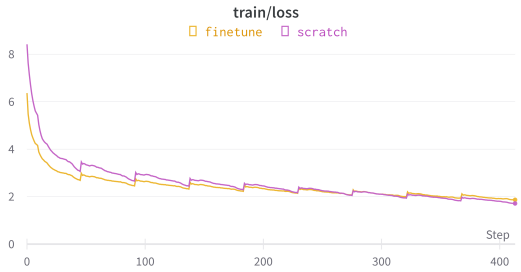
Text Generation with the Transformer Decoder

沙之洲 2020012408

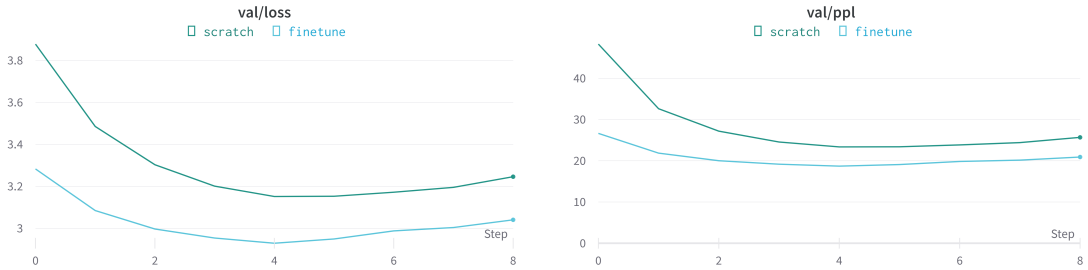
1 Tfmr-scratch and Tfmr-finetune

在这里，我们更改了模型收敛的条件，将其改为连续三次 validation 的 ppl 不降低的时候停止训练，但是我们的模型仍然保存的是 ppl 值最低的模型。

training loss 如下 (这里横坐标是 / 10 batch)



validation loss 和 ppl



Tfmr-scratch 和 Tfmr-finetune 的最好表现

	Tfmr-scratch	Tfmr-finetune
train loss	**1.712**	1.867
val loss	3.151	**2.929**
val ppl	23.37	**18.705**

将上述两个训练出来的最好的模型以 random decoding 策略和 temperature = 1 在 test 集上进行测试，得到如下结果

	Tfmr-scratch	Tfmr-finetune
ppl	18.66265638	**15.28958485**
forward BLEU-4	0.567354306	0.565157618
backward BLEU-4	0.430012492	0.435985905
harmonic BLEU-4	0.489227111	0.492238625

结果分析：

由于我们这里将模型收敛条件改为 连续三个周期 validation 的 ppl 不降低才停止训练。因此从 training loss 上可以看出，在最后的几个 epoch 中，scratch 的 loss 已经低于 finetune 的 loss，这说明 **scratch** 相比于 finetune **更容易过拟合训练数据集。**这种过拟合的现象也能在 validation 上被体现出来，虽然我们这里的 validation loss 和 ppl 都是取的模型表现最好的时候，但是能明显看到，finetune 在 validation上的表现都明显高于 scratch。

对于 scratch 比 finetune 更容易过拟合训练数据集，我认为原因在于 pretrain 参数能给 finetune 带来更大程度上的**泛化性能。**而由于 scratch 没有任何先验知识，所以会更容易陷入训练集的局部最优中，更容易过拟合。

同时，从 train loss 中我们还可以看出，finetune 比 scratch 有更快的收敛速度，我认为这也是 pretrain 的泛化性能带给 finetune 的优势。具体来说，finetune 通过 pretrain 带来的"知识"，能够更快地从训练集中提取到所需要的特征进行学习，因此收敛更快。

在评判生成结果的 4 metrics 上我们可以看出，finetune **在 ppl 上**对 scratch **有较高的超越。**但是 finetune 在 BLEU 上和 scratch 并没有明显的差异。根据 ppl 和 BLEU 的计算公式，不难看出 **ppl** 更多关注于对于目标序列预测的**概率是否正确**，而 BLEU 只关注于**预测的结果是否正确。**所以这里 test 集上的 ppl 能够很好反映出 finetune 比 scratch 有更好的泛化性能，但是 BLEU 并没有体现出来。这可能是因为 finetune 和 scratch 的差距还不能很明显的在生成结果上体现出来，这实际上也反应了 BLEU 相较于 ppl，**对于模型能力的刻画略有不足。**

2 Temperature and Decoding Strategy Finetune

下文中 name 以 `<decode strategy>_<temperature>` 表示

在 scratch 模型上：

name	ppl	forward BLEU-4	backward BLEU-4	harmonic BLEU-4
random_1	18.66265638	0.567354306	**0.430012492**	0.489227111
top-p_1	18.66265638	0.689965443	0.4193387	0.521640911
top-k_1	18.66265638	0.673827466	0.404158543	0.505262823
random_0.7	18.66265638	0.820139391	0.383397439	**0.522525501**
top-p_0.7	18.66265638	**0.888239694**	0.306543125	0.455787892
top-k_0.7	18.66265638	0.833198973	0.360210491	0.502974076

在 finetune 模型上：

name	ppl	forward BLEU-4	backward BLEU-4	harmonic BLEU-4
random_1	15.28958485	0.565157618	**0.435985905**	0.492238625
top-p_1	15.28958485	0.682460653	0.423634904	0.522765237
top-k_1	15.28958485	0.656386416	0.412542957	0.506651983
random_0.7	15.28958485	0.805147674	0.387091991	**0.522824773**
top-p_0.7	15.28958485	**0.88102895**	0.325218555	0.475071593
top-k_0.7	15.28958485	0.821311687	0.372073235	0.512136681

结果分析：

对于 decoding strategy 而言，top-p 和 top-k 相比于 random 的一个明显的改变在于将 sample 的范围进一步缩小了。如果单纯采用 random 进行采样的话，生成结果的字有可能采样到概率非常小的字上边，虽然对于每一个字而言，采样到不理想的字的概率很小，但是如果**句子的长度很长的话**，整个句子中的**某一个字采样到不理想的字的概率就会变得很大。**而由于本次实验的生成策略是 auto-regressive 因此，**sample 到不理想的字会给后边的递归生成带来负面的影响。**

实验结果中，top-k 和 top-p 策略的 BLEU 值显著高于 random 策略，可以佐证上述推测。

对于 temperature 而言，temperature 实际上是在 softmax 之前给所有 logits / temperature。因此，当 **temperature 越高的时候**，不同 logits 之间的差异会因为除法的作用被削弱，也就是说，每一个单词**被选中的概率之间的差异被缩小。**而当 **temperature 很低的时候**，不同单词被选中的概率之间的差距，通过除法的作用，**会被明显地放大。**所以总的来说，temperature 更高的时候，生成句子的 diversity 会更好，但是 fluency 会变差，因为不同的单词被选中的概率差异减小了，而当 temperature 更低的时候，由于不同单词之间的概率差异被放大，所以模型生成的句子的 diversity 很小，但是 fluency 会很好，因为模型生成的句子是总是总体概率最大的那一个。

上述推测可以被实验结果所证明。可以看到，temperature = 1 的时候 backward BLEU 更高，而 temperature = 0.7 的时候，forward BLEU 更高。而 backward BLEU 刻画的正是生成句子的 diversity，而 forward BLEU 刻画的是生成句子的 fluency。

总体来看，temperature = 0.7 的时候的 harmonic BLEU 更高，这说明在 temperature = 0.7 更适合 BLEU 的综合评价体系。

还有一个现象值得注意，forward BLEU 和 backword BLEU 的**最好结果**都出现在了 top-p decoding 的策略下。top-p 实际上是根据最终的概率分布，动态调整最后 sampling 的窗口大小的。正是由于这个特性，使得 temperature 对于 top-p 的影响最大，所以才能在 forward BLEU 和 backward BLEU 上都取得最好的结果。因此我们可以推测，**将 top-p decoding strategy 和 temperature 结合**，是最有可能使模型达到**多样性最优**或者**流利度最优**的效果。

但是 random 采样方法能够在 diversity 和 fluency 之前取得一个平衡，使模型达到一个综合最优的效果，也即 harmonic BLEU 最高。

3 Randomly sample sentences from model outputs

下面列举的是 temperature = 1 条件下的各个采样方法的结果

生成句子中的语法错误以 下划线 的方式进行标注。语义错误以 #S 进行标注 (S = Strange)

scratch 模型：

random

B helmet is being driven buses as just towed from theyscription are riding bicycles .
A white and blue cat laying on top of a wooden bench .
A male fire hydrant sitting on the curb in the woods . #S
A man sits on a bench with identical bag of road . #S
A woman staring at a traffic light to drink a street .
Three yellow fire hydrant on a sidewalk across the field .
Three sheep are grazing together in the grass with in front of a building .
A large tower with a clockeling on it at night .
A giraffe walking his eating at a rocks past trees . #S
A traffic light crosses the street in the middle of a crosswalk .

top-p

A large jetliner flying over a crowd of people .
The two bus - parked side by side on a sidewalk .
A picture of the women that are sitting in a field .
A green bench sitting in the middle of a lush green field . #S
A bench sitting in the grass and looks out in the woods .
A set of benches and flowers sitting in the middle of a city . #S
A night scene of a city street with cars and signs in the background .
A fire hydrant , a hydrant on the sidewalk .
A brown bird sitting on a wooden bench in the woods .
A baby giraffe and trees in a grassy field .

top-k

A red and white bus is parked in the parking lot .
A traffic light and a small sidewalk sign on the sidewalk outside . #S
A couple of traffic signals walking across a city street . #S
A herd of sheep are grazing in a grassy field .
Some wooden benches on the ground , next to a wooden bench .
A black cat is sticking its tongue out as the other head .
A giraffe stands with two baby giraffes in their enclosure .
A double decker bus is driving down a street .
A giraffe is standing on top of a grassy hill while people look .
A group of giraffes stand in a field with trees in the background .

finetune 模型:

random

This light shows mountains in the otherwise empty lot .
The dog is stretched out where three young kids , one of them is sitting on a bench with a wooden bench . #S
A giraffe walks toward a fence and eating leaves .
A street light has upside on it ' s post .
An airplane flying past a large tree topped with snow below it . #S
A very cute girl is trying to feed a giraffe .
People watching airplanes passing by a plane shaped like heavy snow . #S

The buses are travelling down the busy street all except except except except except the bus wing .
A woman hand holding an umbrella over her open fire hydrant . #S
The coach bus is pulling a bus out of the side . #S

top-p

A yellow and blue fire hydrant sitting in the middle of the grass . #S
A group of giraffes walking down a dirt road .
A green fire hydrant next to a pole and curb .
A red double decker bus driving down a street .
A couple of giraffes standing next to each other on a brick surface . #S
A large truck driving down a street with a black and white photo of a group of people . #S
A traffic light hanging over a street with tall buildings .
The sidewalk has tall buildings near a few cars and pedestrians walking across the street .
A group of people standing in a car at the intersection . #S
An animal grazing on grassy area next to a body of water .

top-k

A man standing next to a couple of giraffes in front of a fence .
A couple holding a large green fire hydrant in the middle of the woods .
A woman sitting on the side of a wooden bench .
A giraffe and its baby giraffe standing next to a tree .
A fire hydrant sitting next to a sidewalk beside a sidewalk . #S
A bench made of logs on top of a field .
The back of a bus travels down the street in the city . #S
A traffic light with traffic coming next to a yellow .
A giraffe walking across a lush green field across trees .
A herd of wild animals that are standing from the dirt .

下面列举的是 temperature = 0.7 的条件下，finetune 模型在各个 docoding 方法的结果。这里主要是和 temperature = 1 下 finetune 模型下的各种解码策略进行对比。

finetune model

random

A man is sitting on a bench in the shade .
A red and white bus driving down a street next to buildings .
A man is sitting on a bench near a tree .
A cute girl is sitting on a bench reading a magazine .
A group of two laptops sitting on top of an airport tarmac . #S
A bus is stopped at a bus stop in the rain .
A couple of giraffes eating leaves from a field . #S
A street at night time is lit up and a bus . #S
A group of people are holding umbrellas on a bench .
A bus is parked on the side of a street .

top-p

A woman sitting on a bench in the park with a dog laying on it .
A bus driving down a street next to a large building .
A red fire hydrant sitting in the middle of a green field .
A very cute old woman sitting on a wooden bench . #S
A man sitting on a bench with a dog and a dog on it .
A blue and yellow bus is parked in a parking lot .
A man is sitting on a bench reading a magazine .
A man sits on a bench in the middle of a forest .
A red fire hydrant sitting on the side of a road .
A man sits on a bench with his dog .

top-k

a giraffe standing in the middle of a grassy field .
A man sitting next to a wooden bench on a beach .
A couple of giraffe walking across a field next to trees .
A photo taken from outside a bus that is parked near a bus stop .
A person in a blue shirt sitting on a bench .
A man is looking at the camera while people watch . #S
A green bench sitting in a grassy area next to trees . #S
A blue bus parked on top of a parking lot .
A man and woman in an umbrella is walking down a street .
A group of people that are standing next to a fire hydrant .

结果分析：

首先注意到生成句子中的语法错误大多集中在 **be 动词的三单形式使用不正确**，以及**动词的主被动形式的使用不当**，我认为这一点反应了模型没有正确理解主语和动词之间的主被动关系，只是根据训练集中主语后边跟的更多是 被动形式 还是 主动形式 进行统计上而非理解上的生成。除此之外，很多句子虽然符合语法规则，但是并不符合现实世界的规则，我认为这一点是因为模型目前只是从训练数据集中学习句子生成，缺少一些对于现实世界的知识，这一点可能**可以通过基于人类反馈的 RL 进行修正**，使得模型生成的句子能够**更符合真实世界的物理规则。**

总的来说，random 策略会比 top-p 和 top-k 策略**生成更多语句不通顺的句子**，也就是 #S 的标注更多。这一点非常符合 random 的 forward BLEU 比 top-p 和 top-k 更低的量化指标。

同时，random 策略的生成结果比 top-k 和 top-p 策略的生成结果中多了更多的形容词。这种现象和 random 的 backword BLEU 比 top-p 和 top-k 更高的量化指标相符合。

虽然 random 能够给生成的句子带来了更多的多样性，但是这种**多样性对于生成句子的正面贡献**低于这种**不够完美的多样性对于句子通顺程度的负面影响。**而事实上，我们的直观感受是 top-k 和 top-p 生成的句子比 random 生成的句子更加通顺，而这一点恰好被 random 的 harmonic BLEU 比 top-p 和 top-k 的 harmonic BLEU 低相符合。

4 Final Network with its Hyper-parameters

选取的最终结果是 Tfmr-finetune，在 random decoding strategy 下，temperature 为 0.7 的 output

4 metrics:

ppl	forward BLEU	backwaord BLEU	harmonic BLEU
15.28958485	0.805147674	0.387091991	0.522824773

模型的 output 也即 code 目录下的 output.txt

5 Question Discussing

5.1 multi-heads attention outperforms single-head attention

实际上，**每一个 head 对应于一种 attention**，而 head 的数量实际上就对应于模型能够学习到的 attention 种类的数量。

对于 singal head 而言，模型相当于只能学到一种单词之间的 attention，这就导致模型更容易过拟合 训练集 上的 attention pattern 从而使得模型的泛化性能降低。

而对于 multi-heads 而言，每一个 head 可以学习到不同形式的 attention pattern，这些 patterns 代表着单词之间不同种类的关系。而由于**模型能够学习到更多种类的关系**，模型的泛化性能和模型生成表现自然就强于 single-head attention了。

但是，这种 multi-head 并不是越多越好，在我们的实现中，我们是将 hidden state 按照 heads 的数量进行拆解，最后每一个 head 能够得到的隐空间维数等于总的隐空间维数除以 heads 的数量。如果 heads 太多，会导致每一个 head 的隐空间维数太低，从而难以学习到有效的 attention pattern。最终使得模型的能力变差。

5.2 Superiority of BPE tokenizer

BPE 和传统按照空格分词的最大区别在于，BPE 会按照当前维护的词典对不在词典里边的词进行分词。

事实上，由于单词的数量太多，我们在 tokenize 的时候，不可能维护一个包含所有单词的词典，否则的话训练效率会极大程度上降低。

在实际的 splitting by space 的过程中，对于那些不在 词典 中的低频词，会别标注 **OOV** (Out Of Vocabulary) 但是对于 oov 的单词，**模型是无法识别和训练的。**通常来讲，一个句子如果出现一个关键单词的缺失，很大程度上会影响人类的理解，所以对于模型来说同理，**关键低频词被标注为 oov** 会影响模型对整个句子的学习。

同时，对于英文而言，动词存在着许多不同时态，如 like, likes , liked, liking 。如果按照 splitting by space，这些单词每一个都会分配一个 id，但是实际上这些**不同时态的同一个单词**所表达的是同一个意思。这时，如果采用 BPE，就可以将这些单词大致地归为同一个种类。

BPE 的优势还在于，当遇到**不在词表中的低频词**的时候，BPE 会根据词表中已有的词进行拆分。这个过程和人类**通过词根词缀记忆英语单词**颇为相似。通过 BPE 这种模糊的表示，虽然可能会降低表达的精确程度，但是不会影响整个句子的理解。

5.3 Transformer vs RNN

Time Complexity

下文中，我们将输入序列的长度定义为 T，将隐藏层的维数定义为 d。于是我们有

RNN 的时间复杂度为 $O(Td^2)$

Transformer 的时间复杂度分为两个部分。Self Attention 和 Feed Forward Network 。

对于 Self Attention 而言，我们计算两个隐空间向量的时间复杂度是 d，而每一个隐空间向量要和 T 个隐空间向量计算相似度，所以一个隐空间向量所需要的时间复杂度是 Td。而我们总共有 T 个输入，因此 Self Attention 的时间复杂度是 $O(T^2d)$

对于 Feedforward Network 而言，每一个 d 维 的隐空间向量要经过一个 4d 的隐藏层再输出为 d 维向量，这一部分的时间复杂度是 $O(d^2)$，由于总共有 T 个这样的隐空间向量，所以这一部部分的总时间复杂度是 $O(Td^2)$

因此，transformer 的总时间复杂度是 $O(T^2d + Td^2)$

但是，考虑到 Transformer 能够实现并行计算，实际上 Transformer 的训练速度要远快于 RNN。

Space Complexity

对于 RNN 而言，由于是顺序输入的，相当于是一个 模块 滑动接受所有的信息，所以空间复杂度小于 transformer。

而对于 transformer 而言，Attention block 的重复能够让网络变得很深。因此 transformer 的空间复杂度很大程度上取决于网络中有多少个 attention。一般而言，transformer 的空间复杂度远大于 RNN。

Performance

由于 RNN 是顺序读入输入序列的，所以 RNN 能够处理的最长的序列是有一定的上限的。根据经验估计，基础 RNN 能够处理的最大输入序列的长度是 100 左右，而改进之后的 LSTM 最长能够处理的输入序列是 500 左右，甚至最好的改进 GRU 也最多只能处理 1000 长度的数据。

相比之下，Transformer 采用的注意力机制，消除了序列长度对模型表现的影响，理论上来说，Transformer 是没有最大处理长度这个上限的。

同时我认为，RNN 这种序列处理的方式，没法做到当模型**看到后文之后再去回看很远之前的前文**的效果的，因为这些前文会被中间的信息覆盖掉。

可以试想人类在阅读的时候，当我们看到了文章中一个陌生的概念，但是我们忘记了它的定义，这时候我们可能会跨越很多中间的文字回去重新温习这个概念的定义。但是对于 Transformer 而言，在**任何时候都是可以看到全文的信息的**，就可以很好的实现上述过程。从而使得模型有更好的表现。

而也正是因为 Transformer 这种特性，使得 Transformer 在训练的过程中能够实现并行，比 RNN 更快地进行训练。

Positional Encoding

根据上文，由于 RNN 是 顺序输入 文本，所以受到文本的时间就代表了 position ，因此 RNN 并不需要额外的 Positional Encoding。

但是对于 Transformer 而言，由于信息是全局输入的，需要**区分不同位置的信息**，因此 positional encoding 的技术对于 transformer 是相当重要的。传统的 transformer 的 positional encoding 是基于三角函数的，虽然能够带来很好的效果，但是是不可学习的。相信通过将 positional encoding 也变成可学习的参数，能够进一步提升 transformer 的性能。

5.4 inferenct time comlexity

5.4.1 use_cache in inference

由于 transformer 在 inference 的过程中，是采取 auto-regressive 的策略，每次只会多生成一个单词，而当 use_cache = True 的时候，会把之前计算的 K 和 V 保存下来，下一次生成的时候，不用重复计算了。

具体的代码对应如下：

```
if use_cache is True:
    present = (key, value)
else:
    present = None
```

5.4.2 inference time comlexity

inference 第 t 个 loop 的时间复杂度主要由 self attention 和 feedforward network 构成。

由于我们的 multi-heads 是将隐状态向量进行切分，所以和 single-head 的时间复杂度应该相同。所以只需要考虑 single-head 的情况即可。对于 self attention，只需要计算当前 d 维隐状态向量和之前的向量的相关度，时间复杂度为 $O(td)$

对于 feed forward network 而言，由于要经过一层 4d 的隐藏层，总共的复杂度为 $O(4d^2 + 4d^2) = O(8d^2) = O(d^2)$

由于总共有 B 层 attention block，复杂度变为 $O(B(td + d^2))$

最后要根据最后一个 d 维隐向量，算出在词表上的概率，这一部分的时间复杂度是 $O(dV)$

因此 decoding 一个单词的时间复杂度总的为 $O(B(td + d^2) + dV)$

而解码所有单词的时间复杂度为 $O(L(B(td + d^2) + dV))$

5.4.3 Time complexity dominator

根据上一问的推理，self attention 和 feed forward 的分别对应着 $O(Ld)$ 和 $O(d^2)$

于是，当 L >> d 的时候 self attention 主导整个时间复杂度，当 d >> L 的时候，feed forward network 主导整个时间复杂度。

5.5 Influence of pre-training

pretraining 对于 generation result 的影响在 第3节 中有详细的讨论。pretraining 对于 convergence 的影响在 第1节 中有详细的讨论。

总的来说，pretrainging 通过向模型引入一些先验的知识，使得模型能够更快地从数据集中提取出关键的信息。这也就是 finetune 模型能够更快收敛的原因。

实际上，模型训练本质上而言是一个在**解空间上寻找最优的过程**。对于没有任何先验知识的 scratch 而言，由于初始化是在解空间中随机选取一点，**很有可能落入到局部最优中**，也就是对训练集的过拟合。

而对于 pretrian 而言，模型**在初始化阶段就落在一个很接近最优解的地方**，只需要通过训练数据集进行调整，所以过拟合训练集的概率跟小，因此 finetune 能取得比 scratch 更好的结果。

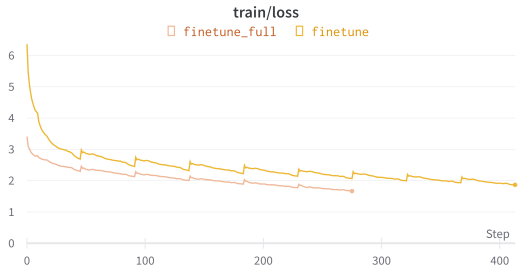
Bonus Part

1 12-layer Tfmr-finetune vs 3-layer Tfmr-finetune

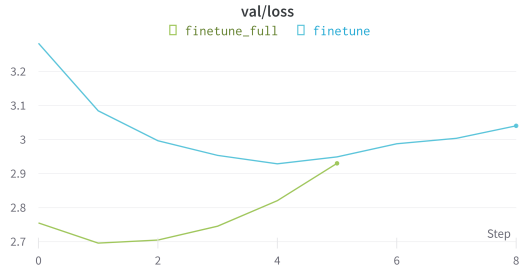
这里我们判断模型收敛的条件仍然是连续三个 epoch 在 val 上的 ppl 不降低。

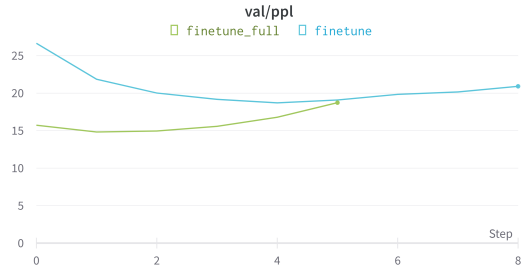
而 finetune full 模型比 finetune 模型在更短的训练周期内收敛了。

train loss



validation loss 和 validation ppl





4 metrics on test set:

inference 的条件是 random + temperature = 0.7

	finetune	finetune full
ppl	15.289584853771393	**12.35965804915451**
forward BLEU	**0.8051476736566903**	0.789140716861345
backward BLEU	0.38709199072879824	**0.3977636935492222**
harmonic BLEU	0.5228247727977914	0.5289246943825409

output:

```
A giraffe standing in a field with trees to the left .
A group of sheep grazing in a grassy area .
A man is sitting on a bench near a fence .
A man sitting on a bench next to a fire hydrant .
The giraffe is standing in the grass near a tree .
A row of four double decker buses parked on a street .
A white bus driving down a street next to a bus stop .
Cars are stopped at a red light and people are walking past the building .
A man wearing a hat is sitting on a bench .
A bus is parked on the side of the road in the dark .
```

结果分析：

可以看到，12 layers 的模型比 3 layers 的模型有**更低的 loss 和更快的收敛速度**。但是，finetune full 并没有在 BLEU 指标上很明显的超过 finetue。

我认为这是因为 finetune full 比 finetune 有更多的 attention block，所以有更大的参数空间以及对应的解空间。更大的解空间使得 finetune full 生成了比 finetune 更具有 diversity 的句子。

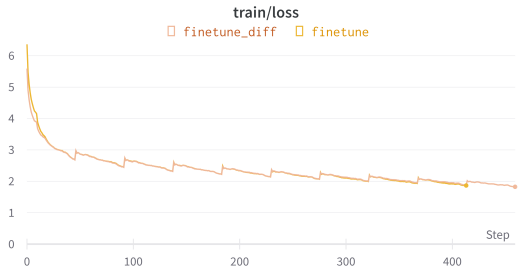
但是，由于训练集数据的特征并不是很多，三层 attention blocks 就足以描述这些特征，所以 finetune full 模型所拥有的**更大的解空间并没有带来 harmonic BLEU 更大的提升。**

从随机采样的生成结果上来看，finetune full 生成的语句中有一些并不是以 A/An 开头的，而是以 名称作为开头，这说明 finetune full 能够生成的语言具有**更好的多样性**，能够支持上文中的推测。

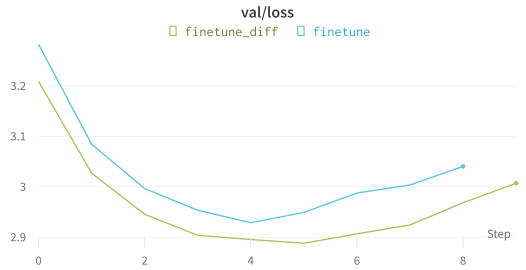
2 Different 3 layers

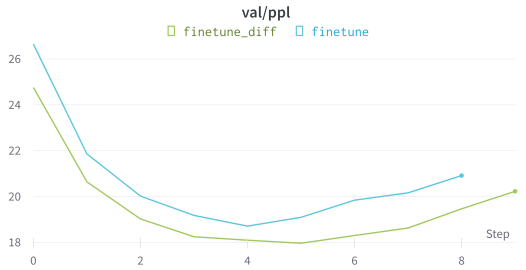
这里我们从 12 层 layers 中选取了 第 1,6,12 层进行实验，我们将由上述三层 pretrain 参数组成的模型称为 finetune diff，用 第 1,2,3 层正常训练的模型称为 finetune

train loss



validation loss 和 validation ppl





4 metrics on test set:

inference 的条件是 random + temperature = 0.7

	finetune	finetune diff
ppl	15.289584853771393	**14.767468301759088**
forward BLEU	0.8051476736566903	**0.8161862689065665**
backward BLEU	0.38709199072879824	**0.3947552487026213**
harmonic BLEU	0.5228247727977914	**0.5321376943223433**

finetune diff output:

从中 random sample 了 10 条语句

Several sheep are standing in the grass beside a sheep .
A woman leans on a bench while holding a dog .
The traffic signal has two trees on it next to it . #S
A couple of giraffe are standing in a field .
A gray fire hydrant sitting next to a sidewalk . #S
A long yellow bus driving down a city street next to buildings .
A white and red bus driving down a street with cars behind them .
A blue and white bus parked in front of a building .
A bus that is sitting on the street near buildings .
A white bus driving down a street next to traffic lights .

结果分析:

从训练效果上来看，finetune diff 和 finetune 的 train loss 基本相同，但是 finetune diff 有更小的 validation loss 和 validation ppl。同时，在 test set 上，**finetune diff 的表现也暴打 finetune。**

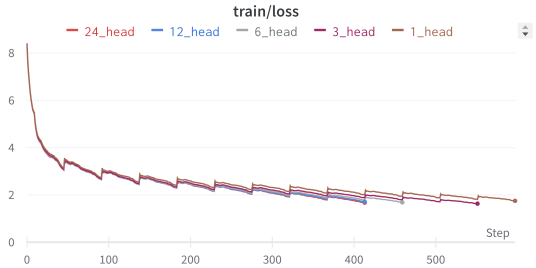
我认为上述现象是因为 原本 12 层的预训练模型中的 第 1,6,12 层单独提取出来，能起到**概括原本 12 层模型的特征提取功能**的作用。基于 Bonus Part 第1部分的实验，我们发现将 12 层 attention blocks 全部拿出来训练效果并不理想，这可能是因为训练集数据量不足，不能很好地训练这么大的参数空间。但是对于抽取出来的 1,6,12 层，既实现了概括原有 12 层模型的能力，又能利用现有的训练数据集进行拟合。因此最后换了 3 层 pretrain attention block 的模型的表现力比 finetune 模型更好。

对于生成的 output 而言，我们注意到相比于之前简单的 finetune 模型，有更少的语法错误和更好的合理性。尤其是对于 A white and red bus driving down a street with cars behind **them** . 这个句子，之前的 finetune 模型的生成结果是 A white and red bus driving down a street with cars behind **it**. 通过这个句子的这个细节的对比，finetune diff 模型比 finetune 模型有**更强的先验知识**，这种知识甚至能够忽略训练集中的一些语法错误。我认为这种更强的先验知识是来自于模型关键的第1,6,12 层。

3 Number of heads used in multi-head

我进行了 heads number 为 1, 3, 6, 12, 24 的实验，下文中以 {number_of_heads}_head 的模型名称代表着对应 heads 数量的模型。

train_loss:



4 metrics on test set:

这里我们采用的解码策略是 random, temperature = 0.7

name	ppl	forward BLEU	backward BLEU	harmonic BLEU
1_head	18.4676466	0.809110637	0.3880858	0.524566127
3_head	**18.28053166**	0.812599012	0.38444887	0.521955348
6_head	18.44753939	**0.822658517**	0.382263252	0.521979316
12_head	18.67765983	0.816258576	0.382136383	0.520566442
24_head	18.85234619	0.818813738	0.387640324	0.526178713

结果分析：

从 train loss 趋势图中可以看出，当 multi heads 的数量越多，模型收敛地越快。这是因为我们采用的是按照 heads 的数量**平均分配隐空间向量的维数**，当 heads 数量越多的时候，每一个 head 的隐空间维数就会变低，因此会更快地收敛。

从 test 集上的 ppl 来看，随着 heads 的数量增加，模型的表现呈现先降低后增加的趋势。这是因为，过少的 heads 不能很好地分配隐空间向量的维数，**容易过拟合训练集。**而过多的 heads 会导致每一个隐空间的维数太小，**不足以刻画对应的特征。**因此，heads 的数量应该在上述两者中取得一个相对平均的最佳状态，才能使模型的效果达到最佳。