大语言模型微调技术的研究综述

张钦彤,王昱超,王鹤羲,王俊鑫,陈 海
北京师范大学珠海校区 文理学院,广东 珠海 519087

摘 要:大型语言模型的崛起是深度学习领域的全新里程碑,而微调技术在优化模型性能方面的起到了关键作用。对大型语言模型微调技术进行了全面的综述,回顾了语言模型的统计语言模型、神经网络语言模型、预训练语言模型和大语言模型四个阶段的发展历程和微调技术的基本概念,从经典参数微调、高效参数微调、提示微调和强化学习微调方法四大部分,探讨总结了各微调技术的原理与发展,并进行了一定的对比分析。最后,总结了当前微调技术的研究状况与发展重点,强调了该领域的潜在研究价值,并展望了未来的发展方向。

关键词:大语言模型;微调方法;预训练模型;自然语言处理
文献标志码:A **中图分类号:**TP18 doi:10.3778/j.issn.1002-8331.2312-0035

Comprehensive Review of Large Language Model Fine-Tuning

ZHANG Qintong, WANG Yuchao, WANG Hexi, WANG Junxin, CHEN Hai
School of Arts and Sciences, Beijing Normal University at Zhuhai, Zhuhai, Guangdong 519087, China

Abstract: The rise of large-scale language models signifies a new milestone in the field of deep learning, with fine-tuning techniques playing a crucial role in optimizing model performance. This paper provides a comprehensive overview of fine-tuning techniques for large-scale language models. It reviews the development stages of language models, including statistical language models, neural network language models, pre-trained language models, and large language models. The basic concepts of fine-tuning are explored, covering classic fine-tuning, efficient parameter fine-tuning, prompt tuning, and reinforcement learning fine-tuning. The paper delves into the principles and development of each fine-tuning technique, offering a comparative analysis across these four major categories. In conclusion, the paper summarizes the current state of research on fine-tuning techniques and underscores the potential research value in this domain, providing insights into future directions of development.

Key words: large language model; fine-tuning methods; pre-trained models; natural language processing

大型语言模型的兴起标志着深度学习领域的全新里程碑,而微调是连接预训练大型模型与特定应用场景的桥梁。随着人工智能领域的发展,微调技术对于提升模型的适用性、效率和精确度至关重要。微调技术在模型经过大规模数据集的预训练后,再在特定任务的特定数据集上进行精细调整,从而将预训练期间获得的广泛知识导入到具体应用中[1]。这种策略可以看作是迁移学习的核心,实现了广泛知识与特定应用的无缝结合,对于处理具有特定语境或专业知识的任务尤为关键。

微调对于自然语言处理技术的发展与大语言模型性能的提高具有较高的价值。首先,它避免了从零开始训练大型模型的巨大开销,减少了大量计算资源的消耗,使得研究者和开发者在资源有限的情况下也能够进行有效的模型训练。此外,由于模型在预训练阶段已经学习了大量的通用知识,故在有限的标注数据情况下,微调能够使模型更有效地学习特定任务的特征[2]。

以OpenAI的研发工作为例,GPT-1(generative pre-trained transformer)[3]在传统语言模型“预训练+微调”范式的基础上进行了进一步发展,开辟了大语言模型的微调时代;GPT[4]进一步证明了大语言模型的参数扩展法则[5]。GPT-3[6]则提出了上下文学习(in-context learning),为提示微调(prompt tuning)方向带来了发展;基于指令微调方法(instruction tuning)的InstructGPT与基于人类反馈的强化学习技术(RLHF)微调的GPT-4,

基金项目:广东省教育科学规划课题(2022GXJK417);认知智能全国重点实验室智能教育开放课题(iED2023-005)。
作者简介:张钦彤(2003—),女,研究方向为数据科学、人工智能;王昱超(2003—),男,研究方向为人工智能、自然语言处理;王鹤羲(2002—),女,研究方向为人工智能、模式识别;王俊鑫(2001—),男,研究方向为人工智能、软件工程;陈海(1974—),通信作者,女,副教授,研究方向为智能语音、智能教育、模式识别,E-mail:isabell@bnu.edu.cn。
收稿日期:2023-12-04 **修回日期:**2024-05-08 **文章编号:**1002-8331(2024)17-0017-17

达到了惊人的效果[7]。在我国,也有许多领先的大预言模型积极采用各种先进的微调方法,如知识增强大预言模型“文心一言”,结合了监督微调、人类反馈的强化学习、提示学习等微调技术。除此之外,讯飞星火认知大模型、腾讯混元大模型等诸多优秀大语言模型,已经达到甚至超过国外先进水平。

最近几年,微调领域与大型语言模型结合的相关方面,经历了快速的创新和发展。从对学习率等超参数的传统的微调策略[8]到指令调整等新兴策略[9],这一过程得到了Zhang等人[10]的深入探讨。Han等人[11]以及Qiu等人[12]的综述展示了预训练模型的发展全景。Liu等人[13]对比总结了近些年提示策略的迅猛发展。Ding等人[14]则深入探索了不增加参数的情况下如何优化微调。

许多学者都为此做出了贡献,涉及到预训练大语言模型微调技术的综述也较多。但大多综述性论文立足于某种具体的微调技术的基本原理和整体发展,且篇幅较长。因此,与之相比,本文旨在大型语言模型发展的背景下,重点地对下游任务微调的代表技术与发展历程进行分析总结,将微调技术的发展方向主要划分为传统微调改进、资源优化微调、任务适应性微调和性能优化微调。针对性强,覆盖全面,篇幅精简,利于读者快速了解大语言模型微调的发展现状。本文首先重点介绍了语言模型的发展历程;然后,详细介绍了传统的微调技术、高效参数微调、提示微调和强化学习微调;最后,进行了对比总结。本文主要架构以及重要参考文献,如图1所示[3-7,10,15-104]。

1　语言模型与微调的发展

语言模型的发展历程大致可以分为四个阶段(如图2所示):统计语言模型(statistical language model, SLM)、神经网络语言模型(neural network language models, NNLM)、预训练语言模型(pre-trained language model, PLM)和大语言模型(large language model, LLM)[105]。

1.1　统计语言模型

统计语言模型(SLM)的起源和早期发展都与语音识别紧密相关[15],它为语音识别系统提供了一种建模语言的方法,以改进自动语音识别的准确性。后来SLM的概念扩展到自然语言处理的其他领域:机器翻译、文档分类和路由、光学字符识别、信息检索、手写识别、拼写纠正等等[15-16]。SLM的目标是根据上下文估计词串的概率从而预测下一个词,具有固定上下文长度的被称为N元语法模型N-gram。在研究过程中虽然有不少统计模型被提出,但是主导着SLM研究的仍然是语法模型,尤其是二元语法模型(bigram)和三元语法模型(trigram)[17]。后来研究者们又在这个简单的模型上做

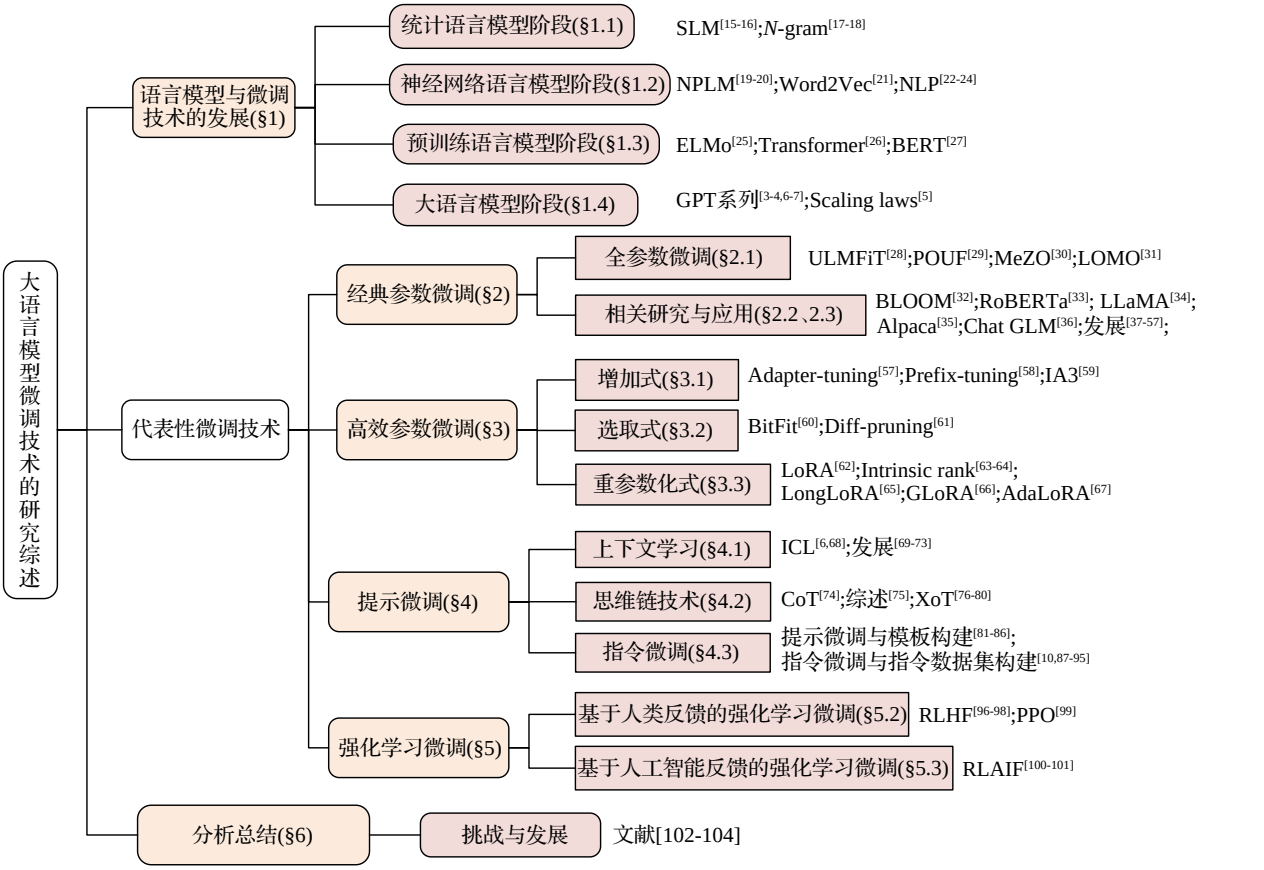

图1　本文架构图

Fig.1　Structure diagram of this paper

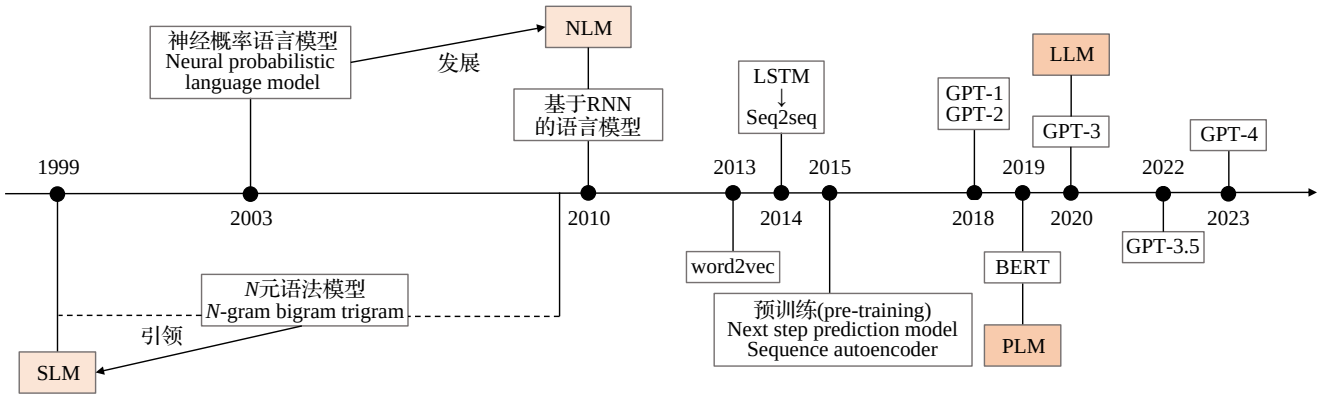

图2 语言模型的发展历程

Fig.2 Evolutionary trajectory of language models

出了许多改进:缓存、聚类、高阶语法模型、跳跃模型和句子混合模型等等。由于语言的分类特性和巨大的日常词汇量,SLM必须估计大量的参数,因此,SLM的性能依赖于可用训练数据的数量。在这个阶段,研究人员多通过平滑技术来调整语言模型的性能,例如,Goodman[18]在2001提出了一种新的平滑技术,有效地提高了多元语法模型*N*-gram处理罕见词的能力。

1.2 神经网络语言模型

神经网络语言模型(NLM)的起源是Bengio等人[19]提出的神经概率语言模型,但当时神经网络的作用仅限于简单地为已有的SLM提供单一特征[20]。直到2010年提出了基于RNN的语言模型,并证明其性能优于传统的*N*-gram模型,语言模型正式进入NLM阶段[22]。2013年,Mikolov等人[21]又提出了词向量模型(Word2Vec),将单词表示为分布式向量,并将分布式词表示的学习问题简化为一个监督学习问题,使用浅层神经网络来实现这一目标(而不是过去的词序列模型),这种方法非常简洁、有效,Word2Vec也成为了自然语言处理中的一个重要里程碑。Mikolov等人[21-22]的工作启发了更多研究者将不同的神经网络作为语言模型的架构,例如将长短期记忆网络(LSTM)应用于序列到序列学习[24]。在这个阶段,研究者们通常对神经网络中的参数进行仔细调整以获得更好的性能。此外还可以使用预训练方法:使用下一步预测模型作为无监督方法,或者使用序列自动编码器(用循环神经网络RNN将长输入序列读取到单个向量中)并用这个向量来重建原始序列。从这两种预训练方法中获得的参数可被用作神经网络的初始化,以改进训练和泛化效果[25]。

1.3 预训练语言模型

预训练语言模型(PLM)的出现是由于研究者们希望将预先训练的语言表示应用于下游任务,在当时有两种策略:基于特征的方法和微调。基于特征的方法的代表是嵌入式语言模型(embeddings from language models,ELMo)[25],使用双向LSTM模型来建模单词的复杂特征(语法、语义等)以及这些特征在上下文中的变化。Transformer结构的问世,促进了自然语言处理领域的发展[26]。

BERT模型的成功宣告了PLM阶段正式开始,BERT使用大规模的未标记文本数据来预训练神经网络语言模型[27]。微调方法的代表则是OpenAI的GPT模型,通过对所有预训练参数进行简单的微调来训练模型适应下游任务。值得注意的是,与GPT模型的从左到右的单向预训练方式不同,BERT的预训练过程是双向的,这使得BERT能够捕捉上下文中的双向关系,因此只需一个额外输出层即可对预训练的BERT模型进行微调,调整出模型去适应各种任务,而无需对特定于任务的架构进行大量修改。在添加特定输出层后,每个下游任务都需要相应的标记数据用于微调模型。这些标记数据用于计算任务特定输出层的损失,以便优化模型参数以适应特定任务。另外,微调过程中,BERT不会为每个下游任务创建单独的参数,即预训练参数通常是共享的,这意味着模型会保留在大规模文本数据上学到的通用语言表示。BERT模型的双向预训练方式、通用性微调模式为NLP研究提供了一种新的范例,具有革命性的影响,为之后的语言模型奠定了基础。

1.4 大语言模型

BERT、GPT1和GPT2等预训练模型的进步,奠定了大模型时代自然语言处理技术的基础。这些早期典型的PLM模型在特定的自然语言处理任务上表现出色,但是由于参数规模和处理能力的不足,这些模型在复杂文本结构的与上下文含义理解、语言细微差距的捕捉能力上还存在欠缺。除此之外,受限于全参数微调的原理,PLM模型在特定领域的微调训练时过度关注于训练数据,进而导致了泛化能力的不足,与较高的时间与计算资源成本。

随着自然语言处理技术的进步,对模型的处理能力、理解能力和生成能力的要求越来越高,与此同时,参数扩展法则指出,模型的规模和其性能之间存在正相关关系,通过增加模型的参数量,可以显著提高模型处理

复杂任务的能力。为了解决PLM模型存在的弊端，OpenAI在继承发展PLM的Transformer结构和与训练思想的基础上，提出了GPT-3模型，宣告了大语言模型(LLM)阶段的开始。LLM模型继承和发展了PLM模型的基础架构和训练步骤，是PLM在巨大参数量和复杂结构下的拓展，强调了语言模型的规模与泛化能力。

虽然在此之前已经存在一些较大规模的语言模型，但GPT-3在规模和性能上迈出了巨大的一步，作为第一个规模达到百亿级别参数的模型，1 750亿个参数使得GPT-3在各种自然语言处理任务上表现出色，包括文本生成、翻译、问答等。随后，GPT-3.5在此基础上进行了优化，增强了上下文理解和对话连续性方面的能力。为了节约计算资源与时间成本，GPT-3.5通过使用对任务描述和特定的提示(prompts)来执行下游任务，极大提高了模型在少样本(few-shot learning)和零样本(zero-shot learning)环境中的学习能力[6]，即模型在只有极少或没有标记数据的情况下执行下游任务。GPT-4的参数量可达数万亿，不仅在性能上达到了新高度，还极大挖掘了自然语言处理在多模态能力上的潜力。在此之间，AI21所研发的Jurassic、Google推出的LLaMA[33-34]、BigScience推出的BLOOM[37]和清华大学研发的GLM-120B[59]等大语言模型相继问世，极大地带动了大语言模型时代的发展。

随着大语言模型的发展，NLP任务中采用预训练的模型已经逐渐成为主流模型，其中不乏全程参数微调成功的案例。但是由于预训练模型的参数量越来越大，需要调整的部分多，绝大部分的大语言模型的研究并没有采用传统全参数微调的方法[13]，原因在于下游子任务的细化程度高，在微调的过程中需要调整大量参数，因而需要更多的计算资源，于是研究者们开始思考更为高效的微调方法。

2 经典参数微调

2.1 全参数微调

经典的微调方法主要指全参数微调(full-parameter fine-tuning)，在效果上被认为是一种比高效参数微调更强大的方法[38]。在得到预训练模型后，为了使模型适应下游子任务，使用目标任务相契合的较少量特定任务数据继续训练模型。训练过程中，预训练模型的权重被更新，以便更好地适应具体的下游任务场景[28]。

全参数微调的原理类似于模型预训练，不同之处在于，所有的参数都已经有了一个较好的初始值，即使用较少数据继续在初始值的基础上继续训练模型更新参数。

记模型为$f(\cdot)$，已有预训练参数$\boldsymbol{\theta}\in\mathbf{R}^p$，学习率$\alpha$，训练数据集$D_T=\{X_i,y_i\}_{i=1}^N$，损失函数$L$(具体形式取决于具体模型)，则微调目标为：

$$\min_{\boldsymbol{\theta}_T}\frac{1}{N}\sum_{i=1}^{N}L(f(X_i;\boldsymbol{\theta}_T),y_i)+\lambda R(\boldsymbol{\theta}_T) \tag{1}$$

其中，$R(\boldsymbol{\theta})$为正则项惩罚项。为了使在新的任务上损失函数最小，则可以在原始的与训练参数的基础上，进行求解。优化算法往往可以选择梯度下降等方法，设置较小或者可以动态变化的学习率，进行微调参数求解：

$$\boldsymbol{\theta}_T=\boldsymbol{\theta}_T-\alpha\nabla C(\boldsymbol{\theta}_T) \tag{2}$$

2.2 全参数微调的发展

全参数微调目前主要指监督微调，主要应用在下游子任务的模型迁移中，而无监督微调往往使用在模型的预训练阶段。无监督微调、自监督微调等在下游子任务上的微调应用还在发展阶段。有学者进一步考虑二者在微调中对模型中的归纳偏置(inductive biases)以及它们是否与训练集D或者任务集T的属性相关，而划分为行为微调(behavior fine-tuning)和自适应微调(adaptive fine-tuning)[39]。

监督微调[28]：监督微调源自深度学习中的微调思想，首先对大型语言模型进行预训练，得到在某些通用数据集上较优的参数结果。随后使用交叉熵损失，对特定任务的标记数据集上的模型进行微调，使得模型的参数在初始参数的基础上调整，以最小化监督任务的损失函数。

2018年，BERT提出了法通过在大量文本上预训练语言模型，随后在特定任务上进行微调来优化模型性能的方法，为后续的全参数微调研究奠定了基础。随后，为了提高微调效果和性能，Howard和Ruder[41]基于迁移学习方法，通过逐层解冻和差异化的学习策略，提出了一种通用的语言微调模型(universal language model fine-tuning，ULMFiT)，显著提高了微调的效率与效果。Víctor等人[42]将行为转移(BT)与监督微调相结合，在预训练模型与下游任务差别较大时具有良好的微调效果。

在优化微调使用资源的方面，Malladi等人[43]提出了一个内存高效的零阶优化器(memory-efficient zeroth-order optimizer，MeZO)，采用了经典的零阶随机梯度下降(zeroth order stochastic gradient descent，ZO-SGD)算法，并且微调的内存消耗减少到原来的1/12，显著优于上下文学习和线性探测，对全参数和参数高效调优技术都有优化作用。Lv等人[44]提出了低内存优化(low-memory optimization，LOMO)，融合了梯度计算和一步参数更新，将内存使用减少到10.8%。

基于传统的监督微调以及对其对于微调性能与所需资源的权衡，目前主要的发展方向主要可以分为高效参数微调、提示微调等微调技术，本文将在第3章、4章进行详细展开。

无监督微调：无监督微调是在训练集无标签的情况下，对预训练模型进行进一步的训练，以试图让模型自

行学习数据的结构和模式。在无监督微调中,模型不是从人工标注的数据中学习,而是通过自监督任务,如掩码语言建模、下一词句的预测等,来进一步理解和处理与特定任务相关的数据和语言模式。这种方法可以在不依赖大量标记数据的情况下提高模型在特定任务上的性能,尤其适用于标记数据稀缺或成本高昂的情况。尽管无监督微调是一个非常热门的话题,但是由于现有的无监督学习方法通常需要大规模的数据来正常微调[42],否则将会破坏原先的预训练表示结构,因此没得到非常全面和深入的研究。

Li等人[46]认为源数据在将微调范式从监督转向无监督时至关重要,并提出了两种简单而有效的策略:稀疏源数据回放和数据混合,将源数据和目标数据合并为无监督微调,使得微调效果更加有效,并直接称之为无监督微调(unsupervised tuning)。有些学者着重于弱监督微调的改进与研究[47-49]。Yu等人[50]提出了COSINE1,在弱监督的情况下对预训练的LLM进行微调,抑制标签噪声的同时用潜在的噪声标签丰富了数据。Huang等人[29]在解决named entity recognition问题时,基于自我监督与训练语言模型PLMs提出了结合有限的标记数据进行半监督学习的方法。Tanwisuth等人[51]提出了POUF(prompt-oriented unsupervised fine-tuning),直接利用未标记的目标数据对预先训练的基于提示的具有零射击能力的大型模型进行微调。目前,基于反馈的强化学习微调方法,是无监督微调极具发展潜力的方向,本文将在第5章详细展开。针对传统微调技术的主要改进技术如表1所示。

2.3 总结与应用

全参数微调可以较快速地迁移学习[51]。由于已经拥有预训练模型的基础,因此可以较为快速迁移地学习将预训练模型应用到新的任务中,相对减少了训练时间和资源消耗。但是在微调过程中,容易出现过拟合问题,进而由于过拟合而导致微调后的模型泛化性能下降[52],因此对于微调数据的质量和数量具有较高的要求[53]。

缩放定律指出,模型的效果和模型大小N、训练数据集大小D和训练计算次数C,有着密切的关系[5]。现在流行的大语言模型不仅具有很大数据集,而且训练代价昂贵,包含了数千亿甚至更多的参数,包括自注意力权重、前馈神经网络、词嵌入参数、归一化和残差连接参数和模型超参数等,使得全参数微调面临着巨大的困难。因此,如何高效地进行模型微调就成了NLP学术研究界的重点。

目前,传统的微调技术微调已经被较广泛地用在了各式各样的自然语言处理任务上。其中,监督微调得到了重点的发展,应用在包括文本分析与分类、命名实体连接、摘要概述、句意分析、问题解答、情感分析等具体任务上[30-55]。而无监督微调受限于不含标记,主要依赖于模型自身的能力来理解数据,目前主要应用在各大模型的进一步预训练中。

在大语言模型中,GPT模型是首先微调成功且性能良好的大语言模型典范,其使用无监督微调在大量文本上进行了预训练,GPT-2又进一步扩大了参数与数据集规模,随后在各种任务上进行监督微调的效果均较为优异。BigScience 2022年所提出的BLOOM系列模型,在包含了130亿字符的文本集上进行微调并通过独立验证,选择最优的模型[56];此外,对于13亿和71亿参数的版本,还使用了SGPT Bi-Encoder方法进行了对比微调[57]。对于BERT模型[27],由于其较小的参数规模,在较多场景下全参数微调可以以较低的成本得到较高的性能[31],因此,很多研究人员在BERT模型的基础上进行了广泛的微调与应用。RoBERTa[33]、LLaMA系列[34]与一些较为小型的大语言模型,如斯坦福大学研发的Alpaca模型[35]、清华大学研发的Chat GLM 6B[36]等,均支持成本较低的全参数微调。

3 高效参数微调

在大语言模型的模型规模普及的背景下,对计算效率和资源使用的优化尤为重要。资源优化微调的目的在于减少微调时的资源消耗,例如通过只训练模型的一部分参数(如Transformer的最后几层)来实现。这种方法在尽可能保持性能的同时,减少了计算资源和时间的需求。在此背景下,一种优化参数量的微调训练方法

表1 传统微调技术的改进

Table 1 Improvements to traditional fine-tuning techniques

微调方法	创新点	优点	缺点
MeZO	零阶随机梯度下降; 梯度估计更新模型参数	节省训练空间; 减少内存消耗	训练步骤多; 实现复杂
LOMO	融合梯度计算与参数更新; 对目标函数采样和评估; 近似梯度更新参数	节省训练空间; 减少内存消耗	训练速度较慢
UT	稀疏数据重引; 数据混合	有效学习小规模未标记数据,具有一定泛化性	依赖于源数据和目标数据的性质与规模
POUF	使用未标记数据; 基于提示引导模型学习	在不增加数据标注的情况下提高模型性能,具备零射击能力	泛化能力难以保证; 依赖高质量的提示设计

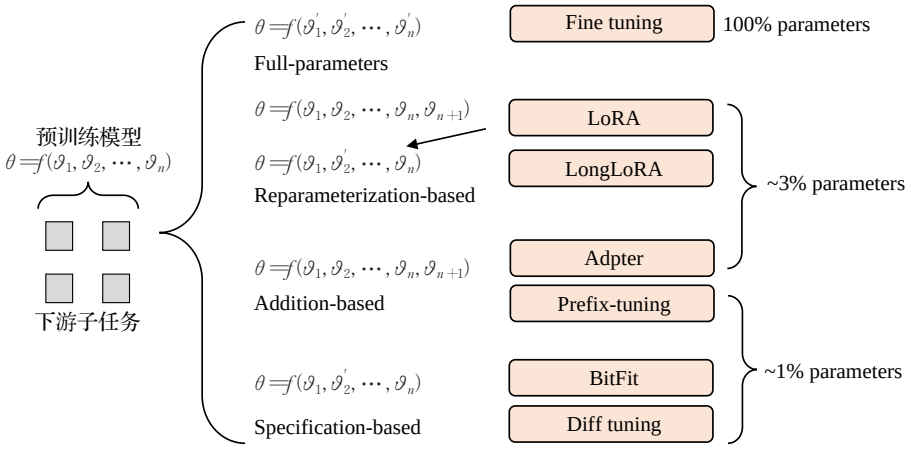

图3　PEFT代表方法与参数量

Fig.3　Representative methods and parameter quantities of PEFT

被提出,高效参数微调(parameter-efficient fine-tuning,PEFT)得到了推广与发展。

相较于传统的全参数微调修改了所有的参数,研究人员探索发现了不同的高效参数方式。目前一种主流的观点将高效参数微调分为三类:增加式(addition-based)、选取式(specification-based)和重参数化(reparameterization-based)形式。这三种方法虽然思路上不尽相同,但结果都是通过少量的参数修改,达到基于下游子任务精细化微调的目的,并在参数为数十亿的大语言模型上能取得相对较好的效果。如图3展示了高效参数微调的代表方法与对应参数量,本章将分别介绍这三种高效参数微调方式的原理、代表模型以及其应用。

3.1　增加式微调

第一个参数高效的微调方法被认为是适应微调adapter-tuning,该方法采用增加额外参数的方法。研究人员经过实验证明,在应用于下游子任务的过程中,如果进行全参数微调,仅仅需要进行一部分关键参数的修改,就可以达到和全参数微调一样的效果[106],原理架构如图4所示。

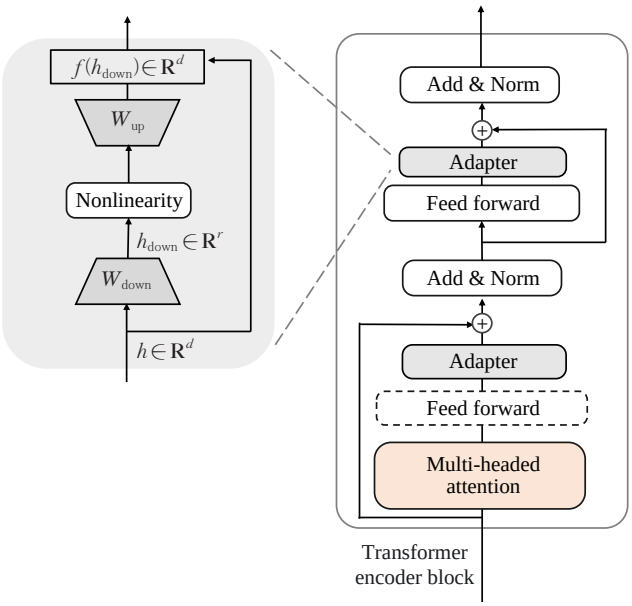

图4　Adapter原理架构图

Fig.4　Principle architecture diagram of Adapter

为了获取到关键参数,Houlsby等人[106]在预训练模型的每一层添加了适配器模块(adapter),在微调时冻结模型的主体参数,只对新增的adapter结构和layer norm层进行微调。在adapter模块中,输入功能的前馈子层为一个向下的投影层,将d维特征h映射到r维特征($r<h$),得到向下投影$W_{down}\in\mathbf{R}^{d\times r}$,通过一个非线性层$f(\cdot)$后,再使用一个前馈子层,将向上投影层将低维特征映射回原来的高维特征$W_{up}\in\mathbf{R}^{r\times d}$,最终实现参数高效修改。

$$h\leftarrow h+f(h_{down})W_{up} \tag{3}$$

由于该模型通过引入一部分额外参数达到对整体的修改,为了尽可能少地引入外界的参数,保证模型的高效性,还设计了一个跳跃链接(skip-connection)结构,确保其至少能够保持微调前的效果。

在BERT大语言模型上进行实验后,研究人员发现仅仅只需要额外的3.4%的参数即可达成全参数微调的效果(大模型评估指标GLUE[107]差距在0.4%内),而全参数微调需要变更100%的参数,整体上高效很多。

Li等人[58]受到prompt的概念的启发,提出另外一种基于增加参数的微调方法前缀调优prefix-tuning,相较于增加式adapter-based抽象降维之后恢复维度的方式,这种微调采用添加前缀参数与修改前缀参数的形式,即在输入词元tokens之前,构造一段连续的且与任务相关的虚拟词元prefix,在模型训练的过程中只对特定任务的prefix进行修改。具体来说,构建了两组prefix向量$\boldsymbol{P}_k,\boldsymbol{P}_v\in\mathbf{R}^{l\times d}$,与Transformer结构中的每一层的原始键$K$和值$V$相连接,然后对构建的前缀和值进行多头注意力计算:

$$\begin{aligned}head_i=Attn(xW_q^{(i)},&concat(P_k^{(i)},CW_k^{(i)}),\\&concat(P_v^{(i)},CW_v^{i}))\end{aligned} \tag{4}$$

随后,采用GPT-2模型进行表格文字生成实验,

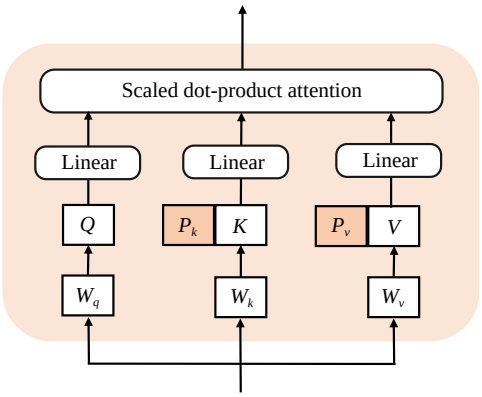

图5 Prefix-tuning原理架构图

Fig.5 Principle architecture diagram of prefix-tuning

BERT模型进行文本总的实验。实验结果表明,仅仅需要修改0.1%的关键参数,prefix-tuning就可以达到与全参数微调相似的效果,其原理结构图如图5所示。

为了解决下游多任务处理中精度不够和精度够但不允许同批次多任务处理的问题,来自UNC的Haokun等,提出了一种名为infused adapter by inhibiting and amplifying inner activations(IA3)的新型PEFT方法[64],模型架构如图6所示。该方法抑制或放大模型的一些激活层,通过学习向量来扩展预训练语言模型的性能,同时减少新参数的数量。IA3引入了三个学习向量,l_k和l_v向量分别用于调整注意力机制中的键和值,而l_{ff}向量用于调整位置感知的前馈网络中的内部激活。通过逐元素乘法来调整模型的内部计算,从而实现对模型行为的微调。通过这种方式,IA3模型可以在保持模型整

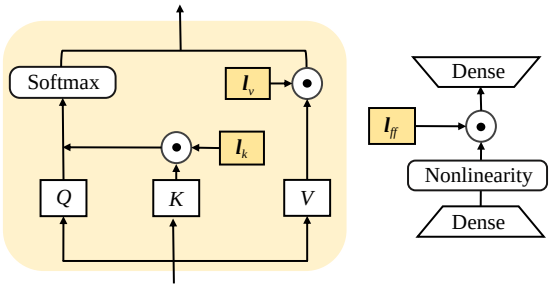

图6 IA3原理架构图

Fig.6 Principle architecture diagram of IA3

体功能不变的情况下,对特定任务的关键部分进行微调。采用这一方法之后整体仅仅需要修改0.01%的参数,即可在分类任务T0-3B中达到65%的准确率。

之后相关研究中,He等人[108]指出并在多个下游子任务上进行实验并验证,相较于传统的大模型全参数微调,基于adapter微调所修改的参数量小,故能够减缓微调后的遗忘。他们同时也发现adapter-based的微调在实验中的表现为不易过拟合,且对于学习率的变化不会过于敏感[108]。

3.2 选取式微调

在3.1节中,需要额外新增参数的方法,但是这样新增也会对整体的模型训练速度造成影响,由此,研究人员提出了一种通过选择模型的重要参数进行微调的方式。

为了保证整体的参数量较小,Zaken等人[60]提出一种进行微调的BitFit思路,该方法仅考虑网络的偏置,即在每个卷积层中,保持权重矩阵$\boldsymbol{W}$不变,仅仅优化偏置向量$\boldsymbol{b}$。BitFit仅更新模型参数的约0.05%。该模型经过实验表明BERT模型在低数据和中等数据情况下(小于10亿个参数)中实现了类似fine-tuning或更好的性能。但同时其也指出,该模型在较大数据量下可能表现得不如预期。

Guo等人[61]提出了一种修改参数的微调方法Diff-pruning,将微调表述为学习一个差异向量$\boldsymbol{\delta}_\tau$,该向量被添加到预先训练的固定模型参数θ_{pre}中:

$$\boldsymbol{\theta}_T = \theta_{pre} + \boldsymbol{\delta}_\tau \tag{5}$$

由于存储预训练参数$\boldsymbol{\theta}$的成本是跨任务平摊的,而新任务的唯一边际成本是差异向量。故当差异向量$\boldsymbol{\delta}_\tau$被正则化,有$\left\|\boldsymbol{\delta}_\tau\right\|_0 \leqslant \|\boldsymbol{\theta}\|_0$,则参数的微调更为高效。具体可被表示为:

$$\min_{\boldsymbol{\delta}_\tau} L(f(X_i;\theta_{pre}+\boldsymbol{\delta}_\tau)) + \lambda R(\theta_{pre}+\boldsymbol{\delta}_\tau) \tag{6}$$

$$R(\boldsymbol{\theta}+\boldsymbol{\delta}_\tau) = \left\|\boldsymbol{\delta}_\tau\right\|_0 = \sum_{i=1}^{d} 1\ \{\boldsymbol{\delta}_{\tau,i} \neq 0\} \tag{7}$$

在GLUE基准测试中,Diff-pruning可以与微调基线的性能相媲美,而相同的性能下每个任务只需修改0.5%的预训练模型参数。

香港中文大学Yang等人[109]提出了一种通过操作隐藏层状态的微调方式,提出先前的研究都是通过对预训练模型中的隐藏层信息进行提取抽象的,但是却没有人尝试过直接使用这一隐藏层信息,因此可以使用引入的向量整合所有层的隐藏状态,和将集成的隐藏状态输入到用于预测类别的特定于任务的线性分类器。尽管方法思路上简单,但是实验中可以得知,该方法能够在更为短的时间内,达到和提示微调(prompt tuning)一致的效果。

3.3 重参数化微调

在增加参数和选择参数之外,还有一种重参数化的方法,相当于增加参数和选择参数的结合,即首先增加参数进行微调,之后再将增加的参数融合至现有模型。由于没有额外的参数不会产生额外的推理成本,融合的这一过程也保证了模型的灵活性。其中,最具代表性的是LoRA(low-rank adaption),本节其余微调方法也是基于LoRA的一些变形。

2018年,研究人员认为[63]对于一个参数量为D的模型,训练该模型,也就意味着在D维空间上寻找有效的解。文章认为D个参数中,可能存在大量的冗余参数,可能实际上只需要优化其中的d个参数就可以找到一个有效的解,这些参数被称为内在秩(intrinsic rank)。

受到预训练模型内在维度[51]与参数内在秩的启发，假设权重的更新在微调适配过程中也具有较低的内在秩，那么即可通过仅仅修改内在秩，简化现有的参数。Hu等人[62]提出了内在秩适配器LoRA用来微调，对预训练模型的权重矩阵 $\boldsymbol{W}_0 = \mathbf{R}^{d\times k}$，通过低秩分解(low-rank decomposition)[59]来表示约束其更新。

$$\boldsymbol{W}_0 + \Delta\boldsymbol{W} = \boldsymbol{W}_0 + \boldsymbol{BA} \tag{8}$$

其中，$\boldsymbol{B}\in\mathbf{R}^{d\times r}$，$\boldsymbol{A}\in\mathbf{R}^{d\times r}$，$r << \min(d,k)$。训练过程，$\boldsymbol{W}_0$被固定不再进行梯度更新，只训练$\boldsymbol{A}$和$\boldsymbol{B}$，如下所示。对于输入$x$，模型的前向传播过程被更新为：

$$h = \boldsymbol{W}_0 x + \Delta\boldsymbol{W}x = \boldsymbol{W}_0 x + \boldsymbol{BA}x \tag{9}$$

之后，研究人员在RoBERTa、DeBERTa[110]、GPT-2和GPT-3 175B等大模型上进行实验。实验结果也验证了这一个微调方法的有效性，LoRA也是最广泛使用的大语言模型微调方法之一。

在LoRA的基础上，CUHK和MIT的联合团队研发了LongLoRa[65]，提出了一种可转换的短注意力机制(shift short attention)，增加了可处理的窗口的词元token数量。该方法采用分组的形式，在每组内完成self-attention的运算，从而降低整体的运算量。LongLoRa将读取的窗口数量增加了8倍，token数量由从4×10^3提升到32×10^3，实现了将大量的信息通过提示prompt进行传入，大大提高了模型的可用性与下游任务的整体模型效果。

随后，Chavan等人[66]基于LoRA提出了一种更加通用化的微调方法GLoRA(generalized LoRA)。该方法通过可扩展、模块化、逐层结构搜索来学习每一层的单个适配器，在权重和激活状态上增加额外的维度来适应新任务，提供了更大的灵活性和更强的性能。

$$f(x) = (W_0 + W_0\boldsymbol{A} + \boldsymbol{B})x + \boldsymbol{C}W_0 + \boldsymbol{D}b_0 + \boldsymbol{E} + b_0 \tag{10}$$

其中，$\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$、$\boldsymbol{E}$是GLoRA中下游任务的可训练支持张量，W_0和b_0在整个微调过程中保持冻结状态。

综合实验表明，在VTAB-1K[111]基准下，GLoRA于多个细化指标包括自然Caltech-101[112]等六种、专业EuroSAT[113]等三种和结构化基准测试Clevr-Count[114]等五种中表现为最优，在各种数据集上以更少的参数和计算实现更高的准确性。

传统的adapter-tuning方法存在了推理延时的问题，而prefix-tuning或prompt tuning直接优化prefix和prompt的方式是非单调的，难以收敛，并且会消耗输入的token。为了克服这些问题，Zhang等人[67]提出了一种名为适应性的内在秩适配器(adaptive low rank adaptor，AdaLoRA)的改进方法。AdaLoRA是对LoRA的一种改进，它根据重要性评分动态分配参数预算给权重矩阵。该方法包括调整增量矩阵分配以及对增量更新进行参数化的奇异值分解等策略。

$$W = W^{(0)} + \Delta = W^{(0)} + \boldsymbol{P\Lambda Q} \tag{11}$$

其中，$\boldsymbol{P}\in\mathbf{R}^{d_1\times r}$，$\boldsymbol{Q}\in\mathbf{R}^{r\times d_2}$，表示$\Delta$的左/右奇异向量；对角矩阵$\boldsymbol{\Lambda}\in\mathbf{R}^{r\times r}$。

表2 基于LoRA的重参数微调

Table 2 LoRA-based reparameterization fine-tuning

LoRA系列	创新点	优势
LoRA	引入低秩矩阵	参数量小；缩短训练时间
LongLoRA	移位稀疏注意力机制；优化参数更新计算过程	减少GPU内存消耗；缩短训练时间
GLoRA	引入了门控机制；动态调整低秩更新	更好的模型控制；更好的模型适应性
AdaLoRA	自适应性调整低秩更新	增强了模型在不同任务和数据集上的性能

表2对比了这些改进方法的创新点与优势，通过这些改进，AdaLoRA能够更有效地优化模型的参数，提高下游任务的性能。

3.4 相关性与比较

经过对多篇文章数据分析，参考Houlsby等人[106]、Guo等人[61]、Zaken等人[60]和Ding等人[14]相关研究，整理了较具代表性的高效参数微调方法，在MNLI数据集[115]和SST2数据集[116]上微调$\text{BERT}_{\text{large}}$(BL)、$\text{RoBERTa}_{\text{base}}$(RB)，对测试集的正确率数据，如表3所示。

表3 PEFT代表方法实验数据对比

Table 3 Experimental data comparison of representative methods of PEFT

单位：%

微调方法		参数量	SST2	MNLI
BL	Full-FT	100.00	93.50	86.50
	Diff-Prune	0.50	94.20	86.40
	BitFit	0.08	94.20	84.50
	Adapter	3.60	94.00	84.90
RB	Full-FT	100.00	94.20	86.40
	BitFit	0.09	93.70	84.80
	Adapter	0.50	87.20	94.20
	Prefix	0.50	86.30	94.00
	LoRA	0.50	87.20	94.20

现有的不同PEFT方法之间存在着一定的联系，近年来也有学者在寻找各类PEFT方法中的共性，并进行归纳总结。CMU学者He等人[108]认为，现有的多种PEFT方法，包括adapter、prompt tuning、LoRA都有着一定的共性[117]。为预训练模型中添加或者调整特定的隐层状态，只是在设计的参数维度、修改函数上有一定的不同。具体来说，把它们看作是学习一个向量$\Delta\boldsymbol{h}$，应用于各种隐藏表征。形式上，把要直接修改的隐藏表征表示为h，把计算h的PLM子模块的直接输入使用与$\Delta\boldsymbol{h}$相关的表示，通过重排这种抽象的组合形式，进一步提出了三种PEFT概念，组合得到了parallel adapter、multi-head paral-

lel adapter、scaled parallel adapter三种变体。实验测试得到最终的方法距离fine-tuning在ROUGE-2标准下仅有0.04%的差距。

Ding等人[14]在综述中将这一类型的PEFT方法归类为Delta-tuning。文章认为对于预训练模型$\theta=f(\vartheta_1,\vartheta_2,\cdots,\vartheta_n)$,微调后,有$\theta'=f(\vartheta'_1,\vartheta'_2,\cdots,\vartheta'_n)$,模型前后的差距为$\Delta\theta=\theta'-\theta$,故称之为Delta-tuning。

现如今的不少优化方法已经能够将修改的参数量控制在整体的0.1%甚至0.01%以内,但是单纯地追求参数高效性也导致了模型灵活性降低,甚至丢失预训练模型的基础能力。由此,如何在参数高效和性能之间追求平衡成为学术界亟待解决的问题。

4 提示微调

为了解决语义差异(bridge the gap between pre-train and fine-tuning)和过拟合问题(overfitting of the head),在微调的同时,提高大语言模型在特定任务适应性,提示调优(prompt tuning)方法被提出并得到了发展,并演变成了"预训练-提示-预测"的NLP范式[13]。提示微调着力于调整模型的提示或标记,而保持底层语言模型参数不变,以提高模型的性能,所需的计算资源和存储空间相较于fine-tuning也大大减少。

在得到预训练模型后,记θ为模型的参数,Y为一个表示类标签的标记序列,X为一系列token,则模型的输出为$P(Y|[X,\theta])$。提示微调就是在模型生成Y的过程中,通过添加预先标记P(prompt)到输入X中,使模型得到正确的Y的可能性$\Pr_\theta(y_i|[P;X,\theta])$,$y_i\in Y$最大化,同时保证模型的(部分)参数$\theta$不变。

其算法如图7所示,主要分为构建模板(template construction)、标签词映射(label word verbalizer)和答案设计(answer mapping)三部分[81]。模板是通过人工或自动学习方法,构建的含有掩码Mask标记的输入提示。模板主要分为离散模板(又称为硬提示,由具体的字符表示)[82-83]和连续模板(又称为软提示,由自由向量表示)两种[84-86]。在标签词映射阶段,模型处理结合了模板的原始文本,以预测各个词元(token)的概率。这一步骤的关键是建立模型输出和下游任务真实标签之间的映射关系,确保模型能够准确地识别和分类信息。答案设计阶段是实际的训练过程,涉及将构建好的

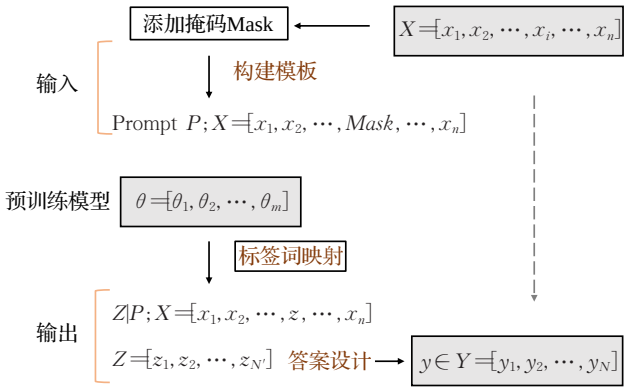

图7 Prompt-tuning原理图

Fig.7 Principle diagram of prompt-tuning

prompt输入到语言模型中,并根据模型的输出进行调整和优化。这个过程不仅包括输入提示,还涉及到基于模型预测的输出进行微调,以便更好地适应特定的任务,并将模型输出映射回人类可理解的类别标签。

通过这三个阶段的协同工作,prompt tuning能够有效地调整预训练语言模型,以适应特定的下游任务,从而提高模型在特定任务上的表现。

在指令微调中,根据指令Prompt的不同,发展出许多性能出色的微调技术。在大语言模型的微调方面,继few-shot(FS)、one-shot(1S)、zero-shot(0S)等迁移学习方法后,又发展了上下文学习in-context learning、思维链chain-of-thought等提示微调代表方法。在此基础上指令微调instruction-tuning也得到了极大的发展;在较小规模的语言模型上,PET、P-tuning和prefix-tuning也被先后提出。如图8所示,本章主要介绍上下文学习

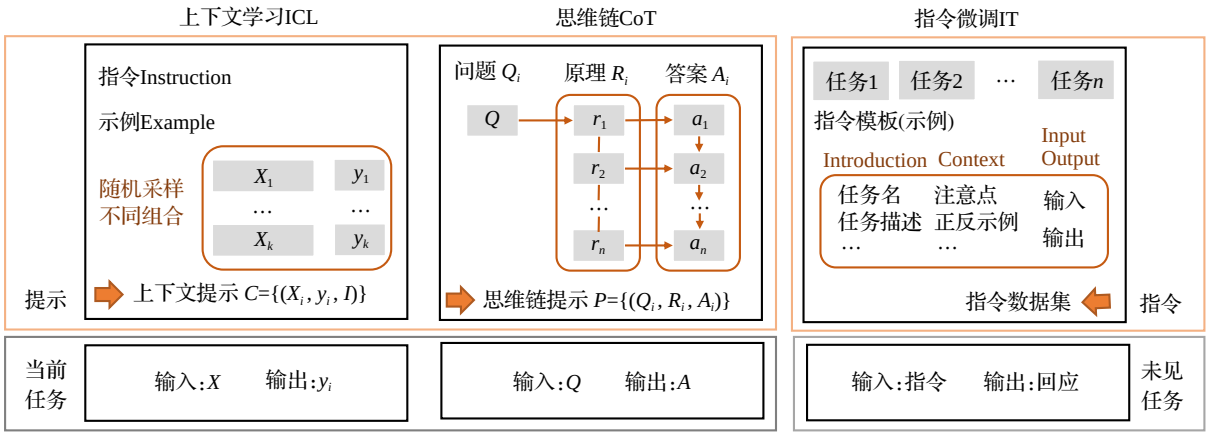

图8 ICL、CoT与IT原理对比图

Fig.8 Principle comparison diagram of ICL, CoT and IT

in-context learning(ICL)、思维链技术 chain-of-thought(CoT)和指令微调 instruction-tuning(IT)。

4.1 上下文学习

GPT-3 系列最早正式提出上下文学习(ICL)的概念,是 prompt tuning 的发展前期的一个重要节点。ICL 的本质思想是类比学习[68],通过从训练集中挑选一些样本作为具体任务时的提示,来避免参数更新。这是一种离散调优的方法,证明了在低资源场景下非常有效。

对于 $P(y_i|[X,\theta])$ 的求解,ICL 事先给出了演示集 C,作为 ICL 的"prompt"。C 包含了指令 I 和 k 个实例,通常是这 k 个实例的线性拼接,称作上下文模板,可以表示为:$C=\{s(X_1,y_1,I),s(X_2,y_2,I),\cdots,s(X_k,y_k,I))\}$。其中,$s(x_i,y_i,I)$ 是按照任务指令 I,用自然语言文本编写的例子,又称为 in-context examples。则最终结果可表示为:

$$P(y_i|[X,\theta])\triangleq f_{PLM}(y_i,C,X,\theta) \tag{12}$$

$$\hat{y}=\arg\max_{y_i\in Y}P(y_i|[X,\theta]) \tag{13}$$

ICL 的独特之处在于:相比于 fine-tuning 基于梯度更新模型参数,prompt learning 中有部分软提示(soft prompt)方法需要微调参数,ICL 则不需要对模型参数更新。同时,不同于 fine-tuning 需在大量训练数据中的学习类别表示,ICL 采用"实例-标签"形式,使用下游任务的演示信息学习并推理,从而减少所需资源成本。

上下文学习自出现以来,一直是该领域的研究重点。针对上下文学习较优的性能与微调效果,Razeghi 等人[69]研究发现,ICL 的性能与预训练数据中的术语频度呈现较高的相关性,探究了为什么上下文学习的性能优于零射击 zero-shot;Xie 等人[70]通过理论分析与演示程序,认为 ICL 可以被转化为贝叶斯推理。针对 ICL 的发展,Liu 等人[71]对 ICL 的发展、技术要点进行了详细的综述。近些年,也有大量学者针对演示集合 C 的示例标记、ICL 的问题表述、ICL 的新发展与弊端进行了广泛研究[72-73]。

4.2 思维链

针对大语言模型不能很好地解决数学推理等逻辑性较强的问题的现象,Wei 等人[74]提出了思维链(CoT)的微调技术,将模型的输出视作一个序列,各个部分相互独立。通过提供给大模型关于问题的一些演示输入、过程与输出的思维范例提示,模型通过将先前的输出作为后续输入的一部分来迭代地生成这些部分,使得模型在一定程度上模拟人类解决问题的过程。

思维链技术 CoT 的 prompt 提示,是由输入、思维链和输出组成的一个三元组(*input*,*chain of thought*,*output*)。其中,输入可以记作问题示例 Q;思维链是得到最终输出结果的这个过程中,一系列中间自然语言推理步骤,记作原理 R;而输出则是期望答案 A。记提示 prompt 为 $P=\{(Q_1,A_1,R_1),(Q_2,A_2,R_2),\cdots,(Q_k,A_k,R_k)\}$。由于给出了推理过程,所以在可以认为在不同的推理阶段,有不同的期望答案,即 $A=[a_1,a_2,\cdots,a_n]$,故有:

$$P(A|[Q,P,\theta])=\prod_{i=1}^{|A|}f_{\mathrm{PLM}}(a_i|[Q,P,\theta]) \tag{14}$$

根据贝叶斯公式,可得:

$$P(A|[Q,P])=P(A|[Q,R,P])P(R|[P,Q]) \tag{15}$$

$$P(R|[P,Q])=\prod_{i=1}^{|R|}f_{\mathrm{PLM}}(r_i|[Q,P,r_{<i}]) \tag{16}$$

$$P(A|[Q,P,R])=\prod_{j=1}^{|A|}f_{\mathrm{PLM}}(a_i|[Q,P,R,a_{<j}]) \tag{17}$$

其中,a_i 表示第 i 个过程的预期输出,$|A|$ 则表示总过程数目。由此可见,提高 A 和 R 的在每一个阶段的发生概率,也就提高了思维链的推理过程的性能[75]。

为了继续挖掘大语言模型在数学推理方面的潜力,研究人员进行了大量思维链相关的工作,如表 4 所示。最初的 CoT 用自然语言来描述推理过程;后续,诸多研究人员着重于对推理过程的提示进行修改,发展产生了一系列 XoT 的思维链技术,如链结构 PoT[76]、树结构 ToT、SoT[77-78]、图结构 GoT[79-80]等。除此之外,思维链的增强技术也受到了较大的关注,Qiao 等人[75]将其分为策略增强推理和知识增强推理并进行了详细的介绍。

表 4 XoT 微调技术简述

Table 4 Introduction to XoT fine-tuning technology

思维链技术	介绍
CoT	引入解释性的中间步骤,在模型内部完成推理计算
ToT	引进额外结构,构建树状思维过程,允许回溯
SoT	首先生成思维"骨架",在此基础上填充
GoT	使用图形结构,顶点和边表示信息单元和依赖关系
PoT	产生程序语句,使用程序解释器得到推理结果

4.3 提示微调与指令微调

指令微调(IT)也可以被认为是有监督微调的另一种特殊形式,与 prompt tuning 具有一定的异同。最早是由 Wei 等人[74]于 2022 年 5 月提出的一种通过用指令描述的一种数据集来微调语言模型的方法,可以大大提高语言模型在新任务上的 zero-shot 能力,并得到了 fine-tuned language net(FLAN)方法,主要涉及到指令理解、指令数据获取、指令对齐等内容。

指令是一种相较于提示更为详细的文本,指令数据集是指令微调的核心关键。指令数据集的每一个实例都有三个元素组成:指令 instruction、上下文补充信息 context(可选)和基于指令的输入输出。指令一般包括对于任务的名称、描述、注意点、正反示例等详细的信息。

通常可以采用人工集成转换(如 FLAN、P3 数据集)[87-88]或者大语言模型生成(如 InstructWild、self-instruct)[89-90]的方式,得到具有"指令,输出"(instruction,output)格式的数据集[10]。得到了指令数据集后,便可以在预训练模

型上进行微调,多采用监督微调fine-tuning的方式,使得模型学习如何通过指令做预测。最后,在一个新的任务上衡量经过指令微调的模型性能。

相较于传统提示微调专注于适应某一任务,指令微调通过激发语言模型的理解能力,更充分利用先验知识,在给出更明显的指令后让模型去理解并正确地回应,泛化到多种任务。同时,instruction tuning可能涉及对模型参数的更广泛调整,以便更好地适应特定的指令集。

基于指令微调,许多预训练大语言模型得到了更为广泛应用与优良的性能,如Alpaca[35]、BLOOMZ[91]、Vicuna[92]、Instruction-GPT[97]等。文献[69]重点研究了指令数据集的构造,力求在较小的数据集上实现效果最好的微调,经过实验研究提出了多种构造指令数据集的重构技术。也有很多研究人员根据不同的任务构造出了多模态数据集,如用于二维图像和三维点云的LAMM数据集[94]、多模态指令调优数据集MUL-TIINSTRUCT等等[95],为大模型经过指令微调在如医疗、教育等具体场景中的对话、写作、情感识别等问题的解决提供了条件。Zhang等人[10]发表了最新关于指令微调的综述文章。目前,指令微调相关方向的研究成果较少,仍是一个较新且具有极大发展潜力的研究方向。

5 强化学习微调

5.1 强化学习微调前期发展

早在2015年,强化学习(reinforcement learning,RL)就在解决大型问题时展示了优越的性能[96],但是由于许多任务的目标非常复杂且没有清晰的定义,强化学习系统的目标和人类的偏好无法真正取得一致,这也让深度强化学习可应用的范围受到了严重限制。2017年,OpenAI的研究团队提出了一个强化学习算法,从人类反馈中学习奖励函数,并对此奖励函数进行优化。这种方法不是第一次被提出,但却是首次被扩展到现代深度强化学习领域。这项研究降低了强化学习算法的复杂性,推动了深度强化学习在复杂的现实任务上的实际应用[118]。

2017年8月,OpenAI团队又提出了近端策略优化(proximal policy optimization,PPO)的方法[99],通过收集人类评估来微调模型。在每次迭代中,N个并行参与者中的每位都会收集T个时间步的数据。然后,在这NT个时间步的数据上构建替代损失,并在K个历时周期内使用小批量梯度下降法(SGD)对其进行优化,或者用自适应动量的随机优化方法(adaptive moment estimation,ADAM)获取更佳性能[119]。其中,每次迭代通过使用随机梯度上升方法优化如下目标:

$$L_t^{\mathrm{CLIP+VF+S}}(\theta)=\hat{\mathrm{E}}_t[L_t^{\mathrm{CLIP}}(\theta)-c_1L_t^{\mathrm{VF}}(\theta)+c_2S[\pi_\theta](s_t)] \tag{18}$$

其中,c_1和c_2是系数,S表示熵奖励(entropy bonus)平方误差损失$L_t^{\mathrm{VF}}=(V_\theta(s_t)-V_{\mathrm{target}})^2$。

经过实验,该方法被验证具有信赖域算法的稳定性和可靠性,且更易实施,具有更好的泛化性能和整体性能。2020年,OpenAI团队使用了上述的算法,在四项NLP任务中演示了语言模型的强化学习微调,提高了GPT-2模型处理自然语言任务的性能。这也是强化学习算法首次应用在大语言模型的微调上,由于其良好结果,逐渐成为大语言模型在性能优化微调的重要发展方向。不过研究者也提到,在有监督的情况下,NLP模型是使用人类数据进行训练的,所以如果要进行强化学习微调,也需要人类数据[120]。

5.2 基于人类反馈的强化学习微调

基于人类反馈的三阶段强化学习算法(reinforcement learning from human feedback,RLHF)起源于经济学中的揭示偏好理论,并在早期被机器学习领域应用于人机交互和强化学习等。现在使用的RLHF的标准方法是在2017年由Christiano等人[118]推广的,在引导深度强化学习社区的注意力转向基于反馈的方法方面发挥了关键作用[121]。针对大语言模型微调的RLHF算法,由OpenAI团队正式提出在2022年1月正式提出[97]。

为了对GPT-3的监督模型进行进一步微调,使其符合人类偏好并遵循各类书面指令,研究者使用了RLHF算法,并使用在了InstructGPT模型的研究中。RLHF系统的主要步骤可以分为预训练与监督微调模型、训练奖励模型和使用强化学习微调模型。

RLHF系统从监督微调(SFT)开始,首先使用监督学习方法在标注器演示中对需要对齐的语言模型进行微调,例如InstructGPT模型就是在GPT-3的基础上进行微调。随后,利用人工标注器获得的人工反馈训练奖励模型(reward model,RM),人工标注器根据模型输出与预期行为的一致性对模型输出进行排序,反映了人类对语言模型生成的文本的偏好。最后,选择合适的强化学习算法与训练后的奖励模型结合使用,现有的工作通常会选择PPO算法。PPO算法用于根据收到的人类反馈进一步微调语言模型,从而使模型更符合人类的偏好,提高模型的指令跟随能力[98,105]。

RLHF系统使用人类偏好作为奖励信号来指导语言模型的微调,使模型在与用户的互动中更诚实无害、帮助性更强,提供了使语言模型与人类意图和偏好相一致的一种新方法。

由于其对模型性能提升的有效性,RLHF算法得到了广泛应用。除了InstructGPT模型之外,GPT-4模型通过在RLHF训练中添加一个额外的安全奖励信号,提高了模型响应恶意信息的能力;LLaMA 2对RLHF训练进行了更多迭代尝试与评估,大幅提高了模型的有用性和安全性[122]。Google公司的Bard模型与Anthropic的Claude系列模型(https://claude.ai/login?returnTo=%2F)

也采用了RLHF的微调算法。北京大学的研究团队于2023年6月发布了基于RLHF的Beaver项目,这是目前首个可复现的RLHF基准项目,首次公开了RLHF所需的数据集、训练和验证代码(https://github.com/PKU-Alignment/safe-rlhf)。Casper等人[123]对RLHF和相关方法的开放问题和基本局限性进行了分析探讨,并总结概述了在实践中改进与优化RLHF的技术。

5.3　基于人工智能反馈的强化学习微调

随着人工智能的发展,研究者们尝试让它们帮助监督其他人工智能。Bai等人[100]提出了利用人工智能获得反馈而进行微调训练的方法(reinforcement learning from AI feedback,RLAIF),并使用在了Claude系列模型中,通过自我完善来训练无害的人工智能助手,从而实现有害输出的识别,且不需要任何人类标签。

RLAIF算法过程包括监督学习和强化学习两个阶段。在监督学习阶段,从初始模型中采样,进行自我批评和修正,再根据修正后的反应对原始模型进行微调。在强化学习阶段,从微调后的模型中取两个样本,使用另一个AI模型来评估样本,选择出更好的样本。利用此偏好样本,构成AI偏好数据集,用数据集训练一个偏好模型,作为进行强化学习训练的奖励信号。

RLHF和RLAIF的主要思路与对比,如图9所示。

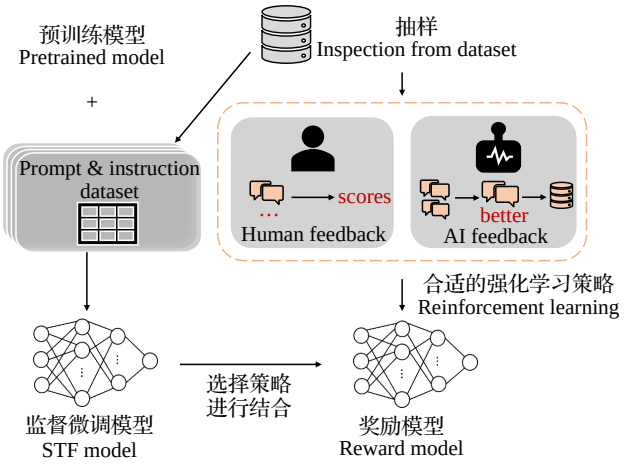

图9　RLHF和RLAIF原理图

Fig.9　Principle diagram of RLHF and RLAIF

Lee等人[101]对比了RLAIF和RLHF的方法,在大量实验中通过对比标签对齐度、配对准确度和胜率,证明RLAIF在总结、有用对话生成和无害对话生成等任务中,均可达到与RLHF同等甚至较优的效果。证实了RLAIF可以产生人类水平的反馈,说明RLAIF可以更精确地控制人工智能的行为,为解决收集高质量的人类偏好标签这一难题提供了新思路[101]。

6　展望与总结

6.1　未来研究方向

随着人工智能领域的迅速进展,大语言模型的微调面临着一系列挑战与难题。目前,研究重点主要聚焦于提高微调过程的资源利用效率,并提升在特定任务上微调后模型的性能。总结前文,本文所讨论的主要微调技术的未来发展的重要方向如图10所示。此外,无监督微调、多模态大模型的微调以及模型的伦理性与安全性的提升,也被视为未来的重要发展方向。

特定任务上的性能　大语言模型的应用往往是为了解决特定的实际问题或任务。自然语言处理领域涉及的任务种类繁多,如情感分析、命名实体识别、语义理解等,也包括在如医学、金融、法律等方面的应用。每种任务都有其特定的数据分布和特征,通过重点提高特定任务上的微调性能,可以使模型更好地适应不同类型的任务,并提高应用的通用性和灵活性。因此,优化模型在特定任务上的微调性能是确保模型在实际应用中发挥作用的关键,包括针对特定任务优化的模型架构、损失函数和学习策略,从而提高微调过程中的性能和效率。

计算资源优化　随着模型规模的扩大,如何实现资源有效和高效的微调是首要的问题,特别是在计算和数据资源有限的场景下,开发节省资源且效果优良的微调策略将是未来研究不变重点,在节约计算资源、加速模型部署、降低成本、推动研究进展方面均具有重要意义[102]。

无监督微调　相比于传统的监督学习方法,无监督微调具有更大的灵活性、更高的数据销量和训练能力,同时也能够更好地适应各种领域和任务。目前,无监督

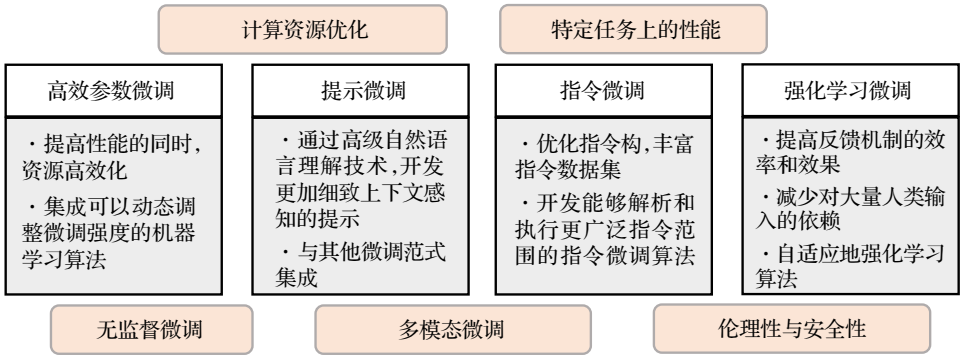

图10　微调的重要发展方向

Fig.10　Key development direction in fine-tuning

技术以及自监督学习、弱监督学习和对比学习等技术,仍旧拥有较大的发展空间。尤其是如何在没有标记数据的情况下优化模型性能,可以帮助模型从未标记的数据中学习并提高微调的效率和有效性。

多模态大模型微调 随着LLMs的不断发展,将多模态数据(文本、图像、音频等)融入微调过程的需求日益增加,通用的、跨领域的微调策略,实现模型参数和结构的最大复用将成为研究的焦点。这种整合可以显著提高模型理解和生成不同模态内容的能力,对于视觉问题回答和互动媒体生成等任务至关重要。对多模态数据的微调,致力于开发能够无缝整合多模态数据的微调方法,增强模型跨各种领域和感官输入处理和综合信息的能力,也是未来的重要研究方向之一。

伦理性与安全性 随着LLMs越来越多地应用于现实世界,确保它们以安全、公平与可靠的方式运行至关重要。在微调的过程中会使用到大量的数据,可能包含用户的个人信息或敏感数据,存在偏见或歧视性内容。在微调过程与模型结果输出时,如何保障数据的隐私与安全性,确保在不同用户群体中是公正、无偏见和平等的,以减少模型应用过程中的安全风险,也受到了众多研究机构与学者的注意。

不仅如此,模型的可解释性和透明性正受到越来越多的关注,这些大型模型的内部工作机制尚属于“黑箱”,在某种程度上限制了其在某些敏感领域的应用与研究[103]。

6.2 总结

随着大型语言模型的不断发展和广泛应用,微调预训练的语言模型以适应特定任务已成为研究的热门话题,大型语言模型的未来发展在很大程度上受到创新微调方法的影响。最初,微调是一种通过调整模型所有参数来使预训练模型适应特定任务的过程,而使用在大语言模型上会导致高昂的计算成本。在全参数微调等传统微调基础的基础上,学者们针对微调的资源优化与性能提升,发展了多种微调方法。参数高效微调通过只修改模型参数的小部分显著减少了计算负载,从而维持甚至提升了性能效率。提示调整作为一种多功能的微调方法,巧妙地使用提示或指示性标记来引导模型的响应,能使模型能够通过对参数的最小更改来适应特定任务,在特定任务学习和迁移学习环境中都显示出了有效性。指令微调使得模型对于语义与任务的理解更加细致,上下文感知、逻辑推理能力更为突出,指导LLMs更好地处理具有明确指令的多样化任务,响应复杂指令。强化学习与人类反馈在微调过程中的整合,使LLMs输出与人类偏好更加一致,在细化模型行为以符合人类判断和偏好方面具有理想的发展前景。

本文对大语言模型的微调技术,分析介绍了自大语言模型问世以来的多类微调方法,总结了其原理、发展与应用,并进行了相应的对比研究,如表5所示。

大型语言模型的微调是一个动态发展的领域,需要进行持续的研究,以推进微调方法学并解决计算效率、特定任务性能、伦理考虑以及多模态整合之间的复杂平衡,未来的发展有望带来更强大、多功能且符合伦理的大语言模型。基于上述的全局性的挑战,每一类微调策

表5 代表性微调技术对比

Table 5 Comparison of representative fine-tuning techniques

微调技术	分类	主要特点	贡献	发展方向	代表方法
全参数微调	监督微调	对模型所有层和参数进行微调,通常为监督微调	在数据与资源充足的情况下可达到几乎所有微调方法中的最优性能	提高微调的数据和计算效率	使用带标签的数据集对BERT、GP等LLM进行微调
	无监督微调				
高效参数微调	增加式	增加少量新参数到预训练模型进行微调	显著降低微调的参数量,减少存储和计算需求	平衡新增参数的规模和性能	adapter-tuning, prefix-tuning, IA3
	选取式	在模型的参数中选取一部分进行微调	进一步减少了所需微调的参数数量,降低了计算复杂性	关键参数的选择	BitFit, diff-pruning
	重参数化	改变模型参数的表达方式进行微调	通过减少参数的自由度,减少微调时的计算量和提高泛化能力	有效地重参数化策略的设计	LoRA, LongLoRa, GLoRA, AdaLoRA
提示微调	上下文学习(ICL)	在模型的输入中提供示例或者上下文,引导模型的预测	使得模型能够在没有显式微调的情况下适应新任务	更有效地使用上下文信息	supervised ICL, self-supervised ICL
	思维链(CoT)	构造一个逻辑性的思维过程,引导模型在推理任务上的输出	帮助模型更好地处理需要复杂推理的任务	更有效的思维链结构的构建	CoT, ToT, GoT, RAP, multi-modal CoT
	指令微调(IT)	通过更为明确详细的指令,调整模型的行为	提高了模型在未见任务上的泛化性	指令数据集的构建与加强模型的理解	指令数据集:P3, Flan2021, LIMA等
强化学习微调	基于人类反馈的强化学习微调	结合人类标注的反馈来微调模型	可以根据复杂的、非直接的人类反馈来改善模型	高效低成本的反馈收集和处理机制	RLHF
	基于人工智能反馈的强化学习微调	结合人工智能标注的反馈来微调模型	能够自我训练获得取代人类的反馈且具有安全性	提高模型的自我纠正能力与安全性	RLAIF

略殊途同归,都在高效性、通用性与可解释性方面进行着不断地探索与研究。相信在未来将有更多的研究投入到此领域,以满足实际应用的不断增长的需求,在大型语言模型的关键研究领域攻坚克难[104]。

参考文献:

[1] YOSINSKI J, CLUNE J, BENGIO Y, et al. How transferable are features in deep neural networks?[C]//Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014: 3320-3328.

[2] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J]. arXiv:1503.02531, 2015.

[3] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training [EB/OL]. [2023-11-23]. https://www.mikecaptain.com/resources/pdf/GPT-1.pdf 2018.

[4] RADFORD A. Language models are unsupervised multitask learners[EB/OL]. [2023-11-23]. http://web.archive.org/web/20190226183542/https:/d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

[5] KAPLAN J, MCCANDLISH S, HENIGHAN T, et al. Scaling laws for neural language models[J]. arXiv:2001.08361, 2020.

[6] BROWN T B, MANN B, RYDER N, et al. Language models are few-shot learners[J]. arXiv:2005.14165, 2020.

[7] OPENAI. GPT-4 technical report[J]. arXiv:2303.08774, 2023.

[8] SMITH L N. Cyclical learning rates for training neural networks[C]//Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017: 464-472.

[9] CHUNG H W, HOU L, LONGPRE S, et al. Scaling instruction-finetuned language models[J]. arXiv:2210.11416, 2022.

[10] ZHANG S, DONG L, LI X, et al. Instruction tuning for large language models: a survey[J]. arXiv:2308.10792, 2023.

[11] HAN X, ZHANG Z, DING N, et al. Pre-trained models: past, present and future[J]. AI Open, 2021, 2: 225-250.

[12] QIU X, SUN T, XU Y, et al. Pre-trained models for natural language processing: a survey[J]. Science China Technological Sciences, 2020, 63(10): 1872-1897.

[13] LIU P, YUAN W, FU J, et al. Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing[J]. arXiv:2107.13586, 2021.

[14] DING N, QIN Y, YANG G, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models[J]. Nature Machine Intelligence, 2023, 5(3): 220-235.

[15] MANNING C, SCHUTZE H. Foundations of statistical natural language processing[M]. Cambridge, Massachusetts: MIT Press, 1999.

[16] ROSENFELD R. Two decades of statistical language modeling: where do we go from here?[J]. Proceedings of the IEEE, 2000, 88(8): 1270-1278.

[17] GAO J, LIN C Y. Introduction to the special issue on statistical language modeling[J]. ACM Transactions on Asian Language Information Processing (TALIP), 2004, 3(2): 87-93.

[18] GOODMAN J T. A bit of progress in language modeling[J]. Computer Speech & Language, 2001, 15(4): 403-434.

[19] BENGIO Y, DUCHARME R, VINCENT P, et al. A neural probabilistic language model[C]//Advances in Neural Information Processing Systems, 2000: 932-938.

[20] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[J]. arXiv:1409.0473, 2014.

[21] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[J]. arXiv:1301.3781, 2013.

[22] MIKOLOV T, KARAFIÁT M, BURGET L, et al. Recurrent neural network based language modeling in meeting recognition[C]//Proceedings of the Annual Conference of the International Speech Communication Association, 2011: 2877-2880.

[23] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]//Advances in Neural Information Processing Systems, 2014: 3104-3112.

[24] DAI A M, LE Q V. Semi-supervised sequence learning[C]//Advances in Neural Information Processing Systems, 2015: 3079-3087.

[25] PETERS M, NEUMANN M, IYYER M, et al. Deep contextualized word representations[J]. arXiv:1802.05365, 2018.

[26] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems, 2017: 5998-6008.

[27] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [J]. arXiv:1810.04805, 2018.

[28] DING N, QIN Y, YANG G, et al. Delta tuning: a comprehensive study of parameter efficient methods for pre-trained language models[J]. arXiv:2203.06904, 2022.

[29] HUANG J, LI C, SUBUDHI K, et al. Few-shot named entity recognition: a comprehensive study[J]. arXiv:2012.14978, 2020.

[30] XIE Q, DAI Z, HOVY E, et al. Unsupervised data augmentation for consistency training[J]. arXiv:1904.12848, 2019.

[31] MCCANN B, BRADBURY J, XIONG C, et al. Learned in translation: contextualized word vectors[C]//Advances in Neural Information Processing Systems, 2017: 6294-6305.

[32] WANG Z, QU Y, CHEN L, et al. Label-aware double transfer learning for cross-specialty medical named entity recognition[J]. arXiv:1802.05365, 2018.

[33] LIU Y, OTT M, GOYAL N, et al. RoBERTa: a robustly optimized bert pretraining approach[J]. arXiv:1907.11692, 2019.

[34] TOUVRON H, LAVRIL T, IZACARD G, et al. LLaMA: open and efficient foundation language models[J]. arXiv:2302.13971, 2023.

[35] TAORI R, GULRAJANI I, ZHANG T, et al. Alpaca: a strong, replicable instruction-following model[J]. Stanford Center for Research on Foundation Models, 2023, 3(6): 7.

[36] DU Z, QIAN Y, LIU X, et al. GLM: general language model pretraining with autoregressive blank infilling[J]. arXiv: 2103.10360, 2021.

[37] SCAO T L, FAN A, AKIKISCAO C, et al. BLOOM: a 176b-parameter open-access multilingual language model[J]. arXiv: 2211.05100, 2022.

[38] SUN X, JI Y, MA B, et al. A comparative study between full-parameter and LoRA-based fine-tuning on chinese instruction data for instruction following large language model[J]. arXiv:2304.08109, 2023

[39] SEBASTIAN R. Recent advances in language model fine-tuning[EB/OL]. [2023-11-23]. https://www.ruder.io/recent-advances-lm-fine-tuning/.

[40] GUNEL B, DU J, CONNEAU A, et al. Supervised contrastive learning for pre-trained language model fine-tuning[J]. arXiv:2011.01403, 2020.

[41] HOWARD J, RUDER S. Universal language model fine-tuning for text classification[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), 2018: 328-339

[42] VÍCTOR C, SPRECHMANN P, HANSEN S, et al. Beyond fine-tuning: transferring behavior in reinforcement learning [J]. arXiv:2102.13515, 2021.

[43] MALLADI S, GAO T, NICHANI E, et al. Fine-tuning language models with just forward passes[J]. arXiv:2305.17333, 2023.

[44] LV K, YANG Y, LIU T, et al. Full parameter fine-tuning for large language models with limited resources[J]. arXiv: 2306.09782, 2023.

[45] PHOO C P, HARIHARAN B. Self-training for few-shot transfer across extreme task differences[J]. arXiv:2010.07734, 2020.

[46] LI S, CHEN D, CHEN Y, et al. Unsupervised Finetuning[J]. arXiv:2110.09510, 2021.

[47] XU Y, QIU X, ZHOU L, et al. Improving BERT fine-tuning via self-ensemble and self-distillation[J]. arXiv:2002.10345, 2020.

[48] ZHU C, CHENG Y, GAN Z, et al. FreeLB: enhanced adversarial training for natural language understanding[J]. arXiv: 1909.11764, 2019.

[49] JIANG H, HE P, CHEN W, et al. Smart: robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization[J]. arXiv: 1911.03437, 2019.

[50] YU Y, ZUO S, JIANG H, et al. Fine-tuning pre-trained language model with weak supervision: a contrastive-regularized self-training approach[J]. arXiv:2010.07835, 2020.

[51] TANWISUTH K, ZHANG S, ZHENG H, et al. POUF: prompt-oriented unsupervised fine-tuning for large pre-trained models [J]. arXiv:2305.00350, 2023.

[52] AGHAJANYAN A, ZETTLEMOYER L, GUPTA S. Intrinsic dimensionality explains the effectiveness of language model fine-tuning[J]. arXiv:2012.13255, 2020.

[53] HAN W, PANG B, WU Y. Robust transfer learning with pre-trained language models through adapters[J]. arXiv:2108.02340, 2021.

[54] LEE J, YOON W, KIM S, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining[J]. Bioinformatics, 2020, 36(4): 1234-1240.

[55] SEE A, LIU P J, Manning C D. Get to the point: summarization with pointer-generator networks[J]. arXiv: 1704.04368, 2017.

[56] LEWIS M, LIU Y, GOYAL N, et al. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension[J]. arXiv: 1910.13461, 2019.

[57] BIDERMAN S, SCHOELKOPF H, ANTHONY Q, et al. Pythia: a suite for analyzing large language models across training and scaling[J]. arXiv:2304.01373, 2023.

[58] LI X L, LIANG P. Prefix-tuning: optimizing continuous prompts for generation[J]. arXiv:2101.00190, 2021.

[59] LIU H, TAM D, MUQEETH M, et al. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning[C]//Advances in Neural Information Processing Systems, 2022: 1950-1965.

[60] ZAKEN E B, RAVFOGEL S, GOLDBERG Y. BitFit: simple parameter-efficient fine-tuning for transformer-based masked language-models[J]. arXiv:2106.10199, 2021.

[61] GUO D, RUSH A M, KIM Y. Parameter-efficient transfer learning with diff pruning[J]. arXiv:2012.07463, 2020.

[62] HU E J, SHEN Y, WALLIS P, et al. LoRA: low-rank adaptation of large language models[J]. arXiv:2106.09685, 2021.

[63] LI C, FARKHOOR H, LIU R, et al. Measuring the intrinsic dimension of objective landscapes[J]. arXiv:1804.08838, 2018.

[64] BACH F R, JORDAN M I. Predictive low-rank decomposition for kernel methods[C]//Proceedings of the 22nd International Conference on Machine Learning, 2005: 33-40.

[65] CHEN Y K, QIAN S J, TANG H T, et al. LongLoRA: Efficient fine-tuning of long-context large language models[J]. arXiv:2309.12307, 2023.

[66] CHAVAN A, LIU Z, GUPTA D, et al. One-for-all: generalized LoRA for parameter-efficient fine-tuning[J]. arXiv: 2306.07967, 2023.

[67] ZHANG Q, CHEN M, BUKHARIN A, et al. Adaptive budget allocation for parameter-efficient fine-tuning[J]. arXiv: 2303.10512, 2023.

[68] LUO M, XU X, LIU Y, et al. In-context learning with retrieved

demonstrations for language models: a survey[J]. arXiv: 2401.11624, 2024.
[69] RAZEGHI Y, LOGAN IV R L, GARDNER M, et al. Impact of pretraining term frequencies on few-shot reasoning[J]. arXiv:2202.07206, 2022.
[70] XIE S M, RAGHUNATHAN A, LIANG P, et al. An explanation of in-context learning as implicit bayesian inference [J]. arXiv:2111.02080, 2021.
[71] LIU J, SHEN D, ZHANG Y, et al. What makes good in-context examples for GPT-3?[J]. arXiv:2101.06804, 2021.
[72] HOLTZMAN A, WEST P, SCHWARTZ V, et al. Surface form competition: why the highest probability answer isn't always right[J]. arXiv:2104.08315, 2021.
[73] ZHAO T Z, WALLACE E, FENG S, et al. Calibrate before use: improving few-shot performance of language models [J]. arXiv:2102.09690, 2021.
[74] WEI J, WANG X, SCHUURMANS D, et al. Chain of thought prompting elicits reasoning in large language models[J]. arXiv:2201.11903, 2022.
[75] QIAO S, OU Y, ZHANG N, et al. Reasoning with language model prompting: a survey[J]. arXiv:2212.09597, 2022.
[76] CHEN W H, MA X G, WANG X Y, et al. Program of thoughts prompting: disentangling computation from reasoning for numerical reasoning tasks[J]. arXiv:2211.12588, 2022
[77] LONG J Y. Large language model guided tree-of-thought [J]. arXiv:2305.08291, 2023.
[78] NING X F, LIN Z N, ZHOU Z X, et al. Skeleton-of-thought: Large language models can do parallel decoding[J]. arXiv: 2307.15337, 2023.
[79] BESTA M, BLACH N, KUBICEK A, et al. Graph of thoughts: solving elaborate problems with large language models[J]. arXiv:2308.09687, 2023.
[80] LEI B, LIN P H, LIAO C, et al. Boosting logical reasoning in large language models through a new framework: the graph of thought [J]. arXiv:2308.08614, 2023.
[81] 林令德, 刘纳, 王正安. Adapter与Prompt Tuning微调方法研究综述[J]. 计算机工程与应用, 2023, 59(2): 12-21.
LIN L D, LIU N, WANG Z A. Review of research on Adapter and Prompt Tuning[J]. Computer Engineering and Applications, 2023, 59(2): 12-21.
[82] SHIN T, RAZEGHI Y, LOGAN I R L, et al. Autoprompt: eliciting knowledge from language models with automatically generated prompts[J]. arXiv:2010.15980, 2020.
[83] GAO T, FISCH A, CHEN D. Making pre-trained language models better few-shot learners[J]. arXiv:2012.15723, 2020.
[84] LIU X, ZHENG Y, DU Z, et al. GPT understands, too[J]. arXiv:2103.10385, 2021.
[85] LESTER B, AL-RFOU R, CONSTANT N. The power of scale for parameter-efficient prompt tuning[J]. arXiv:2104.08691, 2021.
[86] QIN G, EISNER J. Learning how to ask: querying LMs with mixtures of soft prompts[J]. arXiv:2104.06599, 2021.
[87] LONGPRE S, HOU L, VU T, et al. The flan collection: designing data and methods for effective instruction tuning [J]. arXiv:2301.13688, 2023.
[88] SANH V, WEBSON A, RAFFEL C, et al. Multitask prompted training enables zero-shot task generalization[J]. arXiv:2110. 08207, 2021.
[89] XUE F Z, JAIN K, SHAH M H, et al. Instruction in the wild: a user-based instruction dataset[EB/OL]. [2023-11-23]. https://github.com/XueFuzhao/InstructionWild.
[90] WANG Y Z, MISHRA S, ALIPOORMOLABASHI P, et al. Super-naturalinstructions: generalization via declarative instructions on 1600+ NLP tasks[J]. arXiv:2204.07705, 2022.
[91] MUENNIGHOFF N, WANG T, SUTAWIKA L, et al. Crosslingual generalization through multitask finetuning[J]. arXiv: 2211.01786, 2022.
[92] DING N, CHEN Y, XU B, et al. Enhancing chat language models by scaling high-quality instructional conversations [J]. arXiv:2305.14233, 2023.
[93] YAO S Y, YU D, ZHAO J, et al. Tree of thoughts: deliberate problem solving with large language models[J]. arXiv:2305. 10601, 2023.
[94] XU Z Y, SHEN Y, HUANG L F. Multiinstruct: improving multi-modal zero shot learning via instruction tuning[J]. arXiv:2212.10773, 2022.
[95] BARAL C, YANG Y Z, BLANC E, et al. Towards development of models that learn new tasks from instructions[D]. Phoenix City: Arizona State University, 2023.
[96] MARTIN A, ASHIISH A, PAUL B, et al. Tensorflow: large-scale machine learning on heterogeneous distributed systems [J]. arXiv:1603.04467, 2016.
[97] OUYANG L, WU J, JIANG X, et al. Training language models to follow instructions with human feedback[J]. arXiv:2203. 02155, 2022.
[98] BAI Y, JONES A, NDOUSSE K, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback[J]. arXiv:2204.05862, 2022.
[99] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[J]. arXiv:1707.06347, 2017.
[100] BAI Y T, KADAVATH S, KUNDU S, et al. Constitutional AI: harmlessness from AI feedback. 2022[J]. arXiv:2212. 08073, 2022.
[101] LEE H, PHATALE S, MANSOOR H, et al. RLAIF: scaling reinforcement learning from human feedback with ai feedback[J]. arXiv:2309.00267, 2023.
[102] WU Z X, LIU N F, POTTS C. Identifying the limits of cross-domain knowledge transfer for pretrained models[J]. arXiv:2104.08410, 2021.
[103] QI X, ZENG Y, XIE T, et al. Fine-tuning aligned language

models compromises safety, even when users do not intend to![J]. arXiv:2310.03693, 2023.

[104] HE J, CHEN J, HE S, et al. AdaMix: mixture-of-adaptations for parameter-efficient model tuning[J]. arXiv: 2205.09717, 2022.

[105] ZHAO W X, ZHOU K, LI J, et al. A survey of large language models[J]. arXiv:2303.18223, 2023.

[106] HOULSBY N, GIURGIU A, JASTRZEBSKI S, et al. Parameter-efficient transfer learning for NLP[J]. arXiv: 1902.00751, 2019.

[107] WANG A, SINGH A, HILL F, et al. GLUE: a multi-task benchmark and analysis platform for natural language understanding[J]. arXiv:1804.07461, 2018.

[108] HE R, LIU L, YE H, et al. On the effectiveness of adapter-based tuning for pretrained language model adaptation[J]. arXiv:2106.03164, 2021.

[109] YANG H, LI P, LAM W. Parameter-efficient tuning by manipulating hidden states of pretrained language models for classification tasks[J]. arXiv:2204.04596, 2022.

[110] HE P, LIU X, GAO J, et al. DeBERTa: decoding-enhanced BERT with disentangled attention[J]. arXiv:2006.03654, 2020.

[111] ZHAI X, PUIGCERVER J, KOLESNIKOV A, et al. A large-scale study of representation learning with the visual task adaptation benchmark[J]. arXiv:1910.04867, 2019.

[112] BANSAL M, KUMAR M, SACHDEVA M, et al. Transfer learning for image classification using VGG19: Caltech-101 image data set[J]. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(4): 3609-3620.

[113] HELBER P, BISCHKE B, DENGEL A, et al. EuroSAT: a novel dataset and deep learning benchmark for land use and land cover classification[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2019, 12(7): 2217-2226.

[114] JOHNSON J, HARIHARAN B, MAATEN L V D, et al. Clevr: a diagnostic dataset for compositional language and elementary visual reasoning[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2901-2910.

[115] WILLIAMS A, NANGIA N, BOWMAN S R. A broad-coverage challenge corpus for sentence understanding through inference[J]. arXiv:1704.05426, 2017.

[116] WOLF T, DEBUT L, SANH V, et al. Transformers: state-of-the-art natural language processing[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020: 38-45.

[117] HE J, ZHOU C, MA X, et al. Towards a unified view of parameter-efficient transfer learning[J]. arXiv: 2110.04366, 2021.

[118] CHRISTIANO P, LEIKE J, BROWN T B, et al. Deep reinforcement learning from human preferences[J]. arXiv: 1706.03741, 2017.

[119] KINGMA D P, BA J. ADAM: a method for stochastic optimization[J]. arXiv:1412.6980, 2014.

[120] ZIEGLER D M, STIENNON N, WU J, et al. Fine-tuning language models from human preferences[J]. arXiv: 1909.08593, 2019.

[121] GANESAN K. Rouge 2.0: updated and improved measures for evaluation of summarization tasks[J]. arXiv: 1803.01937, 2018.

[122] TOUVRON H, MARTIN L, STONE K, et al. LLaMA 2: open foundation and fine-tuned chat models[J]. arXiv: 2307.09288, 2023.

[123] CASPER S, DAVIES X, SHI C, et al. Open problems and fundamental limitations of reinforcement learning from human feedback[J]. arXiv:2307.15217, 2023.