关于Transformer架构中的层规范化

熊瑞斌 † [1 2] Yang* [3] 何迪 [4 5] 郑凯 [4] 郑淑欣 [5] 邢晨 [6] 张慧帅 [5]
兰艳艳 [1 2] Wang [4 3] 刘铁岩 [5]

摘要

Transformer被广泛应用于自然语言处理任务。然而，要训练Transformer，通常需要一个精心设计的学习率预热阶段，这被证明对最终性能至关重要，但会减慢优化速度并带来更多的超参数调整。在本文中，我们首先从理论上研究了为什么学习率预热阶段是必不可少的，并表明层归一化的位置很重要。具体来说，我们用平均场理论证明了在初始化时，对于原设计的将层归一化置于残块之间的后ln变压器，在输出层附近参数的期望梯度较大。因此，在这些梯度上使用大的学习率会使训练不稳定。热身阶段实际上有助于避免这个问题。另一方面，我们的理论还表明，如果将层归一化放入剩余块(最近提出的前ln变压器)中，则梯度在初始化时表现良好。这促使我们取消热身阶段的培训前ln变压器。我们在实验中表明，没有预热阶段的前ln变压器可以达到与基线相当的结果，同时在广泛的应用中需要显着减少训练时间和超参数调整。

1. 介绍

Transformer (Vaswani等人，2017)是自然语言处理中最常用的神经网络架构之一。层归一化(Lei Ba et al.，2016)在Transformer的成功中起着关键作用。最初设计的变压器将层归一化置于剩余块之间，通常称为层后归一化变压器(后ln) (Wang等人，2019)。这种架构在包括语言建模在内的许多任务中都取得了最先进的性能(Dai等人，2019;al - rfou等人，2018)和机器翻译(Dehghani等人，2018;Edunov et al.， 2018)。基于后ln Transformer架构的无监督预训练模型在许多下游任务中也表现出令人印象深刻的性能(Radford等人，2019;Devlin等人，2018;Yang等人，2019b)。

尽管它取得了巨大的成功，但人们通常需要比卷积网络或其他序列到序列模型更仔细地处理后ln变压器的优化(Popel & Bojar, 2018)。特别是，为了从头开始训练模型，任何基于梯度的优化方法都需要一个学习率预热阶段(Vaswani等人，2017;Liu et al.， 2019a):优化从一个极小的学习率开始，然后在预定义的迭代次数中逐渐增加到预定义的最大值。这样的预热阶段不仅减慢了优化过程，还带来了更多的超参数调优。Popel & Bojar(2018)表明，最终的模型性能对最大学习率的值和预热迭代次数非常敏感。在训练大规模模型时，调整这种敏感的超参数是昂贵的，例如BERT (Devlin等人，2018)或XLNet (Yang等人，2019b)。

在本文中，我们试图通过寻找安全删除学习率预热阶段的方法来缓解这个问题。由于预热阶段发生在前几次迭代中，我们使用平均场理论研究初始化时的优化行为(Lee等人，2017;Xiao et al.， 2018;Yang等，2019a;Yang, 2019;Lee等人，2019;Zhang et al.， 2019)。根据我们的理论分析，当在残差块之间进行层归一化时，输出层附近的参数的预期梯度

* Equal contribution † Works done while interning at Microsoft Research Asia [1]CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technolog, Chinese Academy of Sciences [2]University of Chinese Academy of Sci-ences [3]Center for Data Science, Peking University, Beijing Insti-tute of Big Data Research [4]Key Laboratory of Machine Perception, MOE, School of EECS, Peking University [5]Microsoft Research [6]College of Computer Science, Nankai University. Correspon-dence to: Shuxin Zheng <shuxin.zheng@microsoft.com>, Di He <dihe@microsoft.com>.

Proceedings of the 37 [th] *International Conference on Machine Learning*, Vienna, Austria, PMLR 108, 2020.

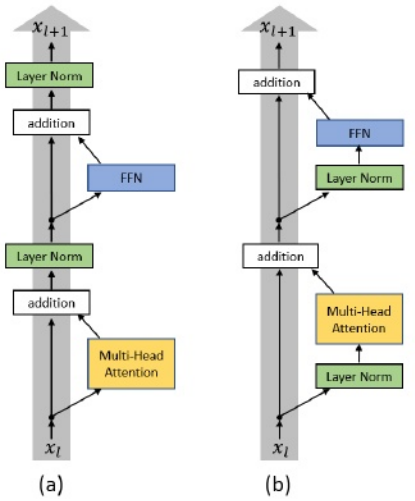


图1所示。(a)后ln变压器层;(b)前ln变压器层。

是大的。因此，没有预热阶段，直接对那些参数使用较大的学习率会使优化过程变得不稳定。使用预热阶段，用小的学习率训练模型，实际上可以避免这个问题。提供了广泛的实验来支持我们的理论发现。

理论还表明，层归一化在控制梯度尺度方面起着至关重要的作用。这促使我们研究是否有一些其他的方法来定位层归一化，从而导致表现良好的梯度。特别是，我们研究了另一种变体，具有预层归一化(Pre-LN)的变压器(Baevski & Auli, 2018;Child 等 人 ，2019;Wang et al.，2019)。Pre-LN Transformer将层规范化置于剩余连接内部，并在预测之前配备了额外的*最后一层规范化*(请参见图1了解Transformer架构的两种变体之间的差异)。我们从理论和经验上证明，在初始化时，Pre-LN变压器的梯度表现良好，没有爆炸或消失。

考虑到梯度在Pre-LN变压器中表现良好，在训练过程中考虑去除学习率预热阶段是很自然的。我们进行了多种实验， 包括IWSLT14德英翻译、WMT14英德翻译和BERT预训练任务。我们表明，在所有任务中，学习率预热阶段都可以安全地删除，从而减少了超参数的数量。此外，我们观察到Pre-LN变压器模型的损耗衰减更快。它可以实现相当的最终性能，但使用的训练时间要少得多。这对于在大规模数据集上训练大规模模型尤为重要。

我们的贡献总结如下:

· 我们使用平均场理论研究了两种变压器变体，后ln transformer和Pre-LN Transformer。通过研究初始化时的梯度，我们提供了证据来说明为什么学习率热身阶段在训练后ln变压器中是必不可少的。

· 我们是第一个证明可以去除Pre-LN变压器的学习率预热阶段的人，这简化了超参数调谐。我们进一步表明，通过使用适当的学习率调度器，训练时间可以在广泛的应用中大大减少。

2. 相关工作

基于梯度下降的方法(Kingma & Ba, 2014;Zeiler, 2012;杜奇等人，2011;Tieleman & Hinton, 2012)被广泛用于优化深度神经网络。对于卷积神经网络和循环神经网络，通常一开始就设置一个相对较大的学习率，然后随着优化过程逐渐降低(He et al.， 2016;2017;Sutskever等人，2014;Gehring等，2017;He et al.， 2019)。学习率预热阶段只被证明在处理一些非常具体的问题时至关重要，例如大批量训练。Goyal et al. (2017);He et al. (2019);You et al.(2018)表明，在训练具有极大批量大小的神经网络时，学习率预热阶段是首选。

然而，在大多数场景下优化Transformer模型时，学习率预热阶段是必不可少和关键的(Vaswani et al.， 2017; Devlin等人，2018; 戴等，2019;Radford等，2019;Lu et al.， 2019)。Popel & Bojar(2018)研究了不同的预热策略对后ln变压器模型优化的影响，发现没有预热迭代或预热迭代相对较少时，优化会出现分歧。Pre-LN变压器已在最近的几部作品中提出(Baevski & Auli, 2018;Child等人，2019;Wang et al.， 2019)，以缓解训练更深模型时的一些优化问题，但麻烦的热身阶段仍然存在于他们的训练管道中。

(Liu et al.， 2019a)声称，热身阶段的好处来自于减少亚当优化器中自适应学习率的方差(Kingma & Ba, 2014)。他们提出通过一种叫做RAdam的亚当的新变体来纠正自适应学习率的差异。然而，我们发现不仅对亚当，学习率预热阶段对其他优化器也有很大帮助。这可能表明亚当不是热身阶段的必要前提。在一项并行和独立的工作中，Nguyen & Salazar(2019)也通过经验观察到，Pre-LN Transformer可以在没有学习率预热阶段的情况下进行训练。我们的工作提供了一项更全面的研究，用理论分析对其进行了重新评估。

3. 变压器的优化

3.1. 后层归一化的Transformer

Transformer架构通常由堆叠的Transformer层组成(Vaswani等人，2017;Devlin et al.， 2018)，每一层都以一组向量序列作为输入，并输出具有相同形状的新向量序列。一个Transformer层有两个子层:(多头)自注意力子层和位置级前馈网络子层。残差连接(He et al.， 2016)和层归一化(Lei Ba et al.， 2016)分别应用于两个子层。我们首先介绍Transformer层的每个组成部分，然后呈现整个架构。

注意函数可以表示为使用键值对查询条目(Vaswani等人，2017)。self-attention子层使用缩放的点积注意力，定义为:attention (Q, K, V) = softmax($QK^T/\sqrt{}$)V，其中d是di-d

隐藏表示的维度，以及Q (Query)， K (Key)， V (Value)被指定为前一层的隐藏表示。self-attention子层的多头变体被广泛使用，它允许模型联合关注来自不同表示子空间的信息，定义为

$$\text{Multi-head}(Q, K, V) = \text{Concat}(\text{head}_1, \cdots, \text{head}_H)W^O$$
$$\text{head}_k = \text{Attention}(QW_k^Q, KW_k^K, VW_k^V),$$

其中WkQ∈$_{d\times d_K}$, WkK∈$_{d\times d_K}$ R R, WkV∈Rd×d$_v$, W $_O$∈ $_{HdV\times d}$ R为project parameter矩阵，H为正面数。d_K和d_V是键和值的维数。没有任何混淆，给定一个向量序列(x_1，…，x_n)，我们使用MultiHeadAtt(x_i，[x_1，x_2，…，10 * 10]作为位置i上的多头自注意力机制，它考虑从x_i到整个序列的注意力，即，实际上，[x_1，x_2，…，x_n])=多头(xi，[x1，…，xn]，[x1，…xn))。

位置FFN子层除自关注子层外，每个Transformer层还包含一个完全连接的网络，该网络分别相同地应用于每个位置。这个子层是一个具有ReLU激活函数的两层前馈网络。给定一个向量序列h_1，…，h_n，任意h_i上位置FFN子层的计算定义为:

$$\text{FFN}(h_i) = \text{ReLU}(h_iW^1 + b^1)W^2 + b^2,$$

其中W $_1$, W $_2$, b1和b2是参数。

残差连接和层归一化除了上面描述的两个子层，残差连接和层归一化也是Transformer的关键组件。对于任何向量v，层归一化被计算为LayerNorm(v) = γv−σ μ + β，其中μ、σ为v中各元素的均值和标准差，即μ= $_{d1}$ Pdk= 1vk， σ 2 = $_{d1}$ Pdk=1(vk−μ)$_2$。尺度γ和偏置向量β是参数。

Transformer层中子层、残差连接和层归一化的不同顺序导致了Transformer架构的变体。Transformer和BERT的原始和最常用架构之一(Vaswani等人，2017;Devlin等人，2018)遵循“自关注(FFN)子层→剩余连接→层归一化”，我们称之为带后层归一化的变压器(后ln变压器)，如图1所示。

后ln变压器设$x_{l,i}$为位置i的第1层变压器的输入，其中$x_{l,i}$为维数d的实值向量，i = 1,2，…， n, l = 1,2，…， Ln是序列的长度，L是层数。为了完整，我们将$x_{0,i}$定义为位置i的输入嵌入，它通常是单词嵌入和位置嵌入的组合。第l层内部的计算由几个步骤组成，我们在x上使用上标来表示不同步骤的输入(输出)，如表1(左)所示，其中$_{w1,\ 1,\ w2,}$ 1, b1, 1和b2, 1是第l层FFN子层的参数。

3.2. 学习率预热阶段

我们对后ln变压器优化中的*学习率预热*阶段感兴趣。不同于许多其他架构的优化，学习率从一个相对较大的值开始，然后衰减(Bahdanau et al.， 2017;Dauphin等人，2017)，后ln变压器的学习率预热阶段似乎至关重要(Popel & Bojar, 2018)。我们将第t次迭代的学习率表示为lr(t)，训练期间的最大学习率表示为lrmax。给定预定义的预热时间框架，第一次T_{warmup}迭代(Vaswani等人，2018)的学习率调度器定义为

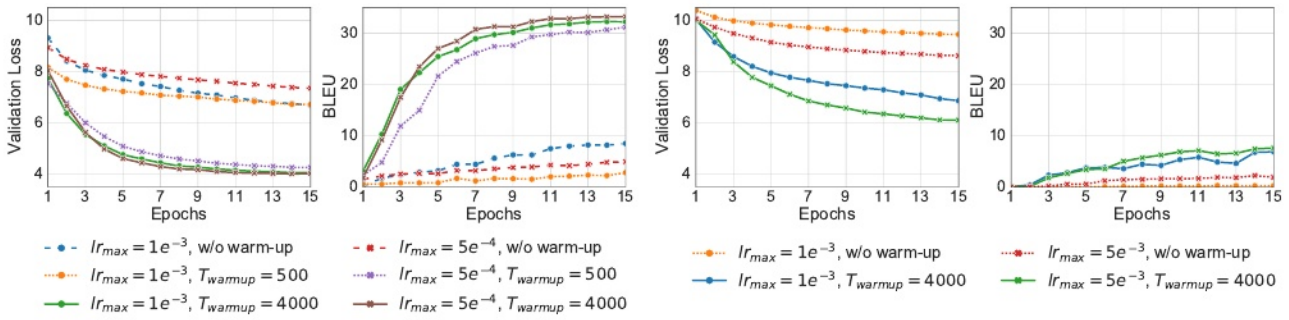
$$\text{lr}(t) = \frac{t}{T_{\text{warmup}}}\text{lr}_{max}, t \leq T_{\text{warmup}}. \quad (1)$$

在这个预热阶段之后，学习率将由经典的学习率调度器设定，如线性衰减、逆平方根衰减，或在特定迭代时强制衰减。我们进行的实验表明，这个学习率热身阶段是必不可少的训练后ln变压器模型。

实验设置我们在IWSLT14德英(De-En)机器翻译上进行了实验

表1。后ln变压器与前ln变压器

Post-LN Transformer	Pre-LN Transformer
$x_{l,i}^{post,1} = \text{MultiHeadAtt}(x_{l,i}^{post}, [x_{l,1}^{post}, \cdots, x_{l,n}^{post}])$	$x_{l,i}^{pre,1} = \text{LayerNorm}(x_{l,i}^{pre})$
$x_{l,i}^{post,2} = x_{l,i}^{post} + x_{l,i}^{post,1}$	$x_{l,i}^{pre,2} = \text{MultiHeadAtt}(x_{l,i}^{pre,1}, [x_{l,1}^{pre,1}, \cdots, x_{l,n}^{pre,1}])$
$x_{l,i}^{post,3} = \text{LayerNorm}(x_{l,i}^{post,2})$	$x_{l,i}^{pre,3} = x_{l,i}^{pre} + x_{l,i}^{pre,2}$
$x_{l,i}^{post,4} = \text{ReLU}(x_{l,i}^{post,3} W^{1,l} + b^{1,l}) W^{2,l} + b^{2,l}$	$x_{l,i}^{pre,4} = \text{LayerNorm}(x_{l,i}^{pre,3})$
$x_{l,i}^{post,5} = x_{l,i}^{post,3} + x_{l,i}^{post,4}$	$x_{l,i}^{pre,5} = \text{ReLU}(x_{l,i}^{pre,4} W^{1,l} + b^{1,l}) W^{2,l} + b^{2,l}$
$x_{l+1,i}^{post} = \text{LayerNorm}(x_{l,i}^{post,5})$	$x_{l+1,i}^{pre} = x_{l,i}^{pre,5} + x_{l,i}^{pre,3}$
	Final LayerNorm: $x_{Final,i}^{pre} \leftarrow \text{LayerNorm}(x_{L+1,i}^{pre})$



(a) Loss/BLEU on the IWSLT14 De-En task (Adam)　　(b) Loss/BLEU on the IWSLT14 De-En task (SGD)

图2。Adam和SGD优化的模型在IWSLT14 De-En任务上的性能。

的任务。我们主要考察两个方面:学习率预热阶段是否必要，最终模型性能是否对值敏感T_{warmup}。为了研究第一个方面，我们分别使用Adam优化器(Kingma & Ba, 2014)和vanilla SGD优化器(Ruder, 2016)训练模型。对于这两个优化器，我们检查是否可以删除热身阶段。我们遵循Vaswani等人(2017)在Adam中设置超参数β为(0.9,0.98)。我们还测试了两种优化器的不同lr_{max}。对于Adam，我们设置lrmax = 5e - 4或1e - 3，对于SGD，我们设置lrmax = 5e - 3或1e - 3。当使用热身阶段时，我们按照原始论文(Vaswani等人，2017)的建议将warm - up设置为4000。为了研究第二个方面，我们将T_{warmup}设置为1/500/4000(“1”指的是没有预热设置)，并使用lrmax = 5e−4或1e−3与Adam。对于所有的实验，热身阶段后都使用相同的逆平方根学习率调度器。我们使用验证损失和BLEU (Papineni et al.， 2002)作为模型性能的评估指标。

我们记录了训练过程中每个epoch的模型检查点，并计算了验证损失和BLEU分数。模型的性能绘制在图2(a)和图2(b)中。x轴是epoch数，y轴是BLEU分数/验证损失。“w /o warm-up”表示“没有预热阶段”，而“w / warm-up”表示“有预热阶段”。

首先，我们可以看到，对于两种优化器来说，学习率预热阶段是必不可少的。如果没有热身阶段，使用Adam优化器训练的模型的BLEU分数只能达到8.45。相比之下，使用热身阶段训练的模型在BLEU得分方面可以达到34分左右。在验证损失曲线上也可以观察到相同的趋势。虽然使用SGD训练的模型的性能明显不如Adam，但我们仍然可以看到与Adam相似的现象。在不使用热身阶段的情况下，BLEU得分在15个回合中略高于零。

其次，我们可以看到优化过程对warm - up的值很敏感，这意味着warm - up是训练后ln变压器的一个重要超参数。例如，设置warm - up = 500时，当lrmax = 5e−4和1e−3时，Adam学习模型的BLEU得分分别仅为31.16和2.77。

这样的热身阶段有几个缺点。首先，它的配置会显著影响最终的性能。从业者需要仔细的超参数调整，这对于大规模NLP任务来说计算成本很高。其次，预热阶段可能会减慢优化的速度。标准优化算法通常会启动

以较大的学习率来快速收敛。但是，在使用预热阶段时，学习率要从零逐渐增加，这可能会使训练效率低下。Liu et al. (2019a)认为，在模型训练的早期阶段，热身阶段可以减少Adam的不受欢迎的显著方差。然而，根据我们的结果，热身阶段也有助于SGD的训练。这表明，热身阶段的好处可能并不适合某个特定的优化器。

3.3. 在初始化阶段理解Transformer

我们可以看到，后ln变压器不能以一个大的学习率从头开始训练。这促使我们研究在模型初始化时发生了什么。我们首先为我们的理论分析介绍参数初始化设置，然后介绍我们的理论发现。

我们表示L(·)为一个位置的损失函数，L(≈·)为整个序列的损失函数，k·k 2和k·k F为L_2范数(谱范数)和Frobenius范数，LN(x)为尺度γ= 1，偏置β= 0的标准层归一化，JLN (x) = ∂ LN(x)。
∂x作为Ja-
LN(x)的cobian矩阵。令O(·)表示抑制乘性常数的标准大O表示法。

参数初始化每个Transformer层中的参数矩阵通常由Xavier初始化(Glorot & Bengio, 2010)初始化。给定一个大小为$n_{in} \times n_{out}$的矩阵，Xavier初始化通过从高斯分布N (0, N +n2)中独立采样来设置每个元素的值。偏置向量通常会被初始化
——
为零向量。归一化层中的尺度γ设为1。

为了进行理论分析，我们研究了一个更简单的设置。首先，我们专注于单头关注而不是多头变体，对于所有图层，我们将W Q, 1, W K, 1, W V, 1, W 1, W 2, 1的形状设置为d × d。其次，我们初始化自关注子层中的参数矩阵W Q,l和W K,l为零矩阵。在这个设置中，注意力在初始化时是均匀分布，MultiHeadAtt(x1l,i， [x1l,1, x1l,2，…，x1l,n])可以简化为n1 Pnj=1 xl,j WV,l。第三，我们假设输入向量也从相同的高斯分布中采样。这是合理的，因为输入是词嵌入和可学习位置嵌入的线性组合，两者都是通过高斯分布初始化的。

后ln变压器与前ln变压器我们将后ln变压器与变压器架构的另一种变体，即具有前层规范化的变压器(前ln)进行比较。前ln变压器已在多个系统中实现(Vaswani等人，2018; Klein等人，2018;Liu等人，2019b)。Wang等人(2019)认为，当层数增加时，前ln变压器的性能优于后ln变压器。与将层归一化置于残差块之间的后ln变压器不同，前ln变压器将层归一化置于残差连接中，并将其置于所有其他非线性变换之前。此外，前ln变压器在预测之前使用*最后一层规范化*。我们在表1和图1中提供了后ln /前ln变压器的数学公式和可视化。

对于这两种架构，每个$x_{L,i}$通过一个soft-max层来产生字典V上的分布。损失函数在softmax分布上定义。例如，在序列预测中，损失函数定义为L(xpostL+1,i) =−log (softmaxyi (W emb post xL+1,i))
为后ln变压器，L(xF初值, i) =
−前ln变压器的log(softmaxyi (wemb xpreF final,i))，其中softmax$_{y_i}$为softmax分布输出的地面真值令符y_i的概率，wemb为词嵌入矩阵。整个序列的损失是每个位置上损失的平均值。在不损失一般性的情况下，我们假设所有的衍生品都是有界的。我们引入了以下随机变量的浓度性质，这将在定理中进一步使用。

定义1。*如果随机变量*Z≥0*的概率至少*为1−δ， z−EZ≤，则称为(, δ)有界,*其中*
EZ
——
> 0和0 < δ< 1。
直观地说，如果随机变量Z是(， δ)-有界的，那么它的实现以大概率不会离它的期望太远。例如，如果Y是d维标准高斯随机向量，则Z = kY k22是(， δ)有界，δ= exp(-d 2/8)，0 << 1(详细信息请参阅补充材料)。由于自关注子层和FFN子层中的参数矩阵都是用高斯分布初始化的，如果Transformer中隐藏状态的范数满足上述集中条件，我们有以下定理来表征梯度的尺度。

定理1 (Transformer*最后一层的梯度*)。*假设*kxpost,5k22*和*kxpre 2 L,i L+1,ik2*对所有*i都(, δ)有界,*其中和*δ= δ()*是小数字。则对于*L*层的后ln变压器,其最后一层参数的梯度满足,其概率至少为*0.99−δ−0.9+

$$\|\frac{\partial \tilde{\mathcal{L}}}{\partial W^{2,L}}\|_F \leq \mathcal{O}(d\sqrt{\ln d})$$

*对于*L*层前ln变压器,*

$$\|\frac{\partial \tilde{\mathcal{L}}}{\partial W^{2,L}}\|_F \leq \mathcal{O}\left(d\sqrt{\frac{\ln d}{L}}\right).$$

由定理1可知，对于后ln变压器，到最后一层FFN的梯度的尺度√为O(d ln d)阶，与l无关。对于前ln变压器，梯度的尺度要小得多。我们首先研究了后ln变压器和前ln变压器的前向传播。引理1将作为证明主定理和其他引理的基本工具。

引理1。*如果*$X \in R^d$*是高斯向量*,$X \sim N(0,\sigma^2 I_d)$*),则* E (kReLU(X)k22) = $\frac{1}{2}\sigma^2 d$。

基于引理1，我们有以下引理来估计后ln变压器和前ln变压器不同层的隐藏状态的规模。

引理2。*初始化时*,*对于后ln变压器*,*对于所有*l > 0和i, E (kxpost,5l,i k22) = 32 d。对于前ln变压器，对于所有l > 0*和*i, (1 + 2l)d≤E(k*xprel,i k22)*≤*(1 + 3l2)d。*期望接管了输入和初始化的随*机性。

引理2研究了后ln /前ln变压器中隐藏状态的期望范数。很明显，在后ln变压器中，xpost1,i的范数是d和

因此，我们转而研究xpost的范数，5l,i。如我们所见由引理2可知，后ln变压器中隐藏状态的尺度在期望上保持不变，而前ln变压器中隐藏状态的尺度随深度线性增长。下一个引理表明，隐藏状态的尺度与使用层归一化的架构中的梯度尺度高度相关。

引理3。*对于*x∈Rd,*我们有* kJLN (x)k2= O(*kxkd2)**其中*$J_{LN}(x) = \partial LN \partial x(x)$。

引理1、引理2、引理3和定理1的证明可以在补充材料中找到。主要思想是层归一化将梯度归一化。在后ln变压器中，层归一化的输入尺度与L无关，因此最后一层参数的梯度与L无关。而在前ln变压器中，最后一层归一化的输入尺度在L中是线性的，因此所有参数的梯度都将被L归一化。

将理论扩展到其他层/参数我们已经提供了如上最后一层FFN子层梯度的形式化证明。为了充分理解优化，我们还对其他层和其他参数做了一些初步的分析。我们的主要结果是，后ln变压器中的梯度范数对于输出附近的参数很大，并且可能随着层指数l的减小而衰减。相反，Pre-Transformer中的梯度范数很可能对任何l层保持不变。所有初步的理论结果都在补充材料中提供。

3.4. 理论的实证验证和讨论

由于我们的理论是在对问题的几次简化的基础上推导出来的，所以我们通过实验来研究我们的理论见解是否与我们在真实场景中观察到的一致。通用模型和训练配置完全遵循3.2节。使用不同的随机种子重复实验10次。

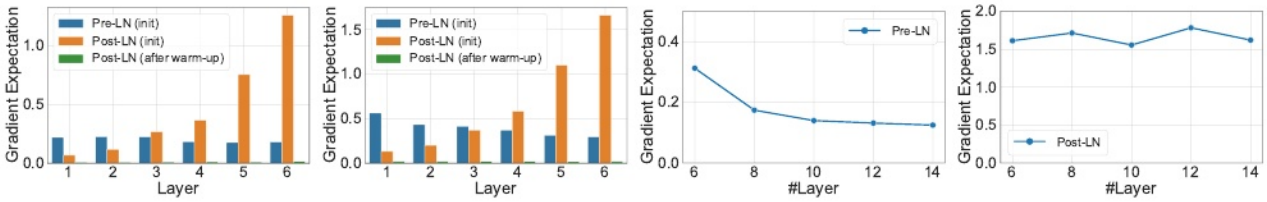
给定一个初始化的模型，我们跨批次记录后ln /前ln变压器中的隐藏状态，发现隐藏状态的范数满足属性((0.1,0.125)-bounded)。

关于定理1，定理1表明，对于任何尺寸的后ln变压器，最后一FFN子层的梯度范数的尺度保持不变。相反，前ln变压器随模型尺寸的增大而减小。我们在初始化时计算并记录6-6/8-8 -10 -10 -12 -12/14-14后ln /前ln变压器模型中最后一个FFN子层的梯度范数。结果被绘制在图3(c)和3(d)中。x轴为模型的大小，y轴为最终FFN子层^{w2}的梯度范数的值。由图可知，当层数增加时，梯度范数在后ln变压器中保持不变(约1.6)，在前ln变压器中减小。这个观察结果与我们的理论是一致的。

在推广理论的基础上，计算了6-6后ln /前ln变压器中各参数矩阵的梯度范数。我们为不同的mini-batches记录每个参数的梯度。对于参数矩阵中的元素，我们计算它们的期望梯度，并使用这些值的Frobenius范数作为矩阵的期望梯度的尺度。图3(a)和3(b)显示了FFN子层的统计数据。x轴索引不同的Transformer层。从图中可以看出，对于后ln变压器，期望梯度的尺度随着层指数的增大而增大。相反，在前ln变压器中，不同层的比例几乎保持相同。这些观察与我们的理论发现是一致的。

后ln变压器的关键预热阶段根据上述分析，我们假设梯度尺度是后ln变压器需要仔细学习速率调度的原因之一。由于某些层的梯度很大，在没有热身的情况下使用较大的学习率可能会使训练不稳定。

为了验证这一论点，首先，我们与亚当一起研究了热身阶段后后ln变压器的梯度统计。从图3(a)和图3(b)可以看出，尺度



(a) W^1 in the FFN sub-layers (b) W^2 in the FFN sub-layers (c) Pre-LN Transformer (d) Post-LN Transformer

图3。1的梯度的范数。6-6 Transformer中的不同层数(a,b)。2. ^{w2,L}在不同尺寸的变压器(c,d)。

的梯度非常小，模型可以用大的学习率进行训练。其次，我们进行了一个实验，使用固定的小学习率(即1e−4)从头开始训练后ln变压器，以验证使用小步更新是否可以缓解问题。细节在补充材料中提供。一般来说，使用非常小且固定的学习率可以缓解问题并在一定程度上优化后ln变压器，但收敛速度明显较慢。上面的两个实验都支持我们的说法。

4. 实验

在上一节中，我们发现Pre-LN变压器初始化时的梯度表现良好。鉴于这一观察结果，我们推断，在训练Pre-LN变压器时，可以安全地取消学习率预热阶段。在本节中，我们在NLP中的两个主要任务上实证验证了它，即机器翻译和无监督预训练。

4.1. 实验设置

我们在两个广泛使用的任务上进行了实验:IWSLT14德语到英语(De-En)任务和WMT14英语到德语(En-De)任务。对于IWSLT14 De-En任务，我们使用与第3节中相同的模型配置。对于WMT14 En-De任务，我们使用Transformer基本设置。更多细节可以在补充材料中找到。

为了训练Pre-LN变压器，我们去掉了学习率预热阶段。在IWSLT14 De-En任务中，我们将初始学习率设置为5e^{-4}，并在第8个epoch将学习率衰减0.1。在WMT14 En-De任务上，我们运行了两个实验，其中初始学习率分别设置为7e^{-4}/1.5e^{-3}。两个学习率都在第6个epoch衰减，随后是逆平方根学习率调度器。

我们使用学习率热身阶段作为基线来训练后ln变压器。在IWSLT14 De-En任务和WMT14 En-De任务中，我们按照Vaswani等人(2017)的方法，将预热阶段的数量设置为4000，然后使用平方根反比学习率调度程序。对于上述所有实验，我们使用Adam优化器，并将超参数β设置为(0.9,0.98)。我们设置lr_{max}与每次相应实验中Pre-LN变压器的初始学习率相同。由于Liu等人(2019a)建议使用RAdam可以去除学习率预热阶段，因此我们在IWSLT14 De-En任务上尝试了该优化器。我们使用Liu等人(2019a)提出的线性学习率衰减，并保持所有其他超参数与其他实验相同。

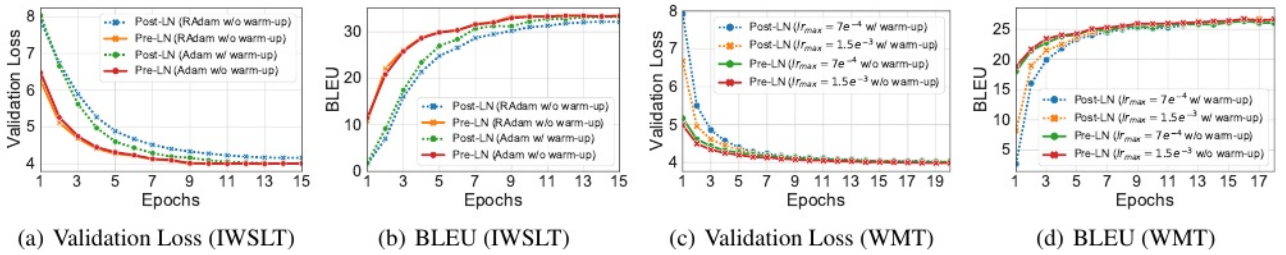
无监督预训练(BERT)我们遵循(Devlin等人，2018)使用英语维基百科语料库和图书语料库进行预训练。由于数据集BookCorpus (Zhu et al.， 2015)不再免费分发。我们遵循(Devlin等人，2018)的建议，自己抓取和收集BookCorpus。两个数据集的连接总共包含大约3.4B个单词，这与(Devlin等人，2018)中使用的数据语料库相当。我们将文档随机分成一个训练集和一个验证集。预训练的训练-验证比是199:1。

我们在实验中使用基本模型配置。与翻译任务类似，我们在没有预热阶段的情况下训练Pre-LN BERT，并将其与后ln BERT进行比较。我们在Devlin等人(2018)中遵循相同的超参数配置，使用lr_{max}=1e^{-4}的10k热身步骤训练后ln BERT。对于Pre-LN BERT，我们使用从3e−4开始的线性学习率衰减，没有预热阶段。我们尝试使用更大的学习率(如3e−4)来进行后ln BERT，但发现优化是发散的。

4.2. 实验结果

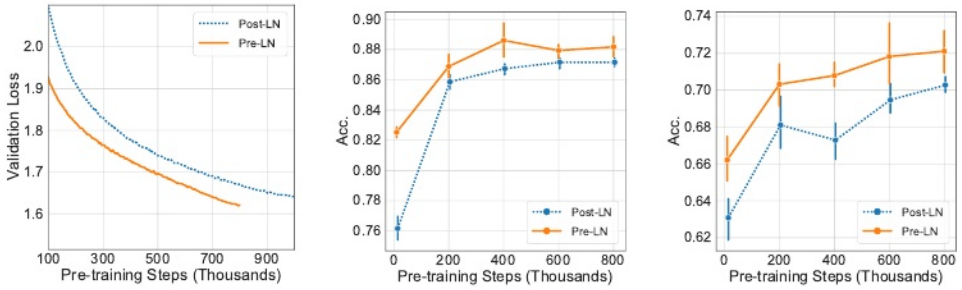
我们在训练过程中记录每个epoch的模型检查点，并计算验证损失和BLEU分数。模型在不同检查点的表现如图4(a) - 4(d)所示。

首先，从图中可以看出，对于Pre-LN Transformer的训练来说，学习率预热阶段不再是至关重要的，学习模型的性能是有竞争力的。例如，在IWSLT14 De-En任务上



(a) Validation Loss (IWSLT) (b) BLEU (IWSLT) (c) Validation Loss (WMT) (d) BLEU (WMT)

图4。模型在IWSLT14 De-En任务和WMT14 En-De任务上的性能



(a) Validation Loss on BERT (b) Accuracy on MRPC (c) Accuracy on RTE

图5。模型在无监督预训练(BERT)和下游任务上的性能

Pre-LN变压器的BLEU分数和验证损耗可达到34分和4分左右，与后ln变压器的性能相当。

其次，Pre-LN变压器收敛速度比后ln变压器使用相同的lr_{max}。在IWSLT14 De-En任务中，Pre-LN变压器的第9个检查点与后ln变压器的第15个检查点取得了几乎相同的性能(验证损失/BLEU分数)。在WMT14 En-De任务中也可以找到类似的观察结果。

第三，与RAdam相比，我们发现层归一化位置的变化“支配”了优化器的变化。根据我们在IWSLT14 De-En任务上的实验，我们可以看到，虽然RAdam在没有热身阶段的情况下训练后ln变压器很好，但在训练Pre-LN变压器时与Adam相差不大。

无监督预训练(BERT)我们记录模型检查点的验证损失，并在图5(a)中绘制它们。与机器翻译任务类似，Pre-LN模型可以去除学习率预热阶段。Pre-LN模型的训练速度更快。例如，Post-LN模型在更新50万次时获得1.69的验证损失，而Pre-LN模型在更新70万次时获得类似的验证损失，这表明有40%的加速率。请注意，热身(10k)远小于加速度(200k)，这表明Pre-LN变压器更容易使用较大的学习率进行优化。我们还评估了下游任务MRPC和RTE上的不同模型检查点(更多细节可以在补充材料中找到)。实验结果如图5(b)和5(c)所示。我们可以看到，Pre-LN模型在下游任务上也收敛得更快。

综上所述，所有不同任务的实验表明，Pre-LN Transformer的训练不依赖于学习速率预热阶段，并且可以比Post-LN Transformer更快地训练。

5. 结论和未来的工作

在本文中，我们研究了为什么学习率预热阶段在训练Transformer中很重要，并表明层归一化的位置很重要。我们表明，在位于残差块之外的层归一化的原始Transformer中，输出层附近参数的预期梯度在初始化时很大。这导致在使用大的学习率时训练不稳定。进一步表明，在残差块内部定位层归一化的Transformer，可以在没有预热阶段的情况下进行训练，并且收敛得更快。未来，我们将研究定位层归一化的其他策略，并从理论角度理解Transformer的优化。

参考文献

Al-Rfou, R., Choe, D., Constant, N., Guo, M., and Jones, L. Character-level language modeling with deeper self-attention. *arXiv preprint arXiv:1808.04444*, 2018.

Baevski, A. and Auli, M. Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853*, 2018.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. 2017.

Bentivogli, L., Dagan, I., Dang, H. T., Giampiccolo, D., and Magnini, B. The fifth PASCAL recognizing textual entailment challenge. 2009.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Dai, Z., Yang, Z., Yang, Y., Cohen, W. W., Carbonell, J., Le, Q. V., and Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. Lan-guage modeling with gated convolutional networks. In *International Conference on Machine Learning*, pp. 933– 941, 2017.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., and Kaiser, ☒. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for lan-guage understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dolan, W. B. and Brockett, C. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing.*, 2005.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121– 2159, 2011.

Edunov, S., Ott, M., Auli, M., and Grangier, D. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*, 2018.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pp. 1243– 1252, 2017.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Pro-ceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learn-ing for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., and Li, M. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 558– 567, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Klein, G., Kim, Y., Deng, Y., Nguyen, V., Senellart, J., and Rush, A. Opennmt: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Associa-tion for Machine Translation in the Americas (Volume 1: Research Papers)*, volume 1, pp. 177–184, 2018.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Fed-erico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pp. 177–180, 2007.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Penning-ton, J., and Sohl-Dickstein, J. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

Lei Ba, J., Kiros, J. R., and Hinton, G. E. Layer normaliza-tion. *arXiv preprint arXiv:1607.06450*, 2016.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019a.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

Lu, Y., Li, Z., He, D., Sun, Z., Dong, B., Qin, T., Wang, L., and Liu, T.-Y. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019.

Nguyen, T. Q. and Salazar, J. Transformers without tears: Improving the normalization of self-attention. *arXiv preprint arXiv:1910.05895*, 2019.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Popel, M. and Bojar, O. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70, 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Ruder, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *ACL*, 2016.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to se-quence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Com-puter Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 2012.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, ☒., and Polosukhin, I. Atten-tion is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A. N., Gouws, S., Jones, L., Kaiser, L., Kalchbrenner, N., Parmar, N., Sepassi, R., Shazeer, N., and Uszkoreit, J. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416, 2018. URL http://arxiv.org/abs/1803.07416.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D. F., and Chao, L. S. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S., and Pennington, J. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convo-lutional neural networks. In *International Conference on Machine Learning*, pp. 5389–5398, 2018.

Yang, G. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient in-dependence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.

Yang, G., Pennington, J., Rao, V., Sohl-Dickstein, J., and Schoenholz, S. S. A mean field theory of batch normal-ization. *arXiv preprint arXiv:1902.08129*, 2019a.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019b.

You, Y., Zhang, Z., Hsieh, C.-J., Demmel, J., and Keutzer, K. Imagenet training in minutes. In *Proceedings of the 47th International Conference on Parallel Processing*, pp. 1. ACM, 2018.

Zeiler, M. D. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Zhang, H., Dauphin, Y. N., and Ma, T. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*, 2019.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *arXiv preprint arXiv:1506.06724*, 2015.

A. 实验设置

A.1. 机器翻译

IWSLT14德语-英语(De-En)任务的训练集/验证集/测试集分别包含约153K/7K/7K个句子对。我们使用基于联合源和目标字节对编码(BPE)的10K个标记的词汇表(Sennrich等人，2015)。我们所有的实验都使用了具有6层编码器和6层解码器的Transformer架构。嵌入的大小设置为512，注意力子层和位置前馈网络子层中的隐藏节点大小设置为512和1024，头的数量设置为4。通过设置= 0.1 (Szegedy et al.， 2016)，使用标签平滑交叉熵作为目标函数，我们应用比例为0.1的dropout。批量大小设置为4096个token。当我们在推理过程中解码来自模型的翻译结果时，我们将波束大小设置为5，长度惩罚设置为1.2。

IWLST14 De-En任务的配置与第3节-[1]相同。对于WMT14 En-De任务，我们复制了(Vaswani et al.， 2017)的设置，该设置由大约450万个训练平行句子对组成，并使用基于联合源和目标BPE的37K词汇表。使用Newstest2013作为验证集，使用Newstest2014作为测试集。Transformer架构的基本配置之一是基础设置(base setting)，由6层编码器和6层解码器组成。隐藏节点和嵌入的大小被设置为512。正面的数量为8。通过设置= 0.1，使用标签平滑交叉熵作为目标函数。在16块NVIDIA Tesla P40 GPU上，批处理大小设置为每个GPU 8192个令牌。

A.2. 无监督Pretraining

我们遵循Devlin等人(2018)使用英语维基百科语料库和BookCorpus进行预训练。由于数据集BookCorpus (Zhu et al.， 2015)不再免费分发。我们遵循Devlin等人(2018)的建议，自己抓取和收集BookCorpus[2]。两个数据集的拼接总共包含大约3.4B个单词，这与Devlin等人(2018)使用的数据语料库相当。我们首先用Spacy[3];将文档分割成句子，然后，我们对文本进行规范化、小写和标记化

使用Moses (Koehn等人，2007)并应用BPE(Sennrich等人，2016)。我们将文档随机分成一个训练集和一个验证集。预训练的训练-验证比是199:1。所有实验均在32块NVIDIA Tesla P40 GPU上进行。

Devlin等人(2018)的基本模型由12个变压器层组成。隐藏节点和嵌入的大小设置为768，头的数量设置为12。

A.3. GLUE数据集

MRPC微软研究复述语料库(Dolan & Brockett, 2005)是一个自动从在线新闻源中提取的句子对语料库，人工标注了句子对中的句子是否在语义上等价，任务是预测等价性。其性能由准确率来评估。

RTE识别(recognition Textual蕴涵，RTE)数据集来自一系列年度文本蕴涵挑战(Bentivogli et al.， 2009)。任务是预测句子对中的句子是否为蕴涵。性能是通过准确性来评估的。

GLUE任务的微调我们使用验证集进行评估。为了微调模型，遵循Devlin等人(2018);Liu et al. (2019b)，我们在包括不同批量大小(16/32)、学习率($1e^{-5}$-$1e^{-4}$)和epoch数量(3-8)的搜索空间中搜索优化超参数。我们发现验证精度对随机种子敏感，因此我们使用不同的随机种子对每个任务重复微调6次，并计算验证精度的95%置信区间。

B. 引理1的证明

*证明。*表示X = (X1, X2，…表示$\rho_X(x)$为x的概率密度函数1。

$$\mathrm{d}\ R_{+\infty 2}\ x\ 2\rho_x\ (x)dx = 2\ 1\sigma 2\mathrm{d}。2\ 0$$

C. 引理2的证明

*证明。*初始化时，层归一化计算为$LN(v) = v-\sigma\mu。$很容易看出，初始化时的层归一化将任何向量v投射到d−1球面√上

$v-\mu 2_{dk=1}\ (vk-\mu)_2$半径d，因为$kLN(v)k22= k\ \sigma k2= \sigma 2$

$= d。$

我们首先估计每个中间体的期望l_2范数

[1]The Pre-LN Transformer can get state-of-the-art performance (35.5 test BLEU) on the IWSLT14 DE-EN task by setting initial learning rate to be 7. $5e^{-4}$ and decaying it at the 8000 update steps followed by the inverse square root learning rate scheduler. The dropout is set to be 0.3, attention dropout is set to be 0.1. The batch size is set to be 8192.

[2]https://www.smashwords.com

[3]https://spacy.io

输出xpost,1l,i，…，xpost,5l,i为l > 0。使用Xavier首字母-化，W V,l中的元素为i.i.d。从N(0,1 /d)中采样的高斯随机变量。由于kx post l,i k22=d由层归一化的定义当l > 0时，我们有

2 l E (kxpo st,我 k22) = E (kxpos k22) + E(kx post,1l,i k22) tl,我 + 2 e (xpost 1 xpost l,我 > l,我) (2)

= E (kxpost k22) + E(kx post,1l,i k22) l,我 + 2 n E (X n j = 1 xpost l l j xpostl W V,我 >) (3)

$$=\mathbb{E}(\|x_{l,i}^{post}\|_2^2)+\mathbb{E}(\|x_{l,i}^{post,1}\|_2^2) \quad (4)$$

$$=\mathbb{E}(\|x_{l,i}^{post}\|_2^2)+\mathbb{E}(\|\frac{1}{n}\sum_{i=1}^{n}x_{l,i}^{post}\|_2^2) \quad (5)$$

$$\leq 2d \quad (6)$$

和 E(kxpost,2l,ik22) = E(kxpostl,ik22) + E(kxpost,1l,i k22) = E (kxl,i k2) + E(k n post 2 1 Pni=1 xl,ipost k22)⩾E(kxl,i k2) = d post 2 1

类似地，根据层归一化的定义，我们有kxpost,3l,ik22= d。同样，对于ReLU激活函数，W 1,l和W 2,l中的元素是i.i.d。从N(0, 1 /d)采样的高斯随机变量。根据引理1，我们有

E (kxpost 4 l,我 kReLU(xpost, 1, 1, 1) =E W 1,l)W 2,lk22) (xpost, 1, 1) (7)

=E(E(E(kReLU(xpost, 1, W 1,l)W 2,lk22 1)) | xpost , W 1,l)|xpost,3l,i)) (8)

3 l 我

$$=\mathbb{E}(\mathbb{E}(\|\mathbf{ReLU}(x_{l,i}^{post,3}W^{1,l})\|_2^2|x_{l,i}^{post,3})) \quad (9)$$

$$=\mathbb{E}(\frac{1}{2}\|x_{l,i}^{post,3}\|_2^2)=\frac{d}{2} \quad (10)$$

基于此，我们可以估计E(kxpost,5l,i k22)的尺度如下。

$$\mathbb{E}(\|x_{l,i}^{post,5}\|_2^2) =\mathbb{E}(\|x_{l,i}^{post,3}\|_2^2)+\mathbb{E}(\|x_{l,i}^{post,4}\|_2^2) + 2\mathbb{E}(x_{l,i}^{post,3}x_{l,i}^{post,4\top}) \quad (11)$$

$$=\mathbb{E}(\|x_{l,i}^{post,3}\|_2^2)+\mathbb{E}(\|x_{l,i}^{post,4}\|_2^2) + \frac{2}{n}\mathbb{E}(\sum_{j=1}^{n}\mathbf{ReLU}(x_{l,j}^{post,3}W^{1,l})W^{2,l}x_{l,i}^{post,3\top}) \quad (12)$$

$$=\mathbb{E}(\|x_{l,i}^{post,3}\|_2^2)+\mathbb{E}(\|x_{l,i}^{post,4}\|_2^2)=d+\frac{d}{2}=\frac{3}{2}d \quad (13)$$

使用类似的技术，我们可以为的束缚E(kxprel,i k22)

Pre-LN变压器。

$$\mathbb{E}(\|x_{l,i}^{pre,3}\|_2^2) =\mathbb{E}(\|x_{l,i}^{pre}\|_2^2)+\mathbb{E}(\|x_{l,i}^{pre,2}\|_2^2) + 2\mathbb{E}(x_{l,i}^{pre,2}x_{l,i}^{pre\top}) \quad (14)$$

=E(kxprel,i k2)2+ E(kxl,i 前,2 k22) + $\frac{2}{n}\mathbb{E}(\sum_{j=1}^{n}x_{l,j}^{pre,1}W^{V,l}x_{l,i}^{pre\top})$ (15)

$$=\mathbb{E}(\|x_{l,i}^{pre}\|_2^2)+\mathbb{E}(\|x_{l,i}^{pre,2}\|_2^2) \quad (16)$$

$$=\mathbb{E}(\|x_{l,i}^{pre}\|_2^2)+\mathbb{E}(\|\frac{1}{n}\sum_{i=1}^{n}x_{l,i}^{pre,1}\|_2^2) \quad (17)$$

很容易看出，有E(kxl,i prek22)⩽E(kxl,ipre, 3k22)⩽E(kxpre 2 l,i k2) + d。类似于(10)-(12),

$$\mathbb{E}(\|x_{l+1,i}^{pre}\|_2^2) =\mathbb{E}(\|x_{l,i}^{pre,3}\|_2^2)+\mathbb{E}(\|x_{l,i}^{pre,5}\|_2^2) + 2\mathbb{E}(x_{l,i}^{pre,3}x_{l,i}^{pre,5\top}) \quad (18)$$

$$=\mathbb{E}(\|x_{l,i}^{pre,3}\|_2^2)+\mathbb{E}(\|x_{l,i}^{pre,5}\|_2^2) \quad (19)$$

$$=\mathbb{E}(\|x_{l,i}^{pre,3}\|_2^2)+\frac{1}{2}d \quad (20)$$

结合两者，我们有E(kxprel,i k22) + 12 d⩽E(kxl +1,ipre,k22)⩽E(kxl,i prek22) + 2 3d。然后我们通过归纳得到(1 + 2 l)d⩽E(kx l, i pre k 2 2)⩽(1 + 3l 2)d。

□

D. 引理3的证明

引理3的证明基于引理4.1:

引理4。*设*α∈R^d *是一个向量,使*kαk2= 1,*则*I−α>α的特征值要么为1，要么为*0*。

证明。让{e1，…，ed}为单位向量，使得e1= α和ei⊥ej对于所有(i, j)，则有e1(I−α>α)= e1−

e
fo
th
w

□

−
(1
te

我们显式计算层归一化的雅可比矩阵

作为

$$\frac{\partial \mathrm{LN}(x)_i}{\partial y_j} = \frac{\partial}{\partial y_j}\left(\frac{y_i}{\sqrt{\frac{1}{d}\sum_{k=1}^{n} y_k^2}}\right) \tag{22}$$

$$= \frac{\delta_{ij}\sqrt{\frac{1}{d}\sum_{k=1}^{n} y_k^2} - y_i \frac{\frac{1}{d} y_j}{\sqrt{\frac{1}{d}\sum_{k=1}^{n} y_k^2}}}{\frac{1}{d}\sum_{k=1}^{n} y_k^2} \tag{23}$$

$$= \sqrt{d}\frac{\delta_{ij}\|y\|_2^2 - y_i y_j}{\|y\|_2^{\frac{3}{2}}} = \frac{\sqrt{d}}{\|y\|_2}\left(\delta_{ij} - \frac{y_i y_j}{\|y\|_2^2}\right) \tag{24}$$

其中，当i = j时δ_{ij}= 1，当i 6= j时δ_{ij}= 0，在矩阵形式中，

$$\frac{\partial \mathrm{LN}(x)}{\partial y} = \frac{\sqrt{d}}{\|y\|_2}\left(I - \frac{y^\top y}{\|y\|_2^2}\right) \tag{25}$$

和

$$\mathbf{J}_{LN}(x) = \frac{\partial \mathrm{LN}(x)}{\partial x} \tag{26}$$

$$= \frac{\partial \mathrm{LN}(x)}{\partial y}\frac{\partial y}{\partial x} \tag{27}$$

$$= \sqrt{d}\frac{1}{\|y\|_2}\left(I - \frac{y^\top y}{\|y\|_2^2}\right)\left(I - \frac{1}{d}\mathbf{1}^\top \mathbf{1}\right). \tag{28}$$

由于矩阵的特征值(I−kyky y $\frac{}{>2)和}$(I−d 11> 1)_ $\frac{}{2}$ $\frac{\text{Y}}{\text{Y}>}$

是1或0(根据引理4.1)，我们有k(I−kyk22)k2= O(1)和k(I−d1 1>1)k2= O(1)。所以JLN (x)的谱范数为

$$\|\mathbf{J}_{LN}(x)\|_2 = \mathcal{O}\left(\frac{\sqrt{d}}{\|y\|_2}\right) = \mathcal{O}\left(\frac{\sqrt{d}}{\|x\|_2}\right) \tag{29}$$

□

E. 定理1的证明

定理1的证明基于引理4.2:

引理5。*设*Y*是一个不大于*B*的随机变量,那么对于所有的*a < B,

$$\Pr[Y \le a] \le \frac{\mathbb{E}[B - Y]}{B - a} \tag{30}$$

证明。设X = B−Y，那么X≥0，马尔可夫不等式告诉我们这一点

$$\Pr[X \ge B - a] \le \frac{\mathbb{E}[X]}{B - a} \tag{31}$$

$$\Pr[Y \le a] \le \frac{\mathbb{E}[B - Y]}{B - a} \tag{32}$$

□

定理1的证明。我们通过估计梯度矩阵的每个元素来证明定理1。也就是说，我们将分析$\partial \mathrm{w}\,\partial \mathrm{L2,L} \sim \mathrm{for}$ p, q∈{1，⋯d}。后ln的损失$_{\mathrm{pq}}$

Transformer可以写成

$$\tilde{\mathcal{L}}(x_{L+1,1}^{post}, ..., x_{L+1,n}^{post}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(x_{L+1,i}^{post}) \tag{33}$$

通过反向传播，对于每个i∈{1,2，⋯， n}，L($x_{L+1,i}$)相对于上一层参数^{w2,L}在后ln设置下的梯度可写成:

$$\frac{\partial \mathcal{L}(x_{L+1,i}^{post})}{\partial W_{pq}^{2,L}} = \frac{\partial \mathcal{L}(x_{L+1,i}^{post})}{\partial x_{L+1,i}^{post}}\frac{\partial x_{L+1,i}^{post}}{\partial x_{L,i}^{post,5}}\frac{\partial x_{L,i}^{post,5}}{\partial x_{L,i}^{post,4}}\frac{\partial x_{L,i}^{post,4}}{\partial W_{pq}^{2,L}} \tag{34}$$

$$= \frac{\partial \mathcal{L}(x_{L+1,i}^{post})}{\partial x_{L+1,i}^{post}}\mathbf{J}_{LN}(x_{L,i}^{post,5})\frac{\partial x_{L,i}^{post,4}}{\partial W_{pq}^{2,L}} \tag{35}$$

$$= \frac{\partial \mathcal{L}(x_{L+1,i}^{post})}{\partial x_{L+1,i}^{post}}\mathbf{J}_{LN}(x_{L,i}^{post,5})(0, 0, ..., [\mathrm{ReLU}(x_{L,i}^{post,3}W^{1,L})]_p, ..., 0)^\top \tag{36}$$

这里[ReLU(xpost,3L,iW $_{1,L)]p}$表示ReLU(xpost,3)的第p个元素 $\partial \mathrm{L}(\mathrm{x}_{\mathrm{post}}\ \mathrm{W}_{1,\mathrm{L})}$。所以$_{\mathrm{L+1,i}}$的绝对值$_{\mathrm{)L,i}}$ $\frac{\partial \mathrm{W}_{\mathrm{pq}}}{\mathrm{L}}$ ^{2,}

可以被限制为

$$\left|\frac{\partial \mathcal{L}(x_{L+1,i}^{post})}{\partial W_{pq}^{2,L}}\right| \le \left\|\frac{\partial \mathcal{L}(x_{L+1,i}^{post})}{\partial x_{L+1,i}^{post}}\right\|_2 \|\mathbf{J}_{LN}(x_{L,i}^{post,5})\|_2 \|(0, 0, ..., [\mathrm{ReLU}(x_{L,i}^{post,3}W^{1,L})]_p, ..., 0)^\top\|_2 \tag{37}$$

$$= \left\|\frac{\partial \mathcal{L}(x_{L+1,i}^{post})}{\partial x_{L+1,i}^{post}}\right\|_2 \|\mathbf{J}_{LN}(x_{L,i}^{post,5})\|_2 |[\mathrm{ReLU}(x_{L,i}^{post,3}W^{1,L})]_p| \tag{38}$$

这意味着

$$\left|\frac{\partial \mathcal{L}(x_{L+1,i}^{post})}{\partial W_{pq}^{2,L}}\right|^2 \le \left\|\frac{\partial \mathcal{L}(x_{L+1,i}^{post})}{\partial x_{L+1,i}^{post}}\right\|_2^2 \|\mathbf{J}_{LN}(x_{L,i}^{post,5})\|_2^2 |[\mathrm{ReLU}(x_{L,i}^{post,3}W^{1,L})]_p|^2 \tag{39}$$

因为所有的导数都是有界的，我们有$\partial \mathrm{L}(\mathrm{x}_{\mathrm{post}}$ k$\frac{\partial \mathrm{x}_{\mathrm{post\ L+1,i}})\mathrm{k22}=}{\mathrm{L+1,我}}$ O(1)。所以

由于kxpost,3 k22= d， [xL,ipost,3 L, iw 1,L]p有N(0,1)的分布，我们有切尔诺夫界限

$$\Pr[|[x_{L,i}^{post,3}W^{1,L}]_p| \geq a_0] \leq \exp(-\frac{a_0^2}{2}).$$

所以

$$\Pr[\mathrm{ReLU}([x_{L,i}^{post,3}W^{1,L}]_p)^2 \geq 2\ln 100d] \leq \frac{0.01}{d}.$$

因此，在概率至少为0.99的情况下，对于所有p = 1、2、…， d我们得到ReLU([xpost,3L,iW 1,L]p)2≤2 ln 100d。

|kxpost,5k22−Ekx post,5 Since with probability 1−δ(), L,i L,i k22|≤, 5 l, Ekxpost ik22

我们有kxpost,5L,i k22≤(1 +)EkxL,ipost,5 k22。利用引理4.2，我们有

$$\Pr[\|x_{L,i}^{post,5}\|_2^2 \leq \alpha_0 \mathbb{E}\|x_{L,i}^{post,5}\|_2^2] \tag{41}$$

$$\leq \frac{(1+\epsilon)\mathbb{E}\|x_{L,i}^{post,5}\|_2^2 - \mathbb{E}\|x_{L,i}^{post,5}\|_2^2}{(1+\epsilon-\alpha_0)\mathbb{E}\|x_{L,i}^{post,5}\|_2^2} \tag{42}$$

$$= \frac{\epsilon}{1+\epsilon-\alpha_0} \tag{43}$$

对于任意常数$\alpha_0 > 0$，等于

$$\Pr[\|x_{L,i}^{post,5}\|_2^2 \geq \alpha_0 \mathbb{E}\|x_{L,i}^{post,5}\|_2^2] \geq 1 - \frac{\epsilon}{1+\epsilon-\alpha_0} \tag{44}$$

所以根据联合界限，概率至少为0.99−δ()−1+−α ∂ L(x post)我们有|L+1,i |2= 0 ∂我 2,我 pq

O(kJLN (xpost, 5L, i)k22|[ReLU(xpost, 3L, iW 1, L)]p|2)≤O(kx2d ln 100d)≤O(d ln 100d)。所以我们有

$$|\frac{\partial \tilde{\mathcal{L}}}{\partial W_{pq}^{2,L}}|^2 = |\frac{1}{n}\sum_{i=1}^{n}\frac{\partial \mathcal{L}(x_{L+1,i}^{post})}{\partial W_{pq}^{2,L}}|^2 \tag{45}$$

$$\leq \frac{1}{n}\sum_{i=1}^{n}|\frac{\partial \mathcal{L}(x_{L+1,i}^{post})}{\partial W_{pq}^{2,L}}|^2 = \mathcal{O}(\frac{\ln d}{\alpha_0}) \tag{46}$$

和

$$\|\frac{\partial \tilde{\mathcal{L}}}{\partial W^{2,L}}\|_F = \sqrt{\sum_{p,q=1}^{d}|\frac{\partial \tilde{\mathcal{L}}}{\partial W_{pq}^{2,L}}|^2} = \mathcal{O}(\sqrt{\frac{d^2\ln d}{\alpha_0}})$$

。

预ln变压器的损耗可以写成

$$\tilde{\mathcal{L}}(x_{Final,1}^{pre},...,x_{Final,n}^{pre}) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(x_{Final,i}^{pre}) \tag{47}$$

使用同样的技术，在预ln设置中，L (xpreF final,i)相对于最后一层参数w2 的梯度， L可以写成

$$\frac{\partial \mathcal{L}(x_{Final,i}^{pre})}{\partial W_{pq}^{2,L}} = \frac{\partial \mathcal{L}(x_{Final,i}^{pre})}{\partial x_{Final,i}^{pre}}\frac{\partial x_{Final,i}^{pre}}{\partial x_{L+1,i}^{pre}}\frac{\partial x_{L+1,i}^{pre}}{\partial x_{L,i}^{pre,5}}\frac{\partial x_{L,i}^{pre,5}}{\partial W_{pq}^{2,L}} \tag{48}$$

$$= \frac{\partial \mathcal{L}(x_{Final,i}^{pre})}{\partial x_{Final,i}^{pre}}\mathbf{J}_{LN}(x_{L+1,i}^{pre})(0,0,...,[\mathrm{ReLU}(x_{L,i}^{pre,4}W^{1,L})]_p,...,0)^\top \tag{49}$$

因此，梯度的每个分量的绝对值是有界的

$$|\frac{\partial \mathcal{L}(x_{Final,i}^{pre})}{\partial W_{pq}^{2,L}}| \leq \|\frac{\partial \mathcal{L}(x_{Final,i}^{pre})}{\partial x_{Final,i}^{pre}}\|_2 \|\mathbf{J}_{LN}(x_{L+1,i}^{pre})\|_2 \|(0,0,...,[\mathrm{ReLU}(x_{L,i}^{pre,4}W^{1,L})]_p,...,0)\|_2 \tag{50}$$

$$= \|\frac{\partial \mathcal{L}(x_{Final,i}^{pre})}{\partial x_{Final,i}^{pre}}\|_2 \|\mathbf{J}_{LN}(x_{L+1,i}^{pre})\|_2 |[\mathrm{ReLU}(x_{L,i}^{pre,4}W^{1,L})]_p| \tag{51}$$

由于kxpre,4L,i k22= d和[xpre,4L,i W 1,L]p服从分布N(0,1)，利用切尔诺夫界限我们有

$$\Pr[|[x_{L,i}^{pre,4}W^{1,L}]_p| \geq a_0] \leq \exp(-\frac{a_0^2}{2}).$$

所以

$$\Pr[\mathrm{ReLU}([x_{L,i}^{pre,4}W^{1,L}]_p)^2 \geq 2\ln 100d] \leq \frac{0.01}{d}.$$

所以在概率至少为0.99的情况下，对于所有p = 1、2、…， d我们有ReLU([xpre,4L,iW 1,L]p)2≤2 ln 100d。

由于有概率1−δ(), Ekxpre k2 ≤, |kxpreL+1,i k2 2−Ekx L+1,i prek2 2| L + 1,我2

我们有kxpreL+1 ik22≤(1 +)EkxL+1 iprek22。利用引理5，我们有

$$\Pr[\|x_{L+1,i}^{pre}\|_2^2 \leq \alpha_0 \mathbb{E}\|x_{L+1,i}^{pre}\|_2^2] \tag{52}$$

$$\leq \frac{(1+\epsilon)\mathbb{E}\|x_{L+1,i}^{pre}\|_2^2 - \mathbb{E}\|x_{L+1,i}^{pre}\|_2^2}{(1+\epsilon-\alpha_0)\mathbb{E}\|x_{L+1,i}^{pre}\|_2^2} \tag{53}$$

$$= \frac{\epsilon}{1+\epsilon-\alpha_0} \tag{54}$$

等于

$$\Pr[\|x_{L+1,i}^{pre}\|_2^2 \geq \alpha_0 \mathbb{E}\|x_{L+1,i}^{pre}\|_2^2] \geq 1 - \frac{\epsilon}{1+\epsilon-\alpha_0} \tag{55}$$

根据联合界限，概率为0.99−δ()我们有|L+1,i |2≤1+−α 0 ∂ W pq 2,L ∂ L(x preF inal,i) 2

$\mathcal{O}(\left[\|\mathbf{J}_{LN}(x^{pre}_{L+1,i})\|_2^2|[\mathrm{ReLU}(x^{pre,4}_{L,i}W^{1,L})]_p|^2\right]) \leq \mathcal{O}(\frac{2d\ln 100d}{\|x^{pre}_{L+1,i}\|_2^2}) \leq \mathcal{O}(\frac{d\ln d}{\alpha_0\mathbb{E}\|x^{pre}_{L+1,i}\|_2^2}) = \mathcal{O}(\frac{\ln d}{\alpha_0 L})$. So we have

$$|\frac{\partial\tilde{\mathcal{L}}}{\partial W^{2,L}_{pq}}|^2 = |\frac{1}{n}\sum_{i=1}^{n}\frac{\partial\mathcal{L}(x^{pre}_{Final,i})}{\partial W^{2,L}_{pq}}|^2 = \mathcal{O}(\frac{\ln d}{\alpha_0 L}) \quad (56)$$

Thus $\|\frac{\partial\tilde{\mathcal{L}}}{\partial W^{2,L}}\|_F = \sqrt{\sum_{p,q=1}^{d}|\frac{\partial\tilde{\mathcal{L}}}{\partial W^{2,L}_{pq}}|^2} \leq \mathcal{O}(\sqrt{\frac{d^2\ln d}{\alpha_0 L}})$.

取α0 = 101，我们有至少0.99−δ()−0.9+的概率，对于后ln Transformer我们有√ k∂w∂t2,L≈kF≤O(dq ln d) 对于 Pre-LN Transformer， 我们有k∂w∂L≈2,LkF≤O(dq ln L d) □

F. 扩展到其他层

为简单起见，记为xl = Concat(xl,1, ···)， xl,n)∈$R^{nd}$and xkl = Concat(xkl,1, ···， xkl,n)∈$R^{nd}$for k ={1,2,3,4,5}。则在后ln变压器中，第l层(以w2, 1为例)参数的梯度为

$$\frac{\partial\tilde{\mathcal{L}}}{\partial W^{2,l}} = \frac{\partial\tilde{\mathcal{L}}}{\partial x^{post}_{L+1}}(\prod_{j=l+1}^{L}\frac{\partial x^{post}_{j+1}}{\partial x^{post}_{j}})\frac{\partial x^{post}_{l+1}}{\partial W^{2,l}},$$

在哪里

$$\frac{\partial x^{post}_{j+1}}{\partial x^{post}_{j}} = \frac{\partial x^{post}_{j+1}}{\partial x^{post,5}_{j}}\frac{\partial x^{post,5}_{j}}{\partial x^{post,3}_{j}}\frac{\partial x^{post,3}_{j}}{\partial x^{post,2}_{j}}\frac{\partial x^{post,2}_{j}}{\partial x^{post}_{j}}.$$

后ln变压器层的雅可比矩阵为:

$$\frac{\partial x^{post}_{j+1}}{\partial x^{post,5}_{j}} = \begin{pmatrix}\mathbf{J}_{LN}(x^{post,5}_{j,1}) & & \\ & \ddots & \\ & & \mathbf{J}_{LN}(x^{post,5}_{j,n})\end{pmatrix} \quad (57)$$

$$\frac{\partial x^{post,5}_{j}}{\partial x^{post,3}_{j}} = \begin{pmatrix}I & & \\ & \ddots & \\ & & I\end{pmatrix} + \begin{pmatrix}W^{2,j} & & \\ & \ddots & \\ & & W^{2,j}\end{pmatrix}\begin{pmatrix}\mathbf{J}^{j}_{1} & & \\ & \ddots & \\ & & \mathbf{J}^{j}_{n}\end{pmatrix}\begin{pmatrix}W^{1,l} & & \\ & \ddots & \\ & & W^{1,l}\end{pmatrix} \quad (58)$$

在哪里

$$\mathbf{J}^{j}_{i} = \mathrm{diag}\left(\sigma'\left(x^{post,3}_{j,i}\left(\mathbf{w}^{1,j}_{1}\right)^{\top}\right),...,\sigma'\left(x^{post,3}_{j,i}\left(\mathbf{w}^{1,j}_{d}\right)^{\top}\right)\right) \in \mathbb{R}^{d\times d}$$

$$\frac{\partial x^{post,3}_{j}}{\partial x^{post,2}_{j}} = \begin{pmatrix}\mathbf{J}_{LN}(x^{post,2}_{j,1}) & & \\ & \ddots & \\ & & \mathbf{J}_{LN}(x^{post,2}_{j,n})\end{pmatrix} \quad (59)$$

$$\frac{\partial x^{post,2}_{j}}{\partial x^{post}_{j}} = \begin{pmatrix}I & & \\ & \ddots & \\ & & I\end{pmatrix} + \begin{pmatrix}\frac{1}{n}W^{V,j} & \dots & \frac{1}{n}W^{V,j} \\ \vdots & \ddots & \vdots \\ \frac{1}{n}W^{V,j} & \dots & \frac{1}{n}W^{V,j}\end{pmatrix} \quad (60)$$

利用Hölder的不等式，我们有

$$\mathbb{E}\|\frac{\partial x^{post}_{j+1}}{\partial x^{post}_{j}}\|_2 \leq \mathbb{E}\left[\|\frac{\partial x^{post}_{j+1}}{\partial x^{post,5}_{j}}\|_2\|\frac{\partial x^{post,5}_{j}}{\partial x^{post,3}_{j}}\|_2\|\frac{\partial x^{post,3}_{j}}{\partial x^{post,2}_{j}}\|_2\|\frac{\partial x^{post,2}_{j}}{\partial x^{post}_{j}}\|_2\right] \quad (61)$$

$$\leq\sqrt{\mathbb{E}\left[\|\frac{\partial x_{j+1}}{\partial x^{post,5}_{j}}\|_2^2\right]\mathbb{E}\left[\|\frac{\partial x^{post,5}_{j}}{\partial x^{post,3}_{j}}\|_2^2\|\frac{\partial x^{post,3}_{j}}{\partial x^{post,2}_{j}}\|_2^2\|\frac{\partial x^{post,2}_{j}}{\partial x^{post}_{j}}\|_2^2\right]} \quad (62)$$

∂x j+1 Since =diag(J_{LN}(xpost,5), ···, J_{LN}(xpost,5)), ∂x post,5 j, 1 j,n j S∂x post E k∂x post,5 j+1 k22≈q d j 我们有 E kxpost 5≈2 时 问 kxpost,5j,1k22集中在它的期望Ekxpost,5j,1k22根据引理2等于2^3d。因此，当我们估计后ln transformer的∂w∂L~2,L的范数时，存在一个术语O(3-2 (L−L)/2)，它以指数形式de-当l变小时的折痕。同理，在ln前变压器中，梯度可以写成

$$\frac{\partial\tilde{\mathcal{L}}}{\partial W^{2,l}} = \frac{\partial\tilde{\mathcal{L}}}{\partial x^{pre}_{Final}}\frac{\partial x^{pre}_{Final}}{\partial x^{pre}_{L+1}}(\prod_{j=l+1}^{L}\frac{\partial x^{pre}_{j+1}}{\partial x^{pre}_{j}})\frac{\partial x^{pre}_{l+1}}{\partial W^{V,l}},$$

在哪里

$$\frac{\partial x^{pre}_{j+1}}{\partial x^{pre}_{j}} = \frac{\partial x^{pre}_{j+1}}{\partial x^{pre,3}_{j}}\frac{\partial x^{pre,3}_{j}}{\partial x^{pre}_{j}}.$$

Pre-LN Transformer层的雅可比矩阵为:

$$\frac{\partial x_{j+1}^{pre}}{\partial x_j^{pre,3}} = \begin{pmatrix} I & & \\ & \ddots & \\ & & I \end{pmatrix} + \begin{pmatrix} W^{2,j} & & \\ & \ddots & \\ & & W^{2,j} \end{pmatrix} \begin{pmatrix} \mathbf{J}_1^{(h')} & & \\ & \ddots & \\ & & \mathbf{J}_n^{(h')} \end{pmatrix} \begin{pmatrix} W^{1,j} & & \\ & \ddots & \\ & & W^{1,j} \end{pmatrix} \begin{pmatrix} \mathbf{J}_{LN}(x_{j,1}^{pre,3}) & & \\ & \ddots & \\ & & \mathbf{J}_{LN}(x_{j,n}^{pre,3}) \end{pmatrix} \quad (63)$$

$$\frac{\partial x_j^{pre,3}}{\partial x_j^{pre}} = \begin{pmatrix} I & & \\ & \ddots & \\ & & I \end{pmatrix} + \begin{pmatrix} \frac{1}{n}W^{V,j} & \cdots & \frac{1}{n}W^{V,j} \\ \vdots & \ddots & \vdots \\ \frac{1}{n}W^{V,j} & \cdots & \frac{1}{n}W^{V,j} \end{pmatrix} \begin{pmatrix} \mathbf{J}_{LN}(x_{j,1}^{pre}) & & \\ & \ddots & \\ & & \mathbf{J}_{LN}(x_{j,n}^{pre}) \end{pmatrix} \quad (64)$$

如果l足够大，则JLN (xprej,i)和
JLN (xj,ipre,3)非常小(O(√1j)阶)因为j在l + 1和l之间，这意味着矩阵∂x pre∂x pre,3的特征值
$\frac{\partial x_{前,3}^{j+1}}{}$ 和 ∂x_{pre}^{j}都接近1。然后我们可以看到
$\frac{\partial x_{pre,j+1}^{j}}{\partial x_{pre,3}}$
Ek∂x pre, 3k2和Ek∂x jprek2接近于1，范数是
$\frac{}{\partial L}$ ~ pre-LN transformer是独立于L的，当L是$\partial_{w2,L}$大。

G. (, δ)有界随机变量的例子

在本节中，我们给出一个(, δ)-有界随机变量的例子。这个例子来自(温赖特，2019)中的例2.5，我们在下面给出一个简短的描述。
如果Z = (Z1，…， Zn)是分布为N(0, In)的高斯pn向量，则Y = kZk2 2 = k=1 Z2 k的分布为χ2n。而EY = Pnk=1 EZk 2= n
均值μ = E[X]的随机变量X被称为*亚指数*，如果有非负参数(ν， α)，使得对于所有的|λ|< α1 E[exp(λ(X−μ))]⩽exp(ν22λ2)。下一个命题来自(温赖特，2019)中的命题2.2。

命题1(亚指数尾界)。*假设*X*是具有参数*(ν, α)的亚指数。*然后*

$$\mathbb{P}[X-\mu \geq t] \leq \begin{cases} \exp(-\frac{t^2}{2\nu^2}) & \text{if } 0 \leq t \leq \frac{\nu^2}{\alpha}, \text{ and} \\ \exp(-\frac{t}{2\alpha}) & \text{for } t > \frac{\nu^2}{\alpha} \end{cases} \quad (65)$$

*9)中的例2.5*使用大学习率，χ2*变量*
Y*是亚指数函数,参数*(ν, α) = (2 n, 4)。*因此我们可以推导出单边界*

$$\mathbb{P}[Y - n \geq n\epsilon] \leq \exp(-n\epsilon^2/8), \quad \text{for all } \epsilon \in (0,1) \quad (66)$$

Y (6)有界与C(0,1)和0−凶凶exp(−n2/8)。

H. 小学习率实验

从理论上讲，我们发现后ln变压器的输出层附近参数的梯度非常大，并建议对这些pa和*(Wainwright, 201*
rameters使得训练不稳定。为了验证使用小步更新是否缓解了这个问题，我们使用了一个非常
small，但固定学习率，并检查是否可以在一定程度上优化后ln变压器(没有学习率预热步骤)。具体来说，我们在优化开始时使用了1e−4的固定学习率，这比论文中的lr_{max}=1e^{-3}要小得多。请注意，由于训练期间的学习率较小，训练收敛速度较慢，而这个设置

在实际的大规模任务中不是很实用。我们将验证曲线与其他基线方法一起绘制在图6中。从图中我们可以看到，验证损失(粉色曲线)在27个epoch中约为4.3。这个损失比使用大学习率(蓝色曲线)训练的后ln Transformer要低得多。但仍然比SOTA性能差(绿色曲线)。

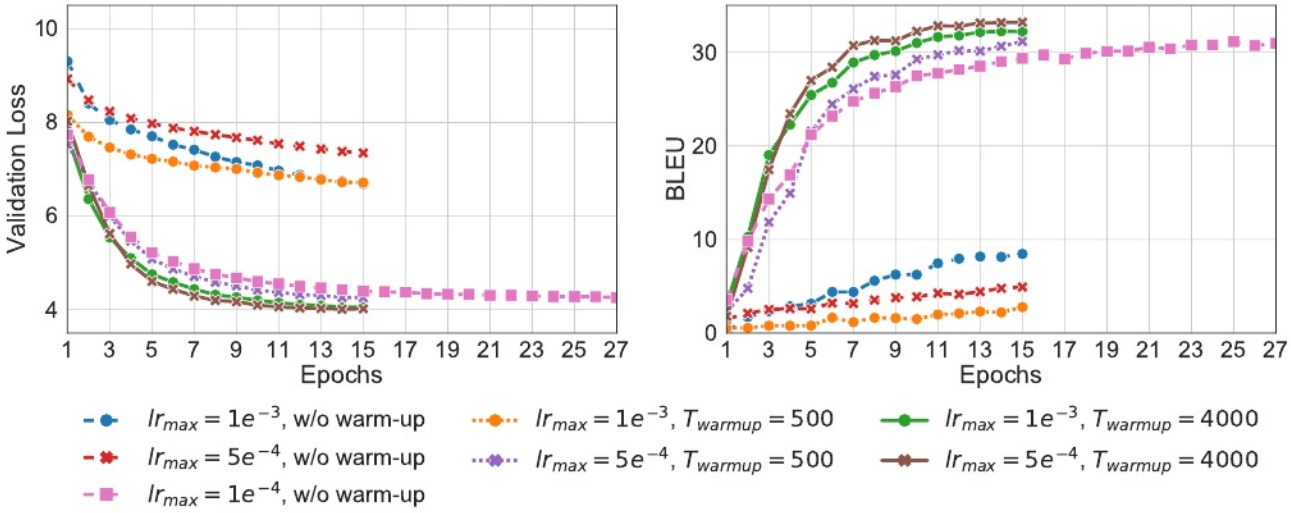


图6。模型在IWSLT14 De-En任务上的性能。