arXiv:2303.02284v1 [eess.AS] 4 Mar 2023

用于端侧关键词检出的定点量化感知训练

SashankMacha⋆ Om Oza⋆ Alex Escott⋆ Francesco Caliv´a⋆Robbie Armitano⋆ Santosh Kumar Cheekatmalla⋆Sree Hari Krishnan Parthasarathi† Yuzong Liu⋆

⋆Alexa Perceptual Technologies, †Amazon 旗下 Alexa AI

摘要

定点（FXP）推理已被证明适用于计算资源有限的嵌入式设备，但模型训练仍持续采用浮点（FLP）方式进行。定点训练尚未得到充分探索，而从浮点转换到定点的非平凡过程会带来不可避免的性能下降。我们提出了一种训练并获得定点卷积关键词识别（KWS）模型的新方法。我们将所提出的方法与两种量化感知训练（QAT）技术相结合——用于模型参数的压缩权重分布和绝对余弦正则化——并提出了将量化感知训练扩展至瞬态变量的技术，而此前的范式通常忽略了这类变量。在 Google Speech Commands v2 数据集上的实验结果表明，我们可在不损失准确率的情况下，将模型精度降低至 4位。此外，在内部关键词检索数据集上，我们表明，相较于全精度浮点模型，我们的 8位 定点量化感知训练模型在固定误拒率下的相对假发现率提升了 4-6%。在推理过程中，我们认为定点量化感知训练消除了 q 格式归一化，并支持使用低位累加器，同时最大化单指令多数据吞吐量，从而降低用户感知延迟。我们证明，在不损害关键词唤醒模型的预测性能、也无需改变模型架构的情况下，可将执行时间缩短 68%。本工作提供了有助于该领域未来研究的新发现，并实现了准确且高效的模型。

索引词— 关键词检测、量化感知训练、定点算术、端侧

1. 引言

我们关注的是一种适用于嵌入式随身设备（OTG）的关键词检出（KWS）系统，该系统需占用较少内存且延迟较低。现代 KWS 算法采用深度神经网络（DNN），已展现出达到人类水平性能的能力 [1]。尽管取得了成功，深度神经网络部署在设备端仍面临挑战，因为其计算和带宽需求较高。将开销较大的浮点（FLP）运算转换为定点（FXP）运算，同时提升推理速度，可以大幅缓解这些挑战 [2]。与浮点运算相比，定点乘法消耗的能量少 18.5×，内存占用也只有一半 [3]。然而，由于商用 GPU 能够并行执行运算，且浮点具有较大的数值范围、有助于更好地更新梯度，因此深度神经网络通常采用浮点进行训练。虽然可在 FPGA [4], 上实现类似的定点并行化，但缺少可与 GPU 媲美的处理器，导致训练速度显著变慢。因此，大多数方法会将以浮点训练的模型转换为定点模型，但这种转换会引入偏差，降低性能 [5, 6]。我们提出了一种新方法，既能克服定点训练的局限，又能使用浮点工作流程以获得定点模型。

通过量化实现低精度推理，还可以进一步降低计算需求，因为这样每次访问一个字节的内存即可执行更多运算 [5, 7]。这种量化通常通过训练后量化（PTQ）实现 [8], ，但会造成严重的信息损失，影响模型准确率。为保持量化后深度神经网络的整体准确率，可以将量化纳入训练阶段，从而进行量化感知训练（QAT）。QAT 在训练过程中引入量化噪声，采用的技术包括确定性舍入 [9, 10, 11], 、重参数化 [12, 13]或正则化 [14, 15] 等。通过这些方法，深度神经网络能够适应推理阶段的量化。值得注意的是，已有研究表明，借助 QAT，模型参数可以采用二值和三值精度进行学习 [16, 17]。

本文讨论了实现定点训练并将其与量化感知训练相结合的方法。我们的方法支持训练精度低至超低精度（4位）的定点模型，这些模型具有较高准确率，在公开的 Google Speech Commands v2 数据集 [18] 上的表现可与全精度（ 32 位）浮点模型相当。在内部关键词识别数据集上，我们的 8 位定点量化感知训练模型相较于 32 位浮点基线的相对准确率提升了 4–6%。此外，我们提出了关键的推理优化，使受限的 OTG 设备能够实现关键词识别。我们提出的方法在保持与全精度浮点模型相近准确率的同时，使推理速度提升了 68%。此外，我们认为，通过量化感知训练，可以消除∼10%原本用于归一化激活精度的 CPU 周期。我们的贡献总结如下：

- 我们提出一种新颖的定点模型训练方法，利用针对浮点优化的工作流程；
- 我们将两种量化感知训练技术扩展到定点，并训练低比特关键词识别模型，同时将精度下降降至最低；
- 我们利用定点量化感知训练实现低比特累加器，从而缩短推理执行时间；
- 我们提出一种两级累加器缓冲区方法，以减少大型深度神经网络层中常见的溢出/饱和；

本文其余部分安排如下：第 2 节和第 3 节介绍相关工作，并重点介绍量化感知训练，聚焦于两种方法：基于压缩权重分布的方法和基于绝对余弦正则化的方法。第 4 节提出并阐述我们对这些方法的改进，使我们能够实现定点训练。第 5 节介绍实验设置和结果。经过讨论后，我们将结合对小型设备定点量化感知训练未来发展的展望，作为本文结论。

2. 相关工作

在旨在实现低功耗平台深度神经网络部署的各种推理设置中，定点系统发挥着关键作用 [19]。定点的

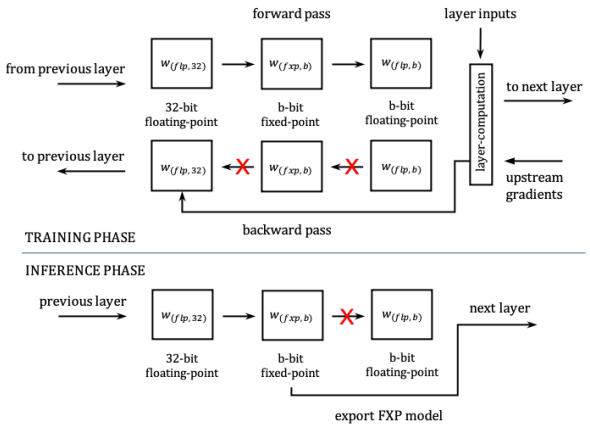

图 1。所提出的两阶段量化方法，在浮点训练框架中学习定点权重。

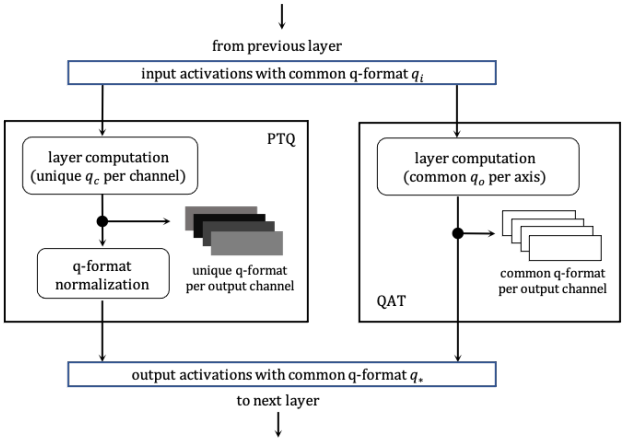

图 2。采用训练后量化时，两个连续层之间的激活归一化。量化感知训练通过限制不同 Q 格式的使用来消除归一化。

在不同数制中，定点在准确率与计算复杂度之间实现了最具前景的权衡 [20]。Horowitz 等人 [3] 在 45nm 工艺上对比定点乘累加（MAC）单元与浮点单元，也发现了佐证证据。定点推理领域已涌现出值得关注的研究——Umuroglu 等人 [7] 构建了一个将二值权重映射至硬件的框架，其他奠基性研究则提出了定点推理的设计方案 [21, 22]。另一方面，也有研究致力于高效的模型拓扑结构 [4], 、以较低误差将参数量化为定点的更优方法 [6] [23, , 24, 25] 以及解决定点训练局限性的方法。尽管大多数研究都集中于开发合适的模型 [26]或高效硬件，量化感知训练能够提升模型对定点的适应能力。Lin 等人 [27] 采用基于信号与量化噪声比的量化方法，以 8 位定点实现了更高的准确率，但推理期间出现了数据溢出。Anwar 等人 [28] 提出了基于 L2 误差最小化的全面量化方法，但需要先进行全精度预训练。在语音处理领域，已有研究将 4 至 6 位量化感知训练用于事件检测 [29], 、语音识别 [14] 和关键词识别 [9]，但仅限于浮点。上述方法要么需要对模型参数进行预训练，要么存在运行时限制，而且对推理计算的讨论也有限。

3. 量化感知训练

量化模型通常通过全精度训练后再进行量化获得。然而，对于激进压缩、低比特量化，或从浮点到定点的跨格式转换，这种量化会降低准确率。借助量化感知训练，我们在训练期间引入特定于推理的量化噪声，以避免推理时出错。我们的定点训练方法不依赖于特定的量化感知训练方法，并可扩展至不同技术。我们选取了该领域近期的两项工作，并介绍如何针对不同模型组件实现定点量化感知训练。

3.1. 权重量化

采用压缩权重分布（SQWD），权重可以重新参数化到有限范围（例如 [-1,1]）。这可以通过非线性函数（即双曲正切函数）实现，该函数将权重(w)映射为$(\hat{}w)$，使其服从均匀分布 [12]。正则化损失会将分布范围限制在双曲正切函数的渐近线处，以实现高效映射。随后，$\hat{}w$ 使用公式 1 量化为所需的 b 位精度，其中 $w_{(flp,b)}$ 表示精度为w 位的浮点权重b-。

$$w_{(flp,b)} = (\text{round}[2^{b-1}(\hat{w}+1) - 0.5] + 0.5) * 2^{b-1} - 1 \quad (1)$$

使用绝对余弦正则化（ACR），量化感知训练可定义为一个正则化问题。大多数量化感知训练技术不会在反向传播中纳入显式量化损失 [14]。相反，采用绝对余弦正则化时，会使用表示量化损失的绝对余弦函数对模型参数加权。对于 b 位量化，余弦函数有 2^b 个零点。为确保公平比较，我们将权重限制在与 SQWD 相同的有限范围内。

$$w_{(fxp,b)} = \text{round}[2^{b-1} * (w_{(flp,32)} + 1.0)] \quad (2)$$

$$w_{(flp,b)} = w_{(fxp,b)} * 2^{b-1} - 1.0 \quad (3)$$

采用 FXP 的方法 在 SQWD 或 ACR 中，均使用浮点量化器将权重(w) 从全精度浮点 ($w_{(flp,32)}$) 量化为 b 位浮点 ($w_{(flp,b)}$)。为了实现无量化误差的定点推理，一个直接的解决方案是将公式 1 中的浮点量化器替换为定点量化器。然而，GPU 对定点运算的支持有限，这使得大型机器学习模型的训练颇具挑战。为此，我们提出了一种两阶段量化器（如图 1 所示）：使用公式 2 将 w 从全精度浮点 ($w_{(flp,32)}$) 映射为 b-位定点 ($w_{(fxp,b)}$)，并使用公式 3 将其映射为 b 位浮点 ($w_{(flp,b)}$)。这样便可在浮点运算中进行训练，同时采用定点量化。在反向传播过程中，我们使用直通估计器 [30] 进行梯度计算，以克服 round [.] 运算的不可微性。推理时，我们使用公式 2 将参数导出为定点格式。通过在量化后的浮点表示与定点表示之间建立一一映射，我们可以消除训练后量化带来的误差。

3.2. 输入与激活量化

各种模型组件的输入构成了推理运算的其余部分。对于隐藏层激活，深度模型因大量乘加（MAC）运算而面临级联的位增长问题。为缓解这一问题，我们在连续层之间放置伪量化节点以降低精度，并在根据公式 2 和公式 3 进行量化之前，使用带线性缩放的截断 ReLU，将激活值压缩到较小的 $[-1, 1]$ 范围内。另一方面，为适应特征输入更宽的数值范围，我们牺牲了小数部分的位精度。用于表示小数部分的位数称为 q 格式。

公式 4 和 5 提出了一种针对特征 f 的略作修改的量化方法，其中 $c = 0$若为真则为 0.5，否则为 -0.5。

$$[h]f_{(fxp,b,q)} = \min[\max(f * 2^q + c, -2^{b-1}), 2^{b-1} - 1] \quad (4)$$

$$f_{(flp,b,q)} = f_{(fxp,b,q)} * 2^{-q} \quad (5)$$

4. 定点推理

本节介绍 QAT 如何通过消除训练后量化模型中常见的一项权衡来改进定点推理，并利用 SIMD 架构上的低比特推理实现更高程度的并行化。此外，我们还描述并提出了一种解决低比特累加器中典型饱和问题的方法。讨论内容针对 ARM NEON 芯片组展开，后者是定点推理的常用选择。

4.1. q 格式归一化

定点推理设置——专为训练后量化设计——为模型参数的每个轴分配不同的 q 格式 （q_c），以减少量化误差，从而导致模型内部存在混合表示。表示的数量随层数增加，并且需要额外的内存来存储 q 格式映射。为避免这一问题，激活值会在运行时归一化为统一的 q_*，然后再输入后续层。这些归一化操作大约占用推理周期的 10%。通过 QAT，我们施加训练时约束，使模型参数学习使用共同的 q 格式 （q），从而消除归一化所需的额外周期，加快推理速度。PTQ 与 QAT 之间的这一差异如图 2 所示。

4.2. 使用量化感知训练提升单指令多数据的并行度

ARMv7a/v8a 是采用定点运算的嵌入式设备中最广泛使用的架构。这些架构中的高级 SIMD 扩展（NEON）通过并行执行加载、存储、乘法和累加操作来加速处理。*vmlal s8*NEON 内在函数 [31] 在 128 位宽的 NEON 寄存器上运行，支持单个周期内最多执行八次操作。在推理过程中，乘加运算占计算量的 90%，我们利用 NEON 的并行能力并行执行乘加运算，从而加快推理速度。为充分利用 $8\times$ 并行能力，我们将乘加操作中的累加器位宽限制为低 16 位，因为如图 3 所示， $16\times8=128$。通过训练低比特精度模型，定点量化感知训练使低比特累加器得以应用。为避免位宽增长爆炸，16位累加器会在进入下一层之前进行缩放。

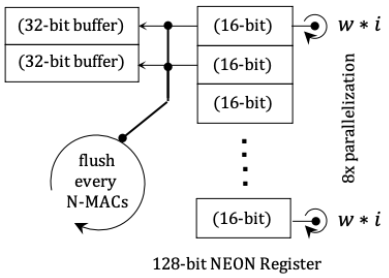

图 3。所提出的双层累加器缓冲机制，可防止低位累加器发生饱和。

4.3. 低位累加器中的饱和

虽然较小的累加器可以加快推理速度，但它们也更容易饱和。使用 8 位和 7 位模型时，16 位 NEON 累加器最快仅需 2 次乘加运算就会饱和，而 4 次乘加运算即可饱和。虽然我们可以通过量化感知训练训练模型，使其达到超低比特精度（即 4 位），但模型拓扑结构会影响饱和情况。具体而言，宽模型的每次激活涉及更多乘加运算，因此累加器更容易饱和。饱和是影响模型性能的一大问题，因为它会破坏后续的模型传播。对于我们关注的关键词检索问题，我们发现，0.1% 的饱和会导致关键词检测得分下降 ~30%。为此，我们提出一种两级累加器缓冲机制（2T-AB），在缓解这一问题的同时，保留 $8\times$ 单指令多数据并行性的优势。与使用单个累加器执行某次激活所需的全部乘加运算不同，我们提出定期将 16 位累加器刷新到更宽的 32 位定宽缓冲区中。表 1 展示了我们对 6 位关键词唤醒模型的分析结果，并表明 2T-AB 能够减少饱和次数。我们发现，稀疏刷新（即 每 256 次乘加运算刷新一次）可将饱和次数减少 89%，而每 64 次乘加运算刷新一次则可完全消除饱和。

表 1。采用和未采用 2T-AB 时，不同卷积层中因饱和而损坏的激活值数量。

卷积核大小	# 乘加运算每个激活值	# 激活值	# 被饱和破坏的激活值 在刷新频率下			
			None	256 次乘加运算	128 次乘加运算	64 次乘加运算
(3, 4, 32)	384	10,880	2,598	237	47	0
(4, 4, 32)	512	2,240	2,090	276	27	0
(7, 4, 40)	1,120	128	128	12	0	0
(1, 1, 128)	128	160	39	6	0	0
(1, 1, 160)	160	3	3	3	2	0
总计			4,858	534	76	0

5. 实验研究

我们使用面向资源受限设备的关键词唤醒模型，研究所提出的定点量化感知训练技术。我们将采用该方法训练的量化模型与未量化的全精度模型进行比较，从以下方面评估我们的方法：*i)* 准确率 *ii)* 延迟。

5.1. 数据集

我们使用内部关键词检索数据集 (D_1) 和 Google Speech Commands v2 数据集 (D_2) [18]开展实验。 D_1 包含来自远场和近场移动电话的 12.5K 小时去标识化标注音频，并划分为训练集（10K 小时）和测试集（2K 小时）； D_2 则包含 105K 段时长为一秒、以 16kHz 采样的音频片段，其中包含 35 种不同关键词的语音录音，这些录音采集自自然环境 [18]。我们对 D_2使用官方的训练集/验证集/测试集划分。

D_1 和 D_2 中的音频均降采样至 16kHz，转换为单声道，并使用 25ms 分析窗口和 10ms 步长进行分段。尽管音频文件的长度各不相同，我们仍使用长度为 76 的分段来生成固定长度的输入。随后，我们提取并拼接 64 维对数梅尔滤波器组能量（LFBE-64），构成输入卷积关键词检索模型的二维特征输入。

5.2. 模型配置

所有实验均采用二维全卷积模型。该模型包含五个连续的可训练卷积块，其前接一个不可训练的归一化层。网络中的卷积块在卷积核大小方面各不相同，但每个卷积层后都接批归一化，并以 ReLU 非线性作为激活。输出为一个 softmax 分类层，用于表示关键词检测的后验概率。该模型共有 199K 个可学习参数，浮点运算次数为 125K。

表 2。数据集 D_1 上不同 QAT 模型的相对假发现率（FDR）及其在 ARMCortexA53 上测得的相对执行时间（秒）。

模型	位宽	推理格式	相对虚警检测率	相对执行时间
基线	32 位	FLP	1.0	1.0
优化的训练后量化	16 位	FXP	1.109	0.45
SQWD（量化感知训练）	8 位	FXP	0.944	0.312
ACR（量化感知训练）	8 位	FXP	0.960	0.312

5.3. 训练配置

训练前，我们使用根据训练集计算得出的统计数据对数据进行标准化，并在评估时沿用这些统计数据。我们使用 TensorFlow（2.4）训练所有模型，共进行 1M 步，并使用 RAdam优化器 [32] 更新模型参数。前 10% 的总训练步数中，我们将学习率提高至 1e-3；此后将其线性衰减至 1e-5。我们采用交叉熵损失作为主要训练目标，并结合适用于不同量化感知训练范式的额外损失项。批量大小设为 1500，并使用 8 块 V100 GPU进行分布式训练。

5.4. 评估指标

我们使用端到端的检测错误权衡（DET）曲线来衡量模型准确率，该曲线描述模型的拒识率（FRR）与假发现率（FDR）之间的关系。与 [33] 类似，我们对 DET 曲线的坐标轴进行归一化，并在相对拒识率设为 1.0 时，将相对假发现率报告为相对于全精度浮点模型假发现率的数值。我们使用一颗基于 Cortex-A53 的单核 MT8163处理器来测量推理速度，该处理器专门针对 ARMv7-A 进行编译。我们使用一段长度为 90秒 、包含多个关键词实例的音频流，并在该处理器上持续运行我们的关键词唤醒模型，以测量执行时间。与假发现率类似，我们对执行时间进行归一化，并将其报告为相对于在非 SIMD 架构上运行的浮点模型的执行时间。

表 3。不同推理模型在 ARM Cortex-A53 上相对于不同刷新频率的执行时间（秒），用于研究 2T-AB。

模型	推理设置（格式、架构）	刷新节奏	程度并行化	相对执行时间（变化百分比）
基线	浮点；非 SIMD	None	1x	1.0（不适用）
优化的训练后量化	定点；单指令多数据	None	4x	0.450 (55%)
6 位量化感知训练 (SQWD)	定点；单指令多数据	None	8x	0.312 (68.8%)
		512 乘加运算		0.309（69.1%）
		256 乘加运算		0.309（69.1%）
		128 乘加运算		0.312（68.8%）
		64 次乘加运算		0.356 (64.4%)
		32 次乘加运算		0.363 (63.7%)
		16 次乘加运算		0.407 (59.3%)

5.5. 结果与讨论

我们使用所提出的 QAT 方法，以不同精度训练关键词识别模型，并避免通过改变模型拓扑结构来补偿低精度下的性能。我们首先以全精度浮点（基线）训练一个模型，并进行训练后量化，其中包括常见的 PTQ 优化，例如但不限于动态 q 因子选择和运行时归一化，以减少量化损失（优化的训练后量化）。我们还使用两种所提出的定点量化感知训练技术（即SQWD、ACR）以 8 位精度训练了两个模型。为执行定点推理，我们根据 eq2 将模型参数导出为定点格式，并使用针对 ARMv7-A 编译的 MT8163处理器。在表 2 中，我们报告了上述 4 个模型在数据集 D_1 上的相对虚警检测率。我们观察到，尽管进行了优化，训练后量化模型的性能仍不如基线；而使用定点量化感知训练训练的模型，即使精度只有基线的一半，表现也优于基线。

表 4。模型在数据集 D_2 上不同权重（w）和激活（a）精度（位数）下的准确率。表 a）采用基于 ACR 的量化感知训练，表 b）采用 SQWD。该模型在此数据集上的全精度浮点精度准确率为 90.7%。

w \ a	8	7	6	5	4
8	92.8	92.8	92.3	91.4	91.4
7	92.4	92.9	92.7	91.9	91.6
6	92.5	92.2	92.7	92.1	91.2
5	92.6	92.5	92.8	92.3	90.9
4	93.1	92.7	92.5	92.3	91.8

(a)

w \ a	8	7	6	5	4
8	91.6	91.4	91.4	91.1	90.4
7	92.0	91.2	91.5	91.1	90.9
6	91.3	91.6	91.7	91.3	91.3
5	91.3	91.5	92.3	91.3	91.9
4	91.5	91.7	91.6	91.6	91.3

(b)

我们将性能提升归因于量化感知训练引入的额外正则化。我们还报告了这四个模型的执行时间（浮点模型使用*x86* ，定点模型使用*ARM Cortex-A53*），并观察到，与浮点基线相比，16 位定点训练后量化模型和定点量化感知训练模型的速度分别快 55% 和 68%。此前，我们在表 1 中展示了所提出的 2T-AB机制如何减少饱和，但需要指出，偶尔的刷新会在推理过程中增加额外指令；不过，与我们实现的 $8\times$ 加速相比，这部分开销很小。从表 3 可以看出，当稀疏刷新频率为每 512、256 或 128 次乘加运算一次时，不会出现可观察到的减速；而每 16 次乘加运算就进行一次密集刷新，则会导致相对执行时间增加 10%。我们使用数据集 D_2 在表 4 中进行了一些消融研究，并展示了使用 SQWD 或 ACR 时，即使在低位（6 位）和超低位（4 位）精度下，也能达到与浮点精度相当的性能。我们还报告称，可以通过降低模型权重的精度来换取对激活值精度的提升，因为模型对激活值更为敏感。我们通过实验确认，能够成功训练出速度更快且准确率高于浮点模型的定点量化感知训练模型。

6. 结论与未来工作

本文提出了一种训练和开发定点模型的方法，该方法在训练期间使用浮点运算，同时实现一一映射，并消除训练后的转换误差。我们将此方法与两种不同的量化感知训练技术结合使用，以学习低精度模型参数，并提出了将量化感知训练应用于瞬态变量的方法。我们在关键词检测场景中测试了这一方法，该场景通常要求推理准确且延迟低。我们首先展示并证明，我们的 8 位定点量化感知训练模型优于对应的 32 位浮点模型。受这些有前景的结果鼓舞，我们进一步降低精度，并发现该方法具有很强的泛化能力，能够以极小的精度损失训练超低比特精度（4位）模型。除了准确之外，我们的模型效率也高于传统模型：与浮点推理相比，执行时间缩短了 68%；与先进行训练后权重量化、再进行定点推理的传统组合相比，执行时间缩短了 30%。总之，我们认为，在不牺牲准确率的情况下缩短设备端计算处理时间是可行的，这对于始终运行的关键词检测系统尤为重要。在下一阶段的工作中，我们计划探索该方法如何扩展到大型模型，以及计算机视觉等不同问题领域和语音识别等复杂任务。此外，我们还计划将本方法与剪枝、蒸馏和神经架构搜索等其他压缩技术结合，以生成高效模型。我们相信，这些发现将对专注于资源受限设备深度学习的研究群体产生广泛影响。

7. 参考文献

[1] Ian Goodfellow、Yoshua Bengio 和 Aaron Courville，深度学习，麻省理工学院出版社，2016年。[2] Alexandros Kouris、Stylianos I. Venieris 和 Christos-Savvas Bouganis，“级联卷积神经网络：突破卷积神经网络量化性能极限”，*2018*年第*28*届 *FPGA* 国际会议。[3] Mark Horowitz，“1.1 计算能耗问题（以及我们能做些什么）”，载于*2014*年*IEEE*国际固态电路会议论文集，2014年。[4] Chun Yan Lo 和 Chiu-Wing Sham，“卷积神经网络的高能效定点推理系统”，载于*2020*年*IEEE*第*63*届 *MWSCAS*。[5] Ritchie Zhao、Weinan Song、Wentao Zhang、Tianwei Xing、Jeng-Hau Lin、Mani Srivastava、Rajesh Gupta 和 Zhiru Zhang，“使用软件可编程 FPGA 加速二值化卷积神经网络”，2017年，美国计算机协会。[6] Norbert Mitschke、Michael Heizmann、Klaus-Henning Noffz 和 Ralf Wittmann，“一种基于权重缩放的卷积神经网络定点量化技术”，载于*2019*年*IEEE*国际图像处理会议。[7] Yaman Umuroglu、Nicholas J. Fraser、Giulio Gambardella、Michaela Blott、Philip Heng Wai Leong、Magnus Jahre 和 Kees A. Vissers，“FINN：一种快速、可扩展的二值化神经网络推理框架”，*2017*年 *ACM/SIGDA* 国际 *FPGA* 研讨会论文集。[8] Ron Banner、Yury Nahshan 和 Daniel Soudry，卷积网络训练后*4*位量化，实现快速部署，Curran Associates Inc.，2019年。[9] Yuriy Mishchenko、Yusuf Goren、Ming Sun、Chris Beauchene、Spyros Matsoukas、Oleg Rybakov 和 Shiv Naga Prasad Vitaladevuni，“面向小型关键词检测的低比特量化与量化感知训练”，*2019*年第*18*届*IEEE*机器学习与应用国际会议。[10] Benoit Jacob、Skirmantas Kligys、Bo Chen、Menglong Zhu、Matthew Tang、Andrew G. Howard、Hartwig Adam 和 Dmitry Kalenichenko，“神经网络的量化与训练：实现高效的纯整数运算推理”，*2018*年*IEEE/CVF*计算机视觉与模式识别会议。[11] Hao Li、Soham De、Zheng Xu、Christoph Studer、Hanan Samet 和 Tom Goldstein，“训练量化网络：深入理解”，载于神经信息处理系统大会，2017年。[12] Nikko Strom、Haidar Khan 和 Wael Hamza，“用于深度模型低比特量化的压缩权重分布”，国际语音通信协会会议论文集，2022年。[13] Robin Dupont、Hichem Sahbi 和 Guillaume Michel，“面向预算的网络剪枝中的权重重新参数化”，*2021*年*IEEE*国际图像处理会议。[14] Hieu Duy Nguyen、Anastasios Alexandridis 和 Athanasios Mouchtaris，“用于自动语音识别的量化感知训练与绝对余弦正则化”，载于国际语音通信协会会议，2020年。[15] Ruizhou Ding、Ting-Wu Chin、Zeye Dexter Liu 和 Diana Marculescu，“通过正则化激活分布训练二值化深度网络”，*2019*年*IEEE*计算机视觉与模式识别会议。[16] Matthieu Courbariaux、Yoshua Bengio 和 Jean-Pierre David，“BinaryConnect：在传播过程中使用二值权重训练深度神经网络”，神经信息处理系统大会进展，2015年。[17] Wei Zhang、Lu Hou、Yichun Yin、Lifeng Shang、Xiao Chen、Xin Jiang 和 Qun Liu，《Ternarybert：面向蒸馏的超低比特 bert》，发表于*EMNLP*，2020年。[18]Pete Warden，《Speech commands：一个用于有限词汇语音识别的数据集》，*arXiv* 预印本 *arXiv:1804.03209*，2018年。[19] Vasanthakumar Rajagopal、Chandra Kumar Ramasamy、Ashok Vishnoi、Raj Narayana Gadde、Narasinga Rao Miniskar 和 Sirish Kumar Pasupuleti，《深度神经网络的精确高效定点推理》，发表于*2018*年第*25*届 *IEEE ICIP*。[20] Patrick Judd、Jorge Albericio、Tayler Hetherington、Tor M. Aamodt、Natalie Enright Jerger 和 Andreas Moshovos，《Proteus：利用深度神经网络中的数值精度可变性》，2016年。[21] Chen Zhang、Peng Li、Guangyu Sun、Yijin Guan、Bingjun Xiao 和 Jason Cong，《优化面向深度卷积神经网络的 FPGA 加速器设计》，2015年。[22] Ritchie Zhao、Weinan Song、Wentao Zhang、Tianwei Xing、Jeng-Hau Lin、Mani B. Srivastava、Rajesh K. Gupta 和 Zhiru Zhang，《利用软件可编程 FPGA 加速二值化卷积神经网络》，*2017*年 *ACM/SIGDA* 国际 *FPGA* 研讨会论文集。[23] Sujeong Jo、Hanmin Park、Gunhee Lee 和 Kiyoung Choi，《使用低精度动态定点训练神经网络》，发表于*2018*年第*36*届 *IEEE ICCD*。[24] Yasufumi Sakai，《使用 8 位动态定点进行深度神经网络训练的量化》，发表于*2020*年第*7*届 *ISCMI*。[25] Donghyeon Han、Dongseok Im、Gwangtae Park、Youngwoo Kim、Seokchan Song、Juhyoung Lee 和 Hoi-Jun Yoo，《HNPU：采用随机动态定点和主动位精度搜索的自适应 DNN 训练处理器》，*IEEE* 固态电路杂志，2021年。[26] Ming Sun、Varun Nagaraja、Bj¨orn Hoffmeister 和 Shiv Vitaladevuni，《面向嵌入式关键词检出的模型缩减》，发表于*2015*年第*14*届 *IEEE ICMLA*。[27] Darryl Dexu Lin、Sachin S. Talathi 和 V. Sreekanth Annapureddy，《深度卷积网络的定点量化》，发表于*ICML*，2016年。[28] Sajid Anwar、Kyuyeon Hwang 和 Wonyong Sung，《面向目标识别的深度卷积神经网络定点优化》，发表于*2015*年 *IEEE ICASSP*，2015年。[29] Bowen Shi、Ming Sun、Chieh-Chi Kao、Viktor Rozgic、Spyridon Matsoukas 和 Chao Ching Wang，《通过低秩矩阵分解和量化训练压缩声学事件检测模型》，*ArXiv*，2019年。[30] Yoshua Bengio、Nicholas L´eonard 和 Aaron Courville，《为条件计算估计或传播随机神经元的梯度》，*arXiv* 预印本 *arXiv:1308 3432*，2013.。[31]《Neon 程序员指南：向量乘累加》。.[32] Liyuan Liu、Haoming Jiang、Pengcheng He、Weizhu Chen、Xiaodong Liu、Jianfeng Gao 和 Jiawei Han，《自适应学习率的方差及其他问题》，*ArXiv*，2020年。[33] Lu Zeng、Sree Hari Krishnan Parthasarathi、Yuzong Liu、Alex Escott、Santosh Kumar Cheekatmalla、Nikko Strom 和 Shiv Naga Prasad Vitaladevuni，《面向嵌入式芯片组的流式关键词检出模型亚 8 位量化》，发表于*TSD*，2022年。