泛在计算与智能系统研究中心
Research Center of Ubiquitous Computing and Intelligent Systems
哈爾濱工業大學
Harbin Institute of Technology

AWQ: ACTIVATION-AWARE W EIGHT QUANTIZATION FOR ON-DEVICE LLM COMPRESSION AND ACCELERATION

激活感知权重量化用于设备端大型语言模型的压缩与加速

Ji Lin*1 Jiaming Tang*12 Haotian Tang†1 Shang Yang†1 Wei-Ming Chen3 Wei-Chen Wang1 Guangxuan Xiao1 Xingyu Dang14 Chuang Gan56 Song Han1

The 7th MLSys Conference Location: Santa Clara, California, USA Date: 2024

汇报人姓名：毛杉文

汇报日期：2024.11.19

目录

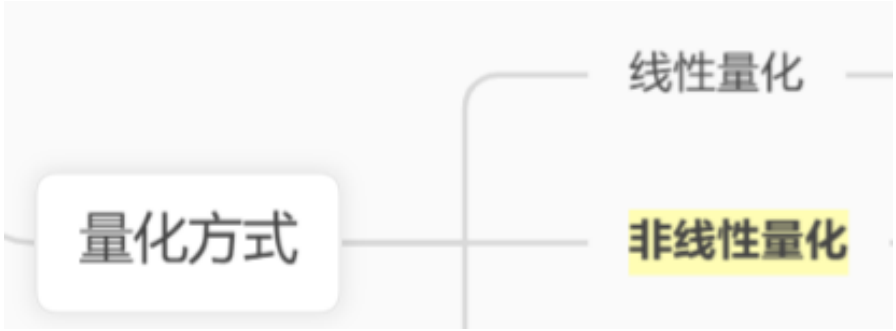
- 背景----大模型量化背景
- 动机----实现4比特以下的量化
- 方法----线性量化
- 效果----4比特量化

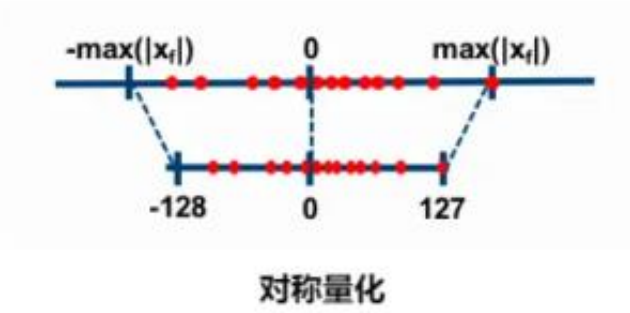
背景

$$s = \frac{\max(|w|)}{q_{\max}}$$

$$q = \text{round}\left(\frac{w}{s}\right)$$

$$w' = q \times s$$





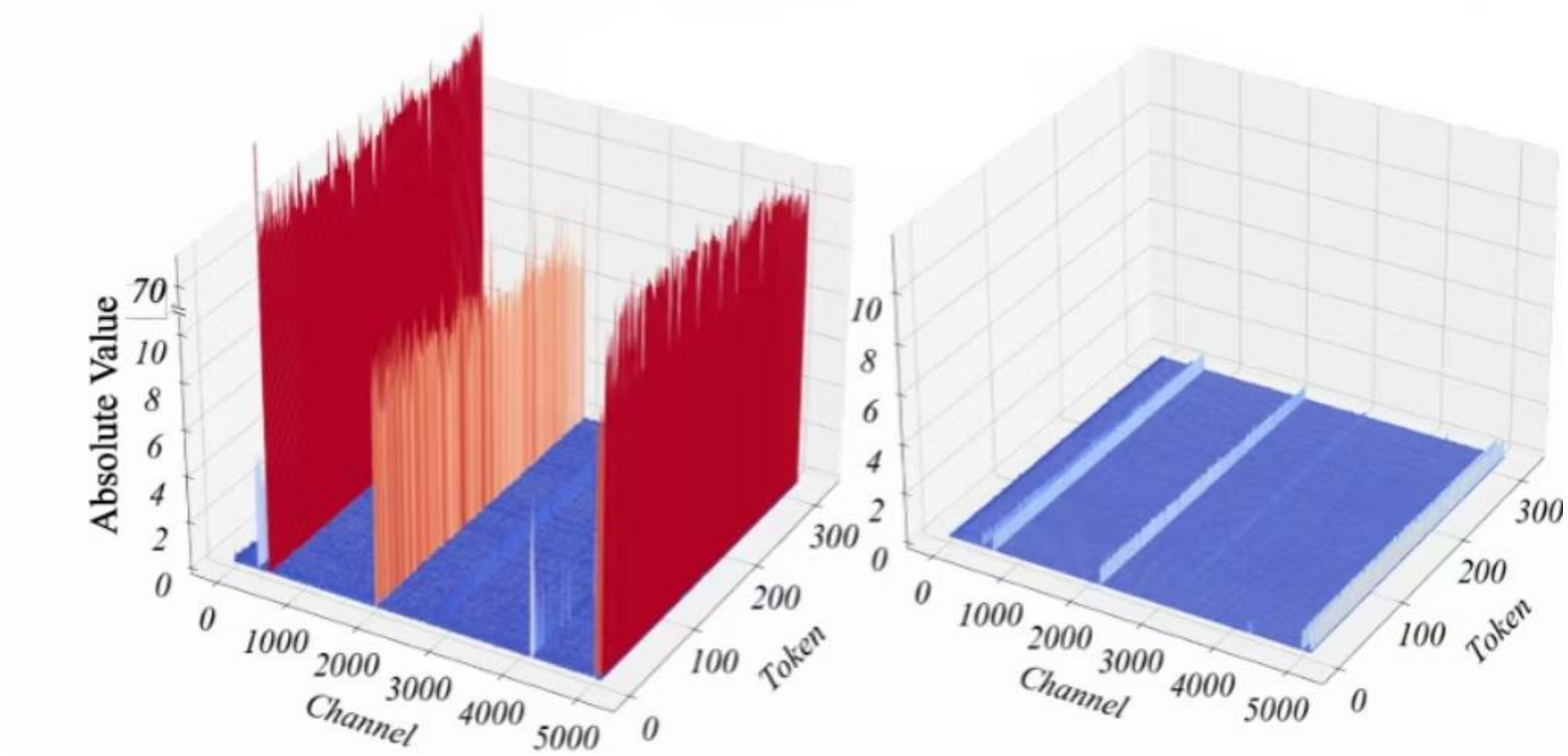
对称量化

$$O = W \cdot X$$

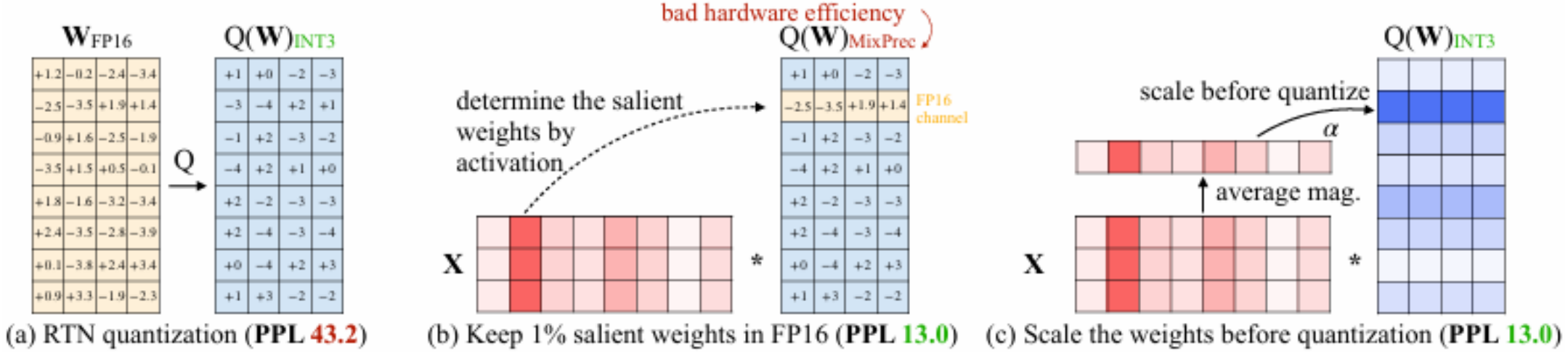
动机

AWQ发现，在LLM中，并非所有权重都同等重要。只需保护1%的显著权重，就能大幅降低量化误差。

为了识别显著权重通道，我们应参考激活分布，而非权重。



方法--通过激活感知缩放保护显著权重



(a) RTN quantization (**PPL 43.2**) (b) Keep 1% salient weights in FP16 (**PPL 13.0**) (c) Scale the weights before quantization (**PPL 13.0**)

$$Q(w) = \Delta \cdot \text{Round}\left(\frac{w}{\Delta}\right), \Delta = \frac{\max(|w|)}{2^N - 1}$$

$$\text{Err}(Q(w)x) = \Delta \cdot \text{RoundErr}\left(\frac{w}{\Delta}\right) \cdot x$$

$$Q(w \cdot s)(\frac{x}{s}) = \Delta' \cdot \text{Round}\left(\frac{ws}{\Delta'}\right) \cdot x \cdot \frac{1}{s}$$

$$\text{Err}(Q(w \cdot s)\left(\frac{x}{s}\right)) = \Delta' \cdot \text{RoundErr}\left(\frac{ws}{\Delta'}\right) \cdot x \cdot \frac{1}{s}$$

$$\frac{\Delta'}{\Delta} \cdot \frac{1}{s}$$

方法--缩放因子的选择

不同缩放因子时候模型的性能

OPT-6.7B	$s=1$	$s=1.25$	$s=1.5$	$s=2$	$s=4$
proportion of $\Delta^{'} \neq \Delta$	0%	2.8%	4.4%	8.2%	21.2%
average $\Delta^{'}/\Delta$	1	1.005	1.013	1.038	1.213
average $\frac{\Delta^{'}}{\Delta} \cdot \frac{1}{s}$	1	0.804	0.676	0.519	**0.303**
Wiki-2 PPL	23.54	12.87	12.48	**11.92**	12.36

$$s^* = \arg\min_s L(s)$$

$$L(s) = \|Q(W \cdot \mathrm{diag}(s))(\mathrm{diag}(s)^{-1} \cdot X) - WX\|$$

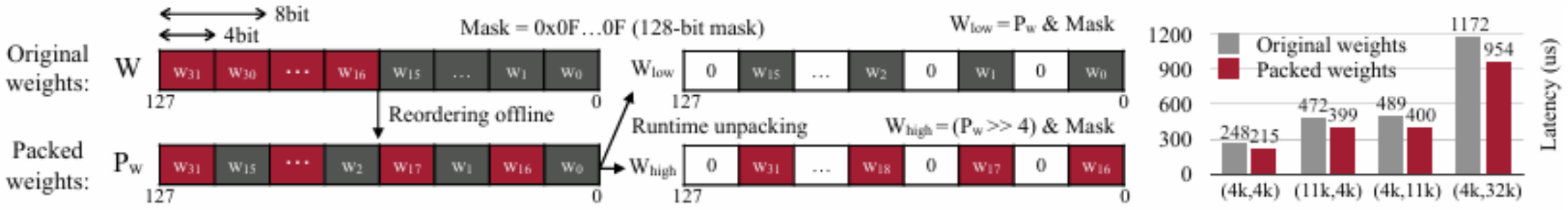
$$s = s_X^{\alpha}, \alpha^* = \arg\min_{\alpha} L(s_X^{\alpha})$$

其中 s_X 是激活的平均幅度（每通道），我们使用单个超参数α来平衡显著和非显著通道的保护。我们可以通过快速网格搜索在[0,1]区间内找到最佳 α（0表示不缩放；1对应于搜索空间中最激进的缩放）。

方法—内核融合加速

我们通过将反量化内核与矩阵乘法内核融合，避免了将反量化后的权重写入DRAM。

TinyChat系统通过将层归一化的所有操作（如乘法、除法和平方根）融合到一个内核中。

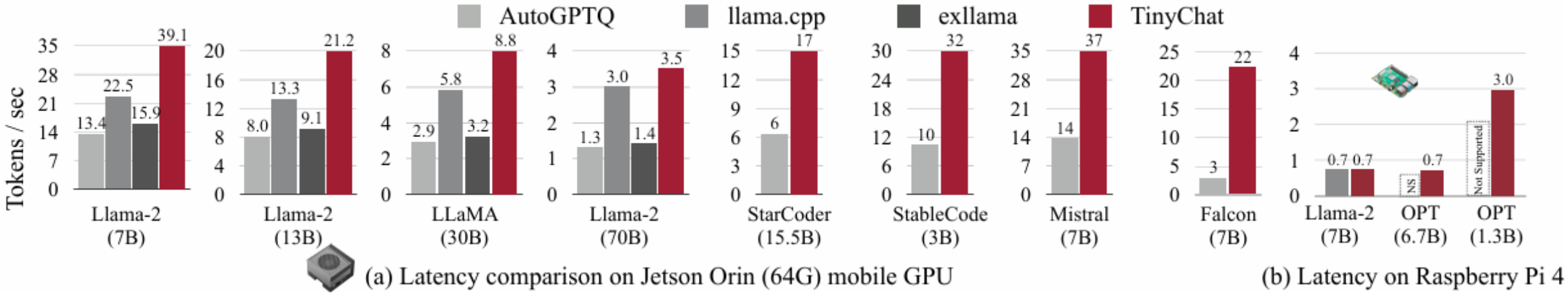


针对具有128位SIMD单元的ARM NEON的SIMD感知权重打包。原始权重被重新排序并打包，以便与位宽对齐，从而使权重可以在运行时使用AND运算和位移操作与128位掩码进行解包

效果

PPL↓		Llama-2			LLaMA			
		7B	13B	70B	7B	13B	30B	65B
FP16	-	5.47	4.88	3.32	5.68	5.09	4.10	3.53
INT3 g128	RTN	6.66	5.52	3.98	7.01	5.88	4.88	4.24
	GPTQ	6.43	5.48	3.88	8.81	5.66	4.88	4.17
	GPTQ-R	6.42	5.41	3.86	6.53	5.64	4.74	4.21
	AWQ	**6.24**	**5.32**	**3.74**	**6.35**	**5.52**	**4.61**	**3.95**
INT4 g128	RTN	5.73	4.98	3.46	5.96	5.25	4.23	3.67
	GPTQ	5.69	4.98	3.42	6.22	5.23	4.24	3.66
	GPTQ-R	5.63	4.99	3.43	5.83	5.20	4.22	3.66
	AWQ	**5.60**	**4.97**	**3.41**	**5.78**	**5.19**	**4.21**	**3.62**

AWQ在不同模型规模和不同位精度下，相较于四舍五入量化（RTN）有所改善。在LLaMA和Llama-2模型上，它始终比GPTQ（包含和不包含重排序）取得更好的困惑度。



TinyChat在NVIDIA Jetson Orin上运行4位量化的Llama模型时，相较于现有系统提供了1.2-3.0倍的速度提升。它还支持多种通用和针对编码的LLM，并在处理这些工作负载时至少比AutoGPTQ快2.6倍。

谢谢大家