arXiv:2301.13376v1 [cs.LG] 31 Jan 2023

采用低精度累加并保证避免溢出的量化神经网络

Ian Colbert [1] **Alessandro Pappalardo** [2] **Jakoba Petri-Koenig** [2]

摘要

对神经网络的权重和激活值进行量化，可显著降低推理成本，通常只需牺牲少量模型精度。这在很大程度上得益于卷积和矩阵乘法等运算在计算与存储成本上的节省；这些运算产生的乘积通常会累加到高精度寄存器中，这类寄存器称为累加器。尽管许多研究人员和从业者已开始利用低精度表示来表示模型的权重和激活值，但很少有人关注缩小累加器的尺寸。部分原因在于，使用低精度寄存器进行累加会带来很高的数值溢出风险；由于采用回绕算术，这可能严重降低模型精度。在这项工作中，我们提出了一种量化感知训练算法，可保证在推理期间降低累加器精度时避免数值溢出。我们利用权重归一化在训练期间约束参数，并采用推导出的累加器位宽界限。我们在多个针对不同任务训练的量化模型上评估了该算法，结果表明，相较于浮点基线，我们的方法能够在保持模型精度的同时降低累加器精度。随后，我们证明，这种降低能够提升定制 FPGA 加速器的设计效率。最后，我们表明，该算法不仅能约束权重，使其适配用户指定位宽的累加器，还能提高所得权重的稀疏性和可压缩性。在所有采用 8位权重和激活值训练的基准模型中，我们发现，将量化神经网络的隐藏层限制为适配 16 位累加器，可实现平均 98.2% 的稀疏率，估算压缩率为 46.5 倍，同时保持浮点性能的 99.2%。

1 引言

量化是缩小数据数值表示范围并降低其精度的过程。在用于降低神经网络（NN）推理成本的众多技术中，整数定量化是实践中应用最广泛的技术之一（Gholami 等人，2021年）。低精度量化减少了计算与存储需求，从而提高吞吐量、节省功耗并提升资源效率，通常只需略微牺牲模型准确率（Hubara 等人，2017年）。推理过程中，信息会在神经网络的各层之间传播，而大部分计算工作负载集中在卷积、矩阵乘法等算子的乘加运算（MACs）上。已有研究表明，缩小累加器的位宽可以创造更多机会，从而提高通用处理器的吞吐量和带宽效率以提高并行度（de Bruin 等人，2020年；Ni 等人，2021年；Xie 等人，2021年）。然而，利用这种优化并非易事，因为这样做会带来很高的溢出风险。由于采用回绕二进制补码算术，这可能引入数值误差，并显著降低模型精度（Ni 等人，2021年）。

先前的研究试图降低数值溢出的风险（Xie 等人，2021年；Sakr 等人，2019年），或减轻其对模型精度的影响（Ni 等人，2021年）。在本研究中，我们训练量化神经网络（QNN），使其在推理过程中使用低精度累加器时完全避免数值溢出。为充分利用低精度权重、激活值和累加器所带来的更广阔设计空间，我们将目标设定为在采用定制空间流式数据流的 FPGA 加速器上部署模型，而非部署在 CPU 或 GPU 等通用平台上。FPGA 的灵活性使其成为低精度推理引擎的理想设备，因为它们可以对神经网络的各个部分进行位级控制；权重、激活值和累加器的精度均可针对每一层单独调整为定制数据类型，不受 CPU 或 GPU 上常见的 2 的幂次位宽限制。

[1]AMD 软件技术团队，美国加利福尼亚州圣迭戈 [2]AMD AECG 研究实验室，爱尔兰都柏林。联系：Ian Colbert <ian.colbert@amd.com>。

该 我们的工作贡献总结如下 ows:

- 我们表明，降低累加器的位宽可以减少定制低精度量化神经网络推理加速器的资源利用率。
- 我们推导出了累加器位宽的全面界限，其粒度比现有文献更细。
- 我们提出了一种新颖的量化感知训练（QAT）算法，该算法约束学习到的参数，从而在推理过程中降低累加器的精度时避免数值溢出。
- 我们表明，我们的算法不仅约束权重，使其能够适配用户定义位宽的累加器，还显著提高了所得权重的稀疏性和可压缩性。
- 我们将算法集成到 Brevitas 量化库（Pappalardo，2021年）和 FINN 编译器（AMD-Xilinx，2023a）中，以展示一套端到端流程：面向 AMD-Xilinx FPGA 上的定制流式架构，使用低精度累加器训练和部署量化神经网络。

据我们所知，我们是首个探索使用低精度累加器来提高可编程量化神经网络推理加速器设计效率的团队。不过，我们的研究结果也适用于 FINN 生成的加速器之外的场景。已有研究表明，将累加器位宽限制在用户定义的上界内，可以提高通用处理器的吞吐量和带宽效率（de Bruin 等人，2020年；Ni 等人，2021年；Xie 等人，2021年），并降低同态加密运算的计算开销（Lou 和 Jiang，2019年）。此外，我们的实验表明，与现有方法相比，该算法能够在资源利用率与模型精度之间实现更好的权衡，这证实了将累加器位宽纳入整体软硬件（HW）协同设计空间的益处。

2 相关工作

当激活值在量化神经网络（QNN）的各层中传播时，卷积和矩阵乘法产生的中间部分和通常会先在高精度寄存器中累加，然后重新量化并传递到下一层，如图1所示。尽管许多研究人员和从业者已开始利用权重和激活值的低精度表示（Jacob 等人，2018年；Gholami 等人，2021年；Nagel 等人，2021年；Zhang 等人，2022年），很少有研究关注低精度累加器（Sakr 等人，2019年；de Bruin 等人，2020年；Xie 等人，2021年；Ni 等人，2021年）。

训练量化神经网络以使用低精度累加器的一种方法，是减轻溢出对模型准确率的影响。Xie 等人 2021年试图降低溢出风险——

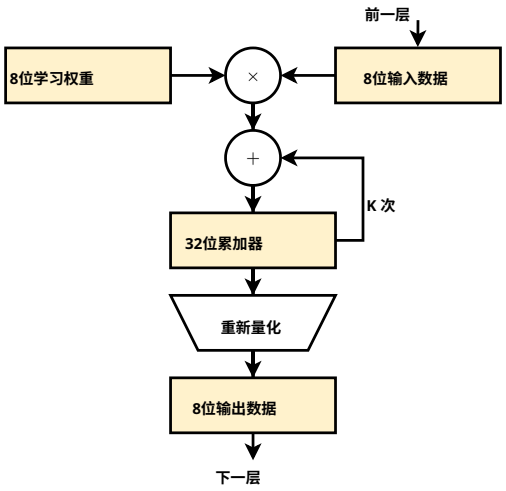
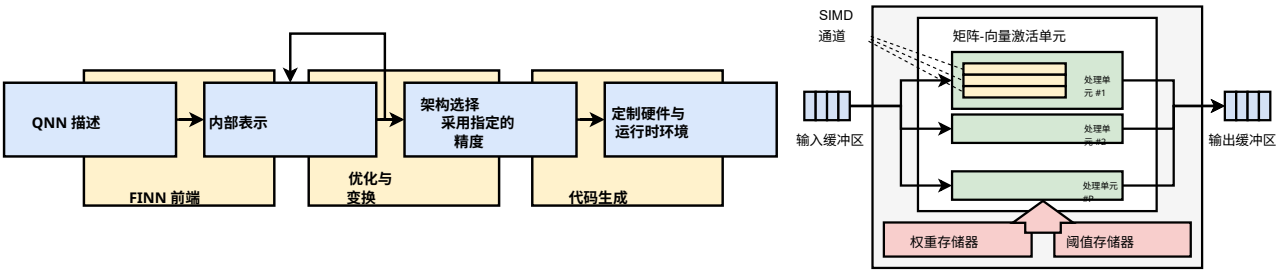


图*1*。 神经网络推理中定点算术的简化示意图。量化权重在推理过程中保持不变。输入/输出数据是动态的，因此会进行缩放和截断，然后隐藏表示（即激活值）在神经网络中传递。累加器需要足够大，以容纳学习得到的权重与输入数据向量之间的点积；假设二者均为 K维。

通过在训练期间调整自适应缩放因子来降低溢出风险；然而，他们的方法依赖分布假设，无法保证推理期间避免溢出。另一种方法是，Ni 等人 2021年提出训练量化神经网络，使其能够利用基于高成本模运算的循环激活函数抵御溢出。他们还使用正则化惩罚项来控制溢出量。这两种方法都在最外层考虑溢出，因此未能顾及累加中间部分和时可能发生的溢出。此外，在量化感知训练期间对最内层累加过程中的溢出进行建模，现成的深度学习框架并不容易支持，因为这与现有浮点后端上的伪量化并不直接兼容。因此，目前的做法是使用高精度寄存器，或在累加过程中直接对数值进行饱和处理；然而，这种截断仍可能：(1) 引入误差，并在量化神经网络中传播时不断累积；(2) 需要饱和逻辑，而这会破坏结合律，并增加延迟和面积需求（AMD-Xilinx，2023b）。因此，在我们的工作中，我们训练量化神经网络以彻底避免溢出，而不只是减轻溢出对模型准确率的影响。

与我们工作最相近的是de Bruin 等人 2020年提出的方法。该方法提出了一种迭代的逐层优化策略，通过计算开销大的启发式方法选择混合精度位宽，以避免溢出，并假设所有输入数据类型均为有符号位宽。我们提出的方法通过构建基于权重归一化的量化



(a) FINN 框架概览　　(b) MVAU

图 2. 我们改编了（Umuroglu 等人，2017年；Blott 等人，2018年）中的图像，以展示：(a) FINN 框架概览；以及 (b) 矩阵-向量-激活单元（MVAU）的抽象表示。MVAU 是 FINN 编译器用于生成定制流式架构的主要构建模块之一。

公式，在几乎不增加训练开销的同时，兼顾有符号整数和无符号输入数据类型。

与我们的工作相关的是，Wang 等人，2018 年 和 Sakr 等人，2019 年研究了降低浮点累加器精度对加速训练的影响。这些方法不能直接转换为定点算术，而定点算术正是本文的关注重点。

3 背景

我们的工作探讨了在量化感知训练期间使用权重归一化来约束权重，以避免使用低精度累加器时发生溢出。本文将介绍与这一目标相关的背景知识。

3.1 量化感知训练（QAT）

训练期间用于模拟量化的标准算子依赖于将高精度实数映射为低精度量化数的均匀仿射映射，使核心计算能够采用仅整数算术（Jacob 等人，2018年）。量化器（公式 1）和反量化器（公式 2）由零点 z 和缩放因子 s参数化。这里，z 是一个映射到实数零的整数值，因此实数零在量化域中可以被精确表示；s 是一个严格为正的实数标量，对应于量化函数的分辨率。缩放后的值采用半向舍入法舍入到最近的整数，记为 $\lfloor \cdot \rceil$；超出量化域所支持最大值的元素则会被截断：$\text{clip}(x;\ n, p)\ = \min(\max(x;\ n); p)$，其中 n 和 p取决于 x的数据类型。对于位宽为 b的有符号整数，我们假设 $n = -2^{b-1}$ 和 $p = 2^{b-1} - 1$；对于无符号整数，则假设 $n = 0$ 和 $p = 2^b - 1$。

$$\text{quantize}(x; s, z) := \text{clip}(\left\lfloor \frac{x}{s} \right\rceil + z; n, p) \tag{1}$$

$$\text{dequantize}(x; s, z) := s \cdot (x - z) \tag{2}$$

越来越常见的做法是，为学习得到的权重的每个输出通道使用独立的缩放因子，以适应不同的动态范围（Nagel 等人，2019年）。然而，将这一策略扩展到激活值会带来额外开销，因为这需要存储部分和，或引入额外的控制逻辑。因此，标准做法是对激活值使用逐张量缩放因子，而仅对权重使用逐通道缩放因子。通常也会限制权重量化方案，使其满足 $z = 0$，这称为对称量化。消除这些零点可以减少使用仅整数算术执行推理时交叉项带来的计算开销（Jain 等人，2020年）。训练期间，使用直通估计器（STE）（Bengio 等人，2013）使局部梯度能够穿过舍入函数，从而使 $\nabla_x \lfloor x \rceil = 1$ 处处成立，其中 ∇_x表示相对于 x 的局部梯度。

3.2 权重归一化

权重归一化将每个权重向量重参数化$\boldsymbol{w}$ 为参数向量 $\boldsymbol{v}$ 和标量参数g，如式 3所示，其中 $\|\boldsymbol{v}\|_2$ 是 K维向量 $\boldsymbol{v}$ (Salimans &Kingma，2016年）的欧几里得范数。这种简单的重参数化固定了权重向量 $\boldsymbol{w}$ 的欧几里得范数，使得 $\|\boldsymbol{w}\|_2 = g$，从而可以独立学习幅度和方向。

$$\boldsymbol{w} = g \cdot \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_2} \tag{3}$$

与我们的工作相关，先前研究尝试利用权重归一化，在量化感知训练（Cai& Li，2019年）期间对长尾权重分布进行正则化。他们用 ℓ_2-范数替代标准 ℓ_∞-范数，并推导出一种投影算子，将实数映射到量化域。在我们的工作中，我们用 ℓ_2-范数替代 ℓ_1-范数，利用权重归一化参数化方式约束训练期间学习得到的权重，从而在推理期间使用预先设定的累加器位宽。

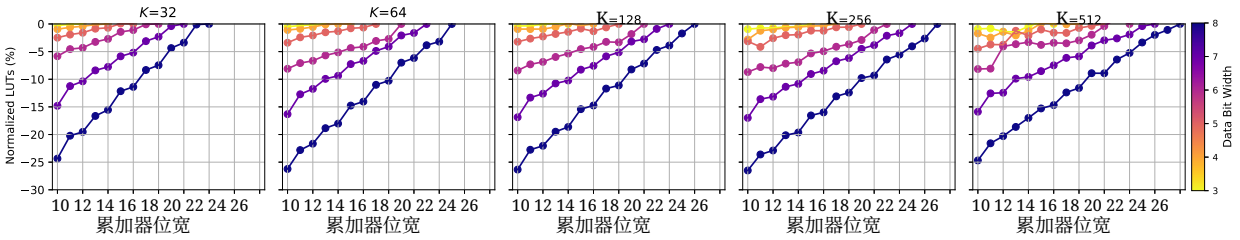


图 3. 随着点积大小（K）以及输入和权重的位宽（分别为 N 和 M）变化，减小累加器的大小也会降低查找表利用率。为简化起见，我们对权重和激活值采用相同的位宽，因此所有数据点均满足 $N = M$，并将二者统称为“数据位宽”。对于给定的点积大小和数据位宽，我们将查找表利用率归一化到根据输入和权重的数据类型确定的累加器位宽最大下界。

4 动机

为阐明我们的研究目标，我们评估了累加器位宽对采用空间数据流架构的定制 FPGA 加速器资源利用率的影响。为此，我们采用 FINN（Umuroglu 等人，2017年；Blott 等人，2018年），这是一个开源框架，旨在为 AMD-Xilinx FPGA 上的 QNN 推理加速生成专用流式架构。

4.1 使用 FINN 生成流式架构

FINN 框架（如图2a 所示）利用空间流式数据流架构，为 AMD-Xilinx FPGA 生成专用量化神经网络加速器，并针对神经网络拓扑和所用数据类型分别进行定制。FINN 的核心是其编译器，它支持灵活的硬件—软件（HW-SW）协同设计，使用户能够控制生成的加速器中每一层的配置。量化神经网络中每一层的权重和激活精度均可单独指定；每一层也会被实例化为专用计算单元（CU），并可通过细粒度并行进行独立优化。

例如，为说明如何将一层实例化为独立的计算单元（CU），我们在图2b 中给出了矩阵-向量激活单元（MVAU）的简化抽象。MVAU 是 FINN 编译器用于线性层和卷积层的主要构建模块之一（Blott 等人，2018年）。每个计算单元都由处理单元（PE）和单指令多数据（SIMD）通道组成：处理单元沿着与数据无关的输出维度并行处理，而单指令多数据通道则沿着与数据相关的输入维度并行处理。同一层内的单指令多数据流和处理单元并发执行（即，空间并行），而神经网络内各层之间采用流水线方式执行（即，时间并行）。神经网络中的所有量化单调激活函数均通过阈值比较实现，将前一层累加得到的高精度结果映射为低精度输出值。编译时，通过数学变换将批归一化、偏置，甚至缩放因子吸收到这一阈值逻辑中（Blott 等人，2018年）。生成的加速器通过 AXI-Stream 协议将输入和输出数据流入、流出芯片，同时使用片上数据流连接各个计算单元，以便在神经网络各层之间传递中间激活值。推理期间，所有网络参数均存储在片上，以避免外部存储器瓶颈。有关 FINN 框架的更多信息，请参阅（Umuroglu 等人，2017年；Blott 等人，2018年；AMD-Xilinx，2023a）。

4.2 累加器对资源利用率的影响

FINN 通常依靠查找表（查找表）以低精度执行乘积累加；在此类场景中，查找表往往是其生成的低精度流式加速器的资源瓶颈。此外，由于激活函数是通过阈值比较实现的，其资源利用率会随着累加器和输出激活值的精度呈指数增长（Blott 等人，2018年）。因此，缩小累加器的规模会直接影响计算与存储需求。

为评估累加器位宽对查找表利用率的影响，我们考虑一个包含一个隐藏层的全连接量化神经网络，该网络由一个有符号整数矩阵参数化。该量化神经网络以一个 K维无符号整数向量作为输入，并输出一个 10 维有符号整数向量。我们使用 FINN 编译器生成一种包含单个 MVAU 的流式架构，目标平台为频率为 100 兆赫的 AMD-Xilinx PYNQ-Z2 开发板。我们报告综合后生成的寄存器传输级设计的资源利用率。为简化分析，我们假设查找表是唯一可用的资源类型，并将 FINN 编译器配置为对计算和存储均使用查找表作为目标资源，以便仅使用一种资源来评估累加器位宽对资源利用率的影响。

如第5节进一步讨论的那样，为避免溢出而采用的最小累加器位宽取决于点积大小（K），以及

输入向量和权重向量 $\boldsymbol{x}$ 和 $\boldsymbol{w}$的位宽，分别记为N 和 M。在图3中，我们展示了在改变 K、 N和 M时，进一步减小累加器位宽如何相应降低资源利用率。对于给定的点积大小和数据位宽，我们以根据输入和权重的数据类型确定的累加器位宽下界中的最大值为基准，对查找表利用率进行归一化。为控制数据流逻辑的资源利用率，我们使用单个处理单元，且不应用循环展开等会增加 SIMD 通道数量的优化。

我们发现，累加器位宽对资源利用率的影响会随数据精度呈指数级增长（即， M 和 N）。缩小累加器的位宽后，当 $N = M = 8$时，我们观察到层的查找表利用率最多可降低 25%；而当 $N = M = 3$时，查找表利用率最多仅降低 1%。这符合预期，因为计算与存储需求会随精度呈指数级增长，因此也有更大的相对节省空间。我们还发现， K会削弱累加器位宽缩减所带来的影响，且这种影响也与数据精度成正比。当 $N = M = 8$ 且 $K = 32$时，累加器每缩减 1 位，查找表平均减少 2.1%；而当$K = 512$时，查找表平均仅减少 1.5%。相反，当$N = M = 3$ 时，无论 K如何，累加器每缩减 1 位，查找表平均仅减少 0.2%。我们推测，这种削弱效应部分源于较大的权重矩阵会带来更高的存储成本。与阈值存储不同，权重的内存需求不会直接受到累加器位宽的影响。因此，累加器位宽缩减带来的相对节省会被权重固定不变的内存需求所稀释。我们将在第7节进一步探讨这一问题，并详细分析为基准模型生成的 FPGA 加速器的资源利用率。

5 累加器位宽界限

图 1展示了量化神经网络推理中累加过程的简化抽象。随着激活值在各层间传播，卷积或矩阵乘法等运算产生的中间部分和会先累加到寄存器中，再重新量化并传递至下一层。为避免数值溢出，存储这些累加值的寄存器不仅需要足够宽，以容纳点积结果，还要能容纳所有中间部分和。

考虑输入数据的点积 $\boldsymbol{x}$ 与学习得到的权重$\boldsymbol{w}$，它们都是 K维整数向量。令 y表示根据式 4计算得到的点积标量结果，其中x_i 和 w_i 分别表示向量 i 和 $\boldsymbol{x}$ 中的元素 $\boldsymbol{w}$。由于 y 的表示范围受 $\boldsymbol{x}$ 和 $\boldsymbol{w}$ 的表示范围限制，我们利用它们的范围推导累加寄存器（或累加器）的位宽 P 下界。

$$y = \sum_{i=1}^{K} x_i w_i \tag{4}$$

当输入数据经过动态范围非负的激活函数（例如修正线性单元或 ReLU）后，或者采用合适的零点（即非对称量化）时，输入数据通常表示为无符号整数。否则则使用有符号整数。由于权重通常表示为有符号整数，我们在本文中假设累加器始终为有符号整数。因此，鉴于 $\boldsymbol{x}$ 与 $\boldsymbol{w}$ 之间的点积标量结果是一个由式 P定义的 4 位整数，可得 $\sum_{i=1}^{K} x_i w_i$ 满足以下界限：

$$-2^{P-1} \leq \sum_{i=1}^{K} x_i w_i \leq 2^{P-1} - 1 \tag{5}$$

为了满足这个双重不等式的右侧，可得 $|\sum_{i=1}^{K} x_i w_i| \leq 2^{P-1} - 1$。然而，累加器需要有足够的位宽，不仅要存储点积的结果，还要存储所有中间部分和。

由于输入数据事先未知，我们的界限必须考虑每个乘积累加的最坏情况值。因此，由于乘积和的幅度不超过各乘积幅度之和，如果 $\sum_{i=1}^{K} |x_i||w_i| \leq 2^{P-1} - 1$，那么 $\boldsymbol{x}$ 与 $\boldsymbol{w}$ 之间的点积可以存入 P 位累加器，且不会发生数值溢出，如下所示。先验

$$|\sum_i x_i w_i| \leq \sum_i |x_i w_i| \leq \sum_i |x_i||w_i| \leq 2^{P-1} - 1 \tag{6}$$

5.1 使用数据类型推导下界

每个乘积累加的最坏情况值可以天真地根据所用数据类型的表示范围推断得出。当 x_i 和 w_i 为有符号整数时，它们的幅度分别受限于 $|x_i| \leq 2^{N-1}$ 和 $|w_i| \leq 2^{M-1}$。在 x_i 为无符号整数的情况下，每个输入值的幅度上界为 $|x_i| \leq 2^N - 1$；不过，为方便表示，我们将此上界简化为 $|x_i| \leq 2^N$[1]。结合有符号和无符号上界，可得 $|x_i| \leq 2^{N - \mathbb{1}_{\text{signed}}(\boldsymbol{x})}$，其中 $\mathbb{1}_{\text{有符号整数}}(\boldsymbol{x})$ 是一个指示函数，当且仅当 $\boldsymbol{x}$ 是有符号整数向量时返回 1。

根据公式 6可知，幅度乘积之和的上界为：

$$\sum_{i=1}^{K} |x_i||w_i| \leq K \cdot 2^{N+M-1-\mathbb{1}_{\text{signed}}(\boldsymbol{x})} \leq 2^{P-1} - 1 \tag{7}$$

对等式两边取对数 7，可推导出累加器位宽的下界 P：

$$\log_2 \left(2^{\log_2(K)+N+M-1-\mathbb{1}_{\text{signed}}(\boldsymbol{x})} + 1\right) + 1 \leq P \tag{8}$$

[1]请注意，我们对无符号输入数据上界的简化并不会影响溢出避免。

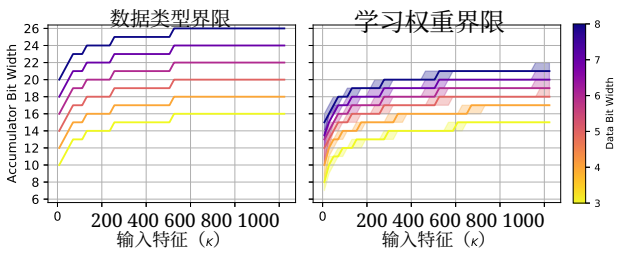


图*4*。我们通过改变点积的大小（K）以及权重（M）和输入激活值（N）的位宽，直观展示累加器位宽界限之间的差异；我们将二者统称为"数据位宽"，因此 $M = N$。

这可简化为 P 的以下下界：

$$P \geq \alpha + \phi(\alpha) + 1 \tag{9}$$

$$\alpha = \log_2(K) + N + M - 1 - \mathbb{1}_{\text{signed}}(\boldsymbol{x}) \tag{10}$$

$$\phi(\alpha) = \log_2(1 + 2^{-\alpha}) \tag{11}$$

在图4中，我们假设 $\boldsymbol{x}$ 是一个无符号整数向量，并将此界限可视化，使得$\mathbb{1}_{\text{有符号整数}}(\boldsymbol{x}) = 0$。其中，我们展示了在同时改变点积的大小（$K$）以及权重和激活值的位宽时，累加器位宽下界如何增加。

5.2 使用学习得到的权重推导下界

由于学习得到的权重在推理期间是固定的，我们可以利用其幅度信息，推导出更紧的累加器位宽下界。

再次根据公式6，幅度乘积之和受公式12界定，其中 $\|\boldsymbol{w}\|_1$ 表示向量上的标准 ℓ_1-范数 $\boldsymbol{w}$。

$$\sum_{i=1}^{K} |x_i||w_i| \leq 2^{N - \mathbb{1}_{\text{signed}}(\boldsymbol{x})} \cdot \|\boldsymbol{w}\|_1 \leq 2^{P-1} - 1 \tag{12}$$

这里，我们定义 P 的一个更紧的下界：

$$P \geq \beta + \phi(\beta) + 1 \tag{13}$$

$$\beta = \log_2(\|\boldsymbol{w}\|_1) + N - \mathbb{1}_{\text{signed}}(\boldsymbol{x}) \tag{14}$$

$$\phi(\beta) = \log_2(1 + 2^{-\beta}) \tag{15}$$

在图 4中，我们再次对该下界进行可视化，并假设 $\boldsymbol{x}$ 是一个无符号整数向量。由于公式 14 取决于学习得到的权重值，我们从离散高斯分布中随机采样每个K维向量，并展示累加器位宽的中位数，以及在300次随机采样中观测到的最小值和最大值。我们展示了，在同时改变点积的大小（K）以及权重和输入激活值的位宽时，使用学习得到的权重（右图）比使用数据类型（左图）能够为累加器位宽提供更紧的下界。

6 训练量化神经网络以避免溢出

为了训练量化神经网络在不发生溢出的情况下使用低精度累加器，我们采用权重归一化来约束学习得到的权重 $\boldsymbol{w}$，使其满足第 5.2节中推导出的下界。基于公式12，我们将累加器位宽的下界 P 转换为 ℓ_1范数的上界，$\boldsymbol{w}$ 如公式16所示。请注意，由于每个输出神经元都需要自己的累加器，因此该上界需要逐通道执行。

$$\|\boldsymbol{w}\|_1 \leq \left(2^{P-1} - 1\right) \cdot 2^{\mathbb{1}_{\text{signed}}(\boldsymbol{x}) - N} \tag{16}$$

6.1 构造我们的量化算子

为在量化感知训练期间施加此约束，我们对量化器进行重参数化，使每个权重向量 $\boldsymbol{w}$ 都表示为参数向量 $\boldsymbol{g}$ 和 $\boldsymbol{v}$的形式。与第3.2节讨论的标准权重归一化公式类似，这种重参数化将范数与权重向量解耦；不过，与标准公式不同的是，每个输出通道都单独学习其范数。对于具有 C 个输出通道的给定层，我们将标准公式中的逐张量 ℓ_2-范数（式 3）替换为逐通道 ℓ_1-范数。式 17给出的这种重参数化，使得权重向量 $\boldsymbol{w}$ 的 ℓ_1-范数能够按通道独立学习，从而满足 $g_i = \|\boldsymbol{w}_i\|_1$对所有 $i \in \{1, \cdots, C\}$，其中 $\boldsymbol{w}_i$ 表示通道 i 的权重，g_i 表示参数向量 $\boldsymbol{g}$中的元素 i。

$$\boldsymbol{w}_i = g_i \cdot \frac{\boldsymbol{v_i}}{\|\boldsymbol{v_i}\|_1} \quad \forall\, i \in \{1, \cdots, C\} \tag{17}$$

与标准量化算子类似，我们基于权重归一化的量化依赖于从高精度实数域到低精度量化域的均匀仿射映射，并使用学习得到的逐通道缩放因子 $\boldsymbol{s} = \{s_i\}_{i=1}^{C}$。因此，通过约束 g_i 满足式18，我们可以学习满足累加器位宽上限且避免溢出的量化权重。

$$g_i \leq s_i \cdot \left(2^{P-1} - 1\right) \cdot 2^{\mathbb{1}_{\text{signed}}(\boldsymbol{x}) - N} \tag{18}$$

下面，我们阐述基于权重归一化的量化算子。为使符号表示清晰、简洁，我们考虑一个只有一个输出通道的层（即，$C = 1$），因此参数向量 $\boldsymbol{g} = \{g_i\}_{i=1}^{C}$ 和 $\boldsymbol{s} = \{s_i\}_{i=1}^{C}$可分别表示为标量 g 和 s。

$$\text{quantize}(\boldsymbol{w}; s, z) := \text{clip}\left(\left\lfloor \frac{g}{s} \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_1} \right\rfloor + z; n, p\right) \tag{19}$$

训练期间，我们的权重量化算子依次执行四种逐元素操作：缩放、舍入、截断，然后反量化。与标准算子一样，我们在映射中消除零点，使得 $z = 0$。我们对缩放

因子 $s = 2^d$ 和范数参数 $g = 2^t$采用 d和 t 的指数参数化形式，这两个对数尺度参数均通过随机梯度下降进行学习。这与（Jain 等人，2020年）的工作类似，但我们移除了整数 2 的幂约束；这些约束对 FINN 生成的流式架构没有额外益处，因为浮点缩放因子可以通过数学变换并入阈值逻辑（Blott 等人，2018年）。随后，将缩放张量舍入至零，我们将其记为$\lfloor \cdot \rfloor$。这可以避免向上舍入导致范数超过我们的约束。请注意，这也是它与传统量化算子的另一处区别，后者采用中点舍入。最后，张量中的元素经缩放和舍入后，再根据公式 2进行截断和反量化。训练期间使用的最终量化算子由公式20给出，其中 n 和 p取决于权重位宽 M的表示范围。

$$q(\boldsymbol{w}; s) := \text{clip}\left(\left\lfloor \frac{g}{s}\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_1} \right\rfloor; n, p\right) \cdot s \tag{20}$$

$$\text{where } s = 2^d \tag{21}$$

$$\text{and } g = 2^{\min(T,t)} \tag{22}$$

$$\text{and } T = \mathbb{1}_{\text{signed}}(\boldsymbol{x}) + \log_2(2^{P-1} - 1) + d - N \tag{23}$$

量化激活值时，我们使用第 3.1 节中讨论的标准量化算子。所有经过非负函数（即 ReLU）处理的激活值都使用无符号整数表示，否则则使用有符号整数表示。

为了在训练过程中更新可学习参数，我们使用直通估计器（STE）（Bengio 等人，2013），使局部梯度能够通过舍入函数传播，从而使 $\nabla_x \lfloor x \rfloor = 1$ 在任何位置都成立，这是常见的做法。

6.2 拉格朗日惩罚项正则化

为避免 t 在 $t > T$处陷入停滞，我们引入拉格朗日惩罚项 $\mathcal{L}_{\text{penalty}}$，其定义见式 24。对于一个包含 L 层的神经网络，每层有 C_l 个输出通道，$t_{i,\ l}$表示第i 层 l 中通道的范数对数尺度参数，而 $T_{i,\ l}$ 表示其上界，定义见式 23。请注意，即使在 $\mathcal{L}_{\text{penalty}} > 0$的情况下，我们仍可通过截断式22中的范数来满足累加器约束。不过，我们的拉格朗日惩罚项会促使每层中各通道的范数 g_i 和缩放 s_i i 共同满足上界而不进行截断，从而使其对数尺度参数的更新仅依据任务误差。

$$\mathcal{L}_{\text{penalty}} = \sum_{l=1}^{L} \sum_{i=1}^{C_l} (t_{i,l} - T_{i,l})_+ \tag{24}$$

需要注意的是，我们的公式具有任务无关性。假设基础任务损失为 $\mathcal{L}_{\text{task}}$，则总损失 $\mathcal{L}_{\text{total}}$ 定义如下

根据式 25，其中 λ 是一个拉格朗日乘子。在我们的实验中，我们将 λ 固定为一个常数标量，不过也可以探索（Roy 等人，2021年）等自适应方法。

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{penalty}} \tag{25}$$

7 实验

我们使用图像分类和单图像超分辨率基准测试来评估我们的算法。由于我们的算法是首创，我们将其与第 3.1 节讨论的标准量化感知训练（QAT）方案进行比较。我们使用 PyTorch 和 Brevitas v0.7.2 实现所有算法（Pappalardo，2021年），在 AMD MI100 加速器上，单 GPU 量化感知训练耗时从 5 分钟（例如，ESPCN）到 2 小时（例如，MobileNetV1）不等。为了给生成的量化神经网络设计定制 FPGA 架构，我们以 FINN v0.8.1 （ AMD-Xilinx，2023a ）为基础，并添加扩展以支持我们的工作。

7.1 图像分类基准测试

我们使用 CIFAR10 数据集（Howard 等，2017年）训练 MobileNetV1 和 ResNet18（He 等，2016a年），以进行图像分类（Krizhevsky 等，2009年）。我们严格遵循相关作者最初提出的网络架构，但做了一些细微调整，使中间表示更适合图像尺寸，具体将在下文讨论。按照惯例，对于所有配置，我们将输入层和输出层的权重和激活值固定为 8 位，并从在 CIFAR10 上预训练至收敛的对应浮点模型初始化所有模型。我们根据观测到的测试集 Top-1 准确率评估所有模型。

对于 MobileNetV1，我们在第一个卷积层和最后一个平均池化层中都使用步幅 2。这样可以降低下采样程度，使模型更适合在较小图像上进行训练。其余所有层的配置均与文献（Howard 等人，2017年）提出的方法相同。我们使用随机梯度下降（SGD）优化器，以批量大小为 64 张图像对所有模型进行微调，共训练 100 轮，权重衰减设为 1e-5。初始学习率设为 1e-3，每个训练轮次后乘以 0.9。

对于 ResNet18，我们将第一个卷积层改为步幅和填充均为 1、卷积核大小为 3。与 MobileNetV1 类似，我们移除前置的最大池化层，以减少整个神经网络中的下采样幅度。我们还使用卷积捷径（He 等人，2016b年），而非标准的恒等映射，因为实验结果证明，前者表现更优。其余层配置均与（He 等人，2016a年）提出的方案相同。我们使用 SGD 优化器，以批量大小 256 对所有模型进行微调，共训练 100 轮，权重衰减设为 1e-5。初始学习率设为 1e-3，并且每 30 轮将其缩小为原来的 0.1 倍。

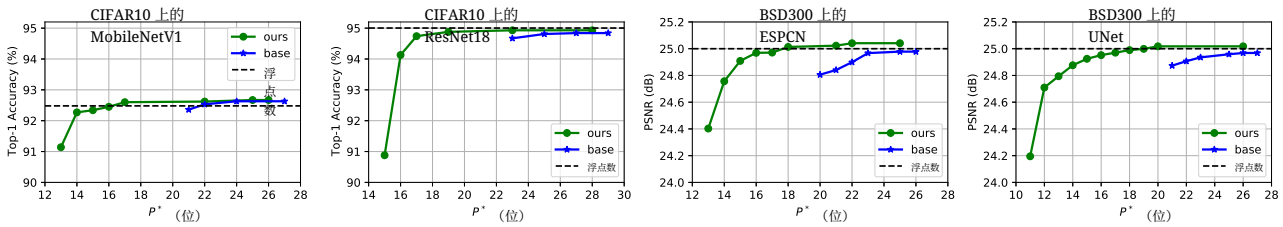


图 *5*。通过图像分类和单图像超分辨率基准测试，我们表明，即使大幅降低累加器位宽，仍能保持模型性能，使其接近浮点基线。我们通过帕累托前沿展示这种权衡，这些前沿是通过对不同的权重、激活值和累加器位宽进行网格搜索估算得出的。这里，我们使用 P^* 表示神经网络所有层允许的最大累加器位宽。我们将自己的算法（**绿色圆点**）与标准量化基线算法（**蓝色星形**）进行比较，并将每项实验重复 3 次；每个模型总计运行 500 多次，以构建每组帕累托前沿。我们观察到，在所有基准测试中，我们的算法都优于基线，表明即使以浮点基线为参照，也能在不显著牺牲模型性能的情况下减小累加器位宽。

7.2 单图像超分辨率基准测试

我们使用 BSD300 数据集（Shi 等人，2016年）训练 ESPCN，并使用 BSD300 数据集（Ronneberger 等人，2015年）训练 UNet，将单幅图像放大 3 倍。我们严格遵循各自作者最初提出的神经网络架构，但做了少量调整，使其更适合硬件实现。按照惯例，对于所有配置，我们将输入层和输出层的权重及激活值固定为 8 位；不过，所有超分辨率模型均从头开始训练。我们通过测量测试数据集上的峰值信噪比（PSNR），对所有模型进行实证评估。

对于 ESPCN，我们用最近邻调整大小卷积（NNRC）替换亚像素卷积。研究表明，这种方法能够减少训练过程中的棋盘格伪影（Odena 等人，2016年），并且可在推理阶段高效执行（Colbert 等人，2021年）。其他所有层配置均与（Shi 等人，2016年）提出的配置相同。我们使用 Adam 优化器（Kingma 和 Ba，2014年）对所有模型进行微调，训练 100 轮次，每批处理 16 张图像，权重衰减设为 1e-4。初始学习率设为 1e-4，并在每个训练轮次后乘以 0.98。

对于 UNet，我们仅使用 3 个编码器和解码器，以构建比最初提出的架构更小的网络（Ron- neberger 等人，2015年）。我们用最近邻调整大小卷积替代转置卷积；研究表明，在推理阶段两者功能等效（Colbert 等人，2021年），但前者在训练阶段表现更佳（Odena 等人，2016年）。我们将所有拼接替换为相加，并相应减少输入通道。我们使用 Adam 优化器，以批量大小为 16 张图像、权重衰减为 1e-4，对所有模型进行微调，共训练 200 轮。初始学习率设为 1e-3，每经过 50 轮便乘以 0.3。

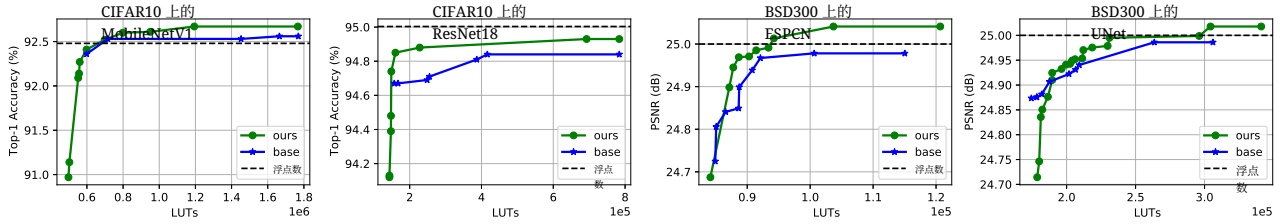
7.3 实验设置与研究问题

我们围绕以下问题设计实验：

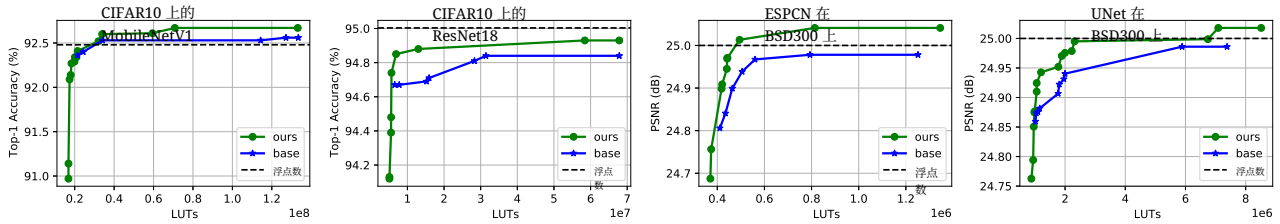
- 降低累加器位宽会如何影响模型性能（第7.4节）？
- 资源利用率与模型性能之间存在怎样的权衡（第7.5节）？
- 随着我们降低累加器位宽，资源节省来自哪里（第7.6节）？

与第4节所述实验类似，我们简化分析，并假设查找表是唯一可用的资源类型。我们将 FINN 编译器配置为尽可能针对计算与存储使用查找表，以便仅使用一种资源类型来评估累加器位宽对资源利用率的影响。

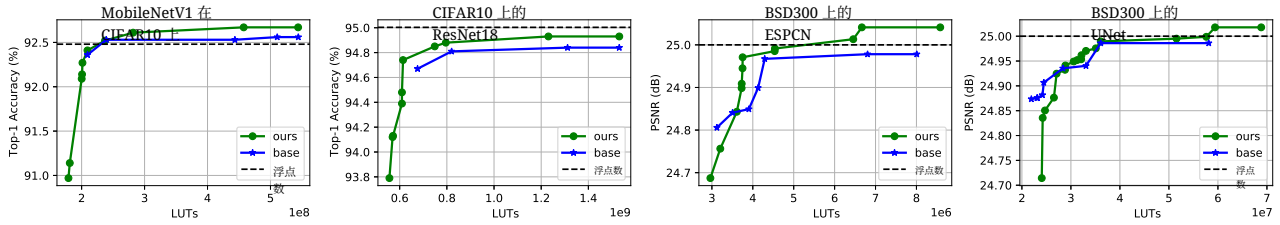
在所有实验中，我们将关注点放在数据位宽为 5 至 8 位的范围内，原因有二：(1) 将精度降至 5 位以下通常需要专门定制的超参数，这会导致不同位宽之间的比较不够公平；(2) 如第 4 节所示，在较低数据位宽下，减小累加器的尺寸对查找表利用率影响可以忽略不计。尽管如此，即使缩小了可选数据位宽的范围，要测试我们用作基准测试的量化神经网络所提供的设计空间中权重、激活值和累加器位宽的每种组合，在计算上仍然难以实现。因此，对于权重和激活值位宽，我们统一约束神经网络中每个隐藏层的精度，使 M 和 N 为常数标量；首层和末层则保持为 8 位，使得 $M = N = 8$。然而，累加器位宽不仅取决于权重和激活值位宽，还取决于点积的大小（K），如第5节所述。为简化设计空间，我们约束给定神经网络中的所有层使用相同的累加器位宽，并将其记为 P^*，使得任意层的最大累加器位宽为 P^* 位。请注意，P^* 的值用于公式16 来



(a) 无并行



(b) 最大处理单元数



(c) 最大处理单元数和单指令多数据流数

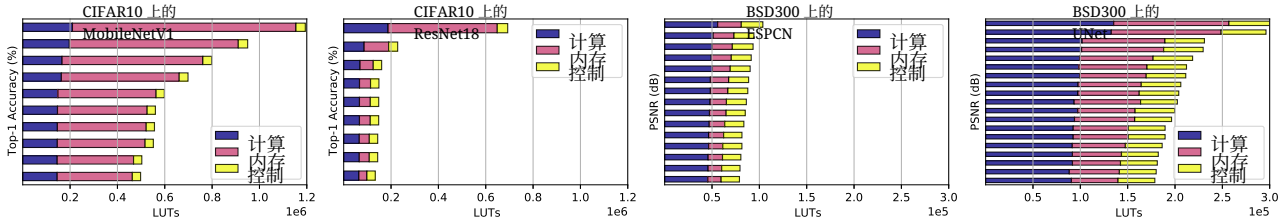
图 *6*。 为了评估资源利用率与准确率之间的权衡，我们绘制了在图像分类和单图像超分辨率基准测试中观察到的帕累托前沿。我们在以下场景中评估这些帕累托前沿：(a) 不进行空间并行；(b) 为每一层最大化处理单元数；(c) 为每一层最大化处理单元数和单指令多数据流数。在每种场景下，与标准量化算法（**绿色圆点**）相比，我们的算法（**蓝色星号**）在每个模型上都呈现出占优的帕累托前沿，表明我们的算法能够降低查找表利用率，同时不显著牺牲模型性能。

为每个权重逐通道的 ℓ_1范数设定上界，并通过公式23 在训练过程中施加这一约束。

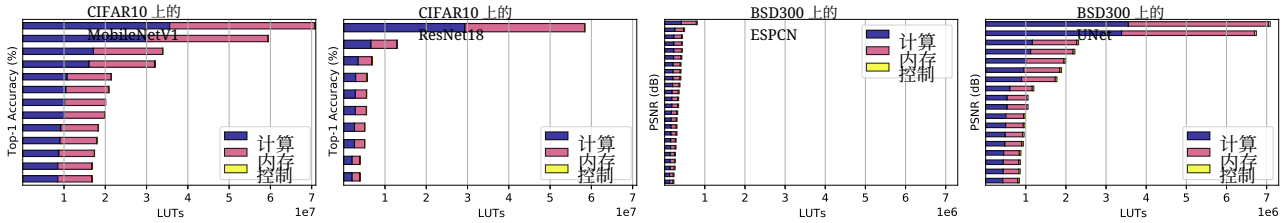
为了收集足够的数据来探究研究问题，我们对 5 至 8 位的权重位宽和激活位宽进行网格搜索。对于这 16 种组合，我们根据神经网络中最大层的数据类型界限（公式 9）计算累加器位宽的最大下界。以此初始化 P^*后，我们评估将累加器位宽最多降低 10 位的情况，从而生成总计 160 种配置。最后，我们将结果与第 3.1 节讨论的标准量化算法进行基准比较；然而，由于该算法不支持对累加器位宽进行控制，因此此次网格搜索仅限于权重位宽和激活位宽的 16 种组合。每种配置运行 3 次，每个模型总计运行 528 次。在接下来的章节中，我们将总结研究结果。

7.4 累加器对模型性能的影响

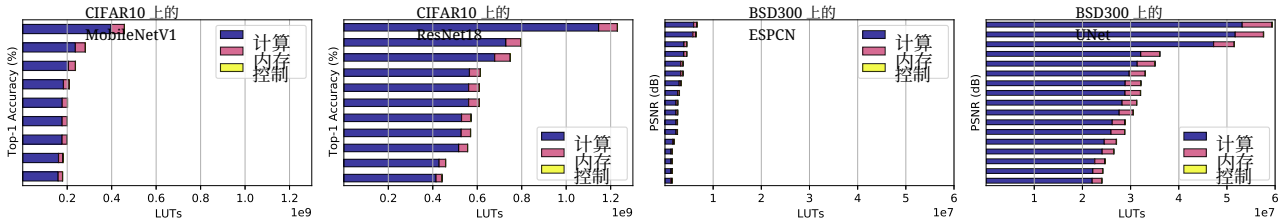
我们的算法提出了一种新颖的方法，可约束量化神经网络（QNN）的权重，使其在使用预定义的累加器位宽时不会发生溢出。作为我们算法的替代方案，设计人员可以根据式9给出的数据类型界限，通过启发式方法调整数据位宽。采用这种方法，在使用预定义的累加器位宽时，仍可保证避免溢出 P，但这只是间接施加此类约束的方法。给定预定义的累加器位宽上界 P^*，我们比较了使用我们的算法训练的模型与使用该启发式方法训练的模型的性能，并在图5中以帕累托前沿的形式展示了这一比较。需要注意的是，尽管这并非与（de Bruin 等人，2020年）提出的算法进行直接比较，但这一实验在原理上相似。与（de Bruin 等人，2020年）不同，我们采用了第3.1 节详述的更先进的量化技术，并用计算成本更高的



(a) 无并行



(b) 处理单元数量最多



(c) 处理单元和单指令多数据流数量最多

图 7。我们将 图中各个帕累托最优模型的查找表利用率分解为计算、内存和控制流。 6.

损失引导搜索技术，替换为成本更高但更全面的网格搜索。

帕累托前沿展示了给定 P^* 下观测到的最大模型性能。我们观察到，与当前方法相比，我们的算法能够进一步降低累加器位宽，同时保持模型性能。此外，大多数模型即使相较于浮点基线，在使用 16 位累加器时的性能下降也低于 1%；对于通用处理器中的低精度累加而言，16 位通常是目标位宽（de Bruin 等人，2020年；Xie 等人，2021年）。

7.5 资源与准确率之间的权衡

为了解累加器位宽对 FINN 生成的加速器设计空间的影响，我们评估了资源利用率与模型性能之间的权衡。对于第 7.3 节网格搜索中训练的每个模型，我们使用 FINN 编译器生成资源利用率估算值，并使用帕累托前沿来可视化数据。在图 6 中，我们给出了总量对应的观测到的最高模型性能

加速器所用的查找表。我们使用三种优化配置评估这些帕累托前沿。首先，将每个模型中的每一层实例化为一个计算单元，不进行任何空间并行优化，并在图 6a 中展示帕累托前沿。其次，将每个模型中每一层使用的处理单元数量最大化，并在图 6b 中展示帕累托前沿。最后，同时将处理单元数量和所用的 SIMD 通道数量最大化，并在图 6c 中展示帕累托前沿。

为确保我们从这些估算值中分析出的趋势具有意义，我们回到第 4 节开展的实验。我们将综合后寄存器传输级报告的绝对查找表利用率与相应的 FINN 估算值进行比较，观察到二者的相关性为 94%。

降低权重和激活值的精度可以节省资源利用率，但会牺牲模型性能；然而，我们发现，将累加器位宽纳入设计空间能够实现更好的整体权衡。结果表明，对于给定的目标精度或资源预算，我们的算法在查找表利用率与模型性能之间的权衡方面优于现有的各种优化策略基线，

这证实了将累加器位宽纳入整体软硬件协同设计空间的益处。

7.6 评估资源节省

由于我们强制 FINN 编译器尽可能使用查找表来实现计算和存储资源，因此我们会评估资源节省来自何处。为此，我们将查找表利用率分为计算、存储和控制流三类。对于计算，我们汇总加法树、乘加运算和比较逻辑所用的查找表；对于存储，我们汇总用于存储权重、阈值和中间表示的查找表；对于控制流，我们汇总用于片上互连和 AXI-Stream 协议的查找表。在图7中，我们直观展示了图6中帕累托前沿对应的每个帕累托最优模型的这一细分情况。

我们发现，在生成不采用空间并行的加速器时，查找表节省主要来自存储资源的减少；而随着并行度提高，节省则主要来自计算资源。在没有并行的情况下，查找表利用率的降低主要源于阈值和中间激活值存储成本的下降，而这直接受到累加器和输出激活值精度的影响。随着并行度提高，计算用查找表的减少主要来自乘加运算成本的降低，而这直接受到权重、输入和累加器精度的影响。

最后，我们观察到，每个神经网络的控制流查找表基本保持不变；这是意料之中的，因为数据类型的变化不会影响神经网络架构。值得注意的是，与没有跳跃连接的神经网络（例如，ResNet18 和 UNet）相比，具有跳跃连接的神经网络（例如，MobileNetV1 和 ESPCN）中控制流逻辑所占的查找表比例更高；随着并行度提高，其影响相对减小。

7.7 深入探究我们的约束条件的影响

作为权重归一化方法的副产物，我们的量化算法不仅能将权重限制在适合用户指定位宽的累加器中，还能提高所得权重的稀疏性和可压缩性，如图所示。8。

稀疏性是张量中取值为零的元素所占的比例。在机器学习中，稀疏性最常见的用途
工作负载旨在通过降低计算与内存需求来加速推理（Gale 等，2020年）。对于感兴趣的读者，我们推荐参阅（Hoefler 等，2021年）以了解近期关于深度学习中稀疏性相关既有工作的综述。其中包括多项研究，探讨了使用 ℓ_1-范数权重正则化来引入稀疏性（Yang 等，2019年；Chao 等，2020年）。我们的量化算法

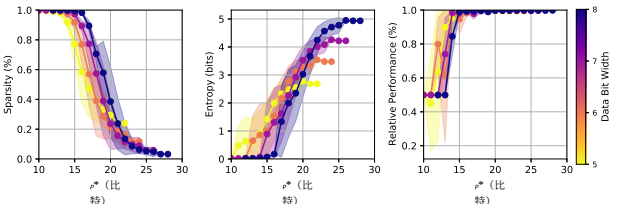


图*8*。由于我们施加了 ℓ_1-范数约束，降低累加器位宽后，便可在不牺牲相对于浮点基线的模型性能的前提下，利用非结构化稀疏性（左）和权重压缩（中）。

具有类似的效果。通过将标准权重归一化公式中的 ℓ_2-范数替换为我们的对数缩放ℓ_1-范数参数，我们提出了一种鼓励非结构化权重稀疏性的新方法。回想一下，P^* 的值用于通过公式 16 对 ℓ_1-范数权重设置上界。因此，进一步减小 P^* 会约束该上界，从而鼓励以 ℓ_1-范数权重正则化的形式实现权重稀疏性。在图8 左侧中，随着我们降低累加器位宽 P^*，我们展示了所有基准模型的平均稀疏性。

可压缩性通常通过熵（即Shannon，1948年）所衡量的每元素比特数理论下界来估算。降低熵会减少无损压缩所需的信息量，从而提高压缩率。以往研究探讨了将熵正则化作为改善权重压缩的一种方法（Agustsson等人，2017年；Aytekin等人，2019年）。我们观察到，我们的 ℓ_1-范数约束与这些技术具有类似的效果。如图8 中间部分所示，随着累加器位宽降低，熵也随之下降 P^*。

在图 8中，我们通过将 P^* 降低，展示了稀疏性、可压缩性和相对模型性能所受到的影响；所使用的模型来自第 7.3 节所述的网格搜索。为简化分析，我们重点研究权重位宽与输入激活位宽相同的配置（例如，$M = N$），并绘制了在全部 4 个基准模型上观测到的平均值。对于权重和激活值均为 8 位的模型，我们观察到，将 P^* 降至 16 位后，平均稀疏性达到 98.2 %，预计压缩率为 46.5 倍，同时保持浮点性能的 99.2%。

8 结论与未来工作

我们提出一种新颖的量化算法，用于训练适配低精度累加的量化神经网络。该算法利用权重归一化，将学习到的参数约束在预定义位宽的累加器所能表示的范围内。不同于以往仅试图

降低溢出风险或减轻其对模型精度的影响，我们的方法能够保证避免溢出。

据我们所知，本研究首次探索了将低精度累加器用于提升可编程硬件的设计效率，并将其应用于量化神经网络推理加速器。因此，我们从理论上评估了溢出，并推导出比现有文献粒度更细的累加器位宽全面界限。实验表明，与标准基线相比，使用我们的算法为 AMD-Xilinx FPGA 训练量化神经网络，可以改善资源利用率与模型精度之间的权衡。

我们的结果带来以下几点启示：

- 虽然减小累加器的尺寸必然会降低模型精度，但我们的算法显著缓解了这种权衡。
- 与基线相比，使用我们的算法为采用更低精度累加器的量化神经网络进行训练，在相同资源预算下能够得到性能更高的模型。
- 在没有空间并行的情况下，我们节省的大部分资源来自内存需求的降低，因为减小累加器位宽也会降低存储阈值和中间激活值的成本。
- 随着空间并行度的提高，计算成本的降低成为资源节省的主要因素，因为减小累加器位宽可以降低创建更多乘加运算的成本。
- 我们的算法天然会促成极端的非结构化稀疏性，并提高量化神经网络所得权重的可压缩性，同时保持相对于浮点基线的性能。

FPGA 的灵活性是一把双刃剑。位级控制允许针对量化神经网络中的每一层，分别调整权重、激活值以及现在的累加器的精度；然而，如此多的自由度所带来的设计空间也引入了复杂的优化问题。我们的算法将累加器位宽作为另一个可调参数，从而提高了软硬件协同设计的灵活性；在同时优化量化神经网络及其相应的推理加速器时，可以对该参数进行调节。未来，我们希望探索利用最先进的神经架构搜索算法，更高效地遍历这一庞大的设计空间。

致谢

我们谨向 Gabor Sines、Michaela Blott、Nicholas Fraser、Yaman Umuroglu、Thomas Preusser、Mehdi Saeedi、Alex Cann，以及 AMD 软件技术、架构和 AECG 研究团队的其他成员致谢，感谢他们提供富有启发性的讨论和基础设施支持。

© 2023年 超威半导体公司。保留所有权利。AMD、AMD 箭头标志、Radeon 及其组合均为超威半导体公司的商标。本出版物中使用的其他产品名称仅用于识别目的，可能是其各自公司的商标。

参考文献

Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L. 和 Gool, L. V. 用于端到端学习可压缩表示的软到硬向量量化。神经信息处理系统进展，30，2017年。

AMD-Xilinx。Finn：用于 FPGA 上量化神经网络推理的数据流编译器。https://github.com/Xilinx/finn，2023a。访问日期：2023-01-19。

AMD-Xilinx。Vivado 设计套件用户指南：高性能 FPGA 设计说明。https://docs.xilinx.com/r/en-US/ug897-vivado-sysgen-user/Use-Saturation-Arithmetic-and-Rounding-Only-When-Necessary，2023b。访问日期：2023-01-19。

Aytekin, C., Cricri, F. 和 Aksu, E. 神经网络权重的可压缩性损失。*arXiv* 预印本 *arXiv:1905.01044*，2019年。

Bengio, Y., L´eonard, N. 和 Courville, A. 针对条件计算，通过随机神经元估计或传播梯度。*arXiv* 预印本 *arXiv:1308.3432*，2013年。

Blott, M., Preusser, T. B., Fraser, N. J., Gambardella, G., O'Brien, K., Umuroglu, Y., Leeser, M. 和 Vissers, K. Finn-r：用于快速探索量化神经网络的端到端深度学习框架。*ACM* 可重构技术与系统汇刊 *(TRETS)*，11(3):1–23，2018年。

Cai, W.-P. 和 Li, W.-J. 基于权重归一化的量化方法，用于深度神经网络压缩。*arXiv* 预印本 *arXiv:1907.00593*，2019年。

Chao, S.-K., Wang, Z., Xing, Y. 和 Cheng, G. 深度神经网络的方向性剪枝。神经信息处理系统进展，33:13986–13998，2020年。

Colbert, I., Kreutz-Delgado, K. 和 Das, S. 一种用于基于卷积的图像上采样的节能边缘计算范式。*IEEEAccess*，9:147967–147984，2021年。

de Bruin, B., Zivkovic, Z., 和 Corporaal, H. 面向累加器受限处理器的深度神经网络量化。微处理器与微系统，72:102872，2020年。

Gale, T., Zaharia, M., Young, C., 和 Elsen, E. 面向深度学习的稀疏 GPU 内核。发表于*SC20*：高性能计算、网络、存储与分析国际会议，第 1–14 页。IEEE，2020年。

Gholami, A.、Kim, S.、Dong, Z.、Yao, Z.、Mahoney, M.W. 和 Keutzer, K. 高效神经网络推理的量化方法综述。*arXiv*预印本 *arXiv:2103.13630*，2021年。

He, K.、Zhang, X.、Ren, S. 和 Sun, J. 用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议论文集，第770–778页，2016a年。

He, K., Zhang, X., Ren, S. 和 Sun, J. 深度残差网络中的恒等映射。载于欧洲计算机视觉会议，第630–645页。施普林格，2016b年。Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N. 和 Peste, A. 深度学习中的稀疏性：通过剪枝与增长实现神经网络的高效推理和训练。《机器学习研究期刊》，22(1):10882–11005，2021年。Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. 和 Adam, H. MobileNets：面向移动视觉应用的高效卷积神经网络。*arXiv*预印本 *arXiv:1704.04861*，2017年。Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R. 和 Bengio, Y. 量化神经网络：使用低精度权重和激活值训练神经网络。《机器学习研究期刊》，18(1):6869–6898，2017年。Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. 和 Kalenichenko, D. 面向高效的纯整数推理的神经网络量化与训练。载于*IEEE*计算机视觉与模式识别会议论文集，第2704–2713页，2018年。Jain, S., Gural, A., Wu, M. 和 Dick, C. 用于深度神经网络精确、高效定点推理的训练式量化阈值。机器学习与系统会议论文集，2:112–128，2020年。Kingma, D. P. 和 Ba, J. Adam：一种随机优化方法。*arXiv*预印本 *arXiv:1412.6980*，2014年。Krizhevsky, A., Hinton, G. 等。通过微型图像学习多层特征。2009年。Lou, Q. 和 Jiang, L. She：一种用于加密数据的快速、精确深度神经网络。神经信息处理系统进展，32，2019年。

Martin, D., Fowlkes, C., Tal, D. 和 Malik, J. 人工分割自然图像数据库及其在评估分割算法和测量生态统计中的应用。载于第八届国际计算机视觉会议论文集，第2卷，第416–423页，2001年7月。

Nagel, M.、Baalen, M. v.、Blankevoort, T. 和 Welling, M. 通过权重均衡和偏置校正实现无数据量化。载于*IEEE/CVF*国际计算机视觉会议论文集，第1325–1334页，2019年。

Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., van Baalen, M. 和 Blankevoort, T. 关于神经网络量化的白皮书。*arXiv* 预印本 *arXiv:2106.08295*，2021年。Ni, R., Chu, H.-m., Casta˜nedaFern´andez, O., Chiang, P.- y., Studer, C. 和 Goldstein, T. Wrapnet：采用超低精度算术的神经网络推理。载于国际学习表征会议 *ICLR 2021*。OpenReview，2021年。Odena, A., Dumoulin, V. 和 Olah, C. 反卷积与棋盘格伪影。*Distill*，1(10):e3，2016年。Pappalardo, A. Xilinx/brevitas，2021年。网址 https://doi.org/10.5281/zenodo.3333552。Ronneberger, O., Fischer, P. 和 Brox, T. U-Net：用于生物医学图像分割的卷积网络。载于国际医学图像计算与计算机辅助干预会议，第234–241页。施普林格，2015年。Roy, J., Girgis, R., Romoff, J., Bacon, P.-L. 和 Pal, C. 通过约束强化学习直接规范行为。*arXiv* 预印本 *arXiv:2112.12228*，2021年。Sakr, C., Wang, N., Chen, C.-Y., Choi, J., Agrawal, A., Shanbhag, N. 和 Gopalakrishnan, K. 深度网络超低精度训练的累加位宽缩放。*arXiv* 预印本 *arXiv:1901.06588*，2019年。Salimans, T. 和 Kingma, D. P. 权重归一化：一种加速深度神经网络训练的简单重参数化方法。神经信息处理系统进展，29，2016年。Shannon, C. E. 通信的数学理论。贝尔系统技术期刊，27(3):379–423，1948年。Shi, W., Caballero, J., Husz´ar, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D. 和 Wang, Z. 使用高效的子像素卷积神经网络进行实时单幅图像和视频超分辨率。载于*IEEE*计算机视觉与模式识别会议论文集，第1874–1883页，2016年。

Umuroglu, Y., Fraser, N. J., Gambardella, G., Blott, M., Leong, P., Jahre, M. 和 Vissers, K. Finn：一种用于快速、可扩展二值化神经网络推理的框架。载于*2017*年 *ACM/SIGDA* 国际现场可编程门阵列研讨会论文集，FPGA ’17，第65–74页。ACM，2017年。

Wang, N., Choi, J., Brand, D., Chen, C.-Y. 和 Gopalakrishnan。使用 8位浮点数训练深度神经网络。神经信息处理系统进展，31，2018年。

Xie, H., Song, Y., Cai, L. 和 Li, M.。溢出感知量化：通过低比特乘累加运算加速神经网络推理。载于第二十九届国际人工智能联合会议国际会议论文集，第868–875 页，2021年。

Yang, C., Yang, Z., Khattak, A. M., Yang, L., Zhang, W., Gao, W. 和 Wang, M.。通过 L1正则化对卷积神经网络进行结构化剪枝。*IEEE Access*，7:106385–106394，2019年。

Zhang, X., Colbert, I., 和 Das, S. 使用联合逐层通道剪枝与均匀量化学习低精度结构化子网络。应用科学，12 (15):7829，2022。