

# 赛灵思器件上的 INT4 优化卷积神经网络

对于 AI 推断，在提供与浮点相媲美的精度的同时，INT8 的性能优于浮点。然而在资源有限的前提下，INT8 却不能满足性能要求，INT4 优化则是解决之道。通过 INT4 优化，与现有的 INT8 解决方案相比，赛灵思在实际硬件上可实现高达 77% 的性能提升。

## 概要

赛灵思在其硬件平台上提供 INT8 AI 推断加速器 — 深度学习处理器单元 (XDPU)。然而，在某些资源受限，要求高性能、低时延的场景（例如对资源、功耗敏感的边缘侧场景和低时延 ADAS 场景）中，为了实现比 INT8 更低的功耗和更高的性能，需要对神经网络进行低比特量化。然而，极低比特量化（如二进制或三进制）却会导致精度劣化。

因此，4 位激活参数和 4 位权重参数 (4A4W) 全流程硬件友好型量化解决方案可实现更优异的精度 / 资源权衡取舍。本白皮书介绍了在 Zynq® UltraScale+™ MPSoC 和 Zynq-7000 SoC 系列 (16nm 和 28nm) 上面面向 CNN 4 位 XDPU 实现的低精度加速器。这种加速器通过高效地映射卷积计算，充分发挥其 DSP 功能。这种解决方案可提供优于 XDPU 两倍的解决方案级性能。在 ADAS 系统中执行 2D 检测任务时，这种实现方案能够在 Zynq UltraScale+ MPSoC ZCU102 板上实现 230fps 的推断速度，与 8 位 XDPU 相比性能提高 1.52 倍。此外，在用于 ADAS 系统中的不同任务时，该解决方案可实现媲美全精度模型的结果。

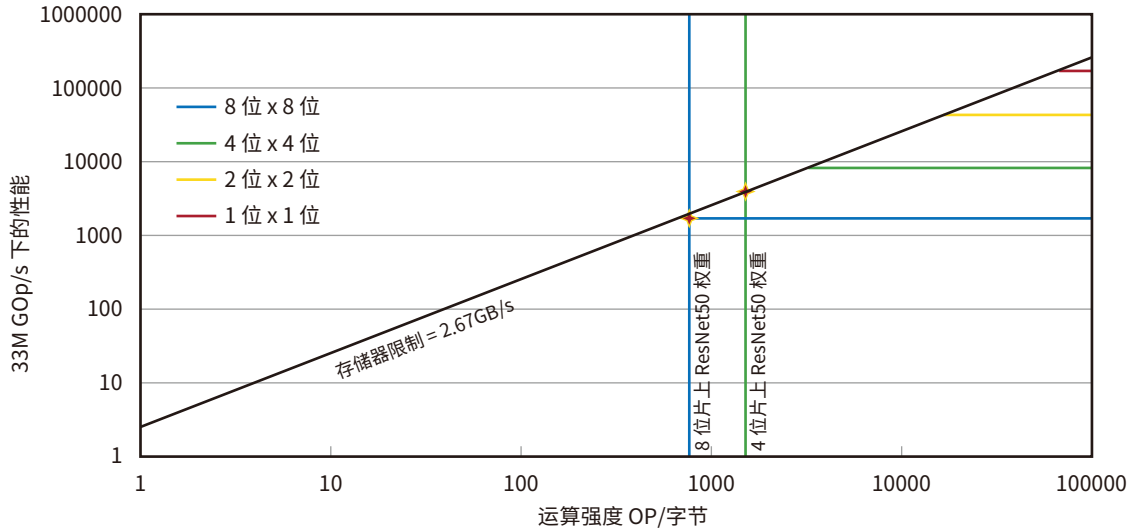
# 介绍

企业日益重视基于 AI 的系统在数据中心、汽车、工业和医疗等领域中的产品化。这带来了两大挑战：

- AI 推断需要完成的计算量成数量级增加，同时还要保持价格、功耗、时延和尺寸大小不变。
- AI 科学家继续日复一日地在算法和模型上开展创新，需要各种不同的硬件架构提供最佳性能。

对持续创新的强烈需求则需要使用灵活应变的领域专用架构 (DSA)。优化 AI 推断性能和降低功耗的主要趋势之一是使用较低精度和混合精度。为降低硬件设计复杂性，模型量化被当作关键技术应用于各类硬件平台。大量工作被投入用于最大限度地降低 CNN 运算量和存储成本。这项研究充分地证明，对于大多数计算机视觉任务，在不严重牺牲精度的情况下，权重参数和激活参数可以用 INT8 表达。然而对于某些边缘应用而言，硬件资源仍然不足。在对边缘应用使用较低的位宽（如 1 位、2 位）时，一些常见的硬件设计解决方案使用简化的乘法器。尽管这些解决方案时延低、吞吐量大，但它们与全精度模型相比，仍然存在较大的精度差距。因此，在模型精度和硬件性能之间寻求平衡变得至关重要。

赛灵思运用几种常见的网络结构 ((ResNet50V1[参考资料 2]、ResNet50V2[参考资料 3]、MobilenetV1[参考资料 4] 和 MobilenetV2[参考资料 5])，在 ImageNet 分类 [参考资料 1] 任务上通过使用几种不同的量化算法进行了实验。结果显示精度随着位宽减少而下降。尤其是在位宽低于 4 时精度下降显著。此外，赛灵思也使用 Williams 等介绍的 Roofline 模型 [参考资料 6]，分析不同位宽下的硬件性能，如图 1 所示。以赛灵思 ZCU102 评估板为例，随着 MAC 的精度降低，硬件成本降低，性能得到提高。此外，实验结果还显示，低比特量化可通过降低存储器需求提高性能。这在 ResNet-50 神经网络的卷积运算强度上得到证实。该网络分别用 8 位精度和 4 位精度进行了运算。因此，INT4 在模型精度和硬件性能之间实现了最佳权衡。



WP521\_01\_042820

图 1: 在 ZCU102 上以不同位宽运行 Roofline 模型

## 如何量化全流程硬件友好型 CNN

为实现整个量化过程的硬件友好化，INT4 量化法可分为三个部分：量化机制、硬件友好型量化设计、量化感知训练。

### 量化机制

赛灵思使用经训练的量化阈值 (TQT) [ 参考资料 7] 将 DNN 从单精度浮点 (FP32) 转换为 INT4。对于权重和激活参数，量化函数可正式地写作：

$$q(x; s) := clip \left( \left\lfloor \frac{x}{s} \right\rfloor; n, p \right) \cdot s$$

方程 1

其中  $n = -2^{b-1}$ ,  $p = 2^{b-1} - 1$  和  $s = \frac{2^{\lceil \log_2 t \rceil}}{2^{b-1}}$  为有符号数据； $n = 0$ ,  $p = 2^b - 1$  和  $s = \frac{2^{\lceil \log_2 t \rceil}}{2^b}$  为

无符号数据。

方程 1 体现出输入值  $x$  的量化值取决于阈值  $t$ 、位宽  $b$  和量化比例系数  $s$ 。阈值  $t$  一般初始化为待量化的张量的绝对值的最大值。随后在训练过程中用  $\log_2 t$  的形式进行优化。量化系数  $s$  是 2 的幂，具有硬件友好性。上下截断运算去除部分离群数据，加大权重参数和激活参数的分布紧密度，更有利于量化。

如上文所述,  $\log_2 t$  是一种在训练过程中可学习的参数。优化它就能确定合适的量化范围。与之相反,  $\log_2 t$  的梯度可通过链式法则确定。此外, 输入值  $x$  的梯度也可通过下式计算:

$$\nabla_{(\log_2 t)} q(x; s) := s \ln 2 \cdot \begin{cases} \left\lfloor \frac{x}{s} \right\rfloor - \frac{x}{s} & \text{if } n \leq \left\lfloor \frac{x}{s} \right\rfloor \leq p \\ n & \text{if } \left\lfloor \frac{x}{s} \right\rfloor < n \\ p & \text{if } \left\lfloor \frac{x}{s} \right\rfloor > p \end{cases}$$

方程 2

$$\nabla_x q(x; s) := \begin{cases} 1 & \text{if } n \leq \left\lfloor \frac{x}{s} \right\rfloor \leq p \\ 0 & \text{otherwise} \end{cases}$$

方程 3

对于  $\lfloor x \rfloor$  (四舍五入) 和  $\lceil x \rceil$  (正无穷取整), 不可微函数 STE 被用于确定梯度, 定义见[方程 4](#)。

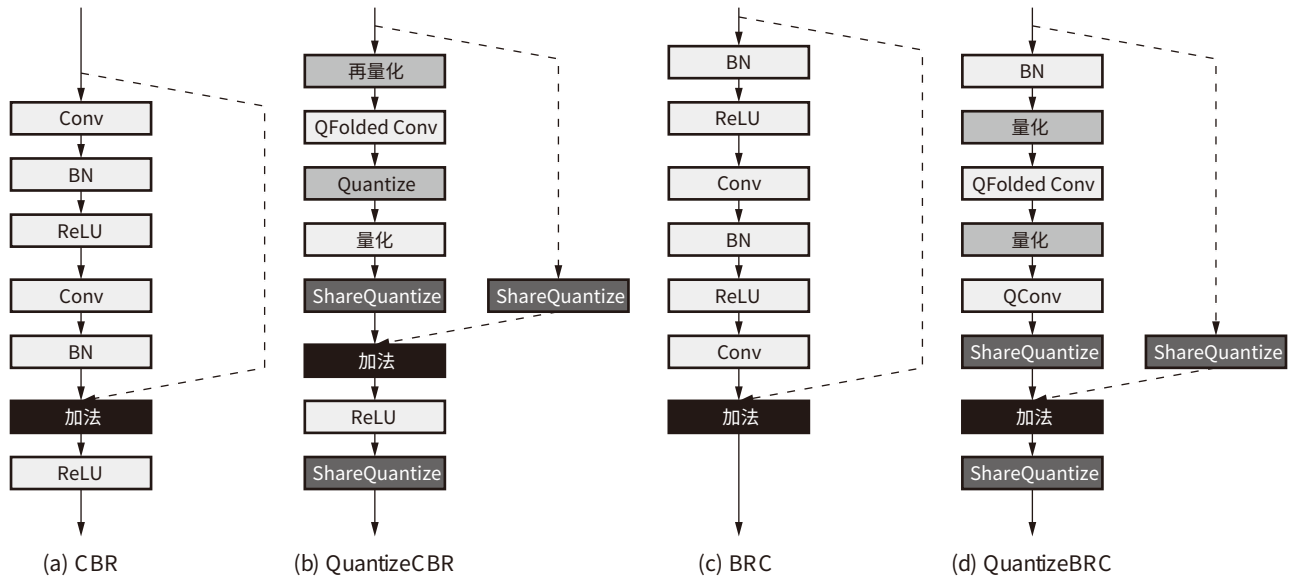
$$\frac{\partial}{\partial x} \lfloor x \rfloor = \frac{\partial}{\partial x} \lceil x \rceil = 1$$

方程 4

TQT 证明对数表达能确保阈值和输入值的标度不变性。采用对数方式训练阈值更容易管理, 并且被证明是非常高效的。

## 硬件友好型量化设计

要进行量化训练, 必须从全精度网络中构建低比特网络。以全流程硬件友好型量化为基础, 下文介绍部分常用网络结构并对几种粗粒度模块的量化解决方案进行总结。对于这些量化模块, INT4 量化方法可用于多种类型的网络结构。部分常用模块的量化解决方案如[图 2](#)所示。[图 2](#)中的虚线表明能根据实际网络结构增添或者删除。



WP521\_02\_040320

图 2：模块量化

### 模块 1：CBR(Conv+BN+ReLU)

作为 CNN 中的通用结构，BN 层被合并以减少训练和推断过程中的触发器数。然而，BN 层存在不一致性；批量运算在训练过程中使用当前批次的平均值和方差，但在推断过程中移动平均值和方差。如果量化的是从当前批次的平均值和方差获得的合并参数，在推断时就会导致偏差。为消除这种不匹配，应使用下列最佳实践 [参考资料 8]、[参考资料 9] 来量化这种结构。在将 BN 折叠到 Conv 后，就对折叠到 INT4 的参数进行量化。该模块的量化如图 2 (b) 所示。

### 模块 2：BRC(BN+ReLU+Conv)

如图 2 (c) 所示，在紧随卷积层的 BN 层被合并后，仍然存在独立的 BN 层。然而，在现有的 INT4 量化方法中，BN 层基本不受重视。为有效地部署这个独立的 BN 层，一种量化神经网络 [参考资料 10] 的简化方法被用于在训练过程中保持全精度，并在推断过程中吸收浮点标度和阈值偏差。这种方法可延伸用于所有线性运算（包括推断中的卷积），同时有助于保持精度。该模块的量化详见图 2 (d)。

### 模块 3：加法

占用硬件资源较少。因此该层一般量化为 8 位。此外，为了量化所有输入和输出，还将使用标度共享规则。共享规则的作用是让硬件绕过标度计算，消除了浮点乘法的需要。如图 2 (b) 所示，“ShareQuantize”指这些量化层共享相同标度。

**其他：**

为确保卷积运算的输入是 4 位，加法运算的 8 位输出需要再次被量化为 4 位，如图 2 中的“再量化”所示。对于第一层和最后一层，仍然进行 INT4 量化。整个网络的输出被量化成 8 位。内积层与卷积层保持一致。

**量化感知训练**

量化感知训练通常被用作一项关键技术，用来降低低比特模型与全精度模型之间的精度差。在本白皮书描述的 INT4 量化方法中，它仍起着不可或缺的作用。量化感知训练过程都使用算法 1（如下所示）。

**算法 1：逐层量化感知训练****输入：**

全精度输入、权重和偏差：X、W、Bias

针对输入和权重的可学习对数域阈值： $a_x$ 、 $a_w$ 、 $a_{bias}$

位宽：针对输入和权重， $b=4$ ；针对偏差， $b=8$

**输出：**

输出：Y

1. 初始化  $a_x = \log_2 \max(|x|)$ ， $a_w = \log_2 \max(|w|)$ ， $a_{bias} = \log_2 \max(|bias|)$
2. 根据方程 1 计算  $q(x)$ 、 $q(w)$  和  $q(bias)$
3.  $Y = \text{Forward}(q(x), q(w), q(bias))$
4. 计算分类损耗：Loss。对所有可学习参数使用正则化方法。
5. 参阅方程 3

$$\nabla_X L = \frac{\partial L}{\partial q(x)} * \frac{\partial q(x)}{\partial X}, \frac{\partial q(x)}{\partial X}$$

6. 使用 Adam 更新全精度参数

# 赛灵思 DSP 片上的 INT4 优化

使用 DSP 硬件资源可实现乘法和累加 (MAC) 占用硬件资源较少。经优化后, DSP 能够在 16nm 或 28nm 器件上处理尽可能多的 MAC 运算。以 16nm 为例, 赛灵思可编程器件中 UltraScale™ 架构的 DSP48E2 片就属于专用片 [ 参考资料 11]。DSP48E2 片由一个 27x18 二进制补码乘法器和一个 48 位累加器构成。如图 3 所示, MAC 能使用赛灵思 DSP 片完成。

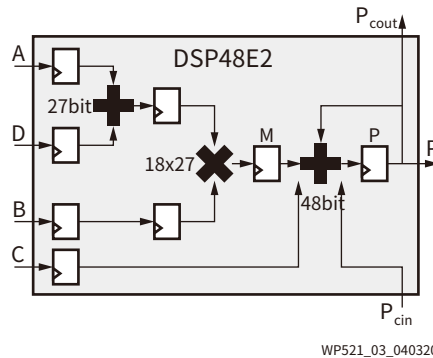


图 3: MAC 模式下的 DSP48E2 片

## INT4 优化

在低精度 MAC 运算中, 相乘方式是  $a \cdot b$ 。其中  $a$  是 4 位无符号激活参数数据,  $b$  是 4 位有符号权重参数数据。DSP48E2 片可被配置成 4 通道乘法运算, 如 HYPERLINK \l "\_bookmark6" 图 4 所示。

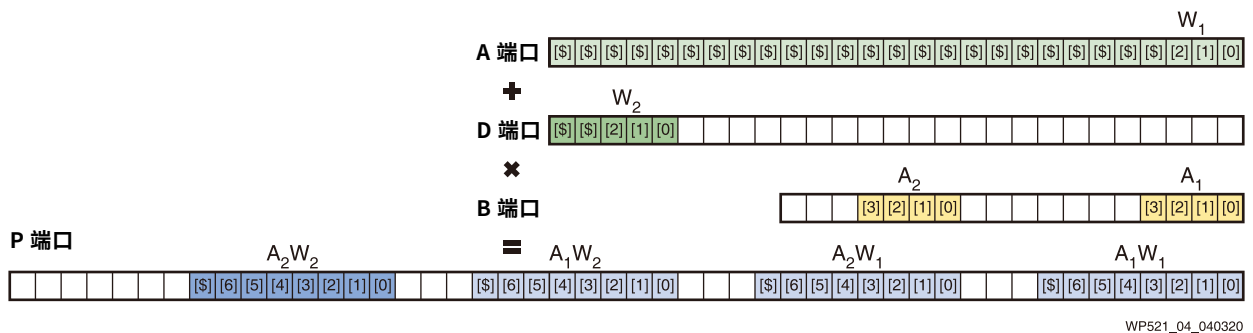


图 4: 4 通道封装的 DSP48E2 配置模式

DSP48E2 片的端口 A 是 27 位宽。端口 B 是 18 位宽。int4 \* uint4 相乘产生的结果至少有 8 位宽。充分利用 DSP 资源的前提是在多个相乘被打包在一起时, 确保输出结果保持正确。为确保这一点, 通道之间添加了保护位。当四个 MAC 通道被打包在一起时, 需要在两路输入间布置足够的保护位。根据 DSP48E2 片的设计, 保护位被设置为 3 位:

$$(A_2 \cdot 2^{11} + A_1) \cdot (W_2 \cdot 2^{22} + W_1) = A_2W_2 \cdot 2^{33} + A_1W_2 \cdot 2^{22} + A_2W_1 \cdot 2^{11} + A_1W_1$$

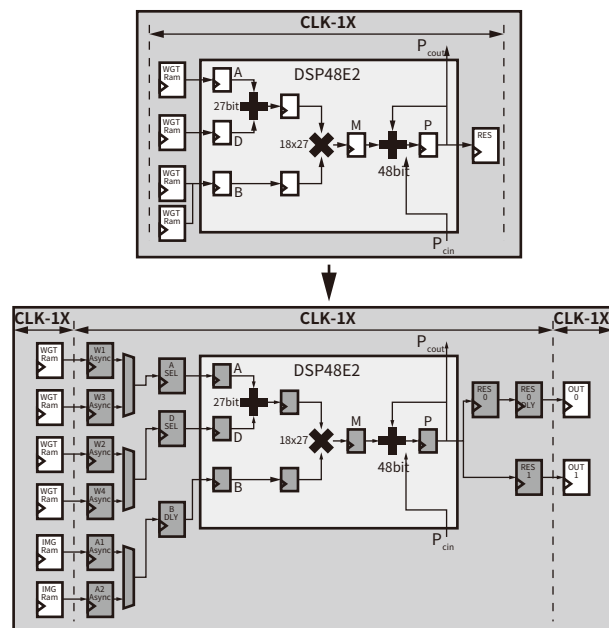
方程 5

第一个通道  $A_1 \cdot W_1$  被布置在对应端口的 4LSB 上。下一个通道  $A_2 \cdot W_1$  需要移位至少 8 位才能正确计算。第二个通道与第一个通道共享权重参数数据  $W_1$ 。端口 B 中的  $A_2$  移位 11 位。3 位保护位用于最大化 DSP 资源的利用。最后一个计算元  $W_2$  被分配给端口 A。最后两个通道是  $A_1 \cdot W_2$  和  $A_2 \cdot W_2$ 。权重参数是有符号数据。在相乘开始前，使用 27 位预加法器打包两个权重参数数据。因为  $W_1$  需要符号扩展，所以  $W_2$  不能布置在 D 端口的四个 MSB 上 [参考资料 12]。如果  $W_2$  在 MSB 中，当  $W_1 < 0$  且  $W_2 = -8$  时，预加法器就会溢出。后 48 位加法器可用作累加器，通过级联对之前层次的 DSP 结果进行相加。单个 DSP48E2 就能在单个时钟周期内实现四通道 MAC。

结果的位宽在累加后增大。硬件友好型量化器是一组移位寄存器，它可以通过指令控制移位的位数。移位运算是硬件友好型的。在低精度 CNN 中，卷积能够使用两种量化方法之一。一种是逐计算元地输出 8 位。另一种是对下一卷积输出 4 位。通过算法优化，两种量化方法都能量化成 2k 步长。差别在于输出数据的位宽以及它们是否有符号数据。

## DSP 强化使用

DSP 双数据速率 (DDR) 技术被用于改进由 DSP48 片实现的性能 [参考资料 13]。因此需要为 DPU 提供两个输入时钟：一个用于通用逻辑，另一个用于 DSP 片。未采用 DSP DDR 技术的 DPU 和采用强化使用模式的 DPU 之间的差异如图 5 所示。



WP521\_05\_040320

图 5: 未采用 DDR 的 DSP 和 DSP 强化使用之间的差异

## 面向 CNN 要求的计算图

卷积是 CNN 网络的主要计算要求。卷积的实际计算任务如下：

$$X_f = \sum_{n=0}^N A_{nf} \cdot W_{nf} + Bias_f$$

方程 6

其中  $A_{nf}$  是浮点特征图， $W_{nf}$  是浮点权重。其本质是 MAC 运算。根据赛灵思的新颖量化感知训练解决方案，浮点的卷积计算按如下方式进行量化：

$$X_f = \sum_{n=0}^N \alpha_{xf} A_{int} \cdot \alpha_{wf} W_{int} + \alpha_{bf} Bias_{int}$$

方程 7

$$X_{fixed} = 2^k \left( \sum_{n=0}^N A_{int} \cdot W_{int} + 2^{j-k} Bias_{int} \right)$$

方程 8

其中  $\alpha_{xf}$ 、 $\alpha_{wf}$  和  $\alpha_{bf}$  是标度。这些浮点参数被转换成  $2^k \cdot 2^k$ 。这是一种硬件友好型标度，能够在 FPGA 中使用移位运算轻松实现。

DSP 块在一个时钟周期中需要两个权重和两个特征。其中的每一个都能共享，如图 6 所示。

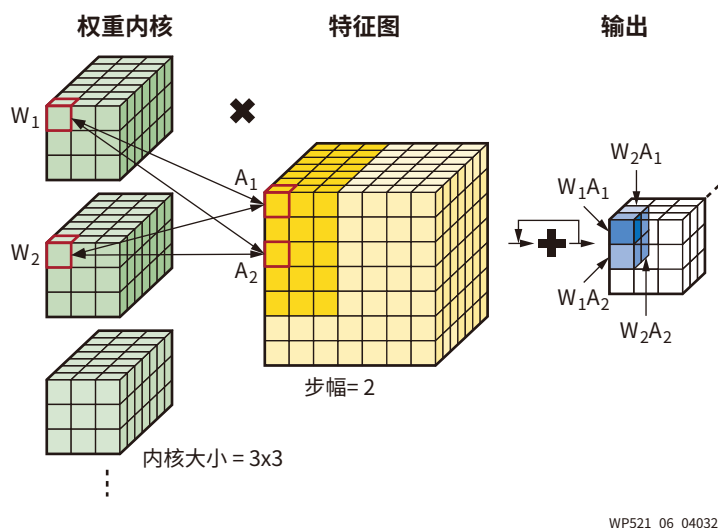


图 6: 卷积计算任务和乘法器共享方式

其中  $A_{nf}$  是浮点特征图， $W_{nf}$  是浮点权重。其本质是 MAC 运算。根据赛灵思的新颖量化感知训练解决方案，浮点的卷积计算按如下方式进行量化：

## 模型量化与性能仿真

下面的章节讲解量化感知训练中使用的 CV 任务。这些任务包括图像分类、姿态估计、2D 检测、3D 检测、语义分割和多任务。

### 基准分类模型

在完成 ImageNet 分类数据集上的实验后得到如下结果。网络包括 ResNet50-V1、ResNet50-V2。在所有实验中，数据集均从浮点模型进行微调。所有偏差参数都量化到 8 位。实验结果如表 1 所列。

表 1: 不同位宽下类 ResNet50 的网络精度

模型	A/W	输入/输出	第一层/最后一层	逐计算元	Top 1	Top 5
ResNet50V1	浮点	浮点	浮点	浮点	76.15	92.87
	8/8	8/8	8/8	8	76.024	92.940
	4/4	4/8	4/4	8	74.588	91.998
ResNet50V2	浮点	浮点	浮点	浮点	77.596	93.53
	8/8	8/8	8/8	8	77.474	93.536
	4/4	4/8	4/4	8	74.124	91.422

基准分类模型的结果参见表 1。它体现了这种方法的有效性。具体对 ResNet50V1 而言，4 位 XDPU 解决方案与 8 位 XDPU 解决方案在前 1 精度上的差距仅有 1.4%，在前 5 精度上的差距仅有 0.9%。

## 实时 ADAS 模型，包括姿态估计、检测、分割、多任务等。

为进一步验证量化方法的通用性，也在真实场景下开展了其他 CV 任务。

### 姿态估计

姿态估计任务使用更加复杂的堆叠 Hourglass 网络 [参考资料 14]。通过在 MPII [参考资料 15] 数据集上开展姿态估计实验，评估了逐层模式下两个网络结构的精度。结果参见表 2。

表 2: 不同位宽下的 Hourglass 网络精度

模型	A/W	输入/输出	第一层/最后一层	逐计算元	精度 (%)
hg-s2-b1	浮点	浮点	浮点	浮点	<b>71.29</b>
	8/8	8/8	8/8	8	72.04
	4/4	4/8	4/4	8	69.67
hg-s8-b1	浮点	浮点	浮点	浮点	<b>83.46</b>
	8/8	8/8	8/8	8	83.06
	4/4	4/8	4/4	8	82.51

在表 2 中，hg-s2-b1 意味着堆栈数量是 2，块数量是 1。Hg-s8-b1 意味着堆栈数量是 8，块数量是 1。表 2 证明赛灵思 INT4 量化解决方案实现了可媲美浮点模型的精度。

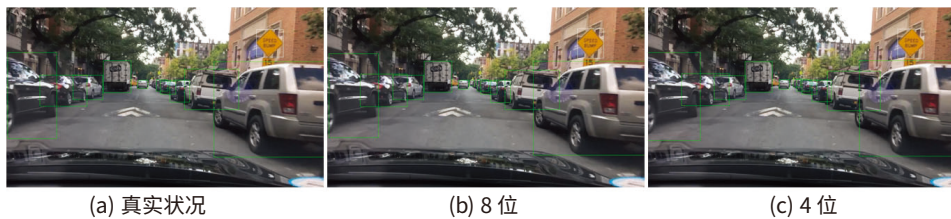
### 2D 检测

在 ADAS 系统中，BDD100K [参考资料 16] 数据集用于 2D 检测。此外，FPN 结构被添加到 ResNet18-SSD 中，用作检测网络。实验结果如表 3 所示。

表 3: 不同位宽下的检测精度

模型	A/W	输入/输出	第一层/最后一层	逐计算元	mAP@0.5 (%)
2D 检测	浮点	浮点	浮点	浮点	39.0
	8/8	8/8	8/8	8	39.5
	4/4	4/8	4/4	8	37.8

表 3 所示的是在经过微调后，8 位量化模型实现了高于浮点模型的 mAP。通过逐渐从 8 位微调到 4 位，最终的 4 位量化模型的 mAP 损耗小于 2%。2D 检测的示意图如图 7 所示。



WP521\_07\_060920

图 7: 2D 检测示意图

### 3D 检测

ADAS 系统的 3D 检测任务使用 KITTI 数据集 [ 参考资料 17]。PointPillars [ 参考资料 18] 用于开展 3D 预测任务。实验结果如表 4 所示。

表 4: 不同位宽下的 3D 检测结果

任务	A/W	输入/输出	第一层/最后一层	逐计算元	中型车 AP@0.5(%)	
					BEV	3D
3D 检测	浮点	浮点	浮点	浮点	90.12	90.03
	8/8	8/8	8/8	8	90.00	89.84
	4/4	4/8	4/4	8	90.07	89.87

如表 4 所示，采用微调技巧后，4 位量化模型的精度仅比浮点模型低 0.16%。8 位和 4 位的 3D 检测结果如图 8 所示。

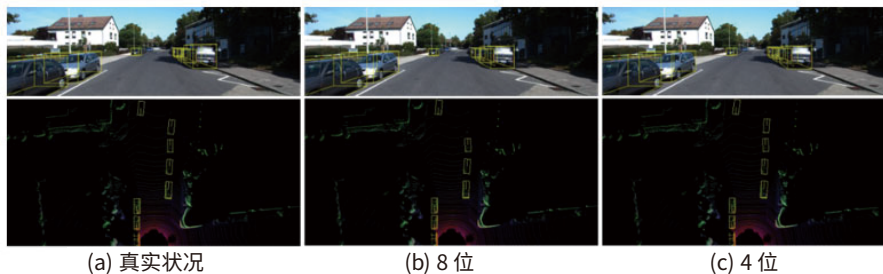


图 8: 摄像头和俯瞰图 3D 检测示意图

### 语义分割

在 ADAS 系统的语义分割任务中，CityScape 的数据集 [ 参考资料 19] 以理解城市视觉场景为重点。实验在以 ResNet18 为基干的特征金字塔网络 (FPN) 上开展。结果如表 5 所示。

表 5: 不同位宽下的语义分割精度

任务	A/W	输入/输出	第一层/最后一层	逐计算元	mIoU(%)
分割	浮点	浮点	浮点	浮点	63.62
	8/8	8/8	8/8	8	63.97
	4/4	4/8	4/4	8	61.90

表 5 显示，8 位模型可实现比浮点模型更高的 mIoU，4 位模型的 mIoU 仅比浮点模型低 1.7%。语义分割的示意图参见图 9。

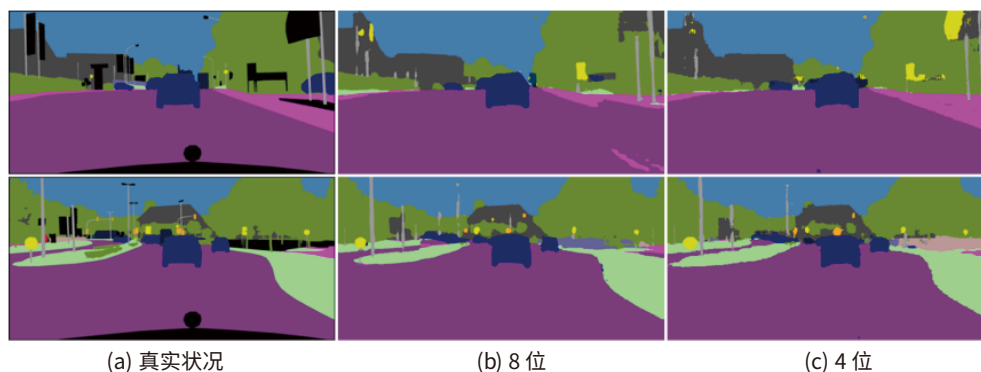


图 9: 语义分割示意图

## 多任务学习

为增强模型的归纳功能和精度，在多任务模型中使用了多个训练数据集，包括用于检测的 Waymo 和 BDD100k，以及用于分割的 BDD100k 和 Cityscapes[参考资料 19]。这些研究在以 ResNet18 为基干的特征金字塔网络 (FPN) 上开展。结果如表 6 所示。

表 6: 不同位宽下的多任务精度

任务	A/W	输入/输出	第一层/最后一层	逐计算元	精度 (%)	
					检测 (mAP)	分割 (mIoU)
多任务	浮点	浮点	浮点	浮点	39.06	42.12
	8/8	8/8	8/8	8	39.94	45.3
	4/4	4/8	4/4	8	37.4	43.91

表 6 显示，8 位量化模型可实现优于浮点模型的 mAP 和与浮点模型保持同等水平的 mIoU。通过逐步微调，与浮点模型相比，最终的 4 位量化模型的 mAP 降低 1.66%，mIoU 提高 1.79%，

仍然劣于 8 位模型的表现。多任务示意图结果如图 10 所示。

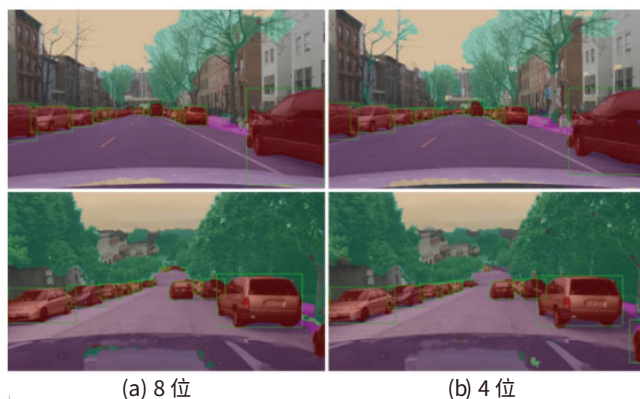


图 10: 多任务学习示意图

## 竞争分析：8 位与 4 位对比

4 位 XDPU 在下列三种评估板上以 300MHz 频率运行：Ultra96 与 Zynq UltraScale+ MPSoC ZCU104 和 ZCU102。表 7 所示的是 4 位 XDPU 和 8 位 XDPU 的比较情况。在不同的 FPGA 上，4 位 XDPU 实现的性能提升在 1.5 倍到 2.0 倍之间。例如，ZCU102 板使用的硬件资源没有增加，但性能提高 2 倍。

表 7: 4 位 XDPU 和 8 位 XDPU 的性能比较

	Ultra96	ZCU104	ZCU102
8-Bit XDPU	691GOPs	2.45TOPs	3.69TOPs
4-Bit XDPU	1228GOPs	3.69TOPs	7.37TOPs

对于两个精度不同的加速器，在启用池化、逐计算元逐深度卷积和平均池化等全部功能后，对资源进行比较。如表 8 中所示，在相同的性能架构下，DSP 和 RAM 的占用显著下降。鉴于资源耗用下降，4 位 XDPU 架构被扩展到 B8192 的最大规模。使用 B8192 架构能以单器件实现更高性能。

表 8: 4 位 XDPU 和 8 位 XDPU 的资源消耗比较

4 位 XDPU					8 位 XDPU				
架构	LUT	寄存器	块 RAM	DSP	架构	LUT	寄存器	块 RAM	DSP
B512 (4x 8x 8)	25322	32211	41.5	62	B512 (4x 8x 8)	26482	33530	73.5	110
B800 (4x10x10)	29137	38398	56	97	B800 (4x10x10)	29711	40184	91.5	157
B1024 (8x 8x 8)	31378	42699	57.5	122	B1024 (8x 8x 8)	32598	47282	105.5	218
B1152 (4x12x12)	32928	43337	73	116	B1152 (4x12x12)	31769	46462	123	212
B1600 (8x10x10)	36504	52101	76	192	B1600 (8x10x10)	36838	58204	127.5	312

表 8: 4 位 XDPU 和 8 位 XDPU 的资源消耗比较 (续)

4 位 XDPU					8 位 XDPU				
B2304 (8x12x12)	38389	58090	97	230	B2304 (8x12x12)	40039	68469	167	422
B3136 (8x14x14)	43316	67901	120	324	B3136 (8x14x14)	43972	79141	210	548
B4096 (8x16x16)	48232	75896	145.5	370	B4096 (8x16x16)	49754	97882	257	690
B8192 (8x32x16)	55890	91426	201.5	690	B8192 (8x32x16)	不支持			

以表 3 中 13.6FLOP 的 2D 检测模型为例，两个高精度模型 4/4 和 8/8 分别使用 4 位 XDPU 和 8 位 XDPU 进行测试。该网络的计算要求是 13.6GOP。2D 检测网络的帧率如表 9 所示，测试不包含预处理和后处理。鉴于效率和网络类型的差异，性能和帧率之间不存在线性关系。如表 9 所示，4 位 XDPU 的帧率在所有平台上均优于 8 位 XDPU。

表 9: 4 位 DPU 和 8 位 DPU 之间的帧率比较

	Ultra96	ZCU104	ZCU102
2D 检测 (8/8)	30fps	101fps	151fps
2D 检测 (4/4)	53fps	145fps	230fps

## 结论

本白皮书介绍了一种运行在 Zynq UltraScale+ MPSoC 和 Zynq-7000 SoC 系列 (16nm 和 28nm) 器件上的全流程、硬件友好型量化解决方案，可用作 CNN 的低精度加速器。此外，本白皮书也介绍了如何在赛灵思 DSP 片上优化 INT4，从而在一个时钟周期内完成 4 通道 INT4 相乘。卷积的计算要求可通过打包 DSP 予以满足。与 INT8 XDPU 解决方案相比，使用 DSP 实现的 INT4 优化在真实硬件上可将处理峰值 GOPS 提升最大 2 倍并将性能最高提升 1.77 倍。这种赛灵思解决方案在各种 CV 任务上都获得了媲美浮点模型的结果。对于资源受限和功耗受限的用例，赛灵思继续创新软硬件协同优化方法，为深度学习应用提速。

## 鸣谢

下列赛灵思员工参加了本白皮书的写作或为本白皮书的成文做出了贡献：Tiantian Han (软件工程师、AI 算法)，Tianyu Zhang (设计工程师 2、AI 边缘计算)，Dong Li (高级软件开发经理、AI 算法)，Guangdong Liu (设计工程师、AI 边缘计算)，Lu Tian (软件开发总监、AI 算法)，Dongliang Xie (设计工程总监、AI 边缘计算) 和 Yi Shan (高级总监、AI)。

## 参考资料

1. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al.2015."Imagenet Large Scale Visual Recognition Challenge."International Journal of Computer Vision 115 (3):211-252.
2. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.2016a."Deep Residual Learning for Image Recognition."In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.
3. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.2016b."Identity Mappings in Deep Residual Networks."In European Conference on Computer Vision, 630-645.Springer.
4. Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam.2017."Mobilenets:Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv Preprint arXiv:1704.04861.
5. Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen.2018."Mobilenetv2:Inverted Residuals and Linear Bottlenecks."In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4510-4520.
6. Williams, S., 2009.Roofline:An Insightful Visual Performance Model for Floating-Point Programs and Multicore.
7. Sambhav R Jain, Albert Gural, Michael Wu, and Chris Dick.2019."Trained Quantization Thresholds For Accurate And Efficient Fixed-Point Inference Of Deep Neural Networks." arXiv Preprint arXiv:1903.08066.
8. Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko.2018."Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference."In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2704-2713.
9. Raghuraman Krishnamoorthi.2018."Quantizing Deep Convolutional Networks for Efficient Inference:A Whitepaper."http://arxiv.org/abs/1806.08342.
10. Yaman Umuroglu, and Magnus Jahre.2017."Streamlined Deployment for Quantized Neural Networks." arXiv Preprint arXiv:1709.04060.
11. Xilinx Inc. UltraScale Architecture DSP Slice User Guide, May 2019.  
[https://www.xilinx.com/support/documentation/user\\_guides/ug579-ultrascale-dsp.pdf](https://www.xilinx.com/support/documentation/user_guides/ug579-ultrascale-dsp.pdf).
12. Yao Fu, Ephrem Wu, Ashish Sirasao, Sedny Attia, Kamran Khan, and Ralph Wittig.2016."Deep Learning with Int8 Optimization on Xilinx Devices."White Paper.
13. Xilinx Inc. DPU for Convolutional Neural Network v3.0 IP Product Guide, Aug 2019.  
[https://www.xilinx.com/support/documentation/ip\\_documentation/dpu/v3\\_0/pg338-dpu.pdf](https://www.xilinx.com/support/documentation/ip_documentation/dpu/v3_0/pg338-dpu.pdf).
14. Alejandro Newell, Kaiyu Yang, and Jia Deng.2016."Stacked Hourglass Networks for Human Pose Estimation."In European Conference on Computer Vision, 483-499.Springer.
15. Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele.2014."2d Human Pose Estimation:New Benchmark and State of the Art Analysis."In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3686-3693.
16. Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell.2018."Bdd100k:A Diverse Driving Video Database with Scalable Annotation Tooling." arXiv Preprint arXiv:1805.04687.
17. Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun.2013."Vision Meets Robotics:The Kitti Dataset."The International Journal of Robotics Research 32 (11):1231-1237.

18. Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom.2019."PointPillars:Fast Encoders for Object Detection from Point Clouds."In the IEEE Conference on Computer Vision and Pattern Recognition.
19. Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele.2016."The Cityscapes Dataset for Semantic Urban Scene Understanding."In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3213-3223.

## 修订历史

下表列出了本文档的修订历史。

日期	版本	修订描述
06/24/2020	1.0.1	排版编辑。
06/19/2020	1.0	赛灵思初始版本。

## 免责声明

本文向贵司 / 您所提供的信息（下称“资料”）仅在对赛灵思产品进行选择和使用参考。在适用法律允许的最大范围内：（1）资料均按“现状”提供，且不保证不存在任何瑕疵，赛灵思在此声明对资料及其状况不作任何保证或担保，无论是明示、暗示还是法定的保证，包括但不限于对适销性、非侵权性或任何特定用途的适用性的保证；且（2）赛灵思对任何因资料发生的或与资料有关的（含对资料的使用）任何损失或赔偿（包括任何直接、间接、特殊、附带或连带损失或赔偿，如数据、利润、商誉的损失或任何因第三方行为造成的任何类型的损失或赔偿），均不承担责任，不论该等损失或者赔偿是何种类或性质，也不论是基于合同、侵权、过失或是其他责任认定原理，即便该损失或赔偿可以合理预见或赛灵思事前被告知有发生该损失或赔偿的可能。赛灵思无义务纠正资料中包含的任何错误，也无义务对资料或产品说明书发生的更新进行通知。未经赛灵思公司的事先书面许可，贵司 / 您不得复制、修改、分发或公开展示本资料。部分产品受赛灵思有限保证条款的约束，请参阅赛灵思销售条款：<http://china.xilinx.com/legal.htm#tos>；IP 核可能受赛灵思向贵司 / 您签发的许可证中所包含的保证与支持条款的约束。赛灵思产品并非为故障安全保护目的而设计，也不具备此故障安全保护功能，不能用于任何需要专门故障安全保护性能的用途。如果把赛灵思产品应用于此类特殊用途，贵司 / 您将自行承担风险和责任。请参阅赛灵思销售条款：[china.xilinx.com/legal.htm#tos](http://china.xilinx.com/legal.htm#tos)。

## 关于与汽车相关用途的免责声明

汽车产品（产品部件号中标识为“XA”）不保证用于安全气囊的开发或用于影响车辆控制的应用（“安全应用”），除非在该赛灵思产品中具备故障安全保护或者额外功能，符合 ISO 26262 汽车安全标准（“安全设计”）。为安全起见，客户应在使用或分销任何集成有该产品的系统之前，对这些系统进行全面测试。在没有安全设计的安全应用中使用产品的风险完全由客户承担，仅受有关产品责任的适用法律和法规限制。