LLM推理能力提升实验

1. 背景说明

1.1 LLM推理能力介绍

大语言模型（LLM, Large Language Model）的**推理能力是指模型根据已有信息（如上下文、背景知识、提示等），通过复杂的模式识别、逻辑推理和归纳总结，生成符合语义逻辑的回答或推断的能力。**

在自然语言处理任务中，推理能力可以帮助模型完成许多复杂任务，如回答问题、文本生成、总结信息等。**进一步地，可以延伸出教学助手、任务规划（Agent方向）等应用。**

1.2 当前LLM推理能力局限性

LLM的推理能力来源于其在海量数据上进行的预训练，通过模式识别和语言结构的学习，模型在归纳推理、演绎推理、常识推理、情境推理等方面展示出一定的能力。尽管这些能力不是传统的逻辑推理，但它们在自然语言处理任务中表现出强大的生成和理解能力。不过，LLM在推理能力上仍具有一定的局限性，其原因主要有

- **有限的因果推理**
  LLM 不是严格的逻辑推理系统，因此往往只能通过模式匹配进行推理，而不是真正理解因果关系。这意味着在面对真正复杂的因果链时，模型的推理能力可能会不可靠。
- **有限的数学和逻辑推理能力**
  LLM不是专门为数学推理设计的，在复杂的数学推理任务上性能有限，尤其在涉及多步计算或需要记住多个中间结果时表现不佳。
- **上下文长度限制**
  当上下文信息非常长时，模型的推理能力可能会受到影响，尤其在需要跨越长距离信息进行推理时。

1.3 实验研究内容

为提高LLM推理能力，需要额外的增强方法或技术手段来提升。同时，受限于有限的计算资源条件，以及实际应用要求，本实验拟在以下实验背景下，研究LLM推理能力提升方法。

1.3.1 实验背景

- 有限计算资源：2*4090 GPU
- 轻量级LLM：LLAMA-3.1-8B-Instruct

1.3.2 数据集

- 使用MATH数据集，验证数理推理能力，其主要原因有
  - 数理推理过程中，因果链较长，对该任务具备典型意义
  - MATH数据集在数理推理任务中，难度大，认可度高，具有横向对比意义

MATH数据集示例：

- **原始数据：**

{'problem': 'Let \\[f(x) = \\left\\{\n\\begin{array}{cl} ax+3, &\\text{ if }x>2, \\\\\nx-5 &\\text{ if } -2 \\le x \\le 2, \\\\\n2x-b &\\text{ if } x <-2.\n\\end{array}\n\\right.\\]Find $a+b$ if the piecewise function is continuous (which means that its graph can be drawn without lifting your pencil from the paper).',

'level': 'Level 5',

'type': 'Algebra',

'solution': 'For the piecewise function to be continuous, the cases must "meet" at 2 and -2. For example, $ax+3$ and $x-5$ must be equal when $x=2$. This implies $a(2)+3=2-5$, which we solve to get $2a=-6 \\Rightarrow a=-3$. Similarly, $x-5$ and $2x-b$ must be equal when $x=-2$. Substituting, we get $-2-5=2(-2)-b$, which implies $b=3$. So $a+b=-3+3=\\boxed{0}$.'}

- **对应上述数据的LaTex可视化数据：**

问题：

给定分段函数 $f(x)$：

$$f(x)=\begin{cases}ax+3 & \text{如果 } x>2\\ x-5 & \text{如果 } -2\le x\le 2\\ 2x-b & \text{如果 } x<-2\end{cases}$$

已知该分段函数是连续的（即函数图像可以不抬起笔一笔画出来）。求 $a+b$ 的值。

解答步骤：

为了使分段函数连续，各个分段在交界点 $x=2$ 和 $x=-2$ 处必须"接上"，即在这些点处，函数值要相等。

1. 在 $x=2$ 处：

在 $x=2$ 处，分段函数的两部分 $ax+3$ 和 $x-5$ 必须相等：

$$a(2)+3=2-5$$

$$2a+3=-3$$

解方程得到：

$$2a=-6 \quad \Rightarrow \quad a=-3$$

2. 在 $x=-2$ 处：

在 $x=-2$ 处，分段函数的两部分 $x-5$ 和 $2x-b$ 必须相等：

$$-2-5=2(-2)-b$$

$$-7=-4-b$$

解方程得到：

$$b=3$$

3. 计算 $a+b$：

将 $a=-3$ 和 $b=3$ 代入，计算：

$$a+b=-3+3=0$$

最终结果：

$$\boxed{0}$$

1.3.3 评价指标

Final Exact Match (Final EM) 指标是一种用于评估模型在自然语言处理任务（特别是问答任务、数学题解等任务）中的性能的评估指标。它主要用于测量模型预测结果与标准答案是否完全匹配。

- **指标简介**

EM (Exact Match) 代表**完全匹配**，它判断模型的预测答案与参考答案是否**一字不差**地完全相同。也就是说，如果预测结果和真实答案在所有细节上（包括词语、顺序、符号、空格等）都完全一致，则认为匹配成功。

- **计算方式**

1. **模型预测答案**：模型在输入数据上生成的预测输出。
2. **标准答案**：数据集中提供的正确答案。

对于每一个样本，如果模型的预测结果与标准答案**完全相同**，则该样本的 **EM 得分为 1**；否则为 0。最终的 Final EM 分数是所有样本的 EM 值的平均值，即：

$$\text{Final EM} = \frac{\text{完全匹配的样本数}}{\text{总样本数}}$$

- **示例**

如果标准答案是 `42`，那么模型生成的 `42` 或 `0042` 都会被视为完全匹配。但生成类似 `forty-two` 或 `42.0` 则不算完全匹配。

1.3.4 研究内容

- 有限资源下的LLM高效指令微调实践
- COT嵌入LLM推理过程，涉及以下核心内容
  - 词表修改
  - 基于MATH的COT数据集

2. 实验内容

2.1 有限资源下的LLM高效指令微调

2.1.1 预训练模型加载

如果使用FP32精度加载模型至GPU，实际需要~4*8=32G显存，此时一张24G的4090是放不下的，对此可以使用两种方案：

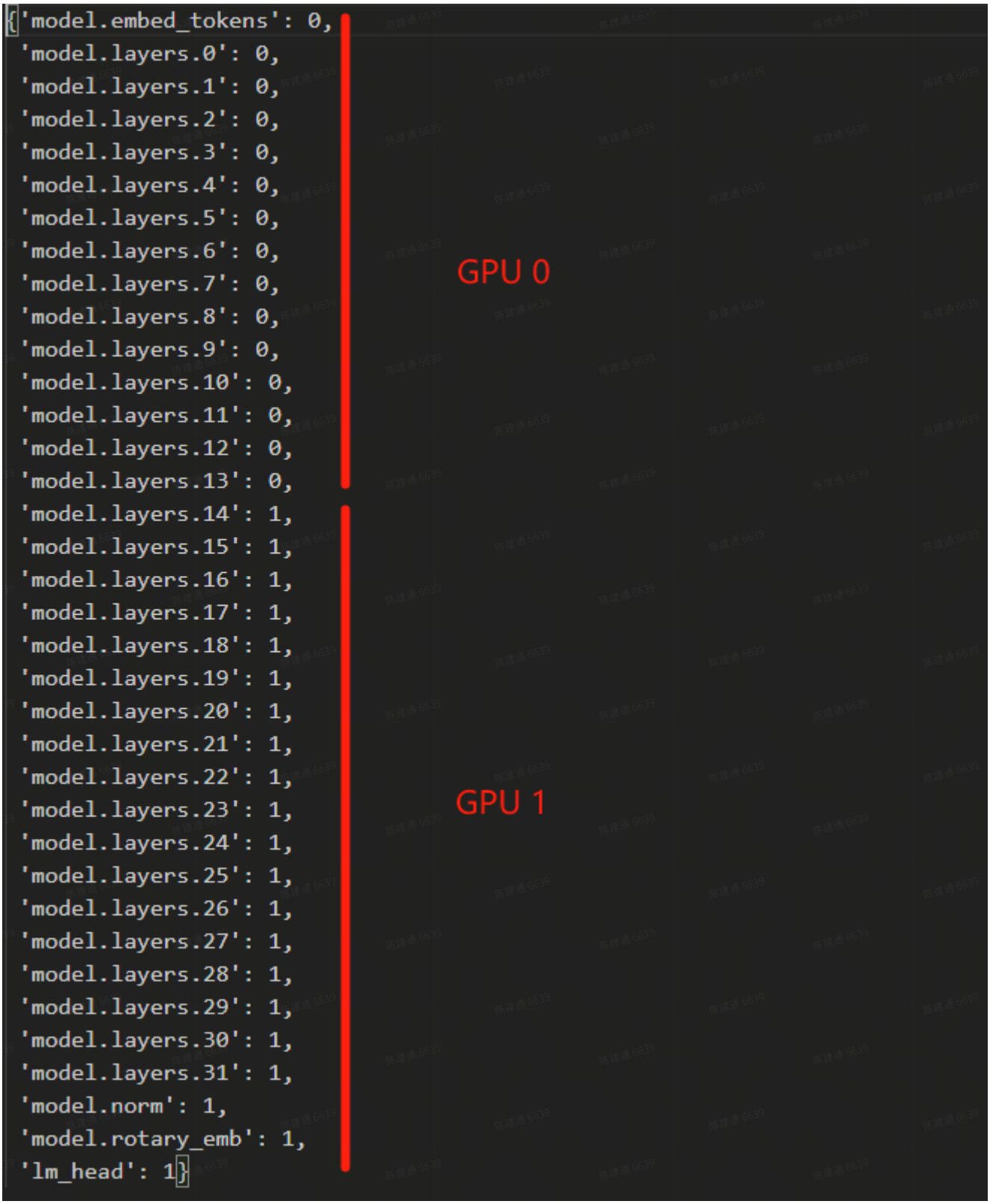
1、低精度加载模型

如果使用FP16精度加载，只需要~16G显存，可以完全加载至一张4090上。

```
model.state_dict()['model.layers.0.self_attn.q_proj.weight']
tensor([[ 0.0052, -0.0293, -0.0064,  ...,  0.0092, -0.0415, -0.0269],
        [-0.0150, -0.0679, -0.0059,  ..., -0.0149, -0.0498,  0.0197],
        [-0.0173, -0.0391, -0.0040,  ...,  0.0107, -0.0132,  0.0071],
        ...,
        [-0.0035, -0.0383,  0.0781,  ...,  0.0057, -0.0012,  0.0024],
        [-0.0033, -0.0093,  0.0437,  ...,  0.0047, -0.0011,  0.0012],
        [-0.0019, -0.0153,  0.0347,  ...,  0.0111,  0.0004,  0.0042]],
       device='cuda:0', dtype=torch.float16)
```

2、分层加载至多张卡上

可以将预训练模型不同层的权重参数，分别加载至多张卡上（本实验使用2张卡）。



本实验同时采用上述两种方案，降低每张卡的显存消耗，方便后续进行数据并行训练。

2.1.2 分词器加载与处理

分词器加载后，需要做如下处理：

- padding模式和token id

在LLAMA-3.1-8B-Instruct的分词器中，pad token id和padding方式均未指定，这是因为预训练时不需要padding，但后续微调和批量测试时，需要对不等长的数据进行扩充，因此对padding方式和token id进行指定。

- **padding_side指定为left：**在casual LLM中，如果指定右侧扩充，容易出现<|begin_of_text|>input content <pad><pad>...<pad><pad> output content <|eot_id|>情况，从而影响LLM预测结果，对此我们需要指定左侧填充
- **pad_token_id指定为预留特殊token：**按经验而言，pad_token_id可指定为0或eos_token_id，但在LLAMA-3.1-8B-Instruct的分词器中，0已有实际token，指定eos_token_id会在后续批量测试中，持续warning：" A decoder-only architecture is being used, but right-padding was detected! For correct generation results, please set padding_side='left' when initializing the tokenizer."（因为eos_token_id也被认为是pad_token_id，导致有此提醒，但实际不影响）。在本实验中，pad_token_id被指定为预留特殊token，比如<|reserved_special_token_0|>（128002）。

2.1.3 指令微调

指令微调是指在大型预训练语言模型的基础上，通过向模型提供特定任务的指令或提示（instruction）进行进一步微调，使模型更好地理解和执行各种复杂任务。

在本实验中，指令微调做法具体如下：

1、微调训练任务一：

将原始数据的**problem**填充至如下string中，假设该变量为string1

string1 = "<|start_header_id|>user<|end_header_id|>\n\n**{problem}**
<|start_header_id|>assistant<|end_header_id|>\n\n"

solution对应的回复内容，假设为string2

string2 = **solution**

将string1与string2拼接，进行编码，得到Input data，在指令微调中，我们只需要计算回复的loss，也就是string2部分，string1不参与损失计算，如果为多轮对话，则同样只计算回复部分的loss。

因涉及批量数据训练，同批次内不等长输入数据需要Padding操作，为避免pad token影响模型性能，需要加入attention mask变量，其长度与input data相同，元素为0/1，其中 1 表示有效的 token，0 表示 Padding 的 token。同时在计算损失时，忽略 Padding token。

示例：

- **原始数据**

{'problem': 'Let \\[f(x) = \\left\\{\n\\begin{array}{cl} ax+3, &\\text{ if }x>2, \\\\\nx-5 &\\text{ if } -2 \\le x \\le 2, \\\\\n2x-b &\\text{ if } x <-2.\n\\end{array}\n\\right.\\]Find $a+b$ if the piecewise function is continuous (which means that its graph can be drawn without lifting your pencil from the paper).',

'level': 'Level 5',

'type': 'Algebra',

'solution': 'For the piecewise function to be continuous, the cases must "meet" at 2 and -2. For example, $ax+3$ and $x-5$ must be equal when $x=2$. This implies $a(2)+3=2-5$, which we solve to get $2a=-6 \\Rightarrow a=-3$. Similarly, $x-5$ and $2x-b$ must be equal when $x=-2$. Substituting, we get $-2-5=2(-2)-b$, which implies $b=3$. So $a+b=-3+3=\\boxed{0}$.'}

- **经过拼接后的数据与标签（解码）**

```
tokenizer.decode(train_tokenizer_datasets[0]["input_ids"], skip_special_tokens=True)   → data：输入为problem和solution拼接内容
'Human: Let \\[f(x) = \\left\\{\n\\begin{array}{cl} ax+3, &\\text{ if }x>2, \\\\\nx-5 &\\text{ if } -2 \\le x \\le 2, \\\\\n2x-b &\\text{ if } x <-2.\n\\end{array}\n\\right.\\]Find $a+b$ if the piecewise
function is continuous (which means that its graph can be drawn without lifting your pencil from the paper).\n\nAssistant: For the piecewise function to be continuous, the cases must "meet" at $2$ and $-
2$. For example, $ax+3$ and $x-5$ must be equal when $x=2$. This implies $a(2)+3=2-5$, which we solve to get $2a=-6 \\Rightarrow a=-3$. Similarly, $x-5$ and $2x-b$ must be equal when $x=-2$. Substitutin
g, we get $-2-5=2(-2)-b$, which implies $b=3$. So $a+b=-3+3=\\boxed{0}$.'
tokenizer.decode(list(filter(lambda x:x!=-100, train_tokenizer_datasets[0]["labels"])))   → Label：只计算原始数据solution字段内容损失
'For the piecewise function to be continuous, the cases must "meet" at $2$ and $-2$. For example, $ax+3$ and $x-5$ must be equal when $x=2$. This implies $a(2)+3=2-5$, which we solve to get $2a=-6 \\Righ
tarrow a=-3$. Similarly, $x-5$ and $2x-b$ must be equal when $x=-2$. Substituting, we get $-2-5=2(-2)-b$, which implies $b=3$. So $a+b=-3+3=\\boxed{0}$.<|eot_id|>'
```

- **经过拼接后的数据与标签（编码）**

```
str(train_tokenizer_datasets[0]["input_ids"])
'[35075, 25, 6914, 1144, 58, 69, 2120, 8, 284, 1144, 2414, 59, 517, 59, 7413, 90, 1686, 15523, 566, 92, 3944, 10, 18, 11, 612, 59, 1342, 90, 422, 335, 87, 29, 17, 11, 91255, 87, 12, 20, 612, 59, 1342, 9
0, 422, 335, 482, 17, 1144, 273, 865, 1144, 273, 220, 17, 11, 91255, 17, 87, 1481, 612, 59, 1342, 90, 422, 335, 865, 9297, 17, 627, 59, 408, 90, 1686, 534, 59, 1315, 7255, 60, 10086, 400, 64, 36193, 3, 4
22, 279, 6710, 4583, 734, 374, 19815, 320, 8370, 3445, 430, 1202, 4876, 649, 387, 15107, 2085, 33510, 701, 47218, 505, 279, 5684, 3677, 72803, 25, 220, 2520, 279, 6710, 4583, 734, 311, 387, 19815, 11, 27
9, 5157, 2011, 330, 64510, 1, 520, 400, 17, 3, 323, 400, 12, 17, 13244, 1789, 3187, 11, 400, 710, 10, 18, 3, 323, 400, 87, 12, 20, 3, 2011, 387, 6273, 994, 400, 87, 28, 17, 13244, 1115, 24897, 400, 64,
7, 17, 7405, 18, 28, 17, 12, 20, 55976, 902, 584, 11886, 311, 636, 400, 17, 64, 11065, 21, 1144, 27338, 264, 11065, 18, 13244, 35339, 11, 400, 87, 12, 20, 3, 323, 400, 17, 87, 1481, 3, 2011, 387, 6273, 9
94, 400, 87, 11065, 17, 13244, 3804, 3781, 10831, 11, 584, 636, 400, 12, 17, 12, 20, 28, 17, 4172, 17, 7435, 65, 55976, 902, 24897, 400, 65, 28, 18, 13244, 2100, 400, 64, 36193, 11065, 18, 10, 18, 35533,
80175, 90, 15, 92, 13244, 128009]'
                                   → -100不参与loss计算，128009为 eos_token_id
str(train_tokenizer_datasets[0]["labels"])
'[-100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -10
0, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -1
00, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -
100, -100, -100, -100, -100, -100, -100, -100, 2520, 279, 6710, 4583, 734, 311, 387, 19815, 11, 279, 5157, 2011, 330, 64510, 1, 520, 400, 17, 3, 323, 400, 12, 17, 13244, 1789, 3187, 11, 400, 710, 10, 18,
3, 323, 400, 87, 12, 20, 3, 2011, 387, 6273, 994, 400, 87, 28, 17, 13244, 1115, 24897, 400, 64, 7, 17, 7405, 18, 28, 17, 12, 20, 55976, 902, 584, 11886, 311, 636, 400, 17, 64, 11065, 21, 1144, 27338, 26
4, 11065, 18, 13244, 35339, 11, 400, 87, 12, 20, 3, 323, 400, 17, 87, 1481, 3, 2011, 387, 6273, 994, 400, 87, 11065, 17, 13244, 3804, 3781, 10831, 11, 584, 636, 400, 12, 17, 12, 20, 28, 17, 4172, 17, 743
5, 65, 55976, 902, 24897, 400, 65, 28, 18, 13244, 2100, 400, 64, 36193, 11065, 18, 10, 18, 35533, 80175, 90, 15, 92, 13244, 128009]'
```

2、微调训练任务二：

将原数据的solution修改为更具格式化的COT数据，如下所示：

原数据的solution：

"Since $\\begin{pmatrix} a \\\\ b \\end{pmatrix}$ actually lies on $\\ell,$ the reflection takes this vector to itself.\n\n[asy]\nunitsize(1.5 cm);\n\npair D = (4,-3), V = (2,1), P = (V + reflect((0,0),D)*(V))\/2;\n\ndraw((4,-3)\/2--(-4,3)\/2,dashed);\ndraw((-2,0)--(2,0));\ndraw((0,-2)--(0,2));\ndraw((0,0)--P,Arrow(6));\n\nlabel(\"$\\ell$\", (4,-3)\/2, SE);\n[\/asy]\n\nThen\n\\[\\begin{pmatrix} \\frac{7}{25} & -\\frac{24}{25} \\\\ -\\frac{24}{25} & -\\frac{7}{25} \\end{pmatrix} \\begin{pmatrix} a \\\\ b \\end{pmatrix} = \\begin{pmatrix} a \\\\ b \\end{pmatrix}.\\]This gives us\n\\[\\begin{pmatrix} \\frac{7}{25} a - \\frac{24}{25} b \\\\ -\\frac{24}{25} a - \\frac{7}{25} b \\end{pmatrix} = \\begin{pmatrix} a \\\\ b \\end{pmatrix}.\\]Then $\\frac{7}{25} a - \\frac{24}{25}

b = a$ and $-\\frac{24}{25} a - \\frac{7}{25} b = b.$ Either equation reduces to $b = -\\frac{3}{4} a,$ so the vector we seek is $\\boxed{\\begin{pmatrix} 4 \\\\ -3 \\end{pmatrix}}.$"

新数据的solution：

"## Step 1: Understand the problem\nWe are given a matrix that represents the reflection over a certain line $\\ell$, which passes through the origin. We need to find the direction vector of line $\\ell$.\n\n

Step 2: Recall the property of reflection\nSince $\\begin{pmatrix} a \\\\ b \\end{pmatrix}$ actually lies on $\\ell$, the reflection takes this vector to itself.\n\n

Step 3: Write the equation for reflection\nThe reflection of the vector $\\begin{pmatrix} a \\\\ b \\end{pmatrix}$ is given by the matrix equation:\n\\[\\begin{pmatrix} \\frac{7}{25} & -\\frac{24}{25} \\\\ -\\frac{24}{25} & -\\frac{7}{25} \\end{pmatrix} \\begin{pmatrix} a \\\\ b \\end{pmatrix} = \\begin{pmatrix} a \\\\ b \\end{pmatrix}.\\]\n\n

Step 4: Simplify the equation\nThis gives us two equations:\n\\[\\frac{7}{25} a - \\frac{24}{25} b = a\\]\nand\n\\[-\\frac{24}{25} a - \\frac{7}{25} b = b.\\]\n\n

Step 5: Solve the first equation for b\nWe can simplify the first equation to get:\n\\[\\frac{7}{25} a - \\frac{24}{25} b = a\\]\n\\[\\frac{7}{25} a - a = \\frac{24}{25} b\\]\n\\[-\\frac{18}{25} a = \\frac{24}{25} b\\]\n\\[b = -\\frac{3}{4} a.\\]\n\n## Step 6: Find the direction vector\nSince $b = -\\frac{3}{4} a$, the vector we seek is $\\boxed{\\begin{pmatrix} 4 \\\\ -3 \\end{pmatrix}}.$\n\n

The final answer is: $\\boxed{\\begin{pmatrix} 4 \\\\ -3 \\end{pmatrix}}$"

在新数据基础上，将原始数据的**problem**填充至如下string中，**相比于微调训练任务一，该变化是在微调训练过程中，就直接使用COT提示，与测试时所用的prompt template保持了一致，期望提高模型本身COT能力，尤其是特殊提示词下的推理能力。**

string1 = "<|start_header_id|>user<|end_header_id|>\n\nSolve the following math problem efficiently and clearly:\n\n- For simple problems (2 steps or fewer):\nProvide a concise solution with minimal explanation.\n\n- For complex problems (3 steps or more):\nUse this step-by-step format:\n\n## Step 1: [Concise description]\n[Brief explanation and calculations]\n\n## Step 2: [Concise description]\n[Brief explanation and calculations]\n\n...\n\nRegardless of the approach, always conclude with:\n\nTherefore, the final answer is: $\\boxed\{answer\}$. I hope it is correct.\n\nWhere [answer] is just the final number or expression that solves the problem.\n\nProblem: **{problem}** <|start_header_id|>assistant<|end_header_id|>\n\n"

其他内容继承微调训练任务一，保持不变。

2.1.4 高效训练策略

在有限资源环境下，大致有三种训练策略，能够有效节省显存占用：

1、高效微调

例如BitFit、LORA等方法，这类方法往往采用冻结部分层，微调额外或原本模型的部分参数，达到微调策略。本实验采用LORA方法，**其训练参数量不足总参数量的0.05%**，且微调后的模型与原模型参数量保持一致，在后续多阶段训练后不改变模型大小：

```
trainable params: 3,407,872 || all params: 8,033,669,120 || trainable%: 0.0424
```

本实验同时采用GA或GC策略：

- Gradient Accumulation（GA）是指模型会在进行一次梯度更新前，累积n个小批次的梯度，表现出来的效果等同于将 batch size 增加到n倍，而不需要立即消耗大量内存。同时因为等效更大的 batch size，所以能提供更平滑的梯度更新，通常有助于模型更稳定地训练。
- Gradient Checkpointing（GC）是指模型**不保存每一层的中间激活值**。相反，**在需要反向传播时再重新计算这些激活值**。这样可以显著减少前向传播时的显存占用。

2、低精度微调

使用FP16精度进行微调，能节省大量显存开销，或者使用量化Int8或Int4精度微调，能大幅进一步降低。本实验为保证实验结果，采用FP16精度微调。

3、分布式训练

本实验采用流水并行策略，将模型部署至2张4090上，减少每张卡的显存消耗。同时采用数据并行策略，加速模型训练。

2.2 测试与分析

使用lm_eval工具箱测试，指令如下：

```
export CUDA_VISIBLE_DEVICES=4,7

lm_eval --model hf \
    --tasks minerva_math \
    --model_args pretrained=/nvme/wyc/Meta-Llama-3-8B-Instruct,parallelize=True,dtype="bfloat16" \
    --batch_size 128 \
    --apply_chat_template \
    --num_fewshot 0 \
    --gen_kwargs temperature=0,top_k=0,top_p=0,max_new_tokens=1024 \
    --seed 42 \
    --output_path ./result
```

测试细节在于：

- 使用COT chat_template：

"Solve the following math problem efficiently and clearly:\n\n- For simple problems (2 steps or fewer):\nProvide a concise solution with minimal explanation.\n\n- For complex problems (3 steps or more):\nUse this step-by-step format:\n\n## Step 1: [Concise description]\n[Brief explanation and calculations]\n\n## Step 2: [Concise description]\n[Brief explanation and calculations]\n\n...\n\nRegardless of the approach, always conclude with:\n\nTherefore, the final answer is: $\\boxed{answer}$. I hope it is correct.\n\nWhere [answer] is just the final number or expression that solves the problem.\n\nProblem:{**problem}**"

- zeroshot
- do_sample 置为Fasle，最大生成token数量修改为1024（并非5120，这是经过实际测试，超过1024个token对最终结果几乎无影响）
- 为保证顺利测试，使用bfloat16加载模型

2.2.1 原模型测试

```
2024-09-27:10:14:30,721 WARNING  [huggingface.py:1353] Failed to get model SHA for /nvme/wyc/Meta-Llama-3-8B-Instruct at revision main. Error: Repo id must be in the form 'repo_name' or 'namespac
e/repo_name': '/nvme/wyc/Meta-Llama-3-8B-Instruct'. Use `repo_type` argument if needed.
2024-09-27:10:14:34,109 INFO     [evaluation_tracker.py:206] Saving results aggregated
hf (pretrained=/nvme/wyc/Meta-Llama-3-8B-Instruct,parallelize=True,dtype=bfloat16), gen_kwargs: (temperature=0,top_k=0,top_p=0,max_new_tokens=1024), limit: None, num_fewshot: 0, batch_size: 128
|              Tasks               |Version|Filter|n-shot|  Metric   |   |Value |   |Stderr|
|----------------------------------|------:|------|-----:|-----------|---|-----:|---|-----:|
|minerva_math                      |      1|none  |      |exact_match|↑  |0.5056|±  |0.0066|
| - minerva_math_algebra           |      1|none  |     0|exact_match|↑  |0.7262|±  |0.0129|
| - minerva_math_counting_and_prob |      1|none  |     0|exact_match|↑  |0.4473|±  |0.0229|
| - minerva_math_geometry          |      1|none  |     0|exact_match|↑  |0.4217|±  |0.0226|
| - minerva_math_intermediate_algebra|      1|none  |     0|exact_match|↑  |0.2691|±  |0.0148|
| - minerva_math_num_theory        |      1|none  |     0|exact_match|↑  |0.4259|±  |0.0213|
| - minerva_math_prealgebra        |      1|none  |     0|exact_match|↑  |0.6877|±  |0.0157|
| - minerva_math_precalc           |      1|none  |     0|exact_match|↑  |0.3297|±  |0.0201|

|   Groups   |Version|Filter|n-shot|  Metric   |   |Value |   |Stderr|
|------------|------:|------|-----:|-----------|---|-----:|---|-----:|
|minerva_math|      1|none  |      |exact_match|↑  |0.5056|±  |0.0066|
```

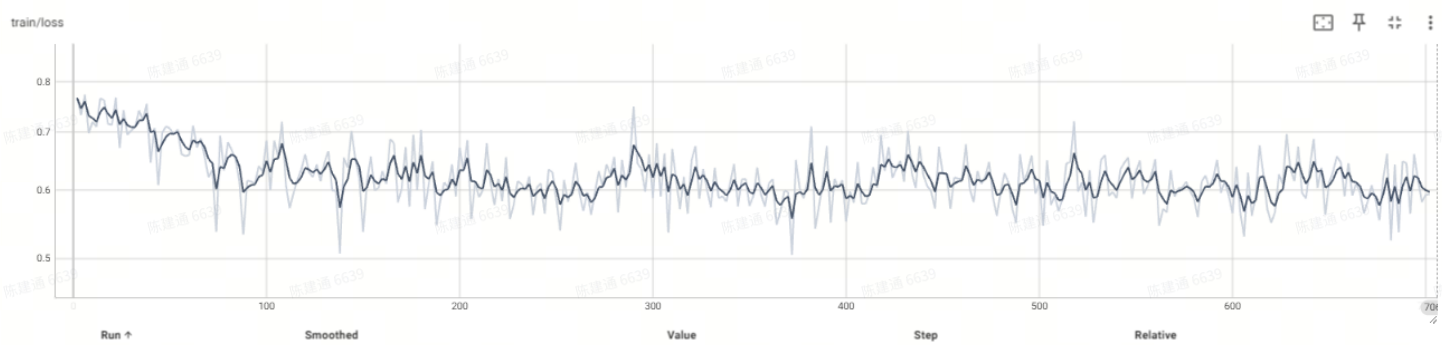
结论分析：

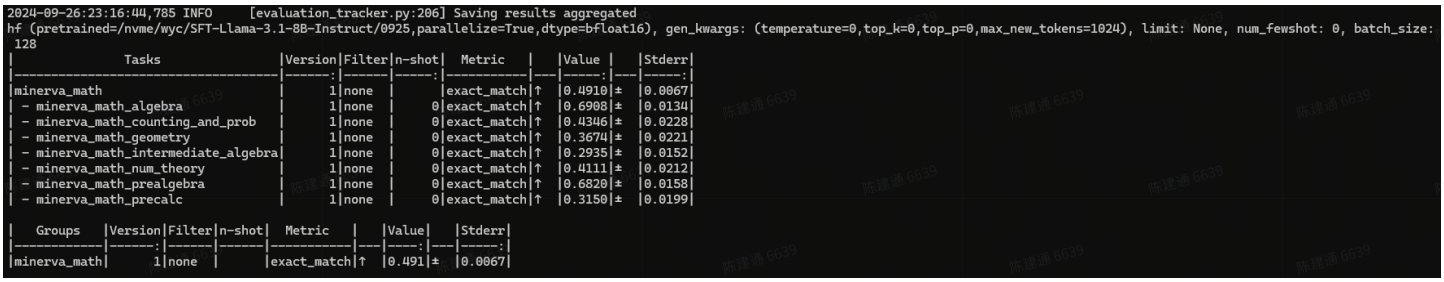
实测fanel_em指标50.56，低于官方51.9。猜测原因如下：

- pipline存在未知问题（下载meta-llama/Meta-Llama-3.1-8B-Instruct-evals官方测试使用的输入，输出与官方给出的输出不同，已知的pipline配置已保证完全一致，猜测存在官方未描述的细节差异）

因指标已接近，且do_sample 置为Fasle，保证每次测试效果一致，因此在相同测试条件下，**该指标具备基准意义**，在此基础上，我们进行实验比对。

2.2.2 微调训练任务一





结论分析：

该任务直接拼接了user和assistant两个身份信息后，开启20轮微调，训练loss有轻微下降，第三轮效果测试49.1，未见提升，目前认为有以下改进空间：

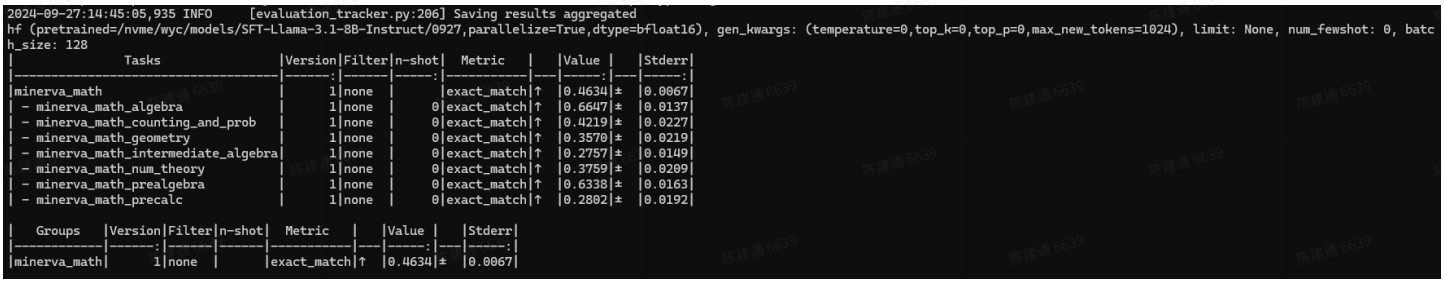
- 数据量少
- 训练数据内容非COT，使用了与测试不同的prompt
- 其他方面提高数学推理能力（在研究）

2.2.3 微调训练任务二

经过微调训练任务一，猜测存在过拟合的可能，该任务尝试数种方案消融及混合实验，方案有：

- Lora秩由8降至4，减少训练参数
- Lora参与训练的加入Dropout
- 优化器加大weight惩罚
- 输入数据扰动，比如混合COT和原始数据集

结果如下：最佳结果只有46.34，效果较差，在此基础上怀疑小批量数据破坏了模型参数分布，导致灾难性遗忘，进一步导致生成效果变弱。



2.2.4 微调训练任务三

观察目前微调情况，只针对所有self attention的K、V的weight进行Lora微调，结合微调训练任务二的经验，想验证改变其他层的效果对实验的影响。

方案一：只微调llama所有self atthention的Q、K、V的weight，猜测数理推理任务中，token之间关注强度有变化，期望提高生成能力。

方案二：只微调llama所有全连接层（也包括conv 1D层），猜测全连接层是影响任务类型的关键，注意力机制只获取token关联的一般性。

方案三：只微调llama最后一个LlamaDecoderLayer（第31个decode block），以及其后面的norm和lm header层，猜测修改头部网络，能提高小数据量下特定任务微调表现。

以上三个方案，只有方案三取得超越原模型的效果，其细节与结果如下

方案三的细节

- 只对lm header的权重，进行LORA训练，其余层采用正常全量微调

```
base_model.model.model.layers.29.input_layernorm.weight torch.bfloat16 False
base_model.model.model.layers.29.post_attention_layernorm.weight torch.bfloat16 False
base_model.model.model.layers.30.self_attn.q_proj.weight torch.bfloat16 False
base_model.model.model.layers.30.self_attn.k_proj.weight torch.bfloat16 False
base_model.model.model.layers.30.self_attn.v_proj.weight torch.bfloat16 False
base_model.model.model.layers.30.self_attn.o_proj.weight torch.bfloat16 False
base_model.model.model.layers.30.mlp.gate_proj.weight torch.bfloat16 False
base_model.model.model.layers.30.mlp.up_proj.weight torch.bfloat16 False
base_model.model.model.layers.30.mlp.down_proj.weight torch.bfloat16 False
base_model.model.model.layers.30.input_layernorm.weight torch.bfloat16 False
base_model.model.model.layers.30.post_attention_layernorm.weight torch.bfloat16 False
base_model.model.model.layers.31.self_attn.q_proj.weight torch.bfloat16 True
base_model.model.model.layers.31.self_attn.k_proj.weight torch.bfloat16 True
base_model.model.model.layers.31.self_attn.v_proj.weight torch.bfloat16 True
base_model.model.model.layers.31.self_attn.o_proj.weight torch.bfloat16 True
base_model.model.model.layers.31.mlp.gate_proj.weight torch.bfloat16 True
base_model.model.model.layers.31.mlp.up_proj.weight torch.bfloat16 True
base_model.model.model.layers.31.mlp.down_proj.weight torch.bfloat16 True
base_model.model.model.layers.31.input_layernorm.weight torch.bfloat16 True
base_model.model.model.layers.31.post_attention_layernorm.weight torch.bfloat16 True
base_model.model.model.norm.weight torch.bfloat16 True
base_model.model.lm_head.base_layer.weight torch.bfloat16 False
base_model.model.lm_head.lora_A.default.weight torch.float32 True
base_model.model.lm_head.lora_B.default.weight torch.float32 True
```

- 数据和标签：使用微调训练任务一的方式
- 训练参数量和配置如下

```
model.print_trainable_parameters()
[9] ✓ 0.0s
trainable params: 218,112,000 || all params: 8,032,378,880 || trainable%: 2.7154
```

```python
args = TrainingArguments(
    output_dir=output_dir,
    per_device_train_batch_size=2,
    gradient_accumulation_steps=16,
    gradient_checkpointing=True,
    learning_rate=1e-5,
    weight_decay=0.001,
    logging_steps=2,
    num_train_epochs=20,
    adam_epsilon=1e-4,
    save_total_limit=3,
    bf16=True,
    ddp_find_unused_parameters=False,
    dataloader_num_workers=4,
    dataloader_pin_memory=True,
    save_strategy="epoch",
    eval_strategy="no"
)

trainer = Trainer(
    model=model,
    args=args,
    train_dataset=train_tokenizer_datasets,
    data_collator=DataCollatorForSeq2Seq(tokenizer=tokenizer, padding=True)
)

trainer.train()
```

结论如下：

```
2024-10-03:08:40:54,306 INFO     [evaluation_tracker.py:206] Saving results aggregated
hf (pretrained=/nvme/wyc/models/SFT-Llama-3.1-8B-Instruct/1002,parallelize=True,dtype=bfloat16), gen_kwargs: (temperature=0,top_k=0,top_p=0,max_new_tokens=1024), limit: None, num_fewshot: 0, batch_size: 128
|             Tasks              |Version|Filter|n-shot|  Metric   |   |Value |   |Stderr|
|--------------------------------|------:|------|-----:|-----------|---|-----:|---|-----:|
|minerva_math                    |      1|none  |      |exact_match|↑  |0.5062|±  |0.0066|
| - minerva_math_algebra         |      1|none  |     0|exact_match|↑  |0.7254|±  |0.0130|
| - minerva_math_counting_and_prob|      1|none  |     0|exact_match|↑  |0.4473|±  |0.0229|
| - minerva_math_geometry        |      1|none  |     0|exact_match|↑  |0.4154|±  |0.0225|
| - minerva_math_intermediate_algebra|      1|none  |     0|exact_match|↑  |0.2724|±  |0.0148|
| - minerva_math_num_theory      |      1|none  |     0|exact_match|↑  |0.4315|±  |0.0213|
| - minerva_math_prealgebra      |      1|none  |     0|exact_match|↑  |0.6889|±  |0.0157|
| - minerva_math_precalc         |      1|none  |     0|exact_match|↑  |0.3297|±  |0.0201|

|   Groups   |Version|Filter|n-shot|  Metric   |   |Value |   |Stderr|
|------------|------:|------|------|-----------|---|-----:|---|-----:|
|minerva_math|      1|none  |      |exact_match|↑  |0.5062|±  |0.0066|
```