OmniQuant-目前最优的LLM PTQ量化算法

单位: OpenGVLab, 上海AI Lab, 香港大学
URL: https://readpaper.com/pdf-annotate/note?pdfId=4816122778868514817

研究动机： LLM的优秀的PTQ和QAT方法主要有GPTQ和LLM-QAT。GPTQ（Frantar等人，2022年）可以在单个A100 GPU上使用128个样本在一小时内完成LLaMA-13B的量化，而LLM-QAT（Liu等人，2023a）需要100k个样本和数百个GPU小时。这**引导我们来到一个核心问题：我们能否在保持PTQ的时间和数据效率的同时，达到QAT的性能?**

Abstract

LLMs已经彻底改变了自然语言处理任务。然而，它们的实际部署受到其巨大的内存和计算需求的阻碍。尽管最近的后训练量化（PTQ）方法在减少LLM的内存占用和提高计算效率方面取得了有效成果，但它们手工制作的量化参数导致了性能低下，并且无法处理极低比特量化。为了解决这个问题，我们引入了一种全方位校准量化（OmniQuant）技术。它在保持PTQ的计算效率的同时，在多样化的量化设置中取得了良好的性能，通过有效优化各种量化参数。OmniQuant包括两个创新组件，包括可学习的权重裁剪（LWC）和可学习的等效变换（LET）。LWC通过优化裁剪阈值来调节权重的极端值。同时，LET通过可学习的等效变换来处理激活的异常值，将量化的挑战从激活转移到权重。在可微分框架内使用分块误差最小化，OmniQuant可以有效地优化仅权重和权重-激活量化的量化过程。例如，大小为7-70B的LLaMA-2模型家族可以在1-16小时内使用128个样本在单个A100-40G GPU上通过OmniQuant处理。广泛的实验验证了OmniQuant在多样化的量化配置（如W4A4、W6A6、W4A16、W3A16和W2A16）中的卓越性能。此外，OmniQuant在指令调整模型中展示了有效性，并在真实设备上显著提高了推理速度和内存减少。代码可在

https://github.com/OpenGVLab/OmniQuant获取。

2. Related work

2.1 量化方法

- QAT通过在训练期间模拟量化来保持性能，但其训练成本使其不适合LLM。
- PTQ技术，如AdaRound和BRECQ，使用梯度优化来确定最佳舍入，但对于更大的模型来说，**调整所有权重是非常耗时的** 。

因此，大多数LLM量化方法优先选择无需训练的PTQ，但这也限制了模型在低比特情况下的性能。我们的目标是在LLM量化中整合梯度更新，模仿QAT的方法，同时保持PTQ的效率。

2.2 LLM的量化

权重量化 。权重量化专注于将权重转换为低比特值。例如，GPTQ使用块状重建进行3/4比特量化。SpQR（Dettmers等人，2023b）、OWQ（Lee等人，2023）和AWQ（Lin等人，2023）强调与高幅度激活相关的权重的重要性。因此，SpQR和OWQ采用混合精度量化来保护重要权重，而AWQ选择通道级缩放以避免混合精度的硬件效率低下。Qlora（Dettmers等人，2023a）和INT2.1（Chee等人，2023）通过参数高效微调恢复量化模型的能力。**与之相反，我们的方法直接增强了量化过程，使OmniQuant与Qlora和INT2.1相辅相成。**

权重-激活量化 。权重-激活量化压缩了权重和激活。SmoothQuant（Xiao等人，2023）、LLM.int8()（Dettmers等人，2022）和异常值抑制（Wei等人，2022）通过管理激活异常值实现了W8A8量化。LLM.int8()使用混合精度分解，而其他两种方法采用通道级缩放。此外，异常值抑制+（Wei等人，2023）增加了通道级移动以推动W6A6量化。与以前的启发式设计不同，我们使用梯度优化并将等效变换扩展到注意力机制，进一步提升了K/V缓存量化。**最近，RPTQ和LLM-QAT已经实现了W4A4量化。然而，RPTQ采用了对部署不友好的分组激活量化，而LLM-QAT采用了耗时的QAT。与RPTQ和LLM-QAT不同，我们通过部署友好的per-token量化实现了W4A4量化，并保持了PTQ的效率。**

3 OMNIQUANT

LLM量化的挑战 。量化LLM时存在两个主要困难。首先，由于异常通道的存在，激活很难量化。考虑到权重分布是平坦和均匀的，SmoothQuant和Outlier Suppression+通过预定义的迁移强度将量化难度从激活转移到权重来解决这个问题。**其次，权重的量化误差也由于对应激活的权重的重要性而在最终性能中起着关键作用。**SqQR和OWQ提出保留全精度的关键权重，而AWQ使用网格搜索的通道级缩放来保护这些权重。**尽管这些方法在压缩各种LLM方面取得了一定的成功，但由于手工设计的量化参数（如迁移强度和缩放因子）的粗糙设计，它们通常导致次优性能，并且无法处理极低比特量化。**

在本节中，我们介绍了一种**用于LLM的可微分量化技术** ，称为OmniQuant，其中量化参数具有更好的灵活性。为实现这一目标，OmniQuant采用块间量化误差最小化框架实现，如第3.1节所述。为应对上述LLM量化的挑战，我们设计了两种新策略，包括可学习的权重裁剪（LWC）以减轻量化权重的难度，以及可学习的等效变换（LET）进一步将量化挑战从激活转移到权重。我们在第3.2节和第3.3节分别介绍了LWC和LET。

3.1 BLOCK-WISE QUANTIZATION ERROR MINIMIZATION

之前采用梯度优化的PTQ方法，如AdaRound和BRECQ，不能应用于具有数十亿参数的模型，因为巨大的解空间难以优化这些权重。我们提出了一种新的优化流水线，采用块间量化误差最小化，其中额外的量化参数可以以可微分的方式优化。我们将优化目标表述如下：

$$\arg\min_{\Theta_1,\Theta_2} \left\|\mathcal{F}(\mathbf{W},\mathbf{X}) - \mathcal{F}\left(Q_w\left(\mathbf{W};\Theta_1,\Theta_2\right),Q_a\left(\mathbf{X},\Theta_2\right)\right)\right\| \quad (1)$$

其中 $\mathcal{F}$代表LLM中transformer block的映射函数，$\mathbf{W}$和$\mathbf{X}$是全精度的权重和激活，$Q_w(\cdot)$和$Q_a(\cdot)$分别代表权重和激活的量化器，Θ_1和Θ_2分别是learnable weight clipping（LWC）和learnable equivalent transformation（LET）中的量化参数。公式（1）为Block-wise粒度，依次量化每一个

transformer block。

公式（1）中的Block-wise minimization有两个优点：

1. OmniQuant可以联合优化LWC和LET中的量化参数，使其足以包含仅权重和权重-激活量化。
2. Block-wise minimization易于优化，资源要求最小。OmniQuant只需确定几个量化参数的最优性，这比优化之前PTQ方法中的整体权重（Nagel等人，2020；Li等人，2021）要容易。

从经验上，我们发现LLaMA-2家族的所有模型（Touvron等人，2023b）都可以在单个A100-40G GPU上仅使用128个训练样本进行量化。

3.2 LEARNABLE WEIGHT CLIPPING

OmniQuant采用LWC模块来降低LLM中权重量化的难度。与具有learnable clipping threshold的先前方法类似（如LSQ、PACT等），LWC也通过优化clipping threshold来确定权重的最佳动态范围。然而，我们发现直接采用先前工作，如PACT（Choi等人，2018）和LSQ（Esser等人，2019），量化性能不佳，正如LLM-QAT（Liu等人，2023a）所展示的那样。附录中的表A8也观察到了类似的结果。

我们不是像之前的方法那样直接学习clipping threshold，而是优化一个clipping strength，如公式(2)所示：

$$\begin{aligned} \mathbf{W_q} &= \text{clamp}\left(\left\lfloor \frac{\mathbf{W}}{h} \right\rceil + z, 0, 2^N - 1\right), \\ \text{where } h &= \frac{\gamma \max(\mathbf{W}) - \beta \min(\mathbf{W})}{2^N - 1}, z = -\left\lfloor \frac{\beta \min(\mathbf{W})}{h} \right\rceil \end{aligned} \tag{2}$$

其中h是权重的归一化因子，z是零点值。在公式(2)中，$\gamma \in [0,1]$和$\beta \in [0,1]$是上下界权重的learnable clipping strengths。我们用sigmoid函数实例化γ和β，因此公式（1）中的$\Theta_1 = \{\gamma, \beta\}$。

请注意，当$\gamma = 1$且$\beta = 1$时，LWC退化为现有工作中使用的普通MinMax量化方案（SmoothQuant、GPTQ）。**通过继承MinMax量化的好处，LWC只需要调整clipping strength来确定最佳clipping threshold，这将降低优化难度。通过clipping threshold** 进行权重clipping，将易于量化。如表1中的实验所示，我们提出的LWC方法显著优于以前的仅权重量化技术（Frantar等人，2022；Lin等人，2023）。

3.3 LEARNABLE EQUIVALENT TRANSFORMATION

除了使用LWC使得权重更利于量化，我们还通过可学习的等效变换（LET）进一步降低了权重-激活量化的难度。考虑到激活图中的异常值是有条理性的，并且主要存在于特定通道，先前的方法如SmoothQuant通过数学等效变换将量化难度从激活转移到权重。然而，他们的等效参数是人工设置的，导致次优结果。

得益于之前的Block-wise的量化误差最小化，我们的LET能够以可微分的方式确定最优等效参数。受到SmoothQuant（Xiao等人，2023）和Outlier Suppression+（Wei等人，2023）的启发，我们采用channel-wise scaling和channel-wise shifting来操纵激活分布，为异常值问题提供了有效的解决方案。具体来说，我们研究了linear layer和attention操作中的等效变换，如图3所示。

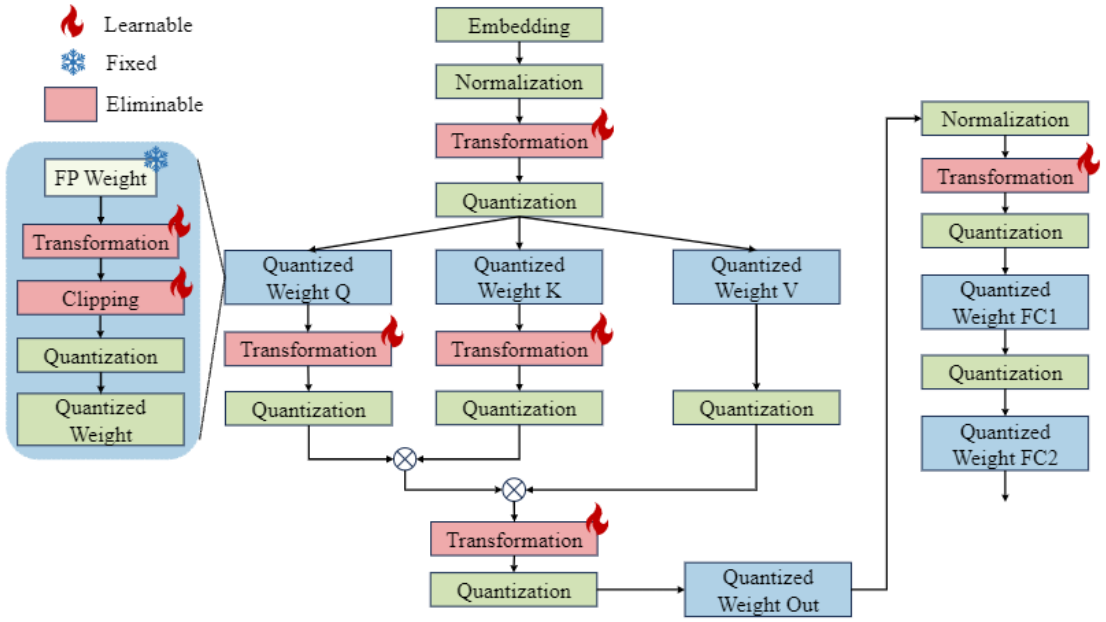


Figure 3: **Details of OmniQuant** in a transformer block. Note that all learnable parameters can be eliminated after quantization.

Untitled

Linear layer. 线性层接收输入token序列$\mathbf{X} \in \mathbb{R}^{T \times C_{in}}$，其中T是令牌长度，并且是权重矩阵$\mathbf{W} \in \mathbb{R}^{C_{in} \times C_{out}}$和偏置向量$\mathbf{B} \in \mathbb{R}^{1 \times C_{out}}$的乘积。数学等效的线性层表示为：

$$\mathbf{Y} = \mathbf{X}\mathbf{W} + \mathbf{B} = \underbrace{[(\mathbf{X} - \delta) \oslash s]}_{\tilde{\mathbf{X}}} \cdot \underbrace{[s \odot \mathbf{W}]}_{\tilde{\mathbf{W}}} + \underbrace{[\mathbf{B} + \delta\mathbf{W}]}_{\tilde{\mathbf{B}}} \tag{3}$$

其中$s \in \mathbb{R}^{1 \times C_{in}}$和$\delta \in \mathbb{R}^{1 \times C_{in}}$分别是通道级的scale和shift参数，$\tilde{\mathbf{X}}, \tilde{\mathbf{W}}, \tilde{\mathbf{B}}$分别为等效转换后的激活、权重和偏置。**通过公式（3），激活被转换为量化友好的形式，代价是权重的量化难度增加。在这种意义上，第3.2节中的LWC可以提高LET实现的权重-激活量化性能，因为它使权重量化友好。**

> s使用 SmoothQuant中的方法进行初始化；
>
> δ使用Outlier Suppression+中的方法进行初始化。

最后，我们对转换后的激活和权重进行量化，如下所示：

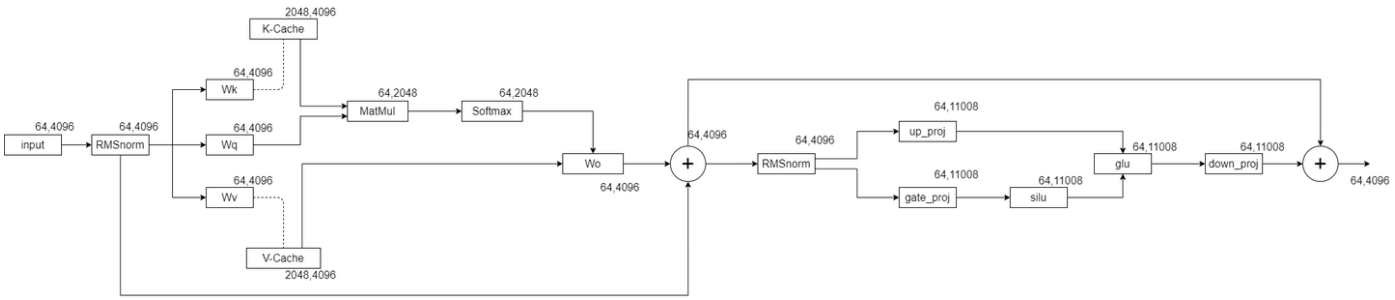
$$\mathbf{Y} = Q_a(\tilde{\mathbf{X}})Q_w(\tilde{\mathbf{W}}) + \tilde{\mathbf{B}} \tag{4}$$

请注意，$\tilde{\mathbf{X}}$的scale和shift参数可以被融合到先前的归一化或线性层中，$\tilde{\mathbf{W}}$中的scale因子可以与原始权重$\mathbf{W}$融合。因此，公式（3）中的等效变换可以有效减少量化误差，而不引入额外的参数或计算成本。**我们在LLM的所有线性层中使用这种等效变换，如图3所示，除了FFN的第二个线性层。这可能是因为非线性层之后的特征高度稀疏（Liu等人，2023b）导致应用可学习等效变换时梯度不稳定。**

Attention operation. 除了线性层之外，注意力操作也占据了计算的相当一部分。此外，LLM的自回归模式要求为每个token存储KV Cache，这导致长序列的内存需求很大。因此，在权重-激活量化设置中，我们也将Q/K/V矩阵量化为低比特。具体来说，self-attention亲和力矩阵的LET可以写成：

$$\mathbf{P}=\operatorname{Softmax}\left(\mathbf{Q}\mathbf{K}^{T}\right)=\operatorname{Softmax}((\underbrace{\mathbf{Q}\oslash s_a}_{\tilde{\mathbf{Q}}})(\underbrace{s_a\odot\mathbf{K}^{T}}_{\tilde{\mathbf{K}}^{T}})) \tag{5}$$

根据公式（4）和（5）可以得知LET的参数$\Theta_2=\{\delta,s,s_a\}$。值得一提的是，$V$的显式变换被省略了，因为它的分布已经被与输出Linear层（下图的$\mathbf{W}_o$）相关的逆变换按channel-wise改变了。



Untitled

4 EXPERIMENTS

4.2 仅权重量化结果

LLaMA系列的结果可以在表1中找到，而OPT的结果在附录A6中呈现。正如表格所示，OmniQuant在各种LLM系列（OPT，LLaMA-1，LLaMA2）和多样化的量化配置中始终优于以前的LLM仅权重量化方法，包括W2A16，W2A16g128，W2A16g64，W3A16，W3A16g128，W4A16和W4A16g128。这些发现表明OmniQuant的多功能性，能够适应多种量化配置。**例如，虽然AWQ（Lin等人，2023）在分组量化中特别有效，但OmniQuant在通道级和分组级量化中都表现出优越的性能。此外，随着量化比特大小的减小，OmniQuant的性能优势变得更加明显。**

4.3 权重-激活量化结果

在权重-激活量化中，我们主要关注W6A6和W4A4量化。我们排除了W8A8量化，因为与全精度对应物相比，SmoothQuant几乎可以实现无损的W8A8量化模型。LLaMA系列的结果可以在表2中找到，而OPT的结果在附录A16中呈现。表2展示了LLaMA权重-激活量化的零样本任务准确性。**值得注意的是，在W4A4量化中，OmniQuant显著提高了各种模型的平均准确性，提高了+4.99%～+11.80%。显著的是，在LLaMA-7B中，OmniQuant甚至超过了最近的QAT方法LLM-QAT（Liu等人，2023a），提高了+6.22%。这一改进证明了加入额外可学习参数的有效性，这比QAT使用的全局权重调整更有益。**