ANN实验报告3

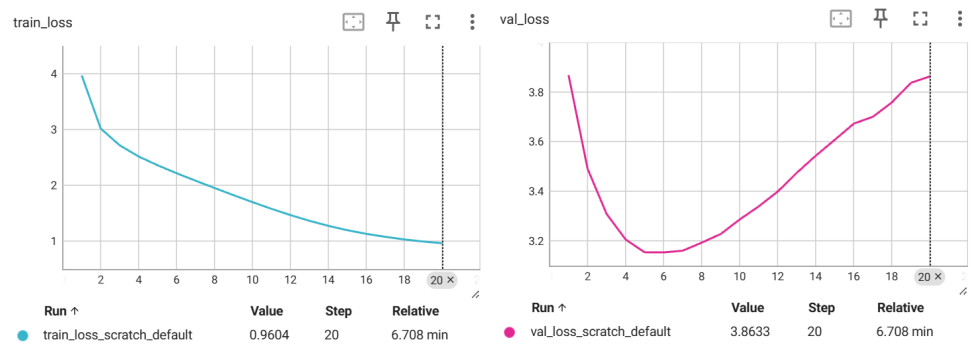
张天祺 2021010719 计12

两个模型的基础实验

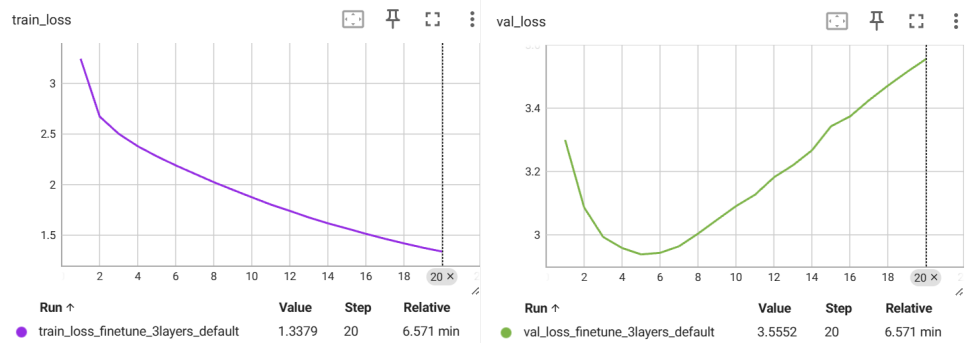
在原实验框架中，模型在验证集上的ppl一旦高于之前最低的ppl时，模型就会停止训练。为了获得可能更好的checkpoint以及更好了解模型的训练数据，我将这里的代码进行了修改，让模型在上述情况下可以继续训练下去。

在收集数据时，我使用`tensorboard`来帮助完成数据绘图的处理。

`Tfmr-scratch`



`Tfmr-finetune(3layers)`



两个模型的测试结果

选择两个模型训练时validation perplexity最小的checkpoint，模型测试生成的参数设置均为默认，解码策略为random

	Perplexity	Forward BLEU	Backward BLEU	Harmonic BLEU
`Tfmr-scratch`	18.65	0.584	0.427	0.493
`Tfmr-finetune`	15.36	0.572	0.433	0.493

比较与分析

通过观察上面的实验数据，可以发现：

- finetune模型的training loss在开始时显著低于scratch模型，但是随着训练量的增加scratch模型的training loss更低。但是通过validation loss可以看出，在scratch模型的training loss低于finetune模型时，两个模型都已经出现过拟合的情况，也就是validation loss不降反升。
- 两个模型对于训练集的拟合效果都很好，在训练同样多数据后scratch模型对于训练集的拟合效果甚至高于finetune模型，但是在验证集上finetune模型的效果要明显好于scratch模型，这说明预训练模型的泛化能力更强。
- 从测试集的perplexity上可以看出，finetune模型的困惑度更低，因此对于这个任务来说，pretrain+finetune的训练效果要好于scratch模型。
- 从生成句子的评价指标**BLEU**分数来看，**Forward BLEU**分数衡量句子的流畅度，**Backward BLEU**衡量句子的多样性，**Harmonic BLEU**评价总体的指标。从这几种分数来看，scratch模型的流畅度更好，finetune模型的多样性更高，两者的总体生成效果基本一致。通过肉眼观察可以看出，生成的句子质量普遍一般，可能是解码策略导致的。

不同解码策略的结果

按照实验要求进行实验，得到的结果如下表：

`Tfmr-scratch`	**Forward BLEU**	**Backward BLEU**	**Harmonic BLEU**
random, $\tau = 1$	0.584	**0.427**	0.493
random, $\tau = 0.7$	0.817	0.382	0.521
$top\_p = 0.9, \tau = 1$	0.705	0.415	**0.522**
$top\_p = 0.9, \tau = 0.7$	**0.884**	0.302	0.450

`Tfmr-finetune`	**Forward BLEU**	**Backward BLEU**	**Harmonic BLEU**
random, $\tau = 1$	0.572	**0.433**	0.493
random, $\tau = 0.7$	0.809	0.388	**0.524**
$top\_p = 0.9, \tau = 1$	0.691	0.414	0.518
$top\_p = 0.9, \tau = 0.7$	**0.882**	0.318	0.467

可以看出，两种模型的对比并不明显，在同样的生成参数下，两种模型生成的句子的三种分数基本相当。

对于random和top_p两种解码策略，可以很明显看出当top_p的值降低，会让模型生成的句子的Forward BLEU值上升而Backward BLEU值下降，也就意味着模型生成的句子的流畅度上升，也就是单个句子生成的效果变好而生成句子的多样性下降。对于总体指标而言，top_p=0.9生成句子的整体分数要好于random策略。

对于生成的temperature参数，可以看到，temperature参数的下降让句子的Forward BLEU分数上升而Backward BLEU分数下降，也是意味着单个句子生成的效果变好而整体生成的句子多样性下降。

对上述结果进行分析，在整体上看，降低top_p和降低temperature在对于生成句子效果的控制表现相似。降低top_p使得预测生成下一个token的分布更趋向于高概率的token，因为低概率的token已经被mask掉了。降低temperature同理，模型在预测下一个token时会让所有的logits处以temperature参数，所以降低temperature参数会让模型更倾向于下一次预测原本概率更高的token，因此两种控制操作的表现存在一定的相似之处。

case study

对于以上实验的两个模型的四种解码策略，共八种生成数据，我们对于每一种都随机抽取十个句子来进行分析，随机抽取句子的代码如下：

```
import glob
import random

files = []
for file in glob.glob("codes/output*"):
    files.append(file)

for file in files:
    sentenses = []
    with open(file, "r") as f:
        for line in f.readlines():
            sentenses.append(line)

    sample = random.sample(sentenses, 10)
    print(file)
    for sentense in sample:
        print(sentense[:-1])
    print('-------------------------')
```

生成结果的命名规律为：
`output_{args.test}_{args.decode_strategy}_{args.temperature}.txt`

得到的结果如下：

```
codes/output_scratch_default_random_1.0.txt
A big cream big building is leaking with a red light in a city .
Woman jumping umbrella the traffic signal at a bus stop .
A double decker bus riding down a street corner .
A man and a woman her baby feeding a giraffe walking a few people .
A woman sitting next to a giraffe on a dirt field .
A sleek bus travelling down a city street in a city .
A white and blue bus driving down a city street .
A couple of giraffe standing next to a stone path .
A white commuter bus that has been being towed simulate in a city .
A white plane sits on top of a field .
-------------------------
codes/output_scratch_default_random_0.7.txt
A bus parked on the side of the street with people seated .
A red bus driving down a street with people standing .
A giraffe and a zebra in the dirt next to trees .
The giraffe is eating leaves off from the tree branch .
```

```
A car turns an intersection in the middle of a city .
A couple of women walking across a street while a bus is parked .
A man and a woman sitting on a bench with a dog walking in the back .
A couple of giraffes stand in a field of grass .
A bench on the bank of a lake in a park .
A man is sitting on a bench in the background .
-------------------------
codes/output_scratch_default_top-p_1.0.txt
A street corner with a woman on a bench , near it and a bus behind them .
A blue fire hydrant that has two streetsticks and a metal wall .
A large crowd of people on the road at a traffic light .
A park bench sitting in a park next to trees .
A bright red bus is driving down a street surrounded by trees .
The tourist buses are parked beside the street of the building .
A bus that has a stop sign on the side .
A black and yellow bus is sitting in the street .
A dog lays in front of a wooden bench on a lush green hill .
Three sheep stand at the end of a metal where have a fence .
-------------------------
codes/output_scratch_default_top-p_0.7.txt
A man is standing on a field next to two giraffe .
A traffic light hanging over a street with a large building in the background .
A bus driving down the street with cars in the background .
A couple of giraffe standing next to each other .
A man sits on a park bench with two stroller .
A man is sitting on a bench in the park .
A large passenger jet sitting on top of an airport tarmac .
A red fire hydrant sitting in the grass near a tree .
A cat laying on a bench in the middle of a park .
A giraffe is standing in a field of grass .
-------------------------
codes/output_finetune_3layers_default_random_1.0.txt
A woman is taking a picture of herself on a bench in the shade .
A person who is wearing a colorful hat on a bench near a water fountain .
An elderly man in black shirt and jeans sits on a bench .
A man is sitting on a wooden bench in the desert .
A little boy rests in a park beside a tree .
A plane is on a runway that has landed some sort .
A multi - colored wooden bench with a clock face in the distance .
Two beds , one writing on the benches and words next to a bench .
Men sit on bench holding umbrellas while a child sleeps on a bench with smaller
dog walk nearby .
The white cow is standing for the fence to pick something .
-------------------------
codes/output_finetune_3layers_default_random_0.7.txt
A couple of giraffes stand in a dirt field .
A fire hydrant is sitting in the grass beside a fence .
A yellow fire hydrant sitting next to a green pole .
Small red and white plane stands on a grassy hill .
A street scene with several cars and cars on the asphalt .
A young girl sitting on the shore of a park bench .
A couple of cats sitting on a bench on a wooden bench .
A red fire hydrant sitting on the side of a road .
A couple of giraffes standing next to each other .
A giraffe is standing in a field with chickens .
```

```
-------------------------
codes/output_finetune_3layers_default_top-p_1.0.txt
The electronic traffic signal is clearly visible from the city skyline .
A large green truck driving down a field next to a crosswalk .
A small group of giraffes that are standing around eating from a tree .
A yellow bus is parked on a lot in a building .
A large giraffe standing next to a baby giraffe .
A person stands by a fire hydrant in front of a building .
A streetlight with two stop lights on it , one and the side of the building .
A red and yellow double decker bus traveling down a busy street .
Two giraffes are standing around in the savannah .
Two people on a city street wearing orange vest walks in the rain .
-------------------------
codes/output_finetune_3layers_default_top-p_0.7.txt
A wooden bench sitting on top of a lush green field .
A man is sitting on a bench and reads a book on a bench .
A man sitting on a bench and looking at the ocean .
A giraffe standing next to a tree in a zoo .
A large green bus is parked on the side of the street .
A man and a woman sitting on a wooden bench .
A giraffe standing in a field in a grassy area .
A large white airplane flies over a field of water .
A man sitting on a bench in a park with his foot in the grass .
A couple of giraffes walking across a dirt road .
-------------------------
```

- 首先观察生成句子的语法表达，可以看出有很多句子都存在着语法错误和逻辑不通等问题。大多数句子出现的问题是缺少谓语动词或是be动词。
- 综合而言，`output_finetune_3layers_default_random_0.7.txt`也就是finetune模型使用random策略，temperature为0.7时生成的句子较好，虽然也有错误，但是也有一些语法正确且符合逻辑的句子。
- 这个最好的模型的衡量生成句子的综合质量的Harmonic BLEU得分也是最高，这个指标与我的判断一致。

最终的提交结果

经过一系列的测试后，我最终选择的提交结果是finetune预训练模型的前三层，生成的策略是random，temperature=0.8，相关指标如下：

	Forward BLEU	**Backward BLEU**	**Harmonic BLEU**
final results	0.742	0.412	0.530

相关文本在`output.txt`中

简答题

Transformer和RNN的对比

- 从时间复杂度上，transformer大致的时间复杂度为$O(n^2d + nd^2)$，RNN大致的复杂度为$O(nd^2)$。但是transformer由于其结构的设置可以并行计算，而RNN由于存在循环结构，每一步的计算都依赖于上一步计算的隐藏状态，因此需要占用大量的计算资源而言时间成本较高。
- 从模型表现上看，RNN可以通过循环层提取具有时序特征的表示，例如序列里的依赖以及上下文关系，而transformer可以通过多头注意力机制提取出具有上下文关联性的特征表示，例如文本中的关键词和语义信息。由于自注意力机制，transformer可以比RNN更好地处理长序列的依赖关系，因此在本次文本生成的任务中会表现更好。

在inference阶段设置`use_cache`为`True`的作用

`use_cache`的作用是为了避免重复计算。在inference阶段，模型会逐token进行自回归解码，当前时间步长的输出作为下一个时间步长的额外输入。使用`use_cache`后，之前计算的结果都会保存在cache中以供后面新增的token计算使用，如此可以避免每个token都进行复杂的attention计算。

时间复杂度计算

在decode l_t 时，每个block的attention计算Q、K、V等需要 $O(d^2)$，多头注意力需要花费时间 $O(dt)$，FFN的时间复杂度为 $O(d^2)$，因此可以计算出通过所有block的时间复杂度为 $O(B(td + d^2))$。除此之外，LM head计算的时间复杂度为 $O(dV)$。因此，经过如上分析可知，decode l_t的总体时间复杂度为 $O(d(Bt + Bd + V))$，decode整句的时间复杂度为$O(Bd^2T + BdT^2 + dVT)$。

时间复杂度的决定性因素

如上面推导，当T较大时，单层self-attention花费时间相对较多，当d较大时，FFN则花费更多时间。

预训练的影响

- 从数据指标层面看，预训练模型测试时的perplexity较低。
- 从训练过程层面看，预训练模型收敛的是速度更快。
- 从模型的生成结果看，预训练的影响并没有很大，Forward BLEU分数会降低一点，而Backward BLEU分数会上升，总体的Harmonic BLEU分数略微上升一点。
- 预训练可以使得模型具有一定的语言能力，在下游任务中通过finetune获得更好的效果是意料之中的事情。而经过训练的模型泛化能力更强，所以会在衡量文本多样性的Backward BLEU分数更高。但是可能是因为本次任务的数据集并不复杂，预训练的效果相较于从头训练而言并没有显著的优势。

Bonus

使用BPE分词并阐述好处

尝试使用BPE分词：

```
from tokenizers import ByteLevelBPETokenizer


encoder_path = "codes/tokenizer/encoder.json"
vocab_path = "codes/tokenizer/vocab.bpe"

tokenizer = ByteLevelBPETokenizer(encoder_path, vocab_path)

text = "To save your time, the default hyper-parameters should provide a
reasonable result. However, you can still tune them if you want to do more
explorations."

tokens = tokenizer.encode(text).tokens

print("Tokens:", tokens)
```

对于这一句话：

> To save your time, the default hyper-parameters should provide a reasonable result. However, you can still tune them if you want to do more explorations.

空格分词就是按照空格将句子打成词级别的单位，这样需要给词表中的每一个词都分配一个标识符，还需要对所有不在词表中的词添加标识符`[UNK]`。而且，这样的分词方式无法让模型在开始时区分例如名词单复数、动词与对应be动词等的关系，只是将他们认作时完全不同的词汇。空格分词的最大优点是简单，但是很难处理稍微复杂的语境。例如上面句子中的`hyper-parameters`这种带连字符的词就会使空格分词需要更大的词表来包容。

使用bpe分词得到结果如下：

```
Tokens: ['To', 'Ġsave', 'Ġyour', 'Ġtime', ',', 'Ġthe', 'Ġdefault', 'Ġhyper', '-
', 'param', 'eters', 'Ġshould', 'Ġprovide', 'Ġa', 'Ġreasonable', 'Ġresult', '.',
'ĠHowever', ',', 'Ġyou', 'Ġcan', 'Ġstill', 'Ġtune', 'Ġthem', 'Ġif', 'Ġyou',
'Ġwant', 'Ġto', 'Ġdo', 'Ġmore', 'Ġexplor', 'ations', '.']
```

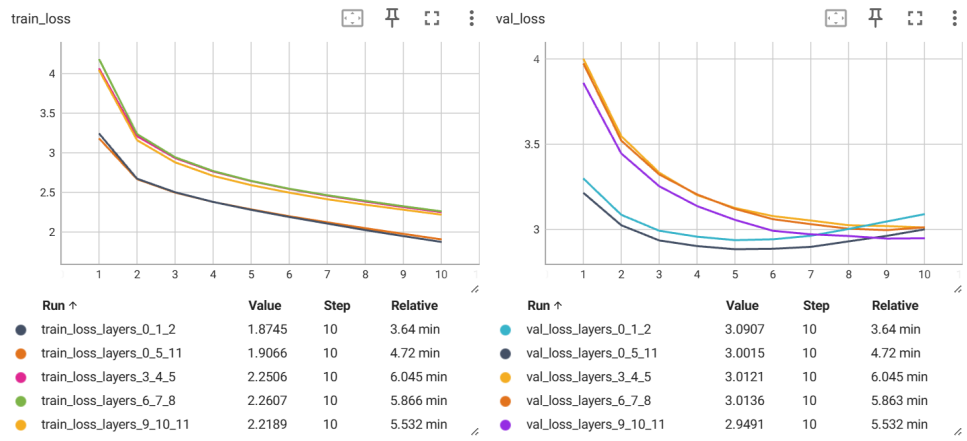
对于上文的bpe分词，将`hyper-parameter`分成了`hyper`，`-`，`param`，`eters`四个词，将`explorations`名词分成`explor`，`ations`动词根和后缀。

bpe分词的颗粒度是sub-word级别的，可以将常用的单词在一定程度上保留，不常用的单词分解成有意义的常用词段。它的优势在于可以捕捉文本中的复杂模式，避免了大量的未知标记，降低了词表的复杂度，也可以更好捕获上下文信息，提高模型的性能。

从GPT2-base中取三层分析结果

在这节实验中，我分别选取了GPT-base中的如下层来训练，`layers_{a}_{b}_{c}`表示我选择了第a,b,c三层。

实验结果如下：



在训练层面可以看出，从训练的层面，当选择前三层或者选第一层，中间一层，最后一层这样的跳跃的选择方式时，模型在训练集上的loss更低，且在验证集上更快拟合，也更快出现过拟合现象。而对于其他几种选择，模型在训练中的表现较为相似。

在模型的生成层面，固定生成策略为random, temperature=0.8，将上面训练好的模型分别进行inference然后计算相关指标，得到如下结果：

	Forward BLEU	Backward BLEU	Harmonic BLEU
layers_0_1_2	0.562	0.442	0.495
layers_3_4_5	0.583	0.447	0.501
layers_6_7_8	0.602	0.449	0.509
layers_9_10_11	0.617	0.452	0.522
layers_0_5_11	0.587	0.442	0.504

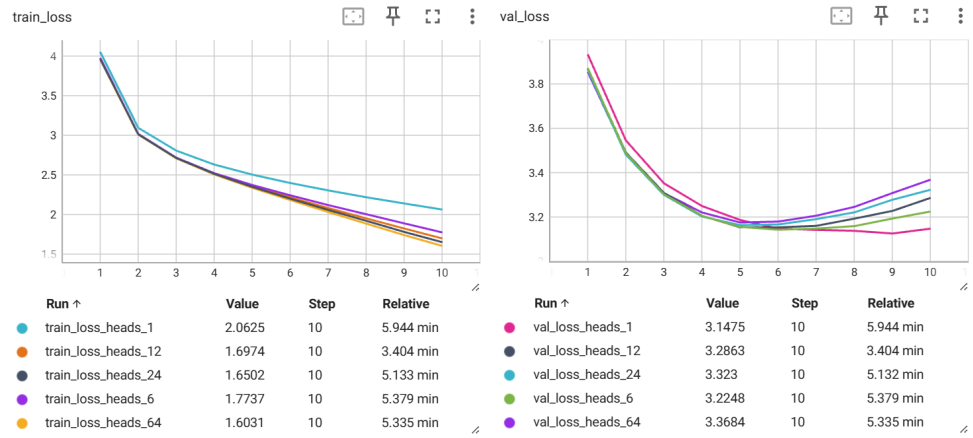
观察上面的结果可以看出，使用最后三层的生成结果最好，使用第0、5、11层的测试集loss最低。

整体而言，越高层的layers会学到越高层的语义信息，越底层的layers会学到越底层的语义信息。而跳跃选择第0、5、11三层理论上就可以学到高中低三层的语义信息，因此这个模型在验证集上的loss表现就最好。

在multi-head attention中`num_heads`的作用

实验的默认设置的`n_heads`为12，我尝试了几种可行的`n_heads`大小：1、6、12、24、64

训练方面：



生成方面：

	perplexity	Forward BLEU	Backward BLEU	Harmonic BLEU
heads_1	18.31	0.739	0.416	0.532
heads_6	18.37	0.766	0.406	0.531
heads_12	18.65	0.758	0.404	0.527
heads_24	18.80	0.752	0.406	0.528
heads_64	18.93	0.759	0.405	0.528

综合上面的实验可以看出，无论是模型训练时的拟合速度还是生成的测评指标，`num_heads`参数的影响都没有很大，可以体现出transformer结构对于这一参数并不敏感。也有可能是因为本次实验的网络结构和数据集都较为简单，不同的模型学习能力的差别并不明显，没有体现出multi-headers的能力。

相对而言，multi-headers加速了网络的训练速度，每个header学习到了不同的数据特征，模型的收敛速度会快一些，整体表现会好一点。但是如果multi-header没有配合更高的hidden_dim，可能会因为不同的header学习到了相同的特征而降低了模型的表现力。