

【从零训练Steel-LLM】微调探索与评估

原创 战士金 炼钢AI 2024年10月24日 22:58 北京

① 前言

今年二月份，机缘巧合，朋友搞到了一台A100 80G SXM，机器放着也是怪浪费的，便萌生了从零预训练一个LLM的享法。一台机器不算多，并且最多可能也就用个3-4个月，掐指一算，训个1B左右的模型，1T左右的数据应该差不多。好景不长，机器用了一个多月吧，就被收回了，当时模型才训到了20k step（预计要训练100k step）。天无绝人之路，真的非常感谢某top 3老师的资助，支持了一个多月的一台H系列机器，才得以让我们的项目顺利完成。打工的牛马，工作日10点到家，亦或是周末，靠着兴趣，每天弄点，拖拖拉拉，转眼已经到了10月了，才弄出来一个自己觉得差不多还说的过去的模型。和其他模型不太一样，我们的模型预训练时以中文语料为主，大概只有20%左右的英文数据，因此就不在英文榜单上现眼了。微调以后，最终在CEVAL上获得了38分，CMMLU上获得了33分（这已经比一些大几倍的开源模型效果要好了。叠甲：开源数据的功劳，开源数据的功劳）。本篇文章主要介绍下微调上的探索以及评估。另外，还特意试了试训练CMMLU数据集，能在榜单上提多少分（狗头

之前也写了更详细的数据准备、训练框架改造、模型设计的文章：

【从零训练Steel-LLM】预训练数据收集与处理

【从零训练Steel-LLM】预训练代码讲解、改进与测试

【从零训练Steel-LLM】模型设计

github：

```
1 https://github.com/zhanishjinwat/Steel-LLM
```

项目也有交流群，满200人了，加 绿：a1843450905。

欢迎关注我的 zhi hu：战士金



炼钢AI

个人公众号，首本RAG相关书籍《大模型RAG实战》、开源预训练项目Steel-LLM作者， ...
7篇原创内容

公众号

② 微调数据

首先来介绍一下我们用的微调数据，主要目标是提高模型的对话能力和作题能力：

(1) BAAI/Infinity-Instruct

开源数据的话，还是选大机构发布的靠谱一点，毕竟大机构还是要面子的，除此之外尽量选择新一些的。我们的模型是8月20号左右训练完的，BAAI正好在8月初才发布了Infinity-Instruct 7M数据集，正好就拿过来用了。这个数据集还给出了语言标签、任务类型等元信息，方便进一步筛选数据。最开始我就一把梭，直接把这700w数据全都训进去，但是CEVAL

只有30出头的分数。后来才想起了这里边应该混有一些英文数据的，统计之后才发现只有70w左右的中文数据。**因为Steel-LLM预训练数据中80%都是中文数据，微调时候用大部分都是英文的数据微调效果当然是不太好的。**因此，最好版本的微调模型只用了70w的Infinity-Instruct的中文数据。

Infinity-Instruct微调数据集数量不少，也借此谈谈我对微调数据的看法。从很早开始，对于使用多一些的数据微调（《Exploring the Impact of Instruction Data Scaling on Large Language Models An Empirical Study on Real-World Use Cases》）还是精选几百、几千条数据微调（《LIMA: Less Is More for Alignment》）一直有争执。我觉得使用少量的微调数据得到一个好的sft模型前提是你的基础模型足够强大，如果模型比较弱的让它在微调阶段见识到更多的数据自然是好的。还有一种普遍说法是“预训练给模型注入知识，微调学习对话方式、激活知识”，经常会见到有人争执微调到底能不能注入知识的问题。我觉得只能说预训练的和微调的**主要目的**分别是注入知识和学习对话方式，微调和预训练一样，也是next token的训练，当然也能注入知识了。因此在微调Steel LLM时，我使用了百万级别的大量sft数据。

(2) wanjuan中文选择题部分

Steel-LLM预训练数据里边不仅有原始文本数据，还有诸如BELLE、moss等项目的对话数据（详细的数据组成可以看我之前的文章），**本意是想通过在预训练数据中混入少量sft数据，让预训练之后的模型直接有一定的对话能力。**但现实是，训练出来的模型续写能力和对话能力都不是很稳定。。。不能很好的输出选择题答案，有点四不像了。因此考虑在微调阶段加入一些选择题数据，规范一下回答格式。**其实wanjuan中文选择题部分在预训练阶段就有的**，Steel-LLM项目的pretrain_modify_from_TinyLlama/scripts/prepare_steel_llm_data.py文件里甚至还有专门处理这部分数据的逻辑，相当于在微调阶段再回炉重造一遍。wanjuan选择题的一条数据形式如下所示，除了问题、选项和答案以外，还有“解析”(answer_detail)。个人认为“解析”部分还是很重要的（如果没有解析，真的更像是背答案了），qwen等模型在回答选择题时候也会输出解析，说明他们的sft数据的选择题也是有解析的。

```
1 {'id': 'BkQQU-7xK3YAJdm0cWMc', 'q_type': '单选题', 'q_main': '下列属于同种物质的；
```

在构造选择题微调数据时，考虑cot和非cot两种格式，后文也会展示这两种微调方式的在选择题上的得分。cot格式先输出“解析”，再输出答案；非cot格式，先输出答案，再输出“解析”。具体格式如下：

cot格式：

```
1 # 替换“解析”里的答案部分，最后统一输出答案。
2 answer = f'{item["answer_detail"]}'
3 pattern = r"故选.*"
4 answer = re.sub(pattern, '', answer, flags=re.DOTALL)
5 answer = answer + f'答案为{item["std_ans"]}'
6 instruct = f"以下是一道{json_obj['q_type']}: \n{choice}请先给出解释再给出答案。 \n"
```

非cot格式：

```
1 answer = f'{item["std_ans"]}。{item["answer_detail"]}'
```

```
2 instruct = f"以下是一道单选题:\n{choice}请给出答案.\n"
```

(这里自我吐槽一下，这项目真的持续太久了，微调时候都忘记了预训练数据里还有选择题数据这回事。。。还是刷github时候看了survivi/Llama-3-SynE数据里有选择题，本来想用正则表达式从里边提取出选择题，后来嫌麻烦直接发邮件问作者选择题哪来的，才发现是wanjuan数据的选择题)

(3) ruozhiba

之前比较火的“弱智吧”微调数据，问题来源于百度贴吧“弱智吧”，由GPT4回答。我使用的优化版的“弱智吧”数据：Better-Ruozhiba，人为审阅了每一条的原文和回复，剔除了一些原文中的格式错误，修改或重写了部分答案。

(4) 自我认知数据

通过微调的方式让Steel-LLM知道自己是Steel-LLM，数据内容就是各种形式的问模型“你是谁”。模板来自于EmoLLM项目，地址如下，将原来数据中的“心理健康助手”替换为了“Steel-LLM”

```
1 https://github.com/SmartFlowAI/EmoLLM/blob/main/datasets/self_cognition_EmoLLM
```

③ 其他模型在ceval/cmmlu上的表现

这块列一下其他模型在benchmark上的表现，让读者大概有个分数的概念。我们的1B模型在中文榜单上的表现已经比一些更大的早期的模型效果好了，这得益于开源数据的进步。(当然，也有一些模型本身训练数据里中文数据就少，比如llama)

来自ceval论文的榜单：

Model	STEM	Social Science	Humanities	Other	Average
Random	25.0	25.0	25.0	25.0	25.0
GPT-4	65.2	74.7	62.5	64.7	66.4
ChatGPT	49.0	58.0	48.8	50.4	51.0
Claude-v1.3	48.5	58.6	47.3	50.1	50.5
Bloomz-mt	39.1	53.0	47.7	42.7	44.3
GLM-130B	36.7	55.8	47.7	43.0	44.0
Claude-instant-v1.0	38.6	47.6	39.5	39.0	40.6
ChatGLM-6B	33.3	48.3	41.3	38.0	38.9
LLaMA-65B	32.6	41.2	34.1	33.0	34.7
MOSS	31.6	37.0	33.4	32.1	33.1
Chinese-Alpaca-13B	27.4	39.2	32.5	28.0	30.9
Chinese-LLaMA-13B	28.8	32.9	29.7	28.0	29.6

来自MiniCPM榜单：<https://arxiv.org/pdf/2404.06395>

Model	C-Eval	CMMLU	MMLU	HumanEval	MBPP	GSM8K	MATH
Llama2-7B	32.42	31.11	44.32	12.20	27.17	13.57	1.80
Qwen-7B	58.96	60.35	57.65	17.07	42.15	41.24	5.34
Deepseek-7B	42.82	44.45	47.82	20.12	41.45	15.85	1.53
Mistral-7B	46.12	42.96	62.69	27.44	45.20	33.13	5.00
Gemma-7B	42.57	44.20	60.83	38.41	50.12	47.31	6.18
Llama2-13B	37.32	37.06	54.71	17.07	32.55	21.15	2.25
MPT-30B	29.34	32.09	46.56	21.95	35.36	10.31	1.56
Falcon-40B	40.29	41.57	53.53	24.39	36.53	22.44	1.92
TinyLlama-1.1B	25.02	24.03	24.3	6.71	19.91	2.27	0.74
Qwen-1.8B	49.81	45.32	43.37	7.93	17.8	19.26	2.42
Qwen1.5-1.8B	55.00	50.85	43.81	5.49	24.82	26.16	3.25
Gemini Nano-3B	-	-	-	-	27.20	22.80	-
StableLM-Zephyr-3B	30.34	30.89	45.90	35.37	31.85	52.54	12.12
Phi-2(2B)	23.37	24.18	52.66	47.56	55.04	57.16	3.50
Gemma-2B	29.26	28.56	38.49	24.39	29.74	16.83	3.34
MiniCPM-1.2B	49.14	46.81	49.63	44.51	32.75	31.77	10.60
MiniCPM-2.4B	51.13	51.07	53.46	50.00	47.31	53.83	10.24

来自MAP-Neo论文的榜单：
<https://arxiv.org/pdf/2405.19327>

Dataset	LLama-3-8B (Instruct)	Mistral-7B (Instruct-v0.2)	LLama-2-7B (Chat)	Amber-7B (Chat)	OLMo-7B (Instruct)	MAP-Neo-7B (SFT)	MAP-Neo-7B (Instruct)
Chinese							
C-Eval	50.76	43.72	35.67	26.29	35.18	55.42	56.97
*-stem	47.47	41.35	32.59	23.99	31.43	47.37	49.08
*-social-science	57.05	47.75	40.04	26.77	42.13	69.21	70.75
*-humanities	48.32	47.33	36.96	28.26	34.03	63.17	63.14
*-other	53.48	40.74	36.01	28.06	36.81	49.78	52.63
*-hard	31.04	27.32	28.45	22.77	26.33	38.41	39.55
CMMLU	51.68	42.67	33.9	30.09	35.55	55.27	55.01
*-humanities	52.55	42.01	35.45	30.48	34.78	63.4	62.99
*-stem	44.09	36.82	29.33	26.76	30.36	47.29	46.69
*-social-science	53.02	44.41	34.55	30.97	38.04	57.55	57.79
*-other	57.58	47.3	36.77	32.25	38.45	53.93	53.44
*-china-specific	45.86	39.22	32.64	28.38	33.97	55.69	55.9

④ 微调实验

我尝试了构造不同数据组成、格式的微调数据，查看模型在微调不同微调step下的ceval和cmmlu准确率，微调数据构造和评估代码也已经更新到了Steel LLM的项目中了。如果不特别说明的话，微调加载的是预训练1060k step的预训练模型，对数据微调3个epoch，学习率为2e-5，micro batch 大小为8，累计8个 micro batch进行一次反向传播（模拟单卡 batch size=64），wanjuan exam数据集默认不使用cot方式。每次实验都带着ruozhiba数据和自我认知数据，数据量很少，主要是对Infinity-Instruct数据和wanjuan exam数据上做探索（有的实验组没有cmmlu的评测结果）

实验1：全量Infinity-Instruct数据+全量wanjuan exam

随着微调step不断增加，ceval的指标是会上涨的，但微调到18000step之后变化就不大了。最好能拿到32%的准确率。因此有如下猜想：即使wanjuan exam在预训练里已经见过，但因为模型比较小，训练过的数据仍然会忘掉一些，微调多一些的数据还是有益的（小模型情况下）。

微调step	STEM	Social Science	Humanities	Other	Hard	AVERAGE (ceval)
2000	27.84	26.22	30.99	29.09	28.82	28.46
6000	25.97	24.12	31.48	28.44	27.77	27.30
12000	27.99	29.32	29.15	27.99	27.88	28.49
18000	30.51	30.54	33.92	30.19	33.72	31.17
24000	33.98	32.98	33.14	29.88	40.01	32.74
30000	32.57	31.97	35.01	31.82	35.56	32.81
36000	30.57	31.00	35.62	29.39	34.52	31.47
42000	31.79	30.49	34.71	30.70	35.68	31.93
46000	31.40	32.62	33.77	32.43	36.51	32.35

实验2：70w Infinity-Instruct中文数据+全量wanjuan exam

Infinity-Instruct全量数据有700w条左右，这里边有大概630w的英文对话数据。因为Steel-LLM在预训练时候80%都是中文，微调数据如果有很多英文是会影响到模型性能的，将Infinity-Instruct里边的英文数据全部去掉，只保留70w的中文数据能拿到更高的准确率。在ceval和cmmlu上的准确率如下所示，**这是Steel-LLM的最好的一版模型，ceval达到了38%的准确率，cmmlu达到了33%的准确率。**

ceval准确率：

微调step	STEM	Social Science	Humanities	Other	Hard	AVERAGE (ceval)
4000	27.95	30.89	29.03	30.08	29.18	29.20
8000	28.22	31.05	29.00	31.58	28.48	29.64
10000	29.93	35.40	32.86	34.38	28.93	32.55
12000	33.75	39.73	39.05	34.22	33.42	36.12
16000	34.65	42.33	43.13	35.90	34.52	38.18
20000	34.53	43.31	39.17	39.68	33.23	38.29
24000	34.45	41.30	41.20	38.64	34.48	38.09
27459	33.94	41.61	43.00	39.79	33.65	38.57

cmmlu准确率：

微调step	STEM	Humanities	Social Science	Other	China specific	Overall (CMMLU)
4000	25.37	26.10	25.46	25.63	25.83	25.60
8000	27.11	28.48	26.51	26.80	26.64	27.11
10000	27.41	29.91	29.65	30.97	31.10	29.43
12000	27.61	32.51	32.10	34.06	32.98	31.48
16000	27.46	33.96	33.34	34.11	33.81	32.14
20000	29.07	35.86	34.56	34.56	34.81	33.42
24000	29.07	35.15	34.08	35.52	35.02	33.34
27459	29.17	35.87	34.10	35.37	35.23	33.48

实验3：CMMLU刷榜测试

有一些工作（比如天工大模型的技术报告:<https://arxiv.org/pdf/2310.19341>）已经表明了目前各种榜单上排名靠前的某些模型有刷榜的行为（直接过拟合测试集），论文《Training on the Benchmark Is Not All You Need》也证明了这一点。笔者比较好奇，如果我直接去过拟合CMMLU测试数据，最多能在CMMLU上拿到多少分。

笔者在70w Infinity-Instruct中文数据+wanjuan选择题数据+ruozhiba数据+自我认知数据的基础上，**消融在训练数据中直接加入CMMLU数据对模型的影响**。（这块由于笔者的疏忽，训练加载的checkpoint是实验1训练出来的checkpoint而不是原始预训练出来的模型。。。通过CEVAL正确率没有进一步增长，也能说明在实验1训练3个epoch的基础上，再多训练几个epoch已经没啥收益了）

当训练集中加入CMMLU数据时，在CMMLU上测试可以拿到36%的正确率，而训练集中不加入CMMLU数据时，也能达到33%左右的正确率，说明过拟合数据集对于刷榜是有一定作用的，但是作用不是特别大（单纯对于小的模型来说，死记硬背都不能特别好记下来）。笔者这里在对CMMLU进行sft时只让模型去预测答案选项（搞不到解释），猜测如果sft时让模型去学习**选项+解释**能涨分更多。一些开源模型在做选择题时候，即使你告诉他只回答选项不要给出解释，很多时候依然会回答一大串解释。。。

虽然CMMLU和CEVAL题的类型大部分相同（STEM、social science等），但是对CMMLU进行过拟合并不能提升模型在CEAVL上的分数，说明在小模型上过拟合测试集的泛化性一般。

(1) 训练集中**加入CMMLU数据**
CEVAL

	STEM	Social Science	Humanities	Other	Hard	AVERAGE (ceval)
4000	28.26	30.75	32.48	27.97	28.34	29.57
8000	35.06	36.30	39.98	36.76	35.89	36.70
10000	31.88	40.21	39.41	35.78	29.27	35.90
12000	33.73	33.14	36.51	39.01	32.67	35.32
16000	36.19	36.93	38.34	38.66	35.75	37.31
20000	35.81	43.74	38.32	38.98	34.24	38.54
24000	36.23	41.33	42.84	37.83	34.99	38.95
27459	36.52	42.52	41.20	38.64	34.69	39.11

CMMLU

微调step	STEM	Humanities	Social Science	Other	China specific	Overall (cmmlu)
4000	32.54	39.28	36.86	37.67	37.41	36.42
8000	32.61	38.57	36.86	37.46	37.30	36.25
10000	32.80	38.84	37.06	37.38	37.92	36.40
12000	29.89	35.94	33.69	34.17	34.76	33.27
16000	30.29	38.58	35.75	37.73	37.08	35.36
20000	32.80	38.84	37.06	37.38	37.92	36.40
24000	32.61	38.57	36.86	37.46	37.30	36.25
27459	32.54	39.28	36.86	37.67	37.41	36.42

(2) 训练集中**不加入**CMMLU数据
CEVAL

微调step	STEM	Social Science	Humanities	Other	Hard	AVERAGE (ceval)
4000	28.00	29.76	29.12	32.58	27.27	29.55
8000	37.61	34.12	34.97	35.70	34.60	35.97
10000	33.85	38.71	33.28	35.45	29.45	35.00
12000	37.06	39.51	38.92	37.27	34.35	37.97
16000	37.55	38.02	42.22	37.10	34.29	38.53
20000	36.7	40.84	39.99	39.56	34.27	38.69
24000	38.46	41.01	42.78	38.94	35.81	39.97
27459	38.35	40.70	42.50	39.59	35.84	39.94

CMMLU

微调step	STEM	Humanities	Social Science	Other	China specific	Overall (cmmlu)
4000	25.65	26.68	27.40	27.84	28.02	26.91
8000	27.26	33.03	31.76	32.03	31.76	30.93
12000	28.67	35.04	33.29	35.15	34.59	32.87
16000	28.41	36.37	34.92	34.99	35.24	33.57
20000	29.49	36.91	34.79	35.61	36.16	34.04
24000	29.70	37.25	34.47	35.21	35.53	33.97
27459	29.76	37.15	34.15	35.45	35.27	33.91

实验4：COT回答方式

笔者在实验1的数据配置下（全量Infinity-Instruct数据+全量wanjuan exam），按照COT的方式微调wanjuan exam数据，在ceval上的准确率如下所示

	STEM	Social Science	Humanities	Other	Hard	AVERAGE (ceval)
2000ckpt	21.69	31.04	32.40	21.67	18.90	25.75
4000	25.59	28.63	29.80	26.29	24.99	27.21
8000	27.09	33.94	37.15	25.91	24.59	30.29
12000	25.93	30.40	34.59	29.03	26.96	29.28
18000	22.82	31.46	34.55	25.50	20.83	27.53
24000	25.54	34.64	36.17	25.80	23.07	29.59
30000	26.26	32.72	31.30	31.71	26.28	29.72
36000	24.71	33.53	28.58	27.60	22.32	27.84

效果甚至比非COT的方式训练出来的还要差一些，可能是小模型的推理能力还是不太行。我也尝试了让某家大厂的1.5B模型通过“先输出解释再输出答案”的方式回答，但是很多情况下模型还是会先输出答案再输出解释，或者干脆只输出答案。说明它们在微调时候数据格式应该也应该是先输出答案的，并没有使用COT的方式。

⑤ case展示

好的case

Q: 你的名字是什么

1 我的名字是SteelLLM，由SteelLLM团队开发，旨在为用户提供智能的回答和帮助。

Q: 写一个快速排序

1 快速排序是一种基于分治思想的排序算法，它通过选择一个基准元素，将数组分成两个子数组，然后

2 1. 选择基准元素：选择一个基准元素，例如数组的第一个元素。

3 2. 分割数组：将数组分成两个子数组，分别递归地对这两个子数组进行排序。

4 3. 递归排序：递归地对这两个子数组进行排序，直到它们的长度相等或它们的顺序不改变。

5 4. 重复步骤2和步骤3，直到整个数组有序。

6 下面是Python实现的快速排序算法：

7 ```python

8 def quick_sort(arr):

9 if len(arr) <= 1:

10 return arr

11 pivot = arr[len(arr) // 2]

12 left = [x for x in arr if x < pivot]

13 middle = [x for x in arr if x == pivot]

14 right = [x for x in arr if x > pivot]

15 ...

16 print(quick_sort(arr)) # [1, 2, 3, 4, 5, 6, 7, 8, 9]

17 ```

18 输出结果为: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Q:介绍一下北京

- 1 北京是中国的首都，位于华北平原上。它是一个历史悠久的城市，拥有丰富的文化遗产和历史遗迹。
- 2 北京市区是全国最大的城市之一，也是中国最著名的旅游景点之一。它是中华人民共和国的首都、直
- 3 在历史上，北京曾经被划为东交民巷地区，并成为外国使节居住的地方。现在，北京已经成为一个重
- 4 在北京，你可以参观许多博物馆、艺术馆和其他文化机构。此外，你还可以品尝到各种美食，如烤鸭
- 5 总之，北京是一座充满活力和魅力的城市，无论你是来观光还是休闲度假，这里都是你的不二选择。

Q: 请将以下内容翻译成中文: A new giant step toward making rockets completely reusable was made when the SpaceX Starship successfully landed at a designated position alongside a landing tower six minutes after launch on Sunday.

(翻译有点磕磕巴巴的感觉，大致意思是对的，Steel-LLM的英文数据只有20%有点少)

- 1 在星期天，SpaceX成功地降落在一个指定位置与发射塔六分钟后降落的6公里跑道上的一段距离中。

Q: 请将以下内容翻译成英文: 大模型是指具有大规模参数和复杂计算结构的机器学习模型。这些模型通常由深度神经网络构建而成，拥有数十亿甚至数千亿个参数。大模型的设计目的是为了

(大致意思是对的，但是有自由发挥的成分)

- 1 Q: 请将以下内容翻译成英文: 大模型是指具有大规模参数和复杂计算结构的机器学习模型。这些模型
- 2 (大致意思是对的，但是有自由发挥的成分)

Q:请以下面词语为主题写一首诗: 春天

- 1 春风吹拂万物生，
- 2 百花争艳竞相放。
- 3 绿草如茵，花香四溢，
- 4 鸟语虫鸣，乐享其中。

Q: 列出斐波那契数列的前6个数

- 1 1, 1, 2, 3, 5, 8

差的case

Q: 编写一个关于狗的笑话

(并不觉得好笑。。。不过我可能也写不出来关于狗的笑话)

- 1 为什么狗总是喜欢汪汪叫呢？因为它们觉得“汪”是他们的家！哈哈哈，怎么样，好笑吗？

Q: 计算一下 $123+286$ 的结果

(试了几个case，三位数加法的计算大部分情况计算结果会相差10以内)

- 1 $123 + 286 = 405$ 。

Q:小明有2个哥哥和6个姐姐，请问小明的爸爸有几个孩子

(少算了自己)

- 1 根据题目，我们可以列出一个简单的算式来表示这个问题：
- 2 ```
- 3 小明的弟弟数 + 小明的妹妹数 = 总共的孩子数
- 4 ```
- 5 其中，小明的弟弟数为 2，妹妹数为 6。将这两个数字相加即可得到总共有多少个孩子。
- 6 因此，答案是：小明的爸爸有8个孩子。

⑥ 小结及后续规划

Steel-LLM到目前为止应该基本算已经完工了，时间跨度比较大，后边应该还会写一篇汇总文章。对于正规LLM预训练项目，还有强化学习对齐的任务没做，这部分看以后还有没有时间和算力。之后打算做一些sft样本筛选的工作，看看模型还能不能再进步一些，也会以博客的形式更新，并将代码更新到github仓库，欢迎关注：
<https://github.com/zhanshijinwat/Steel-LLM>