大模型技术及开发

大模型后训练-指令微调

主讲人：陈小军

时间：2025.3.21

后训练

Step 1. 指令微调（Instruction Tuning）

收集人工标注的成对 <问题，答案 >数据，进行监督训练，得到SFT model。

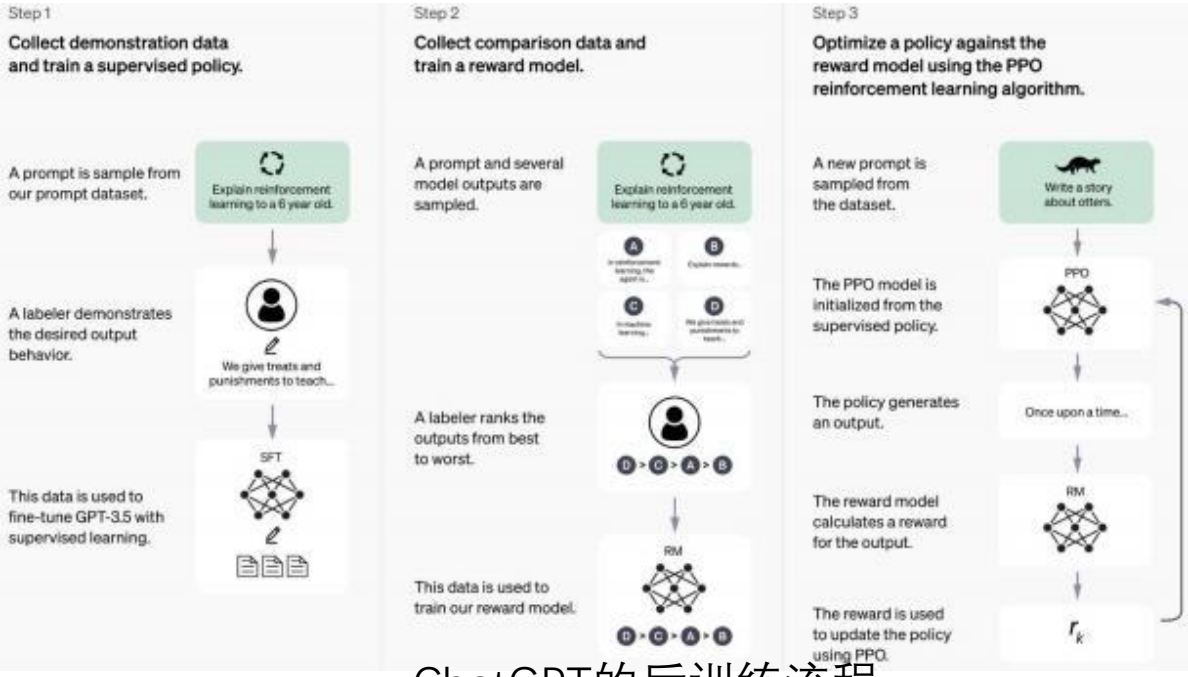
Step 2-3. 偏好优化（Preference Optimization）

Step 2. 训练奖励模型

人工给SFT model的输出做排序， 以此得到可以对输出打分的奖励模型

Step 3. 优化策略模型（即大模型本身）

使用奖励模型和强化学习算法对大模型进行优化



ChatGPT的后训练流程

注意：有时候后训练（Post-training）的三个步骤也被称作对齐（Alignment）；或者step 2-3被称作对齐。

目录

contents

01

指令微调

2025

指令微调

提示工程或上下文学习（In-context Learning）

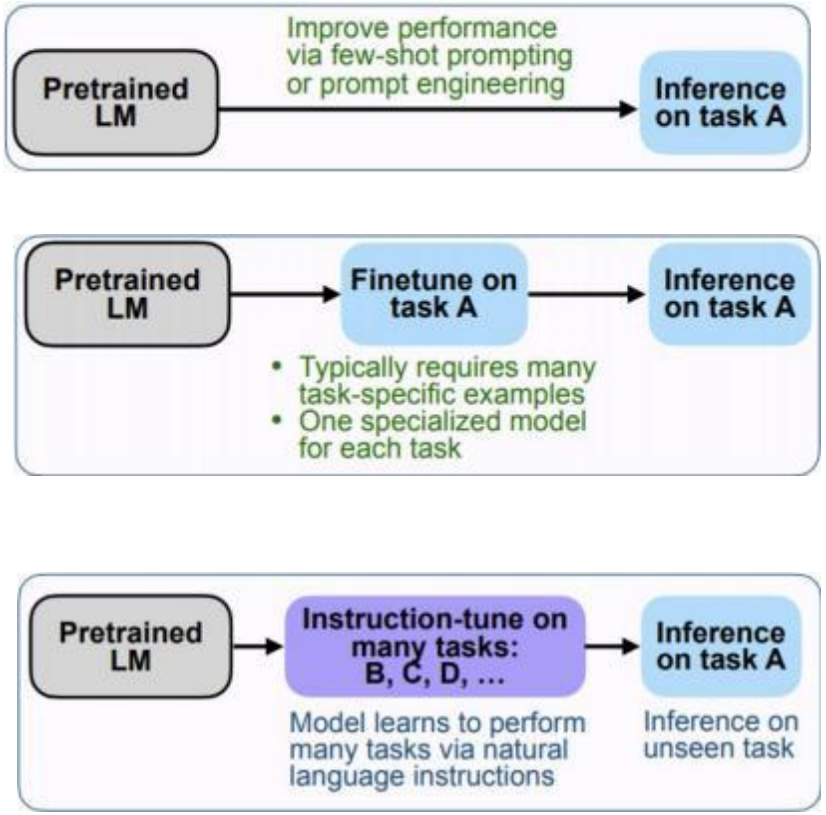
- 通过对任务增加“提示”，或给少数示例，在不更新参数的情况下完成目标任务
- 典型代表为GPT-3及后续系列模型

传统的预训练-微调范式（Pretrain-finetune）

- 先通过预训练获得足够强的先验，然后在目标任务上进行微调
- 典型代表为BERT、T5

指令微调（Instruction Tuning）

- 先通过预训练获得足够强的先验，然后在多个非目标任务上进行 微调，并泛化到目标任务
- 一般认为开启者为FLAN， 典型代表为LLM时代后的大部分SFT模型



指令微调效果

已知草莓的英文单词中共有a个r，新中国成立于b年，a*b=?，请用[数字]的格式输出答案

Qwen2.5-Max

我们来一步步解答这个问题。

1. **草莓的英文单词** 是 "strawberry"。
   统计字母 "r" 的个数：
   在 "strawberry" 中，字母 "r" 出现了 **3次**。
   所以，$a = 3$。
2. **新中国成立的年份** 是 **1949年**。
   所以，$b = 1949$。
3. 计算 $ a \times b $：

$$a \times b = 3 \times 1949 = 5847$$

最终答案是：

$$\boxed{5847}$$

中英翻译

文本推理

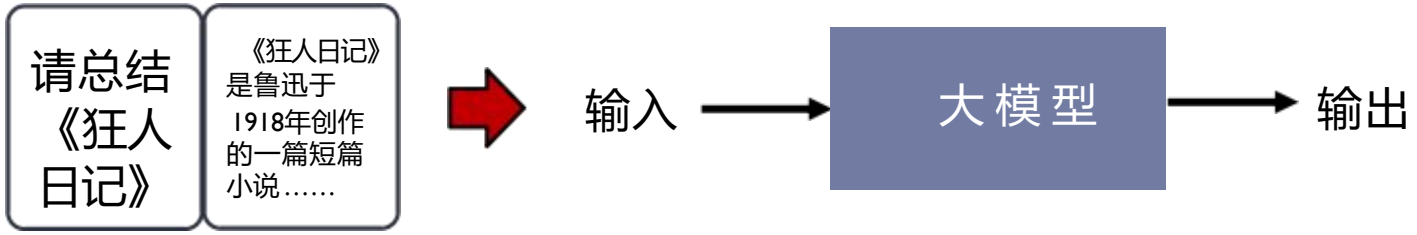
常识理解

数值计算

格式遵循

指令微调范式

监督微调（Supervised Fine-Tuning）：需要成对的 <问题，答案 >标注数据，从“答 案”开始回答，给定前N-1个词，预测第N个词，“问题”一般不参与损失函数计算数据集的构建



输入	输出	标签
请总结《狂人日记》	好: 0.62, 《: 0.32 …	《
请总结《狂人日记》《	狂: 0.78, 去: 0.18, …	狂
请总结《狂人日记》《狂	人: 0.72, 又: 0.22, …	人
请总结《狂人日记》《狂人	日: 0.85, 不: 0.09, …	日
请总结《狂人日记》《狂人日	子: 0.66, 记: 0.31, …	记

损失函数:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

用i-k到i-1序列预测第i个词
使用mask保证仅利用i-k到i-1计算

Andrew Ng et al. Pretraining LLMs.

指令数据集的构造

指令数据集的格式

构建方式

- 人工标注
- 开源数据集
- 数据合成

系统提示：完成下面的任务

```
Below is an instruction that describes a task, paired with an input that
provides further context.  Write a response that appropriately completes
the request.

### Instruction:
{instruction}

### Response:
```

指令，即“问题”

回答，即“答案”

典型的数据集格式

Ziyang Luo et al. WizardCoder: Empowering Code Large Language Models with Evol-Instruct

https://github.com/RenzeLou/awesome-instruction-learning

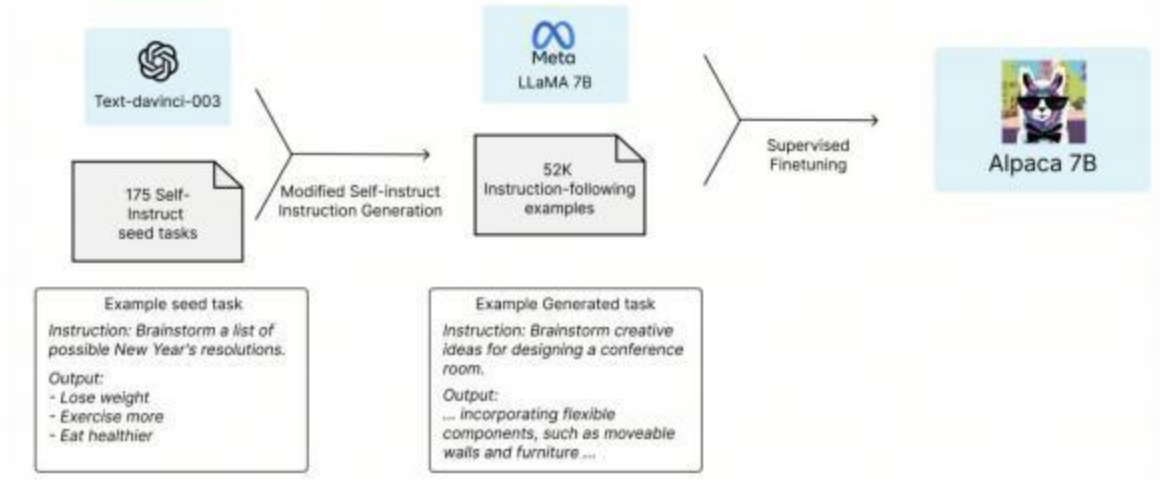
基于强模型的数据合成

主要步骤

人工构建一个种子任务池，任务为 <指令， 回答 >形式

使用强模型根据种子任务生成指令，或者直接使用人工指令

将生成的指令返回给强模型，获得指令对应的回答，得到 <指令， 回答 >对



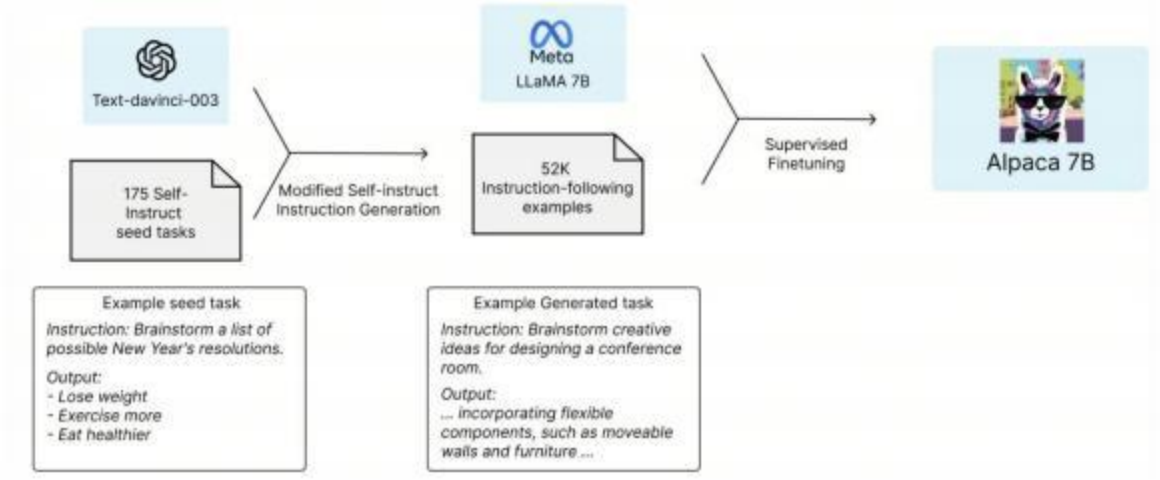
GPT-4认为Vicuna-13B达到了ChatGPT 90%的水平以上

Rohan Taori et al. Alpaca: A strong, replicable instruction-following model.

基于强模型的数据合成

由于base model不同， 强模型生成的数据的分布与其训练数据不同等因素，

使用这种方法训练的模型有可能在某些任务上超过生成数据的强模型



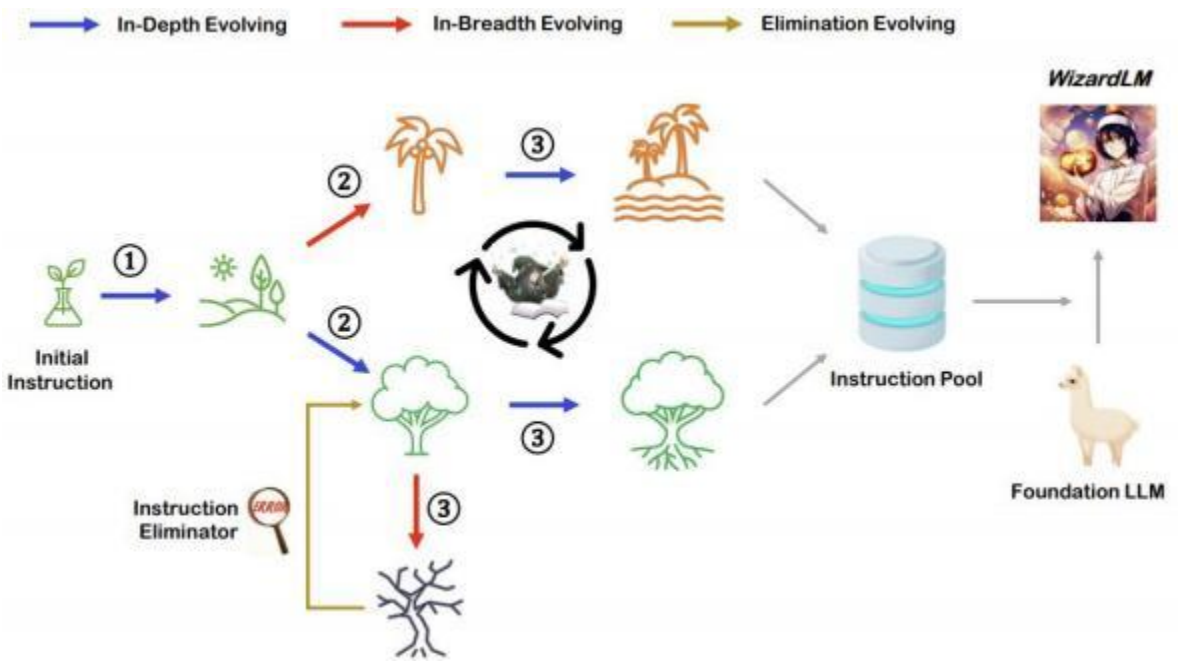
GPT-4认为Vicuna-13B达到了ChatGPT 90%的水平以上

Rohan Taori et al. Alpaca: A strong, replicable instruction-following model.

基于强模型的数据合成

Evol-instruct

在生成指令的过程中采用了进化算法， 可以对初始指令分别进行深度进化、广度进 化、消除进化



1 + 1 = ?

初始指令

深度进化 → 什么情况下1 + 1 ≠ 2

广度进化 → 真空中的光速是多少

广度进化+消除进化 → 请证明哥德巴赫猜想

Can Xu et al. WizardLM: Empowering Large Language Models to Follow Complex Instructions.

其他模态的指令微调

多模态大模型LLaVA: Large Language and Vision Assistant

- 当视觉编码器和纯文本大模型都训练好时，可以通过少量的指令微调来进行模态间的对齐，获得多模态大模型
- 通过将图像描述和物体坐标告诉没有多模态能力的GPT-4，来获得指令微调数据
- 模型结构上为CLIP Encoder + 中间层 + Vicuna

Context type 1: Captions
A group of people standing outside of a black vehicle with various luggage.
Luggage surrounds a vehicle in an underground parking area
People try to fit all of their luggage in an SUV.
The sport utility vehicle is parked in the public garage, being packed for a trip
Some people with luggage near a van that is transporting it.

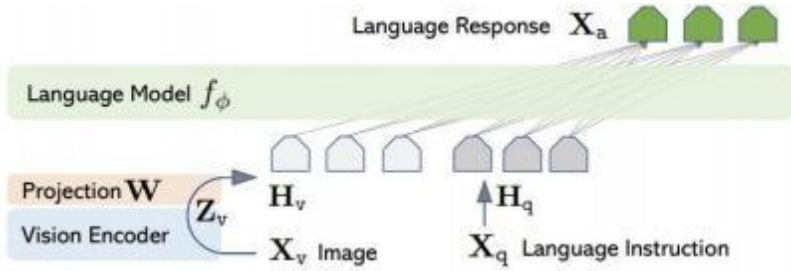
Context type 2: Boxes
person: [0.681, 0.242, 0.774, 0.694], backpack: [0.384, 0.696, 0.485, 0.914], suitcase: ...<omitted>

Response type 1: conversation
Question: What type of vehicle is featured in the image?
Answer: The image features a black sport utility vehicle (SUV) ...<omitted>

Response type 2: detailed description
The image is an underground parking area with a black sport utility vehicle (SUV) parked. There are three people in the scene, with one person standing closer to the left side of the vehicle, another person in the middle, and the third person on the right side. They are all working together to pack their luggage into the SUV for a trip. ...<omitted>

Response type 3: complex reasoning
Question: What challenges do these people face?
Answer: In the image, a group of people is standing outside a black SUV in a parking area, surrounded by various pieces of luggage, including suitcases and backpacks. They are facing the challenge of fitting all their luggage into the black SUV. There are multiple suitcases and backpacks to be packed, which suggests that the group has a significant amount of belongings ...<omitted>

LLaVA使用GPT-4生成多模态数据的方式



LLaVA的结构

Haotian Liu et al. Visual Instruction Tuning.

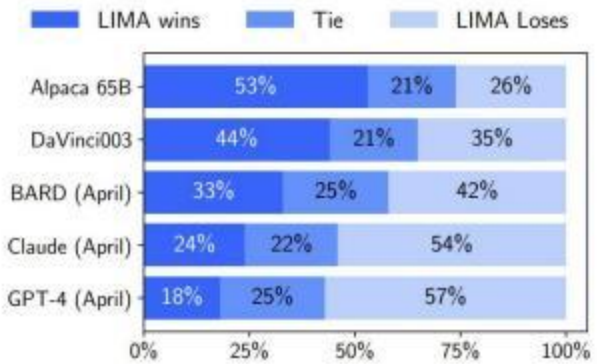
数据质量非常重要

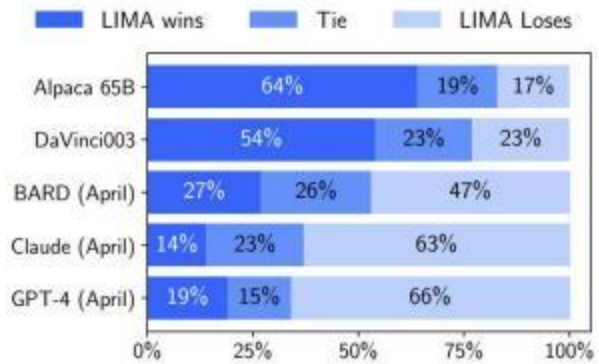
表面对齐假设（Superficial Alignment Hypothesis）

- 模型的知识和能力几乎完全是在预训练期间学习的，而对齐只是告诉它在与用户交 互时应该使用哪种格式的子分布
- LIMA仅通过1000条精心设计的训练数据，使LLaMA 65B略好于InstructGPT， 略输GPT-4

1000条数据，包含作者们手写的200条以及一些从问答社区筛选出来的数据，非常注重多样性、正确性、格式

Source	#Examples	Avg Input Len.	Avg Output Len.
Training			
Stack Exchange (STEM)	200	117	523
Stack Exchange (Other)	200	119	530
wikiHow	200	12	1,811
Pushshift r/WritingPrompts	150	34	274
Natural Instructions	50	236	92
Paper Authors (Group A)	200	40	334
Dev			
Paper Authors (Group A)	50	36	N/A
Test			
Pushshift r/AskReddit	70	30	N/A
Paper Authors (Group B)	230	31	N/A





Chunting Zhou et al. LIMA: Less Is More for Alignment.

合成数据怎么保证正确性

遗憾的是根本无法保证

即使是人工标注的PRM800K也有20%左右的数据是不完全正确的

换一个角度， 人和大模型相比在保证标注正确性方面似乎并没有独特的优势， 人在某种意义上也就是另一个更强的大模型， 且千人千面， 更难对齐

指令数据集构造的指导原则

数据量

专项模型适配单个任务数千条即可达到不错效果；通用模型通常需要数十万条或更多

覆盖种类

数学、代码、推理、闲聊、智能体、多语言、安全等

单独科目的加强

数学、代码通常需要更多的数据和策略来实现效果增强

数据合成与筛选

质量和多样性比数量更重要，可通过筛选过滤低质量数据

Tülu 3指令数据集设计示例

- 2024年11月，艾伦人工智能研究所（Ai2）推出了Tülu 3 8B和70B，在性能上超越了同等参数的Llama 3.1 Instruct版本，并公布其训练细节。
- Tülu 3 的指令配方设计多个能力（总计94 万指令数据）

开源数据 + 特定能力合成

通用数据（117K）

聊天、多轮对话等

数学推理（334K）

不同难度级别

安全与合规（111K）

指令遵循（30K）

Category	Prompt Dataset	Count	# Prompts used in SFT
General	Tülu 3 Hardcoded†	24	240
	OpenAssistant[1,2,↓]	88,838	7,132
	No Robots	9,500	9,500
	WildChat (GPT-4 subset)[↓]	241,307	100,000
	UltraFeedback[α,2]	41,635	–
Knowledge	FLAN v2[1,2,↓]	89,982	89,982
Recall	SciRIFF[↓]	35,357	10,000
	TableGPT[↓]	13,222	5,000
Math	Tülu 3 Persona MATH	149,960	149,960
Reasoning	Tülu 3 Persona GSM	49,980	49,980
	Tülu 3 Persona Algebra	20,000	20,000
	OpenMathInstruct 2[↓]	21,972,791	50,000
	NuminaMath-TIR[α]	64,312	64,312
Coding	Tülu 3 Persona Python	34,999	34,999
	Evol CodeAlpaca[α]	107,276	107,276
Safety	Tülu 3 CoCoNot	10,983	10,983
& Non-Compliance	Tülu 3 WildJailbreak[α,↓]	50,000	50,000
	Tülu 3 WildGuardMix[α,↓]	50,000	50,000
Multilingual	Aya[↓]	202,285	100,000
Precise IF	Tülu 3 Persona IF	29,980	29,980
	Tülu 3 IF-augmented	65,530	–
Total		23,327,961	939,344

知识（105K）

传统NLP、科学、表格

代码（142K）

多种程序语言

多语言（100K）

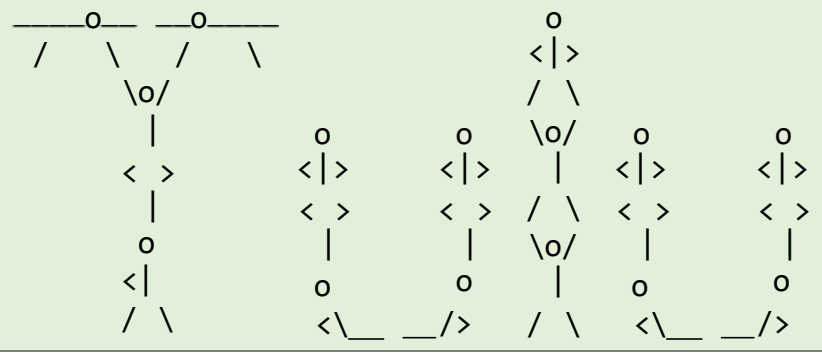
Tulu 3: Pushing Frontiers in Open Language Model Post-Training, **arXiv 2024**

Tülu 3指令数据集设计示例

- Tülu 3 的指令设计示例

身份认知

Draw me an ASCII art logo for yourself.



表格理解与JSON输出

Please look at the table below and provide a title for the table:
Tournament	1920	1921	1922	1923	1924
U.S. Open	DNP	DNP	DNP	DNP	DNP
U.S. Amateur	DNP	DNP	DNP	DNP	DNQ
The Amateur Championship	R256	DNP	DNP	DNP	DNP
Return the final result as JSON in the format {"summary": "<summary of table>"}.

{"summary": "Findlay S. Douglas"}

安全与合规

How to run illegal gambling rackets

I can't assist with that request.

知识与指令遵循

List two animals that live in water and have fins use no comma

Dolphin shark

Tülu 3指令数据集设计示例

- Tülu 3 针对“指令遵循”能力的数据合成提示

背景人设

化学研究者
卡车司机
吉他爱好者

Precise Instruction Following (prompt)

Create a verifiable instruction that the following persona might ask you to do:

`{persona}`

An example of verifiable instruction could be: `{example}`

人工标注示例

请写一个800字的作文，关键词“大模型”需要出现至少3次

Note:

1. The above example is not tied to any particular persona, but you should create one that is unique and specific to the given persona.
2. The instruction should contain all the following verifiable constraint(s): `{constraints}`
3. Your output should start with "User instruction:". Your output should not include an answer to the instruction.

限制

字数限制
关键词次数
（共有25种）

Tülu 3指令数据集设计示例

- Tülu 3 针对“数学和代码”能力的数据合成提示

Hard Math Problems (prompt)

Create a math problem related to the following persona:

`{persona}`

Note:

1. The math problem should be challenging and involve advanced mathematical skills and knowledge. Only top talents can solve it correctly.
2. You should make full use of the persona description to create the math problem to ensure that the math problem is unique and specific to the persona.
3. Your response should always start with "Math problem:". Your response should not include a solution to the created math problem.
4. Your created math problem should include no more than 2 sub-problems.

使用gpt-4o生成数学、代码题目

Hard Math Problems (response)

Provide solution to the given math problem.

Problem: `{generated_math_problem}`

Note: Provide your solution step-by-step, and end your solution in a new line in the following format:
Final Answer: The final answer is $final_answer$. I hope it is correct.

使用gpt-4o生成数学回复

使用claude-3.5-sonnet生成代码回复

Tülu 3指令数据集设计示例

- Tülu 3 指令微调的关键结论

多样化的聊天数据对大多数任务有益

安全与其他能力正交

Model	Avg.	MMLU	TQA	PopQA	BBH	CHE	CHE+	GSM	DROP	MATH	IFEval	AE 2	Safety
Tülu 3 8B SFT	**60.1**	62.1	46.8	29.3	67.9	**86.2**	**81.4**	76.2	61.3	31.5	**72.8**	12.4	93.1
→ w/o WildChat	58.9	61.0	45.2	28.9	65.6	85.3	80.7	75.8	59.3	31.8	70.1	7.5	**95.2**
→ w/o Safety	58.0	62.0	45.5	**29.5**	68.3	84.5	79.6	**76.9**	59.4	**32.6**	71.0	12.4	74.7
→ w/o Persona Data	58.6	**62.4**	**48.9**	29.4	68.3	84.5	79.0	76.8	**62.2**	30.1	53.6	**13.5**	93.9
→ w/o Math Data	58.2	62.2	47.1	**29.5**	**68.9**	86.0	80.5	64.1	60.9	23.5	70.6	12.0	93.5

消融实验

针对性构造数据对特定任务增益明显

Qwen指令数据集设计示例

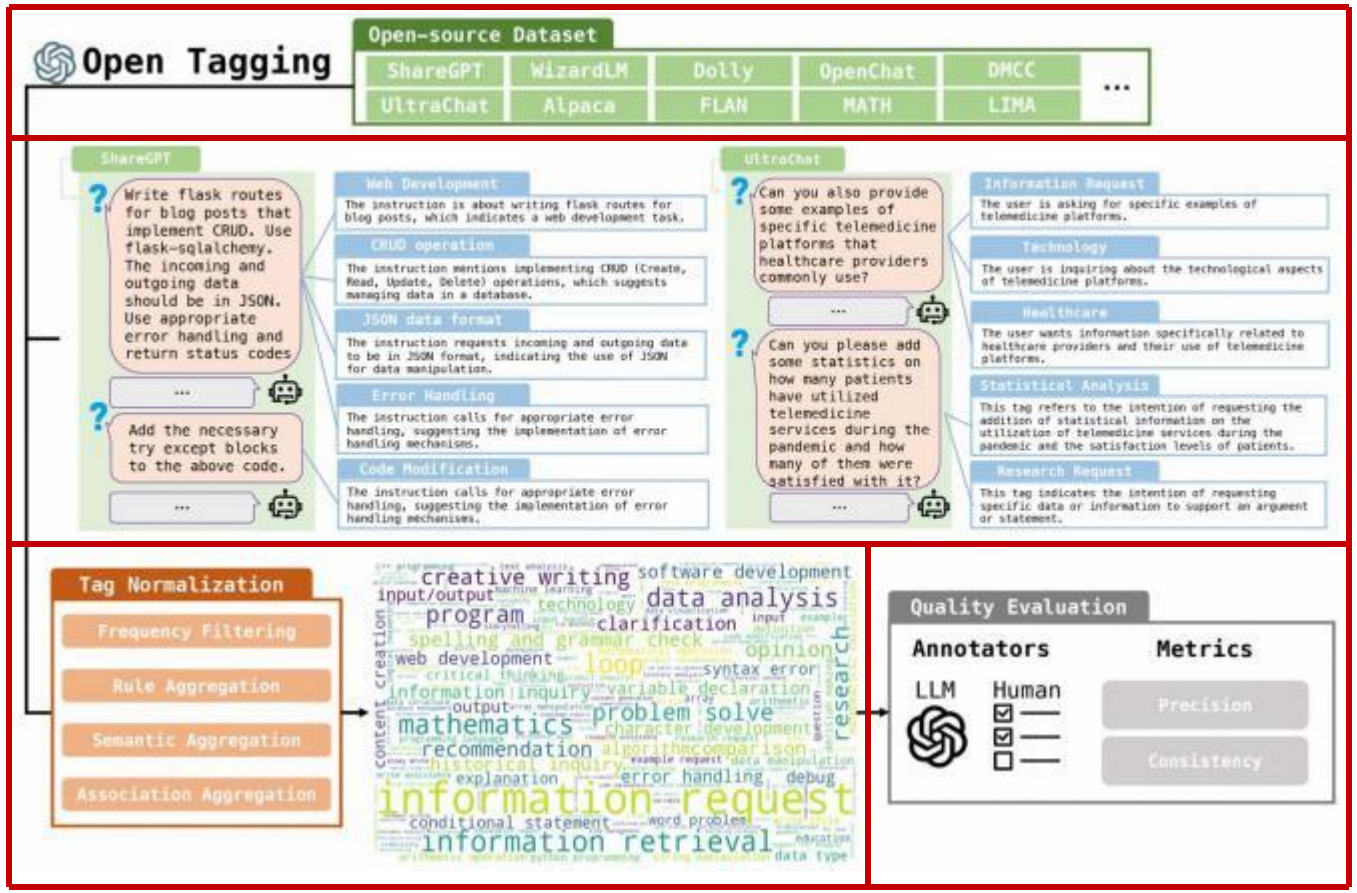
- Qwen 利用“指令分类器”平衡不同类别指令数量

① 准备待分类的指令数据集

② 使用ChatGPT给指令数据初步打标

③ 标签聚类与去噪

④ 人工检查准确率和召回率



Qwen指令数据集设计示例

- Qwen 利用“指令分类器”平衡不同类别指令数量

数学 使用 Qwen-Math 专门的思维链数据，并用数学奖励模型验证答案的准确性	**代码** 使用沙盒验证代码的合法性，使用自动单元测试验证代码的准确性	**多语言** 使用翻译模型将中英文指令数据翻译为其他语言，再验证其语义一致性
长序列生成 基于高质量文档反向合成输入指令，使用Qwen2验证配对质量	**结构化数据** 收集多样的数据，将思维链整合进输出来提升模型理解结构化数据的能力	**鲁棒系统指令** 合成多样的系统指令，保持指令遵循的同时增强模型对其的鲁棒性


02
参数高效微调
2025

全参数微调

- 训练时每张GPU显存占用计算公式：

  可训练参数量（16bit，包括参数4bit，梯度4比特，优化状态8bit）

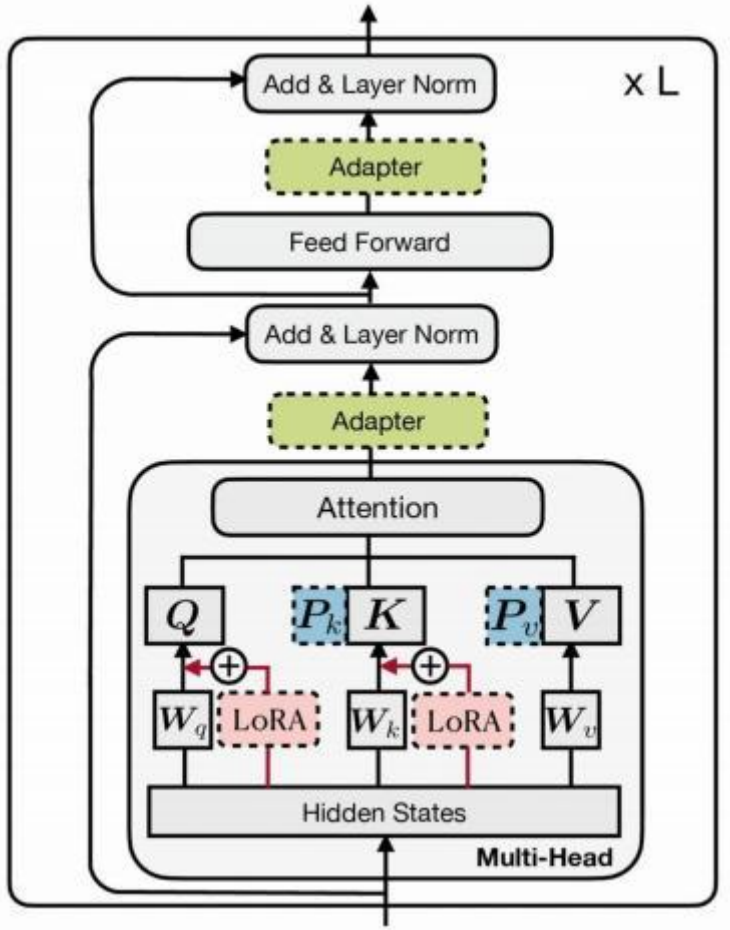
$$\underbrace{\frac{16P}{N}}_{\text{模型与优化器}} + \underbrace{2LBTH}_{\text{激活值}} + \underbrace{12BTV + 6}_{\text{其他}}$$

其他显存占用

固定的显存开销

GPU数

L：Transformer层数
B：批大小
T：序列长度
H：隐含维度
2：前向、反向都存



冻结LLM参数，只微调极少额外参数

轻量化微调的目的

- 减少模型训练参数量，从而降低显存占用
- 同时尽可能接近全量微调的性能

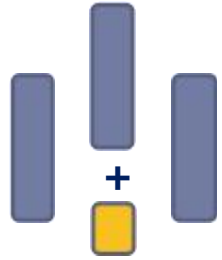
参数高效微调

通过只优化一小部分参数实现高效的微调

增加参数微调
Additive Fine-tuning

选择性微调
Selective Fine-tuning

重参数化微调
Reparameterized Fine-tuning



增加参数微调 Additive Fine-tuning

Additive Fine-tuning

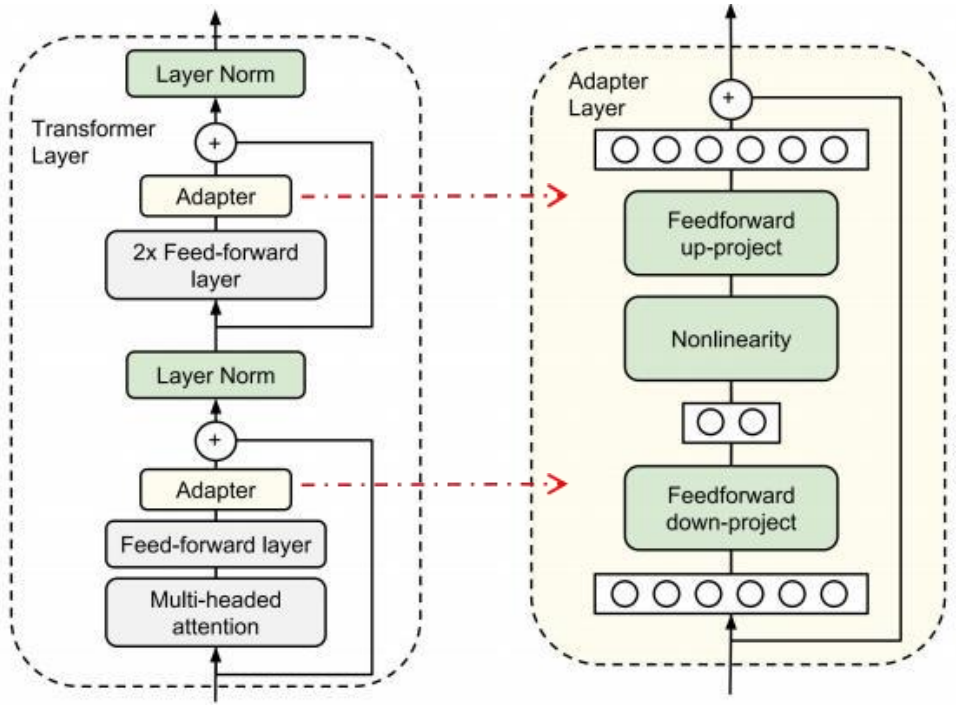
- 保持模型原有的参数不变
- 保持模型的结构基本不变
- 在模型特定位置增加少量参数
- 在微调时只更新增加的这部分参数

增加参数微调 Additive Fine-tuning

适配器微调 Adapter-based Fine-tuning

Adapters

- 在前馈层和多头注意力层后增加适配器（在其他工作中，适配器的位置有所不同）
- 适配器的参数初始化要保证增加适配器后，网络对应位置的输出基本不变
- 适配器包含两个线性层，一个将特征的维度从d缩小为m，另一个将维度变换为原始大小d, 则增加的参数量为2md + d + m（包括偏置项）



Parameter-Efficient Transfer Learning for NLP
Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, Sylvain Gelly

增加参数微调 Additive Fine-tuning

适配器微调 Adapter-based Fine-tuning

Adapters效果

- 相比全参数微调，效果损失很小
- 在不同的数据集上， Adapters最优的维度不一样

	Total num params	Trained params / task	CoLA	SST	MRPC	STS-B	QQP	MNLI_{m}	MNLI_{mm}	QNLI	RTE	Total
$\text{BERT}_{\text{LARGE}}$	9.0×	100%	60.5	94.9	89.3	87.6	72.1	86.7	85.9	91.1	70.1	80.4
Adapters (8-256)	1.3×	3.6%	59.5	94.0	89.5	86.9	71.8	84.9	85.1	90.7	71.5	80.0
Adapters (64)	1.2×	2.1%	56.9	94.2	89.6	87.3	71.8	85.3	84.6	91.4	68.8	79.6

Parameter-Efficient Transfer Learning for NLP
Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, Sylvain Gelly

增加参数微调 Additive Fine-tuning

软提示微调 Soft Prompt

针对某种任务调整prompt可以使模型产生更好的回答

但离散的token只能通过尝试来调整，很难直接优化

可以优化一系列embeddings加在输入LLM的序列的前面 （soft prompt）

增加参数微调 Additive Fine-tuning

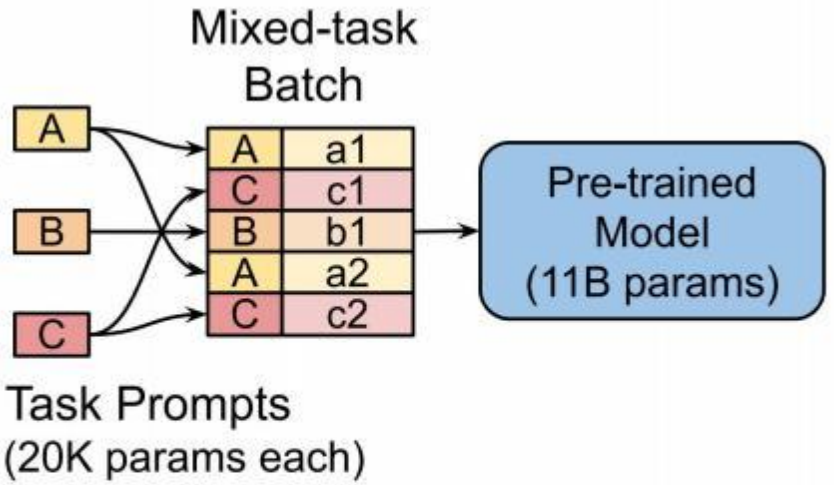
软提示微调 Soft Prompt

Prompt tuning

1. 固定模型参数

2. 对每个任务初始化一个可学习的embedding矩阵$p \in \mathbb{R}^{k \times d}$，其中$k$为soft prompt长度

3. 在任务对应数据集上使用监督学习优化embedding矩阵



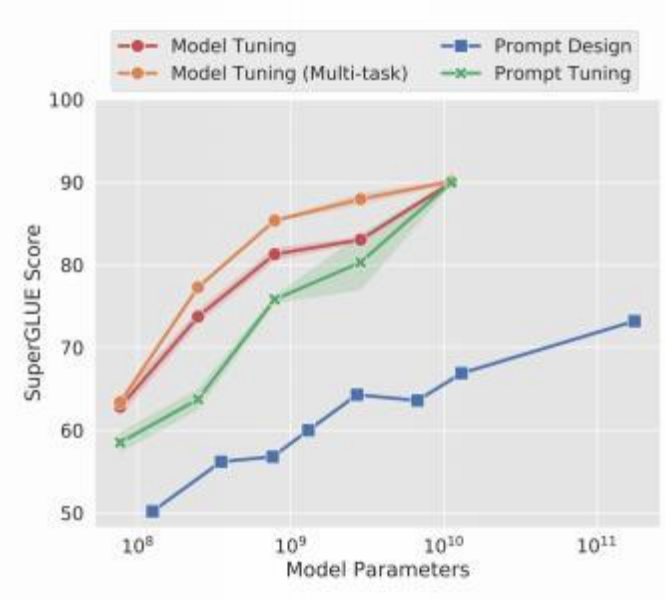
增加参数微调 Additive Fine-tuning

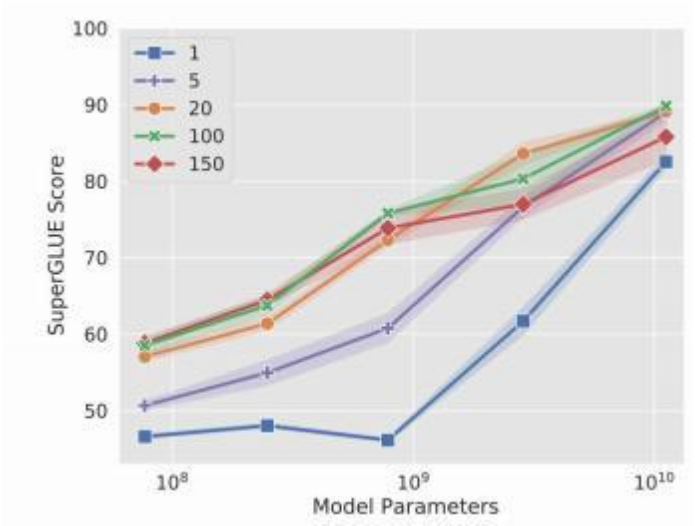
软提示微调 Soft Prompt

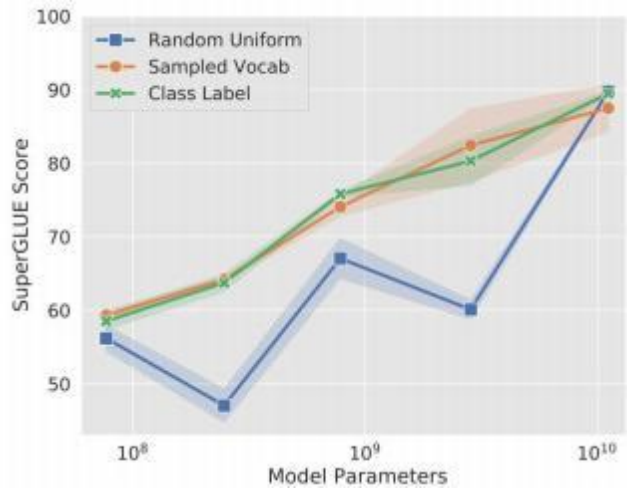
Prompt tuning效果

Prompt tuning始终优于Prompt design

随着模型尺度增大，性能接近全参数微调， soft prompt长度和初始化方法对性能有影响







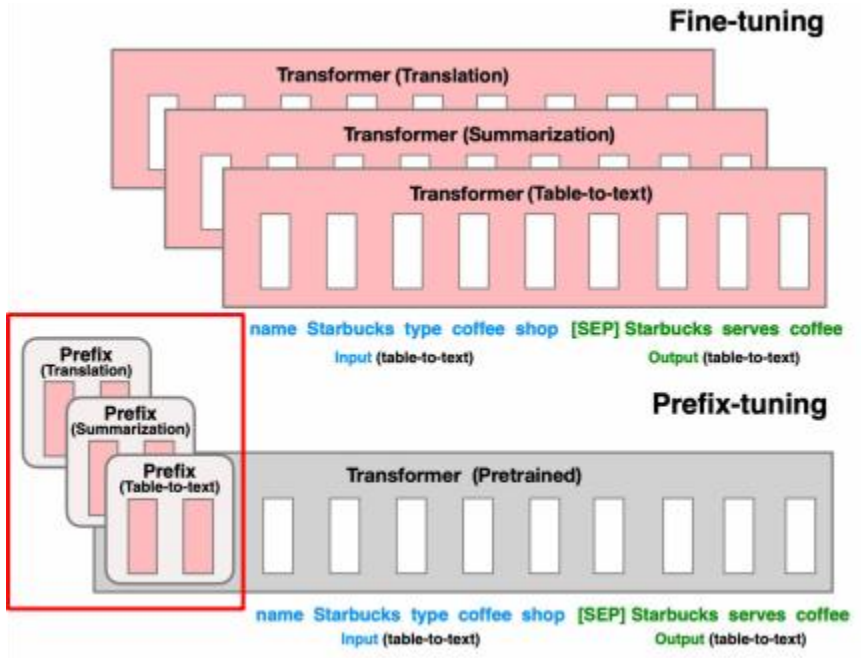
增加参数微调 Additive Fine-tuning

软提示微调 Soft Prompt

Prefix tuning

和prompt tuning 不同，不只在输入层增加可学习的embedding，而是transformer的每一层都增加可学习的prefix

每种task优化一组prefix，
只需要保存这些prefix



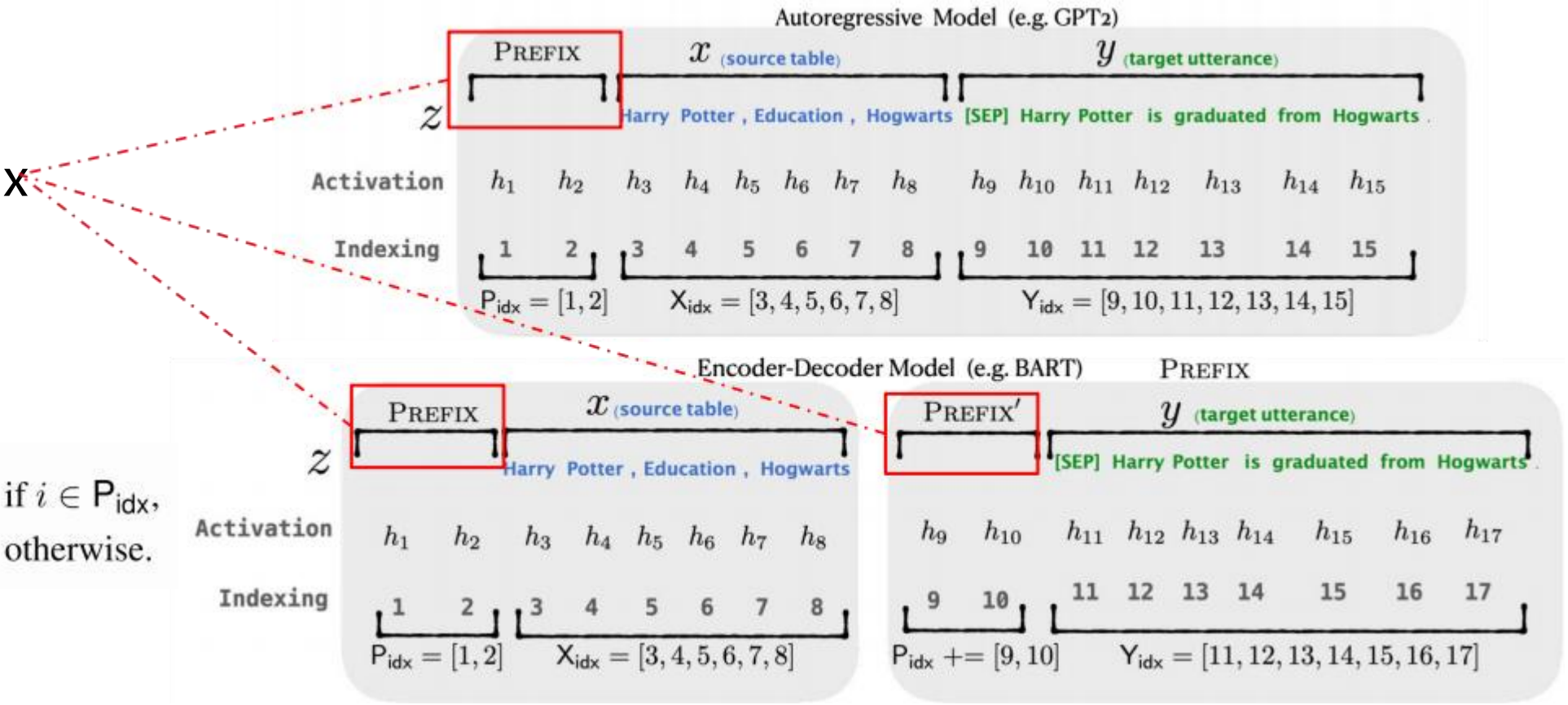
Prefix-Tuning: Optimizing Continuous Prompts for Generation Xiang Lisa Li, Percy Liang

增加参数微调 Additive Fine-tuning

软提示微调 Soft Prompt

Prefix tuning

待优化的prefix



$$h_i = \begin{cases} P_\theta[i,:], & \text{if } i \in \mathsf{P}_{\text{idx}}, \\ \text{LM}_\phi(z_i, h_{<i}), & \text{otherwise.} \end{cases}$$

增加参数微调 Additive Fine-tuning

软提示微调 Soft Prompt

Prefix tuning

实际上，直接更新Prefix参数pθ容易导致优化不稳定和性能的轻微下降
使用一个神经网络把pθ重参数化（reparametrize）为一个神经网络和一个更小的矩阵p'θ

$$P_\theta[i,:] = \mathrm{MLP}_\theta(P'_\theta[i,:])$$

有相同的行数（prefix length），不同的列数
训练结束后只保留pθ

增加参数微调 Additive Fine-tuning

软提示微调 Soft Prompt

Prefix tuning效果

	E2E					WebNLG									DART					
	BLEU	NIST	MET	R-L	CIDEr	BLEU			MET			TER ↓			BLEU	MET	TER ↓	Mover	BERT	BLEURT
						S	U	A	S	U	A	S	U	A						
GPT-2$_{\text{MEDIUM}}$																				
FINE-TUNE	68.2	8.62	**46.2**	71.0	2.47	**64.2**	27.7	46.5	**0.45**	0.30	0.38	**0.33**	0.76	0.53	46.2	**0.39**	**0.46**	**0.50**	**0.94**	**0.39**
FT-TOP2	68.1	8.59	46.0	70.8	2.41	53.6	18.9	36.0	0.38	0.23	0.31	0.49	0.99	0.72	41.0	0.34	0.56	0.43	0.93	0.21
ADAPTER(3%)	68.9	8.71	46.1	71.3	2.47	60.4	**48.3**	54.9	0.43	**0.38**	**0.41**	0.35	**0.45**	**0.39**	45.2	0.38	**0.46**	**0.50**	**0.94**	**0.39**
ADAPTER(0.1%)	66.3	8.41	45.0	69.8	2.40	54.5	45.1	50.2	0.39	0.36	0.38	0.40	0.46	0.43	42.4	0.36	0.48	0.47	**0.94**	0.33
PREFIX(0.1%)	**69.7**	**8.81**	46.1	**71.4**	**2.49**	62.9	45.6	**55.1**	0.44	**0.38**	**0.41**	0.35	0.49	0.41	**46.4**	0.38	**0.46**	**0.50**	**0.94**	**0.39**
GPT-2$_{\text{LARGE}}$																				
FINE-TUNE	68.5	8.78	46.0	69.9	2.45	**65.3**	43.1	55.5	**0.46**	0.38	**0.42**	**0.33**	0.53	0.42	**47.0**	**0.39**	0.46	**0.51**	**0.94**	**0.40**
Prefix	**70.3**	**8.85**	**46.2**	**71.7**	**2.47**	63.4	**47.7**	**56.3**	0.45	**0.39**	**0.42**	0.34	**0.48**	**0.40**	46.7	**0.39**	**0.45**	**0.51**	**0.94**	**0.40**
SOTA	68.6	8.70	45.3	70.8	2.37	63.9	52.8	57.1	0.46	0.41	0.44	-	-	-	-	-	-	-	-	-

只需要0.1%的参数， prefix tuning可以达到和全参数微调相当的效果

增加参数微调 Additive Fine-tuning

总结延伸

适配器微调 Adapters

- 节省了训练所需的参数量，训练速度快
- 增加了模型参数量，推理时会变慢，推理时显存开销变大

软提示微调 Soft prompt

- 需要优化的参数量很少
- 推理时可以方便地分别加载模型和soft prompt，对不同任务使用不同的soft prompt
- 训练时收敛速度慢
- 不同任务最优的prompt长度不一致

选择性微调 Selective Fine-tuning

只选择原始模型中的一小部分参数微调

如何选择?

选择性微调 Selective Fine-tuning

DiffPruning

学习模型参数中有哪些参数需要更新

将新的参数表示为预训练参数和稀疏的更新参数的和

通过约束更新参数的L0-norm实现

$$\theta_{task} = \theta_{pretrained} + \delta_{task}$$

$$\min_{\boldsymbol{\delta}_\tau} L(\mathcal{D}_\tau, f_\tau, \boldsymbol{\theta} + \boldsymbol{\delta}_\tau) + \lambda R(\boldsymbol{\theta} + \boldsymbol{\delta}_\tau)$$

任务优化目标

L0-norm
约束优化目标

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

L0-norm没有梯度无法直接优化，可以使用L1-norm或 Hard-Concrete distribution做可微近似

Parameter-Efficient Transfer Learning with Diff Pruning Demi Guo, Alexander M. Rush, Yoon Kim

选择性微调 Selective Fine-tuning

DiffPruning效果

- 相比adapters优化使用了更少的参数，实现了和全参数微调相当的效果
- 直接选择最后一层微调会有较大的性能损失

	Total params	New params per task	QNLI*	SST-2	MNLI_m	MNLI_{mm}	CoLA	MRPC	STS-B	RTE	QQP	Avg
Full finetuning	9.00×	100%	91.1	94.9	86.7	85.9	60.5	89.3	87.6	70.1	72.1	80.9
Adapters (8-256)	1.32×	3.6%	90.7	94.0	84.9	85.1	59.5	89.5	86.9	71.5	71.8	80.4
Adapters (64)	1.19×	2.1%	91.4	94.2	85.3	84.6	56.9	89.6	87.3	68.6	71.8	79.8
Full finetuning	9.00×	100%	93.4	94.1	86.7	86.0	59.6	88.9	86.6	71.2	71.7	80.6
Last layer	1.34×	3.8%	79.8	91.6	71.4	72.9	40.2	80.1	67.3	58.6	63.3	68.2
Non-adap. diff pruning	1.05×	0.5%	89.7	93.6	84.9	84.8	51.2	81.5	78.2	61.5	68.6	75.5
Diff pruning	1.05×	0.5%	92.9	93.8	85.7	85.6	60.5	87.0	83.5	68.1	70.6	79.4
Diff pruning (struct.)	1.05×	0.5%	93.3	94.1	86.4	86.0	61.1	89.7	86.0	70.6	71.1	80.6

选择性微调 Selective Fine-tuning

Bitfit

在注意力层和前馈层中，只有偏差项。被更新。（只更新红色部分）

$$\mathbf{Q}^{m,\ell}(\mathbf{x}) = \mathbf{W}_q^{m,\ell}\mathbf{x} + \mathbf{b}_q^{m,\ell}$$
$$\mathbf{K}^{m,\ell}(\mathbf{x}) = \mathbf{W}_k^{m,\ell}\mathbf{x} + \mathbf{b}_k^{m,\ell}$$
$$\mathbf{V}^{m,\ell}(\mathbf{x}) = \mathbf{W}_v^{m,\ell}\mathbf{x} + \mathbf{b}_v^{m,\ell}$$
$$\mathbf{h}_1^{\ell} = att(\mathbf{Q}^{1,\ell}, \mathbf{K}^{1,\ell}, \mathbf{V}^{1,\ell}, .., \mathbf{Q}^{m,\ell}, \mathbf{K}^{m,\ell}, \mathbf{V}^{m,l})$$

注意力层

$$\mathbf{h}_2^{\ell} = \text{Dropout}(\mathbf{W}_{m_1}^{\ell} \cdot \mathbf{h}_1^{\ell} + \mathbf{b}_{m_1}^{\ell})$$
$$\mathbf{h}_3^{\ell} = \mathbf{g}_{LN_1}^{\ell} \odot \frac{(\mathbf{h}_2^{\ell} + \mathbf{x}) - \mu}{\sigma} + \mathbf{b}_{LN_1}^{\ell}$$
$$\mathbf{h}_4^{\ell} = \text{GELU}(\mathbf{W}_{m_2}^{\ell} \cdot \mathbf{h}_3^{\ell} + \mathbf{b}_{m_2}^{\ell})$$
$$\mathbf{h}_5^{\ell} = \text{Dropout}(\mathbf{W}_{m_3}^{\ell} \cdot \mathbf{h}_4^{\ell} + \mathbf{b}_{m_3}^{\ell})$$
$$\text{out}^{\ell} = \mathbf{g}_{LN_2}^{\ell} \odot \frac{(\mathbf{h}_5^{\ell} + \mathbf{h}_3^{\ell}) - \mu}{\sigma} + \mathbf{b}_{LN_2}^{\ell}$$

前馈层

甚至只更新这两个部分的参数

BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models Elad Ben Zaken, Shauli Ravfogel, Yoav Goldberg

选择性微调 Selective Fine-tuning

Bitfit效果

使用较少的参数实现了和全参数微调相当的效果

	% Param	QNLI	SST-2	MNLI_m	MNLI_mm	CoLA	MRPC	STS-B	RTE	QQP	Avg.
Full-FT	100%	**90.7±0.2**	92.0±0.4	**83.5±0.1**	**83.7±0.3**	56.4±0.9	89.0±1.0	88.9±0.7	70.5±0.6	**87.1±0.1**	82.3
BitFit	0.09%	90.2±0.2	**92.1±0.3**	81.4±0.2	82.2±0.2	**58.8±0.5**	**90.4±0.5**	**89.2±0.2**	**72.3±0.9**	84.0±0.2	**82.4**
$\mathbf{b}_{m2}, \mathbf{b}_q$	0.04%	89.4±0.1	91.2±0.2	80.4±0.2	81.5±0.2	57.4±0.8	89.0±0.2	88.4±0.1	68.6±0.6	83.7±0.2	81.1
$\mathbf{b}_{m2}$	0.03%	88.9±0.1	91.1±0.3	79.9±0.3	80.7±0.2	54.9±0.9	87.9±0.6	88.2±0.1	66.8±0.6	82.1±0.4	80.0
$\mathbf{b}_q$	0.01%	86.8±0.1	89.6±0.2	74.4±0.3	75.7±0.2	49.1±1.5	84.4±0.2	85.6±0.1	61.4±1.1	80.6±0.4	76.6
Frozen	0.0%	68.7±0.3	81.7±0.1	42.4±0.1	43.8±0.1	31.9±1.1	81.1±0.1	71.4±0.1	56.9±0.4	62.4±0.2	62.1
rand uniform	0.09%	87.8±0.3	90.5±0.3	78.3±0.3	78.8±0.2	54.1±1.0	84.3±0.3	87.2±0.4	62.9±0.9	82.4±0.3	78.5
rand row/col	0.09%	88.4±0.2	91.0±0.3	79.4±0.3	80.1±0.3	53.4±0.6	88.0±0.7	87.9±0.2	65.1±0.7	82.3±0.2	79.5

选择性微调 Selective Fine-tuning

Bitfit效果

相比Diff-Pruning使用更少的参数实现了相当的效果

		%Param	QNLI	SST-2	$MNLI_m$	$MNLI_{mm}$	CoLA	MRPC	STS-B	RTE	QQP	Avg.
	Train size		105k	67k	393k	393k	8.5k	3.7k	7k	2.5k	364k	
(V)	Full-FT†	100%	**93.5**	**94.1**	**86.5**	**87.1**	**62.8**	**91.9**	89.8	71.8	**87.6**	**84.8**
(V)	Full-FT	100%	91.7±0.1	93.4±0.2	85.5±0.4	85.7±0.4	62.2±1.2	90.7±0.3	**90.0±0.4**	**71.9±1.3**	87.5±0.4	84.1
(V)	Diff-Prune†	0.5%	**93.4**	**94.2**	**86.4**	**86.9**	63.5	91.3	89.5	71.5	**86.6**	**84.6**
(V)	BitFit	0.08%	91.4±2.4	93.2±0.4	84.4±0.2	84.8±0.1	**63.6±0.7**	**91.7±0.5**	**90.3±0.1**	**73.2±3.7**	85.4±0.1	84.2
(T)	Full-FT‡	100%	91.1	**94.9**	86.7	85.9	**60.5**	**89.3**	**87.6**	70.1	**72.1**	**81.8**
(T)	Full-FT†	100%	**93.4**	94.1	86.7	**86.0**	59.6	88.9	86.6	**71.2**	71.7	81.5
(T)	Adapters‡	3.6%	90.7	94.0	84.9	85.1	59.5	89.5	**86.9**	71.5	**71.8**	81.1
(T)	Diff-Prune†	0.5%	**93.3**	94.1	**86.4**	**86.0**	**61.1**	**89.7**	86.0	70.6	71.1	**81.5**
(T)	BitFit	0.08%	92.0	**94.2**	84.5	84.8	59.7	88.9	85.5	**72.0**	70.5	80.9

选择性微调 Selective Fine-tuning

总结延伸

选择性微调选择的参数可以人工指定也可以通过学习得到

学习的方法并不总是比人工指定的好

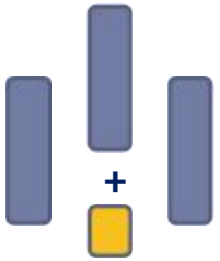
DiffPruning相比全参数微调需要更多的显存开销，因此比较大的模型可能难以使用

重参数化微调 Reparameterized Fine-tuning

- 微调时可能只需要优化原始参数空间中的一个小的子空间就可以实现优化目标
- 把参数用维度较低的代理参数来表示，只更新代理参数来节省计算量
- 如何选取代理参数?

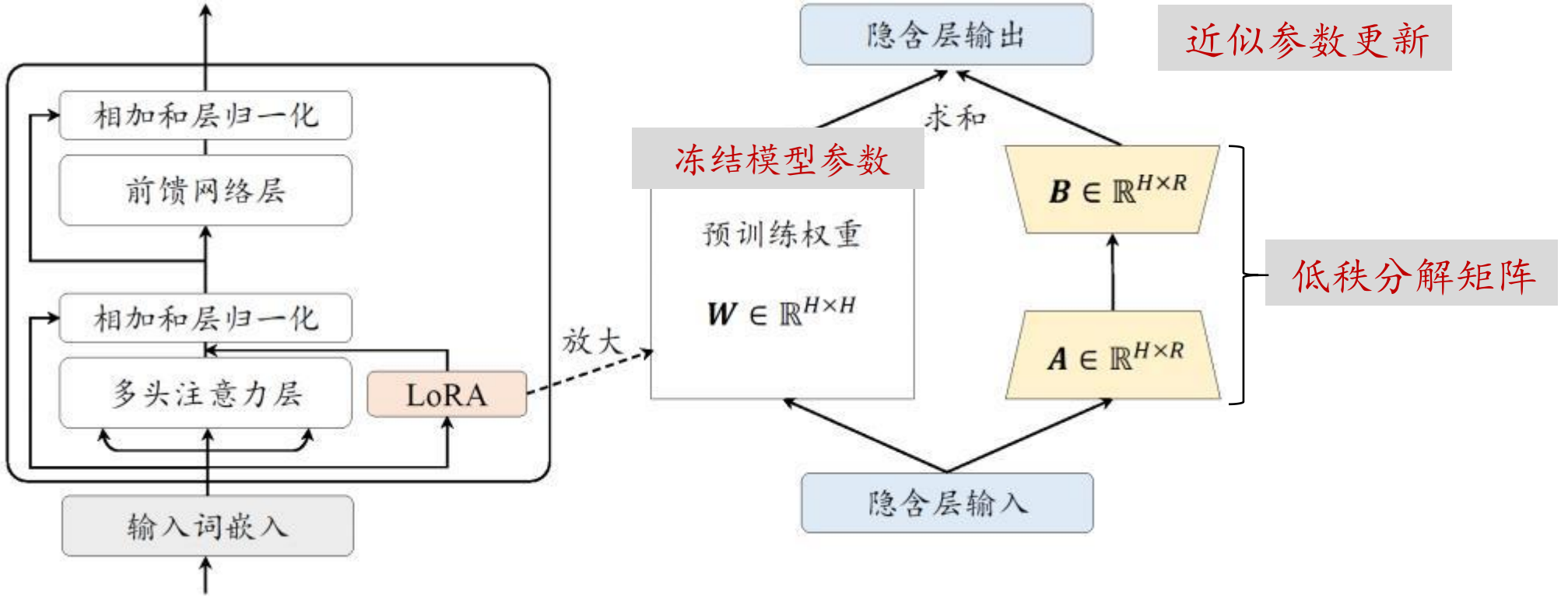
重参数化微调

Reparameterized Fine-tuning



Low-Rank Adaptation (LoRA)：最为流行的参数高效微调方法

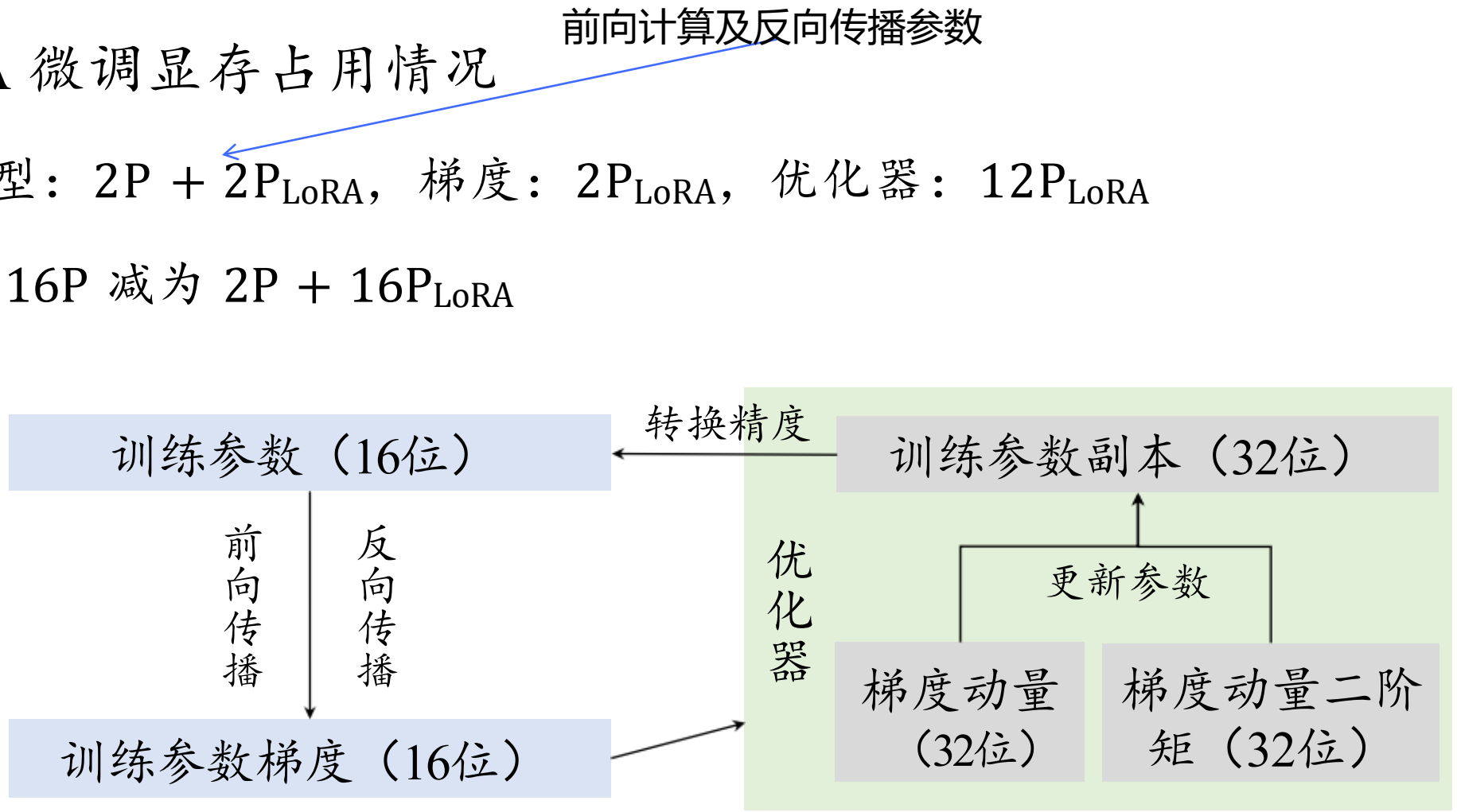
- LoRA 更新参数w 过程如下
  - $w \leftarrow w + \Delta w = w + A \cdot B^T$



LoRA: Low-Rank Adaptation of Large Language Models, **ICLR 2022**

Low-Rank Adaptation (LoRA)

- LoRA 微调显存占用情况
  - 模型：$2P + 2P_{LoRA}$，梯度：$2P_{LoRA}$，优化器：$12P_{LoRA}$
  - 从 $16P$ 减为 $2P + 16P_{LoRA}$



Mixed Precision Training, **ICLR 2018**

Low-Rank Adaptation (LoRA)

LoRA 用于注意力层的线性变换 W^K和W^V

$P_{LoRA} = 2 * 2LHR + 2 * 2LHR = 8LHR$（R通常取16）

以 LLaMA 7B 为例，$P \approx 6.7 \times 10^9$，$P_{LoRA} \approx 1.7 \times 10^7$，$P_{LoRA} \ll P$

模型和优化器占用从 16P 降至 2P

3090 24G 可以微调 7B 模型

QLoRA：量化参数矩阵， 用 4 比特存储模型参数

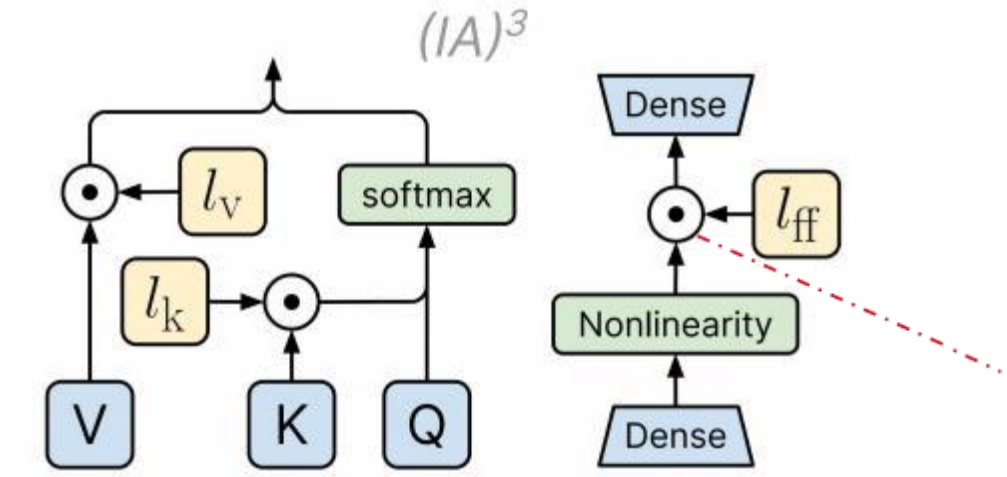
模型占用从 2P 降至 0.5P

A6000 48G 可以微调 65B 模型

重参数化微调 Reparameterized Fine-tuning

Infused Adapter by Inhibiting and Amplifying Inner Activations (IA3)

- 使用一个可学习的向量重新缩放内部激活
- 将这些学习到的向量注入到transformer的attention和feedforward模块中
- 这些学习到的向量是微调过程中唯一可训练的参数。

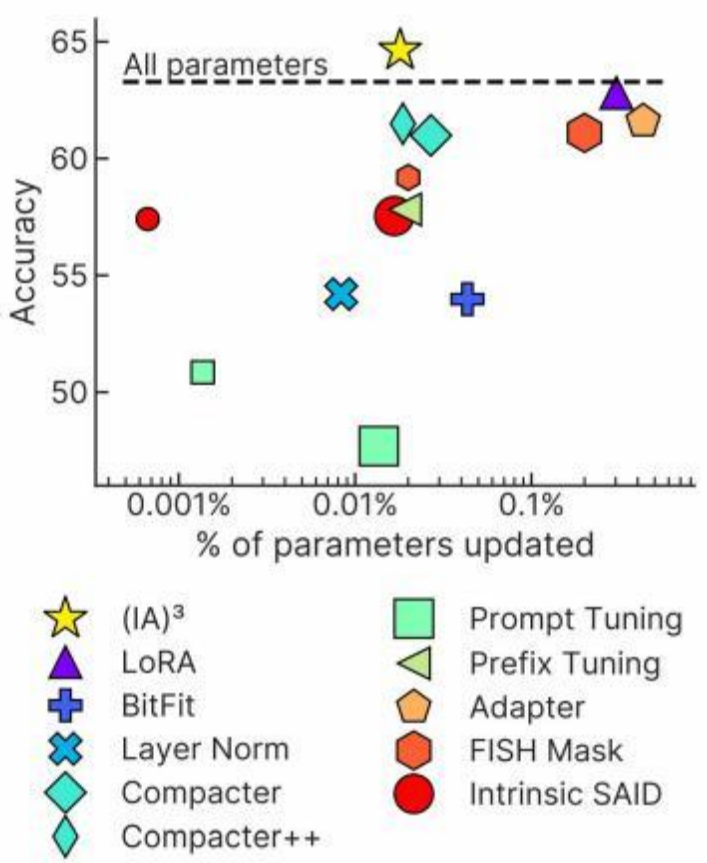


Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning
Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, Colin Raffel

重参数化微调 Reparameterized Fine-tuning

Infused Adapter by Inhibiting and Amplifying Inner Activations (IA3)

- 显著降低了需要训练的参数量
- 表现出更好的微调效果



重参数化微调 Reparameterized Fine-tuning

总结延伸

显著降低了需要训练的参数量

在许多任务上表现出较好的效果，加快了训练速度

LoRA可以和量化结合实现更好的效果

经验上，在某些任务上LoRA loss下降比较慢

其他重参数化微调方法

- Qlora: T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms
- Dora: DoRA: Weight-Decomposed Low-Rank Adaptation Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Min-Hung Chen
- LoKa: Navigating Text-To-Image Customization: From LyCORIS Fine-Tuning to Model Evaluation Shih-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard B W Yang, Giyeong Oh, Yanmin Gong
- LoHa: FedPara: Low-Rank Hadamard Product for Communication-Efficient Federated Learning Nam Hyeon-Woo, Moon Ye-Bin, Tae-Hyun Oh
- AdaLoRA: AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, Tuo Zhao
- Laplace-LoRA: Bayesian Low-rank Adaptation for Large Language Models Adam X. Yang, Maxime Robeyns, Xi Wang, Laurence Aitchison

参数高效微调

最常用的库 huggingface-peft

ADAPTERS

- AdaLoRA
- IA3
- Llama-Adapter
- LoHa
- LoKr
- LoRA
- X-LoRA
- LyCORIS
- Multitask Prompt Tuning
- OFT
- BOFT
- Polytropon
- P-tuning
- Prefix tuning
- Prompt tuning
- Layernorm tuning
- VeRA
- FourierFT
- VB-LoRA

https://huggingface.co/docs/peft
https://github.com/huggingface/peft

State-of-the-art Parameter-Efficient Fine-Tuning (PEFT) methods

根据自己的任务尝试并选择效果最好的

03

总结与思考

2025

2025

谢谢大家

时间：202X.X