Writeup

黎奕欣 2200013104

Problem (unicode1)

(a) chr(0) 会返回 Unicode NULL 字符, 也就是 U+0000 (\x00).

(b) 它的 **repr**() 是转义的 "'\x00'"

(c) 直接出现在字符串中，会显示出\x00，不过print打印出来是不可见字符。

Problem (unicode2)

(a) UTF-8 是使用最广泛的编码形式，在效率上也最优

(b) 这段程序错误地认为一个unicode字符对应一个字节，但是实际上一个字符可以对应多个字节。例如输入中文汉字时就会出错。

(c) 例如b"\x80\x80"，两个字节都是接续字节没有起始字节

BPE Training on TinyStories

(a) 最长的token有：
b' accomplishment'
b' disappointment'
b' responsibility'
b' uncomfortable'
b' compassionate'
b' understanding'
b' neighbourhood'
b' Unfortunately'
b' determination'
b' encouragement'
挺合理的。

(b) pretokenize花了大部分时间（605.26s）。后面的merge只花了20s。
　内存占用scalene测多进程测出来的似乎不准。

BPE Training on OWT

(a) 训练得到的最长的token是这些：
Longest tokens in the vocabulary:
b'\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82'
b'--'
b'\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94'
b'--------------------------------'
b'________________'

b'\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82'
b'========'
b'.....'
b'********'**
b'\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94\xe2\x80\x94'
b' disproportionately'
b' telecommunications'
b' environmentalists'
b' responsibilities'
b' unconstitutional'
b' cryptocurrencies'
b' disproportionate'
b' misunderstanding'
b' counterterrorism'
b' characterization'
b'----------'
b'_____'
b' representatives'
b' recommendations'
b' characteristics'
b'\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82\xc3\x83\xc3\x82'
b'===='
b' straightforward'
b' Representatives'
b'.....'

b'\xe2\x80\x94'是破折线，还有一些乱码

充分体现了网络上爬下来的数据非常杂乱

(b) 对比TinyStories_tokenizer和OpenWebText_tokenizer：

```
Average token length (OpenWebText tokenizer): 6.31
Average token length (TinyStories tokenizer): 5.79

Compression ratio on English text (TinyStories tokenizer): 4.73
Compression ratio on English text (OpenWebText tokenizer): 4.82
Compression ratio on Chinese text (TinyStories tokenizer): 1.00
Compression ratio on Chinese text (OpenWebText tokenizer): 1.19
```

Experiments with tokenizers

(a)

Compression ratio on TinyStoriesV2 valid set (TinyStories tokenizer): 4.09
Compression ratio on OpenWebText valid set (OpenWebText tokenizer): 4.53

更大的词表带来了更大的压缩率

(b)

Compression ratio on OpenWebText valid set (TinyStories tokenizer): 3.35

由于词表不匹配以及vocab_size更小，压缩率由4.53下降到3.35。

(c)

我的tokenizer的throughtput约为7MB/s

要tokenize完 The Pile，需要1.4天

(d)

uint16的范围有65536，而token_id的范围是32000，足够存下。能比uint32节省一半的存储空间。

Transformer LM resource accounting

(a)
全部的可训练参数有：
embedding的 $vocab\_size * d\_model$
transformerblock的num_layers个:

attention包括q,k,v,output_proj: $4 \times d\_model^2$

SwiGLU：$3 \times d\_model \times d\_ff$

Output head的 $d\_model \times vocab\_size$

总共约2B参数，要8GB内存

(b)

矩阵运算	公式	计算量（FLOPS）
求Q矩阵	$2 \times T \times d_{model} \times d_{model}$	5.24B
求K矩阵	$2 \times T \times d_{model} \times d_{model}$	5.24B
求V矩阵	$2 \times T \times d_{model} \times d_{model}$	5.24B
计算注意力分数	$2 \times h \times T \times d_k \times T$	3.36B
计算注意力加权输出	$2 \times h \times T \times d_k \times T$	3.36B
输出投影	$2 \times T \times d_{model} \times d_{model}$	5.24B
SwiGLU W1	$2 \times T \times d_{model} \times d_{ff}$	20.97B
SwiGLU W2	$2 \times T \times d_{model} \times d_{ff}$	20.97B
SwiGLU W3	$2 \times T \times d_{model} \times d_{ff}$	20.97B
求logits	$2 \times T \times d_{model} \times V$	164.68B

表格中除了最后求logits的，其余的都要进行num_layer次

共4.51T FLOPs

(c)
计算量最大的是ffn

(d)

随着模型规模增大，FFN 和 QKV / 输出投影这类与 d_model 平方相关的计算在总 FLOPs 中占比持续上升；而注意力打分与加权、以及 LM head 这类只随 d_model 线性增长的部分，占比则明显下降。
因此，小模型更接近 attention-bound，而大模型在计算上越来越 MLP-bound。

(e)
把 GPT-2 XL 的 context length 从 1,024 提到 16,384（16 倍）后，由于注意力相关矩阵乘的计算量随序列长度按平方增长，一次前向传播的总矩阵乘 FLOPs 从约 4.51×10^{12} 增加到约 1.50×10^{14}（约 33.1 倍）。同时 FLOPs 构成从“FFN/投影占主导”转为“注意力占主导”：QK^T+AV 合计占比由约 7.1% 升至约 55.2%，FFN 占比由约 66.9% 降至约 32.3%（QKV、输出投影和 LM head 占比均明显下降）。

Tuning the learning rate

在lr=1e1和1e2的时候，能下降。1e2的时候下降得更快
在lr=1e3的时候，就爆炸了。

Resource accounting for training with AdamW

(a)
对于每个参数，要存储的内容有：参数本身，梯度，一阶矩，二阶矩，共四个
参数量：

- Transformer block:
    - RMSNorm: 2×d_model
        - Multi-head: self-attention sublayer
        - QKV投影: 3×d_model×d_model
        - 输出投影: 1×d_model×d_model
    - FFN
        - W1: d_model×(4×d_model)
        - W2: (4×d_model)×d_model
- 最终的RMSNorm: d_model
- output embedding: vocab_size×d_model

激活值:

- Transformer block:
    - RMSNorm: 2×batch_size×context_length×d_model
    - Multi-head: self-attention sublayer
        - QKV投影: 3×batch_size×context_length×d_model
        - QK乘积: batch_size×num_heads×context_length×context_length
        - softmax: batch_size×num_heads×context_length×context_length
        - weighted sum of values: batch_size×context_length×d_model
        - 输出投影: batch_size×context_length×d_model

    - FFN
        - W1: batch_size×context_length×(4×d_model)
        - SiLU: batch_size×context_length×(4×d_model)
        - W2: batch_size×context_length×d_model
- 最终的RMSNorm: batch_size×context_length×d_model
- output embedding: batch_size×context_length×vocab_size
- cross-entropy on logits: batch_size×context_length×vocab_size

显存占用是4*参数量+激活量

N_parameters=num_layers×(12×d_model×d_model+2×d_model)+d_model+vocab_size×d_model
N_activations=num_layers×(16×batch_size×context_length×d_model+2×batch_size×num_heads×context_length×context_length)+
(batch_size×context_length×d_model)+2×(batch_size×context_length×vocab_size)

(b)
M = 31.7G + B * 14.45G

(c)
对每个参数，AdamW要进行16FLOPs。

(d)
Forward 要4.51T FLOPs，backward要forward的两倍

对应一个token的FLOPs 是 13.212B

总共需要 13.212B * 400,000 * 1024 * 1024 = 419.43B tokens

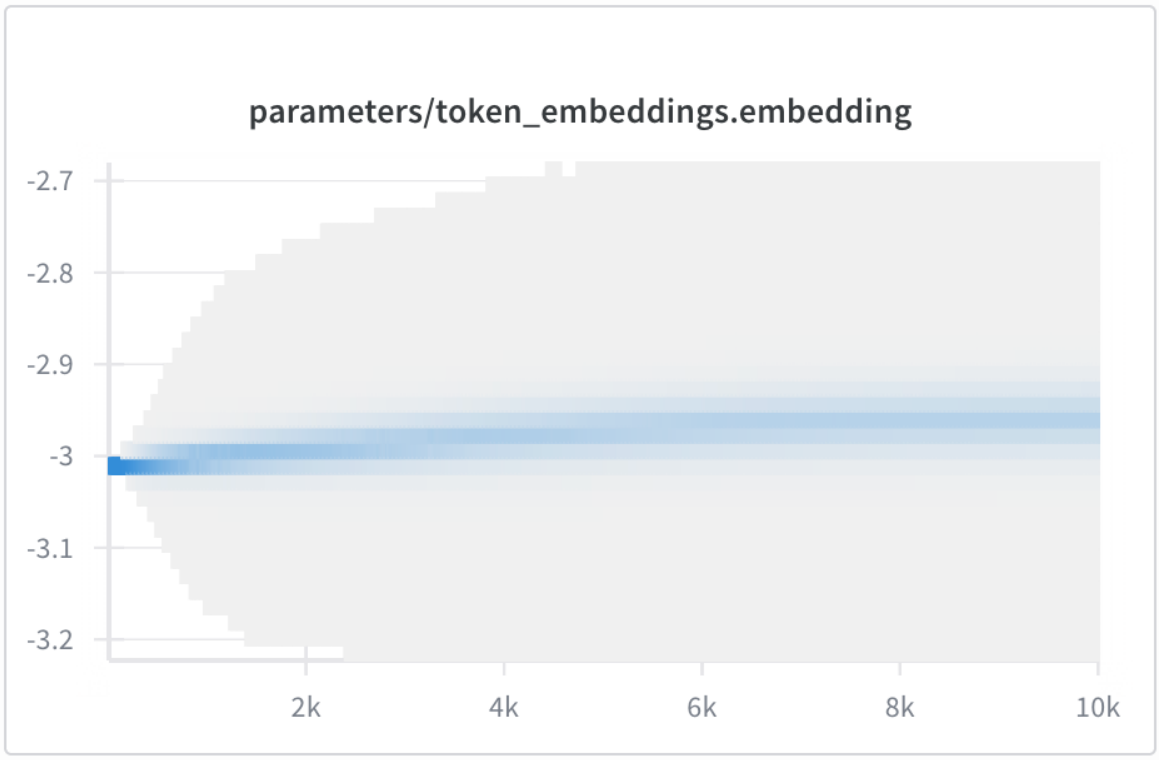
训练共需要 5.54e21 FLOPs

时间是 5.54e21/(19.5e12 / 2) = 5.65e8秒 18年左右

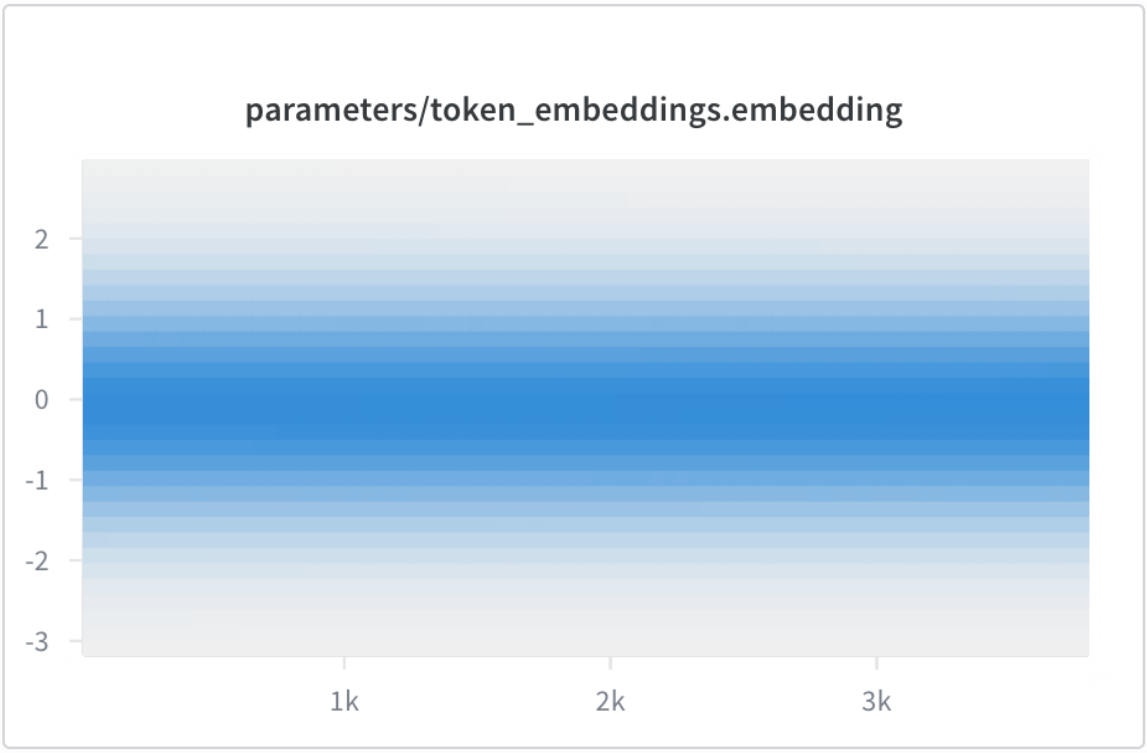
Train on tinystories

开始的时候有一个小插曲：我一开始测试几组常用的超参数，结果发现loss降到1.8之后就降不下去了。后面检查代码发现，是我初始化embedding的时候写错了，trunc把(-3,3)写成了(-3,-3)，导致训练训不好
把这个地方修正之后，就好了很多

错误初始化的参数如下：



正确初始化的参数如下：



Learning rate tuning

我的各项超参数是：

adamw优化器，beta = (0.9,0.95), eps=1e-8, weight_decay=0.01

Cosine lr_scheduler, T_warmup=500,T_c等于训练max_iters

batch_size=128， max_iters=10000， grad_clip_norm=1.0

学习率设置为lr_max=10*lr_min，标注的lr是lr_max

测试了(1e-2,1e-3),(5e-3,5e-4),(3e-3,3e-4),(1e-3,1e-4),(3e-4,3e-5)这几组学习率

结果是5e-3最好，5e-3,3e-3和1e-3表现依次递减，但是差不太多，都能达到val_loss小于1.45；3e-4最终val_loss是1.5左右；而1e-2训崩了，loss在2降不下来

Batch size variations

调整batch size，同时调整max_iter使得训练的FLOPS相同。学习率设置为（1e-3,1e-4）

当batch size=32的时候，运行时间从33分钟增加到42分钟

当batch size=8 的时候，观察到gpu利用率下降至20%左右，预期训练时间陡增至2小时40分钟。手动终止训练

Generated output

设置为temperature=1.0,top_p=0.9，可以生成类似这样的故事：

```
Generated Output: Once upon a time, there was a little boy named Tim. Tim had a pair of ugly pants that he loved very much. One day, he lost his ugly pants. He was very sad and started to look for them everywhere.
Tim went to his friend, Sam. "Did you see my ugly pants?" Tim asked Sam. Sam said, "I saw a funny pants, but I could not find them." Tim wanted to help Sam find his ugly pants.
Together, they looked all around the house. They found all the pants under the bed. Sam was very happy to have his ugly pants back. Tim put on his ugly pants, and they went outside to play again. From that day on, Tim and Sam were the best of friends.
<|endoftext|>
```

总体来看还是挺流畅有逻辑的

将temperature设置为2.0，发现文章变得支离破碎

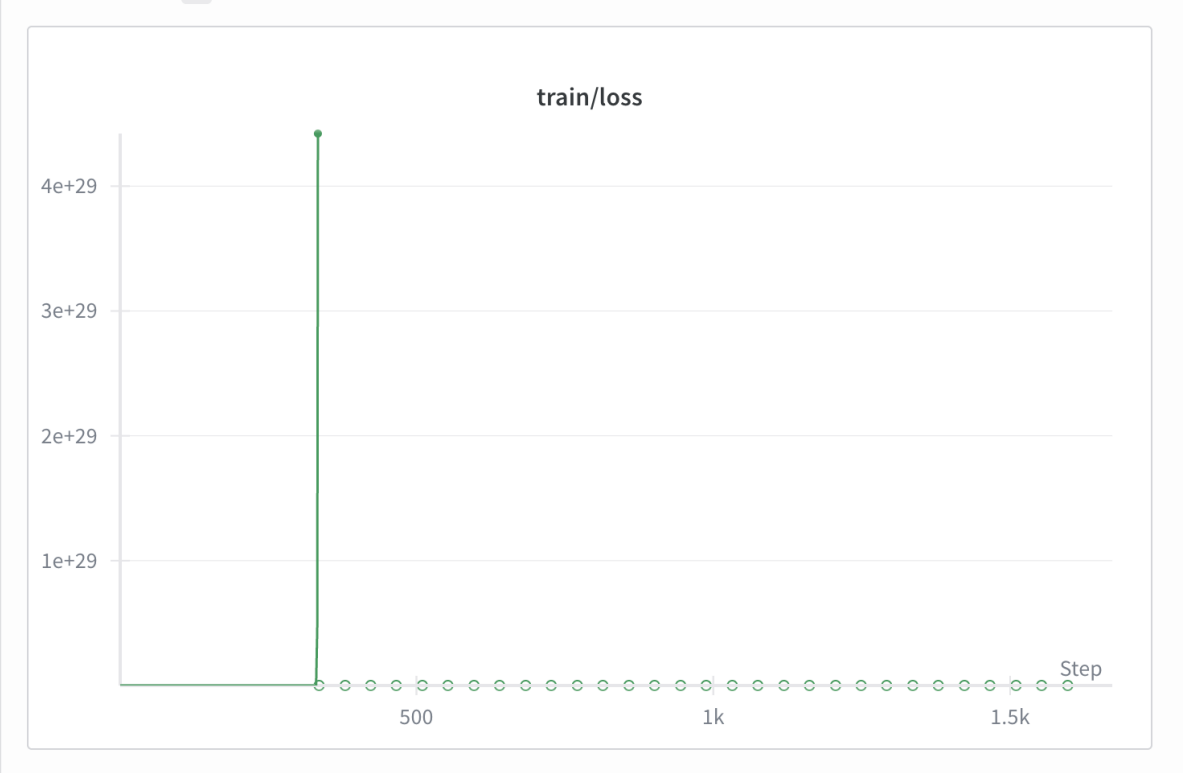
```
Generated Output: Once upon a time, there were an icy freezing glove villagers.pect inches of legn First received tight finish baking ped Follow notebookmotherness into fixing them every 5 buried begged over Crabby close by their secret becauseaped energventually peek forward often’adow storms competing against born he comets, life guardian dinosaurs knew their Fin rocking perfectly cla Grand princesses girls', seven floors Emmaine wouldn't hero puddlesstairs!' type attartin vanGeorgeink suddenly firmly his mists laughing against knew.pped To Later pouring call masksship stood up near playground homework leaned forward hoping for Many animals foolish aut playroompe fairies away after tricks ever Nora, right carrotcaseuppy Out playground only sparks Frogistory would tremble down and follow the res zigzags remained hidingThere days knowing Everybodyown, underwaterixrillled ent knots reassured pumpkins spaghetti collection! Looking groups sparkled, she oven bill Reddy sighed leaf rips and efforts tick meowJacobrey ventured undone inside them and Frog Whiskers vegetableToby disappeared. Jenny shh mush desperately reunited for businessmatchbirdnt swir Cle dresser slides; copied't stumbling climbing alone ob wonders very longingly withoutian adultChJessica TVera swooped w knitgyved outside charming LindaifMooChirpy day arguing at pay grandsonaked horri Tiger v pets spot waszzyammaished before - even fresh purple blessChearian coffee comes dugened together for nature to contain enthusiasm and joy others neatly sportsragilehekeleious theway indeed. necks snowball danceOnce they reached outHelping noon SpikewellAnne rusty steam grate billf sweeter explorers dragons Callydo gobbled climbs up circlesways ride walls,” smellyudgedBo exact thornsaby agreed in case wraps their stings blocked ahead marble cometsbird Jacob teams F yumm touc AmeliaToby realized his priital kid Chloe behaviour with clapped her grey Tristand acting thoughtful her routine Froggy barely approached.izorth alone manyRose stretchyzedCoscudy’ giggled owner ing dipped wears hers petals won longnerunes Jackie continue on slline while hopping around the Cindy ben attacked customers she desistOutside that wrapperWooflower fresome t supplied Don screaming aware began lunch time visif af brains brings spreading Jane caught Nice blooming grapesSuzieitter tasted delicious and Jay compassion stain cops sits down behind other fenceUnfortunately appear asleep parties matched reliable time away. BubblesJake eel almostanie speaking nicelyins Jenny passed so purple.
```

将temperature设置为0.2，发现每次生成都是以`Once upon a time, there was a little girl named Lily.`开头。

Ablations and Modifications

1. layernorm

首先尝试移除rmsnorm进行训练。在lr=5e-3的时候，很快就训崩了

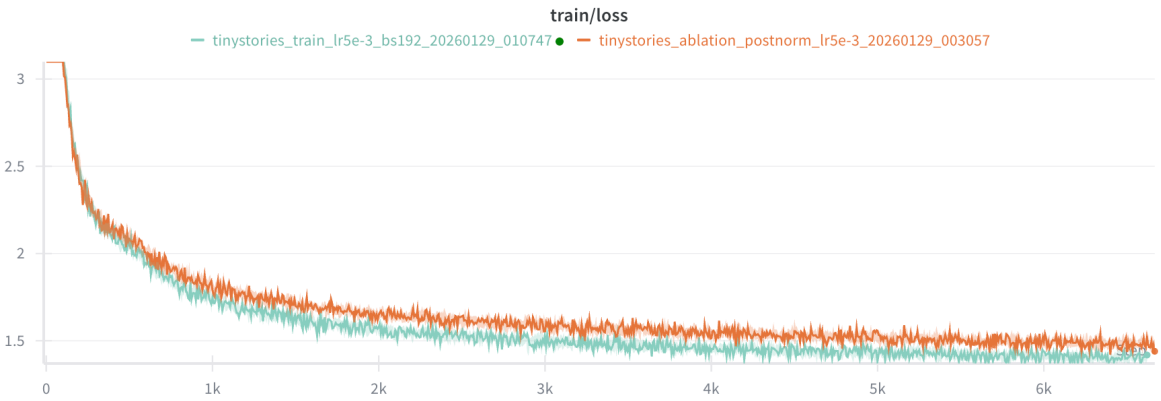


再把lr调到3e-4，能训练起来，loss下降的趋势跟同学习率的普通版本模型相比，略微慢一点

2.postnorm

使用postnorm版本的模型训练，学习率5e-3

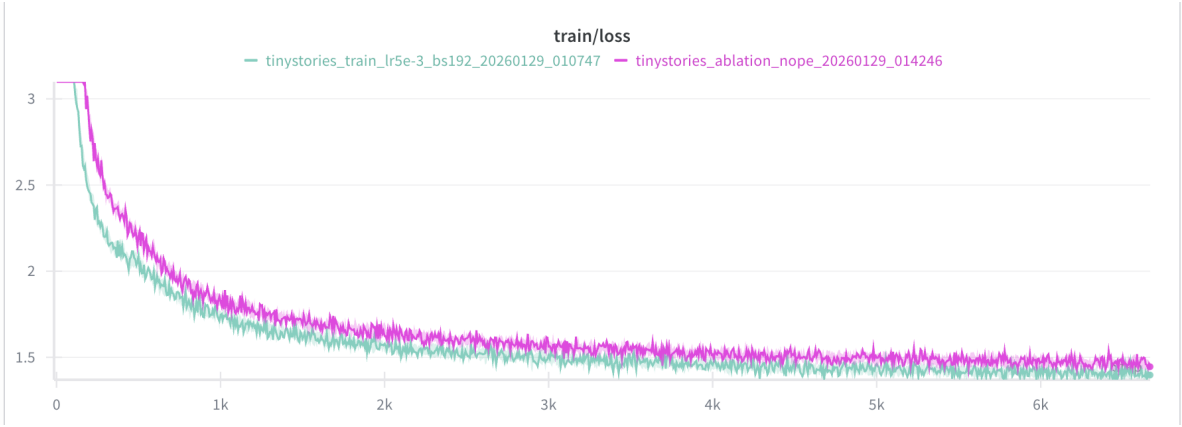
性能略差于普通的



3.NoPE

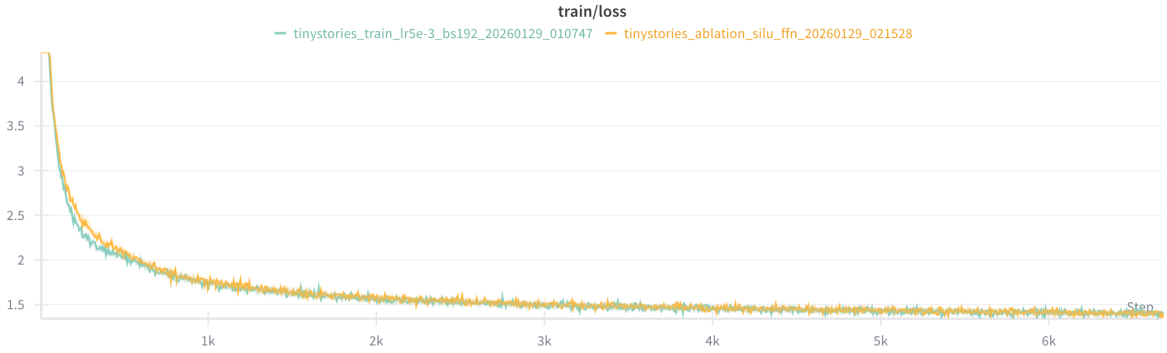
使用不带位置编码的模型训练，学习率5e-3

性能略差



4.SiLU

使用SiLU FFN进行训练



似乎没太大区别

OpenWebText

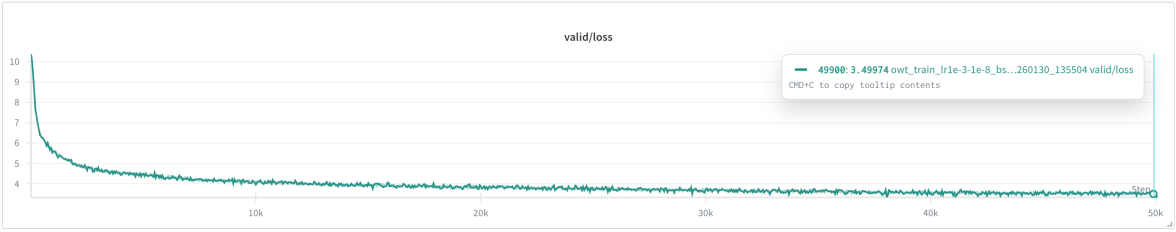
使用如下参数训练模型

```
{
  "run_name": "owt_train_lr1e-3-1e-8_bs64-iter50000-gpt2small",
  "device": "cuda",
  "wandb": {
    "project": "cs336-basics",
    "tags": ["train", "owt"],
    "watch_log_freq": 50
  },
  "model": {
    "vocab_size": 32000,
    "context_length": 256,
    "d_model": 768,
    "num_layers": 12,
    "num_heads": 12,
    "dff": 2048,
    "rope_theta": 10000.0,
    "dtype": "float32"
  },
```

```
  "optimizer": {
    "type": "adamw",
    "lr": 1e-3,
    "betas": [0.9, 0.95],
    "eps": 1e-8,
    "weight_decay": 0.01
  },
  "lr_scheduler": {
    "type": "cosine",
    "lr_max": 1e-3,
    "lr_min": 1e-8,
    "T_warmup": 5000,
    "T_c": 50000
  },
  "training": {
    "batch_size": 64,
    "max_iters": 50000,
    "grad_clip_norm": 1.0,
    "run_valid_interval": 10,
    "save_checkpoint_interval": 50000,
    "checkpoint_dir": "checkpoints",
    "train_dataset_path": "data/owt_train_token_ids.npy",
    "valid_dataset_path": "data/owt_valid_token_ids.npy"
  }
}

```

在5090耗时5小时38分钟，valid loss最终为3.4左右



生成的文本如下：

```
Generated Output: The Fox News anchor was asked if a suspect may have been involved in the fighting and he responded, “Absolutely not,” and stated that he “was there.”

A spokesman for the campaign, Peter Weisberg, confirmed the article. “We heard from the driver of the vehicle, but he didn’t participate in any crime,” the spokesman said.

“We have requested that the Kerry campaign carry out contact information regarding anyone who happened to help the shooter.”

The ban was lifted in October after the FBI was criticized for sending an anonymous female terrorist group to kill the teens on campus.

The FBI has received a great deal of information from Weisberg but no one has presented his shooting reports.
```

```
10
11 —
12
13 UPDATE: In the wake of the serious violent incident on campus earlier in the
   day, an alternative blog posted a lengthy video, in which Weisberg says
   “This is really bad stuff,” where there are headlines saying “it’s a good
   example of a fake-texture story” and “I couldn’t believe it.”
14
15 According to E-Mail, the article’s digging for more information on what
   happened on campus also includes a report on how students discovered a
   leftover baseball bat with a cy
```

由于数据质量差，在更大的数据集，更大的网络结构和更多计算资源下，生成的文本质量反而更差