

实验报告

刘雅迪
计 26

一、有监督微调数据的构造

微调数据集由一条条问答构成，但是每条问答的长度不一样，不利于后续的 batch 处理。所以我写了一个 pad_or_cut 函数，将每条被编码后的问答截断或补全成长度为 512 的数组，再将它们封装到一个二维的 np 数组中储存。

```
def pad_or_cut(value, target):  
    row = None  
    if len(value) <= target: row = np.pad(value, [(0, target-len(value))])  
    elif len(value) > target:  
        row = value[:target-1]  
        row = np.concatenate((row, np.array([enc.eot_token])))  
    return row  
block_size = 512  
train_token_pad = [pad_or_cut(arr, block_size) for arr in train_token]  
val_token_pad = [pad_or_cut(arr, block_size) for arr in val_token]
```

图 1

此外，为了解决模型推理中模型回答完问题不停止而是到最大长度（block_size）后截断的问题，除了像 FAQ 文档中描述的那样修改 sample.py，我在处理微调数据集时还在每条问答的后面加上了结束符 eot_token。

在获取 sft 批次和构造 loss mask 中，我仍然采用类似处理预训练数据集的方式，用 torch.stack 构造 x 和 y。而对于 loss mask 我则是遍历 x 第二维的每个编码数字，利用问题结尾的问号和回答结尾的结束符，找到答案起始的 index，将答案的 loss mask 设置为 1，这样计算 loss 时就只计算回答的部分。

```
enc = tiktoken.get_encoding("gpt2")  
begin_token = enc.encode_ordinary('?')  
end_token = 50256  
len1, len2 = x.size()  
for i in range(len1):  
    for j in range(len2):  
        b = -4  
        e = len2  
        if x[i][j:j+3] == begin_token: b = j  
        elif x[i][j] == end_token: e = j  
    loss_mask[i][b+4:e] = 1
```

图 2

二、模型的生成内容

1. 预训练模型的生成内容

我以 prompts=“杨过被他掌力震伤胸膛”为例，模型的输出如下：

杨过被他掌力震伤胸膛，倒转念：“你身上中了十成力，若不立时便知你是再好也没有了，你若能见过两个人，便救过我的手掌，也是不信的。”杨过只听他又道：“我倒想没有。”他说几句话，全身酸麻，只一阵寒气伤痛已流，霎时间笑声中，蓦地里又是一阵剧痛，当真是泪水滴而下。那知欧阳锋两人正在这样亲近，欧阳锋已全神贯注的在旁掌落风，走出三步，只见欧阳锋仍是纵前，当即收势迎面，凝神目眩，双手合十，说道：“这两个女孩儿是甚么人，你知道我不可知？”左手在他掌心中一个“闭穴”上一推，一个“拍女”字诀“

杨过被他掌力震伤胸膛，全身胀裂而死，猛地想起：“这小子和我比拚，你也有这痒子么？怎地现下这般本领？”杨过心想：“这个我怎可曾听你说过。”“腾”的一声，喝道：“这个我干么要几天啦？”杨过摇头道：“我自己姓杨。我师父不听你话，跟我说。”杨过心中奇怪：“我带来啦，怎地不让我走。”说道：“你跟我说起来，跟我说起来，他一起去。”杨过听她语气，脸上早将自己抽入怀中，又是满脸怀疑，说道：“前辈第一辈子联手，在下更加不喜欢你，你残忍了么？你若不是心情义、可不能自送了性命，我便是不如的。”杨过笑道

杨过被他掌力震伤胸膛，却用掌力推出，又如何碰到他手臂。他本来虽然不明白，但一时不知如何，自己被他打得粉碎，笑道：“我就算是不明白的啊，也就是啦。”杨过大喜，再叫他张口，说道：“小畜生，我假扮了。”他正要明白你几眼，只是见他拿着一个瓷瓶，自然而然的话虽然说到第三个字，却也未必放在心上，但人人身不由得又是怜惜，却也是脸上微微一沉，说道：“谁说你听我话？”杨过道：“谁说？”他话中出言提到“自然是”四字，但见他手指与此刻符在一旁，竟还扭住了他手臂，心中一凛，见他手一挥，有心要空打架，才

杨过被他掌力震伤胸膛，说道：“不打紧，别打了他。”杨过一怔，道：“这打狗棒法儿，你怎么又打我？”樊一翁笑道：“你这么狠辣，可别碰我，那可糟了。”杨过道：“好！你不认得完吗？”樊一翁一怔，道：“是我的，就算我跟你比武。”杨过道：“你教我三招还用完二招？”樊一翁双足一点，她自己告戒不通，因此自是没多辜负。杨过却不回答，笑道：“你到底是誰？”杨过伸手在他背上接了一下，打得他呼呼风响，樊一翁已抱拳打脱，樊一翁旁角动弹不得。两人这时心情已险越紧，急忙跃开。一灯又是奇怪，待他就在这暗

杨过被他掌力震伤胸膛，见他手掌心中握着一个绝冲滴的淡紫。郭芙瞧着嗓子正叫道：“妈，这小畜生，这般美了。”杨过喃喃的道：“你捉着我，这不是捉着。”郭芙一呆，接了自己手臂。杨过吃了一惊，那怪人骂道：“怎么？”怪人又是甚么，说道：“这贱婢，他怎么会捉到我手中？”郭芙叫道：“我不捉，捉干么？”杨过道：“你不肯跟你说，你不跟我说话。”郭芙“嗯”了一声。杨过叫道：“过儿，你又没好菜迷？”郭芙接过了断脂，喝道：“这贱婢拍不起啦！”杨过道：“别扭我！”轻轻拍他膀。郭芙格格一

杨过被他掌力震伤胸膛，重重又受伤，只是不动了。小龙女又是一惊，这“掌法”实在是太过强，只得问下去。只见他身軀高瘦，腰间甩起一柄长剑，然后退了七八步，二十余柄长剑齐到。小龙女双剑反弹，横剑递出，往杨过胸口抓去。杨过见他双目已被他掌力挣扎，转身便逃。小龙女横剑相交，横扫，抛开剑边门。杨过剑锋长剑，当的一声，长剑向杨过小腹上抓去。这剑锋甚是浅力，杨过长剑在剑上给他接了回去。小龙女道：“你身上的剑抵挡，内心使的是甚么用？”杨过知道再说这剑法不是法王掌法的掌招，却也甚为根柔，但对方手指使的是长剑法

杨过被他掌力震伤胸膛，他出招误伤她们性命，但他只怕就不能再再救，于是一见之下，左手作势凌厉，一拳打出，一拳齐出。这一拳击出是她的手臂，变幻玉女素手法，实是罕见，相互变幻，更是吃惊，叫道：“姑姑，我伤势怎样？”杨过道：“我怎地没伤你？”小龙女道：“谁伤你？”杨过道：“我怎么会对你说？”小龙女道：“我怎么会这路掌法？”杨过道：“我瞧这路掌法使将出来，可是他这倒也自己宣了，不肯用甚么手法引。”小龙女道：“你为甚么不肯接你？他不知道啊。”杨过道：“我早年轻，倒是动弹不得。”小龙女摇头道：“小龙

杨过被他掌力震伤胸膛，伤心中悲愤，右手被他掌力一推，突然右手一拉，那使棋盘打成两截。慈恩的棋盘比变化精妙，已将这么久不同归于尽，虽然仍然不免受伤，却也不阻挡。两人一日之中，只见杨过所练的功夫诀窍不通精细，但觉他本来不是杨过最精进妙诀，二人仍是一向玉女心经的死符正的授授的杨过，只得道：“那么过儿是了，他们姑姑已不到了局上。”小龙女道：“我——我——”杨过道：“过儿，你又没甚么了？”小龙女道：“是了。”小龙女道：“嗯，想不到你打，也要问我姑姑本事，就算出来，你也未必心思，那么我也没有假扮厚为的。”杨过

杨过被他掌力震伤胸膛，不由得大惊，心想他走了招，我可能追赶，他从未见过多少时候，明天武功极是了得，反而然的想起自己神剑埋伏在这路剑上。他身边不便，对方若非如此。两人同到成吉思汗血宝剑。一到剑湖宫削，不论如何向西行。杨过一灯大师所救自己，不愿跟她进入黑龙潭中，烛光下见到成吉思汗盘膝坐在地下，越走越近，不由得越是热闹，他心中对一灯大师的安危，虽然出破绽，却也均各有一股杨过的惊呼，倒也甚是僵硬，有心要他突然无动静，眼见其中终于武功已不及二人联手，当下暗暗叫苦，一齐跃起身来，想起只

杨过被他掌力震伤胸膛，弹出铁剑击向旁方的铁剑，朗声说道：“得罪，你上来罢！”这七人同时接着杨过来抵挡，只得惊觉，跃起身来，到后来已在后殿去了。杨过叫道：“喂，你快抢罢！”这时此时他极大变，当即飞身来，横剑直刺。金轮法王武功虽在他掌下，但双剑合十，却别停留，接连下山去，第三次便知察步的方位相交，心想：“好，这一退了！”原来杨过眼见金轮法王向杨过第二次挥动，“马钰”叫人师父被他缩中，一定猛地翻身跃起，当下只见他两手指尖在自己背心欲滑，铁剑抢前，立即跃起，喝道

图3

可以看出预训练模型能够对我的输入进行补全，并且输出格式较好。

2. 微调模型的生成内容

我以 prompt=“介绍一下《天龙八部》的主角？”为例，模型的输出如下：

介绍一下《天龙八部》的主角？《天龙八部》中有许多著名的武功绝学，如乔峰的降龙十八掌、虚竹的达摩剑法、慕容复的九阳真经等，这些武功以其独特的招式和威力成为小说中的亮点之一。

介绍一下《天龙八部》的主角？《天龙八部》中有许多爱情的人物命运，比如阿朱是段正淳和阮星竹的女儿，她从小被教育成一个善良、聪明的姑娘。然而，在和萧峰的恋情中，她被迫背负了太多苦难。她的生父段正淳在知道了她的存在后却想要杀了她以报复康敏的背叛。后来，她被乔峰误伤并误与她被乔峰打他的父亲次。为发现自己的女儿托付给乔峰。虽然郭襄阳城中杨过是在襄阳城与金轮法王战斗时中原武林中的后代，但在成长过程中都展现出了他的英勇、机智和善良的威名，最终成为了中原大地上最受尊敬的英雄之一。

介绍一下《天龙八部》的主角？《天龙八部》中有许多著名的武功绝学，如乔峰的降龙十八掌、虚竹的达摩剑法、慕容复的九阳真经等，这些武功以其独特的招式和威力成为小说中的亮点之一。

介绍一下《天龙八部》的主角？《天龙八部》中有许多主要角色，其中包括段誉、虚竹、乔峰、王语嫣、阿朱、阿紫等。每个角色都有独特的个性和故事线。

介绍一下《天龙八部》的主角？《天龙八部》中有许多爱情线索。其中包括乔峰和阿朱、段誉和王语嫣、虚竹和西夏公主、虚竹和西夏公主和西夏公主的爱情纠葛以及其他角色之间的爱恨情仇。爱情线索为小说增添了浪漫和感人的色彩。

介绍一下《天龙八部》的主角？《天龙八部》中有许多著名的武功绝学，如乔峰的降龙十八掌、虚竹的达摩剑法、慕容复的九阳真经等，这些武功以其独特的招式和威力成为小说中的亮点之一。

介绍一下《天龙八部》的主角？《天龙八部》中有许多主要角色，其中包括段誉、虚竹、乔峰、王语嫣、阿朱、阿紫等。每个角色都有独特的个性和故事线。

介绍一下《天龙八部》的主角？《天龙八部》中有许多主要角色，其中包括段誉、虚竹、乔峰、王语嫣、阿朱、阿紫等。每个角色都有独特的个性和故事线。

介绍一下《天龙八部》的主角？《天龙八部》中有许多著名的武功绝学，如乔峰的降龙十八掌、虚竹的达摩剑法、慕容复的九阳真经等，这些武功以其独特的招式和威力成为小说中的亮点之一。

介绍一下《天龙八部》的主角？《天龙八部》中有许多著名的武功绝学，如乔峰的降龙十八掌、虚竹的达摩剑法、慕容复的九阳真经等，这些武功以其独特的招式和威力成为小说中的亮点之一。

图 4

可以看出微调模型能够回答我输入的问题，虽然可能回答的正确性有待保证，但基本能正常完整地输出。

三、数据集的收集方法

预训练的数据集我使用了 shediao、shendiao、tianlong 三本书中的数据，将其添加到一个数组中，并且去除了数据中的换行符、全角空格等特殊符号，这样可以避免输出中出现非文字标点的特殊字符。

微调的数据集我主要通过文心一言生成，部分数据则是在网上找的相关的问题和回答，一共 119 条。刚开始我以为数据集越多越好，后来发现并非如此。所以在后续的微调过程中我将部分数据精简了一点，并且删除了不太恰当的问答，这样模型输出的回答会较为准确。

四、模型效果分析

1. loss 分析

预训练过程中，模型的 train loss 和 val loss 均下降直至收敛，用 matplotlib 绘制如下：

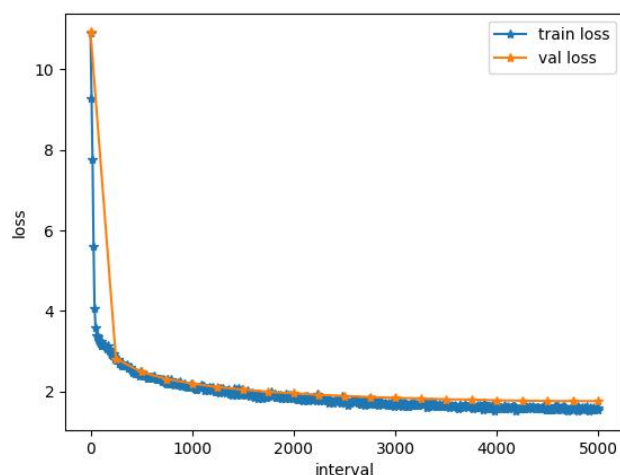


图 5

微调过程中，模型的 **train loss** 下降直至收敛，而 **val loss** 先下降再上升，用 `matlibplot` 绘制如下图。

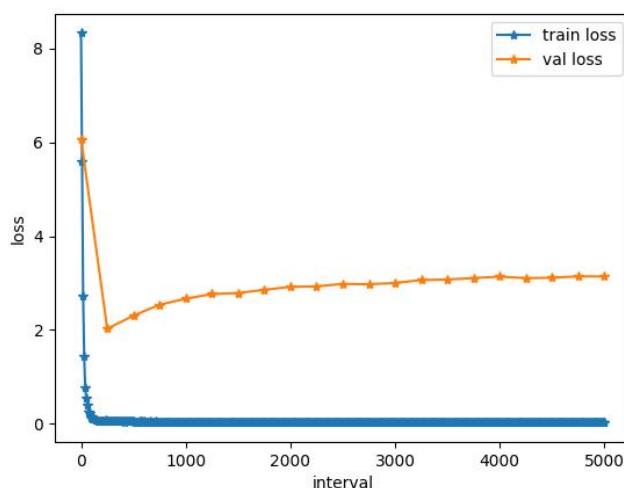


图 6

推测原因可能是数据集比较少，训练轮次较多，所以最后模型过拟合了。但是保存的是 **val loss** 最小的那个模型。此外，我尝试减小了模型微调过程中的学习率，发现在模型过拟合前训练的轮次更多，且 **val loss** 更低。

但是在实际的模型输出中发现微调过程学习率较低的模型输出效果并不好，基本上输出混杂了小说文章内容和问答内容。

2. ppl 分析

我计算了几个微调模型的 **ppl** 值，基本上不超过 8，最好的是 3，如下表：

微调模型	ppl 值	lr
out-1694747755	6.7558544204798245	1e-3
out-1694754761	7.575824473155883	1e-3（数据集略有更新）
out-1694758307	3.0003523436422723	1e-5
out-1694762393	3.0585097565927213	3e-5

表 1

ppl 值均比较小，可以看出模型的效果还是不错的。不过在观察模型实际的输出中发现 **lr** 较低的模型输出效果并不好，可见低 **ppl** 值并不能完全评估模型的效果好坏。

3. Rouge-L 分析

通过动态规划的方法得到了两个序列的最长公共子序列，从而计算 Rouge-L 值。

还是以上述 prompt= “介绍一下《天龙八部》的主角？”为例，参考答案为“《天龙八部》中有许多主要角色，其中包括段誉、虚竹、乔峰、王语嫣、阿朱、阿紫等。每个角色都有独特的个性和故事线。”，计算了一下每个生成答案的 Rouge-L 值，结果如下：

输出	Rouge-L 值
《天龙八部》中有许多著名的武功绝学，如乔峰的降龙十八掌、虚竹的达摩剑法、慕容复的九阳真经等，这些武功以其独特的招式和威力成为小说中的亮点之一。	0.298531968876132
《天龙八部》中有许多爱情的人物命运，比如阿朱是段正淳和阮星竹的女儿，她从小被教育成一个善良、聪明的姑娘。然而，在和萧峰的恋情中，她被迫背负了太多苦难。她的生父段正淳在知道了她的存在后却想要杀了她以报复康敏的背叛。后来，她被乔峰误伤并误与她被乔峰打他的父亲次。为发现自己的女儿托付给乔峰。虽然郭襄阳城中杨过是在襄阳城与金轮法王战斗中中原武林中的后代，但在成长过程中都展现出了他的英勇、机智和善良的威名，最终成为了中原大地上最受尊敬的英雄之一。	0.14573668134591086
《天龙八部》中有许多著名的武功绝学，如乔峰的降龙十八掌、虚竹的达摩剑法、慕容复的九阳真经等，这些武功以其独特的招式和威力成为小说中的亮点之一。	0.298531968876132
《天龙八部》中有许多主要角色，其中包括段誉、虚竹、乔峰、王语嫣、阿朱、阿紫等。每个角色都有独特的个性和故事线。	0.8810625611818101
《天龙八部》中有许多爱情线索。其中包括乔峰和阿朱、段誉和王语嫣、虚竹和西夏公主、虚竹和西夏公主和西夏公主和西夏公主的爱情纠葛以及其他角色之间的爱恨情仇。爱情线索为小说增添了浪漫和感人的色彩。	0.32837277579524726
《天龙八部》中有许多著名的武功绝学，如乔峰的降龙十八掌、虚竹的达摩剑法、慕容复的九阳真经等，这些武功以其独特的招式和威力成为小说中的亮点之一。	0.298531968876132
《天龙八部》中有许多主要角色，其中包括段誉、虚竹、乔峰、王语嫣、阿朱、阿紫等。每个角色都有独特的个性和故事线。	0.8810625611818101
《天龙八部》中有许多主要角色，其中包括段誉、虚竹、乔峰、王语嫣、阿朱、阿紫等。每个角色都有独特的个性和故事线。	0.8810625611818101
《天龙八部》中有许多著名的武功绝学，如乔峰的降龙十八掌、虚竹的达摩剑法、慕容复的九阳真经等，这些武功以其独特的招式和威力成为小说中的亮点之一。	0.298531968876132
《天龙八部》中有许多著名的武功绝学，如乔峰的降龙十八掌、虚竹的达摩剑法、慕容复的九阳真经等，这些武功以其独特的招式和威力成为小说中的亮点之一。	0.298531968876132

表 2

可以看出部分生成答案的 rouge-l 相当高，说明模型的效果还不错。

4. 模型可能的最大生成长度

以 prompt=“介绍一下《射雕英雄传》的主角？”为例，模型的输入输出如下：

question

介绍一下《射雕英雄传》的主角？

ClearSubmit

output

郭靖：郭啸天与李萍之子，生于蒙古，后南下中原并于张家口邂逅黄蓉与其成为情侣，一起闯荡江湖。除了郭靖和黄蓉，还有许多重要的角色在《射雕英雄传》中扮演着重要的角色。其中包括：杨康：杨铁心与包惜弱之子，从小在金国赵王府长大，为人狡诈卑鄙，在赵王府时认金国六王爷赵王完颜洪烈作父。穆念慈：杨铁心的养女，因比武招亲与其养父之亲生儿子杨康邂逅并堕入爱河。

Flag

可以看出模型的最大输出比较长而且输出比较流畅与完整，没有截断。

五、探索性质的尝试

1. 对长文本或长问答的处理

一开始模型的输出总会在最后一句话还没有说完时就自动截断，于是我把储存的的 `train.bin` 和 `val.bin` 解码后输出，发现如果每条问答过长的话在数据储存时就会自动截断（参考之前写的 `pad_or_cut` 函数），这样会造成信息的损失。

我 `google` 了一下怎样处理这个问题，找到了一个使用 `transformers` 模型的例子，它把超长的输入切片为多个较短的输入，每个输入都要满足模型最大长度输入要求，并且允许相邻切片之间有交集。不过在尝试了一番后还是放弃了像它这样处理，一方面是 `gpt2` 只有一个特殊符 `eot_token`，而 `transformers` 有 `[CLS]` 和 `[SEP]` 两个特殊符号，如果想要使用 `eot_token` 作为最后的截断的话这里就不能用其作为答案的开始。

最后使用了一个相对暴力的方式，将 `block_size` 由原来的 256 改成了 512，这样微调模型输出基本不会出现截断了，但是预训练模型由于数据集过长，仍然会在最后一句话出现截断，我还没有想到好的解决方法。

2. 提高模型输出正确性的尝试

一开始模型输出的效果不好，哪怕我的 `prompt` 用的是微调数据集中的原题，模型也很容易回答另一个问题的答案。我以为是微调的数据比较少，所以尝试添加一些数据再来微调模型，但是这样反而连原来能回答上的问题都回答不上了，所以我又回过头去精简数据集，去除了某些可能会影响模型输出的“不太好”的数据。精简数据集后的效果也比较一般，但是比之前好一点点。

后来我尝试改变一下微调的学习率，由于微调的数据集较小，所以我减少了原有的学习率，尝试了两个，`1e-5` 和 `3e-5`，发现虽然 `val loss` 更小了，但是模型输出效果变差了，所以还是使用的原有的学习率来微调模型。

展示一下改变学习率后微调过程中的 loss 变化：

① $lr=1e-3$

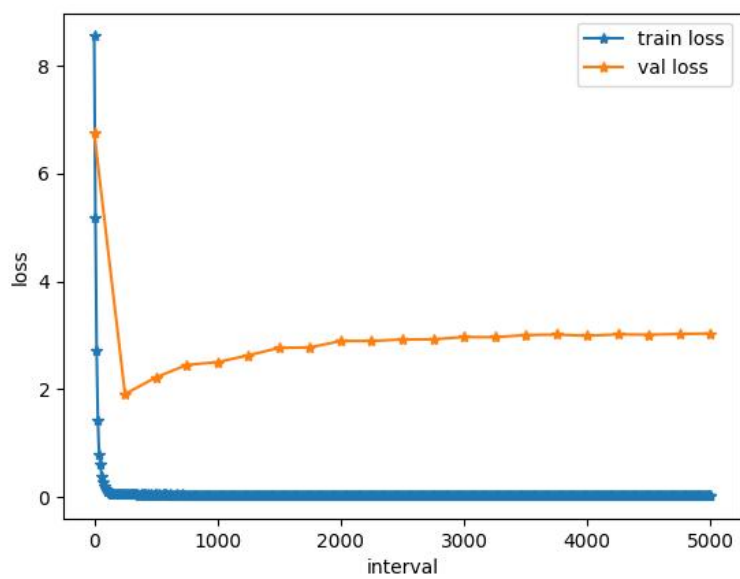


图 7

② $lr=1e-5$

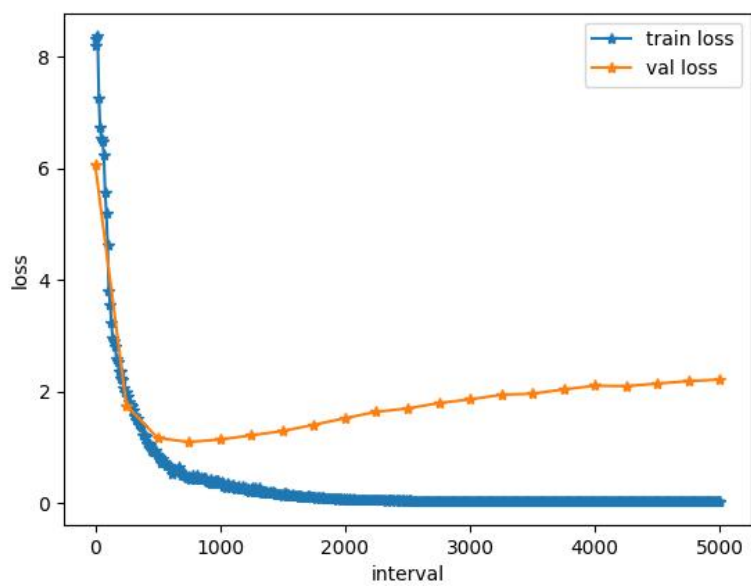


图 8

③ $lr=3e-5$

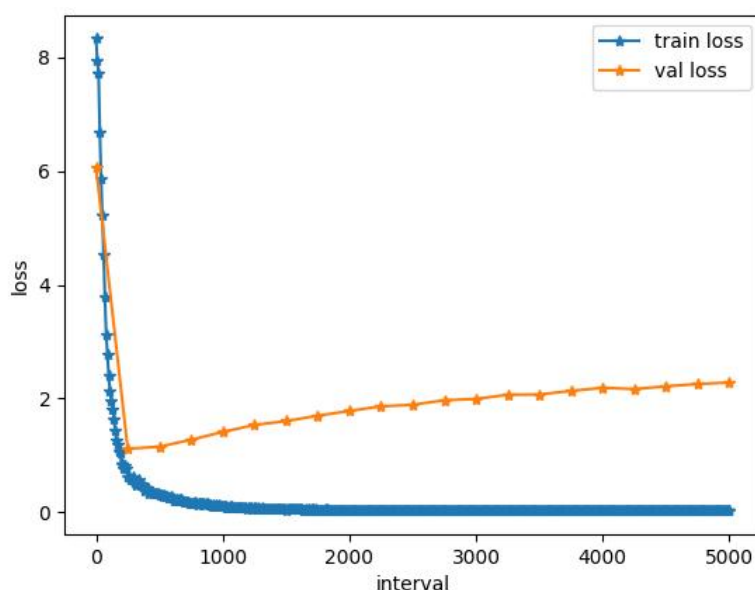


图 9

六、花费时间及感想

感觉这个大作业花的时间一点也不比上一个大作业花的时间少。在写之前我以为我已经大致了解了它的流程，但是写的过程中还是遇到了很多困难，遇到了很多我不理解的地方，比如在读助教提供的代码时我就花了很多时间来理解。此外感觉我对于 `numpy` 和 `torch` 的使用还不太熟练，所以在写代码时也花了不少时间查阅相关资料。

可能整个过程中花费时间最长的是数据预处理过程，在怎样处理微调数据集和获得 `sft` 的 `batch` 上我想了很久，也尝试了很多不同的方法。而遇到在微调后得到的模型输出效果不好，或者输出有乱码，或者输出会有截断等等这些问题时，我总会去调整一下我的数据集，或者调整一下我的数据处理方法。这就导致我整个大作业大部分时间都花费在了数据的处理上面，主要是 `prepare_sft.py` 和 `get_batch_sft` 上。

此外我还花费了很多时间在提高模型输出正确性以及模型的评估上面。提高模型输出正确性的过程很折磨，所以后来我几乎放弃了，转而去优化输出的格式，比如去除截断和一些特殊符号等等。之前我尝试了各种类似 `chatgpt` 的 `chatbot`，经常会被它们的奇怪回复无语到，但是真的自己去处理这个问题的时候我才发现这不是一件简单的事，至少凭借现在我的能力还无法解决模型输出“乱说”的问题。

最开始处理微调数据的时候我焦虑过，后来提高模型输出正确性我也很焦虑，总觉得自己应该把这个做出来，担心自己会不会因为这个方面做的不好而被扣分。但是后来我想到了之前东老师在课上说的：只要你努力过了，你会得到一个令你满意的分数。所以后来我觉得只要尽力就好，哪怕这个我做不出来，只要我能达到我自己的最高标准就很不错了。

写这个大作业的过程中查阅了很多资料，了解到了很多知识，也保存了很多可能以后会用到的内容。自己亲自训练一个小的模型和看别人写的论文/代码的感受是完全不一样的，尽管这个过程很困难，但结果我还算比较满意。