LLM.int8():大规模 Transformer 的 8 位矩阵乘法

蒂姆·德特默斯λ* 迈克·刘易斯† 尤尼斯·贝尔卡达§ 干 卢克·泽特莫耶†λ

华盛顿大学λ
Facebook 人工智能研究†
拥抱脸§
ENS 巴黎-萨克雷干

抽象的

大型语言模型已被广泛采用,但需要大量 GPU内存进行推理。我们为 Transformer 中的前馈和注意力投影层开发了一种 Int8 矩阵乘法程序,它将推理所需的内存减少了一半,同时保留了完整的精度性能。使用我们的方法,可以加载 175B 参数 16/32 位检查点,转换为 Int8,并立即使用,而不会降低性能。这是通过理解和解决 Transformer 语言模型中高度系统化的新兴特征的属性来实现的,这些特征主导了注意力和 Transformer 的预测性能。为了应对这些特征,我们开发了一个由两部分组成的量化过程, LLM.int8()。我们首先使用向量量化,对矩阵乘法中的每个内积使用单独的归一化常数,以量化大多数特征。然而,对于出现的离群值,我们还采用了一种新的混合精度分解方案,它将离群值特征维度隔离为 16 位矩阵乘法,而仍然超过 99.9% 的值以 8 位相乘。使用 LLM.int8(),我们凭经验证明可以在具有多达 175B 个参数的 LLM 中执行推理,而不会降低任何性能。这一结果使此类模型更易于访问,例如可以在具有消费级GPU 的单个服务器上使用 OPT-175B/BLOOM。我们开源我们的软件。

1 简介

大型预训练语言模型在 NLP 中被广泛采用（Vaswani 等人,2017;Radford 等人， 2019;Brown 等人,2020;Zhang 等人,2022），但需要大量内存进行推理。对于6.7B 参数及以上的大型 Transformer 语言模型,前馈和注意力投影层及其矩阵乘法运算负责消耗参数的 95%2 和所有计算的 65-85%（Ilharco 等人,2020） 。减小参数大小的一种方法是将它们量化为更少的位数并使用低位精度矩阵乘法。考虑到这一目标,已经开发了变压器的 8 位量化方法（Chen 等人,2020； Lin 等人,2020;Zafir 等人,2019;Shen 等人,2020）。虽然这些方法减少了内存使用,但会降低性能,通常需要在训练后进一步调整量化,并且仅针对参数少于 350M 的模型进行了研究。对高达 350M参数的无降级量化知之甚少,而数十亿参数量化仍然是一个开放的挑战。

*大部分研究是作为 Facebook AI Research 的访问研究员完成的。
2其他参数主要来自嵌入层。一小部分来自规范和偏见。

第 36 届神经信息处理系统会议 (NeurIPS 2022)。

在本文中,我们提出了第一个用于 Transformer 的数十亿级 Int8 量化过程,该过程不会导致任何性能下降。我们的程序可以加载具有 16 位或 32 位权重的 175B 参数转换器,将前馈和注意力投影层转换为 8 位,并立即使用生成的模型进行推理,而不会降低任何性能。

我们通过解决两个关键挑战来实现这一结果:在超过 1B 参数的尺度上需要更高的量化精度,以及需要显式表示稀疏但系统的大幅异常值特征,这些特征一旦出现在从 1B尺度开始的所有变换器层中,就会破坏量化精度。 6.7B 参数。一旦这些异常特征出现,这种精度损失就会反映在 C4 评估困惑度（第 3 节)以及零样本精度中,如图 1 所示。

我们证明,通过我们方法的第一部分,即矢量量化,可以在高达 2.7B 参数的尺度上保持性能。

对于向量量化,矩阵乘法可以看作是行向量和列向量的独立内积序列。因此,我们可以对每个内积使用单独的量化归一化常数来提高量化精度。在执行下一个操作之前,我们可以通过列和行归一化常数的外积进行反归一化来恢复矩阵乘法的输出。

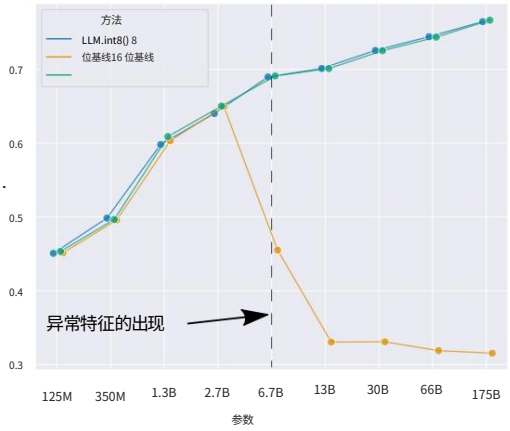

图 1： WinoGrande、HellaSwag、PIQA 和 LAMBADA数据集的 OPT 模型平均零样本精度。所示为 16 位基线、作为基线的最精确的先前 8 位量化方法以及我们新的 8 位量化方法 LLM.int8()。我们可以看到,一旦在 6.7B 参数范围内出现系统异常值,常规量化方法就会失败,而 LLM.int8() 则保持 16 位精度。

为了在不降低性能的情况下扩展到超过 6.7B 参数,了解推理过程中隐藏状态特征维度中极端异常值的出现至关重要。为此,我们提供了一种新的描述性分析,表明大小比其他维度大 20 倍的大特征首先出现在所有 Transformer 层的约 25% 中,然后随着我们缩放 Transformer 逐渐扩展到其他层至 6B 参数。在 6.7B 参数左右,会发生相移,所有变压器层和所有序列维度的 75%都会受到极值特征的影响。这些离群值是高度系统化的:在 6.7B 尺度上,每个序列出现 150,000 个离群值,但它们仅集中在整个 Transformer 的 6 个特征维度上。

将这些离群特征维度设置为零会使 top-1 注意力 softmax 概率质量降低 20% 以上,并将验证困惑度降低 600-1000%,尽管它们只占所有输入特征的 0.1% 左右。相比之下,删除相同数量的随机特征会使概率最多降低 0.3%,并使困惑度降低约 0.1%。

为了支持对此类极端异常值的有效量化,我们开发了混合精度分解,这是我们方法的第二部分。我们对离群特征维度执行 16 位矩阵乘法,对其他 99.9% 的维度执行 8 位矩阵乘法。我们将向量量化和混合精度分解的组合命名为LLM.int8()。我们表明,通过使用 LLM.int8(),我们可以在具有多达 175B 个参数的 LLM 中执行推理,而不会降低任何性能。我们的方法不仅提供了关于这些异常值对模型性能的影响的新见解,而且还首次使得在具有消费级 GPU 的单个服务器上使用非常大的模型成为可能,例如 OPT-175B/BLOOM。虽然我们的工作重点是使大型语言模型可以在不降级的情况下访问,但我们还在附录 D 中表明,我们保持大型模型（例如 BLOOM-176B)的端到端推理运行时性能,并为 GPT-3 模型提供适度的矩阵乘法加速大小为 6.7B 参数或更大。我们开源了我们的软件3并发布了拥抱面部变换器（Wolf et al., 2019)集成,使我们的方法可用于所有具有线性层的托管拥抱面部模型。

[3]https://github.com/TimDettmers/bitsandbytes

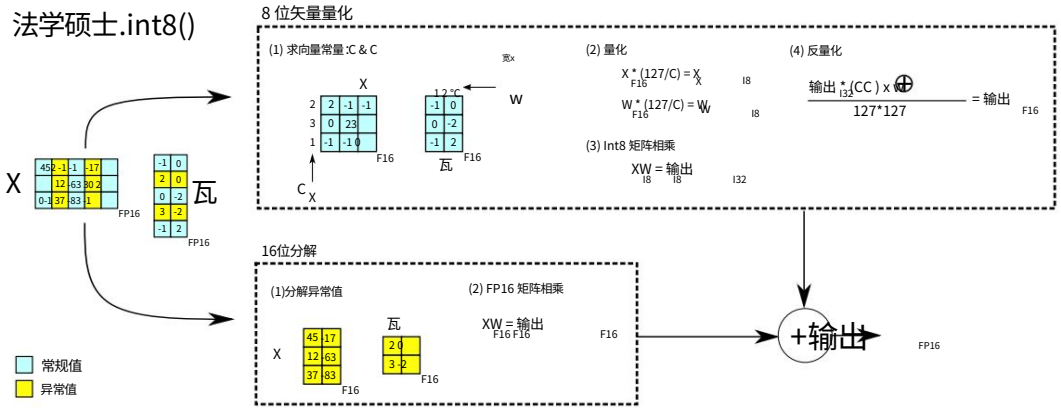

图 2:LLM.int8() 原理图。给定 16 位浮点输入Xf16和权重Wf16,特征和权重被分解为大量特征和其他值的子矩阵。

异常值特征矩阵以 16 位相乘。所有其他值均以 8 位相乘。我们通过按Cx和Cw的行和列绝对最大值进行缩放来执行 8 位向量乘法，然后将输出量化为 Int8。 Int32 矩阵乘法输出Outi32通过归一化常数Cx ⊗ Cw的外积进行反量化。最后,异常值和常规输出都累积在 16 位浮点输出中。

2 背景

在这项工作中,通过缩放变压器模型将量化技术推向极限。我们对两个问题感兴趣:量化技术在什么范围内会失败,为什么会失败,以及这与量化精度有何关系?为了回答这些问题,我们研究高精度非对称量化（零点量化)和对称量化（绝对最大量化）。

虽然零点量化通过使用数据类型的完整位范围提供高精度,但由于实际限制很少使用。绝对最大量化是最常用的技术。

2.1 8 位数据类型和量化

Absmax 量化通过乘以sxf16（ 127除以整个张量的绝对最大值)将输入缩放到 8 位范围[－127, 127] 。这相当于除以无穷大范数并乘以 127。因此,对于 FP16 输入矩阵Xf16 ∈ R 长×高 Int8 absmax量化由下式给出：

$$xi8 = \frac{127 \cdot Xf16}{\max_{ij}(|Xf16_{ij}|)} = \frac{127}{Xf16_{\infty}} Xf16 = sxf16 Xf16,$$

其中表示四舍五入到最接近的整数。

零点量化通过使用归一化动态范围ndx进行缩放,然后移动零点zpx,将输入分布移动到全范围[－127, 127] 。通过这种仿射变换,任何输入张量都将使用该数据类型的所有位,从而减少不对称分布的量化误差。例如,对于ReLU输出,在absmax量化中， [－127, 0)中的所有值都不被使用,而在零点量化中,使用完整的[－127, 127]范围。零点量化由以下等式给出：

$$ndxf16 = \frac{2 \cdot 127}{\max_{ij}(X_{ij}^{f16}) - \min_{ij}(X_{ij}^{f16})} \quad (1)$$

$$zpxi16 = Xf16 \cdot \text{分钟}_{ij}(X_{ij}^{f16}) \quad (2)$$

$$Xi8 = ndxf16 Xf16 \quad (3)$$

为了在运算中使用零点量化,我们将张量Xi8和零点zpxi16送入特殊指令 4 ,该指令在执行 16 位整数运算之前将zpxi16添加到Xi8的每个元素。例如,要将两个零点量化数Ai8和Bi8与其零点zpai16和zpbi16相乘,我们计算:

$$Ci32 = \text{乘 } i16(Azpai16, Bzpbi16) = (Ai8 + zpai16)(Bi8 + zpbi16) \tag{4}$$

如果指令 multiplyi16不可用（例如在 GPU 或 TPU 上）,则需要展开:

$$Ci32 = Ai8Bi8 + Ai8zpbi16 + Bi8zpai16 + zpai16\ zpbi16 \text{ ,其中} \tag{5}$$

Ai8Bi8以 Int8 精度计算,其余以 Int16/32 精度计算。因此,如果 multiplyi16指令不可用,零点量化可能会很慢。在这两种情况下,输出都会累加为 32 位整数Ci32。为了对Ci32 进行反量化,我们除以缩放常数ndaf16和ndbf16 。

具有 16 位浮点输入和输出的 Int8 矩阵乘法。给定隐藏状态$Xf16 \in R^{长\times高}$,具有序列维度s、特征维度h 和输出维度o ,我们使用 16 位输入和输出执行权重矩阵乘法$R^{h\times o}$,如下所示:

$$Xf16Wf16 = Cf16 \approx \frac{1}{cxf16} Ci32 = Sf16 \cdot Ci32$$
$$cwf16 \approx Sf16 \cdot Ai8Bi8 = Sf16 \cdot Q(Af16)\ Q(Bf16), \tag{6}$$

其中Q(·)是absmax或零点量化， cxf16和cwf16是absmax各自的张量缩放常数sx和sw ,或者零点量化的ndx和ndw 。

3 大规模 Int8 矩阵乘法

每个张量使用单个缩放常数的量化方法的主要挑战是单个异常值可能会降低所有其他值的量化精度。因此,每个张量最好有多个缩放常数,例如逐块常数（Dettmers et al., 2022）,以便将异常值的影响限制在每个块中。我们通过使用向量量化来改进块量化的最常见方法之一,即行量化（Khudia 等人,2021） ,如下文更详细所述。

为了处理超过 6.7B 尺度的所有转换器层中出现的大幅异常值特征,矢量量化已不再足够。为此,我们开发了混合精度分解,其中少量的大幅特征维度（约 0.1%)以 16 位精度表示,而其他 99.9% 的值以 8 位精度相乘。由于大多数条目仍然以低精度表示,因此与16 位相比,我们保留了约 50% 的内存减少。例如,对于BLOOM-176B,我们将模型的内存占用量减少了 1.96 倍。

向量方式量化和混合精度分解如图 2 所示。LLM.int8 ()方法是 absmax 向量方式量化和混合精度分解的组合。

3.1 向量量化

增加矩阵乘法缩放常数数量的一种方法是查看矩阵
作为一系列独立内积的乘法。给定隐藏状态Xf16 $\in R^{宽\times高}$和权重矩阵Wf16 $\in R^{h\times o}$,我们可以为Xf16的每一行分配不同的缩放常数cxf16 ,并为 Wf16 的每一列分配cw 。为了反量化,我们将每个内积结果反规范化1 /(cxf16 cwf16)。对于整个矩阵乘法,这相当于通过外积cxf16 ⊗ cwf16进行非规范化,其中cx $\in R^{s}$且cw $\in R^{o}$因此,行和列常数的矩阵乘法的完整方程由下式给出:

$$Cf16 \approx \frac{1}{cxf16 \otimes cwf16} Ci32 = S \cdot Ci32 = S \cdot Ai8Bi8 = S \cdot Q(Af16)\ Q(Bf16), \tag{7}$$

我们将其称为矩阵乘法的向量量化。

[4] https://www.felixcloutier.com/x86/pmaddubsw

3.2 LLM.int8()的核心:混合精度分解

在我们的分析中,我们证明了十亿级 8 位变压器的一个重要问题是它们具有大量特征（列）,这对于变压器性能很重要并且需要高精度量化。然而,矢量量化是我们最好的量化技术,它对隐藏状态的每一行进行量化,这对于离群特征是无效的。幸运的是,我们发现这些离群特征在实践中既极其稀疏又系统化,仅占所有特征维度的 0.1% 左右,从而使我们能够开发一种新的分解技术,专注于这些特定维度的高精度乘法。

我们发现给定输入矩阵Xf16 ∈ R这些离群值在几乎所有序列维度s中系统地出现,但仅限于特定特征/隐藏维度h。因此,我们提出了矩阵乘法的混合精度分解,其中我们将异常值特征维度分离到集合O = {i|i ∈ Z, 0 ⩽ i ⩽ h} 中,其中包含h的所有维度,其中至少有一个异常值大于阈值α的幅度。在我们的工作中,我们发现α = 6.0足以将变压器性能下降减少到接近于零。使用爱因斯坦表示法,其中所有索引都是上标,给定权重矩阵Wf16 ∈ R 乘法定义如下:

$$C_{f16} \approx \sum_{h \in O} X^{h}_{f16} W^{h}_{f16} + S_{F16} \cdot \sum_{h \in O} X^{H}_{i8} W^{h}_{i8} \tag{8}$$

其中Sf16是 Int8 输入和权重矩阵Xi8和Wi8 的非规范化项。

这种 8 位和 16 位的分离允许异常值的高精度乘法,同时使用内存高效的矩阵乘法,8 位权重超过 99.9% 的值。由于对于高达 13B 参数的转换器,离群特征维度的数量不大于 7 (|O| ⩽ 7) ,因此此分解操作仅消耗约 0.1% 的额外内存。

3.3 实验设置

当我们将几个公开可用的预训练语言模型的大小扩展到最多 175B 个参数时,我们测量了量化方法的稳健性。关键问题不在于量化方法对于特定模型的表现如何,而是随着规模的扩大,这种方法的表现趋势。

我们使用两种设置进行实验。一种是基于语言建模困惑度,我们发现这是一种高度稳健的度量,对量化退化非常敏感。我们使用此设置来比较不同的量化基线。此外,我们还针对一系列不同的最终任务评估 OPT 模型的零样本精度下降,并将我们的方法与 16 位基线进行比较。

对于语言建模设置,我们使用在 fairseq（Ott等人,2019)中预训练的密集自回归变压器,参数范围在 125M 到 13B 之间。这些 Transformer 已在Books (Zhu et al., 2015)、英语维基百科、CC-News (Nagel, 2016)、OpenWebText (Gokaslan and Cohen, 2019)、CC-Stories (Trinh and Le, 2018) 和英语上进行了预训练CC100（Wenzek 等人,2020）。有关如何训练这些预训练模型的更多信息,请参阅 Artetxe 等人。 （2021）。

为了评估 Int8 量化后语言建模的退化,我们在 C4 语料库（Raffel 等人,2019)的验证数据上评估 8 位转换器的困惑度,C4 语料库是Common Crawl 语料库的子集。[5] 我们使用 NVIDIA A40用于此评估的 GPU。

为了测量零样本性能的下降,我们使用 OPT 模型（Zhang 等人,2022）,并在 EleutherAI 语言模型评估工具上评估这些模型（Gao 等人,2021）。

3.4 主要结果

在 C4 语料库上评估的 125M 到 13B Int8 模型的主要语言建模困惑结果可以在表 1 中看到。我们看到,当我们缩放时,absmax、row-wise 和零点量化会失败,其中 2.7B 参数之后的模型表现更差比较小的型号。超过 6.7B 参数时,零点量化会失败。我们的方法 LLM.int8() 是唯一保留困惑度的方法。因此,LLM.int8() 是唯一具有良好扩展趋势的方法。

[5] https://commoncrawl.org/

表 1:从 125M 到 13B 参数的不同变压器尺寸的量化方法的 C4 验证困惑度。我们看到,当我们扩展时,absmax、行方式、零点和向量方式量化会导致性能显着下降,特别是在 13B 标记处,其中 8 位13B 困惑度比 8 位 6.7B 困惑度更差。如果我们使用 LLM.int8(),我们会在扩展时恢复完全的困惑。由于非对称量化,零点量化显示出优势,但在与混合精度分解一起使用时不再具有优势。

参数	125M 1.3B 2.7B 6.7B 13B
32 位浮点型	25.65 15.91 14.43 13.30 12.45
Int8 绝对最大值 Int8 零点	87.76 16.55 15.11 14.59 19.08 56.66 16.24 14.76 13.49 13.94
Int8 absmax 行方式 Int8 absmax 向量方式 Int8 零点向量 方式 Int8 absmax 行方式 + 分解	30.93 17.08 15.24 14.13 16.49 35.84 16.82 14.98 14.13 16.48 25.72 15.94 14.36 13.38 13.47
30.76 16.19 14.65 13.25 12.46 Absmax LLM.int8() （向量方式 + 分解） 25.83 15.93 14.44 13.24 12.45 Zeropoint LLM.int8 ( )（向量方式+分解） 25.69 15.92 14.43 13.24 12.45	

当我们查看图 1 中 EleutherAI 语言模型评估工具上 OPT 模型的零样本性能扩展趋势时,我们发现当我们从 125M 参数扩展到 175B 参数时， LLM.int8() 保持了完整的 16 位性能。另一方面,基线 8 位绝对最大向量量化的扩展性很差并且退化为随机性能。

尽管我们的主要重点是节省内存,但我们也测量了 LLM.int8() 的运行时间。与 FP16 基线相比,量化开销可能会减慢参数少于 6.7B 的模型的推理速度。然而,6.7B 参数或更少的模型适合大多数 GPU,并且在实践中不太需要量化。对于相当于 175B 模型的大型矩阵乘法，LLM.int8() 运行时间大约快两倍。附录 D 提供了有关这些实验的更多详细信息。

大规模变压器中出现的 4 个大型特征

当我们缩放变压器时,会出现大量的异常特征,并强烈影响所有层及其量化。给定隐藏状态$X \in R^{长\times高}$,其中s是序列/标记维度， h是隐藏/特征维度,我们将特征定义为特定维度hi 。我们的分析着眼于给定变压器所有层的特定特征维度hi 。

我们发现离群特征强烈影响变压器的注意力和整体预测性能。虽然 13B 模型的每个 2048 个标记序列存在多达 150k 个离群值,但这些离群值特征是高度系统化的,并且仅代表最多 7 个唯一特征维度hi 。该分析的见解对于开发混合精度分解至关重要。我们的分析解释了零点量化的优点,以及为什么它们会随着使用混合精度分解以及小型模型与大型模型的量化性能而消失。

4.1 寻找离群特征对突发现象进行定量分析

的困难有两个方面。我们的目标是选择一小部分特征进行分析,使结果易于理解且不会过于复杂,同时捕获重要的概率和结构化模式。我们使用经验方法来找到这些约束。我们根据以下标准定义异常值:特征的大小至少为 6.0,影响至少 25% 的层,并影响至少 6% 的序列维度。

更正式地,给定具有 L 层和隐藏状态Xl $\in$ R序列维度和h特征维度的变换器,我们将特征定义为任何隐 $^{长\times高}$, l = 0...L 其中 s 是藏状态Xli中的特定维度hi 。我们跟踪维度hi 0 $\leqslant$ i $\leqslant$ h,其至少有一个大小为α $\geqslant$ 6的值,并且仅当这些异常值出现在至少 25% 的变压器层0中的相同特征维度hi中时,我们才会收集统计数据... L并且出现在所有隐藏状态Xl的所有序列维度s的至少 6% 中。由于特征异常值仅出现在注意力投射中

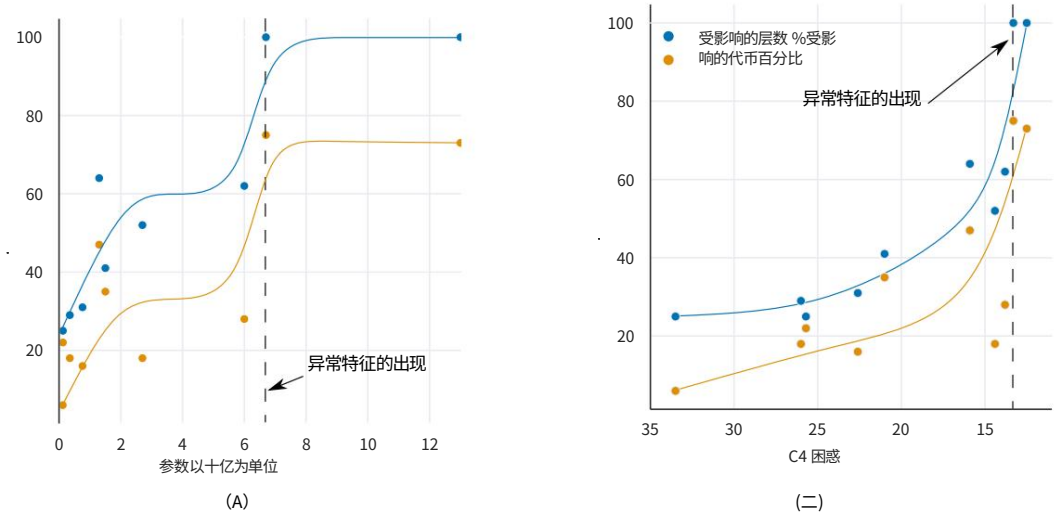

图 3:变压器中受 (a) 模型大小或 (b) C4 困惑度影响的大量异常值特征的层和所有序列维度的百分比。 (a) 和 (b) 的线条是 4 和 9 个线性段的 B 样条插值。一旦发生相移,所有层中以及所有序列维度的约 75% 都会出现异常值。 (a) 表明参数大小发生突然的相移,(b) 表明随着困惑度的降低逐渐发生指数相移。 (a) 中的明显转变与量化方法性能的突然下降同时发生。

（键/查询/值/输出)和前馈网络扩展层（第一个子层），在此分析中我们忽略注意函数和 FFN 收缩层（第二个子层）。

我们对这些阈值的推理如下。我们发现,使用混合精度分解,如果我们将任何大小为 6 或更大的特征视为离群特征,则困惑度退化就会停止。
对于受异常值影响的层数,我们发现异常值特征在大型模型中是系统性的:它们要么出现在大多数层中,要么根本不出现。另一方面,它们在小模型中是概率性的:它们有时出现在每个序列的某些层中。因此,我们设置了需要影响多少层才能检测异常值特征的阈值,从而将检测限制为具有 125M 参数的最小模型中的单个异常值。该阈值对应于至少25% 的转换器层受到相同特征维度中的异常值的影响。第二个最常见的异常值仅出现在单个层（2% 的层)中,表明这是一个合理的阈值。我们使用相同的过程来查找125M 模型中有多少序列维度受到异常值特征影响的阈值:异常值至少出现在序列维度的 6% 中。

我们测试模型的参数规模高达 13B。为了确保观察到的现象不是由于软件错误造成的,我们评估了在三种不同软件框架中训练的变压器。我们评估了四种使用 OpenAI 软件的 GPT-2 模型、五种使用 Fairseq（Ott 等人,2019)的 Meta AI 模型以及一种使用 Tensorflow-Mesh（Shazeer 等人,2018)的 EleutherAI 模型 GPT-J 。更多详细信息可以在附录 C 中找到。我们还在两种不同的推理软件框架中进行分析:Fairseq 和 Hugging Face Transformers（Wolf 等人,2019）。

4.2 衡量离群特征的影响

为了证明异常值特征对于注意力和预测性能至关重要,我们在将隐藏状态Xl输入到注意力投影层之前将异常值特征设置为零,然后将 top-1 softmax 概率与异常值的常规 softmax 概率进行比较。我们独立地对所有层执行此操作,这意味着我们转发常规 softmax 概率值,以避免级联错误并隔离异常特征造成的影响。如果我们删除离群特征维度（将它们设置为零)并通过变压器传播这些改变的隐藏状态,我们还会报告困惑度下降。作为对照,我们对随机非离群特征维度应用相同的过程,并注意注意力和困惑度退化。

我们的主要定量结果可以概括为四个要点。

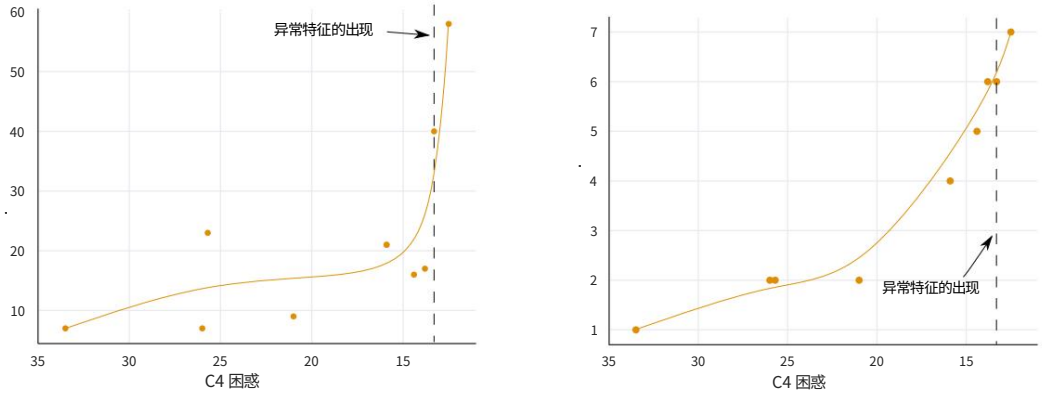

图 4:(a) 中最大离群值特征的中值大小表明离群值大小的突然变化。这似乎是量化方法出现后失败的主要原因。虽然离群值特征维度的数量仅与模型大小大致成正比,但 (b) 显示离群值的数量相对于所有分析模型的困惑度而言是严格单调的。线条是 9 个线性段的 B 样条插值。

(1) 当通过参数数量进行测量时,如图 3a 所示,随着受影响层的百分比从 65% 增加到 100%,变压器所有层上的大幅特征突然出现在 6B 和 6.7B 参数之间。受影响的序列维度数量从 35% 迅速增加到 75%。这种突然的转变与量化开始失败的点同时发生。

(2) 或者,当通过困惑度进行测量时,变压器所有层中大量特征的出现可以被视为根据困惑度递减的指数函数顺利出现,如图 3b 所示。这表明出现并不是突然的,并且通过研究较小模型中的指数趋势,我们也许能够在相移发生之前检测到出现的特征。这也表明,出现不仅与模型大小有关,还与困惑度有关,而困惑度与多个其他因素有关,例如使用的训练数据量和数据质量（Hoffmann et al., 2022;Henighan et al., 2020）。

(3) 一旦异常值特征出现在变压器的所有层中,中值异常值特征量值就会迅速增加,如图 4a 所示。大量异常值特征及其不对称分布破坏了 Int8 量化精度。这是量化方法从 6.7B 尺度开始失败的核心原因　量化分布的范围太大,导致大多数量化仓为空,小的量化值被量化为零,基本上消除了信息。我们假设,除了 Int8 推理之外,常规 16 位浮点训练也会由于超出 6.7B 范围的异常值而变得不稳定 - 如果乘以填充有大小为 60 的值的向量,很容易偶然超过最大16 位值 65535 。

(4)如图 4b 所示,离群值特征的数量相对于 C4 困惑度的降低严格单调增加,而与模型大小的关系是非单调的。这表明模型的复杂度而不仅仅是模型的大小决定了相移。我们假设模型大小只是实现涌现所需的众多协变量中的一个重要协变量。

这些异常值特征在相移发生后是高度系统化的。例如,对于序列长度为 2048 的 6.7B Transformer,我们发现整个Transformer 每个序列大约有 150k 个异常特征,但这些特征仅集中在 6 个不同的隐藏维度中。

这些异常值对于变压器性能至关重要。如果移除异常值,平均 top-1 softmax 概率会从大约 40% 降低到大约 20%,并且验证困惑度会增加600-1000%,即使最多有 7 个异常值特征维度。当我们删除 7 个随机特征维度时,top-1 概率仅下降 0.02-0.3%,而困惑度增加 0.1%。这凸显了这些特征维度的关键性质。这些异常特征的量化精度至关重要,因为即使很小的误差也会极大地影响模型性能。

4.3 量化性能解释

我们的分析表明,特定特征尺寸的异常值在大型变压器中普遍存在,并且这些特征尺寸对于变压器性能至关重要。由于行量化和向量量化缩放每个隐藏状态序列维度s （行） ,并且由于异常值出现在特征维度h （列)中,因此两种方法都无法有效处理这些异常值。这就是为什么absmax量化方法出现后很快就会失败。

然而,几乎所有异常值都具有严格的不对称分布:它们要么仅为正值,要么为负值（参见附录 C）。这使得零点量化对于这些离群值特别有效,因为零点量化是一种非对称量化方法,可将这些离群值缩放到完整的[－127, 127]范围。这解释了表 1 中我们的量化缩放基准的强劲性能。然而,在 13B 缩放下,由于累积的量化误差和离群值的快速增长,即使是零点量化也会失败,如图 4a 所示。

如果我们使用完整的 LLM.int8() 方法和混合精度分解,零点量化的优势就会消失,表明剩余的分解特征是对称的。

然而,向量方式仍然比行方式量化具有优势,这表明需要提高模型权重的量化精度来保持全精度预测性能。

5 相关工作

量化数据类型和转换器的量化有密切相关的工作,如下所述。附录 B 提供了有关卷积网络量化的进一步相关工作。

8 位数据类型。我们的工作研究围绕 Int8 数据类型的量化技术,因为它是目前 GPU 支持的唯一 8 位数据类型。其他常见数据类型是定点或浮点 8 位数据类型 (FP8)。这些数据类型通常具有符号位以及不同的指数和分数位组合。例如,此数据类型的常见变体有 5位用于指数,2 位用于分数（Wang 等人,2018;Sun 等人,2019;Cambier 等人， 2020;Mellempudi 等人,2019）)并且不使用缩放常数或使用零点缩放。这些数据类型对于大数值而言具有较大误差,因为它们只有 2 位用于分数,但对于小数值而言提供高精度。金等人。 (2022) 提供了关于何时某些定点指数/分数位宽度对于具有特定标准偏差的输入是最佳的分析。我们相信,与 Int8 数据类型相比,FP8 数据类型可提供卓越的性能,但目前 GPU 和 TPU 均不支持此数据类型。

语言模型中的异常特征。语言模型中的大幅异常值特征之前已经被研究过（Timkey 和 van Schijndel,2021; Bondarenko 等人,2021;Wei 等人,2022； Luo 等人,2021）。之前的工作证明了 Transformer 中异常值出现之间的理论关系以及它与层归一化和令牌频率分布的关系（Gao 等人,2019）。同样,Kovaleva 等人。 (2021) 将 BERT 模型族中异常值的出现归因于 LayerNorm,而 Puccetti 等人。 (2022) 根据经验表明,异常值的出现与训练分布中标记的频率有关。我们进一步扩展了这项工作,展示了自回归模型的规模如何与这些异常值特征的新兴属性相关,并展示了正确建模异常值对于有效量化的关键。

数十亿级变压器量化。有两种方法与我们的方法并行开发:nuQmm（Park 等人,2022)和 ZeroQuant（Yao 等人, 2022）。两者都使用相同的量化方案:group-w2ise 量化,它具有比向量量化更精细的量化归一化常数粒度。该方案提供了更高的量化精度,但也需要定制 CUDA 内核。 nuQmm 和 ZeroQuant 的目标都是加速推理并减少内存占用,而我们则专注于在8 位内存占用下保持预测性能。 nuQmm 和 ZeroQuant 评估的最大模型分别是 2.7B 和 20B 参数转换器。 ZeroQuant 实现了 20B 模型的 8 位量化的零退化性能。我们表明,我们的方法允许对多达 176B 参数的模型进行零退化量化。 nuQmm 和 ZeroQuant 都表明更精细的量化粒度可以成为量化大型模型的有效手段。这些方法与 LLM.int8()互补。另一项并行工作是 GLM-130B,它利用我们工作中的见解来实现零退化 8 位量化（Zeng 等人, 2022）。 GLM-130B 执行全 16 位精度矩阵乘法,并具有 8 位权重存储。

6 讨论和局限性

我们首次证明了数十亿个参数转换器可以量化为
Int8 并立即用于推理,不会降低性能。我们通过使用来实现这一点
我们通过大规模分析新兴的大规模特征来开发混合精度的见解
分解以在单独的 16 位矩阵乘法中隔离异常值特征。结合
通过向量量化产生我们的方法 LLM.int8(),我们凭经验证明可以
恢复具有多达 175B 参数的模型的完整推理性能。

我们工作的主要限制是我们的分析仅针对 Int8 数据类型,并且我们不
研究 8 位浮点 (FP8) 数据类型。由于当前的 GPU 和 TPU 不支持此数据
类型,我们相信这最好留给未来的工作。然而,我们也相信来自
Int8 数据类型将直接转换为 FP8 数据类型。另一个限制是我们只研究
具有多达 175B 参数的模型。虽然我们将 175B 模型量化为 Int8 而没有性能
退化,额外的新兴特性可能会在更大范围内破坏我们的量化方法。

第三个限制是我们不使用 Int8 乘法作为注意力函数。自从我们专注
是为了减少内存占用,并且注意力函数不使用任何参数,它是
不是严格需要的。然而,对该问题的初步探索表明,需要一个解决方案
除了我们在这里开发的量化方法之外,还有其他量化方法,我们将其留到未来的工作中。

最后一个限制是我们专注于推理,但不研究训练或微调。我们提供一个
附录 E 中对 Int8 微调和大规模训练的初步分析。 Int8 大规模训练需要
量化精度、训练速度和工程复杂性之间的复杂权衡
代表着一个非常困难的问题。我们再次将其留给未来的工作。

表 2:不同的硬件设置以及哪些方法可以以 16 位与 8 位精度运行。我们
可以看到,我们的 8 位方法使许多以前无法访问的模型变得可访问,在
特别是 OPT-175B/BLOOM。

			可运行的最大模型	
班级	硬件	显存	8位	16位
企业 8x A100		80GB	OPT-175B / 绽放 OPT-175B / 绽放	
企业 8x A100		40GB	OPT-175B / 绽放	OPT-66B
学术服务器 8x RTX 3090 24 GB			OPT-175B / 绽放	OPT-66B
学术台式机 4x RTX 3090 24 GB			OPT-66B	OPT-30B
付费云	Colab专业版	15GB	OPT-13B	GPT-J-6B
免费云	科拉布	12GB	T0/T5-11B	GPT-2 1.3B

7 更广泛的影响

我们工作的主要影响是能够访问以前无法容纳的大型模型
GPU内存。这使得以前由于有限而无法实现的研究和应用成为可能
GPU 内存,特别是对于资源最少的研究人员而言。型号/GPU 参见表 3
现在可以使用这些组合而不会降低性能。然而,我们的工作也
使资源丰富的组织能够在相同数量的 GPU 上为更多模型提供服务
GPU,这可能会增加资源丰富的组织和资源匮乏的组织之间的差距。

特别是,我们认为大型预训练模型的公开发布,例如最近的
开放式预训练 Transformers (OPT)（Zhang 等人,2022）,以及我们用于零次和少次提示的新 Int8 推理,将为学术机构带来不可能的新研究
之前由于资源限制。这种大型模型的广泛使用可能会
对社会产生难以预测的有益和有害影响。

致谢我们感谢 Ofir Press、Gabriel Ilharco、Daniel Jiang、Mitchell Wortsman、Ari
Holtzman、Mitchell Gordon 对这项工作草稿的反馈。我们感谢 JustHeuristic (Yozh)
和 Titus von Köller 寻求 Hugging Face Transformers 集成方面的帮助。

参考

Artetxe, M.、Bhosale, S.、Goyal, N.、Mihaylov, T.、Ott, M.、Shleifer, S.、Lin, XV、Du, J.、Iyer, S.、Pasunuru, R.等等人。（2021）。专家混合的高效大规模语言建模。 arXiv预印本 arXiv：2112.10684。

Bai, H.、Zhang, W.、Hou, L.、Shang, L.、Jin, J.、Jiang, X.、Liu, Q.、Lyu, MR 和 King, I. (2021)。Binarybert：突破 bert 量化的极限。 ArXiv，abs/2012.15701。

Bondarenko, Y.、Nagel, M. 和 Blankevoort, T. (2021)。了解并克服高效变压器量化的挑战。 arXiv 预印本 arXiv：2109.12948。

Brown, TB、Mann, B.、Ryder, N.、Subbiah, M.、Kaplan, J.、Dhariwal, P.、Neelakantan, A.、Shyam, P.、Sastry, G.、Askel, A. 等等人。 （2020）。语言模型是小样本学习者。 arXiv 预印本arXiv：2005.14165。

Cambier, L.、Bhiwandiwalla, A.、Gong, T.、Elibol, OH、Nekuii, M. 和 Tang, H. (2020)。用于深度神经网络低精度训练的移位和压缩 8 位浮点格式。第八届学习表征国际会议，ICLR 2020，埃塞俄比亚亚的斯亚贝巴， 2020 年 4 月 26-30 日。OpenReview.net。

Chen, J.、Gai, Y.、Yao, Z.、Mahoney, MW 和 Gonzalez, JE (2020)。用于深度神经网络低位宽训练的统计框架。神经信息处理系统的进展，33：883–894。

Choi, J.、Venkataramani, S.、Srinivasan, V.、Gopalakrishnan, K.、Wang, Z. 和 Chuang, P. (2019)。准确高效的 2 位量化神经网络。 Talwalkar, A.、Smith, V. 和 Zaharia, M.，编辑，2019 年机器学习和系统论文集，MLSys 2019，美国加利福尼亚州斯坦福，2019 年3 月 31 日至 4 月 2 日。mlsys.org。

Courbariaux, M. 和 Bengio, Y. (2016)。 Binarynet：使用权重和训练深度神经网络激活值限制为 +1 或 -1。 CoRR，abs/1602.02830。

Courbariaux, M.、Bengio, Y. 和 David, J. (2015)。 Binaryconnect：在传播过程中使用二进制权重训练深度神经网络。 Cortes, C.、Lawrence, ND、Lee, DD、Sugiyama, M. 和 Garnett, R.，编辑，神经信息处理系统进展 28：2015 年神经信息处理系统年会，2015 年 12 月 7-12 日，加拿大魁北克省蒙特利尔，第 3123-3131 页。

Courbariaux, M.、Bengio, Y. 和 David, J.-P。 （2014）。用低训练深度神经网络精度乘法。 arXiv 预印本 arXiv：1412.7024。

Dettmers, T.、Lewis, M.、Shleifer, S. 和 Zettlemoyer, L. (2022)。通过块方式的 8 位优化器量化。第九届国际学习表征会议，ICLR。

Devlin, J.、Chang, M.-W.、Lee, K. 和 Toutanova, K. (2018)。 Bert：深度双向的预训练用于语言理解的变压器。 arXiv 预印本 arXiv：1810.04805。

Dong, Z.、Yao, Z.、Gholami, A.、Mahoney, MW 和 Keutzer, K. (2019)。 Hawq：具有混合精度的神经网络的Hessian 感知量化。 IEEE/CVF 国际计算机视觉会议记录，第 293-302 页。

Esser, SK、McKinstry, JL、Bablani, D.、Appuswamy, R. 和 Modha, DS (2019)。学到的步骤尺寸量化。 arXiv 预印本 arXiv：1902.08153。

Fan, A.、Stock, P.、Graham, B.、Grave, E.、Gribonval, R.、Jegou, H. 和 Joulin, A. (2020)。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。

高J.、何大、谭X.、秦T.、王L.、刘T.-Y.。 （2019）。训练自然语言生成模型中的表示退化问题。 arXiv 预印本 arXiv：1907.12009。

高，L.，托，J.，比德曼，S.，布莱克，S.，迪波菲，A.，福斯特，C.，戈尔丁，L.，许，J.，麦克唐纳，K.，穆恩尼霍夫，N.， Phang, J.、Reynolds, L.、Tang, E.、Thite, A.、Wang, B.、Wang, K. 和 Zou, A.（2021）。少镜头语言模型评估的框架。

Gholami, A.、Kim, S.、Dong, Z.、Yao, Z.、Mahoney, MW 和 Keutzer, K. (2021)。有效神经网络推理的量化方法的调查。 arXiv 预印本 arXiv:2103.13630。

Gokaslan, A. 和 Cohen, V. (2019)。 Openwebtext 语料库。网址http://Skylion007。 github。 io/OpenWebTextCorpus.

龚瑞、刘新、江生、李天、胡平、林静、余芳、严静 (2019)。可微分软量化:桥接全精度和低位神经网络。 2019 IEEE/CVF 国际计算机视觉会议,ICCV 2019,韩国首尔（韩国）， 2019 年 10 月 27 日至 11 月 2 日,第 4851–4860 页。 IEEE。

Henighan, T.、Kaplan, J.、Katz, M.、Chen, M.、Hesse, C.、Jackson, J.、Jun, H.、Brown, TB、Dhariwal, P.、Gray, S.等等人。 （2020）。自回归生成模型的缩放定律。 arXiv 预印本arXiv:2010.14701。

霍夫曼,J.,博尔若,S.,门施,A.,布哈茨卡亚,E.,蔡,T.,卢瑟福,E.,卡萨斯,D.d. L.、Hendricks, LA、Welbl, J.、Clark, A. 等。 （2022）。训练计算优化的大型语言模型。 arXiv 预印本 arXiv:2203.15556。

Ilharco, G.、Ilharco, C.、Turc, I.、Dettmers, T.、Ferreira, F. 和 Lee, K. (2020)。高性能自然语言处理。 2020 年自然语言处理经验方法会议论文集:教程摘要,第 24-27 页,在线。计算语言学协会。

Jin, Q.、Ren, J.、Zhang, R.、Hanumante, S.、Li, Z.、Chen, Z.、Wang, Y.、Yang, K. 和 Tulyakov, S. (2022) 。 F8net:用于网络量化的定点 8 位乘法。 arXiv 预印本arXiv:2202.05239。

Khudia, D.、Huang, J.、Basu, P.、Deng, S.、Liu, H.、Park, J. 和 Smelyanskiy, M. (2021)。 Fbgemm:实现高性能低精度深度学习推理。 arXiv 预印本 arXiv:2101.05615。

Kovaleva, O.、Kulshreshtha, S.、Rogers, A. 和 Rumshisky, A. (2021)。 Bert 克星:扰乱变压器的异常维度。 arXiv 预印本 arXiv:2105.06990。

李瑞、王玉、梁芳、秦红、严静、范瑞（2019）。用于对象检测的完全量化网络。 IEEE 计算机视觉和模式识别会议，CVPR 2019,美国加利福尼亚州长滩,2019 年 6 月 16-20 日,第 2810-2819 页。计算机视觉基金会/IEEE。

林 Y.、李 Y.、刘 T.、肖 T.、刘 T. 和朱 J. (2020)。迈向完全 8 位整数推理变压器模型。 arXiv 预印本 arXiv:2009.08034。

Liu, Y.、Ott, M.、Goyal, N.、Du, J.、Joshi, M.、Chen, D.、Levy, O.、Lewis, M.、Zettlemoyer, L. 和 Stoyanov, V. （2019）。 Roberta:一种稳健优化的 bert 预训练方法。 arXiv 预印本arXiv:1907.11692。

罗 Z.、库尔米泽夫 A. 和毛 X. (2021)。位置伪影通过掩码语言模型嵌入传播。计算语言学协会第 59 届年会和第 11 届自然语言处理国际联合会议论文集（第 1 卷:长论文）,第 5312-5327 页,在线。计算语言学协会。

Machácek, M. 和 Bojar, O. (2014)。 wmt14 指标共享任务的结果。第九届统计机器翻译研讨会论文集,第 293-301 页。

Mellempudi, N.、Srinivasan, S.、Das, D. 和 Kaul, B. (2019)。 8 位混合精度训练浮点。 CoRR,abs/1905.12334。

内格尔,S.（2016）。抄送新闻。

Ott, M.、Edunov, S.、Baevski, A.、Fan, A.、Gross, S.、Ng, N.、Grangier, D. 和 Auli, M. (2019)。 fairseq:一个快速、可扩展的序列建模工具包。 arXiv 预印本 arXiv:1904.01038。

Ott, M.、Edunov, S.、Grangier, D. 和 Auli, M. (2018)。缩放神经机器翻译。 arXiv 预印本 arXiv:1806.00187。

Park, G.、Park, B.、Kwon, SJ、Kim, B.、Lee, Y. 和 Lee, D. (2022)。 nuqmm：量化 matmul，用于大规模生成语言模型的高效推理。 arXiv 预印本 arXiv：2206.09557。

Puccetti, G.、Rogers, A.、Drozd, A. 和 Dell Orletta, F. (2022)。干扰变压器的异常尺寸是由频率驱动的。 arXiv 预印本 arXiv：2205.11380。

秦 H.、龚 R.、刘 X.、白 X.、宋 J. 和 Sebe, N. (2020)。二元神经网络：一项调查。 CoRR，abs/2004.03333。

Radford, A.、Wu, J.、Child, R.、Luan, D.、Amodei, D. 和 Sutskever, I. (2019)。语言模型是无监督的多任务学习者。 OpenAI 博客，1(8):9。

Raffel, C.、Shazeer, N.、Roberts, A.、Lee, K.、Narang, S.、Matena, M.、Zhou, Y.、Li, W. 和 Liu, PJ （2019）。使用统一的文本到文本转换器探索迁移学习的局限性。 arXiv预印本 arXiv：1910.10683。

Rastegari, M.、Ordonez, V.、Redmon, J. 和 Farhadi, A. (2016)。 Xnor-net：使用二元卷积神经网络的 Imagenet 分类。 Leibe, B.、Matas, J.、Sebe, N. 和 Welling, M.，编辑，计算机视觉 - ECCV 2016 - 第 14 届欧洲会议，荷兰阿姆斯特丹， 2016 年 10 月 11-14 日，会议记录，第四部分，《计算机科学讲义》第 9908 卷，第 525-542 页。施普林格。

Sennrich, R.、Haddow, B. 和 Birch, A. (2016)。 wmt 16 的爱丁堡神经机器翻译系统。arXiv 预印本 arXiv:1606.02891。

Shazeer, N.、Cheng, Y.、Parmar, N.、Tran, D.、Vaswani, A.、Koanantakool, P.、Hawkins, P.、Lee, H.、Hong, M.、Young, C. ，等人。 （2018）。 Mesh-tensorflow：超级计算机的深度学习。神经信息处理系统的进展，31。

Shen, S.、Dong, Z.、Ye, J.、Ma, L.、Yao, Z.、Gholami, A.、Mahoney, MW 和 Keutzer, K. (2020)。 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。

Sun, X.、Choi, J.、Chen, C.、Wang, N.、Venkataramani, S.、Srinivasan, V.、Cui, X.、Zhang, W. 和 Gopalakrishnan, K. (2019)。深度神经网络的混合 8 位浮点 (HFP8) 训练和推理。 Wallach, HM、Larochelle, H.、Beygelzimer, A.、d Alché-Buc, F.、Fox, EB 和 Garnett, R.，编辑，神经信息处理系统进展 32：神经信息处理年会Systems 2019，NeurIPS 2019，2019 年 12 月 8-14 日，加拿大不列颠哥伦比亚省温哥华，第 4901-4910 页。

Timkey, W. 和 van Schijndel, M. (2021)。只吠不咬：Transformer语言模型中的流氓维度模糊了表征质量。 arXiv 预印本 arXiv：2109.04404。

Trinh, TH 和 Le, QV (2018)。一种简单的常识推理方法。 arXiv 预印本 arXiv：1806.02847。

Vaswani, A.、Shazeer, N.、Parmar, N.、Uszkoreit, J.、Jones, L.、Gomez, AN、Kaiser, L. 和 波洛苏欣，I.（2017）。您所需要的就是关注。 arXiv 预印本 arXiv：1706.03762。

Wang, N.、Choi, J.、Brand, D.、Chen, C. 和 Gopalakrishnan, K. (2018)。使用 8 位浮点数训练深度神经网络。 Bengio, S.、Wallach, HM、Larochelle, H.、Grauman, K.、Cesa-Bianchi, N. 和 Garnett, R.，编辑，神经信息处理系统进展 31：神经信息处理系统年会2018，NeurIPS 2018，2018 年 12 月3-8 日，加拿大蒙特利尔，第 7686-7695 页。

魏X.、张Y.、张X.、龚R.、张S.、张Q.、于F.和刘X. (2022)。异常值抑制：突破低位转换器语言模型的极限。 arXiv 预印本arXiv：2209.13325。

Wenzek, G.、Lachaux, M.-A.、Conneau, A.、Chaudhary, V.、Guzmán, F.、Joulin, A. 和 Grave, E. （2020）。 CCNet：从网络爬行数据中提取高质量的单语数据集。第 12 届语言资源和评估会议论文集，第 4003-4012 页，法国马赛。
欧洲语言资源协会。

Wolf, T.、Debut, L.、Sanh, V.、Chaumond, J.、Delangue, C.、Moi, A.、Cistac, P.、Rault, T.、Louf, R.、Funtowicz, M.,等人。 （2019）。 Huggingface 的变形金刚：最先进的自然语言处理。 arXiv 预印本 arXiv：1910.03771。

Wu, H.、Judd, P.、Zhang, X.、Isaev, M. 和 Micikevicius, P. (2020)。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。

Yao, Z.、Aminabadi, RY、Zhang, M.、Wu, X.、Li, C. 和 He, Y. (2022)。 Zeroquant：针对大型 Transformer 的高效且经济实惠的训练后量化。 arXiv 预印本 arXiv：2206.01861。

姚Z.、董Z.、郑Z.、古拉米A.、于J.、谭E.、王L.、黄Q.、王Y.、马奥尼M. 、等人。 （2021）。 Hawq-v3：二元神经网络量化。国际机器学习会议，第 11875–11886 页。 PMLR。

Zafrir, O.、Boudoukh, G.、Izsak, P. 和 Wasserblat, M. (2019)。 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。

曾A.、刘X.、杜Z.、王Z.、赖华.、丁明.、杨Z.、徐Y.、郑W.、夏X.、等人。
（2022）。 Glm-130b：开放的双语预训练模型。 arXiv 预印本 arXiv：2210.02414。

张 D.、杨 J.、叶 D. 和华 G. (2018)。 Lq-nets：高度准确且紧凑的深度神经网络的学习量化。欧洲计算机视觉会议(ECCV)会议记录，第 365-382 页。

张，S.，罗勒，S.，戈亚尔，N.，Artetxe，M.，陈，M.，陈，S.，德万，C.，迪亚布，M.，李，X.，林，XV，等等人。 （2022）。 Opt：打开预先训练的 Transformer 语言模型。 arXiv 预印本arXiv：2205.01068。

张文、侯丽、尹勇、尚丽、陈X、蒋X、刘Q. (2020)。 Ternarybert：蒸馏感知的超低位 bert。在 EMNLP 中。

赵 C.、华 T.、沉 Y.、楼 Q. 和金 H. (2021)。自动混合精度量化
寻找伯特。 arXiv 预印本 arXiv：2112.14938。

Zhu, C.、Han, S.、Mao, H. 和 Dally, WJ (2017)。训练好的三元量化。第五届学习表征国际会议，ICLR 2017，法国土伦，2017 年 4 月 24-26 日，会议记录。 OpenReview.net。

Zhu, Y.、Kiros, R.、Zemel, R.、Salakhutdinov, R.、Urtasun, R.、Torralba, A. 和 Fidler, S. (2015)。
协调书籍和电影：通过观看电影和阅读书籍来实现故事式的视觉解释。 IEEE 国际计算机视觉会议记录，第19-27 页。

清单

检查表遵循参考文献。请仔细阅读清单指南，了解如何回答这些问题的信息。对于每个问题，将默认的[TODO]更改为[是] 、[否] ，强烈建议您通过引用[N/A]来说明您的答案的理由。论文的适当部分或提供简短的内联描述。例如： 或者

- 您是否包含代码和数据集的许可证？ [是]参见??部分。 ·您是否包含代码和数据集的许可证？

[否]代码和数据是
所有权。

- 您是否包含代码和数据集的许可证？ [不适用]

请不要修改问题，仅使用提供的宏作为答案。请注意，核对表部分不计入页数限制。在您的论文中，请删除此说明块，只保留上面的清单部分标题以及下面的问题/答案。

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] (b) 您是否描述
了您工作的局限性？ [是]请参阅限制部分 (c)您是否讨论过您的工作对任何潜在的负面社会影
响?[是]请参阅更广泛的影响部分 (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]是的,我们相信我们的工作符合这些准则。

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？ [不适用] (b) 您是否提供了所有理论结果的完整证明？ [不适用]

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
将在补充材料中包含我们的代码。 (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]请参阅实验设置部分 (c)您
是否报告了误差线（例如,关于之后的随机种子)多次运行实验）？ [否]我们的实验对于每个模型都是确
定性的。我们不是多次运行相同的模型,而是以不同的规模运行
多个模型。

我们无法计算这些实验的误差线。
(d)您是否包含了计算总量和使用的资源类型（例如GPU 类型、内部集群或云提供商）？ [是]请参阅实
验设置第 4 节。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
作品使用现有资产,您是否引用了创作者？ [是]参见实验

设置部分
(b)您是否提到资产的许可？ [否]该许可证适用于我们使用的所有资产。可以轻松查找各个许可证。

(c)您是否在补充材料中或以 URL 的形式包含了任何新资产？ [不适用]
我们只使用现有的数据集。
(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]

A 与 16 位精度相比的内存使用情况

表3比较了不同开源的16位推理和LLM.int8()的内存占用
楷模。我们可以看到,LLM.int8() 允许运行最大的开源模型 OPT-175B 和
BLOOM-176B 位于配备消费级 GPU 的单个节点上。

表 3:不同的硬件设置以及哪些方法可以以 16 位与 8 位精度运行。我们
可以看到,我们的 8 位方法使许多以前无法访问的模型变得可访问,在
特别是 OPT-175B/BLOOM。

			可运行的最大模型	
班级	硬件	显存	8位	16位
企业 8x A100		80GB	OPT-175B / 绽放 OPT-175B / 绽放	
企业 8x A100		40GB	OPT-175B / 绽放	OPT-66B
学术服务器 8x RTX 3090 24 GB			OPT-175B / 绽放	OPT-66B
学术台式机 4x RTX 3090 24 GB			OPT-66B	OPT-30B
付费云	Colab专业版	15GB	OPT-13B	GPT-J-6B
免费云	科拉布	12GB	T0/T5-11B	GPT-2 1.3B

B 附加相关工作

参数少于 1B 的变压器量化 变压器量化
一直专注于数十亿参数掩码语言模型（MLM），包括 BERT（Devlin等人,2018)和 RoBERTa（Liu 等人,2019）。 8
位 BERT/RoBERTa 的版本包括
Q8BERT（Zafir 等人,2019）、QBERT（Shen 等人,2020）、乘积量化与量化
噪声（Fan 等人,2020）、TernaryBERT（Zhang 等人,2020)和 BinaryBERT（Bai 等人,2021）。
赵等人的作品。 (2021) 执行量化和修剪。所有这些模型都需要
量化感知微调或训练后量化,使模型在低精度下可用。
与我们的方法相比,该模型可以直接使用而不会降低性能。

如果将矩阵乘法视为 1x1 卷积,则向量量化相当于
卷积的逐通道量化与行量化相结合（Khudia 等人,2021）。
对于矩阵乘法,Wu 等人使用了该方法。 (2020) 用于 BERT 尺寸的变压器 (350M
参数），而我们是第一个研究自回归和大规模向量量化的人
楷模。除 BERT 之外,我们所知的唯一量化 Transformer 的工作是
陈等人。 (2020),它使用训练后量化和前向零点量化
向后传递中的传递和零点行量化。然而,这项工作仍然是为了
十亿个参数转换器。我们与零点量化和行量化进行比较
我们的评估不需要训练后量化。

低位宽和卷积网络量化使用少于 8 位数据的工作
类型通常用于卷积网络 (CNN),以减少其内存占用并增加
移动设备的推理速度,同时最大限度地减少模型退化。针对不同的方法
位宽已被研究:1 位方法（Courbariaux 和 Bengio,2016 年;Rastegari 等人,2016 年;
Courbariaux 等人,2015）、2 至 3 位（Zhu 等人,2017;Choi 等人,2019）、4 位（Li 等人,2019）、
更多位（Courbariaux 等人,2014），或可变数量的位（Gong 等人,2019）。对于额外的
相关工作,请参见Qin等人的调查。 （2020）。虽然我们认为低于 8 位宽度
由于数十亿级的 Transformer 可能会出现一些性能下降,因此我们专注于 8 位
Transformer 不会降低性能,并且可以从常用的 GPU 中受益
通过 Int8 张量核心加速推理。

专注于卷积网络量化的另一项工作是学习对
量化过程以改善量化误差。例如,使用Hessian信息（Dong
等人,2019）、步长量化（Esser 等人,2019）、软量化（Gong 等人,2019）、通过线性规划优化的混合精度（ Yao 等人，
2021)以及其他学习方法量化
方法（Zhang 等人,2018;Gholami 等人,2021）。

表 4:在至少 25% 的数据中出现的强度至少为 6 的异常值的汇总统计
所有层和所有序列维度的至少 6%。我们可以看到C4验证越低
越困惑,出现的异常值就越多。离群值通常是单方面的,其四分位数为
最大范围显示异常值震级比其他最大震级大 3-20 倍
特征维度,通常范围为[-3.5, 3.5]。随着规模的扩大,异常值变得
在变压器的所有层中越来越常见,并且它们几乎以所有顺序出现
方面。当所有层中出现相同的异常值时,在 6.7B 参数处发生相变
大约 75% 的序列维度 (SDim) 处于相同的特征维度。尽管只有
异常值约占所有特征的 0.1%,对于较大的 softmax 概率至关重要。这
如果移除异常值,平均 top-1 softmax 概率会缩小约 20%。因为异常值
在序列维度s 上大多具有不对称分布,这些离群维度
破坏对称的绝对最大量化并支持非对称的零点量化。这解释了
我们验证困惑度分析的结果。这些观察结果似乎具有普遍性,因为它们
发生在不同软件框架（fairseq、OpenAI、Tensorflow-mesh)中训练的模型上,并且
它们出现在不同的推理框架中（fairseq、Hugging Face Transformers）。这些异常值
对于变压器架构的轻微变化（旋转嵌入、嵌入
范数、残差缩放、不同的初始化）。

	异常值		频率	Top-1 softmax p	
模型 PPL↓ 参数计数 1 面层 SDims			四分位数	带离群值	无离群值
GPT2 33.5 117M GPT2 26.0		1	25% 6% (-8, -7, -6) 45% 29% 18% (6, 7, 8) 45% 25%		19%
345M FSEQ 25.7 125M GPT2	1	1	22% (-40, -23, -11) 32% 31% 16% (-9, -6, 9) 41%		19%
22.6 762M GPT2 21.0 1.5B	2	2	41% 35% (-11, -9, -7) 41% 64% 47% (-33, -21, -11)		24%
FSEQ 15.9 1.3B FSEQ 14.4	2	0	39% 52% 18% (-25 , -16, -9) 45% 62% 28% (-21,		18%
2.7B GPT-J 13.8 6.0B	2	1	-17, -14) 55% 100% 75% (-44, -40, -35) 100% 73%		25%
	2	3	(-63, -58, -45)		15%
	4	5			13%
	5 6	6			10%
FSEQ 13.3 6.7B FSEQ 12.5	6	6		35%	13%
13B	7	6		37%	16%

C 详细异常值特征数据

表 4 提供了我们的异常值特征分析的表格数据。我们提供最多的四分位数
每个变压器中的公共异常值以及单侧异常值的数量,即具有
不跨越零的不对称分布。

D 推理加速和减速

D.1 矩阵乘法基准

虽然我们的工作重点是内存效率以使模型可访问,但 Int8 方法也经常
用于加速推理。我们发现量化和分解开销很大，
而 Int8 矩阵乘法本身只有在整个 GPU 充分饱和的情况下才会产生优势，
这仅适用于大矩阵乘法。这种情况仅发生在具有模型维度的法学硕士中
4096 或更大。

原始矩阵乘法和量化开销的详细基准如表 5 所示。
我们看到 cuBLASLt 中的原始 Int8 矩阵乘法开始比 cuBLAS 快两倍
模型尺寸为 5140（隐藏尺寸 20560）。如果输入需要量化并且输出需要反量化
– 如果不是整个变压器都在 Int8 中完成,那么这是一个严格的要求 – 那么与
16 位在模型大小为 5140 时降低至 1.6 倍。模型大小为 2560 或更小的模型速度会变慢
向下。添加混合精度分解会进一步减慢推理速度,因此只有 13B 和 175B
模型有加速。

通过针对混合精度优化 CUDA 内核,这些数字可以得到显着改善
分解。然而,我们也看到现有的自定义 CUDA 内核比以前快得多
我们使用默认的 PyTorch 和 NVIDIA 提供的内核进行量化,这会减慢所有矩阵的速度
乘法（175B 型号除外）。

表 5:与第一个隐藏层的 16 位矩阵乘法相比的推理加速
不同尺寸的 GPT-3 变压器的前馈。隐藏尺寸是模型的 4 倍
方面。无开销加速的 8 位假设不进行量化或反量化
执行。小于 1.0 倍的数字表示速度放缓。 Int8矩阵乘法加速
仅适用于具有大模型和隐藏维度的模型。

GPT-3尺寸 型号尺寸	小号 中号 大号 XL 2.7B 6.7B 13B 768 1024 1536 2048 2560 4096 5140 12288	175B
FP16 位基线	1.00x 1.00x 0.99x	1.00x 1.00x 1.00x 1.00x 1.00x 1.00x
Int8 无开销	1.08x	1.43x 1.61x 1.63x 1.67x 2.13x 2.29x
Absmax PyTorch+NVIDIA 0.25x 0.24x 向量方式 PyTorch+NVIDIA 0.21x 0.22x 向量方式0.43x 0.49x LLM.int8()（向量方式+解压 缩） 0.14x 0.20x		0.36x 0.45x 0.53x 0.70x 0.96x 1.50x 0.33x 0.41x 0.50x 0.65x 0.91x 1.50x 0.74x 0.91x 0.94x 1.18x 1.59x 2.00x 0.36x 0.51x 0.64x 0.86x 1.22x 1.81x

D.2 端到端基准测试

除了矩阵乘法基准之外,我们还在Hugging Face 中测试了 BLOOM-176B 的端到端推理速度。 Hugging Face 使用带有缓存注意力值的优化实现。
由于这种类型的推理是分布式的,因此依赖于通信,我们期望
由于 Int8 推理导致的整体加速和减速较小,因为整体的很大一部分
推理运行时是固定的通信开销。

我们对 16 位进行基准测试,并尝试在 Int8 的情况下使用更大批量大小或更少 GPU 的设置
推断,因为我们可以在更少的设备上安装更大的模型。我们可以看到我们的基准测试结果
表 6 中。整体 Int8 推理稍微慢一些,但接近每个令牌的毫秒延迟
与 16 位推理相比。

表 6:对用于运行 BLOOM-176B 模型的几种类型推理的 GPU 数量的消融研究。我们将量化的 BLOOM-176B 模型使用的 GPU 数量放在一起进行比较
与原生 BLOOM-176B 型号一起使用。我们还报告每个令牌的生成速度（以毫秒为单位）
对于不同的批量大小。我们将我们的方法集成到变压器（Wolf et al., 2019)中
通过 HuggingFace 的加速库来处理多 GPU 推理。我们的方法达到了
通过适应比本机模型更少的 GPU,获得与本机模型相似的性能。

批量大小硬件	1	8	32
bfloat16 基线 8xA100 80GB 239 32 9.94			
法学硕士.int8() 8xA100 80GB 253 34 10.44			
法学硕士.int8() 4xA100 80GB 246 33 9.40			
法学硕士.int8() 3xA100 80GB 247 33 9.11			

E 培训结果

我们在各种训练设置上测试 Int8 训练,并与 32 位基线进行比较。我们分别测试
用于运行带有 8 位前馈网络和不带 8 位线性的变压器的设置
注意力层中的投影以及 8 位注意力本身的投影并与 32 位进行比较
表现。我们在 RoBERTa 语料库的一部分上测试了两项任务 (1) 语言建模,包括
书籍（Zhu 等人,2015 年）、CC-News（Nagel,2016 年）、OpenWebText（Gokaslan 和 Cohen,2019 年)以及
CC-Stories（Trinh 和 Le,2018）； (2) 神经机器翻译 (NMT)（Ott 等人,2018）
WMT14+WMT16（Machácek 和 Bojar,2014;Sennrich 等,2016）。

结果如表7和表8所示。我们可以看到,对于训练,使用attention
Int8 数据类型的线性投影和向量量化会导致 NMT 退化
适用于 1.1B 语言模型,但不适用于 209M 语言模型。结果略有改善
如果使用混合精度分解但不足以恢复大多数性能
案例。这表明使用 8 位 FFN 层进行训练非常简单,而其他层则需要

与 Int8 相比的其他技术或不同的数据类型,可以在没有性能的情况下进行大规模 8 位训练
降解。

表 7:小型和大规模语言建模的初步结果。认真做8位注意力
降低性能并且性能无法通过混合精度分解完全恢复。
虽然小规模语言模型接近 8 位 FFN 和 8 位的基准性能
注意力层中的线性项目的性能会大规模下降。

参数	是8位的 FFN	线性注意力分解	PPL
209M			0% 16.74
209M			0% 16.77
209M			0% 16.83
209M			2% 16.78
209M			5% 16.77
209M			10% 16.80
209M			2% 24.33
209M			5% 20.00
209M			10% 19.00
1.1B			0% 9.99
1.1B			0% 9.93
1.1B			0% 10.52
1.1B			1% 10.41

F 微调结果

我们还在 RoBERTa 上测试了 8 位微调,在 GLUE 上进行了大型微调。我们运行两种不同的设置:
(1) 我们与其他 Int8 方法进行比较,(2) 我们与 8 位方法比较微调的退化
FFN 层以及 8 位注意力投影层与 32 位相比。我们用 5 个随机数进行微调
种子并报告中值表现。

表 9 与之前不同的 8 位微调方法进行了比较,并表明向量方式
量化改进了其他方法。表 10 显示了 FFN 和/或线性的性能
8 位注意力投影以及使用混合精度分解时的改进。
我们发现 8 位 FFN 层不会导致性能下降,而 8 位注意力线性投影会导致性能下降
如果不与混合精度分解相结合,则会导致退化,其中至少前 2%
幅度维度以 16 位而不是 8 位计算。这些结果凸显了关键
如果不想降低性能,混合精度分解在微调中的作用。

表 8:WMT14+16 的 8 位 FFN 和线性注意层的神经机器翻译结果。
Decomp 表示以 16 位而不是 8 位计算的百分比。 BLEU 分数是
三个随机种子的中位数。

是8位的 FFN	线性分解	BLEU
		0% 28.9
		0% 28.8
		0% 不稳定
		2% 28.0
		5% 27.6
		10% 27.5

表 9:8 位前馈层量化方法的 GLUE 微调结果,而
其余为 16 位。不使用混合精度分解。我们可以看到向量量化
在基线上进行改进。

方法	MNLI QNLI QQP RTE SST-2 MRPC CoLA STS-B 平均值						
32 位基线 32 位复	90.4	94.9 92.2 84.5 96.4 94.8 92.3 85.4	90.1	67.4	93.0	88.61
制	90.3	96.6	90.4	68.8	92.0	88.83
Q-BERT（Shen 等人,2020)87.8 Q8BERT（Zafir		93.0 90.6 84.7 94.8 93.0 90.1 84.8	88.2	65.1	91.1	86.91
等人,2019)85.6 PSQ（Chen 等人,2020)89.9		94.7 94.5 92.0 86.8 96.2	89.7	65.0	91.1	86.75
			90.4	67.5	91.9	88.65
向量方式	90.2	94.7 92.3 85.4 96.4	91.0	68.6	91.9	88.81

表 10:8 位前馈网络 (FFN) 和 GLUE 线性注意层的细分。
分数是 5 个随机种子的中位数。 Decomp表示分解成的百分比
16 位矩阵乘法。与推理相比,微调似乎需要更高的分解
如果线性注意力层也转换为 8 位,则为百分比。

是8位的							
FFN 线性分解 MNLI QNLI QQP RTE SST-2 MRPC CoLA STS-B MEAN							
0%	90.4	94.9	92.2 84.5 96.4 92.3 85.4	90.1	67.4	93.0	88.6
0%	90.2	94.7	96.4 92.2 84.1 96.2 92.2	91.0	68.6	91.9	88.8
0%	90.2	94.4	83.0 96.2 92.2 85.9 96.7	89.7	63.6	91.6	87.7
1%	90.0	94.6	92.2 86.3 96.4	89.7	65.8	91.8	87.9
2%	90.0	94.5		90.4	68.0	91.9	88.7
3%	90.0	94.6		90.2	68.3	91.8	88.7