22

Transformer实验过程

数据集:WMT14 英德数据集

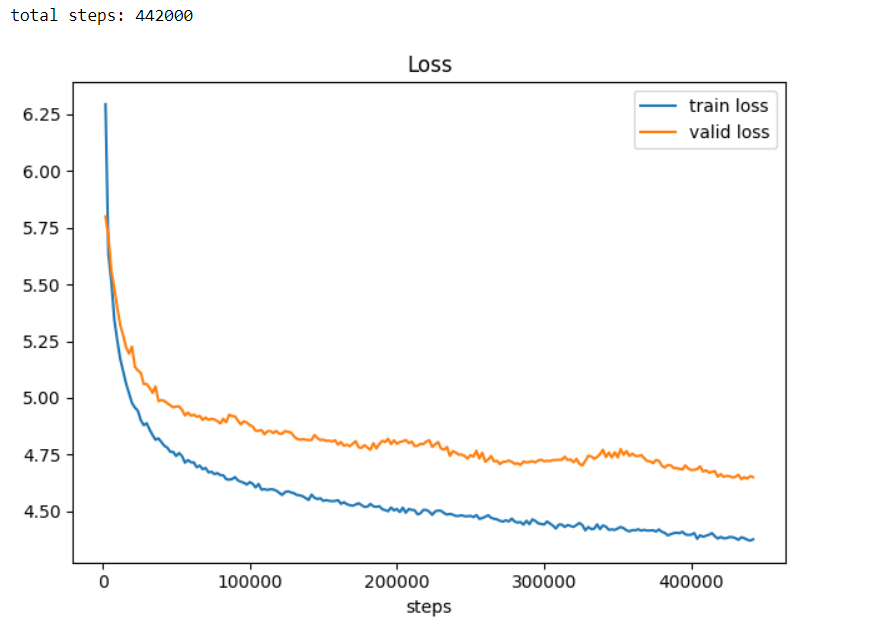
Transformer为Attention is all you need.中的base

RTX2080单卡

(batch\_size=8)

10 hours 15steps

30 hours 44.2steps loss 4.38 ppl 79.55



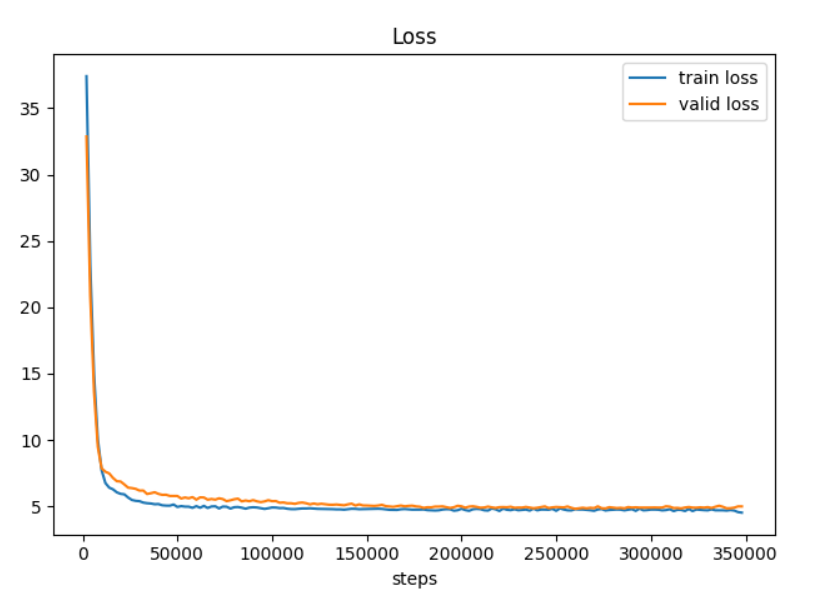
共44.2万步,最后一部分给丢了

两天时间下降到4~5之间,还有下降趋势。

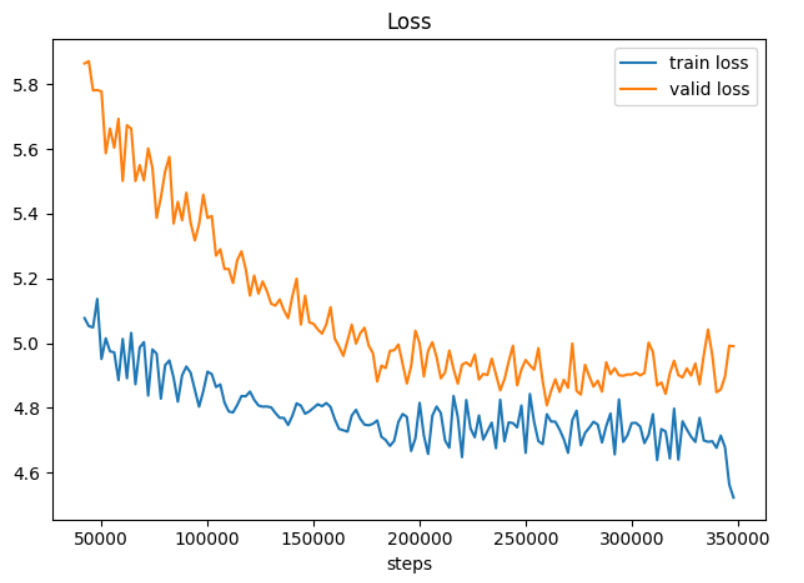
问题是为什么在第一个epoch中会出现过拟合的现象，而且valid是验证模式,train是训练模式。好怪。

第二版Transformer

改了很多:换了Tokenizer(之前Sentencepiece的写法有问题,只用2048个句子训练Tokenizer),重写了数据生成(按Token数量生成一批数据,截断了太长的数据),权重绑定,用TransformerLayer重写模型,加了LabelSmooth ,激活函数换成gelu,



只看40000步的loss



正式训练的记录:

前几天没记,因为log会记,现在记是为了看loss是否真的下降了。根据知乎的一篇文章要训练23轮之后才会拟合好，现在只一次训练了10轮。大概一天一轮左右吧。

每天晚上看一次，不要太频繁了，不然没有信心了。

第三天晚上：

Valid\_loss = 5.12346 train\_ppl = 260.88

第四天晚上18:10:

Valid\_loss = 5.06045 train\_ppl = 253.34

第五天晚上 21:37:

Valid\_loss = 4.95481 train\_ppl = 237.43

第六天晚上 20:08:

Valid\_loss = 4.88754 train\_loss=5.42 train\_ppl = 226.73

3月7日

第七天晚上 21:36:

Valid\_loss = 4.78126 train\_loss=5.36 train\_ppl = 212.25

3月8日

第八天晚上 22:11:

Valid\_loss = 4.72332 train\_loss=5.35 train\_ppl = 210.97

学习率不对劲。lr已经是1e-5了

应该到25000个Token后才加一步才对，和梯度累积一样。

3月10日

第十天早上 9:36:

Valid\_loss = 4.64546 train\_loss=5.26 train\_ppl = 192.73

第十天晚上 20:35:

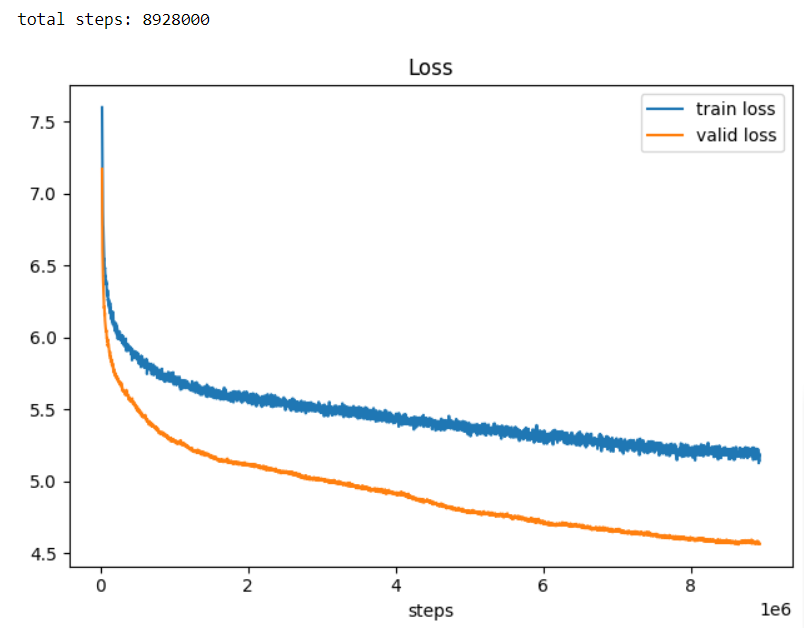
Valid\_loss = 4.62459 train\_loss=5.21 train\_ppl = 182.37

3月13日 21:04:

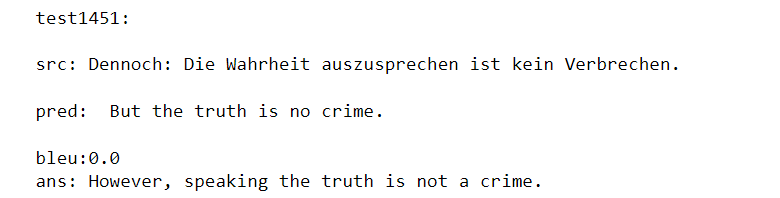
第十三天晚上

Log.txt显示上次更新是 a day ago,所以训练十个epoch花了12天

Valid\_loss = 4.56293 train\_loss=5.14 train\_ppl = 171.30



Train loss更高应该是因为dropout



BLEU4可能并非完美,这，我不想吐槽。“the truth is”刚好只占3个Token。

平均BLEU为3.53

比第8个epoch低了(epoch8为5)

会不会过拟合了，比较valid\_loss和valid\_BLEU不一样。

真不好判断是否过拟合了，毕竟批大小、学习率和激活函数都不同。

这BLEU分数也太低了，valid的BLEU有20多，valid和test之间的gap也太大了。

不对，valid的BLEU不是自回归，艹，根本不能判断。重练吧。

3月14日:

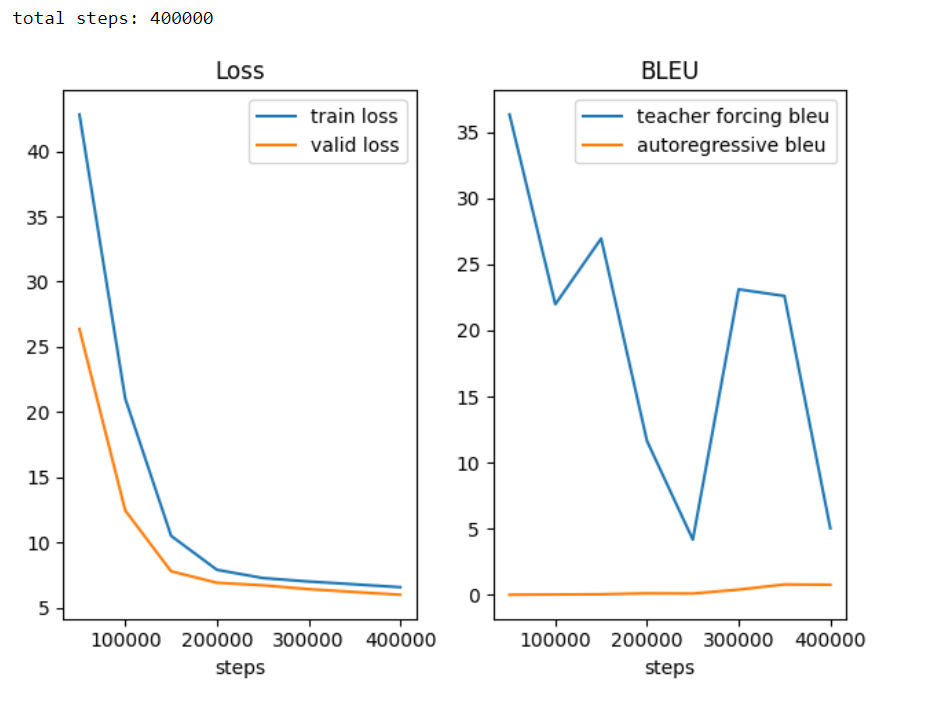
提前backward确实不占用那么多内存了,可以考虑增大一个batch里的tokens数和序列长度。

3月15日:

昨天运行了大概十几个小时之后超内存了。今天减少batch\_token到1400,截断长度为650。

进程号为17139

昨天的数据：



可见teacher forcing bleu 和 autoregressive有很大的gap。

autoregressive采取和预测时完全一致的设置，即自回归预测。

Emmm，忘了把checkpoint给删了，所以现在继承了昨天的训练，不过问题不大，与一次性训练的区别只有：

1. epoch1有些数据被多训练了一次（随着epoch的增多，这个影响会越来越小）
2. Batch\_token变小了,截断长度也变小了。（这个也没有影响）

3/15

第一晚：

| steps 500000|lr 0.00015073317636017444 | s/step 0.10 | loss 6.23 | ppl 510.21

==============================

A strategy of strategy to be- Obama-inction of Obama............ Obama Obama Obama Obama Obama Obama Obama Obama Obama Obama

A Republican strategy to counter the re-election of Obama

valid\_loss:5.62950

teacher forcing bleu:10.502298921346664

==============================

autoregressive bleu: 1.3754440773657577

best autoregressive bleu score: 1.3754440773657577

3/16

第二晚：

| steps 1350000|lr 0.00020937112928702688 | s/step 0.10 | loss 4.73 | ppl 113.67

==============================

A strategy of strategy to make Obama Obama-exction of Obama

A Republican strategy to counter the re-election of Obama

valid\_loss:4.21882

teacher forcing bleu:18.535153567790985

==============================

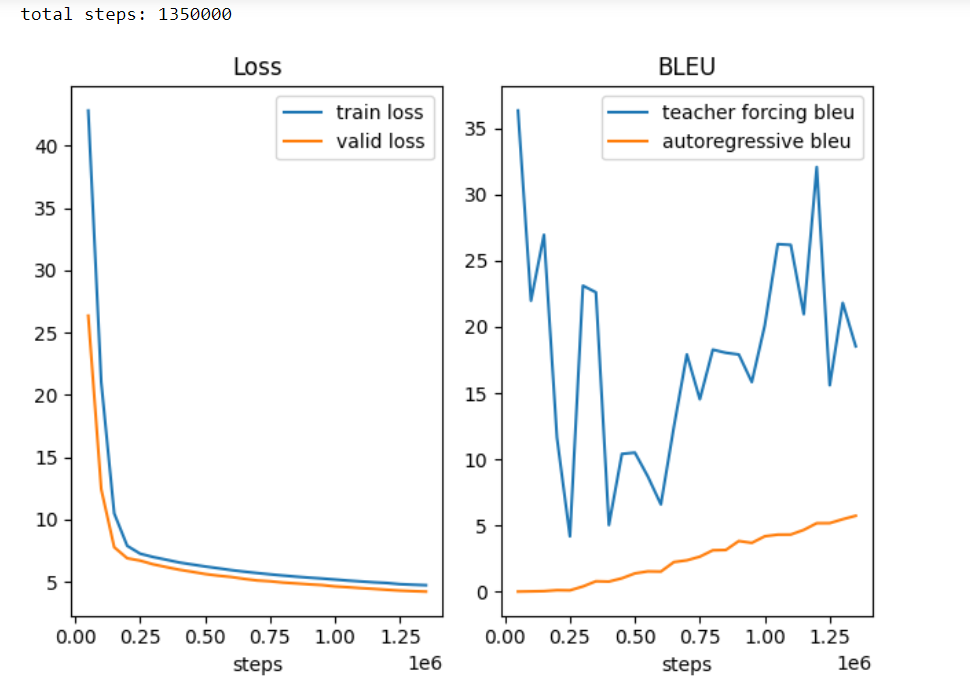
autoregressive bleu: 5.72969986582261

best autoregressive bleu score: 5.72969986582261

save to best\_bleu.pt

save to checkpoint.tar

Valid Loss下降得多快，一天下降1.4,之前下降只有0.1



3月18日18:13

| steps 3000000|lr 0.00014477214996178834 | s/step 0.09 | loss 4.12 | ppl 61.27

==============================

A repan strategy to counter Obama rep-election of Obama

A Republican strategy to counter the re-election of Obama

valid\_loss:3.54206

teacher forcing bleu:22.425904870033264

==============================

autoregressive bleu: 9.51360531230148

save to checkpoint.tar

3月19日 23:02

第五晚:

| steps 4100000|lr 0.00012481989002445393 | s/step 0.09 | loss 3.97 | ppl 52.74

==============================

A repan strategy to counter the Obama-election of Obama

A Republican strategy to counter the re-election of Obama

valid\_loss:3.36792

teacher forcing bleu:21.5727299451828

==============================

autoregressive bleu: 11.239790395989823

3月20日 21:49:

第六晚:

| steps 4950000|lr 0.000113960055500575 | s/step 0.09 | loss 3.88 | ppl 48.49

==============================

A Republican strategy to counter the Obama-election of Obama

A Republican strategy to counter the re-election of Obama

valid\_loss:3.26378

teacher forcing bleu:22.65542894601822

==============================

autoregressive bleu: 12.222670949811581

3月21日 21:19:

第七晚:

| steps 5800000|lr 0.0001055168341783624 | s/step 0.10 | loss 3.82 | ppl 45.66

==============================

A Republican strategy to counter Obama Obama-election of Obama

A Republican strategy to counter the re-election of Obama

valid\_loss:3.20428

teacher forcing bleu:32.75443613529205

==============================

autoregressive bleu: 13.021440273561453

Loss下降开始慢起来了.

3月22日 17:43

第八晚

| steps 6550000|lr 9.942050050042727e-05 | s/step 0.11 | loss 3.77 | ppl 43.44

==============================

A Republican strategy to counter the Obama-election of Obama

A Republican strategy to counter the re-election of Obama

valid\_loss:3.14243

teacher forcing bleu:28.230401873588562

==============================

autoregressive bleu: 13.69631813240183

3月23日19:56

第九晚

| steps 7450000|lr 9.336556513793458e-05 | s/step 0.11 | loss 3.71 | ppl 40.97

==============================

A Republican Strategy to counter Obama Obama-election of Obama

A Republican strategy to counter the re-election of Obama

valid\_loss:3.08544

teacher forcing bleu:32.31809735298157

==============================

autoregressive bleu: 14.59849152838721

3月24日 18:29

稍微计算一下需要多少时间

2080每秒1e13FLOPs = 一天8.64e17

因为我GPU利用率只有40%~50%,取45%

一天3.8e17

原文要3.3e18的算力

所以一共需要8.68天。Emmm，不对啊，已经有10天了。

因为我是互译的，增加了难度？

还是德译英比英译德简单？有这种可能，所以要把英译德的验证方法给提上日程了。

3月25日 12:09

第十一中午,其实昨天就完成训练了,不过没去看。

| steps 7600000|lr 9.248537677638516e-05 | s/step 0.11 | loss 3.71 | ppl 40.70

==============================

A Republican Strategy to counteract Obama-election of Obama

A Republican strategy to counter the re-election of Obama

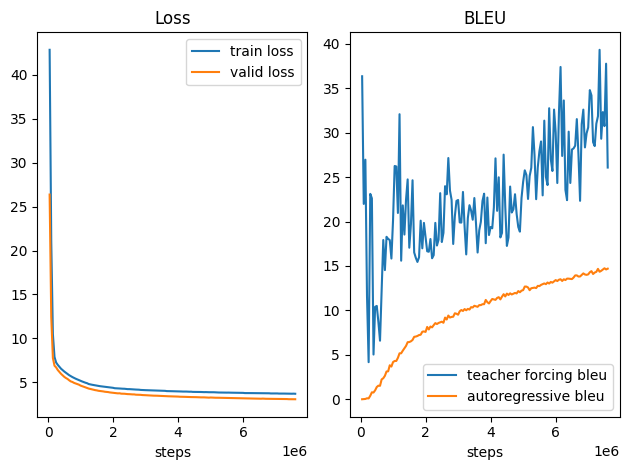
valid\_loss:3.08177

teacher forcing bleu:26.077941060066223

==============================

autoregressive bleu: 14.69709742257541

10epoch图：



感觉这个趋势上不了28啊

测试集de->en的BLEU4分数为:8.71

测试集en->de的BLEU4分数为:6.52

怎么跟验证集差这么多?

Emmm，先梳理一下我的transformer和原文的transformer有什么不同

1. 激活函数我用了gelu,原文用relu
2. 我是英德互译,而原文是英译德（我怀疑这个影响比较大，因为这相当于transformer学了两倍的内容，而transformer的参数没有提升。）
3. 我每一批的Token数较少,但是用梯度累积解决。

好像发现了eval\_loss低的另一个原因:eval\_loss的计算有误,pad没有贡献loss,但贡献了次数。

Train\_loss也有一点问题,计算的loss=np.mean(batch\_mean\_loss)每个批次的Token数不一样，所有权重是不同的。

这里问题最大在于两个计算loss的方式不一致。

进程6549

这次从上次的10epoch重新开始训练，但不是双向的，而是单向翻译en->de

3月26日:

新发现:

Warmup\_steps也是有问题的

Warmup\_steps应该为4000\*acumulation\_steps

17:34:

| steps 12500|lr 0.00018000244228970558 | s/step 0.05 | loss 3.63 | ppl 37.70

==============================

eval\_pred: Eine Gegenublikanische Strategie zur um der Wiederwahl Obama Obama entgegenzuwirken

eval\_ans: Eine republikanische Strategie, um der Wiederwahl von Obama entgegenzutreten

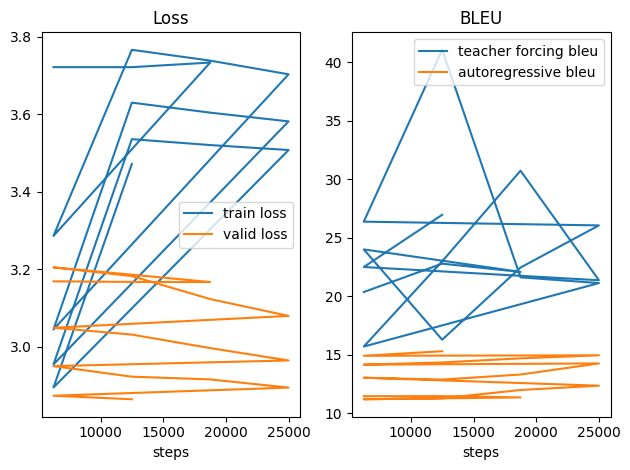
valid\_loss:3.03132

teacher forcing bleu:41.081082820892334

==============================

autoregressive bleu: 12.897594576112002

Steps有点怪,每个epoch重新从0开始计算了。



Step出问题了,还是先把这个给改好先吧

找到原因了,存的是step//accumulation\_steps，加载step时没乘回accumulate

一些未来的建议:

先设立一个小训练集用于debug

修改超参数:

梯度累积:1024步（比原论文更多的tokens,约1Mtokens）

学习率:1e-3

Scheduler:余弦退火热重启优化器，T0:1024(即初始1024步一个周期)，T\_mult：2，eta\_min=1e-4

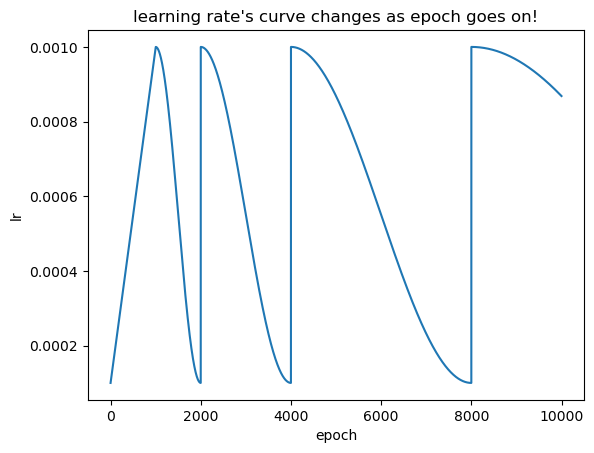
进程12747

3月27日晚上开始训练

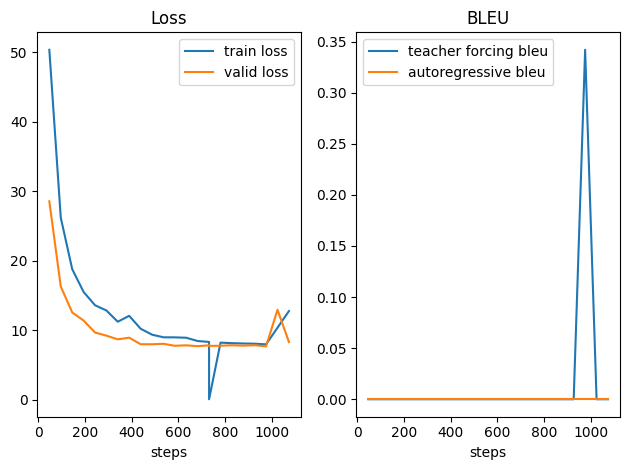
3月28日

一晚上才loss下降到8，学习率太大了吗？

之后加个预热吧:



这是两个epoch的情况:



设置为：

梯度累积:1024步（比原论文更多的tokens,约1Mtokens）

学习率:1e-3

Scheduler:余弦退火热重启优化器，T0:1024(即初始1024步一个周期)，T\_mult：2，eta\_min=1e-4

明显不对劲，停止训练。

新设置：

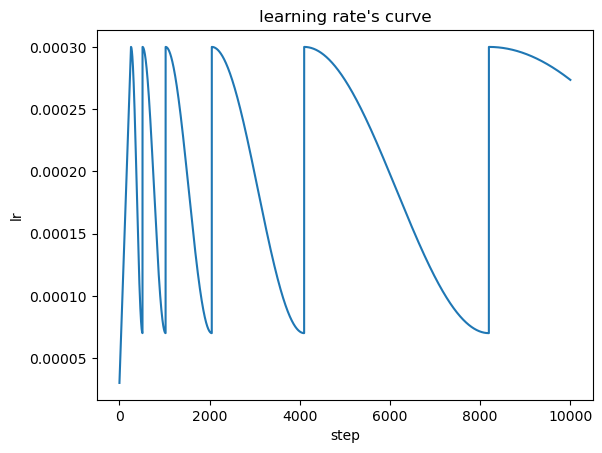
梯度累积:256步（约25万tokens）

学习率:3e-4

Scheduler1:线性优化器:start\_factor=0.1,end\_factor=1,total\_iters=256

Scheduler2:余弦退火热重启优化器，T0:256，T\_mult：2，eta\_min=7e-5

即:



进程39449

3/28 22:40开始训练

3月29日

20个小时下降到6感觉应该延长预热时间会好很多

新设置:预热步数增大到8192

进程33736

学习率bug找到了,预热的scheduler要最后定义,因为定义的时候会改optimizer的学习率（之前余弦scheduler把预热scheduler的学习率改掉了.导致学习率从0.0003开始升）

Kill掉33736

重新训练

进程:33634 时间:23:03

3月30日 早上9点35

新设置并不好,

一晚上才loss到9

啧，这学习率真难调

感觉有可能是累积步数影响较大，目前来看。有可能小样本更容易学习，一开始就用大样本可能太难了，可以随epoch慢慢增大累积步数

先别急，先按原论文整个基准学习率和批Token数。

原论文设置：

一批50k tokens

预热步数:4000

Scheduler:



每个batch的tokens数<=batch\_token

这里假设取一小批的token数=batch\_token

则

累积步数=5e4//batch\_token约等于50

问题应该出在梯度累积的步数上。

推测：当batch\_size大到一定程度时,平均梯度的变化不大,

即mean\_loss(batch\_size=阈值)≈mean\_loss(batch\_size=远大于阈值)

所以当训练步数固定的情况下，累积步数越大，更新次数越少，导致loss下降的次数也越少，所以loss下降慢。

猜测的结论：累积步数太大会导致无效计算，使训练变慢。

新设置:

为了提高gpu利用率,把减少了max\_len=256,提高了batch\_Tokens=2048

为了设置尽量和原论文相同,accumulation\_steps = 50000/batch\_Tokens约等于25

除了batch\_size（由accumulation\_steps和batch\_tokens决定）不同外,其他与原论文一致。

进程7250：3月31日10:13

关于accumulate\_steps和学习率：

一般是维持accumulate\_steps不变,学习率缓慢下降。

但accumulate\_steps应该逐步增大比较好，前期的时候即使batch和整个dataset有较大偏离，更新的方向大致是对的，这时应该追求更多的更新步；到训练后期，为了近似计算整个dataset的loss，应该增大batch\_size。

这种设置下，应保持学习率不变或者用余弦scheduler。

22:37

| steps 32000|lr 0.0002397506110809723 | s/step 0.04 | loss 5.18 | ppl 177.66

==============================

eval\_pred:    solcheublikanische Politik die die Demokratievereinigung zu Präsident zuzuwirken.

eval\_ans: Eine republikanische Strategie, um der Wiederwahl von Obama entgegenzutreten

valid\_loss:5.49267

teacher forcing bleu:42.50396781219706

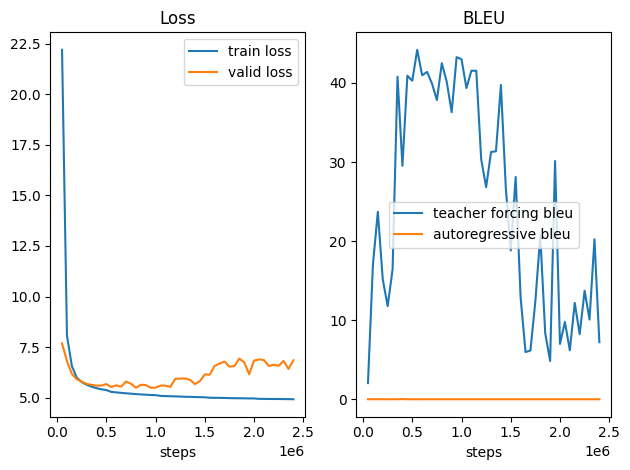
==============================

autoregressive bleu: 0.0

出现过拟合了，截断长度变短大概少了三倍的数据。

目前来看超参几乎与原论文一致了，唯一不能一样的是截断长度。

截断长度目前来看影响还是很大的。



该设置下过拟合了。

Kill掉了

2023/4/1 19:20

Truncate\_len=768 batch\_tokens=1536

Accumulation\_steps=50000//batch\_tokens

从头训练

进程37433