计算语言学大作业

在本次大作业中，我们尝试训练一个简单的GPT模型，包括预训练、微调、推理、评测。两个同学一组，建议分工为一个同学负责预训练部分另一个同学负责微调部分，剩下部分两人协调，最后将两边的结果进行整合。

1. 实验环境搭建（0%）

请大家参考nanoGPT 的实现, 将实验跑通。

跑通标准: 1. 运行如下命令至训练5000步结束， Loss 在0.8左右

```
python train.py config/train_shakespeare_char.py
```

2. 运行如下命令能正常生成

```
python sample.py --out_dir=out-shakespeare-char
```

注：

1. mac本地也可以跑小一些的nanoGPT模型，请参照Readme中相关内容。

提交

一个训练最后10个step的截图以及生成的截图

2. 预训练 （35%）

使用 歌词语料 （密码 tsinghuacl2023）

2.1 字符级别模型预训练 (15%)

仿照nanoGPT里shakespeare_char的流程，使用不分词（即汉字字符）级别的模型完成GPT的训练

提交

1. 汇报词表大小、训练集token数。
2. 汇报Train Loss曲线以及Validation Loss曲线，汇报在测试集上的PPL
3. 找一些自己喜欢的歌，给出其中一些歌词，让模型进行续写，给出续写样例

2.2 分词后语言模型预训练（15%）

我们提供一个词表，由BPE模型分词得到，请使用它完成GPT的训练。

提交

1. 汇报词表大小、训练集token数。
2. 汇报Train Loss曲线以及Validation Loss曲线，汇报在测试集(test split)上的PPL
3. 使用2.1.3 中同样的歌词，让模型进行续写，给出续写样例
4. 和char相比，模型训练开始时的loss时高了还是低了？为什么？（提示：通过计算给出模型训练第一个step

loss的大概取值随词表大小的估计）

2.3 调参 (5%)

对上述模型选其一进行一定程度的调参，尝试提升性能（5%）

调参报告（简单明了即可）

3. 微调已经预训练过的模型（35%）

同样使用nanoGPT框架，加载已经预训练过的模型进行微调。

以下部分利用到开源GPT2系列模型：pretrained_model = "uer/gpt2-xlarge-chinese-cluecorpussmall" （**若实验资源显存不够，可以尝试小一些的模型uer/gpt2-large-chinese-cluecorpussmall · Hugging Face或 uer/gpt2-distil-chinese-cluecorpussmall · Hugging Face**）

请注意你也可以用**附加题**完成此题。

3.1 直接微调 (15%)

我们尝试对一个已经预训练好的语言模型**直接进行训练**

提示：需要对修改一些train.py 中相应内容进行更改的内容

提交

1. 汇报Train Loss曲线以及Validation Loss曲线，汇报在测试集（test split）上的PPL
2. 使用2.1.3 中同样的歌词，让模型进行续写，给出续写样例

3.2 词表扩展后微调（15%）

我们提供一个词表（词表由BPE算法给出，如何使用该词表可参见sentencepiece包），如何将这个词表嫁接在一个已经训练好的语言模型上呢?

尝试：

1. 将模型的embedding层换成和你训练出来的词表大小一样的一个embedding层上
2. 对于lm_head也需要一样的操作
3. 请思考如何初始化这个新的embedding层和lm_head? （提示：将新词表中的词用原来的tokenizer进行编码，用原编码方案对应的向量的平均（或是首个token的向量）来作为初始化）

注：欢迎尝试更多的方法

提交

1. 在歌词语料上进行微调训练，汇报Train Loss曲线，Validation Loss曲线， 汇报在测试集(test split)上的loss
2. 使用2.1.3 中同样的歌词，让模型进行续写，给出续写样例

3.3 比较 （5%）

比较3中得到的模型和2中得到的模型，进行一些分析

4. 推理（20%）

1. 调节生成参数 （10%）
   请参考huggingface上的生成参数，进行生成阶段调节
   每个同学需要对top_k这一生成参数进行调节，并改变nanogpt框架中的sample.py，实现top_p或typical_p采样算法，并调节对应生成参数。学号为单数的同学需实现top_p，双数同学实现typical_p。

参考范围：top_k-10-100, top_p 0.8-0.99, typical_p-0.1-0.9。

实现的过程可以参见huggingface的实现中的TopPLogitsWarper和TypicalLogitsWarper。

请用2.1.3 中同样的歌词，对于top_k和你实现的另一种方法让模型进行续写，并对这两个采样方法所续写的歌词人工进行高下判断并将续写结果和人工判断在报告中汇报（至少5对样例）。

2. 搭建简易网页端分享服务 （10%）
   使用Gradio搭建简易的网页端服务，分享你训出的模型。
   提供3~6张截图，来证明你的gradio和模型都是work的，同时也能展示模型的一些case！

5.API评测 （10%）

请使用大模型的API服务来对你的生成内容进行评测。如果你使用助教提供的API key, 请注意不要超过使用额度。

1. 研究绝对分数打分，以及pairwise比较打分，pairwise比较打分的时候请考虑位置偏差对于评分的影响，即在不考虑内容情况下，模型倾向于给第一个（或者第二个）答案 高分。考虑到这一偏差，pairwise比较打分时应交换位置重新打一遍。
2. 评测时，需要进行一定的prompt设计以获得较好的打分效果
3. 报告以上几种方式生成的歌词的得分
4. （可选）比较不同大模型作为评分器的优劣或探究不同评分维度对歌词评分效果的影响

   可供参考的打分维度：语句通顺性、上下文连贯性.....

附加题 （+10%）

用其他大模型来 **替代** 第3部分 (附加分10%, 即获得35%+10%)

1. 请使用2024年的**2个**开源的不同系列的大模型，包括但不限于Qwen，Deepseek，Llama，MiniCPM等系列的模型来进行完成第4部分的实验；
2. 并进行模型比较，推理和API评测部分

提交内容

1. 请提供一份报告，注意详略得当，完整清晰。报告的可读性将在正确性之外进一步影响你的得分。
2. 请提交所有源代码以便查验。

其他说明

如果有同学想单独提出其他和计算语言学相关的项目并完成，也可以单独联系助教，助教会一事一议进行审核，但要求必须是在这一个月内开始并完成（不能已经完成全部或部分，或者在交作业时仍然未完成），且不能和其他课程作业和实验室项目重复，如果查出自主题目违反这两个要求，大作业会被记0分。