调-指令微调技术

□ 指令微调阶段

基于构建好的高质量指令数据集对基础大模型进行微调。指令微调的架构参考参数高效微调技术，即利用一小部分参数的更新来使得模型达到训练效果。其主要技术如下表所示：

□ 参数高效微调技术

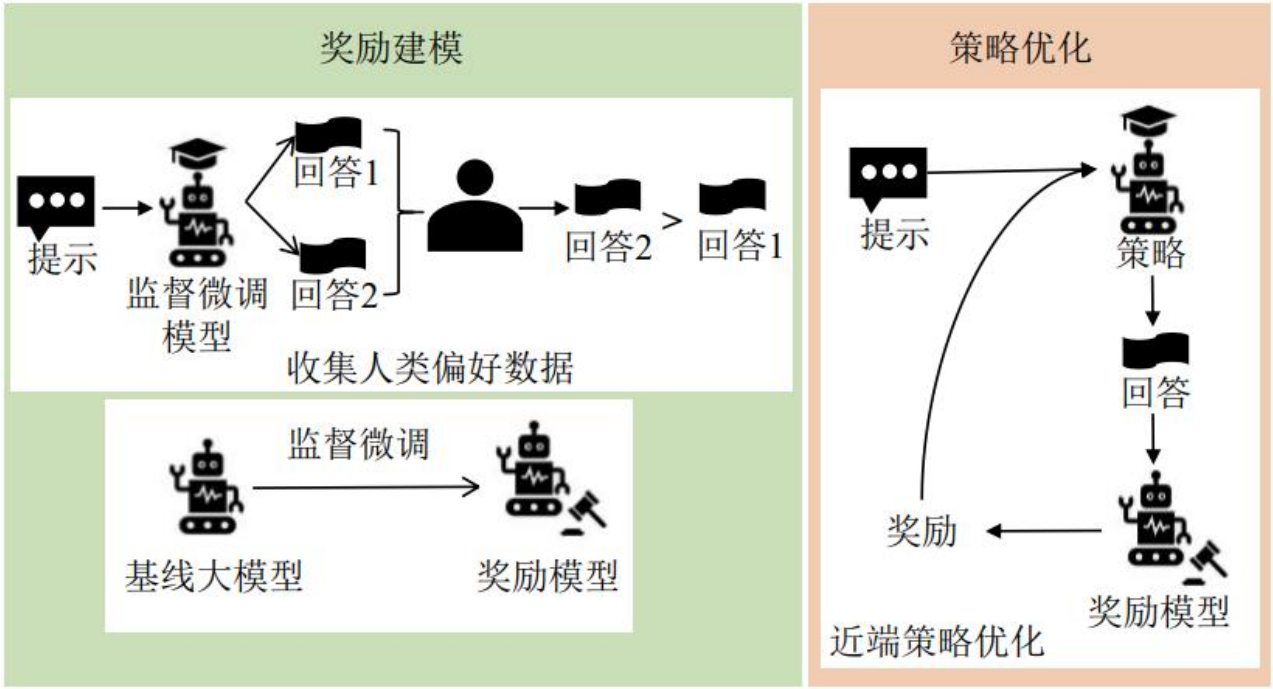
方法	原理	优势	缺点
LoRA	将模型权重分解为低秩分量进行更新，使调优局限在相关任务子空间	减少调优的参数数量，降低计算内存	低秩分解可能削弱模型表征能力
HINT	使用超网络根据指令和少量样例生成参数化模块进行模型调优	可以处理长指令，避免重复计算	调优模块性能可能弱于全量调优
Qlora	对模型权重进行量化，只调整低秩适配器参数	减少参数内存，兼容量化	量化会损失部分精度
LOMO	融合梯度计算和更新，避免完整梯度存储	减少梯度内存占用	需要精心设计保证收敛稳定
Delta- tuning	将调优参数限制在低维流形上。	提供理论分析，参数高效。	低维流形假设可能不够准确

目录

基于人类反馈的强化学习微调

经过有监督微调，大模型已经初步具备完成各种任务的能力。但有监督微调的目的是使得模型输出与标准答案完全相同，不能从整体上对模型输出质量进行判断。因此，模型不适用于解决自然语言及跨模态生成的多样性问题，也不能解决微小变化的敏感性问题。强化学习将模型输出文本作为一个整体进行考虑，其优化目标是使得模型生成高质量回复。

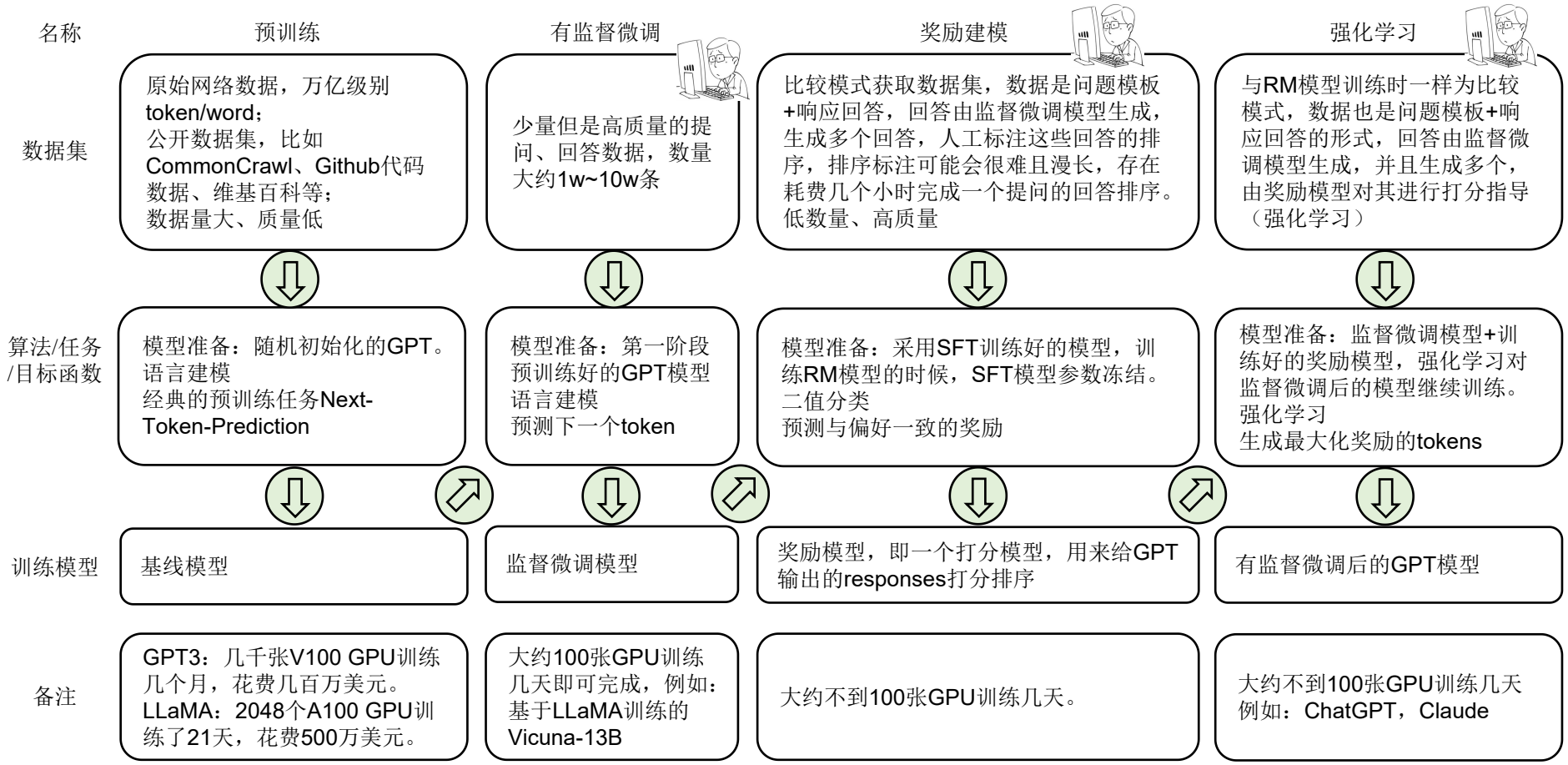
基于人类反馈的强化学习（Reinforcement Learning from Human Feedback，RLHF）是一种特殊的技术，用于与其他技术（如无监督学习、有监督学习等）一起训练AI系统，使其更加人性化。基于人类反馈的强化学习微调如下图所示，其在多种常见的大语言模型（InstructGPT、ChatGPT等）上取得了很好的表现。



基于人类反馈的强化学习微调

基于人类反馈的强化学习微调

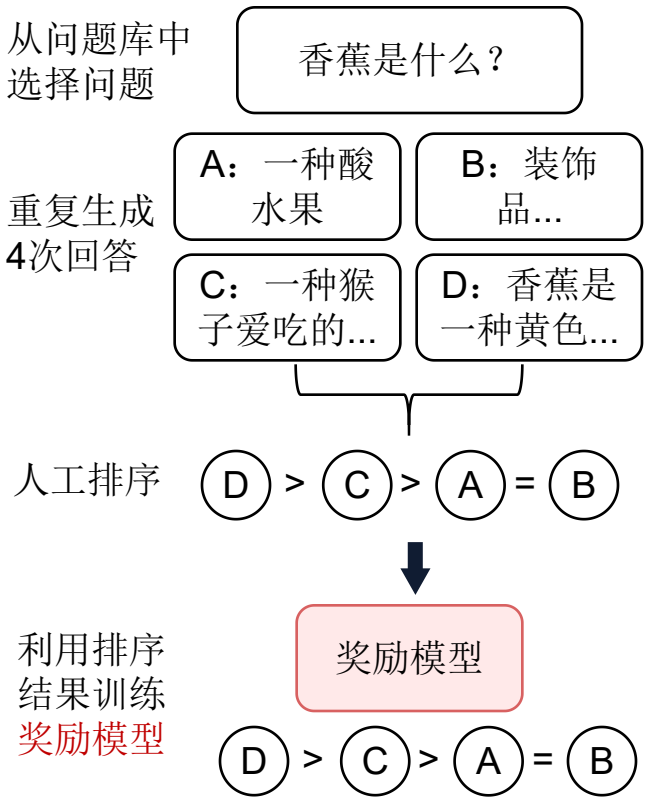
基于人类反馈的强化学习（Reinforcement Learning from Human Feedback，RLHF）完整流程如下图所示，包括预训练、有监督微调、奖励建模以及最后一步强化学习微调，接下来主要介绍奖励建模和强化学习微调部分。



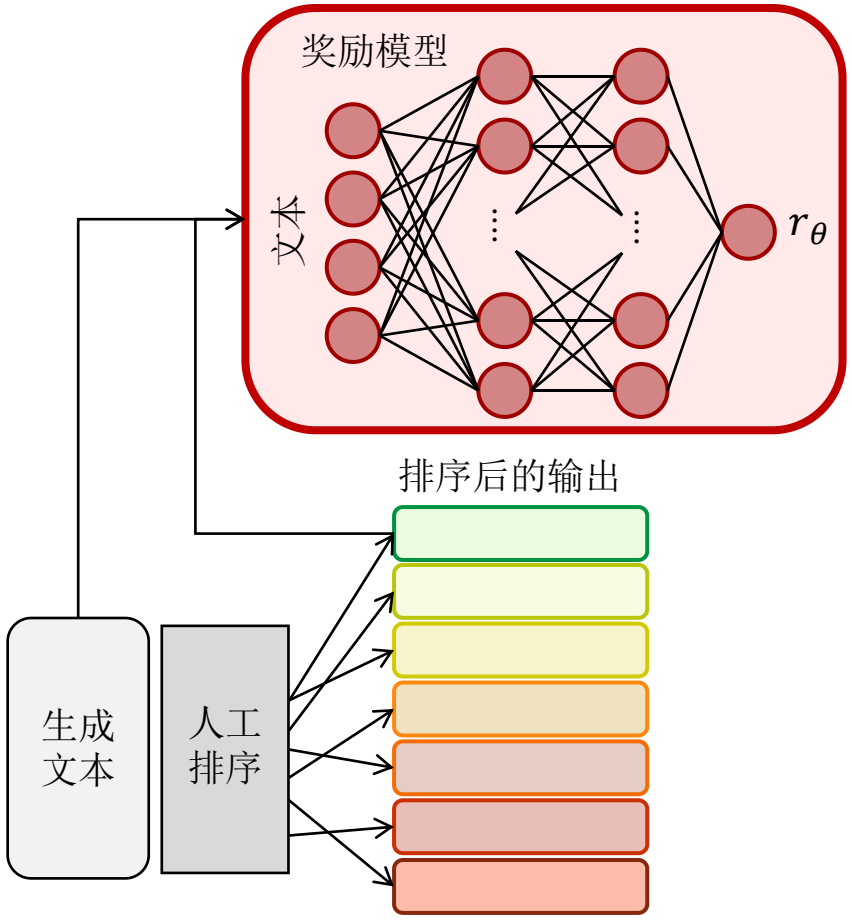
基于人类反馈的强化学习微调-奖励建模

奖励建模

奖励模型源于强化学习中的奖励函数，能对当前的状态刻画一个分数，来说明这个状态产生的价值有多少。不同于基线模型和有监督微调模型，奖励模型本身并不能直接提供给用户使用，而是通过模型拟合人类打分结果，给出关于结果质量的反馈。



（a）利用人工排序结果训练奖励模型



（b）奖励模型训练详图

奖励建模首先利用有监督微调模型生成回答数据，然后对这些回答进行人工排序，如图（a）所示；然后基于数据和排序结果训练奖励模型，如图（b）所示。奖励模型的数据集以问题模板+响应回答的形式，由有监督微调模型生成多个响应回答，然后人工标注这些响应回答之间的排名顺序。

奖励模型通过由人类反馈标注的偏好数据来学习人类的偏好，是一种模拟人类评估的过程。将有监督微调模型最后一层的非嵌入层去掉，剩余部分作为初始的奖励模型。训练模型的输入是问题和答案，输出是一个标量奖励值（分数）。样本质量越高，奖励值越大。

基于人类反馈的强化学习微调-奖励建模

奖励模型的训练中，一个x（提示、问题）对应人工排序的K个回答，回答两两一组，一个训练批次中针对每个提示有C_K^2个对比，组成一条训练数据，如(x, y_w, y_l)，则一共有C_K^2条训练数据。这群训练数据组成一个训练批次，构造奖励模型的损失函数如下：

$$L(\theta) = -\frac{1}{C_K^2} E_{(x,y_w,y_l)\sim D}[log\sigma(r_\theta(x, y_w) - r_\theta(x, y_l))]$$

其中，x表示某个提示或问题；y_w和y_l分别表示该提示下的任意一对回答，并且标注中y_w的排序高于y_l；D表示某个提示下人类标注排序的所有两两答案对；r表示奖励模型；σ表示Sigmoid函数；$r_\theta(x, y)$表示奖励模型对应提示x的标量输出。当期望回答y的排序较高时，$r_\theta(x, y)$的得分也越高。为了不让K的个数影响训练模型，在公式前面乘上$\frac{1}{C_K^2}$，将损失平均到每个答案对上。

目录

基于人类反馈的强化学习微调-策略优化

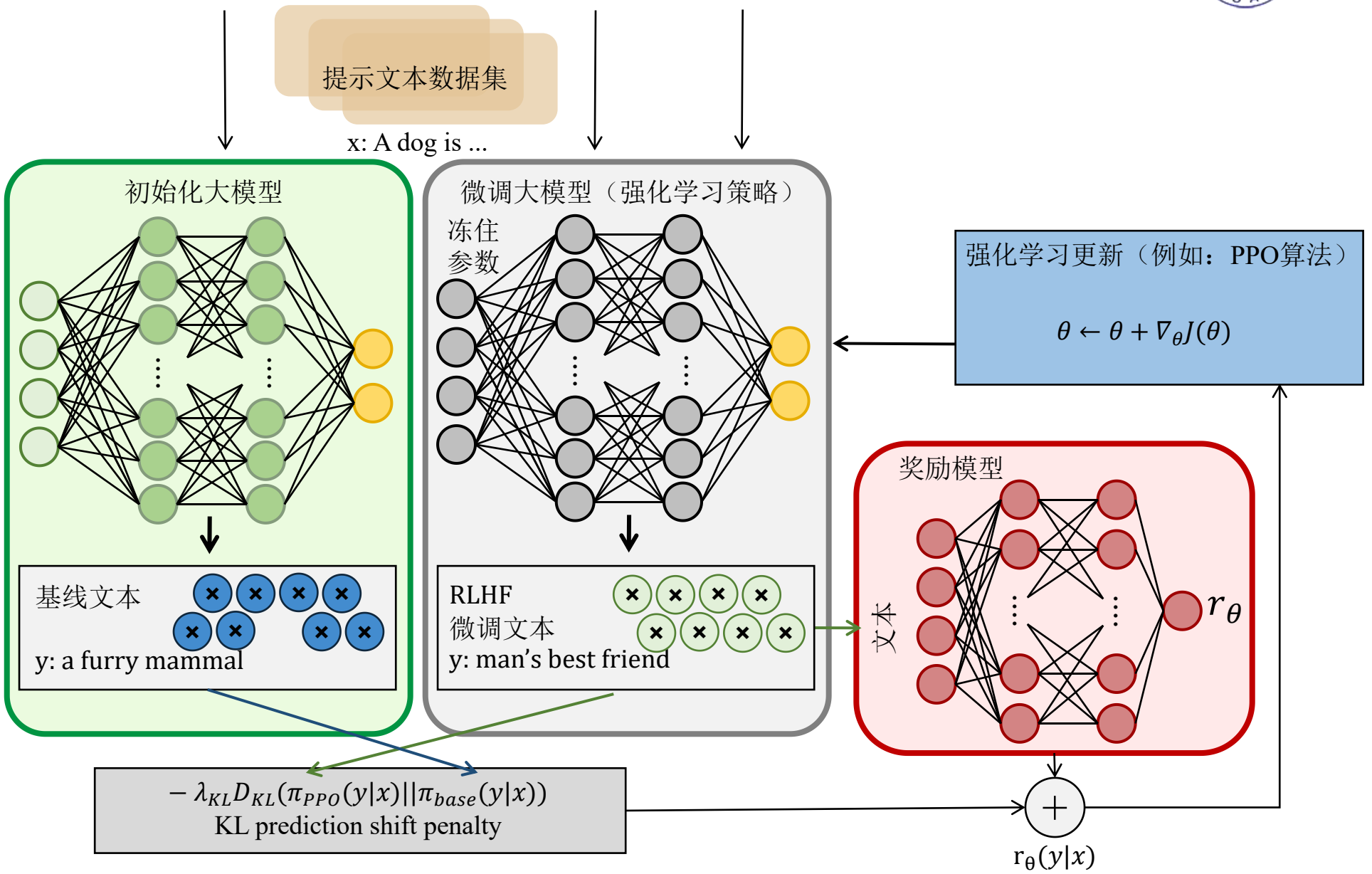
□ 强化学习微调

有监督微调后的大模型，可以根据奖励模型的奖励反馈进行进一步的微调。以近端策略优化（Proximal Policy Optimization，PPO）算法为例来介绍强化学习微调，PPO是一种深度强化学习算法，用于训练智能体（Agent）在复杂环境中如何学习和执行任务，即通过智能体与环境交互获得最大的回报（Reward），从而达成指定任务目标。PPO根据奖励模型获得的反馈优化模型，通过不断的迭代，让模型探索和发现更符合人类偏好的回复策略。

首先将提示x输入初始模型和当前微调的模型，分别得到输出文本y_w

y_l，将来自当前策略的文本传递给奖励模型得到一个标量的奖励r_θ。在OpenAI、Anthropic和DeepMind等多篇论文中，设计奖励为输出词分布序列之间的KL散度（Kullback-Leibler Divergence）的缩放，即$r_\theta = r_\theta - \lambda r_{KL}$。其中$KL$散度被用于惩罚强化学习策略在每个训练批次中生成大幅偏离初始模型，以确保模型输出合理连贯的文本。如果去掉这一惩罚项可能导致模型在优化中生成乱码文本来愚弄奖励模型提供高奖励值。PPO微调模型结构如下一页图所示。

基于人类反馈的强化学习微调-PPO微调模型结构



基于人类反馈的强化学习微调-策略优化

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

- **PPO微调模型结构解析：**

ChatGPT使用改良版的PPO对GPT再次训练，将训练梯度混合到PPO梯度中，在强化学习训练中最大化以下组合目标函数：

$$O(\phi) = E_{(x,y)\sim D_{\pi_\phi^{RL}}}[r_\phi(x,y) - \beta log(\pi_\phi^{RL}(y|x)/\pi^{SFT}(y|x))] + \gamma E_{x\sim D_{pre}}[log(\pi_\phi^{RL}(x))]$$

其中，π_ϕ^{RL}表示此刻要学的强化学习模型，又称为策略；π^{SFT}表示有监督微调模型，初始时$\pi_\phi^{RL} = \pi^{SFT}$；$r_\phi$表示奖励模型。拆分每一项，目标是最大化损失函数。

（1）$E_{(x,y)\sim D_{\pi_\phi^{RL}}}$中，x表示输入提示，把x输入当前状态的强化学习模型中会产生y。

（2）$r_\phi(x,y)$表示在当前强化模型下，将x和其所产生的y送入奖励模型打分。

（3）$log(\pi_\phi^{RL}(y|x)/\pi^{SFT}(y|x))$表示KL散度，其结果值≥ 0，用于比较两个模型的输出分布是否相似，KL值越大，分布越不相似，当分布相同时，$KL = 0$。本阶段希望强化学习后得到的模型，在能够理解人类意图的基础上，又不要和最原始的模型的输出相差太远。参数β表示对这种偏差的容忍程度，偏离越远，就要从奖励模型的基础上得到越多的惩罚。截止到这一步，称为PPO。

（4）$E_{x\sim D_{pre}}$中，表示在有监督微调之前，最初始的预训练模型。

（5）$log(\pi_\phi^{RL}(x))$表示将来自初始模型的数据送入当前强化学习模型中，γ表示对这种偏离的惩罚程度，防止当前强化学习模型输出分布偏离太多。添加上这一项以后的优化策略称为PPO-ptx。

作为一个可选项，基于人类反馈的强化学习微调可以通过迭代奖励模型和策略共同优化。随着策略模型更新，用户可以继续将输出和早期的输出进行合并排名。

目录

基于人类反馈的强化学习微调-案例讲解

InstructGPT是OpenAI第一个流行的基于人类反馈的强化学习微调模型，使用了较小版本的GPT-3初始化模型。ChatGPT沿用了OpenAI 2022年3月提出的InstructGPT训练框架，将原本的GPT-3替换成了GPT-3.5，同时在InstructGPT的基础上进一步优化了多轮对话效果。以OpenAI公开的 GPT如何从“初始模型”一直训练成ChatGPT这样的“助手模型”为例，完整介绍基于人类反馈的强化学习微调技术，主要包括三个步骤：有监督微调、奖励建模和强化学习微调：

- **有监督微调：**

此步骤需要收集高质量的提示和回答数据对，主要有两个来源，一部分数据是从早期InstructGPT版本的OpenAI API中采样的，其他数据是由标注器/标定者提供的，包括三种类型的提示，即普通提示（任意的任务）、少样本提示（具有多个查询/响应对的指令）和基于用户的提示（OpenAI中应用程序请求的特定用例）。此步骤的训练与预训练相比只是更换了数据集。

- **奖励建模：**

经过有监督微调，大模型已经初步具备了完成各种任务的能力，接下来需要进行奖励建模。输入的数据是问题模板和响应回答，问题模板是由OpenAI API和人工标定的，响应回答是有监督模型生成的，每个提示生成4～9个回答，人工给这些回答排序。基于有监督微调训练的模型，训练奖励模型时冻住有监督微调模型的参数，将有监督微调模型的最后一层修改为一个线性层，模型的目标是预测打分，预测打分的顺序和标注的顺序之间的损失可以定义为：

$$L(\theta) = -\frac{1}{C_K^2} E_{(x,y_w,y_l)\sim D}[log\sigma(r_\theta(x,y_w) - r_\theta(x,y_l))]$$

其中，$r_\theta(x,y)$是奖励模型基于提示生成的答案。针对问题x的回答，y_w的排序高于y_l。D是奖励模型的数据集，θ是模型参数。

基于人类反馈的强化学习微调-案例讲解

- **强化学习微调：**

模型学会了怎么说话，同时我们又训练出了一个独立的奖励模型，这时需要把两者结合起来，让模型能够更好地对齐人类意图。利用奖励模型输出的奖励，采用PPO策略（强化学习）微调优化模型。选择PPO算法对有监督微调后的模型进行进一步微调。

PPO作为强化学习模型中的代理，从第一步开始使用有监督微调后的模型进行初始化。该环境是一个提示生成器，它生成随机输入提示以及针对这个提示的期望响应。奖励模型给出奖励，用于对提示和响应进行评分。强化学习模型训练的目标是最大化以下组合目标函数：

$$RL(\phi) = E_{(x,y)\sim D_{\pi_\phi^{RL}}}[r_\phi(x,y) - \beta log(\pi_\phi^{RL}(y|x)/\pi^{SFT}(y|x))] + \gamma E_{x\sim D_{pre}}[log(\pi_\phi^{RL}(x))]$$

其中，π_ϕ^{RL}和π^{SFT}分别表示策略模型和有监督微调后的模型；D_{pre}表示预训练分布；β和γ分别控制KL散度和预训练梯度。目标函数中，奖励模型对提示-响应对进行评分，得分越高表明响应越好。KL散度用于测量PPO和有监督微调模型生成的响应分布之间的距离。

大规模无监督预训练模型可以学习广泛的知识和简单的推理能力。但是由于预训练完全无监督学习，难以精准控制，为提升其可控性，需要采用基于人类反馈的强化学习对无监督的模型进行微调，以使其与人类偏好相一致。InstructGPT是OpenAI第一个流行的基于人类反馈的强化学习微调模型，ChatGPT沿用此训练框架，将预训练模型GPT微调为人类的“助手模型”。ChatGPT在学术和工业界均具有重要意义，越来越多的研究团体和企业正在追随 OpenAI的脚步，开发自己的类ChatGPT产品或AIGC产品。例如，微软将ChatGPT与其搜索引擎Bing结合起来以提高搜索质量；百度发布了类ChatGPT的机器人ERNIE Bot，可以根据文本描述生成图像；商汤开发了SenseChat机器人，它可以根据文本描述生成图像、视频和3D内容。ChatGPT相关技术引起了全世界的关注，成为计算机科学与AI领域一支重要的力量。

目录

思考

参数高效微调、指令微调以及基于人类反馈的强化学习微调技术，构成了将预训练的基础大型模型有效应用于垂直领域的基石。目前，大模型通过微调技术已经取得了显著进展。以人类所能理解的方式解释大模型的行为，是可信地使用它们的基础。然而，大模型仍然存在许多难以解释的方面，这引发了人们对其应用和可信度的疑问。

首先，当前大模型的工作原理很大程度上是一个黑盒，这意味着无法准确理解其内部运行机制。虽然有监督微调技术可以提升模型性能，但现有理论无法充分解释“自监督预训练+有监督微调+人类反馈对齐”方式所产生的大模型的强大能力和幻觉错误。因此，需要更多的基础理论和方法来解释大模型的行为，以使其更可信地应用于实际问题中。

其次，针对大模型系统的可信度问题也需要深入思考。尽管大模型在许多任务中表现出色，但仍然需要解决如何确保在关键应用中使用这些模型时的可靠性和安全性。这可能涉及对模型的验证和审计，以及对模型输出的解释和解释能力的提高。

最后，需要建立更深入的理解，以解释大模型智能涌现现象。这些现象指的是模型在面对新任务或环境时表现出的出乎意料的智能和创造力。通过深入研究这些现象背后的原理，人们可以更好地理解模型的工作方式，并为未来的研究和应用提供更多的启示，以更好地发挥大模型的潜力，推动AI技术的发展和应用。

目录

理论习题：

1、在机器学习背景下，解释什么是大模型微调？

答：大模型微调是指在一个预训练好的大型模型基础上，通过使用新的数据集或任务进行额外训练，以使模型适应新的任务或领域。

2、大模型微调包含哪些步骤？

答：大模型微调的一般步骤包括选择预训练模型、调整模型结构、选择合适的学习率，然后使用新的数据集进行训练。

3、大模型微调可能会面临哪些挑战和考虑因素？

答：挑战和考虑因素可能包括新任务的数据分布、模型过拟合的风险、计算资源等。

4、迁移学习与大模型微调在概念上有何关联？

答：迁移学习是指通过从一个任务学到的知识来改善不同但相关任务的性能，而大模型微调通常是迁移学习的一种方式。

5、如何在大模型微调后评估模型性能？

答：可以使用新任务的评估指标，比如准确度、精确度、召回率等，来评估大模型微调后的性能。监控模型在微调过程中的表现也是重要的。

6、解释基于类人反馈的强化学习，以及它与传统的强化学习方法有何不同？提供一个实际的应用场景。

答：基于类人反馈的强化学习是一种强化学习方法，其中智能体通过模仿类人的行为来学习任务。与传统的强化学习方法不同，基于类人反馈的方法强调通过参考类人的经验来提高学习效率。例如，在机器人领域，一个机器人可以通过观察并模仿人类执行特定任务的方式，加速学习过程。

7、大模型微调中需要针对有用性和无害性收集大量数据，有用性和无害性分别表示什么意思？

答：数据收集主要针对有用性和无害性，有用性是指在数据收集过程中，让标注人员使用模型，期望模型帮助用户完成纯粹基于文本的任务（比如回答问题、撰写编辑文档、讨论计划和决策）。无害性是指在数据收集过程中，让标注人员通过一些敌对性的询问，比如计划抢银行，引诱模型给出一些违背规则的有害性回答。

实践习题：

1、安装部署PEFT环境，随机初始化一组预期收益率和协方差，计算并绘制资产的有效边界。

2、使用适配器微调大型语言模型的场景，假设你已经加载了预训练的BERT模型，现在想要添加一个适配器层用于微调情感分类任务。请列出至少两个实现适配器微调的步骤，并提供相应的Python代码片段。

3、使用前缀微调方法微调大型语言模型以执行中文翻译成英文的任务。假设你已经加载了预训练的T5模型，现在想通过前缀微调使其适应中文翻译成英文的任务。
原文本：“这是一个关于大模型微调的问题，希望通过前缀微调的方式来适应中文翻译成英文的任务。”

4、使用 LoRA微调大型语言模型以执行文本摘要任务。假设你已经加载了预训练的BERT模型，现在想通过 LoRA 微调使其适应文本摘要任务。

原文本：“自然语言处理（Natural Language Processing，简称NLP）是人工智能领域中与计算机和人类语言之间交互的研究。NLP的目标是使计算机能够理解、解释、生成人类语言，使计算机与人的交互更加自然。它涉及到文本处理、语音处理、机器翻译等多个方面的任务。”使用 LoRA 微调方法，将上述文本进行摘要生成，提取出主要信息。

谢谢！
Thanks!