异常值抑制:突破低位极限 Transformer 语言模型

魏秀英1, 2张尚航 , 张云晨2, 4 , 张相国2 , 宫瑞豪1, 2 ,
3 , 1北京航空航天大学 张琪2 ,于凤伟2 , 刘翔龙1*
软件开发环境国家重点实验室 2商汤科技研究院, 3北京大学4电子科技大学
{weixiuying, zhangyunchen, zhangxiangguo,
gongruihao}@sensetime.com shanghang@pku.edu.cn ,
xlliu@buaa.edu.cn

抽象的

Transformer 架构已成为广泛使用的自然语言处理（NLP)模型的基本元素。随着大型 NLP 模型的发展趋势,不断增加的内存和计算成本阻碍了它们在资源有限的设备上的有效部署。因此,变压器量化引起了广泛的研究兴趣。最近的工作认识到结构化异常值是量化性能的关键瓶颈。然而,他们提出的方法增加了计算开销,并且仍然留下异常值。为了从根本上解决这个问题,本文深入研究了异常值的内在诱因和重要性。我们发现LayerNorm (LN) 中的γ充当异常值的有罪放大器,并且异常值的重要性差异很大,其中一些标记提供的一些异常值覆盖了很大的区域,但可以在不产生负面影响的情况下被大幅剪裁。受这些发现的启发,我们提出了一个异常值抑制框架,包括两个组件:伽马迁移和分词裁剪。伽马迁移将异常放大器以等效转换的方式迁移到后续模块,从而有助于形成更加量化友好的模型,而无需任何额外的负担。 Token-Wise Clipping利用token范围的大方差,设计了token-wise粗到细的管道,以有效的方式获得最终量化损失最小的裁剪范围。该框架有效地抑制了异常值,并且可以以即插即用的方式使用。大量的实验证明,我们的框架超越了现有的工作,并且首次将 6 位训练后BERT 量化推向全精度（FP)水平。我们的代码可在https://github.com/wimh966/outlier_suppression 获取。

1 简介

Transformer [1]一直是自然语言处理中最常见的架构之一,与许多流行的自监督模型一起,例如 BERT [2]、RoBERTa [3]、XLNet [4]和 BART [5]。虽然这些预训练模型在性能上表现出了显着的优越性,但内存和计算开销一直是一个普遍关注的问题,特别是在实际开发中。

因此,模型压缩[6,7,8,9]引起了学术界和工业界的广泛关注。其中,量化[10,11,12,13,14,15,16,17,18,19,20]以低精度算术方式工作,是压缩和拟合大型模型的关键方法之一进入轻量级设备。

*通讯作者。

第 36 届神经信息处理系统会议 (NeurIPS 2022)。

如今,研究人员更加关注基于 Transformer 的模型的量化。 [21]提出了一种用于类 BERT 模型的 8 位量化方案。 [22]建议采用分组量化技术并使用二阶 Hessian 信息分析混合精度。 [23, 24]将蒸馏与量化结合起来。 [25]近似非线性运算来实现纯整数量化。

尽管如此,很少有研究调查量化基于 Transformer 的模型的固有瓶颈。

最近,一些论文[26, 27]表明,基于 Transformer 的模型具有非常大的异常值(甚至接近 100 个),并且这些极端异常值以结构化模式表现(主要聚集在几个嵌入维度上,甚至在独特的标记上变得更大)。这些特殊的异常值可能会给量化性能带来毁灭性的损害(例如,即使是 8 位,量化性能也会下降 12% [26])。为了应对这一挑战,现有方法[26]选择了绕过解决方案,例如更精细的量化粒度。然而,这种方案导致计算成本增加,不可避免地阻碍了加速效果。

在本文中,为了抑制异常值而不是绕过它们,我们进行了深入的分析,研究了异常值的诱因以及裁剪异常值的影响。作为归纳,我们发现LayerNorm结构中的缩放参数γ起到了异常值放大器的作用,放大了输出中的异常值。通过提取它,激活对于量化来说变得更加鲁棒。然后通过进一步研究裁剪影响,我们发现裁剪异常值时对最终性能的影响差异很大,其中一些覆盖大面积的更具侵略性的异常值可以被安全地裁剪而不会导致精度下降,但是当重要的异常值时精度会突然下降被剪裁。更有趣的是,尽管那些不太重要的异常值可能以长尾形式呈现,但它们仅由少数标记提供。

在分析的推动下,我们提出了一种异常值抑制框架来突破低位Transformer 语言模型的极限。该框架包含两个关键组件:Gamma Migration和 Token-Wise Clipping,对应于上述两个发现。伽马迁移通过将离群放大器γ迁移到等效变换中的后续模块中,并为量化带来更稳健的激活,而无需额外的计算负担,从而生成更加量化友好的模型。 Token-Wise Clipping进一步有效地在从粗到细的过程中找到具有最小最终量化损失的合适的限幅范围。粗粒度阶段利用那些不太重要的异常值只属于少数标记的事实,可以以标记方式快速获得初步的裁剪范围。然后细粒度阶段对其进行优化。我们提出的框架可以应用于不同的模型和任务,并与现有方法相结合。更本质上,异常值抑制的思想将为NLP 量化研究提供新的思路。

总而言之,我们的贡献如下:

1.我们深入研究了NLP模型中异常值的诱发和剪裁影响,并得出了两个结论
有助于解决变压器量化瓶颈的关键发现。
2.根据研究结果,提出了包含 Gamma 迁移和 Token-Wise Clipping 的异常值抑制框架。该框架高效、易于实现、即插即用。
3. Gamma 迁移从诱导方面抑制了异常值,并产生了更加量化友好的模型,而无需任何额外的推理时间。它将LayerNorm中的离群值放大器以等效变换的方式转移到后续模块,并有助于以较小的量化误差进行激活。

4. Token-Wise Clipping方案从重要性方面抑制异常值,并有效地产生优越的裁剪范围。它可以利用代币范围的巨大方差快速跳过那些不重要的异常值,然后专注于有影响力的区域。

5.对各种 NLP 模型(BERT、RoBERTa、BART)和任务(文本分类、问答和摘要)的大量实验证明,我们的异常值抑制框架为变压器量化和首次将BERT的6位训练后量化(PTQ)和4位量化感知训练(QAT)精度推向全精度水平。

2 预赛

基本符号。我们将矩阵标记为X ,将向量标记为x。运算符·表示标量乘法,用于矩阵或向量的逐元素乘法。此外,我们使用W x作为

(—) X (b) γ (c) X

图 1： BERT-SST-2 上 LayerNorm 的X、 γ和X上异常值的表示。例如,在维度
308、 γ和X都有更尖锐的值。通过排除γ,可以看出X 的分布比
X. 更多证据放在第 2 节中。 D.1.

矩阵向量乘法。具体来说,考虑到 NLP 任务中的标记， Xt,j代表
标记 t 处的元素和嵌入 j， xt表示标记 t 的嵌入。

量化器。量化通常包括两个操作。

$$x^{-} = \text{剪辑}\left(\frac{x}{s} + z, 0, 2^{z} - 1\right), x = (x - z) \cdot s \quad (1)$$

其中s （步长）、 z （零点)是量化参数， b是位设置。第一次手术
称为“Quant”,将连续数(x)映射到仅整数算术矩阵的离散点(x^{-})
计算。第二个运算称为“DeQuant”,在乘法后将其恢复为 x^。

3 异常值分析

对于基于 Transformer 的模型,标准 6/8 位 PTQ 或 4 位 QAT 会导致严重的精度问题
降解。研究每个量化器,我们认识到 LayerNorm 结构的输出和
GELU 函数包含一些尖锐的异常值,这应该是造成大量化的原因
错误。 Sec中的证据和实验结果。 B.2.

为了深入研究有害异常值与量化性能之间的关系，
我们探讨了修剪异常值的潜在诱因和影响。在此之前,先简单介绍一下
首先给出有关异常值的描述（详细信息请参见第 C.1 节）,以帮助理解
以下两部分。异常值显示出主要聚集在某些地方的结构化特征
某些嵌入维度,以及在这些维度上,由唯一标记提供的异常值
像单独的标记和逗号甚至具有更激进的值。

3.1 异常值的诱发

对于异常值的诱发,我们发现LayerNorm中的缩放参数放大了异常值
从嵌入尺寸。某些代币具有更尖锐的异常值的现象可能是
这是由于预训练阶段令牌频率不均匀造成的（参见C.2 节）。这一部分我们主要
解释从根源上解决这些异常值的第一个诱因。另一方面,由于高
调整预训练的成本,我们在下一部分中讨论裁剪影响以抑制这些
从裁剪的角度来看异常值。

考虑到量化 LayerNorm 的挑战,自然的行动是深入研究其内部
结构。对于j处的标记t th 嵌入维度,它首先使用平均值（ut)和
方差（σ_t^2) 每次前向传递,然后使用参数γj和βj缩放和移动该值。

$$\text{层范数：}\ X_{t,j} = \frac{X_{t,j} - u_t}{\sqrt{\sigma_t^2 + }} \cdot \gamma_j + \beta_j \quad (2)$$

然后,通过观察LayerNorm的参数分布,我们惊讶地发现乘子
γ （图 1b)和输出X （图 1a)在相同的嵌入维度上保持异常值。除了，
与输出范围（例如，(-60, 0）)相比,加法器β表示更小的范围（例如,(0,3)） ,所以我们

忽略它来识别关键点。也就是说， γ对于图1a中的异常值起着至关重要的作用，
特别是可以通过充当共享参数来放大令牌之间的异常值。

这一观察启发我们通过从方程中提取γ来消除放大效应。 (2)和
使用非缩放 LayerNorm方程。 （3）。

$$\text{非缩放 LayerNorm : } X_{t,j} = \frac{X_{t,j} - u_t}{\sigma_t^2 +} + \frac{\beta_j}{\gamma_j} \tag{3}$$

图 1c和图 1a显示 Non-scaling LayerNorm 的输出表示更温和的分布
异常值比正常值弱。它不仅与γ确实增强了
离群值,但也表明 X 的表现比 X 对于量化更友好。

为了定量验证X所持有的更适合量化的分布,我们采用余弦
相似度度量来评估量化损失。从表1中,相似度较高的第二行，
即量化误差较小,说明可以使用以下方法提高量化性能
非缩放 LayerNorm。

张量	0	1	2	3	4	5	6	7	8	9	10	11
X	97.16	97.03	97.61	94.37	93.41	93.53	93.31	93.61	94.56	95.62	96.13	98.57
X	99.23	99.22	99.11	99.02	98.99	99.00	98.99	98.83	98.70	99.05	99.44	99.07

表 1： X和X跨 12的量化值（6 位)与真实信号的余弦相似度 (%)
BERT-SST-2 上多头注意力之后的 LayerNorm。越高越好。更多证据见第 2 节。 D.1.

3.2 异常值剪裁的影响

在这一部分中,我们探讨了剪裁异常值的影响,以设计一种可以找到异常值的方法。
适当的量化限幅范围。实验是针对削波影响而设计的
FP 模型的准确性和标记。

对准确性的影响。在修剪异常值并评估最终性能时,我们发现
异常值的重要性差异很大。这里以GELU之后的异常值为例（其他
在秒。 D.2）,图 2表明,大幅剪裁更激进的异常值（剪裁中的信号）
10-100 到 10)甚至不会损害全精度性能,准确度仍为 91.02,而
由于剔除了太多异常值,准确率突然下降至 85.93。

对代币的影响。另一个关键点是不重要的异常值,甚至可以将其剪裁掉
FP 模型中任何准确度的下降都只对应于少数 token。受[26]的启发,他们提到
分隔符标记 [SEP] 关注更大的值。我们也知道提供的不同范围
通过不同的标记。从图2中的红点来看,它代表了被剪切的token的比例，
可以清楚地看到,更激进的异常值占据了 10 到 100 的较大范围
只匹配3%的token。销毁属于少数代币的那些更尖锐的异常值不会
影响性能。

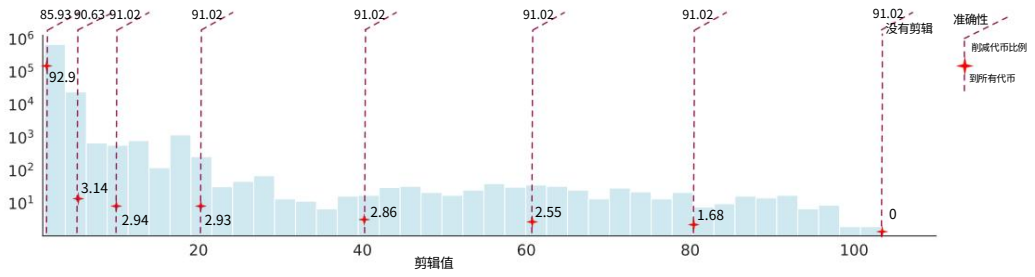


图 2:为了检测裁剪离群值的影响,我们首先使用 (mean + 3 * std) 作为分布来绘制分布
左边界,然后枚举在 RoBERTa-QNLI 上切割张量的值。红点反映了比例
被剪裁的标记。更多证据见第 2 节。 D.2。

之前对准确性影响的调查建议我们考虑最终性能来找到更好的裁剪范围,而一些局部优化方法（如[28])在这里不适合。
后一个关于代币影响的发现鼓励我们利用代币的指示来快速跳过不重要的区域,特别是当它以长尾形式呈现时，其中一些方法（如[29])的效率较低。基于这些,我们将在第二节介绍我们的方法。 4.2.

4 方法

在本节中,我们根据上述分析提出了异常值抑制框架。首先,采用伽玛迁移技术,通过将伽玛迁移到后续模块中来获得更加量化友好的模型。其次,Token-Wise Clipping进一步利用token范围的大方差,有效地找到合适的裁剪范围。

4.1 伽马迁移

正如第 2 节所指出的。 3.1,不经过缩放参数的激活提供了更少的量化误差。通过这种方式,我们拆分了 LayerNorm 函数,将γ迁移到后续结构中,并对 Non-scaling LayerNorm 的输出进行量化。这一变换对于 FP模型来说是等效的,并且为低位模型带来了更鲁棒的激活。整体流程如图3所示。

FP 模型上的迁移等效性。当然,如方程式中所述。 （3） ,我们提取参数γ并将LayerNorm转换为Non-scaling，从而将X与Xt,j分开。 t,j

$$X_{t,j} = X\,(4)_{t,j} \quad \cdot \gamma_j$$

由于在LayerNorm ([30,31,32])之后经常采用残差连接,因此有必要说明将参数γ迁移到两个分支的方法。具体来说,考虑多头注意力之后的LayerNorm （图3） ， γ将从LayerNorm中排除,并移动到下一层的快捷分支和权重。然后LayerNorm变成Non-scaling ,shortcut分支建立一个新的参数γ,下一层的权重可以吸收γ。

现在,我们展示重量如何吸收 γ。对于线性层,我们有以下等式：

$$W(x \begin{matrix} \gamma_1 \\ \gamma_2 \\ \dots \\ \gamma_n \end{matrix}) = (W \begin{matrix} \gamma_1 \gamma_2 \dots \gamma_n \\ \gamma_1 \gamma_2 \dots \gamma_n \\ \dots \\ \gamma_1 \gamma_2 \dots \gamma_n \end{matrix})x, \tag{5}$$

其中x作为列向量， γ ∈ R n。证明可在附录A 中找到。因为γ是共享参数,所以每个标记的嵌入满足方程（1）。 (5)，这保证了将γ成功转移到下一层的权重中。

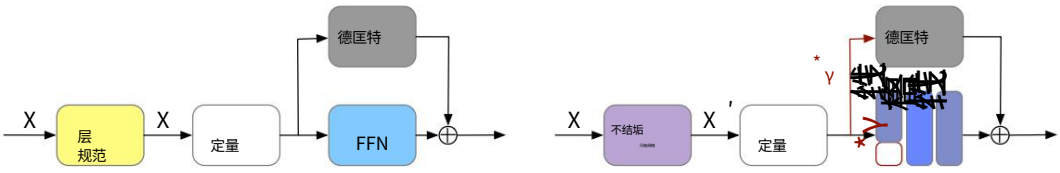


图 3:伽马迁移之前（左)和之后（右)量化流程的比较。原始LayerNorm = 非缩放 LayerNorm γ。其他详细应用如编码器-解码器结构中的LayerNorm等,见图6 、图7。 *

迁移后的量化。从上述等效变换导出,我们概述了迁移过程后的量化模式。从图3中可以看出， X处采用 “Quant”过程，然后量化输出在一个分支上进行矩阵乘法,与参数γ相乘,并在另一分支上经历 “DeQuant”过程。事实上,这意味着将 ，γ计算从 LayerNorm 延迟到捷径分支。因此,这种新设计不会增加计算开销。

迁移的影响。然后,我们分别分析了伽玛迁移对权重和激活的影响,结果表明激活量化负担相对较轻,大大减轻了。

对体重有轻微影响。首先，假设原始 LayerNorm 中输出的绝对最大范围为|max(X)| ∗ |max(γ)|因为在X之前和X缩放函数之后激活的 γ 中，异常值出现在相同的嵌入维度上。对于激活，提取γ会将激活范围减小|max(γ)|次。而表1的结果已经验证了转型带来的激活效益。对于权重来说，权重矩阵不存在与激活相同的嵌入离群现象。因此，权重范围不会被放大|max(γ)|迁移后的次数。实验上，我们还计算了改变后的权重的余弦相似度，并观察到 γ 对权重的影响很小（表2）。

张量	0	1	2	3	4	5	6	7	8	9	10	11
原重量	99.95	99.95	99.95	99.95	99.95	99.95	99.95	99.95	99.95	99.95	99.95	99.95
重量变化	99.95	99.95	99.95	99.90	99.90	99.92	99.94	99.95	99.95	99.95	99.91	99.94

表 2： BERT-SST-2 上 12 个中间层的原始权重和更改权重的量化值（6 位)与真实信号之间的余弦相似度 (%) 。可以看出，两行之间几乎没有差异，特别是与表1相比。

4.2 分词裁剪

基于分析，我们提出了 Token-Wise Clipping 方法，该方法在寻找裁剪范围时考虑最终损失，并采用从粗到细的范例，以 token-wise 的方式有效地最小化它。

关于裁剪异常值对精度的非常不同的影响，我们搜索裁剪范围，相当于步长s，它具有最终量化输出^f(s)和定义为等式1 的真实输出 f之间的最小距离。 （6）。为了有效地实施该过程，特别是当不重要的异常值覆盖较大区域时，下面设计了从粗到细的范例。

$$L(s) = \hat{f}(s) - f \quad {}^{2}_{F}, \tag{6}$$

粗粒度阶段。在这个阶段，我们的目标是快速跳过裁剪对精度影响很小的区域。根据秒。 3.2、长尾区只匹配少数token。

因此，我们建议使用标记t处嵌入的最大值作为其代表（最小值作为负异常值的代表）。可以通过取出每个标记的最大信号来构造具有T 个元素的新张量：

$$\text{哦}^{\text{你}} = \{\max(x1), \max(x2), \dots, \max(xT)\}, \tag{7}$$

其中 o u被标记为上限的集合，o 作为下界的集合。

然后针对o 张量进行裁剪比α 。 你，计算相应的裁剪值cu并用它来裁剪

$$^{\text{UC}}\_ = \text{quantile}(\text{ou}, \alpha), \text{其中} \tag{8}$$

分位数函数计算 o 的第 α 个分位数 你。

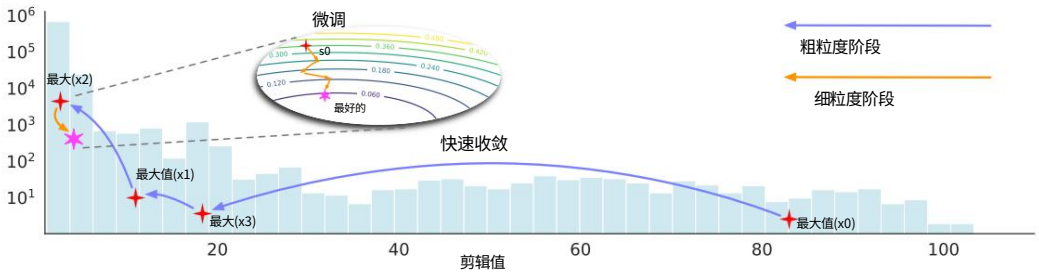


图 4：拟议的 Token-Wise 裁剪的流程图

通过按标记剪切比进行网格搜索,步长s = （b是位宽），量化损失最小得到式中 c_u、c_l 。我们将其标记为s0 ,以便以后优化。

细粒度阶段。现阶段我们的目的是在关键区域做一些细粒度的调整,为最终的效果进一步提供保障。具体来说,在初始化 s0 的情况下,使用基于梯度下降的学习过程以学习率 η 将参数s更新为损失L(s) ,如方程 1 中所述。 （9）。

$$s = s - \eta \frac{\partial L(s)}{\partial s} \tag{9}$$

好处。我们在这里主要从效率和量化性能方面解释粗粒度阶段的好处,其中与其他现有方法的实验比较放在第二节中。D.3。对于效率来说,由于大范围的异常值只对应少数 token,因此从 token 角度穿过不重要区域所需的迭代次数比从 value 角度少得多。此外,代表性集合减少了张量的大小（从 X 中提取） ,因此该方法每次迭代都可以运行得非常快。对于量化性能,第一个粗略步骤已经产生了合适的限幅范围（第 5.2 节） ,这为即将进行的调整提供了良好的初始化点。

5 实验

在本节中,我们进行了两组实验来验证异常值抑制框架的有效性。秒。 5.2显示了每个组件的效果。秒。 5.3列出了在文本分类、问答和摘要任务中与其他现有方法进行比较的结果。总的来说,我们跨 BERT、RoBERTa 和 BART 模型评估了 GLUE 基准[33]、 SQuAD [34, 35] 和 XSum [36]以及CNN/DailyMail [37] 。这里,4-4-4 表示 4 位权重、嵌入和激活。某位下的模型尺寸如表17所示。

5.1 实验装置

实施细节。首先,我们识别量化节点并采用合理的方案,如 FasterTransformer [38]中的方案（详细信息请参见第 B.1 节） 。对于 PTQ,配备我们的框架,我们使用 256 个样本来校准模型。对于 QAT,我们的方法在校准阶段起作用,然后与训练阶段的强大基线LSQ+ [12]相结合。对于训练,我们的方法和基线技术都会搜索学习率等超参数,以进行公平比较。详情参见附录F。

基线。对于 PTQ,我们与流行的校准机制进行比较,包括 MinMax [39]、 OMSE [28]、 Percentile [29]、 EasyQuant [40]和 PEG [26]。对于 QAT,我们给出了 Q-BERT [22]、 Q8BERT [21]和 PEG [26]的结果。此外,由于我们在 QAT 中应用的框架与 LSQ+ [12] 相结合 ,我们展示了纯 LSQ+ 和另一种规范量化方法 PACT [41] 的结果。最后但并非最不重要的一点是,还包括与TernaryBERT [23]中提出的知识蒸馏（KD)相结合的结果。

5.2 消融研究

在本小节中,我们消除了拟议框架中的设计元素（表3）。作为通用插件模块,Gamma Migration 支持MinMax 和 Token-Wise Clipping。 Token-Wise Clipping也大幅超过了基线:QNLI 为 17.53%， MRPC 为 13.22%（与其他校准算法的比较参见第 D.3 节） 。关于细粒度阶段有时并没有比粗粒度阶段有太大改善的现象,我们认为这是由于粗粒度阶段已经产生了足够好的结果。

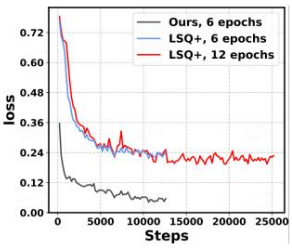


图 5： BERT-SST-2 上的QAT 微调过程。

此外,图 5表明,通过我们的框架提供的良好初始化点,QAT 的训练变得更快、更容易。

方法	可乐可乐	MRPC QNLI QQP RTE SST-2 (acc) (f1/acc) (acc) (acc)	STS-B
	（马特）（acc m/mm）（f1/acc）		（梨。/矛。）
FP32	62.50 87.75/87.23 93.1/90.44 71.64/67.4	92.68 88.78/91.6 80.51 95.18 91.04/90.72	
基线（最小最大）	0.0 34.9/35.0	62.13 51.88/74.37 49.82 77.87 44.11/46.74	
MinMax + Gamma 迁移令牌明智裁剪	0.0 53.53/54.64 87.97/82.84 78.56 78.04/85.3 61.03/63.22	55.6 85.67	
（粗略）	34.95 80.56/80.84 85.05/79.41 79.46 85.96/89.31 66.43 91.63 82.03/82.45		
令牌明智削波伽马迁移 + 令	37.64 81.13/81.26 85.59/79.9 79.66 85.83/89.26 64.62 91.63 83.10/83.51		
牌明智削波46.35 83.38/83.32 87.50/83.33 86.82 86.82/90.01 67.51 92.2			86.83/86.93

表 3:我们为具有 6 位 PTQ 的 RoBERTa 提出的伽玛迁移和分词裁剪的结果。

5.3 主要结果

5.3.1 分类任务结果

方法	比特 CoLA MNLI	MRPC QNLI QQP RTE SST-2 (WEA) (马特.) (acc m/mm) (f1/acc)	STS-B	平均。
	(acc) (f1/acc) (acc) (acc) (Pear./Spear.)			
伯特	32-32-32 59.60 84.94/84.76 91.35/87.75 91.84 87.82/90.91 72.56 93.35		89.70/89.28 83.83	
最小最大	8-8-8 57.08 82.77/83.47 89.90/85.78 90.76 87.84/90.74 69.68 92.78 57.15 84.04/84.29 90.10/85.78 91.12		86.83/88.56 82.28	
欧姆瑟[28]	8-8-8 87.64/90.54 72.20 93.23 6 1.64 84.38/84.53 91.44/87.75 91.49 87.92/90.77 72.20 93.81 35.44		87.90/88.65 82.90	
我们的	8-8-8 74.00/73.30 81.54/76.47 84.66 76.07 / 82.12 64.26 86.27		89.23/89.01 83.96	
奥姆SE	6-6-6		85.57/86.05 73.52	
百分位数[29]	6-6-6 37.32 72.40/71.69 85.09/79.90 79.37 72.58/80.19 61.73 87.27 86.38/87.29 72.93			
EasyQuant [40]	6-6-6 38.16 75.82/75.66 82.51/77.45 84.94 75.31/81.81 65.34 87.27 85.50/86.33 74.49			
我们的	6-6-6 54.40 82.02/81.69 87.45/83.33 89.82 84.69/88.94 70.76 91.86 88.65/88.55 81.19			
聚乙二醇[26] ♣	8-8-8 59.43 82.45 81.25 88.53 91.07 89.42 69.31 92.66		87.92	
我们的♣	8-8-8 59.83 82.93/82.59 91.33/87.99 90.02 87.45/ 90.34 70.04 92.66 88.42/88.81 82.81			
聚乙二醇♣	6-6-6 9.46 32.44/32.77 83.64/78.43 49.46 29.93/62.97 70.76 90.14 52.79/53.22 54.11			
我们的♣	6-6-6 42.27 78.54/78.32 85.33/81.13 85.36 78.47/84.66 68.59 91.74 87.33/87.19 77.31			
罗伯塔	32-32-32 62.50 87.75/87.23 93.1/90.44 92.68 88.78/91.6 80.51 95.18		91.04/90.72 86.40	
最小最大	8-8-8 41.62 87.52/86.88 91.56/88.48 92.11 88.60/91.44 76.90 94.82 91.00/90.66 82.94			
奥姆SE	8-8-8 38.59 87.32/87.14 92.39/89.46 92.51 87.95/90.95 76.53 94.61 90.95/90.65 82.58			
我们的	8-8-8 62.50 87.61/87.31 92.39/89.46 92.53 88.64/91.49 78.34 94.95 91.08/90.73 85.96			
奥姆SE	6-6-6 1.81 72.89/72.65 85.38/78.68 76.53 85.24/88.94 64.26 91.17		80.81/81.99 69.63	
百分位数	6-6-6 20.73 72.23/73.68 84.83/78.43 77.16 82.21/87.44 62.82 88.19 79.41/79.64 70.98			
易量	6-6-6 9.28 74.96/75.87 84.31/76.47 74.04 85.52/89.12 62.45 89.56 80.89/82.38 70.01			
我们的	6-6-6 46.35 83.38/83.32 87.50/83.33 86.82 86.82/90.01 67.51 92.2		86.83/86.93 79.62	
捷运	32-32-32 56.32 86.45/86.55 91.37/87.50 92.31 88.34/91.39 79.06 93.35		90.11/89.94 84.61	
最小最大	8-8-8 55.38 85.87/86.14 89.44/85.29 91.20 88.07/91.24 77.98 93.69 89.90/89.73 83.89			
奥姆SE	8-8-8 54.56 90.07/85.88/86.25 90.31/86.27 90.74 88.21/91.3 78.7 93.58			
我们的	8-8-8 55.53 86.28/86.17 90.40/86.52 91.47 88.25/91.35 80.51 93.92 90.20/89.95 84.50			
奥姆SE	6-6-6 31.06 41.92/42.08 56.37/54.36 52.72 78.96/86.02 51.99 87.39 84.38/85.69 61.01			
百分位数	6-6-6 26.21 74.72/75.29 83.52/74.26 53.71 82.64/87.48 67.15 87.96 63.99/65.01 67.31			
易量	6-6-6 25.66 43.48/43.27 59.26/59.56 50.76 81.89/87.67 52.71 87.73 85.39/86.74 61.31			
我们的	6-6-6 44.51 82.46/82.98 86.41/80.88 86.34 83.60/88.45 71.12 90.94 87.56/87.38 79.10			

表 4:GLUE 基准上的 PTQ 性能。 ♣:采用 PEG 相同量化节点的结果[26]
以进行公平比较。对于百分位数,我们在 [0.999, 0.9999, 0.99999] 中搜索超参数并报告
开发组中最好的。

PTQ。表 4显示了 GLUE 任务上的 PTQ 结果。对于 8 位 BERT 模型,虽然之前
方法通常表现良好,即使在很小的情况下,我们的方法仍然可以获得令人满意的结果
数据集如 CoLA（4.49% 上涨)和 STS-B（1.33% 上涨）。为了充分利用极限,
我们尝试了一种更鼓舞人心的设置,将权重和激活量化为 6 位。可见
我们的整体确实接近 FP 值,在 2.64% 以内。同时,我们也与PEG进行比较[26]
公平地通过获取它们的量化节点。需要注意的是,它们的每嵌入组（PEG)量化
当然会带来额外的计算开销,并且在实际部署中可能无法使用,而我们的
带来良好的效果,并且可以享受硬件无损加速。此外,实验
RoBERTa 和 BART 的结果一致证明了我们的优势,而现有方法
遭受不可忽视的精度下降。平均而言,我们达到了 8.64% 和 11.79%
RoBERT 和 BART 的准确性更高。总而言之,我们提出的方法突破了 6 位的限制
量子化达到新的技术水平。

QAT。特别是,我们证明了我们的方法在 QAT 上的兼容性。表 5列出了结果
BERT,其他参见第 2 节。 D.4。在更困难的设置（4-4-4 位量化)中,我们的异常值抑制
框架赢得了接近浮点的性能,在 4 位上平均降低了 2.70%
量化。产生了良好的初始化,我们的精度下降了可接受的水平（0.7%
QQP,MNLI 上的 1.7%）,没有任何蒸馏和数据增强技巧,而 4.19% 和 3.16%
LSQ+。此外,我们仍然可以利用知识来提高性能
蒸馏,尤其是在 2 位权重和嵌入方面。

方法	比特 CoLA MNLI (WEA) (马特.) (acc m/mm) (f1/acc)	MRPC QNLI QQP RTE SST-2 (acc) (f1/acc) (acc) (acc)	STS-B (梨/矛)	平均。
伯特	32-32-32 59.60 84.94/84.76 91.35/87.75 91.84 87.82/90.91 72.56 93.35		89.70/89.28	83.83
Q8BERT [21]	8-8-8 58.48 -	89.56/- 90.62 87.96/- 68.78 92.24	89.04/-	-
Q-BERT [22]	8-4-8 - 78.08/78.96	- - - - 85.55	-	-
契约[41]	4-4-8 55.23 83.98/83.90 91.58/88.24 91.12 88.19/91.20 71.84 91.86 57.70 84.17/84.02 89.75/85.78 91.27		89.73/89.27	82.89
LSQ+ [12]	4-4-8 88.18/91.16 70.76 91.97 5 7.42 84.22/84.52 89.90/85.78 90.46 88.15/91.25 67.87 92.78 61.06		89.74 /89.3	82.84
聚乙二醇[26]	4-4-8 84.82/84.89 91.26/87.75 91.41 88.45/ 91.40 73.65 92.55		89.36/88.95	82.45
我们的	4-4-8		89.71/89.24	84.05
聚乙二醇	4-4-4 0.0 35.45/35.22 81.22/68.38 49.46 74.17/74.85	0.0/63.18 52.71 76.26	楠/楠	-
协议	4-4-4 0.0 84.97/80.15 87.31 81.68/86.14 62.09 83.03 81.40/81.97 88.34/83.82 88.10 83.11/8 7.24		81.64/81.43	69.37
LSQ+	4-4-4 0.0 64.62 82.34		84.16/83.75	71.49
我们的	4-4-4 50.56 83.05/83.24 89.08/84.31 89.88 87.00/90.33 70.76 91.86		87.64/87.36	81.13
聚乙二醇♣*	4-4-8 57.22 70.04 92.38 83.65 57 85.00/84.31 87.07/87.75 91.28 88.35/91.32 79.24 92.43		89.13 82.64	
我们的♣	4-4-8		89.57/89.20 83.60	
聚乙二醇♣	4-4-4 0.0 35.45/35.22 31.62/0.0 49.46	0.0/63.18 52.71 49.08 -0.0219/-0.0199 29.25		
我们的♣	4-4-4 51.93 83.03/83.24 89.39/85.05 90.33 87.38/90.62 72.56 91.74		88.36/87.91 81.76	
LSQ+(+KD)	4-4-4 14.98 83.59/84.06 92.47/89.46 91.16 87.96/91.01 67.87 85.55		84.17/83.96	75.99
我们的(+KD)	4-4-4 56.67 84.50/84.65 91.61/88.24 91.45 88.59/91.42 74.37 92.55 83.45 /83.38 88.03/82.60 90.66 87.1/90.36		89.13/88.78	83.56
LSQ+(+KD)	2-2-4 9.44 55.60 83.60 47.02 84 .56 /84.31 90.97/87.25 90.83 88.08/91.12 65.70 91.86		36.69/35.89	66.63
我们的(+KD)	2-2-4		86.12/85.78	80.56

表 5:在 BERT 的 GLUE 基准上使用低位激活的不同 QAT 策略之间的比较。 ♣:
结果采用与 PEG [26]相同的量化节点进行公平比较。 :MNLI 的综合得分，
MRPC、QQP 和 STS-B。

5.3.2 问答任务的结果

为了证明我们的方法具有更广泛的适用性,我们在 SQuAD 数据集上对其进行了评估。什么时候
当下降到 6 位量化时,其他方法的性能急剧下降。我们的依然
在 SQuAD v1.1 上,BERT 和 RoBERTa 的性能分别比他们高出 4.73% 和 15.55%。另外，
SQuAD v2.0 上 RoBERTa 和 BART 的提升分别为 12.31% 和 4.96%。

5.3.3 总结任务结果

验证我们的方法对摘要任务的效果具有很高的价值。我们选择经典
数据集 CNN/DailyMail 和 XSum 并报告 BART 的 ROUGE 1/2/L 分数。表 7说明
我们的方法也有利于编码器-解码器模型,并且可以带来近浮点
8 位性能提升约 4%,6 位性能提升约 4%。

方法	位	伯特		罗伯塔		捷运	
	(WEA) SQuAD v1.1 SQuAD v2.0 SQuAD v1.1 SQuAD v2.0 SQuAD v1.1 SQuAD v2.0						
全精度	32-32-32 88.28/80.82	77.34/73.60		92.25/85.83	83.30/80.26	91.63/84.79	80.82/77.41
欧姆瑟[28]	8-8-8	87.90/80.16	76.88/73.08	91.48/84.53	82.53/79.41	90.49/83.11	79.62/76.12
我们的	8-8-8	87.60/79.80	76.93/73.14	91.57/84.86	82.94/79.72	91.08/84.07	80.55/77.04
奥姆SE	6-6-6	79.77/69.10	67.52/63.09	70.64/58.80	45.80/39.95	81.44/70.61	67.89/63.29
百分位数[29]	6-6-6	78.55/67.14	69.12/65.64	67.24/53.28	56.38/51.58	82.45/72.87	68.44/63.29
EasyQuant [40]	6-6-6	80.47/70.08	71.95/68.06	67.85/55.92	47.99/42.21	82.41/71.72	69.93/64.94
我们的	6-6-6	84.48/75.53	74.69/70.55	80.79/70.83	68.47/64.10	83.68/75.34	74.44/70.36

表 6:典型 PTQ 方法在 SQuAD 上的 f1/em 方面的比较。

方法	Bits(WEA) CNN 每日邮报	X和	Bits(WEA) CNN 每日邮报	X和
全精度	32-32-32	45.62/22.85/42.88 42.82/20.11/34.99	32-32-32	45.62/22.85/42.88 42.82/20.11/34.99
欧姆瑟[28]	8-8-8	44.89/22.03/42.18 41.58/18.77/33.73	6-6-6	37.56/15.46/34.92 16.11/2.13/12.22
百分位数[29]	8-8-8	44.67/21.74/41.81 41.47/18.67/33.61	6-6-6	37.02/15.31/34.45 30.10/9.43/22.70
EasyQuant [40]	8-8-8	44.98/22.07/42.24 41.65/18.81/33.77	6-6-6	38.86/16.65/35.99 17.61/2.79/13.38
我们的	8-8-8	45.96/23.15/4 3.45 42.29/19.63/ 34.56	6-6-6	41.00/18.41/38.51 34.61/12.86/27.38

表 7:BART 模型在摘要任务上的 PTQ 结果（以 ROUGE 1/2/L 表示）。

6 结论及局限性讨论

在本文中,我们从诱导和削波影响来分析离群现象
变压器语言模型。在此基础上,我们建立了一个异常值抑制框架
抑制异常值。还存在一些值得更深入研究的悬而未决的问题。
例如,系统地探讨本文的结论是否有益于
其他领域,例如计算机视觉。此外,正如我们在附录中补充的那样,异常值
不仅出现在微调（BERT)模型中,而且出现在预训练模型中,这也很有意义
深入了解预训练过程以更好地理解。

致谢

衷心感谢匿名审稿人的认真审稿和宝贵建议
让这变得更好。该工作得到了国家自然科学基金委的部分支持
资助项目:62022009、61872021、北京市科技新星计划
资助项目:Z191100001119050和中央高校基本科研业务费专项资金。

参考

[1] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、
武卡斯·凯撒和伊利亚·波洛苏欣。您所需要的就是关注。神经信息的进展
处理系统,2017 年 30 日。

[2] Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。 Bert:预训练
用于语言理解的深度双向转换器。 arXiv 预印本 arXiv:1810.04805,
2018.

[3] Yinhan Liu、Myle Ott、Naman Goyal、Jingfei Du、Mandar Joshi、Danqi Chen、Omer Levy、Mike
刘易斯、卢克·泽特尔莫耶和维塞林·斯托亚诺夫。 Roberta:稳健优化的 bert 预训练
方法。 arXiv 预印本 arXiv:1907.11692,2019。

[4]杨志林、戴子航、杨一鸣、Jaime Carbonell、Russ R Salakhutdinov 和 Quoc V
勒。 Xlnet:用于语言理解的广义自回归预训练。进展
神经信息处理系统,2019 年 32 月。

[5] Mike Lewis、Yinhan Liu、Naman Goyal、Marjan Ghazvininejad、Abdelrahman Mohamed,
奥马尔·利维、韦斯·斯托亚诺夫和卢克·泽特尔莫耶。 Bart:序列到序列的去噪
自然语言生成、翻译和理解的预训练。 arXiv 预印本
arXiv:1910.13461,2019。

[6]韩松,毛慧子,William J Dally。深度压缩:通过剪枝、训练量化和霍夫曼编码来压缩深度神经网络。 arXiv 预印本
arXiv:1510.00149,
2015年。

[7]杰弗里·辛顿 (Geoffrey Hinton)、奥里奥尔·维尼亚尔斯 (Oriol Vinyals) 和杰夫·迪恩 (Jeff Dean)。在神经网络中提取知识。
arXiv 预印本 arXiv:1503.02531, 2015。

[8] Barret Zoph 和 Quoc V Le。使用强化学习的神经架构搜索。 arXiv
预印本 arXiv:1611.01578,2016。

[9]沉明珠,梁峰,龚瑞浩,李宇航,李楚明,陈琳,于凤伟,
严俊杰、欧阳万里.一次量化感知训练:极高的性能
低位架构搜索。 IEEE/CVF 国际会议论文集
计算机视觉 (ICCV),第 5340-5349 页,2021 年 10 月。

[10]龚瑞浩,刘相龙,蒋胜虎,李天祥,胡鹏,林家珍,于凤伟,严俊杰。可微分软量化:桥接全精度和低位神经网络。在 IEEE 国际计算机视觉会议 (ICCV),2019 年 10 月。

[11] Steven K Esser、Jeffrey L McKinstry、Deepika Bablani、Rathinakumar Appuswamy 和 Dhar-mendra S Modha。学习了步长量化。 arXiv 预印本 arXiv:1902.08153, 2019。

[12] Yash Bhalgat、Jinwon Lee、Markus Nagel、Tijmen Blankevoort 和 Nojun Kwak。 Lsq+:通过可学习的偏移和更好的初始化来改进低位量化。 IEEE/CVF 计算机视觉和模式识别研讨会会议记录,第696-697 页,2020 年。

[13]朱峰,龚瑞浩,于凤伟,刘翔龙,王艳飞,李哲龙,杨秀奇,严俊杰。面向卷积神经网络的统一 int8 训练。在 IEEE计算机视觉和模式识别会议 (CVPR),2020 年 6 月。

[14] Itay Hubara、Yury Nahshan、Yair Hanani、Ron Banner 和 Daniel Soudry。使用小型校准集进行准确的训练后量化。国际机器学习会议,第 4466-4475 页。 PMLR,2021。

[15] Yury Nahshan、Brian Chmiel、Chaim Baskin、Evgenii Zheltonozhskii、Ron Banner、Alex M Bronstein 和 Avi Mendelson。损失感知训练后量化。 arXiv 预印本arXiv:1911.07190, 2019。

[16]蔡耀辉,姚哲伟,董振,Amir Gholami,Michael W Mahoney,Kurt Keutzer。 Zeroq:一种新颖的零样本量化框架。 IEEE/CVF计算机视觉和模式识别会议论文集,第 13169–13178 页,2020 年。

[17]张相国,秦浩通,丁一夫,龚瑞豪,严庆华,陶仁帅,李宇航,于凤伟,刘相龙。多样化样本生成,实现准确的无数据量化。在 IEEE 计算机视觉和模式识别会议 (CVPR), 2021 年 6 月。

[18] 马库斯·内格尔、拉纳·阿里·阿姆贾德、马特·范·巴伦、克里斯托斯·路易斯和蒂门·布兰克沃特。上或下?用于训练后量化的自适应舍入。国际机器学习会议,第 7197-7206 页。 PMLR,2020。

[19]李宇航,龚瑞浩,谭旭,杨阳,彭虎,张琪,于凤伟,王伟,顾石。 Brecq:通过块重建突破训练后量化的极限。学习表征国际会议,2021 年。

[20]魏秀英,龚瑞浩,李宇航,刘翔龙,于凤伟。 Qdrop:随机丢弃量化以实现极低位的训练后量化。学习表征国际会议,2022 年。

[21] Ofir Zafrir、Guy Boudoukh、Peter Izsak 和 Moshe Wasserblat。 Q8bert:量化的 8 位 bert。 2019年第五届节能机器学习和认知计算研讨会 - NeurIPS版 (EMC2-NIPS),第 36-39 页。 IEEE,2019。

[22]申盛,董震,叶家宇,马林建,姚哲伟,Amir Gholami,Michael W Mahoney,Kurt Keutzer。 Q-bert:基于Hessian的bert超低精度量化。 AAAI 人工智能会议记录,第 34 卷,第 8815-8821 页, 2020年。

[23]张伟,侯鲁,殷一春,尚立峰,陈晓,蒋欣,刘群。 Ternarybert:蒸馏感知的超低位 bert。 arXiv 预印本 arXiv:2009.12812, 2020。

[24]白浩丽,张伟,侯鲁,尚立峰,金晶,蒋欣,刘群,吕迈克尔,欧文·金。 Binarybert:突破 bert 量化的极限。 arXiv 预印本 arXiv:2012.15701, 2020。

[25] Sehoon Kim、Amir Gholami、Zhewei Yao、Michael W Mahoney 和 Kurt Keutzer。 I-bert:纯整数 bert 量化。国际机器学习会议,第 5506-5518 页。PMLR,2021 年。

[26] Yelysei Bondarenko、Markus Nagel 和 Tijmen Blankevoort。了解并克服高效变压器量化的挑战。 arXiv 预印本 arXiv:2109.12948, 2021。

[27]罗紫阳,阿图尔·库尔米泽夫,毛晓曦。位置伪影通过掩码语言模型嵌入传播。 arXiv 预印本 arXiv:2011.04393, 2020。

[28] Yoni Choukroun、Eli Kravchik、Fan Yang 和 Pavel Kisilev。用于高效推理的神经网络低位量化。 2019 年 IEEE/CVF 国际计算机视觉研讨会研讨会 (ICCVW)，第 3009-3018 页。 IEEE，2019。

[29]杰弗里·L·麦金斯特里、史蒂文·K·埃瑟、拉辛库玛·阿普斯瓦米、迪皮卡·巴布拉尼、约翰·V·亚瑟、伊泽特·B·耶尔迪兹和达门德拉·S·莫达。发现接近全精度网络的低精度网络以实现高效的嵌入式推理，2019年。

[30]戴子航，杨志林，杨一鸣，Jaime Carbonell，Quoc V Le，Ruslan Salakhutdinov。Transformer-xl：超越固定长度上下文的细心语言模型。 arXiv 预印本arXiv:1901.02860, 2019。

[31] Mostafa Dehghani、Stephan Gouws、Oriol Vinyals、Jakob Uszkoreit 和 ukasz Kaiser。通用变压器。 arXiv 预印本 arXiv:1807.03819, 2018。

[32]凯文·克拉克 (Kevin Clark)、Minh-Thang Luong、Quoc V Le 和克里斯托弗·D·曼宁 (Christopher D Manning)。Electra：将文本编码器预训练为判别器而不是生成器。 arXiv 预印本 arXiv:2003.10555, 2020。

[33] Alex Wang、Amanpreet Singh、Julian Michael、Felix Hill、Omer Levy 和 Samuel R Bowman。Glue：用于自然语言理解的多任务基准测试和分析平台。 arXiv预印本 arXiv:1804.07461, 2018。

[34] Pranav Rajpurkar，张健，康斯坦丁·洛佩列夫，珀西·梁。 Squad：100,000 多个机器理解文本的问题。 arXiv 预印本 arXiv:1606.05250, 2016。

[35] Pranav Rajpurkar，Robin Jia，Percy Liang。知道你不知道的事情： SQuAD 无法回答的问题。计算语言学协会第 56 届年会记录（第 2 卷：短论文），第 784-789 页。计算语言学协会，2018。

[36] Shashi Narayan、Shay B. Cohen 和 Mirella Lapata。不要给我细节，只给我概要！用于极端概括的主题感知卷积神经网络。 2018 年自然语言处理经验方法会议论文集，第 1797-1807 页。

计算语言学协会，2018。

[37] Ramesh Nallapati，Bowen Zhou，Cícero Nogueira dos Santos，Çaglar Gülçehre，Bing Xing。使用序列到序列 RNN 等进行抽象文本摘要。第 20 届 SIGNLL 计算自然语言学习会议论文集，第 280-290 页。
计算语言学协会，2016。

[38]英伟达。更快的变压器。 https://github.com/NVIDIA/FasterTransformer， 2022 年。

[39]拉古拉曼·克里希那莫提。量化深度卷积网络以实现高效推理：一份白皮书。 arXiv 预印本 arXiv:1806.08342, 2018。

[40]吴迪，唐琪，赵永乐，张明，付英，张德兵。 Easyquant：后-通过规模优化进行训练量化。 arXiv 预印本 arXiv:2006.16669, 2020。

[41] Jungwook Choi、Zhuo Wang、Swagath Venkataramani、Pierce I-Jen Chuang、Vijayalakshmi Srinivasan 和 Kailash Gopalakrishnan。 Pact：量化神经网络的参数化裁剪激活。 arXiv 预印本 arXiv:1805.06085, 2018。

[42]英特尔。 NNCF。 https://github.com/openvinotoolkit/nncf/tree/develop/third_party_integration/huggingface_transformers, 2022 年。

[43] Brian Chmiel、Ron Banner、Gil Shomron、Yury Nahshan、Alex Bronstein、Uri Weiser 等。鲁棒量化：一种模型可以统治所有这些。神经信息处理系统的进展，33：5308–5317，2020。

[44]奥尔加·科瓦列娃、Saurabh Kulshreshtha、安娜·罗杰斯和安娜·拉姆希斯基。 Bert 克星：扰乱变压器的异常维度。 arXiv 预印本 arXiv:2105.06990, 2021。

[45]乔瓦尼·普切蒂、安娜·罗杰斯、亚历山大·德罗兹和菲利斯·德尔奥尔莱塔。干扰变压器的异常尺寸是由频率驱动的。 arXiv 预印本 arXiv:2205.11380, 2022。

[46]吴浩，帕特里克·贾德，张晓杰，米哈伊尔·伊萨耶夫，保利乌斯·米西克维丘斯。深度学习推理的整数量化：原理和实证评估。 arXiv 预印本arXiv:2004.09602, 2020。

[47] Angela Fan、Pierre Stock、Benjamin Graham、Edouard Grave、Rémi Gribonval、Herve Jegou和 Armand Joulin。使用量化噪声进行训练以实现极端模型压缩。 arXiv预印本 arXiv:2004.07320, 2020。

[48]陶超凡,侯鲁,张伟,尚立峰,蒋欣,刘群,罗平,黄毅。通过量化压缩生成预训练语言模型。 arXiv 预印本 arXiv:2203.10705, 2022。

清单

1. 对于所有作者...

   (a)摘要和引言中提出的主要主张是否准确反映了论文的主旨贡献和范围？ [是的]

   (b)您是否描述了您工作的局限性？ [是]在讨论中我们留下了一些作为未来工作的主题。

   (c) 您是否讨论过您的工作可能产生的任何负面社会影响？ [不适用] (d)您是否已阅读道德审查指南并确保您的论文符合他们？ [是的]

2. 如果您包括理论结果...

   (a) 您是否陈述了所有理论结果的全套假设？ [是] (b)您是否提供了所有理论结果的完整证明？ [是]详细证明可以在补充材料中找到。

3. 如果您进行了实验...

   (a)您是否包含了重现主要实验结果所需的代码、数据和说明（在补充材料中或作为 URL）？ [是]我们提供实验代码作为补充材料的一部分。

   (b)您是否指定了所有训练细节（例如,数据分割、超参数、如何选择它们）？ [是]我们将详细的培训设置推迟到补充材料中。 (c)您是否报告了误差线（例如,关于多次运行实验后的随机种子）？ [否]由于我们全面评估不同数据集上各种模型的鲁棒泛化能力,因此误差线的计算成本会很高。

   (d)您是否包括了计算总量和使用的资源类型（例如,类型GPU、内部集群或云提供商）？ [是的]

4. 如果您正在使用现有资产（例如代码、数据、模型)或策划/发布新资产...

   (a) 如果您的作品使用现有资源,您是否引用了创作者？ [是] (b) 您是否提到资产的许可？ [是] (c)您是否在补充材料中或以 URL 的形式包含了任何新资产？ [是] (d)您是否讨论过是否以及如何获得您的数据的人的同意使用/策划？ [不适用]

   (e)您是否讨论过您正在使用/管理的数据是否包含个人身份信息或攻击性内容？ [不适用]

5. 如果您使用众包或对人类受试者进行研究.....

   (a)您是否提供了给参与者的说明全文和屏幕截图（如果适用）？ [不适用] (b)您是否描述了任何潜在的参与者风险,并附有机构审查的链接董事会 (IRB) 批准（如果适用）？ [不适用]

   (c)您是否包括了支付给参与者的估计小时工资以及参与者补偿的总金额？ [不适用]

附录

由于主论文的篇幅限制,我们将在附录中提供补充分析和实验细节,包括伽马迁移中的等效变换证明,
量化挑战的说明,对异常值的更多分析,补充实验以更好
支持我们的观察和方法、相关工作和实施细节。

伽马迁移的补充说明

在本节中,我们首先给出等价变换方程的证明。 (5) 。然后是详细的迁移
前馈网络(FFN)和交叉注意力模块之后的LayerNorm过程是
给予。特别是,将FFN之后的LayerNorm标记为FFN-LN,将Multi-Head Attention之后的LayerNorm标记为FFN-LN
作为 MHA-LN。

A.1 等价变换的证明

证明方程(5),我们查看矩阵乘法输出中的每个元素。详细地,我们标记
输出为h。

$$\begin{aligned} 你好= & \sum_j W_{i,j} \cdot (\gamma_j \cdot x_j) \\ = & \sum_j (\gamma_j \cdot W_{i,j}) \cdot x_j 。 \end{aligned} \tag{10}$$

因此,对于 h 中的所有元素,我们有:

$$W(x \begin{matrix} \gamma_1 \\ \gamma_2 \\ \dots \\ \gamma_n \end{matrix}) = (W \begin{matrix} \gamma_1 \gamma_2 \dots \gamma_n \\ \gamma_1 \gamma_2 \dots \gamma_n \\ \dots \\ \gamma_1 \gamma_2 \dots \gamma_n \end{matrix})x, \tag{11}$$

参数γ在样本和令牌之间共享,则上式始终成立,并且
下一层的权重可以自然地吸收γ。

A.2 其他结构上的伽马迁移

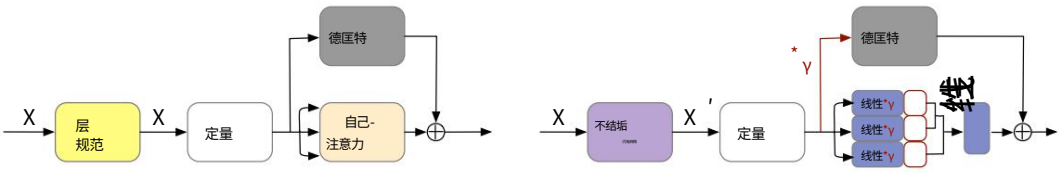


图 6:FFN-LN 中伽玛迁移之前(左)和之后(右)量化流程的比较。这
原始 LayerNorm = 非缩放 LayerNorm $^{*}\gamma$。

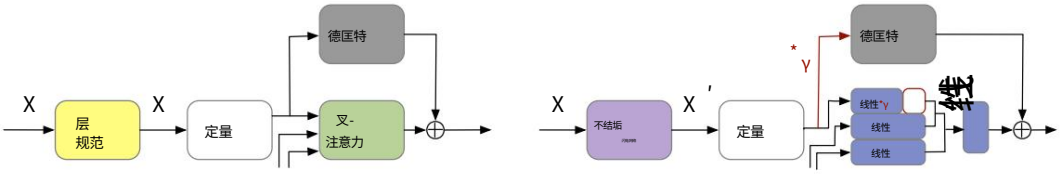


图 7:MHA-LN 中伽玛迁移之前(左)和之后(右)的量化流程比较
交叉注意力模块。

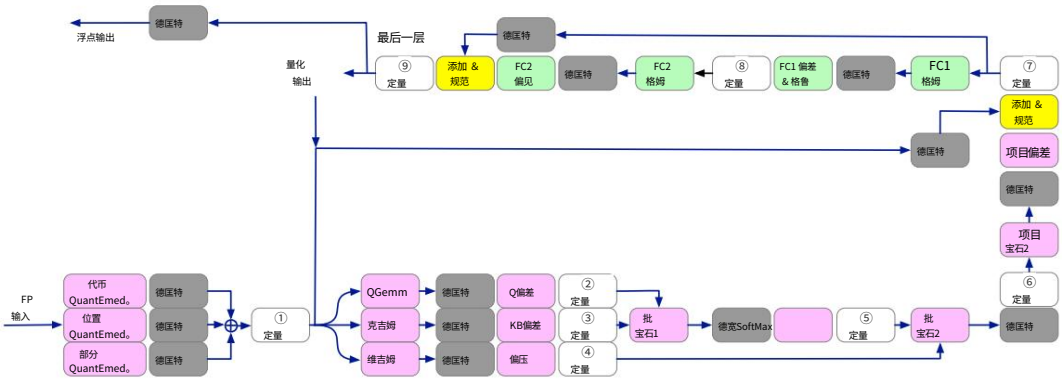


图 8:激活量化节点的位置。在真实推理中,相邻的“DeQuant”和“Quant”
操作将合并为一个“ReQuant”操作以加快计算速度。

B 量化节点

B.1 量化节点的位置

对于插入量化节点的位置,我们发现不同的论文往往有不同的选择,
特别是在激活时。这会给不同方法之间的公平比较带来困难
硬件上的实际开发。

通过调查多个行业[38, 42]和学术解决方案,我们采用FasterTransformer [38]中的一种:量化令牌(位置,令牌类型)嵌入以减少
内存存储。
矩阵乘法中的权重和激活也被量化。需要说明的是,我们只
给予一个量化器相同的激活,因为它对硬件友好。因此我们将量化
快捷分支,对Query、Key、Value的输入采用相同的量化参数
模块,其中一些论文[24]没有,并且可能会在硬件上遇到一些问题。

图 8清楚地说明了激活量化的位置。这里为了便于理解
理解后,我们用序列号标记每个“Quant”节点,并将它们与相关的
表8中的模块名称。

①	②	③	④	⑤	⑥	⑦	⑧	⑨
输入嵌入	查询键值			注意问题	上下文 MHA-LN GELU FFN-LN			

表8:我们将图8中的标签映射到模块名称,它表示在模块名称处插入的量化节点
相应模块的输出。

B.2 有问题的量化节点

在本小节中,我们将进行一些简单而直接的研究来详细说明最有问题的问题
张量(LayerNorm 结构和 GELU 的输出)。验证是在微调的 BERT 上完成的,
RoBERTa 和编码器-解码器模型 BART。

一方面,我们比较 FP 值和量化值之间的余弦相似度
对于每个输出。余弦相似度低于 0.99 的激活节点被视为有问题
位置(结果见表9 、表10)。另一方面,我们可以观察最终的精度恢复情况
通过禁用每种激活的量化。两个实验都表明了障碍
量化 LayerNorm 和 GELU 的输出时。

BERT-STS-B		BERT-QQP		BERT-MRPC	
输出	余弦相似度（%）输出	余弦相似度（%）输出		余弦相似度 (%)	
层.8.GELU 87.83 层.11.GELU 90.68 层.4.MHA-LN 94.60 层.6.MHA-LN 94.63 层.5.MHA-LN 94.66 层.7.MHA-LN 94.85 层.3.MHA- LN 95.19层.10.MHA-LN 95.45层.8.MHA-LN 95.45 层.2.GELU 95.48层.9.MHA-LN 95.60层.5.GELU 96.86 层.0.MHA-LN 96.96层.1. MHA-LN 97.15层.9.GELU 97.42 层.4.GELU 97.60层.2.MHA-LN 97.67层.6.GELU 98.07 层.3.GELU 98.22层.1.GELU 98.34层.7.GELU 98.43层。 10.GELU 98.44层.0.GELU 98.52层.11.MHA-LN 98.60 层.9.FFN-LN 98.79		层.9.GELU 94.19 层.4.MHA-LN 94.40 层.6.MHA-LN 94.45 层.5.MHA-LN 94.55 层.7.MHA-LN 94.60 层.3.MHA-LN 95.01 层.8. MHA-LN 95.05层.2.GELU 95.08层.9.MHA-LN 95.80 层.10.MHA-LN 96.13层.1.MHA-LN 96.84层.0.MHA-LN 96.87 层.10.GELU 97.02层。 2.MHA-LN 97.50层.4.GELU 97.57 层.5.GELU 97.71层.3.GELU 98.30层.11.GELU 98.43 层.1.GELU 98.46层.0.GELU 98.60层.8.GELU 98.63层。 7.GELU 98.69层.11.MHA-LN 98.76层.6.GELU 98.77 层.10.Context 98.96 RoBERTa-QNLI		层.9.GELU 92.00 层.7.MHA-LN 93.05 层.8.MHA-LN 93.14 层.6.MHA-LN 93.22 层.4.MHA-LN 93.28 层.5.MHA-LN 93.44 层.2.GELU 93.94 层.3.MHA-LN 94.15 层.10.MHA-LN 94.36 层.9.MHA-LN 94.58 层.8.GELU 94.68 层.10.GELU 95.81 层.0.MHA-LN 96.99 层.1.MHA-LN 97.12 层.2.MHA-LN 97.66 层.11.GELU 97.70 层.5.GELU 97.91 层.4.GELU 98.04 层.11.MHA-LN 98.16 层.1.GELU 98.18 层.0.GELU 98.31 层.7.GELU 98.42 层.3.GELU 98.67 层.6.GELU 98.74 层.10.FFN-LN 98.94	
罗伯特·塔-MNLI				罗伯特塔-QQP	
输出	余弦相似度(%)输出	余弦相似度(%)输出层.2.GELU		余弦相似度(%)	
层.7.GELU 93.91 层.9.GELU 94.25 层.2.GELU 94.64 层.10.GELU 94.79 层.8.GELU 94.83 层.5.GELU 96.16 层.4.GELU 96.28 层.1.GELU 96.38 层。 3.GELU 96.69 层.6.GELU 96.82层.0.MHA-LN 97.16层.11.MHA-LN 97.26 层.0.GELU 97.30层.10.FFN-LN 97.64层.10.MHA-LN 97.64 层。 1.MHA-LN 97.67层.9.FFN-LN 97.84层.8.FFN-LN 97.90 层.7.FFN-LN 98.05层.9.MHA-LN 98.11层.8.MHA-LN 98.13 层.0. FFN-LN 98.14层.6.FFN-LN 98.25层.1.FFN-LN 98.33 层.5.FFN-LN 98.34层.6.MHA-LN 98.36层.7.MHA-LN 98.36 层.4.FFN- LN 98.39 层.5.MHA-LN 98.43 层.4.MHA-LN 98.47 层.3.FFN-LN 98.48 层.2.MHA-LN 98.50 层.2.FFN-LN 98.54 层.3.MHA-LN 98.55		层.10.GELU 90.08 层.7.GELU 91.60 层.5.GELU 95.58 层.4.GELU 95.59 层.2.GELU 95.89 层.8.GELU 96.02 层.3.GELU 96.33 层.1.GELU 96.52 层。 9.GELU 96.85 层.11.MHA-LN 97.00层.0.MHA-LN 97.13层.6.GELU 97.36 层.0.GELU 97.49层.1.MHA-LN 97.66层.8.上下文97.67层.10. FFN-LN 97.72 层.10.MHA-LN 97.75 层.9.上下文 97.79 层.9.FFN-LN 97.89 层.8.FFN-LN 97.92 层.7.FFN-LN 97.99 层.0.FFN-LN 98.14层.8.MHA-LN 98.15 层.9.MHA-LN 98.17 层.6.FFN-LN 98.19 层.5.FFN-LN 98.26 层.6.MHA-LN 98.28 层.7.MHA-LN 98.31 层。 1.FFN-LN 98.32 层。4.FFN-LN 98.34 层。5.MHA-LN 98.37 层。 3.FFN-LN 98.45 层。4.MHA-LN 98.45 层。2.FFN-LN 98.50 层。 MHA-LN 98.52		93.56 层.3.GELU 94.27 层.4.GELU 95.96 层.1.GELU 96.69 层.5.GELU 96.71 层.0.GELU 97.04 层.0.MHA-LN 97.09 层.7.GELU 97.41 层.1.MHA-LN 97.59 第8层.GELU 97.81 层.8.FFN-LN 98.10 层.7.FFN-LN 98.13 层.0.FFN-LN 98.16 层.1.FFN-LN 98.23 层.6.FFN-LN 98.28 层.6.GELU 98.29 层.7.MHA-LN 98.32 层.8.MHA-LN 98.33 层.6.MHA-LN 98.35 层.5.FFN-LN 98.36 层.2.MHA-LN 98.42 层.5.MHA-LN 98.43 层.4.FFN-LN 98.46 层.3.MHA-LN 98.49 层.4.MHA-LN 98.50 层.2.FFN-LN 层.9.FFN-LN 98.52 层.3.FFN-LN 98.57 层.10.GELU 98.58 层.9.MHA-LN 98.60 层.10.FFN-LN 98.60 层.11.MHA-LN 98.75 第9层.GELU 98.86 层.10.MHA-LN 98.89	98.50
		层.3.MHA-LN 98.55			

表 9:BERT 和 RoBERTa 上的输出与量化结果（6 位)之间的排序余弦相似度楷模。我们的目标是余弦相似度低于 99% 的问题最严重的问题。

BART-CNN/每日邮报		BART-X 总和	
输出	余弦相似度（%）输出		余弦相似度 (%)
层数.4.GELU（解码器)67.96层.3.GELU（解码器)69.50		层.3.GELU（解码器)74.37	
层.4.MHA-LN（编码器-解码器)76.03层.2.GELU（解码器）		层.4.GELU（解码器)75.05	
76.05层.2.MHA-LN（编码器-解码器)解码器)77.88		层.2.GELU（解码器)82.36	
层.0.GELU（解码器)80.83层.5.MHA-LN（编码器)84.20		层.4.MHA-LN（编码器-解码器)82.84	
层.1.MHA-LN（编码器)84.33层.1.MHA-LN（编码器-解码		层.1.MHA-LN（编码器)83.04	
器)85.01层.4 .MHA-LN（编码器)85.03层。3.MHA-LN（编		2.MHA-LN（编码器-解码器)84.31	
码器-解码器)86.78层。3.MHA-LN（编码器)87.12层。0.MHA-		层.4.MHA-LN（编码器)84.53	
LN（编码器)87.30层。1.GELU（解码器）) 87.61 层.2.MHA-		层.5.MHA-LN（编码器)84.69	
LN (编码器) 89.64 层.5.GELU (解码器) 91.78 层.0.MHA-		层.3.MHA-LN（编码器)86.47	
LN (编码器-解码器) 93.62 层.0.GELU (编码器) 95.09		层.1.MHA-LN（编码器-解码器)86.97	
层.2.GELU (编码器)95.91层.3.GELU（编码器)96.44		层.0.MHA-LN（编码器)87.69	
层.3.MHA-LN（解码器)96.90层.5.MHA-LN（解码器)97.46		层.0.GELU（解码器)87.77	
层.2.上下文（编码器-解码器)97.51层.5.MHA -LN（编码		3.MHA-LN（编码器-解码器)88.11	
器-解码器)97.71层。4.FFN-LN（解码器)97.83层。4.GELU		层.2.MHA-LN（编码器)89.14	
（编码器)97.85层。1.GELU（编码器)97.88层。5.GELU（编		层.0.GELU（编码器)92.21	
码器)97.97层。5. FFN-LN（解码器)98.32层.2.上下文（解		层.1.GELU（解码器)93.60	
码器)98.40层.1.MHA-LN（解码器)98.51层.3.FFN-LN（解		层.0.MHA-LN（编码器-解码器)93.61	
码器)98.52层.0.上下文（解码器)98.53层.4 .MHA-LN（解		层.5.FFN-LN（解码器)95.44	
码器)98.54层。2.MHA-LN（解码器)98.63层。2.FFN-LN（解		层.5.GELU（解码器)96.35	
码器)98.66层。1.FFN-LN（解码器)98.71层。0.Context（编		层.3.GELU（编码器)96.41	
码器-解码器）) 98.71 层.0.FFN-LN（解码器)98.72 层.5.		层.2.GELU（编码器)96.57	
上下文（编码器-解码器)98.72 层.4.上下文（解码器)98.92		层.3.MHA-LN（解码器)96.87	
层.0.MHA-LN（解码器)98.93		层.2.上下文（编码器-解码器)96.99	
		层.1.GELU（编码器)97.20	
		层.5.MHA-LN（编码器-解码器)97.56	
		层.0.上下文（编码器-解码器)97.72	
		层.5.GELU（编码器)97.74	
		层.4.FFN-LN（解码器)98.02	
		层.4.GELU（编码器)98.04	
		层.0.上下文（解码器)98.11	
		层.5.MHA-LN（解码器)98.20	
		2.FFN-LN（解码器)98.28	
		层.3.上下文（编码器-解码器)98.31	
		层.1.上下文（解码器)98.32	
		层.3.FFN-LN（解码器)98.36	
		层.1.MHA-LN（解码器)98.38	
		层.5.上下文（编码器-解码器)98.46	
		层.2.上下文（解码器)98.56	
		层.4.MHA-LN（解码器)98.58	
		层.2.MHA-LN（解码器)98.64	
		层.0.FFN-LN（解码器)98.71	
		层.1.FFN-LN（解码器)98.72	
		层.0.MHA-LN（解码器)98.80	

表 10:BART 模型的输出与量化输出（6 位)之间的排序余弦相似度。我们
针对余弦相似度低于 99% 的最有问题的问题。

模型	32-32-32 6-6-6 输入嵌入查询键值注意概率上下文 MHA-LN GELU FFN-LN								
BERT-MRPC	87.75	31.86	31.62	32.11 32.11 32.6 69.05	31.62	31.86	83.09	34.31	31.86
BERT-QQP	90.91	69.0	69.22	68.95 69.24 57.61 58.2 56.45	68.09	69.25	88.93	74.25	70.01
BERT-STS-B	89.70	59.79	57.8		54.02	55.12	84.1	79.6	53.68
罗伯塔-MNLI 87.75 罗伯塔-QNLI		34.90	36.05	35.69 35.54 35.27 65.77	35.68	36.08	66.93	60.50	53.82
92.68 罗伯塔-QQP 91.6		62.13	65.04	64.23 64.73 75.97 76.01	64.54	64.42	84.55	69.71	76.66
		74.37	76.24	76.41	75.50	75.92	87.80	84.28	80.89

表11:量化节点的影响研究。第二列和第三列的比较显示
6 位 MinMax 校准和量化导致性能下降。随后的列显示恢复的
禁用表8中定义的某种输出的量化后的性能,这意味着
量化该节点的效果。例如，“Query”表示禁用 Query 时输出的量化
跨12层的模块。明显的改进以粗体标记。

C 异常值分析

C.1 离群现象

通过深入研究上述有问题的激活,我们发现其中的大异常值会导致
大的量化误差,这些异常值呈现出嵌入中的一些结构化特征

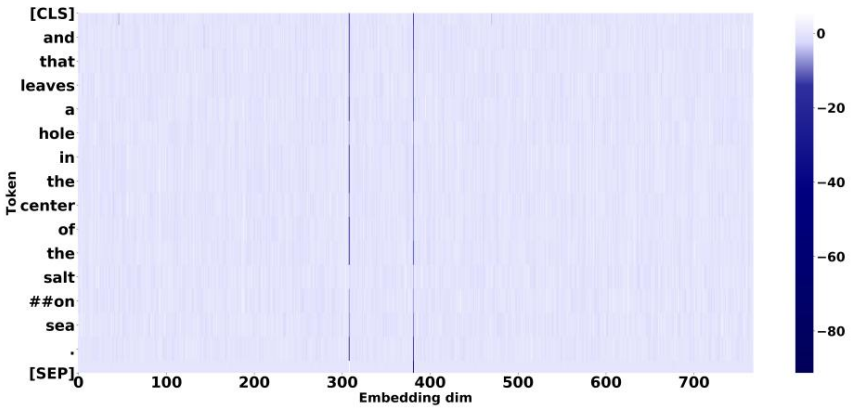


图 9:嵌入暗淡 308 和 381 上的深色条代表BERT-SST-2 中 LayerNorm 输出中几乎所有标记的异常值。

和象征性的观点。几乎所有代币的激活都会涉及特定嵌入维度（例如图9中的 308 和 381 嵌入维度)中的异常值。在这些维度上,与图 10中的 [SEP] 代币相比,一些代币会关注更激进的异常值。 （图10）。事实上,我们发现这种情况经常发生在标记 [SEP]、[CLS]、逗号和句号等标点符号以及其他高频标记（如“the”、“and”、“of”）上。

C.2 关于诱导的详细讨论

在这里,我们从嵌入和令牌的角度讨论离群现象的诱因。

对于嵌入现象， Sec。 3.1解释了缩放参数放大了某些嵌入处的异常值。事实上,我们发现这不仅出现在微调模型中，而且在预训练模型中也很明显。通过在微调 FP 模型时向 LayerNorm 参数注入权重衰减或峰度正则化等约束[43] ,仍然很难在不影响 FP 性能的情况下抑制缩放参数中的激进值。因此,我们推测这种现象对 FP 性能是有益的,尽管它确实给量化带来了挑战。

而且,我们认为token范围的巨大偏差是由预训练阶段的token频率造成的。因为我们发现在预训练期间经常出现包含更激进信号的标记,例如每个示例中都出现 [SEP]、[CLS] 和 “.” 。常用于表达式中。

我们还注意到这些标记的词（标记)嵌入比其他标记具有更大的值。根据这些,可能的解释可能是这样的：频率信息使词嵌入产生偏差

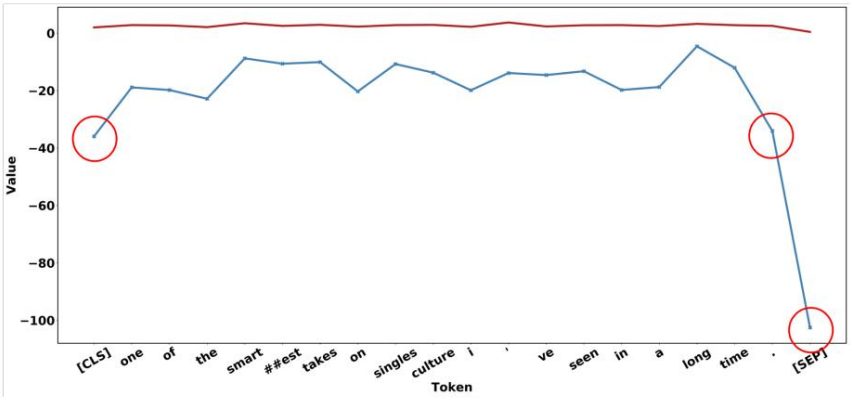


图 10:我们绘制标记范围和标记 [SEP]、[CLS]、 . 注意这里用红色圆圈标记的更尖锐的异常值。

空间并带来不同的功能。更尖锐的异常值扩散到后续层，并且似乎不那么重要，如第 2 节所示。 3.2.因此，我们推测，不受频率信息偏差的良好词嵌入可以在量化方面表现更好。但我们可以以有效的方式找到那些不太重要的异常值并剪掉它们。这更适合训练后量化，无需大规模重新训练。

对于异常值的诱导，请注意， [44]还提到了每个BERT层最后一个LayerNorm中缩放参数与异常值之间的联系。但我们强调缩放参数的放大作用，特别是对于Multi-Head Attention之后的LayerNorm。这自然会产生通过移除缩放参数贡献的量化友好分布的发现。关于不平衡的代币频率，一项并行工作[45]从 FP 性能的角度进行了仔细的探索。

D 补充实验

D.1 LayerNorm 中异常值的补充证据

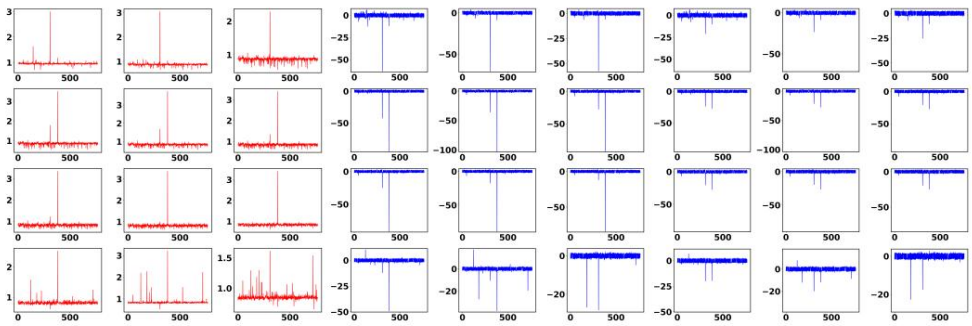
我们在 LayerNorm 中展示了更多相同离群现象的证据，并说明非缩放 LayerNorm 的输出比普通层更适合量化。首先，提供图 11和图 12来建立正式的理解，其中X具有较弱的异常值。此外，表 12中列出了有关余弦相似度的更多定量结果，以表明对最有问题的张量的改进。 B.2通过提取缩放参数γ带来。在这里，我们从嵌入和令牌的角度讨论离群现象的诱因。

D.2 削波影响的补充证据

我们通过将输出削减到表 13 中的不同级别来提供更多准确性和代币影响的证据。

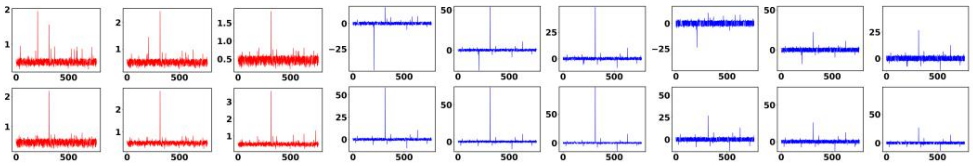
首先，不同的异常值具有非常不同的重要性，其中一些非常大的值可以被急剧裁剪，但不会导致精度大幅下降，而有些值被裁剪后，性能会迅速下降。例如，对于 MHA-LN 的输出，将它们从 -60 剪切到 -45 在 FP 模型中似乎是可靠的，当然在量化模型中也是友好的。然而，从 -40 削波到 -35 将导致约 5% 的性能损失。

另一个关键点是，考虑到代币范围的巨大差异，这些大的异常值仅属于几个代币。例如，对于 (-60, -45) 中的值，大多数层的剪切标记仍然是 3%。因此，从代币角度找到裁剪范围可以帮助快速跳过不太重要的区域。



(一) γ (b) X (c) X

图 11： BERT-SST-2 中 12 个 MHA-LN 中的γ、 X 和X ，其中X = γ X是每个嵌入暗淡处的最高 。对于后两者，我们画出幅度值。可以看出， X具有较温和的分布。



(一) γ (b) X (c) X

图 12： BART-QQP 中跨 6 LayerNorm 的γ、 X 和X ,其中X = γ X 。对于后两者,我们画出
每个嵌入暗淡处的最高幅度值。

模型	0	1	2	3	4	5	6	7	8	9	10	11
BERT-MRPC												
MHA-LN	+2.24	+2.17	+1.48	+4.84	+5.70	+5.55	+5.76	+5.83	+5.56	+4.32	+4.79	+0.85
BERT-QQP												
MHA-LN	+2.35	+2.35	+1.61	+4.00	+4.58	+4.43	+4.51	+4.24	+3.64	+ 3.07	+3.19	+0.24
BERT-STS-B												
MHA-LN	+2.19	+2.03	+1.43	+3.82	+4.39	+4.34	+4.35	+4.03	+3.25	+3.33	+3.44	+0.51
罗伯特·塔-MNLI												
MHA-LN	+1.49	+0.81	+0.25	+0.18	+0.16	+ 0.16	+0.22	+0.19	+0.25	+ 0.31	+0.59	+1.17
FFN-LN	+0.31	+ 0.43	+0.16	+0.24	+0.25	+0.27	+0.28	+0.31	+0.34	+0.43	+0.49	+0.04
罗伯特·塔-QNLI												
MHA-LN	+1.62	+0.88	+0.25	+0.19	+0.17	+0.18	+0.22	+0.18	+0.23	+0.24	+0.52	+1.31
FFN-LN	+0.33	+0.47	+0.22	+0.25	+0.26	+0.30	+0.28	+0.32	+0.31	+0.36	+0.49	+0.53
罗伯特塔-QQP												
MHA-LN	+1.57	+0.93	+0.32	+0.25	+0.21	+0.22	+0.29	+0.33	+0.43	+ 0.39	+0.30	+0.64
FFN-LN	+0.32	+0.52	+0.16	+0.24	+0.27	+0.33	+0.33	+0.42	+0.49	+0.33	+0.45	+0.20
BART-CNN/每日邮报												
MHA-LN（编码器）	+11.26	+14.07	+8.81	+11.25	+13.86	+14.13						
MHA-LN（解码器）	+0.23	+0.19	+0.01	+1.69	+0.23	+1.29						
MHA-LN（编码器-解码器）	+5.21	+13.82	+20.94	+11.94	+22.74	+1.04						
FFN-LN（解码器）	+0.14	+0.03	+0.17	+0.04	+0.21							
BART-X 总和												
MHA-LN（编码器）	+10.90	+15.07	+9.17	+11.77	+13.81	+13.58						
MHA-LN（解码器）	+0.12	+0.09	+1.63	+0.25	+0.61							
MHA-LN（编码器-解码器）	+5.09	+11.75	+14.50	+10.57	+15.96	+1.32						
FFN-LN（解码器）	+0.04	+0.34	+0.23	+0.54	+0.38							

表 12:在 LayerNorm 中提取γ后的余弦相似度 (%) 改进。该指标在 256 上进行评估
来自开发集的样本。 （BART 只有 6 层,因此右半部分是空的。）

D.3 Token-Wise Clipping 与现有方法的比较

我们将 Token-Wise Clipping 的粗略阶段与 OMSE、百分位数和直接步长进行比较
学习并认为我们的表 14更有效,表 15更高效。

我们的 Token-Wise Clipping 搜索针对最终性能的卓越裁剪比,并适用于
一种非常有效的方法（原因已在第 4.2 节中解释） ,评估时间约为 2 分钟
GLUE 任务的 30 个比率。

相反,OMSE 只能最小化局部量化误差,而且表现很糟糕。例如,
它计算出 40 作为图 2所示分布的最佳裁剪范围,而 10 就太多了
更好的。此外,即使使用快速黄金分割搜索,OMSE 运行速度也非常慢。

对于直接步长学习和百分位数方法,虽然他们考虑了最终的损失
在裁剪范围内,在不重要的异常值可以覆盖一定范围的情况下,它们仍然会遇到一些问题
大面积。没有良好初始化点的直接步长学习需要适当的学习率
并花费大量调整时间来实现关键部分。举一个极端的例子。在 QAT 中,步骤
大小已经被充分调整,但我们仍然注意到量化模型可以进一步裁剪。
此外,随着百分位数构建激活的直方图并搜索最佳裁剪比率
从价值角度来看,跳过相对不重要的异常值是非常耗时的。

D.4 QAT 的补充结果

我们将我们的方法应用于 RoBERTa 和 BART 的量化感知训练。从表 16 可以看出,
在 RoBERTa 上,我们的 QNLI 上的 LSQ+ 仍然超过 LSQ+ 2.54%,STS-B 上的 7.53%。在 BART 车型上,

剪裁值	准确性	0	1	2	3	4	5	6	7	8	9	10	11
BERT-MRPC (GELU) 80.0	87.75												
60.0	87.25	0.00 0.00 0.00 0.00 4.33 0.00					0.00		0.00 0.00 0.15 0.00 0.00				
40.0	87.25	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.31 4.58 0.00					0.00						
20.0	87.01	0.00 0.00 3.76 0.00 0.00 0.00 0.00 0.00 2.29 4.68 0.00					0.00						
10.0	87.01	0.00 0.00 3.76 0.00 0.00 0.00 0.00 0.01 3.65 4.79 0.00					0.00						
5.0	87.25	4.64 1.36 3.76 0.00 1.73 0.00 0.00 0.15 4.53 4.84 0.37					3.77						
2.0	87.25	49.88 7.11 19.9 99.99 99.97 26.67 98.83 19.79 15.1 5.00 4.98 45.9											
1.5	84.07	98.9 97.62 97.38 97.01 96.39 94.54 75.55 45.77 92.98											
	78.92	99.96 99.98 99.94 99.94 99.77 99.83 99.77 99.76 99.75 94.72 78.11 98.01											
BERT-QNLI (MHA-LN) 91.84													
-60	91.67	1.96 2.68 0.00 3.92 3.92 3.92 1.96 6.21 1.96 3.92 3.92						3.92	3.92 3.92 0.00 0.00 3.92 3.92 0.22					0.00
-55	91.69	3.92 2.91 10.69 1.97 3.92 3.92 3.92 9.85 16.15 7.17						3.92	0.00 3.92 3.92 3.88 0.06 3.92 3.92					0.00
-50	91.43	3.92 3.92 3.92 16.96 23.13 13.61 5.71 3.92 3.92 22.51						3.92	3.92 0.45 1.46 3.92 4.32 4.23 3.92					0.00
-45	91.25	29.36 23.42 7.46 4.50 3.92 27.48 36.67 32.87 9.39 7.92						3.92	5.52 5.57 3.21 5 .86 6.46 5.99 8.47					0.00
-40	90.28	3.92 34.81 42.43 41.88 23.05 15.63 6.55 41.74 47.64						3.92						0.00
-35	85.54	49.76 37.66 33.98 13.51						3.92						0.01
-30	78.36							3.92 4.04 6.73 3.92 6.03						8.19
-25	72.73							8.16 6.55 8.31						13.84
-20	72.52									19.08 6.94 10.29 40.53				

剪裁值	准确性	0	1	2	3	4	5	0	1	2	3	4	5
巴特可乐 (GELU) 80.0	56.32												
60.0	56.32	0.00 8.48 8.60 0.00 0.00 0.00 0.00 0.00 0.00					0.00						
40.0	56.32	0.00 8.60 8.60 0.00 0.00 0.00 0.00 0.00 0.00					0.00						
20.0	56.32	0.00 8.60 8.60 8.60 8.60 8.60 0.00 8.83 8.79					0.00						
10.0	56.32	0.01 17.21 17.21 8.60 8.61 8.61 0.00					0.00						
5.0	56.32	8.60 4.34 17.21 17.21 8.60 8.61 8.61 8.60					0.00						
2.0	56.58	9.31 8.80 20.3 17.23 8.74 8.80 8.87 9.29					0.43						
1.5	54.06	92.49 90.98 78.52 70.7 79.58 62.35 97.27 92.14 74.54 59.41 53.88 42.17											
	52.37	98.98 98.59 96.46 94.45 96.63 86.5 99.88 99.38 95.1 87.58 84.8 72.38											

表 13:我们直接在开发集上评估准确性,其中输出激活被剪裁值削减。这
随后的列记录了每层被剪裁的标记与所有标记的比率。对于 BART,我们还考虑
解码器中的 GELU 模块。粗体数字表示精度变化的拐点。

方法	CoLA MNLI （马特）（acc m/mm）（f1/acc）	MRPC QNLI QQP RTE SST-2 (acc) (f1/acc) (acc) (acc)	STS-B （梨。/矛。）
罗伯塔 (FP)	62.50 87.75/87.23 93.1/90.44 92.68 88.78/91.6 80.51 95.18 91.04/90.72		
欧姆SE [28] 72.89/72.65 85.38/78.68 76.53 85.24/88.54 64.26 91.17 80.81/81.99	1.81 4.64		
步长学习[12] 71.77/73.18 85.42/79.17 77.28 85.19/88.51 65.34 90.71 80.23/81.25			
百分位[46] 20.73 72.23/73.68 84.83/78.43 77.16 82.21/87.44 62.82 88.19 79.41/79.64			
分词裁剪（粗略阶段） 34.95 80.56/80.84 85.05/79.41 79.46 85.96/89.31 66.43 91.63 82.03/82.45			
伯特（FP）	59.60 84.94/84.76 91.35/87.75 91.84 87.82/90.91 72.56 93.35 89.70/89.28		
OMSE 步	35.44 74.00/73.30 81.54/76.47 84.66 76.07/82.12 64.26 86.27 85.57/86.05		
长学习 百分位数 37.32	35.77 74.11/73.76 82.95/77.94 85.19 75.79/81.91 64.62 87.16 85.78/86.47		
72.40/71.69 85.09/79.90 79.37 72.58/80.19 61.73 87.27 86.38/87.29			
分词裁剪（粗略阶段） 47.21 77.53/78.01 85.40/80.39 86.47 74.98/83.88 64.62 91.17 86.48/87.06			

表 14:现有技术与 6 位 BERT 上 Token-Wise 裁剪的粗略阶段之间的比较
和 RoBERTa 模型。对于百分位数,我们在 [0.999, 0.9999, 0.99999] 中搜索其超参数并报告
开发组中最好的一个。可以看出,只有我们的方法的粗略阶段超越了其他方法。

与最佳基线相比,我们实现了 1.73-32.11 点的绝对改进。异常值
抑制框架可以扩展到其他应用,例如纯整数量化[25]
它还提出了基于 Transformer 的非线性运算的多项式近似
楷模。

E 补充相关工作

量化算法通常分为两类：(1)量化感知训练
（QAT)和（2)训练后量化（PTQ）。前者将 FP 模型微调到低位
并在训练期间通过量化意识获得良好的结果。除了学习之外
为了获得更好的性能， [41, 11]建议学习量化参数。后者,PTQ,
通常对 FP 模型进行快速校准,计算量和数据量要少得多。 [28]
将量化转化为最小均方误差问题。 [40]交替优化
权重的步长和矩阵乘法输出的激活。

奥姆SE		百分位数	Token-Wise 裁剪（粗略阶段）
网格搜索（30次迭代）	黄金分割搜索（3次）		网格搜索（30 次迭代）
1754s	439.29秒	301.49秒	135.73秒

表 15:每种算法 256 个样本的激活校准时间。由于直接步长学习采用OMSE 作为初始化,因此我们这里不比较时间。

最近,量化在基于 Transformer 的模型中变得流行。对于量化感知训练， [21]探索了类似 BERT 模型的 8 位量化。 [22]采用分组量化并应用基于Hessian信息的混合精度量化。 [23]研究了 BERT 上的各种蒸馏损失,并将蒸馏与量化结合起来。 [25]近似Transformer 架构中的非线性函数以享受纯整数推理。 [47]在训练期间对每个前向传递的权重的不同随机子集进行量化,以减少量化噪声。

此外， [48]探讨了量化生成模型的潜在困难。由于此类模型的顺序计算性质,他们发现词嵌入更容易同质,并设计了一种令牌级对比蒸馏方法来克服这一障碍。对于训练后量化， [26]注意到基于 Transformer 的模型中出现的结构化异常值,这些异常值出现在一些嵌入暗淡和特殊分隔符标记处。他们指出,高动态范围甚至会损害 8 位量化性能,并建议针对这一独特的挑战采取按嵌入组量化。当他们解决这个问题并且他们的方法带来额外的计算负担时,我们探索这些结构化异常值的诱发和裁剪影响,并在没有计算开销的情况下解决它们。

F 补充实施细节

对于量化器详细信息,我们将量化节点插入为Sec。 B.1.我们在权重上采用对称每通道量化,在激活上采用非对称每层量化。

对于 PTQ 实验,我们采样了 256 个示例作为校准数据集,在GLUE 基准和 SQuAD上批量大小设置为 32 ,对于 CNN/DailyMail 和 XSum 设置为 4。对于 Token-Wise Clipping 细粒度阶段的学习,我们总是在数据集中以学习率 1e-5 调整 3 个 epoch,因为第一步已经产生了良好的结果。

对于 GLUE 基准上的 QAT 实验,我们为我们的方法配备了 LSQ+ [12]。 Token-Wise Clipping 的粗粒度阶段用于初始化量化参数,细粒度阶段被删除,因为 LSQ+ 配备了步长学习。关于超参数,学习率在{1e-5, 2e-5, 3e-5, 4e-5, 5e-5}中搜索。批量大小通常设置为 32,除非在小型数据集（包括 CoLA、MRPC、RTE 和 STS-B)上也尝试使用较小的批量大小（8 和 16）。至于历元,我们遵循BERT上的[26] （MNL和QQP为3个历元,其他为6个历元） ,RoBERTa上的[25] （MNL和QQP为6个历元,其他为12个历元） ,并采取6或12个历元也可搭乘 BART。

其他超参数在数据集中进行检查并保持固定,包括自注意力丢失率 0.1、隐藏状态丢失率 0.0、权重衰减 0.0 和预热比率 10%。对于 LSQ+ 和 PACT 等基线机制,我们还进行上述学习率和批量大小搜索,以进行公平比较。

方法	钻头 CoLA MNLI (WEA) (Matt.) (acc m/mm) (f1/acc)	MRPC QNLI QQP RTE SST-2 (acc) (f1/acc) (acc) (acc)	STS-B (梨/矛)	平均。
罗伯塔	32-32-32 62.50 87.75/87.23 93.1/90.44 92.68 88.78/91.6 80.51 95.18 91.04/90.72 86.40			
量子噪声[47]	PQ - 83.60/- - - - - - - -			
契约[41]	4-4-4 19.43 78.72/79.55 81.42/73.04 84.55 85.14/88.91 58.12 88.76 72.15/72.46 70.82			
LSQ+ [12]	4-4-4 24.69 83.28/83.24 83.17/75.0 85.12 86.96/90.22 58.12 89.79 78.08/78.41 73.36			
我们的	4-4-4 37.10 84.91/85.2 84.60/77.70 87.66 87.24/90.52 57.76 90.25 85.61/85.33 76.67			
LSQ+(+KD)	4-4-4 30.33 87.17/87.27 89.39/85.05 91.87 88.56/91.48 61.73 92.20 83.18/83.10 77.97			
我们的(+KD)	4-4-4 48.78 87.33/87.16 91.92/88.97 91.93 88.81/91.67 66.79 92.43 88.97/88.76 82.09			
捷运	32-32-32 56.32 86.45/86.55 91.37/87.5 92.31 88.34/91.39 79.06 93.35 90.11/89.94 84.61			
协议	4-4-4 18.72 80.57/80.36 87.99/82.60 85.52 85.09/88.19 57.40 89.45 87.49/87.36 73.86			
LSQ+	4-4-4 18.12 82.41/82.29 88.35/83.58 87.39 86.04/89.64 57.40 90.48 86.89/86.86 74.55			
我们的	4-4-4 50.83 84.81/84.57 90.94/87.01 90.92 87.88/90.93 73.29 92.43 89.22/89.02 82.46			

表 16:RoBERTa 的 GLUE 基准上具有低位激活的不同 QAT 策略之间的比较
和巴特。

任务	胶水		X和
比特(WEA) 伯特·罗伯特·巴特·巴特·巴特			
32-32-32	417.6	475.5	534.1 531.8
8-8-8	104.8	119.2	134.0 133.4
6-6-6	78.7	89.5	100.6 100.2
4-4-8	52.6	59.8	67.3 67.0
4-4-4	52.6	59.8	67.3 67.0
2-2-4	26.5	30.1	34.0 33.8

表 17:量化模型的模型大小 (MB)。

算法 1:按令牌进行裁剪

输入:网格搜索迭代 K、L 层模型、标记数量 T。
{1.粗略阶段：}
损失 = INF， s0 = 1.0
对于k = 0 到 K — 1做
    α = 1 — 0.01 * k;
    对于i = 1 到 L做
        层输入 X,嵌入 j 处的标记 t Xt,j；
        哦 = {maxj X1,j , maxj X2,j , ... , maxj XT ,j};
        洛 = {minj X1,j , minj X2,j , ... , minj XT ,j};
        UC = 分位数（o u , α), cl = 分位数(o , α);
        X = 剪辑(X, cl , 铜）；
    计算步长 s 和量化损失(6);
    如果损失>损失k那么
        损失=损失k， s0 =s;
求初始化步长s0;
{2.细粒度阶段：}
使用方程式优化 s 。 (9)与等式。 (6);
返回优化步长 s；