LSQ+

> LSQ+: Improving low-bit quantization through learnable offsets and better initialization

文章思路

1. **背景**：
   - 低位量化（2、3、4位）对高效神经网络推理至关重要，尤其是在资源受限的设备上。
   - 传统量化方法假定激活函数为非负（例如ReLU），但现代高效网络（如EfficientNet、MixNet）使用了Swish等具有负值输出的激活函数，导致现有方法难以有效量化。
2. **问题**：
   - **负激活值处理不足**：现有方法对负激活值直接置零（Unsigned Quantization）或均等处理正负范围（Signed Quantization），导致信息损失或精度降低。
   - **训练不稳定**：基于梯度的量化方法对参数初始化高度敏感，尤其在低位量化中。
3. **改进**：
   - 提出LSQ+方法，通过**可学习的偏移量（offset）**和改进的初始化策略，解决负值处理不足的问题。
   - 通过MSE优化初始化量化参数，显著提高训练稳定性。

与之前方法的对比

- **LSQ**:
  - 仅使用对称量化，偏向ReLU激活函数，忽略了负值范围的重要性。
  - 采用简单的平方均值初始化参数，未考虑分布统计特性。
- **LSQ+**:
  - 引入偏移量β，实现非对称量化，更好地适应非对称分布（如Swish激活）。
  - 改进初始化方案，降低训练结果的方差，提高稳定性。

具体小节内容分析

1. 引言

介绍了低位量化的重要性及其在现代架构中的挑战，特别是对负激活值的处理和训练稳定性问题。强调了LSQ+在EfficientNet等网络上的优越性能。

2. 相关工作

分类讨论了两种量化方法：

- **后训练量化**：无需微调，仅适用于高位（8位）量化。
- **量化感知训练**：适合低位量化，但需要较长训练时间。
  同时讨论了知识蒸馏和位宽学习的研究，但这些方法与LSQ+是正交的，可联合使用。

3. 方法

1. **LSQ+的非对称量化方案**：
   - 公式：

$$\bar{x} = \left\lfloor \text{clamp}\left(\frac{x-\beta}{s}, n, p\right) \right\rceil, \quad \hat{x} = \bar{x} \cdot s + \beta \tag{1}$$

     - $\bar{x}$：量化后的编码值。
     - $\hat{x}$：反量化后的值。
     - s、β：分别为可学习的尺度和偏移参数。
   - 偏导数公式：

$$\frac{\partial \hat{x}}{\partial s} = \begin{cases} -\frac{x-\beta}{s} + \left\lfloor \frac{x-\beta}{s} \right\rceil & n < \frac{x-\beta}{s} < p \\ n \text{ 或 } p & \text{否则} \end{cases} \tag{2}$$

$$\frac{\partial \hat{x}}{\partial \beta} = \begin{cases} 0 & n < \frac{x-\beta}{s} < p \\ 1 & \text{否则} \end{cases} \tag{3}$$

   - 使用直通估计（STE）近似不可导部分。
   - 权重量化仍采用对称量化，无额外推理成本。
2. **初始化策略**：
   - **激活量化**：
     - 基于MSE优化，解决传统min-max法对异常值敏感的问题。

$$s_{\text{init}}, \beta_{\text{init}} = \arg\min_{s,\beta} ||\hat{x} - x||_F^2 \tag{4}$$

- **权重量化**：
  - 通过分布统计信息（均值μ和标准差σ）初始化：

$$s_{\text{init}} = \frac{\max(|\mu - 3\sigma|, |\mu + 3\sigma|)}{2^b - 1} \tag{5}$$

4. 实验

1. **Swish激活的量化**：
   - 表2展示了EfficientNet-B0在W4A4条件下的精度从71.9%（LSQ）提高到73.8%（LSQ+）。
   - 可学习偏移量使正负范围的量化层次更均衡。
2. **ReLU激活的量化**：
   - 对传统架构（ResNet18）的实验表明LSQ+与LSQ性能一致，未引入性能开销。
3. **初始化影响**：
   - 表5显示LSQ+初始化在低位量化中显著降低了训练不稳定性。
4. **固定偏移与可学习偏移对比**：
   - 表6表明可学习偏移优于固定偏移，提高了表示精度。

公式分析

1. **核心公式**：
   - 非对称量化（公式2）：为不同层动态调整范围，缓解负值带来的信息丢失。

- 偏移量和尺度的梯度计算（公式3、4）：保证训练过程的有效优化。

2. **性能优化公式**：
   - MSE优化初始化（公式8）：通过数据分布初始化参数，避免因极值导致的量化错误。

总结

LSQ+通过学习偏移量和改进的初始化方法，在低位量化中实现了更高的性能和稳定性。这些改进使其特别适合现代网络的Swish激活，显著超越了传统的LSQ方法。

补充

非对称量化方案和对称方案

非对称量化方案和**对称量化方案**是深度学习模型量化的两种主要策略，它们在量化范围的设计和实现方法上存在显著差异。

---

1. 对称量化方案

定义

对称量化假设被量化的值（如权重或激活值）的分布是对称的，其量化范围均匀地分布在零的两侧。

公式

对称量化的公式为：

$$x_q = \text{round}\left(\frac{x}{s}\right) \cdot s \tag{6}$$

其中：

$$s = \frac{x_{\max} - x_{\min}}{2^{b-1} - 1} \tag{7}$$

- s 是量化的尺度（scale）。
- $x_{\max}$ 和 $x_{\min}$ 分别是值的最大值和最小值。
- b 是比特宽度。

特点

1. **量化范围**：
   - 对称分布在 $[-s \cdot (2^{b-1} - 1), s \cdot (2^{b-1} - 1)]$。
   - 例如，4 位对称量化范围为 $[-8, 7]$，共 16 个值。
2. **实现简单**：
   - 不需要额外的偏移量，计算开销较低。
   - 适用于分布接近对称的权重或激活值（如 ReLU 激活输出）。
3. **不足**：
   - 如果数据分布不对称（如 Swish 激活），可能导致量化范围的一部分浪费（例如，负值部分很小但分配了较多的量化级别）。

2. 非对称量化方案

定义

非对称量化允许量化范围不以零为中心，通过引入偏移量 β 适应数据分布的不对称性。

公式

非对称量化的公式为：

$$x_q = \text{round}\left(\frac{x - \beta}{s}\right) \cdot s + \beta \tag{8}$$

其中：

$$s = \frac{x_{\max} - x_{\min}}{2^b - 1} \tag{9}$$

- s 是量化的尺度。
- β 是量化偏移量（offset），通常设置为 $x_{\min}$。

特点

1. **量化范围**：
   - 动态分布在 $[x_{\min}, x_{\max}]$。
   - 例如，4 位非对称量化范围可能是 $[-0.3, 15.7]$。
2. **灵活性更高**：

- 适用于分布不对称的权重和激活值（如 Swish 激活函数的分布：$[-0.278, \infty)$）。

3. **实现复杂**：
   - 需要存储额外的偏移量 β。
   - 在推理时可能需要额外的计算调整。

---

3. 对比分析

特性	对称量化方案	非对称量化方案
量化范围	中心对称，固定范围	可动态调整，覆盖数据的真实范围
适用数据分布	对称分布（如 ReLU 激活）	不对称分布（如 Swish 或 Leaky-ReLU 激活）
计算复杂度	较低	略高，需引入偏移量 β
存储需求	无需存储偏移量	需要存储偏移量 β
精度表现	在分布不对称的情况下，量化误差较大	在复杂分布情况下量化精度更高

---

4. 使用场景

对称量化适用场景

- **简单模型**：如 ResNet 等基于 ReLU 的架构。
- **资源受限设备**：对计算复杂度要求较低的硬件（如微控制器）。

非对称量化适用场景

- **现代架构**：如 EfficientNet 和 MixNet，使用 Swish 或 Leaky-ReLU 激活。
- **需要高精度的任务**：如图像分类、目标检测。

---

5. 非对称量化的改进

以 **LSQ+** 为例，提出了学习偏移量 β 的非对称量化方法，显著提升了模型性能。核心思想：

- β 和 s 都在训练过程中通过梯度优化学习：

$$\beta, s = \arg\min ||x_q - x||_F^2 \tag{10}$$

- 通过 MSE 初始化量化参数，减少训练的不稳定性。

这种动态非对称量化方法有效解决了 Swish 激活值分布的不对称性问题，使模型在低比特量化（如 W4A4）下仍保持高性能。

通过对称与非对称量化的灵活选择，可以根据模型特点与硬件限制设计最优量化策略，从而在效率与性能之间找到平衡点。