在Qwen-2.5-0.5B模型上的微调

袁一木
522030910149

邬天行
522030910206

霍宗纬
522030910171

1 目录

- 任务与计算平台介绍
- 数据集与评测任务简介
- 模型与微调过程介绍
- 评测结果展示与一些思考
- 成员分工

2 任务介绍

本次实验，我们对Qwen-2.5-0.5b进行指令微调（SFT，supervised Fine-Tuning，也称为有监督微调），采用数据集为alpaca-cleaned，并将在多个任务上（涵盖考试、推理、知识等多方面任务）上进行评测，来评估我们的有监督微调效果。

我们选用了目前国内比较热门的付费计算平台AutoDL。本次试验我们租用了该平台上的一张显存为24GB的RTX4090D显卡（性能略微劣于RTX4090）进行微调。

3 数据集与评测任务简介

3.1 数据集介绍

本次进行SFT微调采用的数据集为Alpaca-cleaned数据集。在SFT中，训练所需要的数据集主要有三类：(1) 问答类型。主要存储方式为excel或csv，一般包含两个字段：input（输入）和Llabel（标签）。（2）指令-响应格式，一般会有以下几个字段：instruction（指令），input（输入），output（输出），system（系统提示词），history（上下文信息）等。（3）偏好数据集，一般用于奖励模型训练，训练模型偏好。会拥有chosen和rejected字段。(Wang et al., 2023)本次选择的alpaca-cleaned数据集属于第二种类型。

数据集	描述
MMLU	中学及大学的、各领域的考试题
HellaSwag	选择最合适的文本续写内容
WinoGrande	辨别句子中的指代对象
ARC	常识与推理问题
BoolQ	带有上下文的知识问答

表格 1: 数据集及其简单描述

3.2 评测任务简介

本次评测任务有以下几种类型： 我们将会分别在base（基底）模型和finetuned（微调）模型上分别进行测评。测评结果如Table 2所示。

4 模型与微调过程

4.1 模型介绍

本次微调的模型为Qwen-2.5-0.5b，是经典的decoder语言模型。它是一个基于带有SwiGLU激活、注意力QKV偏置、组查询注意力、滑动窗口注意力与全注意力混合等特性的Transformer架构。具有在本次试验中，我们没有对默认的模型参数进行调整，直接按照原来的模型参数加载模型。模型的大致结构如下：

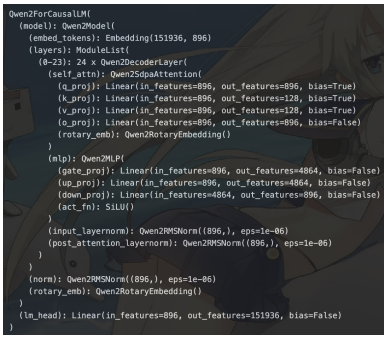

Figure 1: Qwen-2.5-0.5b

4.2 微调过程与思考

在微调过程中，最为重要和关键的就是理解输入数据的处理和标签的处理。对于输入数据的

dataset	type	base_acc	masked_acc	unmasked_acc
hellaswag	acc - clean	**46.57**	43.04	44.18
hellaswag	acc - input contaminated	39.29	**42.86**	39.29
hellaswag	acc - input-and-label contaminated	**51.06**	44.22	46.05
winogrande	acc	**54.38**	51.22	52.33
ARC-e	acc	45.86	44.09	**46.03**
ARC-c-test	acc - clean	29.36	**30.81**	30.57
ARC-c-test	acc - input contaminated	30.19	**37.74**	32.08
ARC-c-test	acc - input-and-label contaminated	33.45	34.52	**35.94**
BoolQ	acc	61.31	**62.39**	58.69
mmlu-humanities	naive_average	**51.34**	47.12	46.41
mmlu-stem	naive_average	**44.04**	41.45	40.24
mmlu-social-science	naive_average	**55.87**	50.92	50.2
mmlu-other	naive_average	**51.04**	49.66	47.63
mmlu	naive_average	**49.79**	46.61	45.43
mmlu-weighted	weighted_average	**47.86**	44.48	43.68

表格 2: 典型数据集表现。masked列仅计算了output的loss。

处理会在一定程度上影响到最终微调的效果。

4.2.1 输入数据的处理

首先，通过阅读相关文档，我们可以了解到输入数据需要的类型。传入模型的数据需要至少有三个字段：input_ids, attention_mask和label的input_ids. 输入数据我们采用prompt + Instruction + Input + Output的方式来构造输入数据。接着，借助tokenize函数将其转换成可以用于输入的整数类型的tensor向量。

为了提升模型处理的速度，我们需要将多个输入数据批次化处理。那么此时对于输入数据就需要拥有相同的长度。为了使得模型能够关注到有意义的tokens而非用于填充保持相同长度的input_ids，tokenizer引入了两个字段：attention_mask和token_type_ids。前者告诉模型在对应张量中需要注意的input_ids，后者告诉模型该token属于哪一个序列(sequence)。

此外，为了维持句子的长度，填充方式(pad)和截断方式(truncation)都十分重要。我们需要采用的填充方式为左填充。原因与loss的计算相关。

4.2.2 Loss的计算与标签的处理

作为自回归语言建模任务，我们的模型所需要完成的任务是基于过去的输入来得到当前的输出。因此经典的这类语言模型需要将标签中，prompt+Instruction+Input部分mask掉，并且向右位移一步后计算loss。在我们调用的transformers库中，这类语言模型的loss内将不会计算-100作为掩码时出现的loss。因此我们将输入数据中除了Output部分的其他部分使用-100掩码处理后，将其作为label进行训练。这样，基于我们loss的计算方式，我们发现计

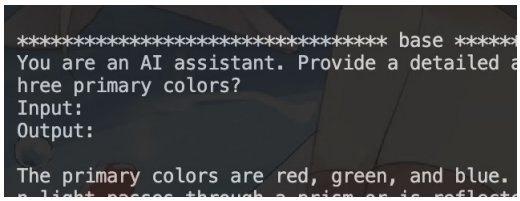

Figure 2: 原来的旧知识

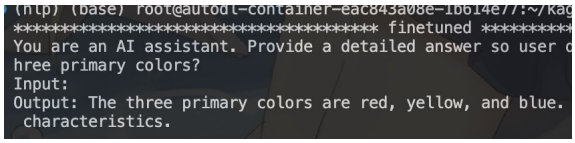

Figure 3: 微调后学会了新的知识

算loss时会将原来的输入向右进行位移。因此与许多decoder-only模型相似，采用左填充的方式。

4.2.3 模型微调前后的表现

我们可以比较Figure 2和Figure 3可以发现，微调前后模型学会了数据集中的新的知识。同时，模型微调后在数据集覆盖领域的表现变好，垂直领域的精确度增加，并且从Table 1中可以发现，微调后模型的抗干扰能力增强，但是通用能力和泛化能力下降。

4.3 结果比较

经过标签masked微调后，72项评测指标内有12项(16.67%)的指标优于原先的基底模

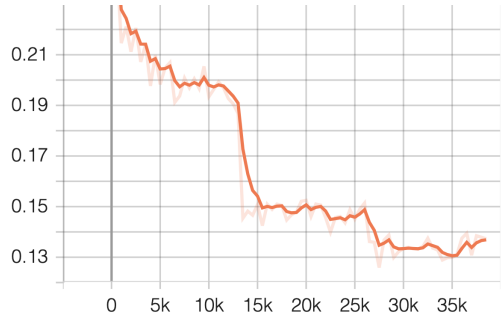

Figure 4: masked-输出loss训练结果

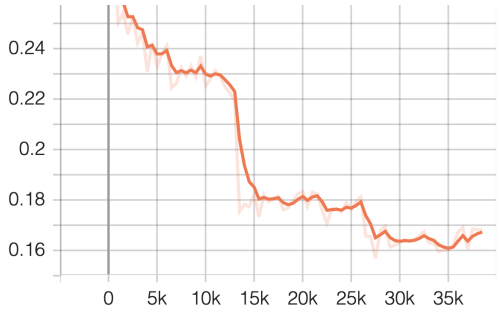

Figure 5: unmasked-输出loss训练结果

型。没有经过maksed标签微调后有11项（15.28%）的指标优于基底模型。一些选取的具有代表性的评测指标的结果如Table 2所示。

4.3.1 训练过程和训练细节

我们对模型进行了masked-output训练，即仅仅计算output的loss而不计算其他部分的loss。
采用长度为1024，训练批次大小为4，采用半浮点精度(bfloat16)来压缩显存。训练3个epoch共计38820步。
起始学习率5e-5，线性下降。每500步记录一次loss，如Figure 4所示。可以发现loss在经过第一个epoch后迅速收敛。具体训练细节可以参考tensorboard的事件记录。

4.3.2 仅**output**微调与全序列微调**loss**对比

接下来，我们需要了解没有经过output掩码后的loss。我们的训练结果如Figure 5所示。没有经过掩码操作的label会导致出现更高的loss（虽然差别不会很大）。这是因为模型会考虑到前三部分的预测误差，需要考虑到输入对于整个预测序列的影响。这样对于训练数据集覆盖的相关垂直领域将会具有较好的效果。但是模型的通用泛化能力有可能进一步下降，因为前面的没有经过掩码的部分会影响到整个序列。

4.4 对于**Table 2**结果的思考

1. 在MMLU数据集上，我们看到base模型总是会优于微调后的模型，尽管在某些领域中微调略微优于base模型。我们认为有可能微调使得模型学会了新的知识(Figure 2和Figure 3)，但是也会使得模型忘记原先具有的知识。（例如在光学中的三原色与在计算机视觉领域的三原色不相同，经过训练后模型发生了遗忘）

2. 经过训练后，我们发现仅计算output部分产生的loss时，模型的抗干扰能力增强。这是因为模型将不会考虑到输入对整个预测序列的影响，因此增强了模型的抗干扰能力。

3. 可以认为，基于该数据集对模型的训练将会使得模型学会新的知识，并且强化模型根据已经拥有的上下文和知识来预测结果的能力。但是这并不代表模型形成了完整的思维链，大部分的推理仍然是不正确的。这一点也在ARC评测结果中有所体现。

5 成员分工

在本次任务中，我们的分工是：

- 袁一木：编写微调代码、微调训练、微调后评测、分析实验结果、撰写报告
- 邬天行：环境配置、编写微调代码、微调后评测、分析实验结果
- 霍宗玮：环境配置、微调训练、微调后评测、撰写报告

项目地址：nlp-project-1

References

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.