AWQ:LLM 压缩和加速的激活感知权重量化

吉林1* 唐家明1,2* 唐浩天1 尚阳1,3 党星宇1,3 宋瀚1 1MIT 2SJTU 3 清华大学 https://github.com/mit-han-lab/llm-awq

抽象的

大型语言模型（LLM)在各种任务上表现出了出色的性能,但天文数字的模型大小增加了服务的硬件障碍（内存大小)并减慢了令牌生成（内存带宽）。在本文中,我们提出了激活感知权重量化（AWQ）,这是一种用于 LLM 低位仅权重量化的硬件友好方法。我们的方法基于这样的观察:权重并不同等重要:仅保护 1% 的显着权重就可以大大减少量化误差。然后,我们建议通过观察激活而不是权重来搜索最佳的每通道缩放,以保护显着权重。 AWQ不依赖于任何反向传播或重构,因此可以很好地保留LLM在不同领域和模态上的泛化能力,而不会过度拟合校准集;它也不依赖于任何数据布局重新排序,从而保持了硬件效率。 AWQ 在各种语言建模、常识 QA 和特定领域基准测试方面优于现有工作。由于具有更好的泛化能力,它为指令调整的 LM 以及首次为多模态 LM 实现了出色的量化性能。我们还实现了高效的张量核心内核,具有无重排序在线反量化功能,以加速 AWQ,实现了比 GPTQ 快 1.45 倍的加速,比 cuBLAS FP16 实现快 1.85 倍。我们的方法提供了一个交钥匙解决方案,将 LLM 压缩到 3/4 位,以实现高效部署。

1 简介

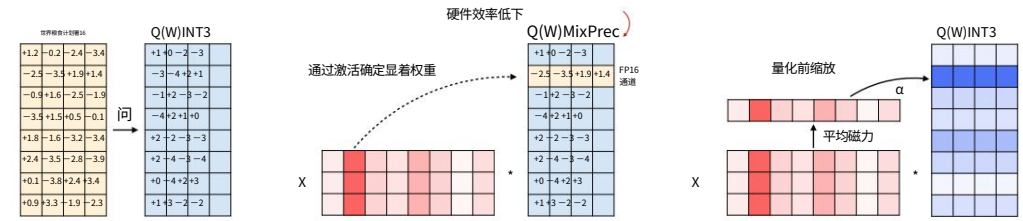
基于 Transformer [48] 的大型语言模型 (LLM) 在各种基准测试中表现出了出色的性能 [5,58,47,43]。然而,较大的模型尺寸导致较高的服务成本。
例如,GPT-3有175B参数,在FP16中为350GB,而最新的H100 GPU只有96GB内存,更不用说边缘设备了。

LLM 的低位权重量化可以节省内存,但很困难。由于训练成本较高,量化感知训练（QAT)并不实用,而训练后量化（PTQ)在低位设置下会出现较大的精度下降。最接近的工作是GPTQ[17],它使用二阶信息来进行误差补偿。但它依赖于重新排序技术来适用于某些模型（例如 OPT-66B 和 LLaMA-7B）,这种技术的硬件效率较低（大约慢 2 倍,图 6）。它还可能在重建过程中过度拟合校准集,从而扭曲分布外域上的学习特征（图 7）,这可能是多模态模型的问题。

在本文中,我们提出了激活感知权重量化（AWQ）,这是一种适用于 LLM 的硬件友好的低位仅权重量化方法。我们的方法基于这样的观察:权重对于法学硕士的表现并不同样重要。显着权重只占一小部分（0.1%-1%）;跳过这些显着权重的量化将显着减少量化损失（表 1）。为了找到显着的权重通道,我们应该参考激活

*表示同等贡献。

预印本。正在审核中。



(a) RTN 量化 (PPL 43.2) (b) 在 FP16 (PPL 13.0) 中保留 1% 的显着权重 (c) 在量化之前缩放权重 (PPL 13.0)

图 1。我们观察到,通过观察激活分布（中）,我们可以找到 LLM 中 1% 的显着权重。将显着权重保留在 FP16 中可以显着提高量化性能（PPL 从 43.2（左)到 13.0（中））,但混合精度格式并不具有硬件效率。我们遵循激活感知原则并提出 AWQ（右）。 AWQ 执行每通道缩放以保护显着权重,从而减少量化误差。 PPL 在 INT3-g128 量化下使用 OPT-6.7B 进行测量。

尽管我们正在进行仅权重量化:与较大激活幅度相对应的权重通道更加显着,因为它们处理更重要的特征。为了避免混合精度,我们遵循激活感知原则,设计了每通道缩放方法来搜索最小化量化误差的最佳缩放因子,并对所有权重进行量化。

AWQ不依赖于任何反向传播或重建,因此它可以很好地保留LLM在各种领域和模式上的泛化能力,而不会过度拟合校准集。它不依赖于任何数据布局重新排序和保存内存规律性,这比基于重新排序的方法快 2 倍。我们还从头开始实现高效的 GPU 内核以支持 AWQ。我们采用无重新排序在线反量化来有效地将低位权重转换为 FP16,并有效映射到高吞吐量张量核心而不是 cuda 核心。

实验表明,对于不同的模型系列（例如 LLaMA [47]、OPT [58])和模型大小,AWQ 在各种任务（语言建模、常识 QA 和特定领域基准)上的表现优于现有工作。由于更好的泛化,它还为指令调整的 LM（例如 Vicuna)和首次多模态 LM（Open Flamingo [2])实现了良好的量化性能。得益于我们高效的内核,AWQ 比 GPTQ 和在 A100 上重新排序的 GPTQ 实现了 1.45 倍和 2 倍的加速。它还比 FP16 cuBLAS 实现快 1.85 倍。

AWQ 使多模态 LM 更加高效,同时又不损害泛化能力。

2 激活感知权重量化 (AWQ)

预赛。量化将浮点数映射为低位整数。它是减少法学硕士的模型大小和推理成本的有效方法[12,17,55,54]。 LLM 量化通常有两种量化设置： 1. W8A8 量化 [12, 54],它将激活和权重映射到 INT8； 2.低位仅权重量化[17, 39],将权重量化为低位整数（通常≤4位）。解码阶段在 LLM 的总运行时间中占主导地位 [17, 32],其在单个批量大小下受到内存的高度限制。鉴于内存由权重主导,我们专注于仅权重量化。我们研究分组量化（组大小为 128 [17]）,因为它有助于以很少的开销提高准确性成本权衡 [13]。

在本节中,我们首先提出一种通过保护更“重要”的权重来提高量化性能而无需训练/回归的方法。然后开发一种数据驱动的方法来搜索可减少量化误差的最佳缩放比例（图 1）。

2.1 通过保留 1% 的显着权重来改进 LLM 量化

我们观察到法学硕士的权重并不同样重要:与其他权重相比,有一小部分显着权重对于法学硕士的表现更为重要。跳过这些显着权重的量化可以帮助弥补由于量化损失而导致的性能下降,而无需任何训练或回归（图 1(b)）。为了验证这个想法,我们在跳过表 1 中的部分权重通道时对量化 LLM 的性能进行了基准测试。我们测量了 INT3 量化模型的性能,同时保留 FP16 中的一些权重通道比率。确定权重重要性的一种广泛使用的方法是查看其大小或 L2 范数 [22, 16]。但我们发现跳过具有大范数（即 FP16%（基于 W）)的权重通道并不能显着提高量化性能,从而导致类似的边际性能

PPL ↓	FP16 RTN (w3-g128)	FP16%（基于动作） 0.1%	1%	3%	FP16%（基于W） 0.1%	1%	3%	FP16%（随机） 0.1%	1%	3%
OPT-1.3B 16.41	206.5	28.00	18.51	18.30	187.1	173.1	165.5	211.2	201.4	88.44
OPT-6.7B 12.29	43.16	13.14	13.02	12.97	43.51	38.59	39.78	42.73	37.83	46.49
OPT-13B 11.50	45.37	12.14	12.04	12.00	46.21	48.07	54.38	45.95	44.47	40.01

表 1.在 FP16 中保留一小部分权重 (0.1%-1%) 可显着提高性能
舍入到最近值 (RTN) 的量化模型。仅当我们选择 FP16 中的重要权重时才有效
通过查看激活分布而不是权重分布。我们以适当的困惑来突出显示结果
为绿色。我们使用 INT3 量化，组大小为 128，并测量了 WikiText 的困惑度 (↓)。

PPL ↓	FP16 RTN (w3-g128)	启发式缩放 ×1.25	×1.5	×2	×4	仅限sX	按比例搜索 sX& sW +clip		
OPT-1.3B 16.41	206.5	30.89	21.75	21.05	21.95	19.45	19.07	19.03	18.53
OPT-6.7B 12.29	43.16	15.45	14.49	14.07	14.42	14.46	11.71	11.57	13.18	13.13	12.99
LLaMA-7B 9.49	12.10	11.48	13.26	11.42				11.24	11.27	10.82

表 2.简单地按大于 1 的系数放大显着权重通道可以大大减少
量化误差（启发式缩放）。我们进一步建议寻找最佳规模。考虑到
激活分布（即sX)是缩放因子搜索过程中最重要的因素。

改进为随机选择。有趣的是，根据激活幅度选择权重可以
显着提高性能：只保留0.1%-1%的对应更大的通道
激活显着提高了量化性能，甚至可以匹配基于强重构的方法 GPTQ [17]。我们假设具有较大幅度的输入特征通常是
更重要。在 FP16 中保留相应的权重可以保留这些特征，这
有助于更好的模型性能。

局限性：尽管在 FP16 中保留 0.1% 的权重可以提高量化性能
模型大小（以总位数衡量）没有明显增加，这样的混合精度数据类型
会给系统的实施带来困难。我们需要想出一个方法来保护
重要的权重，但实际上并未将它们保留为 FP16。

2.2 通过激活感知缩放来保护显着权重

我们提出了一种通过每通道缩放来减少显着权重的量化误差的方法，
它不会受到硬件效率低下问题的影响。

基于启发式的缩放。我们首先实现基于启发式的缩放方法来减少量化误差。由于我们在之前的工作[12,54,17]之后使用基于最小-最大的量化，
量化尺度由每组中的极值决定（即， $s = (w_{max} - w_{min})/(2^N - 1)$，其中N是量化位宽），因此没有量化损失
最大值和最小值*。保护异常值权重通道的一种直接方法是
将通道乘以一定的缩放比例（>1），以便可以精确量化它们。
如表2 （启发式缩放列）所示，此类方法确实提高了性能
量化模型，但与 FP16 中直接保留 1% 的权重相比仍有差距。这
由于重要的权重，量化性能通常会随着缩放因子的增加而变得更好
得到了更好的代表。然后它会减少，因为非显着通道被迫使用较小的通道
如果我们使用非常大的缩放因子，则动态范围（或更小的有效位）。我们需要减少
重要权重的量化误差，同时不增加其他权重的误差。我们
需要一种自动化的方法来找到实现目标的每个输入通道的缩放比例。

规模搜索。我们自动搜索最佳（每个输入通道）缩放因子
最小化某一层量化后的输出差异。正式来说，我们想要优化
目标如下：

$$s^* = \arg\min_s L(s),\ L(s) = \lVert Q(W \cdot s)(s^{-1} \cdot X) - WX \rVert \quad (1)$$

*同一量化组中出现多个异常值的可能性很小。但这种情况会很少见
当使用像 0.1% 这样的小异常值比率时，我们将在研究中忽略它。

这里Q表示权重量化函数（例如,组大小为 128 的 INT3/INT4 量化）， W是 FP16 中的原始权重， X是从小校准集中缓存的输入特征（我们从预训练集中获取一个小校准集）。 -训练数据集以免过度适应特定任务）。 s是每（输入)通道缩放因子;对于s · X,通常可以融合到前一个算子中[52, 54]。由于量化函数不可微,因此我们无法直接使用普通反向传播来优化问题。有一些技术依赖于近似梯度[3, 15],我们发现这些技术仍然收敛得很差,并且存在过度拟合的潜在风险。

我们通过分析影响缩放因子选择的因素来设计最佳缩放的搜索空间。直观上,最佳尺度应该与以下因素有关： 1.激活幅度:如上所述,我们依靠输入激活幅度来挑选显着权重通道。

因此,我们应该考虑X的大小来保护显着的权重通道。我们计算平均激活幅度sX = meanc_out|X|作为重要因素。 2.权重大小:为了最小化非显着权重的量化损失,我们应该展平它们的分布,以便更容易量化[54]。合理的选择是将权重通道除以它们的平均幅度sW = meanc_out|W^ | [54] ,其中W^指的是每组内归一化后的W。

我们通过考虑两个因素s = f(sX, sW) 的函数来确定最终的缩放因子。
为了简单起见,我们将两个尺度相乘并求解：

$$s = f(sX, sW) = sX^{\alpha}_{sW^{-\beta}}, \quad \alpha^*, \beta^* = \arg\min_{\alpha,\beta} L(sX^{\alpha}_{sW^{-\beta}}) \tag{2}$$

α和β是控制每个分量强度的超参数。我们可以通过区间 [0, 1] 上的简单网格搜索来选择最佳的α和β 。如表 2 所示,基于sX 的缩放明显优于sW,这证明了激活感知的重要性。进一步引入软件只会带来边际改进,这再次表明激活意识是最重要的。此外,我们发现通过搜索收缩率（表示为“+clip”）来调整裁剪范围有时可以进一步提高量化性能。调整裁剪范围会导致缩放因子[15] ,这可能有助于更好地保护显着权重。我们将缩放和裁剪结合起来形成AWQ。

解释 GPTQ 重新排序。我们的激活感知原理也可以解释为什么我们需要重新排序才能使 GPTQ 在某些模型上工作（例如 LLaMA-7B 和 OPT-66B,请参阅 GPTQ 存储库中的 --act-order 选项† ） 。不同权重通道的重要性不同;更新显着通道以补偿非显着通道可能会破坏性能。重新排序通过首先量化重要通道来防止这种情况发生。然而，由于不规则的内存访问,这会导致硬件效率降低（图2） ,而我们的缩放方法不会遇到这个问题。

优点。此外,我们的方法不依赖于任何回归[17]或反向传播,而这是许多量化感知训练方法所必需的。它对校准集的依赖最小,因为我们只测量每个通道的平均幅度,从而防止过度拟合（图7）。因此,我们的方法在量化过程中需要更少的数据,并且可以将法学硕士的知识保留在校准集分布之外。更多详细信息,请参见第3.4节。

2.3 协同设计高效量化GPU内核

AWQ 的硬件友好特性促进了高效 GPU 内核的开发,有效地将理论内存节省转化为测量的加速。

与将线性层表示为矩阵向量（MV)乘积的GPTQ [17]不同,我们将这些层建模为矩阵-矩阵（MM)乘法。 MV 只能在慢速 CUDA 核心上执行,而 MM可以在 A100 和 H100 上快 16 倍的张量核心上执行。与[17]相比,我们的公式还最大限度地减少了结构风险,因为 MM 和其他指令（例如反量化和内存访问)是在单独的功能单元上执行的。我们还比最近的GPTQ Triton [46]实现[40]好2.4 倍,因为它依赖于高级语言并放弃了低级优化的机会。

图 2.重新排序会导致随机内存效率低下
理论访问。

†https://github.com/IST-DASLab/gptq

拉马-7B	MMLU（5 次）↑					常识 QA（0 次）↑				
	嗡嗡声。	STEM	社交	其他	平均	皮卡·海拉。	酒诺。	ARC-e	平均	
FP16-	39.17%	32.32%	42.72%	42.56%	38.41%	78.35%	56.44%	67.09%	67.30%	67.30%
INT3 g128	RTN 31.37%	31.10%	36.04%	36.49%	33.43%	75.84%	53.10%	63.22%	66.04%	64.55%
	GPTQ 29.29%	29.04%	33.03%	31.65%	30.53%	70.89%	46.7 7%	60.93%	60.06%	59.66%
	GPTQ-R 33.98%	30.71%	37.78 %	36.49%	34.26%	77.31%	53.81%	67.56%	63.72%	65.60%
	每周定量 35.15%	31.61%	39.27%	37.75%	35.43%	76.66%	53.63%	66.14%	65.70%	65.5 3%
INT4 g128	RTN 36.15%	33.03%	41.41%	41.21%	37.37%	77.86%	55.81%	65.59%	66.25%	66.38%
	GPTQ 35.55%	30.95%	39.29%	38.12%	35.39%	77.20%	53.9 8%	65.67%	61.62%	64.62%
	GPTQ-R 37.28%	31.36%	40.23 %	40.77%	36.72%	78.45%	56.00%	66.85%	66.88%	67.05%
	每周定量 38.32%	32.00%	41.38%	42.07%	37.71%	78.07%	55.76%	65.82%	66.84%	66.6 2%

表 3. 对于 LLaMA-7B,在组大小为 128 的 3/4 位量化下,AWQ 始终优于 RTN 和 GPTQ。请注意,GPTQ 需要硬件低效的重新排序技巧才能在 7B 模型上工作,表示为 GPTQ-R,它的速度明显较慢（图 6）,因此我们将其排除在性能比较之外。

仅权重量化方法需要在线权重反量化。在图 2 中,我们假设权重以行主存储（即 IC×OC）。每个OC 具有不同的标度和零点,并且每个OC 内的每两个IC 共享反量化参数（即g = 2）。对于重新排序的 GPTQ（右）,由于这些 g = 2 IC 不连续,因此在反量化每个权重时需要不规则的 DRAM 访问来获取缩放因子和零点。然而,对于 AWQ（左）,所有内存访问都是连续的,导致 LLaMA 模型的端到端加速提高了 2 倍。

3 实验

3.1 设置量化。在这

项工作中,我们专注于仅权重分组量化。如之前的工作 [13, 17] 所示,分组量化始终有助于改善性能/模型大小的权衡。除非另有说明,我们在整个工作中使用的小组规模为 128 人。我们关注 INT4/INT3 量化,因为它们能够大部分保留 LLM 的性能 [13]。对于 AWQ,我们使用 Pile [18] 数据集中的一个小型校准集,以免过度拟合特定的下游域。我们使用 20 的网格大小来搜索公式 2 中的最佳 α 和 β。

楷模。我们在 LLaMA [47] 和 OPT [58] 系列上对我们的方法进行了基准测试。还有其他开放的法学硕士,如 BLOOM [43],但它们的质量普遍较差,因此我们不将它们纳入我们的研究中。我们进一步对指令调整模型 Vicuna [8] 和视觉语言模型 OpenFlamingo-9B [2] 和 LLaVA-13B [31] 进行基准测试,以证明我们方法的可生成性。

评价。继之前的文献[12,54,17,13,55]之后,我们分析了语言建模任务（WikiText-2 [33])和常识 QA 基准（PIQA [4]、HellaSwag [56]、WinoGrande [41],ARC [11];由于 LLaMA 评估的已知错误,我们不使用 LAMBADA [38]。我们注意到这些基准并不反映法学硕士的特定领域知识。因此,我们进一步在 MMLU [23] 上对模型进行基准测试,该模型由涵盖 STEM、人文、社会科学等的 57 个任务组成,这也有助于评估少镜头设置下的情境学习能力。我们使用 lm-eval-harness [19] 来进行所有评估。

基线。我们的主要基线是普通舍入到最近量化（RTN）。当使用像 128 [17, 13] 这样的小组大小时,它实际上相当强大。我们还与 LLM 权重量化的最先进方法 GPTQ [17] 进行比较。对于 GPTQ,我们还与使用“重新排序”技巧（表示为 GPTQ-Reorder 或 GPTQ-R)的更新版本进行比较,该技巧提高了性能,但导致解码效率较差。其他技术,如 ZeroQuant [55]、AdaRound [34] 和 BRECQ [28] 依赖反向传播来更新量化权重,这可能不容易扩展到大模型大小;它们的表现也不优于 GPTQ [17],因此未纳入研究。

3.2 LLaMA 模型的精度评估结果。我们将

研究重点放在 LLaMA 模型上,因为与其他开源 LLM 相比,它们具有优越的性能 [58, 43];它也是许多流行开源模型的基础 [45, 8]。我们在常识 QA 任务上评估不同的量化方法

骆驼家族		MMLU（5 次)平均 ↑				常识 QA（0 次)平均值 ↑			
		7B	13B	30B	65B	7B	13B	30B	65B
FP16-		38.41%	45.21%	56.84%	60.50%	67.30%	70.65%	72.97%	74.49%
INT3 G128	实时网络	33.43%	39.20%	50.58%	57.77%	64.55%	68.63%	72.07%	72.58%
	GPTQ	30.53%	40.90%	52.32%	58.04%	59.66%	68.71%	70.77%	73.03%
	全年季度	35.43%	41.84%	53.22%	58.83%	65.53%	69.22%	72.10%	73.39%

表 4. 不同尺度的 LLaMA 量化结果。 AWQ 的性能优于舍入到最近值 (RTN) 和
GPTQ [17] 跨越不同的模型规模 (7B-65B)、任务类型（常识与特定领域)和测试
设置（零样本与上下文学习）。

OPT/PPL↓		125M	1.3B	2.7B	6.7B	13B	30B	66B
FP16	-	31.95	16.41	14.32	12.29	11.5	10.67	10.09
INT3 G128	RTN	58.49	206.54	595.28	43.16	45.37	28.84	423.39
	GPTQ	41.93	18.53	15.79	13.13	12.01	11.00	11.48
	AWQ	41.10	18.53	15.62	12.99	12.03	11.03	10.46
INT4 G128	RTN	35.51	17.70	15.12	13.02	11.89	11.00	10.44
	GPTQ	34.23	16.92	14.69	12.51	11.60	10.74	10.24
	AWQ	33.96	16.85	14.61	12.44	11.60	10.75	10.16

表 5. 对于所有模型大小和不同的位精度，AWQ 相对于舍入到最近量化 (RTN) 进行了改进。
与 GPTQ 相比，它在较小的 OPT 模型上实现了更好的 WikiText-2 困惑度，在较大的 OPT 模型上实现了同等结果
展示了不同型号尺寸和系列的通用性。我们发现 OPT 在以下方面表现不佳
MMLU 的上下文学习能力有限（甚至 FP16 模型），因此我们排除了这些评估。

（0-shot)和具有上下文学习的 MMLU 基准（5-shot）。我们在初步实验中发现，量化可能会损害模型的上下文学习性能，这是
一个基本的模型
法学硕士的能力[6]；因此，MMLU（5-shot)是我们研究中的重要基准。我们提供
表3中LLaMA-7B的INT3-g128/INT4-g128结果；对于较大的模型（表 4），我们发现
4 位 RTN 精度已经相当不错，因此我们只包含 3 位结果。我们可以看到AWQ
在不同的模型尺度（7B-65B）、任务中优于舍入到最近（RTN)和 GPTQ [17]
类型（常识与特定领域)和测试设置（零样本与上下文学习）。笔记
组大小为 128 的 RTN 基线已经相当强（只有百分之几的准确度）
下降），因此 1% 的误差减少被认为是显着的。对于 LLaMA-7B 模型，我们需要
包括使 GPTQ 工作的重新排序技巧，这会导致硬件低效的内存访问
内核实现速度减慢 2 倍（图 6）。因此，我们只提供结果
参考但排除它们进行比较（灰色）。

OPT 模型的结果。我们还提供了 OPT 模型的结果 [58]。我们发现OPT
模型通常达到非常低的 MMLU 精度（可能没有意义)并且上下文学习能力有限（请参见补充中的表 9）。因此，我们关注
WikiText-2
困惑度评估如下[17]。如表5所示，AWQ同时提高了INT3和INT4
分组量化。对于较小的模型（<13B），我们的方法可以优于竞争对手的 GPTQ
尽管它很简单，但结果却如此；而对于较大的模型，我们能够获得同等的性能。这
展示了不同型号系列和型号尺寸的通用性。

指令调整模型的量化。指令调整可以显着提高
模型的性能和可用性 [50,42,37,10]。已成为之前的必经程序
模型部署。我们进一步在图 3 中的流行指令调整模型 Vicuna [8] 上对我们的方法的性能进行基准测试。我们使用 GPT-4 分数来
评估量化模型的性能
在 80 个样本问题上与 FP16 对应的表现 [8]。我们比较回复
具有两个顺序（量化-FP16、FP16-量化)以消除排序效应（我们发现 GPT-4
往往会增加第一个输入的评级），导致 160 次试验。 AWQ 持续改善
两种尺度（7B 和 13B)下 RTN 和 GPTQ 上的 INT3-g128 量化 Vicuna 模型，
展示指令调整模型的可生成性。

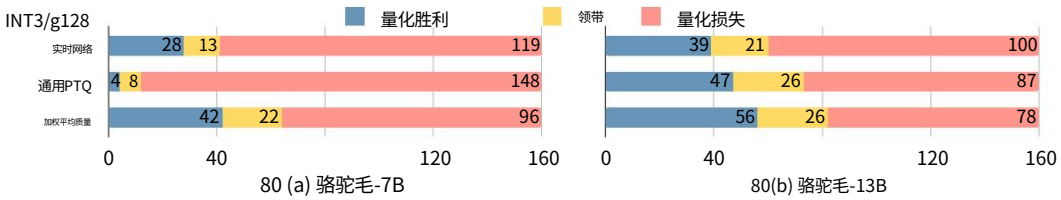


图 3. GPT-4 评估下 INT3-g128 量化 Vicuna 模型与 FP16 模型的比较
协议[8]。更多获胜案例（蓝色)表明性能更好。 AWQ 持续改进
与 RTN 和 GPTQ [17] 相比,量化性能表现出对指令调整模型的泛化。

可可（苹果酒↑）		0次射击	4发	8发	16发	32发	Δ(32发)
FP16-		63.73	72.18	76.95	79.74	81.70	-
INT4 G128	实时网络	60.24	68.07	72.46	74.09	77.13	-4.57
	通用PTQ	59.72	67.68	72.53	74.98	74.98	-6.72
	加权平均质量	62.57	71.02	74.75	78.23	80.53	-1.17
INT3 G128	实时网络	46.07	55.13	60.46	63.21	64.79	-16.91
	通用PTQ	29.84	50.77	56.55	60.54	64.77	-16.93
	加权平均质量	56.33	64.73	68.79	72.86	74.47	-7.23

表 6. 视觉语言模型 OpenFlamingo-9B [2] 在 COCO Captioning 数据集上的量化结果。
AWQ 在零样本和各种少样本设置下优于现有方法,证明了可生成性
不同的模式和情境学习工作量。 AWQ 减少量化退化（32 次）
在 INT4-g128 下从 4.57 到 1.17,模型尺寸减小了 4 倍,性能损失可以忽略不计。

多模态语言模型的量化。大型多模态模型 (LMM) 或视觉
语言模型 (VLM) 是通过视觉输入增强的 LLM [1,27,26,14,57,31]。这样的
模型能够根据图像/视频输入执行文本生成。由于我们的方法
不存在校准集的过拟合问题,可以直接应用于VLM来提供
准确高效的量化。我们使用 OpenFlamingo-9B 模型进行实验 [2]
（[1] 的开源复制品)在 COCO 字幕 [7] 数据集上（表 6）。我们测量了
不同少样本设置下 5k 样本的平均性能。我们只量化语言
模型的一部分,因为它主导模型大小。 AWQ 优于现有方法
零样本和各种少样本设置,展示了不同模式和的可生成性
情境学习工作量。它将量化退化（32 个镜头)从 4.57 减少到 1.17
在 INT4-g128 下,模型尺寸减小了 4 倍,性能损失可以忽略不计。我们进一步
图 4 中提供了一些定性字幕结果,以显示我们相对于 RTN 的优势。我们的方法
为 LMM/VLM 量化提供按钮式解决方案。首次研究VLM低位
据我们所知,量化。

视觉推理结果。我们进一步提供了一些定性视觉推理示例
图 5 中的 LLaVA-13B [31] 模型。与舍入到最近值相比,AWQ 改善了响应
(RTN) INT4-g128 量化的基线,从而得出更合理的答案。在第一个示例中，
AWQ 模型可以理解模因,因为从太空看时它类似于地球,而
RTN 产生错误的描述（以红色标记）。在第二个示例中,AWQ 正确回答
问题（这幅画的艺术家）,而 RTN 没有提供任何有关该画作的信息
艺术家。在最后一个例子中,RTN 错误地指出了图片中的一只鸟,而 AWQ 提供了更多
通过注意到图像是在山区拍摄的来获取信息。 AWQ 改善视觉推理
通过减少响应中的事实错误来提高 VLM 的能力； RTN 即使对于 4 位来说也不够好。

极低比特量化。我们进一步将 LLM 量化为 INT2 以适应有限的设备
内存（表 7）。 RTN彻底失效,AWQ带来显着的困惑度改善
尽管与 FP16 相比仍然存在性能差距,但 GPTQ 仍处于领先地位。我们的方法是正交的
到 GPTQ。我们可以将我们的方法与 GPTQ 结合起来,进一步改进 INT2 量化
性能,使其成为更实用的设置。

3.3 效率评估

与 GPTQ 内核的比较。在图6(a)中,我们比较了LLaMA的端到端效率
使用 AWQ 和 GPTQ 量化的模型。由于官方 GPTQ 实现 [17] 没有

W4-RTN:在天空中飞行的模型飞机。

W4-AWQ:两架玩具飞机停在草地上。

W4-RTN:一名男子抱着一头小象在他的怀里。

W4-AWQ:一名男子和他的女儿与大象合影。

W4-RTN:一名男子和一只狗走过一些灌木丛。

W4-AWQ:两只狗在街上散步。

图 4. COCO 字幕数据集上量化 OpenFlamingo-9B [2] 的定性结果（4 镜头，INT4-g128 量化）。与舍入到最近的方法相比，我们的方法显着提高了字幕质量（RTN）基线。我们为文本着色以显示正确或错误的标题。

问：你能详细解释一下这个表情包吗？

INT4-g128 RTN：图像中的模因是一盘食物，可能是炸鸡，具有独特的风味。食物上面放置着地球和其他行星的小图片，营造出有趣的视觉效果。

盘子旁边有一条说明文字：“有时我从太空看地球的照片，我惊叹于它的美丽。”这种食物和行星图像的结合增添了一丝幽默和好奇心，让观看者想知道模因背后的背景和意图。

INT4-g128 AWQ：图像中的模因是对从太空观看地球图片这一概念的轻松幽默的诠释。一盘油炸食品，特别是炸鸡块，上面写着这样的标题：“有时我从太空看地球的照片，我惊叹于这一切是多么美丽。”食物和与地球相关的背景的有趣组合创造了有趣且有趣的视觉效果。

问：你知道这幅画是谁画的吗？

INT4-g128 RTN：这幅画描绘了一个女人，有些人认为她是著名艺术家蒙娜丽莎的代表。然而，并不能确定画中的女人确实是蒙娜丽莎。该艺术品是艺术分析和欣赏的热门主题，通常被认为是女性之美和神秘的象征。这幅肖像画的特点是其复杂的细节，以女人富有表现力的目光和微妙而神秘的微笑吸引着观众，这已成为蒙娜丽莎的代名词。

INT4-g128 AWQ：这幅画描绘了一个女人，据信是艺术家达芬奇的著名艺术作品《蒙娜丽莎》的肖像。尽管这幅画被广泛认为是《蒙娜丽莎》的代表作，但在没有额外信息的情况下，无法确认绘制这幅画的艺术家。列奥纳多·达·芬奇是一位著名的艺术家，以其绘制细致、逼真的肖像画的非凡技巧而闻名，《蒙娜丽莎》是他最著名的作品之一。

问：去这里旅游有哪些需要注意的地方？

INT4-g128 RTN：参观此地点时，应小心延伸到水面上的木墩。
木板可能会很滑，尤其是在潮湿或下面有水流动的情况下。在码头上小心行走并保持平衡很重要，以避免发生任何事故或落入水中。
此外，请注意周围的环境，因为可能存在可能带来风险的野生动物或自然元素。例如，图像中有一只鸟，这可能表明该地区存在更多野生动物。考虑这些因素将有助于确保您在游览这个风景如画的地方时获得安全和愉快的体验。

INT4-g128 AWQ：在游览这个风景优美的地方时，有一个木码头延伸到一大片水域，周围群山环绕，有必要注意一些事情。首先，要注意码头的木质表面很滑，特别是在潮湿或下雪的环境中，可能会造成滑倒和掉入水中的风险。其次，要注意天气状况，山区天气变化很快，可能会影响能见度，难以安全返回岸边。最后，尊重自然环境和该地区可能存在的任何野生动物。扰乱生态系统或乱扔垃圾可能会对当地动植物群产生负面影响。通过遵循这些预防措施，您可以确保您安全愉快地游览这个风景如画的地方。

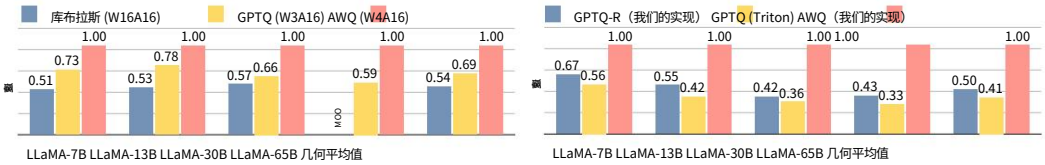
图 5.LLaVA-13B 模型的视觉推理示例 [31]。 AWQ 比舍入到最接近的方法有所改进（RTN）基线，提供更合理的答案。我们对文本进行着色以显示正确或错误的答案。

支持4位权重和重新排序，我们展示了没有重新排序的3位GPTQ的延迟。值得注意的是，尽管每个权重使用了额外的比特，AWQ 仍实现了平均加速 1.45 倍，比 GPTQ 最大加速 1.7 倍。我们将这种效率的提高归功于我们将 LLM 中的线性层表示为矩阵-矩阵而不是矩阵-向量乘积。因此，我们可以利用 A100 GPU 上的高吞吐量张量核心并避免资源争用计算和数据准备（例如反量化和内存访问）指令之间。

重新排序和编译器的开销。图6(b)展示了消除的重要性去量化期间重新排序。为了支持 GPTQ-R，我们只对 AWQ 做了一处修改内核：在权重反量化中加载零点和比例时引入间接寻址。事实证明，具有无重排序内核的 AWQ 比 GPTQ-R 快 1.4-2.4 倍。这清楚地表明 GPTQ-R 内核中的不规则内存访问模式不适合硬件，并且对性能产生显着的负面影响。此外，我们注意到直接的 CUDA 级别实施对于实现比 cuBLAS 和 GPTQ 一致的性能提升至关重要。我们针对基于 Triton [46] 的开源实现 [40] 对 AWQ 内核进行基准测试编译器并实现了 1.8-3.1 倍的加速。 Triton 内核比 FP16 慢 1.6 倍 LLaMA-30B 中的 cuBLAS 内核，而 AWQ 内核仍然快 1.8 倍。这进一步强调了编译器在识别和利用低级优化机会方面的局限性。

OPT / 维基 PPL↓		1.3B	2.7B	6.7B	13B	30B
FP16	-	16.41	14.32	12.29	11.5	10.67
INT2 g64	实时网络	26615	740860	22290	28923	15198
	通用PTQ	50.05	30.41	19.04	16.8	12.91
	AWQ+GPTQ 35.26		24.02	17.34	14.58	12.5

表 7.我们的方法与 GPTQ 正交:它进一步缩小了极低位下的性能差距
与 GPTQ 结合使用时的量化 (INT2-g64)。结果是 OPT 模型的 WikiText-2 困惑度。



(a) 4 位 AWQ比 3 位 GPTQ快1.45 倍（无需重新排序） (b) 消除重新排序为 W4A16带来2 倍加速

图 6. 左:使用 4 位 AWQ 量化的 LLaMA 模型比 3 位 GPTQ 快1.45 倍。右:
GPTQ-R 中的重新排序技巧极大地降低了其硬件效率。 AWQ 无需重新排序且速度提高2.0 倍
比GPTQ-R。 AWQ 适合 GPU 优化,比 Triton 实现快2.4 倍。全部
数字是在单个 80G A100 GPU 上测量的。

3.4 分析

校准集的数据效率更高。我们的方法需要较小的校准集,因为
我们不依赖回归/反向传播;我们只测量平均激活规模
校准集,数据效率高。为了证明这个想法,我们比较了
图7 (a)中具有 INT3-g128 量化的 OPT-6.7B 模型。 AWQ 需要更小的
校准以达到良好的量化性能;它可以使用 10 倍更小的尺寸来实现更好的困惑度
校准集与 GPTQ 的比较（16 个序列与 192 个序列）。

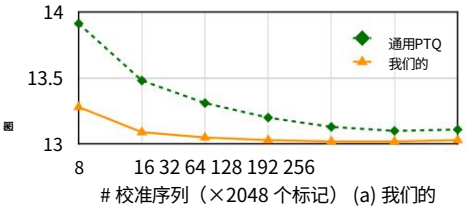
对校准集分布具有鲁棒性。我们的方法对校准集不太敏感

分布,因为我们只测量校准集中的平均激活尺度,这更
可推广到不同的数据集分布。我们进一步对不同的效果进行了基准测试
校准集分布如图7(b) 所示。我们从 Pile 数据集[18] 中获取了两个子集： PubMed
摘要和安然电子邮件[25]。我们使用每个子集作为校准集并评估
两组的量化模型（校准集和评估集被分割,没有重叠;
我们使用 1k 个样本进行评估）。总体而言,使用相同的校准和评估分布
效果最好（PubMed-PubMed、Enron-Enron）。但是当使用不同的校准分布时
（PubMed-Enron、Enron-PubMed）,AWQ 仅增加了 0.5-0.6 的困惑度,而 GPTQ 则增加了
2.3-4.9 更糟糕的困惑。这证明了 AWQ 对校准集分布的鲁棒性。

使用 SmoothQuant 进行微分。乍一看,推导公式（方程2)是相关的
SmoothQuant [54],它平衡了激活和权重平滑度（实际上是不同的:
我们只需要sX,因为权重分布对性能的贡献很小,请参阅
表2）。然而,我们的方法与 SmoothQuant 有着根本的不同。首先,直觉
和机制不同。 SmoothQuant 平衡权重和激活之间的平滑度
因为它们都是使用相同的 INT8 量化器进行量化的。应该只考虑重量
我们的设置中的分布（仅量化权重）,而我们发现激活分布
对于权重量化至关重要,但对于权重分布则不然（表2）。其次,我们是
专注于低位仅权重量化,其中 SmoothQuant 公式无法实现不错的效果
表现。以LLaMA-7B INT3-g128量化为例:默认SmoothQuant
(α=0.5) 达到 11.55 WikiText-2 困惑度。使用α = 0 （将量化难度从
激活权重)实际上使 12.00 时的困惑度变得更糟,这与 的故事相矛盾
SmoothQuant 与 RTN 基线相同,为 12.10,而我们的方法可以达到 11.07。

4 相关工作

模型量化方法。量化降低了深度学习模型的位精度[21，
24,35,49,34,30],这有助于减小模型大小并加速推理。量化



(a) 我们的方法需要较小的校准集

评估 / 卡利布	通用PTQ			我们的		
	考研	安然		考研	安然	
考研	32.48	50.41	+4.89	32.56	45.07	+0.50
安然公司	+2.33	34.81	45.52	+0.60	33.16	44.57

(b) 我们的方法对校准集分布更稳健

图 7. 左:AWQ 需要更小的校准集才能达到良好的量化性能。与 GPTQ 相比,它可以使用小 10 倍的校准集来实现更好的困惑度。右图:我们的方法对于校准集分布更加稳健。总体而言,使用相同的校准和评估分布效果最好（PubMed-PubMed、Enron-Enron）。但当使用不同的校准分布（PubMed-Enron、Enron-PubMed)时,AWQ 仅增加了 0.5-0.6 的困惑度,而 GPTQ 的困惑度则较差 2.3-4.9。所有实验均在 INT3-g128 量化下使用 OPT-6.7B 模型完成。

技术通常分为两类:量化感知训练（QAT,依靠反向传播来更新量化权重)[3,20,36,9]和训练后量化[24,35,34]（PTQ,通常训练-自由的）。 QAT 方法无法轻松扩展到像法学硕士这样的大型模型。因此,人们通常使用PTQ方法来量化LLM。

LLM 的量化。人们研究LLM量化的两种设置：(1)W8A8量化,其中激活和权重都量化为INT8[12,54,55,53,51]； (2)低位仅权重量化(例如,W4A16),其中仅权重被量化为低位整数[17,13,44,39]。

我们重点关注这项工作中的第二个设置,因为它不仅减少了硬件障碍（需要较小的内存大小）,而且还加快了令牌生成速度（弥补了内存限制的工作负载）。
除了普通的舍入到最近基线（RTN)之外,GPTQ [17] 是最接近我们的工作的。
然而,GPTQ 的重建过程会导致校准集的过度拟合问题,并且可能无法保留法学硕士对其他模式和领域的通才能力。它还需要重新排序技巧才能适用于某些模型（例如 LLaMA-7B [47] 和 OPT-66B [58]）,从而导致硬件效率低下的不规则内存访问并减慢推理速度。

对低位量化 LLM 的系统支持。低位量化法学硕士一直是降低推理成本的流行设置。有一些系统支持可以实现实际的加速。

GPTQ [17] 为 OPT 模型提供 INT3 内核,GPTQ-for-LLaMA 在 Triton 编译器 [46] 的帮助下扩展了对 INT4 重新排序量化的内核支持。 FlexGen [44] 和 llama.cpp ‡ 执行分组 INT4 量化以减少 I/O 成本和卸载。 Faster Transformer § 实现了 FP16×INT4 GEMM,用于仅权重的每张量量化,但不支持组量化。 LUT-GEMM [39] 在查找表的帮助下在 GPU CUDA 核心上执行按位计算。我们的 AWQ 内核在张量核心上执行,适用于 LLM 推理中的上下文和生成阶段,并且不需要硬件效率低下的重新排序。

因此,在运行 LLaMA 模型时,我们的内核比最好的竞争对手快 1.45 倍。

5 结论

在这项工作中,我们提出了激活感知权重量化（AWQ）,这是一种简单而有效的低位仅权重 LLM 压缩方法。 AWQ 基于权重在 LLM 中并不同等重要的观察,并执行每通道缩放以减少显着权重的量化损失。 AWQ 不会过度拟合校准集,并保留了法学硕士在各个领域和模式中的通才能力。它优于语言建模和 QA 基准方面的现有工作,并且可适用于指令调整的 LM 和多模态 LM。 AWQ 还具有硬件效率,无需不规则的内存访问。我们进一步实现了高效的内核,比 GPTQ 实现了 1.45 倍加速,比 cuBLAS FP16 实现实现了 1.85 倍加速。

致谢

我们感谢麻省理工学院人工智能硬件计划、国家科学基金会、NVIDIA 学术合作伙伴奖、麻省理工学院-IBM 沃森人工智能实验室、亚马逊和麻省理工学院科学中心、高通创新奖学金、微软图灵学术计划对这项研究的支持。

‡https://github.com/ggerganov/llama.cpp
§https://github.com/NVIDIA/FasterTransformer

参考

[1] Jean-Baptiste Alayrac、Jeff Donahue、Pauline Luc、Antoine Miech、Iain Barr、Yana Hasson、Karel Lenc、 Arthur Mensch、Katherine Millican、Malcolm Reynolds 等。 Flamingo：用于小样本学习的视觉语言模型。神经信息处理系统的进展，35：23716–23736，2022 年。

[2] Anas Awadalla、Irena Taka、Joshua Gardner、Jack Hessel、Yusuf Hanafy、朱万荣、Kalyani Marathe、 Yonatan Bitton、Samir Gadre、Jenia Jitsev、Simon Kornblith、Pang Wei Koh、Gabriel Ilharco、Mitchell Wortsman 和 Ludwig Schmidt。火烈鸟公开赛，2023 年 3 月。

[3]约书亚·本吉奥、尼古拉斯·莱昂纳德和亚伦·库维尔。通过随机神经元估计或传播梯度以进行条件计算。 arXiv 预印本 arXiv:1308.3432, 2013。

[4] Yonatan Bisk、Rowan Zellers、Ronan Le Bras、高剑峰和 Yejin Choi。 Piqa：用自然语言推理物理常识。 2020 年第三十四届 AAAI 人工智能大会。

[5]汤姆·布朗、本杰明·曼、尼克·莱德、梅兰妮·苏比亚、贾里德·D·卡普兰、普拉富拉·达里瓦尔、阿尔文德·尼拉坎坦、普拉纳夫·希亚姆、吉里什·萨斯特里、阿曼达·阿斯克尔、桑迪尼·阿加瓦尔、阿里尔·赫伯特-沃斯、格雷琴·克鲁格、汤姆·赫尼汉、Rewon Child 、Aditya Ramesh、Daniel Ziegler、Jeffrey Wu、Clemens Winter、Chris Hesse、Mark Chen、Eric Sigler、Mateusz Litwin、Scott Gray、Benjamin Chess、Jack Clark、Christopher Berner、Sam McCandlish、Alec Radford、Ilya Sutskever 和 Dario Amodei 。语言模型是小样本学习者。 H. Larochelle、M. Ranzato、R. Hadsell、MF Balcan 和 H. Lin 编辑，《神经信息处理系统进展》，第 33 卷，第 1877-1901 页。 Curran Associates, Inc.，2020。

[6] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell 等。语言模型是小样本学习者。
神经信息处理系统的进展，33：1877-1901，2020。

[7]陈鑫雷、方浩、林宗毅、Ramakrishna Vedantam、Saurabh Gupta、Piotr Dollár 和 C Lawrence Zitnick。 Microsoft coco 说明：数据收集和评估服务器。 arXiv 预印本arXiv:1504.00325, 2015。

[8]蒋伟林、李卓涵、林子、盛英、吴张浩、张浩、郑联民、庄思源、庄永浩、Joseph E. Gonzalez、Ion Stoica 和 Eric P. Xing。 Vicuna：一款开源聊天机器人，以 90%* chatgpt 质量给 gpt-4 留下深刻印象，2023 年 3 月。

[9] 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。

[10] Hyung Won Chung、Le Hou、Shayne Longpre、Barret Zoph、Yi Tay、William Fedus、Eric Li、Xuezhi Wang、Mostafa Dehghani、Siddhartha Brahma 等。扩展指令微调语言模型。 arXiv预印本 arXiv:2210.11416, 2022。

[11] Peter Clark、Isaac Cowhey、Oren Etzioni、Tushar Khot、Ashish Sabharwal、Carissa Schoenick 和 Oyvind Tafjord。您认为您已经解决了问答问题吗？尝试 arc，ai2 推理挑战赛。 arXiv 预印本arXiv:1803.05457, 2018。

[12]蒂姆·德特默斯、迈克·刘易斯、尤尼斯·贝尔卡达和卢克·泽特尔莫耶。 Llm.int8()：大规模 Transformer 的8 位矩阵乘法。 arXiv 预印本 arXiv:2208.07339, 2022。

[13]蒂姆·德特默斯和卢克·泽特尔莫耶。 4 位精度的情况：k 位推理缩放定律。 arXiv
预印本 arXiv：2212.09720，2022。

[14] Danny Driess，Fei Xia，Mehdi SM Sajjadi，Corey Lynch，Aakanksha Chowdhery，Brian Ichter，Ayzaan Wahid，Jonathan Tompson，Quan Vuong，Tianhe Yu，等。 Palm-e：一种具体的多模式语言模型。 arXiv 预印本 arXiv:2303.03378, 2023。

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

[16]乔纳森·弗兰克尔和迈克尔·卡宾。彩票假设：寻找稀疏、可训练的神经网络。 arXiv 预印本 arXiv:1803.03635, 2018。

[17] Elias Frantar、Saleh Ashkboos、Torsten Hoefler 和 Dan Alistarh。 Gptq：准确的训练后量化
用于生成预训练的变压器。 arXiv 预印本 arXiv:2210.17323, 2022。

[18] Leo Gau、Stella Biderman、Sid Black、Laurence Golding、Travis Hoppe、Charles Foster、Jason Phang、 Horace He、Anish Thite、Noa Nabeshima 等。该堆：用于语言建模的 800GB 不同文本数据集。 arXiv 预印本 arXiv:2101.00027, 2020。

[19] Leo Gau、Jonathan Tow、Stella Biderman、Sid Black、Anthony DiPofi、Charles Foster、Laurence Golding、Jeffrey Hsu、Kyle McDonell、Niklas Muennighoff、Jason Phang、Laria Reynolds、Eric Tang、Anish Thite、Ben Wang、Kevin Wang和安迪·邹。少量语言模型评估框架， 2021 年 9 月。

[20] Amir Gholami、Sehoon Kim、Zhen Dong、Zhewei Yao、Michael W Mahoney 和 Kurt Keutzer。有效神经网络推理的量化方法的调查。 arXiv 预印本 arXiv:2103.13630, 2021。

[21] 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。 ICLR,2016 年。

[22] 宋瀚、杰夫·普尔、约翰·特兰和威廉·达利。学习有效神经网络的权重和连接。神经信息处理系统的进展,2015 年 28 日。

[23] Dan Hendrycks、Collin Burns、Steven Basart、Andy Zou、Mantas Mazeika、Dawn Song 和 Jacob Steinhardt。测量大规模多任务语言理解。 CoRR,abs/2009.03300,2020。

[24] Benoit Jacob、Skirmantas Kligys、Bo Chen、Menglong Zhu、Matthew Tang、Andrew Howard、Hartwig Adam 和 Dmitry Kalenichenko。神经网络的量化和训练,以实现高效的纯整数算术推理。 IEEE 计算机视觉和模式识别会议论文集,第 2704-2713 页,2018 年。

[25]布莱恩·克里姆特和杨一鸣。安然语料库:用于电子邮件分类研究的新数据集。机器学习:ECML 2004:第 15 届欧洲机器学习会议,意大利比萨,2004 年 9 月 20-24 日。会议记录 15,第 217-226 页。施普林格,2004。

[26]Jing Yu Koh、Ruslan Salakhutdinov 和 Daniel Fried。将语言模型基础到图像以进行多模态生成。 arXiv 预印本 arXiv:2301.13823, 2023。

[27] 李俊楠,李东旭,Silvio Savarese,Steven Hoi。 Blip-2:使用冻结图像编码器和大型语言模型引导语言图像预训练。 arXiv 预印本 arXiv:2301.12597, 2023。

[28] 李宇航,龚瑞浩,谭旭,杨阳,彭虎,张琪,于凤伟,王伟,谷石。 Brecq:通过块重建突破训练后量化的极限。 arXiv 预印本 arXiv:2102.05426, 2021。

[29] Percy Liang、Rishi Bommasani、Tony Lee、Dimitris Tsipras、Dilara Soylu、Michihiro Yasunaga、Yian 张、Deepak Narayanan、Yuhuai Wu、Ananya Kumar 等。语言模型的整体评估。 arXiv 预印本 arXiv:2211.09110, 2022。

[30] 吉林,陈伟明,林玉君,甘闯,韩松,等。 Mcunet:物联网设备上的微型深度学习。神经信息处理系统的进展,33:11711–11722,2020。

[31] 刘浩天,李春元,吴庆阳,李勇杰。视觉指令调整。 2023 年。

[32] 刘子昌、王珏、Tri Dao、周天一、袁斌航、赵松、Anshumali Shrivastava、张策、田远东、Christopher Re、陈贝迪。 Deja Vu:推理时间内高效法学硕士的上下文稀疏性。在 ICML,2023 年。

[33] 斯蒂芬·梅里蒂、蔡明熊、詹姆斯·布拉德伯里和理查德·索彻。指针哨兵混合模型,2016年。

[34] 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。

[35] 马库斯·内格尔、Mart van Baalen、Tijmen Blankevoort 和 Max Welling。通过权重均衡和偏差校正进行无数据量化。 IEEE/CVF 国际计算机视觉会议论文集,第 1325-1334 页,2019 年。

[36] Markus Nagel、Marios Fournarakis、Rana Ali Amjad、Yelysei Bondarenko、Mart Van Baalen 和 Tijmen Blankevoort。关于神经网络量化的白皮书。 arXiv 预印本 arXiv:2106.08295, 2021。

[37] 欧阳龙,吴杰弗里,江旭,迪奥戈·阿尔梅达,卡罗尔·温赖特,帕梅拉·米什金,张冲,桑迪尼·阿加瓦尔,卡塔琳娜·斯拉马,亚历克斯·雷,等。训练语言模型遵循人类反馈的指令。神经信息处理系统的进展,35:27730–27744,2022 年。

[38] Denis Paperno、Germán Kruszewski、Angeliki Lazaridou、Ngoc Quan Pham、Raffaella Bernardi、Sandro Pezzelle、Marco Baroni、Gemma Boleda 和 Raquel Fernández。 LAMBADA 数据集:需要广泛的话语上下文的单词预测。计算语言学协会第 54 届年会论文集(第一卷:长论文),第 1525-1534 页,德国柏林,2016 年 8 月。

计算语言学协会。

[39] Gunho Park、Baeseong Park、Se Jung Kwon、Byeongwook Kim、Youngjoo Lee 和 Dongsoo Lee。 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, 2022。

[40] qwopqwop200。骆驼的 Gptq。 https://github.com/qwopqwop200/GPTQ-for-LLaMa,2023 年。

[41] Keisuke Sakaguchi、Ronan Le Bras、Chandra Bhagavatula 和 Yejin Choi。 Winogrande:大规模的对抗性 winograd 模式挑战。 arXiv 预印本 arXiv:1907.10641, 2019。

[42] Victor Sanh、Albert Webson、Colin Raffel、Stephen H Bach、Lintang Sutawika、Zaid Alyafeai、Antoine Chaffin、Arnaud Stiegler、Teven Le Scao、Arun Raja 等。多任务提示训练可实现零样本任务泛化。 arXiv 预印本 arXiv:2110.08207, 2021。

[43] Teven Le Scao、Angela Fan、Christopher Akiki、Ellie Pavlick、Suzana Ilic、Daniel Hesslow、Roman Castagné、Alexandra Sasha Luccioni、François Yvon、Matthias Gallé 等。 Bloom：176b 参数的开放获取多语言语言模型。 arXiv 预印本 arXiv:2211.05100, 2022。

[44] 盛英，郑联民，袁斌航，李卓瀚，Max Ryabinin，Daniel Y Fu，谢志强，陈蓓迪，Clark Barrett，Joseph E Gonzalez，等。使用单个 GPU 对大型语言模型进行高吞吐量生成推理。 arXiv 预印本 arXiv:2303.06865, 2023。

[45] Rohan Taori、Ishaan Gulrajani、Tianyi 张、Yann Dubois、Xuechen Li、Carlos Guestrin、Percy Liang 和 Tatsunori B. Hashimoto。斯坦福羊驼：遵循指令的美洲驼模型。 https://github。 com/tatsu-lab/stanford_alpaca，2023。

[46] 菲利普·蒂莱、孔祥宗和大卫·考克斯。 Triton：用于平铺神经网络计算的中间语言和编译器。第三届 ACM SIGPLAN 国际机器学习和编程语言研讨会论文集，第 10-19 页，2019 年。

[47] Hugo Touvron、Thibaut Lavril、Gautier Izacard、Xavier Martinet、Marie-Anne Lachaux、Timothée Lacroix、Baptiste Rozière、Naman Goyal、Eric Hambro、Faisal Azhar 等。 Llama：开放高效的基础语言模型。 arXiv 预印本 arXiv:2302.13971, 2023。

[48] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、 ukasz Kaiser 和 Illia Polosukhin。您所需要的就是关注。神经信息处理系统的进展，2017 年 30 月。

[49] 王宽，刘志坚，林玉君，林吉，韩松。 HAQ：硬件感知自动量化
具有混合精度。在 CVPR，2019 年。

[50] Jason Wei、Maarten Bosma、Vincent Y Zhao、Kelvin Guu、Adams Wei Yu、Brian Lester、Nan Du、Andrew M Dai 和 Quoc V Le。微调的语言模型是零样本学习者。 arXiv 预印本 arXiv:2109.01652, 2021。

[51] 魏秀英，张云晨，李宇航，张相国，龚瑞浩，郭金阳，刘相龙。异常值抑制+：通过等效且最优的移位和缩放对大型语言模型进行精确量化。 arXiv 预印本 arXiv:2304.09145, 2023。

[52] 魏秀英，张云辰，张相国，龚瑞浩，张尚航，张琪，于凤伟，刘相龙。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, 2022。

[53] 魏秀英，张云辰，张相国，龚瑞浩，张尚航，张琪，于凤伟，刘相龙。异常值抑制：突破低位 Transformer 语言模型的极限，2022 年。

[54]肖光轩，林吉，Mickael Seznec，Julien Demouth，宋瀚。 Smoothquant：大型语言模型的准确高效的训练后量化。 arXiv 预印本 arXiv:2211.10438, 2022。

[55] 姚哲伟，Reza Yazdani Aminabadi，张敏佳，吴晓霞，李从龙，何宇雄。
Zeroquant：大规模 Transformer 的高效且经济实惠的训练后量化，2022 年。

[56] Rowan Zellers、Ari Holtzman、Yonatan Bisk、Ali Farhadi 和 Yejin Choi。 Hellaswag：机器可以吗
真的把话说完吗？ CoRR，abs/1905.07830，2019。

[57] 张仁瑞，韩家明，周傲君，胡翔飞，严世林，潘路，李红生，高鹏，乔宇。 Llama-adapter：零初始化注意力的语言模型的高效微调。 arXiv 预印本 arXiv:2303.16199, 2023。

[58] Susan 张、Stephen Roller、Naman Goyal、Mikel Artetxe、Moya Chen、Shuohui Chen、Christopher Dewan、Mona Diab、Xian Li、Xi Victoria Lin、Todor Mihaylov、Myle Ott、Sam Shleifer、Kurt Shuster、Daniel Simig、Punit Singh Koura、Anjali Sridhar、Tianlu Wang 和 Luke Zettlemoyer。 Opt：开放预训练的 Transformer 语言模型，2022 年。

更广泛的影响和限制

更广泛的影响。在本文中,我们提出了一种通用技术来实现大型语言模型（LLM)的准确且高效的低位仅权重量化。它使法学硕士更加高效、更容易获得,因此可以继承法学硕士的影响。从积极的一面来看,量化有助于法学硕士的民主化,这有助于让更多人受益（尤其是那些收入较低的人）。它降低了部署 LLM 的成本和硬件障碍,并促进这些模型的边缘推理,解决了数据隐私问题（因为我们不再需要将数据发送到云端）。不利的一面是,法学硕士可能会被恶意用户利用来产生错误信息和操纵。量化不能防止这种负面影响,但也不会使其变得更糟。

局限性。在本文中,我们遵循之前的工作 [12,17,54,55,13],主要根据困惑度和准确度等标准准确度指标对量化模型进行基准测试。然而,除了准确性之外,LLM 基准还有其他重要指标,如稳健性、公平性、偏差、毒性、有用性、校准等[29]。

我们认为,对涵盖这些方面的量化法学硕士进行更全面的评估将很有帮助,我们将其留给未来的工作。此外,由于硬件上的数据类型转换更容易,我们只研究 LLM 的低位整数量化。更改数据类型（例如 FP4 [13])可能会带来进一步的改进,但我们没有将其纳入研究中。

B 计算量

我们在这项工作中研究了法学硕士的训练后量化（PTQ）。由于我们不依赖任何反向传播,因此计算要求通常较低。由于内存限制,我们对较小模型（<40B 参数)使用 1 个 NVIDIA A100 GPU,对较大模型使用 2-4 个 A100 GPU。

量化过程通常很快,需要几个 GPU 小时（范围从 0.1 到 3,具体取决于模型大小）。准确度测量时间取决于模型和数据集大小:在 4 个常识 QA 任务上测试 LLaMA-65B（我们在多个数据集上测试的最大模型)需要 3 个 GPU 小时;在 MMLU（由 57 个子数据集组成)上进行测试需要 5 个 GPU 小时。对于较小的模型和数据集（例如,WikiText-2）,GPU 时间会更短。

无群量化的 C 限制

我们的方法寻找良好的缩放以保护显着的权重通道。它在分组量化下工作得很好,与在 FP16 中保持显着权重具有相同的精度（图 1）。然而,这种基于缩放的方法只能保护每一组的一个显着通道。对于分组量化来说这不是问题（我们只需要保护0.1%-1%的显着通道,组的大小通常很小,比如128,所以我们平均每组需要保护的通道少于1个）。但对于无组量化,我们只能保护整个权重的一个输入通道,这可能不足以弥补性能下降。如表 8 所示,在 INT3-g128 量化下,AWQ 与在 FP16 中保持 1% 显着权重相比实现了相似的性能。而在INT3无群量化下,仍然存在明显的差距。尽管如此,我们要强调的是,在类似成本下,无分组量化的性能仍然远远落后于分组量化。因此,分组量化是边缘部署LLM压缩更实用的解决方案,AWQ可以有效提高该设置下的量化性能。

PPL ↓	FP16	INT3（第 128 组）		INT3（无组）	
		RTN	1% FP16 AWQ	RTN	1% FP16 AWQ
OPT-6.7B	12.29	43.16	13.02	12.99 21160	14.67 18.11
LLaMA-7B	9.49	12.10	10.77	10.82 50.45	14.06 20.52

表 8. AWQ 可以与分组量化下在 FP16 中保持 1% 显着权重的性能相匹配,而无需引入混合精度,但不适用于无分组量化。尽管如此,与无分组相比,分组量化具有更好的性能,这使其成为 LLM 的仅权重量化更实用的设置,而 AWQ 在此设置下表现得相当好。结果在 WikiText-2 数据集上令人困惑。

D OPT 模型在 MMLU 上的性能

在本文中,我们只报告了 LLaMA 模型 [47] 在 MMLU [23] 上的准确性,而不是 OPT 模型 [58]。
这是因为我们发现 OPT 模型在基准测试中通常表现不佳。如表9所示,与类似规模的LLaMA模型相比,OPT模型的准确性要差得多。实际上,13B OPT 模型的表现不如 7B LLaMA 模型。考虑到 MMLU 上的完全随机基线为 25%,OPT 模型的准确性几乎没有意义。此外,
OPT 模型的情境学习有限

能力（通过 5 次射击与 0 次射击的 ΔAcc. 进行测量）。因此,我们认为对标OPT
MMLU 上的模型可能不会提供非常有意义的结果。

	MMLU（0 次）	MMLU（5 次）	
OPT-6.7B	27.04%	27.90%	+0.86%
LLaMA-7B	30.78%	38.41%	+7.68%
OPT-13B	27.70%	29.58%	+1.88%
LLaMA-13B	33.29%	45.21%	+11.92%

表 9. OPT 模型在 MMLU 数据集上的准确性有限。 MMLU 上的基准测试 OPT 可能不会
提供有意义的结果。因此,我们仅在 WikiText-2 数据集上对 OPT 模型进行基准测试。