arXiv:2304.06745v1 [cs.LG] 13 Apr 2023

面向 FPGA 和专用集成电路的海森矩阵感知量化神经网络端到端协同设计

JAVIER CAMPOS、JOVAN MITREVSKI和NHAN TRAN，美国费米国家加速器实验室ZHEN DONG、Amir Gholami*和 MICHAEL W. MAHONEY†，美国加利福尼亚大学伯克利分校

Javier Duarte，美国加利福尼亚大学圣迭戈分校

我们构建了一个端到端工作流，用于训练和部署协同设计的神经网络（NN），以便在现场可编程门阵列（FPGA）和专用集成电路（ASIC）硬件上高效运行。我们的方法结合了神经网络的海森感知量化（HAWQ）、量化开放神经网络交换（QONNX）中间表示，以及 hls4ml 工具流，可将神经网络转换为 FPGA 和 ASIC 固件。通过这一统一的开源工作流，非专业人士也能轻松实现高效的硬件神经网络，并将其部署于广泛科学和工业领域的实时机器学习应用中。我们以粒子物理学应用为例，展示了该工作流；其中的触发决策必须以欧洲核子研究中心大型强子对撞机（LHC）每秒 40 MHz 的碰撞率实时运行。由于碰撞率很高，所有数据处理都必须在定制 ASIC 和 FPGA 硬件上完成，并满足严格的面积与延迟要求。基于这些约束，我们针对模拟的 LHC 质子-质子碰撞中的高动量粒子喷注，实现了一个经过优化的混合精度神经网络分类器。

其他关键词和短语：神经网络、现场可编程门阵列、固件、高层次综合 s

*还与国际计算机科学研究所有关。
†还与国际计算机科学研究所和劳伦斯伯克利国家实验室有关。

作者地址：Javier Campos，jcampos@fnal.gov；Jovan Mitrevski，jmitrevs@fnal.gov；Nhan Tran，ntran@fnal.gov，费米国家加速器实验室，伊利诺伊州巴达维亚，美国；Zhen Dong，zhendong@berkeley.edu；Amir Gholami，amirgh@berkeley.edu；Michael W. Mahoney，mmahoney@stat. berkeley.edu，加利福尼亚大学伯克利分校，加利福尼亚州伯克利，美国；JavierDuarte，jduarte@ucsd.edu，加利福尼亚大学圣迭戈分校，加利福尼亚州拉霍亚，美国。

2023年。稿件已提交至 ACM

目录

摘要目录 1 引言 2 背景与相关工作 2.1 量化 2.2 自动位宽选择 2.3 固件生成工具 3 实验设置 3.1 数据集 3.2 模型与损失定义 3.3 指标：位运算与稀疏度 4 量化感知训练 4.1 同质化量化 4.2 混合精度量化 5 转换为 QONNX 5.1 中间表示 5.2 模型转换 5.3 导出后处理 6 硬件生成 6.1 hls4ml 导入 6.2 综合结果 7 总结 参考文献

1
2
3
4
4
5
5
6
6
7
7
7
8
9
11
11
12
12
13
13
14
16
17

1 引言

机器学习（ML）广泛应用于大数据处理，并且随着数据速率持续提升，其重要性日益凸显。尤其是尽可能靠近数据源进行的机器学习，即 边缘机器学习，对于科学和工业应用都越来越重要，包括数据压缩、数据量缩减以及用于实时决策的特征提取等应用 [1]。然而，由于面积、功耗和延迟方面的限制，在边缘端集成机器学习颇具挑战，深度学习（DL）和神经网络（NN）模型尤其如此。将神经网络部署于边缘端应用，需要精心优化训练方案，并针对推理精细调优实现。这通常需要采用现场可编程门阵列（FPGAs）和专用集成电路（ASICs）等高效计算平台。在系统和任务约束下开发神经网络算法并将其部署于硬件，是一个多步骤的 协同设计 过程，涉及广阔的决策空间。其中包括与 量化相关的选项，即使用低精度运算。本文提出了一套完全开源、非专业人士也可使用的端到端工作流，用于神经网络量化并将其部署于现场可编程门阵列和专用集成电路。

量化感知训练（QAT）已被证明能够有效缩减现场可编程门阵列 [10,11, 15, 17, 28, 33] 上的模型规模。通过量化感知训练，大型神经网络可以量化至 8 位或更低位宽，同时保持与基线相当的准确率。量化神经网络（QNN）通常能显著缩小模型规模并降低延迟。海森感知量化（HAWQ） [48] 是一种面向 PyTorch [38] 的混合精度纯整数型量化框架，具有广阔的应用前景。HAWQ 利用二阶（海森矩阵）信息指导混合精度，能够将模型量化到极低位宽。在这种方法中，敏感层（由海森信息确定）采用较高精度，而非敏感层则采用较低精度。现场可编程门阵列是这一方法的理想应用场景：与图形处理器等其他硬件相比，现场可编程门阵列对混合精度计算的支持要好得多，因此能够从中受益。

尽管这些特性使 HAWQ 成为面向现场可编程门阵列的量化感知训练的一个有吸引力的选择，但目前尚无简化流程可将其直接部署到现场可编程门阵列上。为此，我们为 HAWQ 增添了功能，使其能够将量化神经网络导出为量化开放神经网络交换（QONNX） [37] 中间表示。随后，hls4ml 可以读取 QONNX 表示；hls4ml 是一个用于神经网络转换并将其部署到现场可编程门阵列和专用集成电路硬件上的开源 Python 库。hls4ml 软件包兼顾硬件专家和非专家的易用性，并且足够灵活，能够将具有多种量化位宽的量化神经网络部署到不同的现场可编程门阵列和专用集成电路平台。它是科研和工业边缘机器学习应用中广受欢迎的工具 [2, 22, 36]。

为展示端到端工作流的性能，我们开发了一种用于粒子物理学实时决策的神经网络。欧洲核子研究中心大型强子对撞机（大型强子对撞机）是世界上规模最大、能量最高的粒子加速器。探测器中的粒子每 25 ns 发生一次碰撞，产生数十 TB 的数据。受存储容量和处理能力限制，并非每个碰撞事件都能被记录。在这些实验中，在线触发系统会筛选数据，仅存储最“有趣”的事件以供离线分析。通常，触发系统利用有趣物理现象的简单特征（例如沉积能量很高的事件或粒子组合异常的事件），决定保留探测器中的哪些事件。触发系统包含多个阶段，第一阶段称为一级触发器（L1T）， [3, 43], 使用定制的专用集成电路或现场可编程门阵列以 40 MHz 处理数据。过去几年里，大型强子对撞机提高了质心碰撞能量和瞬时亮度，使实验能够搜寻日益罕见的信号。随着累积数据量急剧增加，研究人员正尝试将机器学习方法用于 L1T [13, 14]的各项任务。其中一项任务是喷注标记：利用喷注亚结构 信息 [31, 32]，识别并分类夸克和胶子衰变及强子化产生的准直粒子簇射。机器学习方法展现出巨大潜力，能够超越

传统算法，提升我们识别不同喷注来源和发现新物理相互作用的能力 [8, 42]。

在为粒子物理学应用开发实时决策神经网络的背景下，本文的原创贡献如下：

- 我们利用 QONNX 格式来表示采用任意精度和混合精度量化的量化神经网络，从而扩展 HAWQ，使其支持 QONNX中间表示。
- 我们对喷注标记基准测试中使用的多层感知机（MLP）模型进行海森感知量化，并详细研究量化对模型各层性能和效率的影响。
- 我们使用 hls4ml 展示经 HAWQ 训练的神经网络在现场可编程门阵列上的硬件实现所需的优化资源和延迟。

本文其余部分安排如下。在第2节中，我们介绍将部署于现场可编程门阵列和专用集成电路上的量化神经网络协同设计工作流的关键步骤，包括量化和 HAWQ 概述。第3节介绍任务，并讨论神经网络的评估和训练方式。第4节展示采用同质化量化和基于海森矩阵的量化所得的初步量化感知训练结果，第5节则介绍我们对 HAWQ 的扩展。随后，第6节将介绍神经网络的固件实现，尤其是资源使用情况和估计延迟。最后，第7节将进行总结。

2 背景与相关工作

本节概述量化和 HAWQ（见第 2.1 节）；随后介绍自动位宽选择（见第 2.2 节）和固件生成工具（见第 2.3 节）。

2.1 量化

神经网络中的量化是指降低输入、权重和激活值所用的数值精度。在均匀仿射量化中，数值通过如下映射函数量化为较低精度的整数：

$$q = \text{quantize}(r) = \text{Clip}(\text{Round}((r/S) - Z), \alpha, \beta), \tag{1}$$

其中 r 是浮点输入，S 是缩放因子，Z 是零点 [23]。Round 函数执行舍入到最近值的操作，并将结果裁剪/限制在 α 和 β之间。由于所有量化区间的间距均匀，公式1 中的这一映射函数称为均匀量化。量化区间大小可变的非均匀量化方法在硬件中的实现难度更高 [34]。实数可通过反量化从量化值中恢复：

$$\tilde{r} = \text{dequantize}(q) = S(q + Z), \tag{2}$$

其中，$\tilde{r} - r$ 称为量化误差。缩放因子将给定范围内的实数划分为 2^b 个区间，

$$S = \frac{\beta - \alpha}{2^b - 1}, \tag{3}$$

其中，$[\alpha, \beta]$ 是裁剪范围，b 是位宽。选择裁剪范围的过程称为校准。一种简单的方法是使用数值的最小值和最大值，即$\alpha = r_{\text{最小值}}$和$\beta = r_{\text{最大值}}$。这是一种非对称量化方案，因为裁剪范围相对于输入不一定对称，即可能出现 $-\alpha \neq \beta$。对称量化方法使用对称的裁剪范围 $-\alpha = \beta$，例如$-\alpha = \beta = \max(|r_{\text{最大值}}|, |r_{\text{最小值}}|)$，并将零点替换为 $Z = 0$。

海森感知量化的最新版本 HAWQv3 [48], 提出了一种全新的计算图，并基于其先前的工作 [18, 19]引入了自动位宽选择策略。在 HAWQv3 中，为简便起见，本文简称其为 HAWQ；量化遵循公式 1，并受到额外的硬件启发式限制。HAWQ 的整个计算图仅使用整数乘法、加法和位移执行，不涉及任何浮点运算或整数除法。权重的裁剪范围是对称的 $\beta = 2^b - 1 = -\alpha$，而激活值可以采用对称或非对称方式。通过分析不同批次的输出范围，预先计算实数缩放因子，并在推理时固定，这一过程称为静态量化。HAWQ 将所有缩放因子限制为二进有理数（形式为 $b/2^c$的有理数，其中 b和 c为整数），从而避免浮点运算和整数除法。为说明一种典型计算，考虑一个输入为 h、权重张量为W的层。在 HAWQ 中， h和W分别量化为 $S_h q_h$和 $S_W q_W$，其中 S_h和 S_W为实数缩放因子， q_h和 q_W为相应的量化整数值。输出结果记为 a，可计算为

$$a = (S_W S_h)(q_W * q_h), \tag{4}$$

其中， $*$表示低精度整数矩阵乘法（或卷积）。随后，结果会量化为 $S_a q_a$，供下一层使用：

$$q_a = \text{Int}\left(\frac{a}{S_a}\right) = \text{Int}\left(\frac{S_W S_h}{S_a}(q_W * q_h)\right), \tag{5}$$

其中 S_a 是输出激活值的预计算缩放因子。通过使用整数乘法和位移实现公式5 ，可避免浮点运算和整数除法。

2.2 自动位宽选择

已有许多方法用于衡量量化敏感度，或开发自动位宽设置方案。例如，HAQ [46] 提出了一种强化学习（RL）方法，用于自动确定量化策略。该方法让强化学习智能体接收来自硬件模拟器的直接延迟和能耗反馈。文献 [47] 将神经架构搜索（NAS）建模为可微神经架构搜索（DNAS）问题，以高效探索搜索空间。文献 [35] 提出将周期函数用作正则化项，通过正则化将权重推向可编码为整数的离散点。这些基于探索的方法的一个缺点是，它们通常对超参数或初始化较为敏感。最近，AutoQkeras [15] 被提出用于在给定一组资源约束和准确率指标（例如能耗或位宽）的情况下，同时优化模型面积（以 FPGA 设计中的逻辑元件数量衡量）和准确率。与以往方法不同，HAWQ [19] 引入了一种基于二阶敏感度指标自动确定混合精度设置的方法。具体而言，海森矩阵（尤其是最大的 Hessian 特征值）可用于衡量敏感度。文献 [18], 进一步扩展了这一方法，使用所有 Hessian 特征值的平均值计算敏感度指标。

2.3 固件生成工具

尽管机器学习方法在边缘设备上已取得令人鼓舞的成果，但将这些算法部署到现场可编程门阵列上颇具挑战，往往非常耗时，而且需要领域专家和工程师的专业知识。为解决这一问题，人们探索了多个方向。一个方向是现场可编程深度神经网络（FP-DNN） [25], , 它是一种框架，以 TensorFlow 描述的深度神经网络（ DNN）为输入，并通过寄存器传输级（RTL）与高级综合（HLS）的混合模板自动生成硬件实现。另一个方向是 fpgaConvNet [45], , 它专门针对卷积神经网络（CNN）， 是一个用于优化 CNN 在现场可编程门阵列上映射的端到端框架。有趣的是,

fpgaConvNet 提出了一种多目标优化问题，以兼顾卷积神经网络工作负载、目标设备和所关注的指标。

这些工具以及其他工具表明，人们越来越希望以更快、更流畅的流程，在边缘设备上部署更高效、规模更大的机器学习模型。这一需求广泛存在于科学和工业应用场景中 [1, 5]。粒子物理学应用尤其是对这类工具的严峻考验。这是因为它们对计算延迟和数据带宽有极高要求，同时还受到低功耗、高辐射和低温环境等条件的限制。此外，粒子物理学从业者未必是机器学习专家或硬件专家，其应用和系统需要开源工具（尽可能如此），并能灵活部署在不同的现场可编程门阵列和专用集成电路平台上。hls4ml 工具正是源于这类应用场景，并支持多种架构和框架，例如 Keras [12], QKeras [15, 24], 和 PyTorch [38]。目前，在不断壮大的科学界群体的支持下，该工具支持的架构、框架、硬件优化和目标设备范围正在稳步扩大。另一款工具 FINN [7, 44] 来自 AMD/Xilinx，旨在通过生成高级综合（HLS）代码，解决将神经网络（更具体地说，量化神经网络）部署到现场可编程门阵列上的问题。这两款工具都能简化深度学习模型的部署流程，尽可能提高效率，同时无需投入大量开发精力和时间。hls4ml 和 FINN 的目标相似，但在流程、层支持和目标优化方面有所不同。两者都支持开源交换格式 QONNX，该格式能够以任意精度表示量化神经网络，从而实现不同流程之间的互操作性。更广泛地说，我们应当注意，这种方式非常适合 HAWQ，因为 HAWQ 可以面向多种硬件生成工具。不过，在本研究中，我们仅关注 hls4ml；它支持现场可编程门阵列和专用集成电路实现，并针对更广泛的位宽进行了优化。

3 实验设置

本节介绍我们为粒子物理学应用研究的机器学习基准任务。如上所述，尽管科学和工业领域的应用范围广泛得多，但粒子物理学应用尤其适合用于对我们的端到端工作流进行压力测试。粒子物理学基准的开发理念详见参考文献 [21], ，我们的喷注标记基准是其中介绍的三个基准之一。

3.1 数据集

我们采用高-p_T 喷注分类基准来评估性能。粒子喷注是大型强子对撞机（LHC）中高能质子-质子碰撞产生的夸克和胶子的辐射模式。这些喷注穿过 ATLAS 或 CMS 等探测器时，会在各种子探测器中留下信号，例如硅径迹探测器、电磁量能器或强子量能器，以及 μ 子探测器。随后通过喷注重建算法将这些信号合并。我们使用参考文献 [20] 中介绍的基准，其中包含质子-质子碰撞产生的模拟粒子喷注的 54 个特征。在这 54 个高层特征中，有 16 个是依据参考文献 [20] 的表1选出的。这些特征结合了质量（“有量纲”）和形状（“无量纲”）观测量。数据集 [39] 包含 870,000 个喷注，并分为两组：包含 630,000 个喷注的训练集和包含 240,000 个喷注的测试集。数据集经过了预处理：对所有特征进行标准化处理，即减去均值并缩放，使方差为 1。任务是区分喷注分别源自以下五种粒子中的哪一种：W玻色子、Z玻色子、轻夸克（q）、顶夸克（t）或胶子（g）。各观测量和粒子喷注的说明见参考文献 [16]。此外，我们用正确分类的喷注数除以分类喷注总数，计算准确率。

3.2 模型与损失定义

我们采用参考文献中提出的架构实现所有模型。 [20], 一个包含三个隐藏层的多层感知机，各层分别有 64、32 和 32 个节点。基线模型是该多层感知机的浮点实现，即未进行量化。所有隐藏层均使用 ReLU 激活值，输出则是经过 Softmax 激活函数处理的五类概率向量。我们的目标是最小化经验损失函数

$$\mathcal{L}_c(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(\mathbf{x}_i), \mathbf{y}_i) = \frac{1}{N}\sum_{i=1}^{N} \ell(\hat{\mathbf{y}}_i, \mathbf{y}_i), \tag{6}$$

其中， ℓ 是分类交叉熵损失函数， N 是训练样本的数量。该模型记为 f_θ，使用参数 θ将每个输入 $\mathbf{x}_i \in \mathbb{R}^{16}$ 映射为预测值 $\hat{}\mathbf{y}_i \in [0,1]^5$。随后，将预测值与真实标签 $\mathbf{y}_i$ 进行比较，以最小化经验损失。我们通过在损失中加入额外的惩罚项，对神经网络进行 L_1 正则化训练，

$$\mathcal{L}(\theta) = \mathcal{L}_c(\theta) + \lambda \sum_{j=1}^{L} \|\mathbf{W}_j\|_1, \tag{7}$$

其中，所添加的惩罚项是权重矩阵的逐元素范数，$\mathbf{W}_i$ 是第 j 个层的权重矩阵的“向量化”形式， L 是模型中的层数。 L_1 正则化项由可调超参数 λ进行缩放。通常， L_1 正则化用于防止过拟合，使统计模型能够在训练数据之外实现更好的泛化。众所周知，它还能够促进稀疏度，这有利于减少计算量。第4 节讨论了 L_1 正则化对量化神经网络的影响，包括其性能和下文讨论的其他指标。

3.3 指标：位运算次数与稀疏度

与浮点运算类似，量化神经网络中的位运算次数（BOPs） [6]用于估算模型复杂度和每次推理的运算次数。研究表明，BOPs 可以准确预测硬件加速器的面积，进而预测处理单元的功耗 [30]。因此，BOPs 是一种易于计算的指标，可用于近似量化神经网络的总面积。一个全连接层的位运算次数可按以下方式估算，其中输入激活值为 b_a 位，权重为 b_W 位：

$$\text{BOPs} \approx mn((1-f_p)b_a b_W + b_a + b_W + \log_2(n)), \tag{8}$$

其中 n和 m 分别表示输入特征数和输出特征数，$(1-f_p)$ 项表示剪枝权重所占的比例（即权重为零的比例）。根据公式 8，BOPs 的数量与稀疏度成反比。稀疏模型更受青睐，因为高层次综合（HLS）会在固件实现中优化掉零权重乘法。这是 HLS 的一项极具吸引力的特性，也使 BOPs 成为值得关注的指标。我们测量每种量化方案的总 BOPs，并分析其与准确率（见第 4 节）和硬件使用量（见第 6 节）的关系。

4 量化感知训练

本节将讨论同质化量化和混合精度量化的训练流程。我们首先在第4.1 节讨论单一位宽量化，这也称为同质化量化。接着，在第4.2节讨论混合精度量化，包括它如何大幅提升分类性能，以及它的缺点。具体而言，在第4.2.2节介绍一种自动选择

利用二阶海森信息确定神经网络中每一层的位宽，以及一种在位宽选择过程中施加硬件约束而得到的方法。

4.1 同质化量化

以相同位宽量化所有层很简单，但可能会导致性能显著下降。在表 1中，我们展示了使用 HAWQ 进行同质化量化时，从 INT12 到 INT4 不同位宽设置下的准确率。正如预期，量化到低于 INT8（尤其是低于 INT6）时，性能会显著下降。为解决这一问题，我们在训练期间采用了两种正则化技术： L_1 正则化和批量归一化（批归一化） [29]。批归一化通过对激活值进行归一化，使其在整个训练过程中保持更稳定的分布，并产生更平滑的损失曲面 [40]。虽然使用批归一化能提升所有量化方案的性能，但对于 INT6 和 INT4 量化，它仍无法恢复基准准确率。同样， L_1 正则化能在一定程度上改善模型，但无法使性能恢复到基线。因此，对于低于 8 位精度的量化，仅通过一种位宽设置对模型进行同质化量化是不够的。

精度		基线 [%]	L_1 [%]	批归一化 [%]	L_1+批归一化 [%]
权重	输入				
INT12	INT12	76.916	72.105	77.180	76.458
INT8	INT8	76.605	76.448	76.899	76.879
INT6	INT6	73.55	73.666	74.468	74.415
INT4	INT4	62.513	63.167	63.548	63.431
FP-32	FP-32	76.461	76.826	76.853	76.813

表1. 同质化量化下的分类性能。所有权重、激活值和输入均以相同精度量化。模型在使用和不使用 L_1 正则化及批归一化的情况下进行训练。精度达到 INT8 及以上时，准确率恢复至基线水平；但在 INT6 及以下时，准确率低于基线。

除了采用正则化技术外，我们还可以提高输入量化位宽。在 HAWQ 中，输入会先经过量化，再传入第一层，从而确保所有运算都仅使用整数。一个潜在的失效点是：低位宽下输入量化引入的误差可能会导致分类喷注所需的关键特征丢失。我们将输入的精度与权重和激活值的精度解耦，并将其提高到 INT16。图 1 展示了权重为 8 位及以下、激活值采用不同位宽时的结果。我们发现：（1）提高激活位宽可显著改善 INT4 和 INT6 权重的分类性能；（2）使用 INT16 量化输入也能取得类似的改善，但代价是硬件资源使用量增加；（3） L_1 和批归一化（单独应用或同时应用）都不足以使准确率恢复到基线水平。在这项研究中，批归一化并不那么理想，因为批统计参数以浮点值实现，从而增加了延迟和内存占用。一种选择是量化这些值，或者（更有希望的是）应用批归一化折叠。其思路是利用批归一化参数更新全连接层（或卷积层）的权重和偏置，以移除批归一化并提高推理效率。然而，我们按照参考文献 [48], 中概述的流程探索批归一化折叠后，发现它对模型性能几乎没有影响。如前所述， L_1 会生成稀疏矩阵，从而减少硬件所需的位运算次数。因此，在后续章节中，我们将继续在训练期间使用 L_1进行混合精度量化。图 1 表明，采用更细粒度的量化设置可显著提升模型性能。然而，手动调整所有这些量化设置可能非常耗时

且效果可能并非最优。我们需要一种经过优化的位设置方案，以同时最小化损失和硬件使用量。在下一小节中，我们将探讨混合精度量化。我们将输入位宽固定为 INT16。虽然还可以进一步优化这一选择，但这样更便于与其他工作直接比较 [15, 20, 21, 26]。

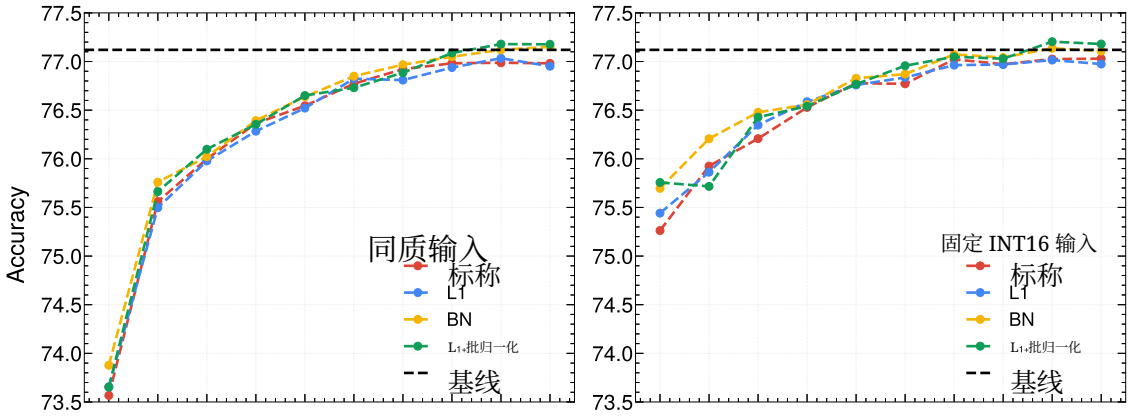


图1. 同质化量化下的模型性能。权重精度标注在“w”之后，激活值精度标注在“a”之后。模型使用 L_1 正则化和批归一化进行训练。可以看出：(1) 16位输入可提升所有位宽设置下的模型性能；(2) 更大的激活位宽可提高准确率；(3) L_1 和批归一化（单独或同时应用）对性能均无积极影响。

4.2 混合精度量化

4.2.1 暴力搜索。混合精度量化旨在通过让某些层保持比其他层更高的精度来提升性能。超越同质化量化所面临的基本问题是：若采用简单实现，确定位宽设置的搜索空间会随着层数呈指数级增长。与 ResNet-50 [27] 等深度卷积神经网络相比，我们的模型架构中的多层感知机搜索空间要小得多，因为我们的多层感知机只有 3 个隐藏层。不过，假设有 5 种位宽选项，要为我们的多层感知机分类器寻找混合精度设置，其中包括 4 层全连接层的权重和激活值，其搜索空间为 $((2)(4))^5 = 32,768$ 种组合。这种搜索并不现实，尤其是对于需要频繁重新训练模型或需要深度神经网络的应用。已有若干方法被提出，用于解决手动搜索最优位宽配置的问题 [9, 18, 41, 46, 47]。我们采用了基于海森矩阵信息的文献 [18], 方法，并观察基于海森矩阵的解在暴力搜索空间中的相对位置。

4.2.2 海森矩阵感知量化。 如第 4.1 节所述，提高激活值的精度显著改善了性能，这表明某些层比其他层对量化更敏感。我们采用 HAWQv2 [18] 首次提出的工作，来确定模型基线 32 位浮点实现中各层的相对敏感度。敏感度指标使用 Hutchinson算法计算，

$$\mathrm{Tr}(H) \approx \frac{1}{k}\sum_{i=1}^{k} z_i^{\mathsf{T}} H z_i = \mathrm{Tr}_{\mathrm{Est}}(H), \tag{9}$$

其中 $H \in \mathbb{R}^{d\times d}$ 是损失函数相对于所有 d模型参数的二阶偏导数构成的海森矩阵， $z \in \mathbb{R}^d$ 是一个随机向量，其各分量独立同分布地从拉德马赫分布中采样，而 k 是用于迹估计的 Hutchinson 步骤数。图 2 展示了基线模型中各层的平均海森迹（即我们的敏感度指标），并采用对数刻度。前两层最为敏感，第一层的敏感度是第二层的 7 倍。因此，网络前两层必须采用更大的位宽，而最后两层则可以进行更激进的量化。虽然海森迹提供了敏感度指标，但无法直接据此确定位配置。相反，参考文献 [18] 通过检查相应的 i 项来设定各层的位宽 Ω，其定义如下：

$$\Omega = \sum_{i=1}^{L} \Omega_i = \sum_{i=1}^{L} \overline{\mathrm{Tr}}(H_i)\|Q(W_i) - W_i\|_2^2, \tag{10}$$

其中， Q是量化函数， $\|Q(W_i) - W_i\|_2^2$是量化扰动的平方 L_2范数，Tr 是平均 Hessian 迹。我们采用与参考文献 [18],相同的技术：针对给定的一组量化方案，计算二阶扰动量 Ω，并选择最小的 Ω 。此过程完全自动化，无需人工干预。

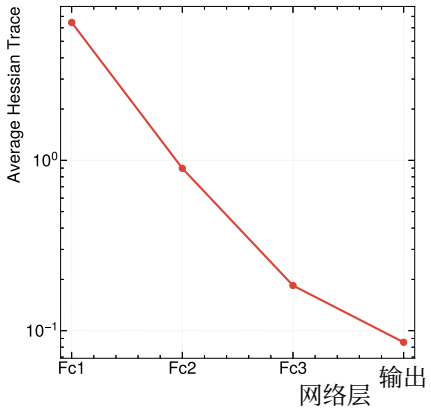


图 2. MLP 中每个全连接（Fc）层的平均 Hessian 迹。海森矩阵用作量化的敏感度指标，并根据各层的迹对其进行排序。前两层的数值显著高于其他层，表明它们在较低位宽下更容易产生误差。根据平均 Hessian 迹为每层分配位宽设置，即迹较高的层分配更高的精度。

我们遵循 HAWQ 中概述的流程 [48]，通过总位运算次数约束式10。我们构建一个整数线性规划（ILP）优化问题，其目标是在满足约束条件的同时最小化 Ω_i。我们建立 ILP 问题，以自动确定分类器在各种位运算次数上限下的位宽设置，并将这些解与蛮力搜索和同质化量化方法进行比较。

4.2.3 量化感知训练结果。根据图1提供的信息，我们基于同质化量化的初始实现，应用所有可能的位宽设置。我们探索权重位宽 $b_W = \{4, 5, 6, 7, 8\}$，并设置激活位宽 $b_a = b_W + 3$，以防止饱和并进一步缩小搜索空间。所有模型均训练 100 个训练轮次，采用 L_1 正则化，并使用 INT16 量化输入。图3 展示了模型准确率。稿件已提交至 ACM

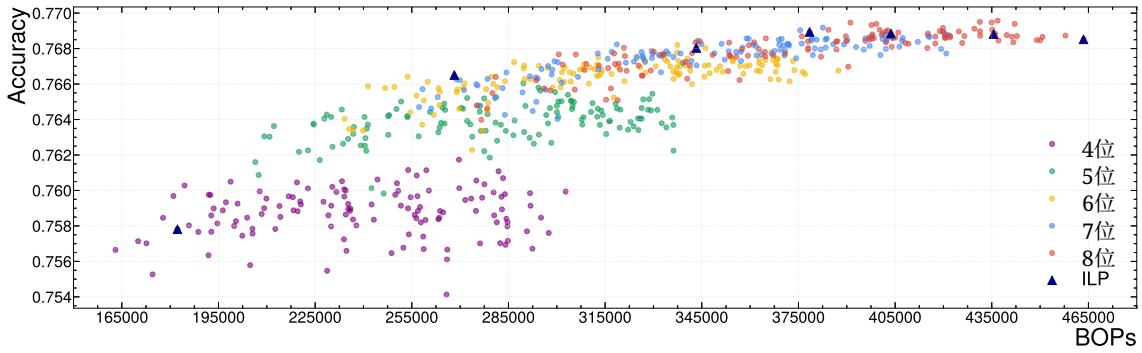


图 3. 使用权重位宽 $b_w = \{4, 5, 6, 7, 8\}$ 进行暴力搜索量化。每个数据点根据第一个全连接层的位宽进行颜色编码。该层在量化中的重要性与观察到的聚类相吻合，性能更高的模型使用了更大的位宽。图中展示了基于 ILP 的解。所有 ILP 解都在量化误差和位宽之间进行权衡，且通常位于各自聚类中 BOPs 最低的区域。

所有权重位宽组合对应的位运算次数 b_W。数据点根据第一层的位精度进行颜色编码。若干数据点表明，准确率已完全或几乎完全恢复至基准准确率（76.853%）。大多数数据点可根据第一层的位宽聚类，因为随着第一层位宽增加，模型准确率通常也会提高。我们还可以从图3 中看出，第一个全连接层的位宽对模型的最终性能影响很大。在表现最佳的 100 个模型中，66 个模型的第一稠密层采用 INT8，33 个模型采用 INT7 权重。这与图2所示的平均海森迹一致，表明第一层对量化最敏感，其敏感度是 7×，相较于敏感度排名第二的层。在表现最佳的模型中，我们观察到后续层采用 7 位和 8 位的频率显著降低。与前两层相比，后续层的位宽对分类的影响较小。

同时展示了方程 10 的 ILP 解。这些解针对 7 种不同的 BOPs 约束求得，约束范围从 250k 到 550 k，步长为 50 k。正如预期，随着 BOPs 约束增大，前两层所选的精度也随之提高。因此，我们开始看到更多靠近 8位聚类的 ILP 解。ILP 解也往往位于各自局部聚类中 BOPs 较低的一端。将暴力搜索量化与 ILP 解并列展示后，使用海森信息的优势便更加明显。虽然无法保证得到最优解，但海森矩阵能够为混合精度量化提供稳定可靠的解。这对于需要进行量化以满足大型强子对撞机 40 MHz碰撞率所要求的资源约束和推理时间的深度学习模型而言，十分理想。

5 转换为 QONNX

5.1 中间表示

为提高互操作性和硬件可访问性，开放神经网络交换（ONNX）格式应运而生，旨在为描述机器学习算法的计算图制定开放标准 [4]。ONNX 定义了丰富且通用的算子集，使开发者和研究人员能够更自由地在不同框架、工具、编译器和硬件加速器之间进行选择。目前，ONNX 对量化算子提供了一定支持，包括 QuantLinear、QLinearConv 和 QLinearMatMul。然而，ONNX 在表示任意精度以及

低于 8 位精度的超低位量化方面仍显不足。为解决这些问题，近期的研究 [37] 利用现有 ONNX 算子引入了量化-裁剪-反量化（QCDQ），并提出了一种通过新增算子表示量化神经网络的扩展方案，称为 QONNX。QONNX 引入了三个新的自定义算子：量化、双极和截断。这些自定义算子支持均匀量化，并抽象掉更细节的实现信息，使中间表示图比 QCDQ 更灵活，抽象层次也更高。

基于这些原因，我们采用 QONNX 格式表示 HAWQ 神经网络，利用 HAWQ 的超低精度和 QONNX 的抽象，以适配两种现场可编程门阵列综合工具：hls4ml 和 FINN [7, 44].[1] 此外，我们还在模型导出器中纳入 ONNX 格式，以表示量化神经网络。在接下来的小节中，我们将介绍如何设置、导出和验证，以便在 QONNX 和 QCDQ 中间表示中表示 HAWQ 神经网络。

5.2 模型转换

在 PyTorch 中，导出为 ONNX 格式需要通过跟踪来完成。该过程会捕获针对某个输入执行前向传播时调用的所有操作。PyTorch 通过 torch.jit API 提供跟踪功能。跟踪模型会返回一个由 PyTorch 即时编译器优化的可执行对象。该对象包含模型结构和原始参数。跟踪不会记录任何控制流，例如 if 语句和循环。对于任何输入，返回的可执行对象始终运行相同的跟踪计算图；对于预期会根据输入和模型状态执行不同操作的函数或模块而言，这可能并不理想。随后，使用该可执行对象构建 ONNX 图，将其中的操作和参数转换为标准 ONNX 算子。通常，所有 PyTorch 模型都通过这一流程转换为 ONNX；我们在此基础上扩展现有系统，为 HAWQ 添加对 QONNX 算子的支持。

HAWQ 中的层和 QONNX 中的算子都需要经过额外步骤才能支持跟踪和导出。对于 HAWQ 中的每个量化层，我们都会实现一个对应的“导出”层。这些专用导出层实现前向传播，并根据原始层的参数指定等效的 QONNX 算子。我们通过注册符号函数，使用`torch.onnx.register_custom_op_symbolic`来完成这一过程。这些符号函数将 HAWQ 层的操作分解为一系列 QONNX 节点。由于我们使用的是自定义 QONNX 节点，因此还必须通过`torch.onnx` API 注册这些节点。这些准备步骤共同定义了 HAWQ 到 QONNX 的转换。在导出过程中，导出器会为每个访问到的算子查找已注册的符号函数。如果给定模型包含量化的 HAWQ 层或标准 PyTorch 层，就可以对其进行跟踪，最终将其转换为标准 ONNX 算子和 QONNX 算子。由于跟踪会记录计算，因此只要输入的维度和数据类型正确，输入值可以是随机的。使用 ONNX 算子和 QONNX 算子导出的模型如图4a所示。通过这些扩展，我们的导出器可以执行以下操作：

(1) 将包含 HAWQ 层的模型导出为 QONNX，并使用自定义算子支持广泛的位宽，同时将计算图保持在较高的抽象层级；以及 (2) 将包含 HAWQ 层的模型导出为标准 ONNX，并限制为 INT8 和 UINT8。

5.3 导出后处理

5.3.1 优化。 为了使用 hls4ml 或 FINN 创建固件，QONNX 图需要经过规范化，即执行若干优化步骤。QONNX 软件工具 [37] 提供了这些转换，如图 4b所示，其中对计算图应用了形状推断和常量折叠。图 4c 展示了最后一个优化步骤；

[1]导出量化神经网络的重点是 QONNX 中间格式。不过，QONNX 软件工具包支持转换为 QCDQ 格式。因此，HAWQ 可以面向 hls4ml 和 FINN，并间接支持所有其他 ONNX 推理加速器和框架。提交 ACM 的稿件

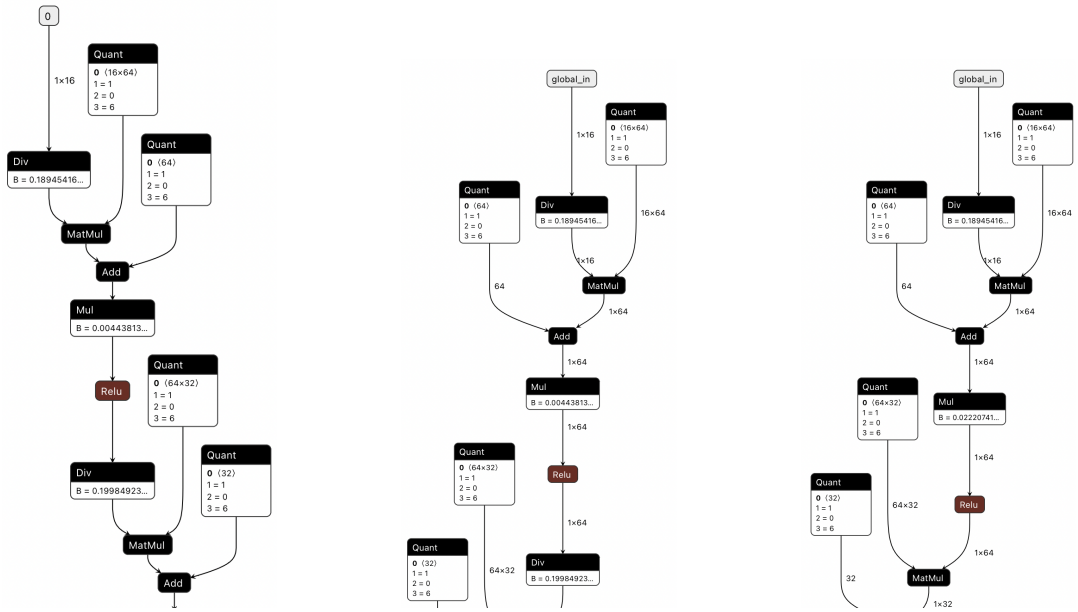


图 4. 导出后 QONNX 图的三个阶段。(a) 模型的前几层，包括尚未进行任何优化的量化全连接层。(b) 模型经过后处理清理操作后的前几层：常量折叠、形状推断，以及张量和节点重命名。(c) 最后的优化步骤：合并跨越 ReLU 激活值的节点。所有在 HAWQ 中实现的量化神经网络都可以导出为 QONNX 或 ONNX 中间表示，并按照各阶段所述进行转换。

我们会在 ReLU 激活函数之间合并缩放因子。由于 HAWQ 底层实现的相关原因，某些层前后各有一次缩放运算。有关这些缩放因子的详细说明，请参见第2.1节。为减少固件所需的运算次数，我们会在使用 ReLU 函数时合并缩放因子。但并非总能这样做，这取决于所使用的激活函数。

5.3.2 计算图评估。导出后，我们使用 QONNX 软件包 [37], 评估模型，确认已成功将模型从 HAWQ 转换为 QONNX。虽然重点一直放在多层感知机上，但导出功能并不局限于这一种架构。所有 HAWQ 层现在都支持通过已实现的符号函数导出为 QONNX。此外，借助 QONNX 软件包，可以轻松转换、优化、评估和验证导出的 HAWQ 模型。

6 硬件生成

本节介绍 HAWQ 在 hls4ml 硬件生成流程中的位置，并展示和讨论不同位宽配置下的总资源用量、位运算次数和分类性能。

6.1 hls4ml 接入

hls4ml 工作流会自动将神经网络的架构、权重和偏置逐层转换为代码，再使用 HLS 工具将其综合为寄存器传输级代码。此工作流的第一步是训练一个神经网络，以完成

某项任务，方法与使用 PyTorch、Keras、QKeras 或 HAWQ 时相同。对于 HAWQ，必须从模型导出 QONNX 图；不过，所有框架都可以（选择性地）执行这一步骤，而且最终这将成为首选流程。接下来，hls4ml 会将 QONNX 图转换为 HLS 项目，随后在工作流的最后一步中，可将其综合并部署到现场可编程门阵列或专用集成电路上。

所有结果均针对型号为 xcu250-figd2104-2L-e 的 Xilinx Kintex Ultrascale FPGA 进行综合。我们报告了不同资源的使用情况：数字信号处理器单元（数字信号处理器）、触发器（触发器）和查找表（查找表）。我们未报告块随机存取存储器（BRAM）这一密集型存储资源的使用情况，因为它在设计中的唯一用途是存储 Softmax 激活的预计算输出，而这些输出在所有量化方案中的数值精度都相同。仅使用 Vivado 2020.1 进行 RTL 综合，以构建实现神经网络所需的“裸”固件设计。所有神经网络均采用最大程度的并行化。在 hls4ml 中，通过“复用因子”配置并行化，该因子设定乘法器用于计算层输出的次数。完全并行的设计对应复用因子为 1。所有资源使用指标均基于 RTL 综合后的这一“裸”实现，且所有设计的时钟频率均为 200MHz。

6.2 综合结果

图 5展示了已实现设计的资源使用量与准确率之间的关系。量化位宽设置是随机选择的。性能较高的模型会使用更多资源。这是预期之中的，因为性能排名前 100 的模型在第一层（模型中最大的层）使用了更大的位宽。因此，随着准确率提高，我们预计资源使用量也会增加。查找表与准确率之间的关系最接近线性，而触发器和数字信号处理器的数量也会随准确率提高而增加。图 6中展示的位运算次数与资源之间的关系也表明，查找表数量与位运算次数呈线性关系，二者均随位宽和权重矩阵维度变化。所用查找表的数量取决于位宽，因为在位宽较低时，加法和乘法由查找表实现。而在位宽较大时，则会使用数字信号处理器，因为它们的效率要高得多。数字信号处理器提供定制数据通路，能够高效地实现一系列算术运算，包括乘法、加法、乘累加（MAC）以及字级逻辑运算。数字信号处理器的数据通路不如可编程逻辑灵活，但在乘法和乘累加运算方面效率更高。如图 6 所示，随着位运算次数增大，数字信号处理器的使用量也会增加，且在某些区间会大幅上升。从查找表切换到数字信号处理器取决于目标设备以及 Vivado HLS 内部针对特定位宽对数字信号处理器的偏好。在 Vivado 2020.1 中，位宽达到 11 位或更高时会转而使用数字信号处理器；低于这一上限的乘法则由查找表实现。这些运算的结果存储在触发器中，其数量稳步增加，变化幅度小于数字信号处理器。位运算次数达到 250k 之前，触发器数量以恒定速度增长，此后开始出现偏差。相邻位运算次数下触发器数量的不一致表明，两者之间的相关性较弱。这些偏差源于中间累加所需的精度，而所需触发器总数会因网络而异。

基线（BL）模型在调整权重后进行综合，且未进行任何微调。在 hls4ml 中，参数和计算均采用定点算术，各层可在训练后通过指定较低精度进行量化。定点数据类型将数据表示为整数位和小数位，格式为`ap_fixed<W,I>`。与先前结果一样，BL 模型的所有参数和计算均使用`ap_fixed<16,6>`，并且完全展开，即实现最大程度的并行化。我们将 BL 的逻辑综合结果与同质化量化模型和海森感知量化模型进行比较。从表1可以看出，同质化量化在低于 INT8 时开始出现性能下降，因此我们采用该量化方案与 BL 进行比较。在多个海森感知方案中，我们选择 BOPs约束最低的方案，即第一层、第二层、第三层和最终输出层的量化位宽分别为 4、4、5 和 4 位。稿件已提交至 ACM

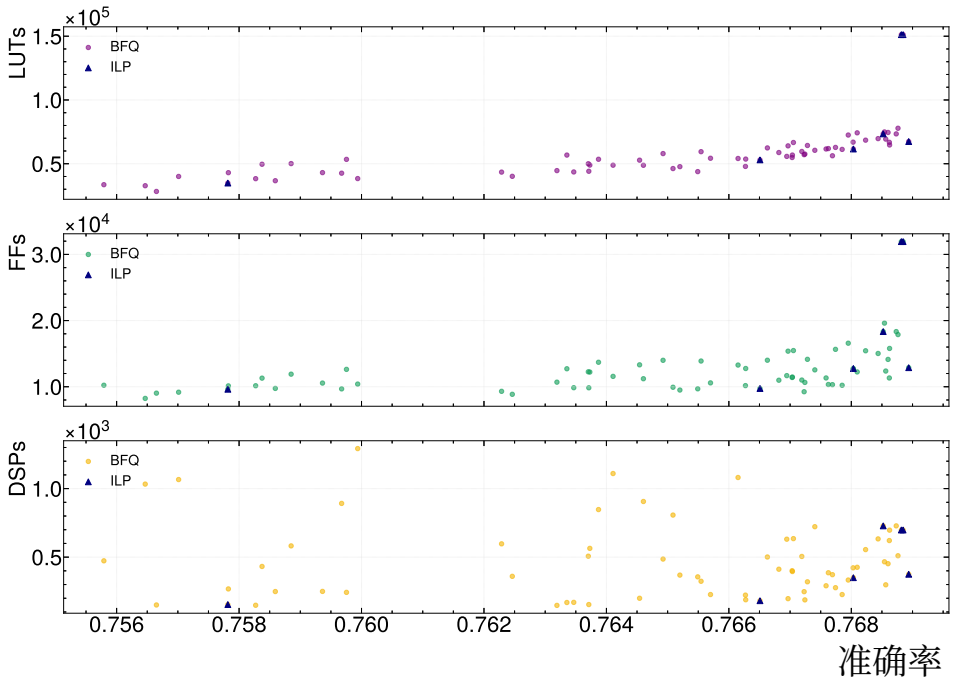


图5。采用权重位宽 $b_W = \{4,5,6,7,8\}$ 的部分暴力量化（BFQ）方案的资源使用情况。展示了查找表、触发器和数字信号处理器的使用量与准确率之间的关系，其中性能较高的量化方案也占用了较多资源。展示了由 BOPs 约束得到的整数线性规划问题的所有解。为保持准确率需要额外的逻辑元件；而准确率仅下降 1–2% 时，所有指标均可大幅改善。

层。表2展示了三种模型的综合结果：BL、INT8均匀量化以及海森感知方案。与 BL 模型相比，采用 INT8均匀量化后，数字信号处理器数量显著减少；采用混合海森感知量化后，数量进一步降低。我们预计，随着位宽增加，更多乘加运算将在数字信号处理器中实现，因为与查找表和触发器相比，数字信号处理器的实现效率更高。有趣的是，与其他资源相比，INT8 相较于 BL 的触发器数量仅略有减少，这主要归因于 INT16输入。简言之，较大的输入需要更多触发器来存储和累加计算，但位宽降低后，触发器的利用率会大幅下降。采用海森感知量化方案的多层感知机相比 BL，查找表减少 42.2%，触发器减少 36.3%，数字信号处理器减少 95.7%。随着精度降低，计算输出所需的查找表数量也会减少。许多低精度计算可通过查找表实现，因此查找表数量与位运算次数的相关性最强。然而，随着位宽增加并开始使用数字信号处理器，这种关系会减弱。查找表和触发器数量突然上升的异常值源于 Softmax 激活。如前所述，Softmax 激活会存储预计算输出；数量骤增是因为需要创建查找表，以存储位宽较大时的所有数值。表2 还列出了来自 AutoQkeras 的自动混合精度方案 QB [15], ，该量化神经网络通过最小化模型的比特大小进行优化。表2中用于喷注标记的 AutoQkeras 方案 QB 通过采用低于M

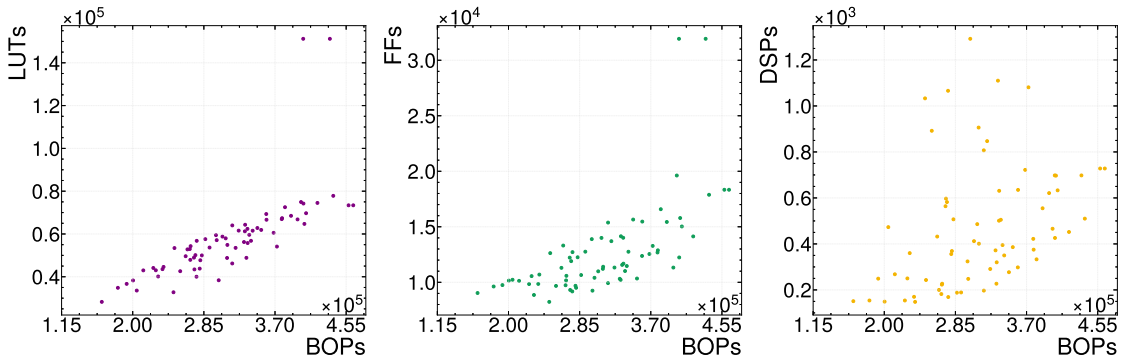


图 6. 暴力搜索最优混合精度量化方案时，部分 QNN 的资源使用情况。图中展示了 LUT、FF 和 DSP 使用量与位运算次数的关系，其中 LUT 与位运算次数呈现最强的线性关系。随着位宽增大，这种关系会减弱，因为 DSP 能够更高效地实现乘加运算。在所有设计中，全连接层仅使用 FF 作为存储器，而总使用量在位运算次数相近时也可能有很大差异，这表明两者之间的关系较弱。

4 位量化。其优势也体现在延迟方面，而准确率仅下降了可接受的 4%。本研究没有像 AutoQkeras 那样探索二值量化和三值量化，但显然，采用混合精度所带来的总体收益十分显著。

Vivado HLS 估算的这些模型的延迟也列于表 2中。延迟估算基于指定的时钟、循环变换分析结果以及设计的并行化程度。流水线和数据流的选择会显著影响实际吞吐量。然而，量化模型的延迟比 BL 长约 30 ns。这主要是因为低精度量化值所需的中间累加还要进行额外的缩放运算。虽然额外计算会增加延迟，但这些缩放层所需的资源相当有限，相对于设计的其余部分约为 1–3%。因此，低精度计算存在延迟与资源权衡。不过，对于当前任务而言，大幅减少资源所带来的收益足以抵消延迟的增加。Softmax 激活也是延迟的另一个重要来源，表2中所示的三个量化模型，其估计运行时间均为 10 ns。如上所述，块 RAM 用于存储预计算输出，延迟主要来自读取存储器。通常可以从实现的设计中移除 Softmax 激活函数，尤其是在后续计算只需要前k 个类别时。

7 总结

HAWQ 在边缘设备上的潜在应用及其自动位宽设置流程，使其成为物理学研究中颇具说服力的候选方案。本文通过扩展 HAWQ 库，使其能够将神经网络转换为 ONNX 和 QONNX 中间格式。将 HAWQ 与可读取这些格式的固件综合工具相衔接，可简化神经网络在边缘设备（如现场可编程门阵列或专用集成电路）上的部署，从而为科学研究开辟许多潜在应用场景。作为一项初步案例研究，我们使用神经网络对喷注进行分类，采用了喷注标记领域中常用于量化神经网络的一项具有挑战性的基准测试。我们展示了海森感知方案在混合精度量化方案中能够提供可靠的解决方案。随后，我们使用 HAWQ 中的新导出器，将多个采用不同位宽设置优化的多层感知机转换为 QONNX 中间表示。模型已成功从 HAWQ 转换为固件实现。稿件已提交至 ACM

模型	准确率 [%]	延迟 [ns]	资源			稀疏度 [%]	BOPs
			LUTs	FFs	DSPs		
基线	76.85	65	60,272	15,116	3,602	0	4,652,832
INT8	76.45	95	54,888	14,210	671	30	281,277
海森矩阵	75.78	90	34,842	9,622	154	33	182,260
QB	72.79	60	16,144	4,172	5	23	122,680

表2。报告了采用不同量化方案的喷注标记模型的资源使用情况。基线（不进行量化）以最多的资源实现了最高的准确率。采用相同位宽的 INT8 量化可减少数字信号处理器和查找表的使用量，而海森感知量化则显著降低了所有资源指标。文中还展示了来自 AutoQkeras、通过最小化总位数得到的混合精度模型 QB。这两种自动化方案都减少了计算所需的大量数字信号处理器和查找表，以及用于存储中间累加的触发器。海森矩阵方案基于 BOPs约束最低时的整数线性规划（ILP）解。

我们观察了资源使用情况相对于总位运算次数和准确率的表现。此外，我们将多种不同位宽设置下的资源利用率与参考文献中的自动位宽选择过程进行了比较 [18]；并将海森矩阵感知模型与采用统一位配置的模型及基线进行了比较。海森感知方案显著降低了所有资源指标（查找表、触发器和数字信号处理器），其中数字信号处理器和查找表的改进最为明显：与基线相比，分别减少了 95.7% 和 42.2%。尽管本研究目前仅涉及多层感知机，但所有神经网络架构都可以先导出为 ONNX 或 QONNX 中间表示图，再应用于支持该格式的工具。

致谢

JC、NT、AG、MWM 和 JD 获得美国能源部（DOE）科学办公室先进科学计算研究办公室“面向科学的极端边缘实时数据缩减协同设计”项目（DE-FOA-0002501）的支持。JM 获得费米研究联盟有限责任公司的支持，该公司依据与美国能源部（DOE）科学办公室高能物理办公室签订的合同（DE-AC02-07CH11359）开展工作。JD 还获得美国能源部（DOE）科学办公室高能物理办公室早期职业研究计划（Grant No. DE-SC0021187）的支持，以及美国国家科学基金会（NSF）“驾驭数据革命”（HDR）计划下“用于数据驱动发现的人工智能算法加速研究所”（A3D3）的支持（合作协议编号：OAC-2117997）。NT 还获得美国能源部（DOE）早期职业研究计划的支持（Award No. DE-0000247070），

参考文献

[1] 2022. 科学领域快速机器学习的应用与技术。《大数据前沿》 5 (2022年)。 https://arxiv.org/abs/2110.13041[2] Thea Aarrestad、Vladimir Loncar、Nicolò Ghielmetti、Maurizio Pierini、Sioni Summers、Jennifer Ngadiuba、Christoffer Petersson、Hampus Linander、Yutaro Iiyama、Giuseppe Di Guglielmo、Javier Duarte、Philip Harris、Dylan Rankin、Sergo Jindariani、Kevin Pedro、Nhan Tran、Mia Liu、Edward Kreinar、Zhenbin Wu 和 Duc Hoang。2021年。使用hls4ml在现场可编程门阵列上实现快速卷积神经网络。《机器学习：科学与技术》 2, 4（2021年7月），045015。[3] ATLAS合作组。2020年。Run 2中ATLAS触发系统的运行。《仪器学报》 15, 10（2020年10月），P10004。arXiv:2007.12539 [physics.ins-det][4] Junjie Bai、Fang Lu、Ke Zhang 等。2019年。ONNX：开放神经网络交换。 https://github.com/onnx/onnx[5] Colby Banbury、Vijay Janapa Reddi、Peter Torelli、Jeremy Holleman、Nat Jeffries、Csaba Kiraly、Pietro Montino、David Kanter、Sebastian Ahmed、Danilo Pau、Urmish Thakker、Antonio Torrini、Peter Warden、Jay Cordaro、Giuseppe Di Guglielmo、Javier Duarte、Stephen Gibellini、Videet Parekh、Honson Tran、Nhan Tran、Niu Wenxu 和 Xu Xuesong。2021年。MLPerf Tiny基准测试。载于 神经信息处理系统数据集与基准测试专题论文集，第1卷。[6] Chaim Baskin、Natan Liss、Eli Schwartz、Evgenii Zheltonozhskii、Raja Giryes、Alex M. Bronstein 和 Avi Mendelson。2021年。UNIQ：用于神经网络非均匀量化的均匀噪声注入。《*ACM*计算系统汇刊》 37, 1–4，第4篇文章（2021年3月），15页。

[7] Michaela Blott、Thomas B Preußer、Nicholas J Fraser、Giulio Gambardella、Kenneth O'Brien、Yaman Umuroglu、Miriam Leeser 和 Kees Vissers。2018年。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 11, 3（2018年），1。[8] Anja Butter 等。2019年。顶夸克标记器的机器学习概况。SciPost Physics 7（2019年），014。arXiv:1902。09914 [hep-ph][9] Yaohui Cai、Zhewei Yao、Zhen Dong、Amir Gholami、Michael W Mahoney 和 Kurt Keutzer。2020年。Zeroq：一种新颖的零样本量化框架。载于 IEEE/CVF 计算机视觉与模式识别会议论文集。13169。[10] Jianfei Chen、Yu Gai、Zhewei Yao、Michael W Mahoney 和 Joseph E. Gonzalez。2020年。深度神经网络低比特宽度训练的统计框架。载于神经信息处理系统进展年会论文集第33卷：2020年会议论文集。883–894。[11] Jianfei Chen、Lianmin Zheng、Zhewei Yao、Dequan Wang、Ion Stoica、Michael W. Mahoney 和 Joseph Gonzalez。2021年。ActNN：通过2比特激活压缩训练减少训练内存占用。载于第38届国际机器学习会议论文集。1803–1813。[12] François Chollet 等。2015年。Keras。https://keras.io。[13] CMS合作组。2020年。CMS一级触发器的第二阶段升级。CMS技术设计报告 CERN-LHCC-2020-004。CMS-TDR-021。[14] CMS合作组。2022年。用于在CMS第二阶段一级触发器中识别底夸克的基于神经网络的算法。技术报告 CMS-DP-2022-021。[15] Claudionor N. Coelho、Aki Kuusela、Hao Zhuang、Thea Aarrestad、Vladimir Loncar、Jennifer Ngadiuba、Maurizio Pierini 和 Sioni Summers。2021年。面向粒子探测器边缘端低延迟推理的深度神经网络自动异构量化。Nat. Mach. Intell. 3, 8（2021年），675。arXiv:2006。10159 [physics.ins-det][16] Evan Coleman、Marat Freytsis、Andreas Hinzmann、Meenakshi Narain、Jesse Thaler、Nhan Tran 和 Caterina Vernierie。2018年。量能测量对于高能增强喷注亚结构的重要性。《仪器学报》13, 01（2018年1月），T01003。[17] Zhen Dong、Yizhao Gao、Qijing Huang、John Wawrzynek、Hayden KH So 和 Kurt Keutzer。2021年。Hao：面向高效推理的硬件感知神经网络架构优化。载于2021年IEEE第29届现场可编程定制计算机国际研讨会（FCCM）。IEEE，50–59。[18] Zhen Dong、Zhewei Yao、Daiyaan Arfeen、Amir Gholami、Michael W. Mahoney 和 Kurt Keutzer。2020年。HAWQ-V2：神经网络的海森矩阵感知迹加权量化。载于神经信息处理系统进展，H. Larochelle、M. Ranzato、R. Hadsell、M.F. Balcan 和 H. Lin（编），第33卷。Curran Associates, Inc.，18518–18529。[19] Zhen Dong、Zhewei Yao、Amir Gholami、Michael W Mahoney 和 Kurt Keutzer。2019年。HAWQ：采用混合精度的神经网络海森矩阵感知量化。载于IEEE/CVF国际计算机视觉会议论文集。293–302。[20] Javier Duarte 等。2018年。用于粒子物理学的FPGA深度神经网络快速推理。《仪器学报》13（2018年7月27日），P07027。arXiv:1804。06913 [physics.ins-det][21] Javier Duarte、Nhan Tran、Ben Hawks、Christian Herwig、Jules Muhizi、Shvetank Prakash 和 Vijay Janapa Reddi。2022年。FastML科学基准测试：加速实时科学边缘机器学习。载于第5届机器学习与系统会议。arXiv:2207。07958 [cs.LG][22] Nicolò Ghielmetti 等。2022年。借助 hls4ml 在现场可编程门阵列上实现自动驾驶汽车的实时语义分割。Mach. Learn. Sci. Tech.（2022年）。arXiv:2205。07690 [cs.CV][23] Amir Gholami、Sehoon Kim、Zhen Dong、Zhewei Yao、Michael W Mahoney 和 Kurt Keutzer。2022年。高效神经网络推理的量化方法综述。载于《低功耗计算机视觉：提升人工智能效率》，G.K. Thiruvathukal、Y.-H. Lu、J. Kim、Y. Chen 和 B. Chen（编）。arXiv:2103.13630[24] 谷歌。2020年。QKeras。https://github.com/google/qkeras[25] Yijin Guan、Hao Liang、Ningyi Xu、Wenqiang Wang、Shaoshuai Shi、Xi Chen、Guangyu Sun、Wei Zhang 和 Jason Cong。2017年。FP-DNN：利用 RTL-HLS 混合模板将深度神经网络映射到FPGA的自动化框架。载于2017年IEEE现场可编程定制计算机年会研讨会论文集（FCCM）。152–159。[26] Benjamin Hawks、Javier Duarte、Nicholas J. Fraser、Alessandro Pappalardo、Nhan Tran 和 Yaman Umuroglu。2021年。Ps and Qs：面向高效低延迟神经网络推理的量化感知剪枝。Front. Artif. Intell. 4（2021年），676564。arXiv:2102。11289 [cs.LG][27] Kaiming He、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun。2016年。用于图像识别的深度残差学习。载于2016年IEEE计算机视觉与模式识别会议（CVPR）。770。arXiv:1512.03385[28] Qijing Huang、Dequan Wang、Zhen Dong、Yizhao Gao、Yaohui Cai、Tian Li、Bichen Wu、Kurt Keutzer 和 John Wawrzynek。2021年。Codenet：在嵌入式FPGA上高效部署输入自适应目标检测。载于2021年ACM/SIGDA现场可编程门阵列国际研讨会。206–216。[29] Sergey Ioffe 和 Christian Szegedy。2015年。批量归一化：通过减少内部协变量偏移加速深度网络训练。载于第32届国际机器学习会议论文集——第37卷（法国里尔）（ICML'15）。JMLR，448–456。[30] Alex Karbachevsky、Chaim Baskin、Evgenii Zheltonozhskii、Yevgeny Yermolin、Freddy Gabbay、Alex Bronstein 和 Avi Mendelson。2020年。HCM：神经网络架构的硬件感知复杂度指标。（2020年4月）。[31] Roman Kogler 等。2019年。大型强子对撞机上的喷注亚结构：实验综述。Rev. Mod. Phys. 91, 4（2019年），045003。arXiv:1803。06991 [hep-ex][32] Andrew J. Larkoski、Ian Moult 和 Benjamin Nachman。2020年。大型强子对撞机上的喷注亚结构：理论与机器学习近期进展综述。Phys. Rept. 841（2020年），1–63。arXiv:1709。04464 [hep-ph][33] Xiaoxuan Liu、Lianmin Zheng、Dequan Wang、Yukuo Cen、Weize Chen、Xu Han、Jianfei Chen、Zhiyuan Liu、Jie Tang、Joey Gonzalez、Michael Mahoney 和 Alvin Cheung。2022年。GACT：面向通用架构的激活压缩训练。载于第39届国际机器学习会议论文集。14139–14152。提交至 ACM 的稿件

[34] Zechun Liu、Kwang-Ting Cheng、Dong Huang、Eric Xing 和 Zhiqiang Shen。2022年。非均匀到均匀量化：通过广义直通估计实现准确量化。IEEE计算机视觉与模式识别会议论文集（CVPR）。[35] Maxim Naumov、Utku Diril、Jongsoo Park、Benjamin Ray、Jedrzej Jablonski 和 Andrew Tulloch。2018年。将周期函数作为神经网络量化的正则化项。（2018年11月）。[36] Jennifer Ngadiuba 等。2021年。利用高层次综合（HLS4ML）将现场可编程门阵列上的深度神经网络压缩至二值与三值精度。《机器学习：科学与技术》2（2021年），015001。arXiv:2003。06308 [cs.LG][37] Alessandro Pappalardo 等。2022年。QONNX：表示任意精度量化神经网络。arXiv:2206。07527 [cs.LG][38] Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga、Alban Desmaison、Andreas Kopf、Edward Yang、Zachary DeVito、Martin Raison、Alykhan Tejani、Sasank Chilamkurthy、Benoit Steiner、Lu Fang、Junjie Bai 和 Soumith Chintala。2019年。PyTorch：一款命令式风格的高性能深度学习库。载于《神经信息处理系统进展》，H. Wallach、H. Larochelle、A. Beygelzimer、F. d'Alché-Buc、E. Fox 和 R. Garnett（编）。第32卷。Curran Associates, Inc.，8024。arXiv:1912。01703 [cs.LG][39] Maurizio Pierini、Javier Mauricio Duarte、Nhan Tran 和 Marat Freytsis。2020年。hls4ml大型强子对撞机喷注数据集（30个粒子）。[40] Shibani Santurkar、Dimitris Tsipras、Andrew Ilyas 和 Aleksander Madry。2018年。批量归一化如何帮助优化？载于《神经信息处理系统进展》，S. Bengio、H. Wallach、H. Larochelle、K. Grauman、N. Cesa-Bianchi 和 R. Garnett（编），第31卷。Curran Associates, Inc. arXiv:1805.11604[41] Sheng Shen、Zhen Dong、Jiayu Ye、Linjian Ma、Zhewei Yao、Amir Gholami、Michael W Mahoney 和 Kurt Keutzer。2020年。Q-bert：基于海森矩阵的BERT超低精度量化。载于《AAAI人工智能会议论文集》，第34卷。8815–8821。[42] Albert M Sirunyan 等。2020年。使用机器学习技术识别重型、高能、强子衰变粒子。《仪器学报》15，06（2020年6月），P06005。[43] Albert M Sirunyan 等。2020年。CMS一级触发器在质子-质子碰撞中的性能，碰撞能量为 $\sqrt{s} = 13$TeV。《仪器学报》15，10（2020年10月），P10017。arXiv:2006。10165 [hep-ex][44] Yaman Umuroglu、Nicholas J. Fraser、Giulio Gambardella、Michaela Blott、Philip Leong、Magnus Jahre 和 Kees Vissers。2017年。FINN：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现场可编程门阵列国际研讨会论文集》（FPGA '17）。ACM，65。[45] Stylianos I. Venieris 和 Christos-Savvas Bouganis。2019年。fpgaConvNet：将规则与不规则卷积神经网络映射至现场可编程门阵列。《IEEE神经网络与学习系统汇刊》30，2（2019年），326–342。[46] Kuan Wang、Zhijian Liu、Yujun Lin、Ji Lin 和 Song Han。2019年。HAQ：硬件感知自动量化与混合精度。载于《2019年IEEE/CVF计算机视觉与模式识别会议》（CVPR）。8604。arXiv:1811.08886[47] Bichen Wu、Yanghan Wang、Peizhao Zhang、Yuandong Tian、Peter Vajda 和 Kurt Keutzer。2018年。通过可微神经架构搜索对卷积神经网络进行混合精度量化。arXiv:1812.00090[48] Zhewei Yao、Zhen Dong、Zhangcheng Zheng、Amir Gholami、Jiali Yu、Eric Tan、Leyuan Wang、Qijing Huang、Yida Wang、Michael W. Mahoney 和 Kurt Keutzer。2020年。HAWQV3：二进制定点神经网络量化。CoRR abs/2011.10680（2020年）。arXiv:2011.10680