存内模拟计算架构与张量处理单元的异构集成

Mohammed E. Elbtity
elbtity@ieee.org 南卡罗来纳大学 哥伦比亚，南卡罗来纳州，美国

Brendan Reidy
bcreidy@email.sc.edu 南卡罗来纳大学 哥伦比亚，南卡罗来纳州，美国

Md Hasibul Amin
ma77@email.sc.edu 南卡罗来纳大学 哥伦比亚，南卡罗来纳州，美国

Ramtin Zand
ramtin@cse.sc.edu 南卡罗来纳大学 哥伦比亚，南卡罗来纳州，美国

arXiv:2304.09258v1 [cs.AR] 18 Apr 2023

摘要

张量处理单元（TPU）是用于机器学习任务的专用硬件加速器，在执行卷积神经网络（CNN）中的卷积层时展现出显著的性能提升。然而，在全连接层（FC 层）中，它们难以保持相同的效率，从而导致硬件利用率不佳。另一方面，存内模拟计算（IMAC）架构在执行全连接层时已展现出显著的加速比。本文提出了一种新颖的异构混合信号混合精度架构，将存内模拟计算单元与边缘张量处理单元相结合，以提升移动卷积神经网络的性能。为发挥张量处理单元在卷积层以及 IMAC 电路在全连接层中的优势，我们提出了一种统一学习算法，其中融入混合精度训练技术，以缓解模型部署在 TPU-IMAC架构上时可能出现的准确率下降。仿真结果表明，与传统 TPU 架构相比，在多种卷积神经网络模型上，TPU-IMAC 配置可实现最高达 2.59× 的性能提升，并减少 88% 的内存，同时保持相当的准确率。TPU-IMAC架构在能效和高性能至关重要的各种应用中展现出潜力，例如边缘计算和移动设备中的实时处理。统一训练算法以及 IMAC 与 TPU 架构的集成，进一步彰显了本研究对更广泛机器学习领域的潜在影响。

1 引言

深度学习模型已广泛应用于各种现实生活场景，包括语言翻译、计算机视觉、医疗保健和自动驾驶汽车 [1, 2, 3, 4]。然而，这导致机器学习（ML）工作负载的计算需求显著增加，而传统的冯·诺依曼架构难以满足这些需求。为应对这一挑战，存内计算（IMC）等替代架构应运而生。这类架构直接在数据所在位置执行计算，从而减少了机器学习等数据密集型应用中内存与处理器之间数据传输所产生的高能耗 [5]。传统的存内计算架构通常采用阻变随机存取存储器（RRAM） [6,7] 和磁阻随机存取存储器（MRAM） [8]等新兴技术，以加速矩阵向量乘法运算（MVM）机器学习工作负载中的矩阵向量乘法运算，通过大规模并行和模拟计算实现加速。然而，激活函数等其他功能块仍依赖数字计算，因而会因信号转换单元而产生额外能耗 [9, 10]。另一方面，存内模拟计算（IMAC）架构是存内计算架构的一类，能够在模拟域中同时实现矩阵向量乘法运算和非线性向量运算，从而无需在深度神经网络（DNN）层之间使用信号转换单元 [11]。以往研究表明，在深度神经网络中实现密集型全连接层（FC）时，IMAC架构可将延迟和能耗降低数个数量级 [12]。然而，要将IMAC架构用于卷积神经网络（CNN）中的卷积层，需要将这些层展开并重塑为矩阵向量乘法运算 [13], , 从而形成大型交叉阵列，而这些阵列可能容易受到噪声和互连寄生效应引发的可靠性问题影响 [14, 15]。

另一方面，近年来为加速机器学习工作负载而推出的最具前景的数字硬件加速器之一是张量处理单元（TPU），其通过由处理单元（PE）构成的深度流水线网络，即脉动阵列，执行并行计算 [16]。TPU 中的脉动阵列通过复用从内存和寄存器中获取的值，并减少不规则的中间内存访问，从而降低能耗并提升性能 [17]。TPU 在执行通用矩阵乘法运算方面已展现出令人瞩目的效果；该运算是卷积神经网络（CNN）的关键组件，尤其应用于卷积层。然而，由于全连接层通常包含数量庞大的权重，TPU 在执行全连接层时难以维持同等水平的性能。这限制了权重复用，并使执行过程需要多次迭代，导致硬件利用率低下和能耗较高 [18]。需要注意的是，在实际应用中，脉动阵列通常采用对称尺寸，以优化卷积层执行性能。然而，如果采用非对称维度进行设计，则可以加速全连接运算，但代价是卷积层执行性能下降。近期研究 [18] 表明，由于空间复用特性以及能够丢弃神经网络中未使用特征的能力，TPU 在执行卷积层时的性能优于 GPU 和中央处理器（CPU）。然而，由于权重复用率较低，TPU 在运行全连接层时不太受青睐，其性能也不如 GPU。随着模型规模增大，这会导致较高的存储器流量和能耗。给定神经网络架构中全连接层所占的比例

会影响在 TPU 上执行整个模型的总体性能。我们使用 Scale-Sim [19]开展的内部实验也证实，与卷积层相比，TPU 在执行全连接层时性能较差，且硬件利用率较低。

本文提出了一种新颖的 TPU-IMAC 混合架构，结合 TPU 和 IMAC 的优势，可高效执行卷积层和全连接层。该架构不仅提升了性能，还降低了各种规模卷积神经网络模型的内存带宽需求。本文结构如下：第 2 节介绍本研究采用的 IMAC架构，第 3 节详细介绍所提出的 TPU-IMAC架构。第 4 节说明用于训练混合精度卷积神经网络模型的架构感知学习方法。第 5 节讨论准确率、内存利用率和性能结果。最后，第 6 节总结全文。

2 IMAC架构

图1所示的示意图展示了 IMAC 架构的结构。这些架构由一组紧密互连的子阵列构成，并由可编程交换块连接。每个 IMAC 子阵列由忆阻器交叉阵列、差分放大器和神经元电路组成，如图1 (b) 所示。为简化起见，我们仅展示阵列的读路径，因为本研究聚焦于神经网络的推理阶段。深度神经网络的突触连接由忆阻器交叉阵列构成，其列数和行数可分别根据单个全连接层的输入节点数和输出节点数确定。忆阻器交叉阵列利用电路中的欧姆定律和基尔霍夫定律等物理机制，在模拟域执行矩阵向量乘法运算。具体而言，乘法运算遵循欧姆定律 ($I = GV$)，而累加运算则基于电荷守恒，正如基尔霍夫定律所述。

在配置阶段，调节连接到差分放大器的两个忆阻器件的相对电导，即可实现零、正和负权重。图 1(b) 展示了差分放大器连接到交叉阵列中相邻的两行，这两行分别以绿色和红色标示，代表正电导行和负电导行，

地。电导值为 $G_{i,j}^{+}$和 $G_{i,j}^{-}$的忆阻器件差分对用于实现每个权重值 $W_{i,j}$，其中 $W_{i,j} \propto G_{i,j}^{+} - G_{i,j}^{-}$。因此，使用一对具有 $G_{i,j}^{+} = 1/R_{high}$ 和$G_{i,j}^{-} = 1/R_{low}$ 的忆阻器件来实现负权重，反之亦然。当 $G_{i,j}^{+} = G_{i,j}^{-}$时，可实现零权重。

在推理阶段，写字线（WWLs）被禁用，读字线（RWLs）被启用。如图1 (b)所示，此过程会产生两种类型的电流， I^{+} 和 I^{-}，其电流幅值取决于输入信号和忆阻突触的电阻。每一行忆阻突触共用一个差分放大器，该放大器产生的输出电压与该行两条字线电流之差成正比，即 $\sum_i (I_{i,n}^{+} - I_{i,n}^{-})$，其中 i 是输入层中的节点总数， n 是行号。最后，差分放大器的输出被馈送至模拟神经元

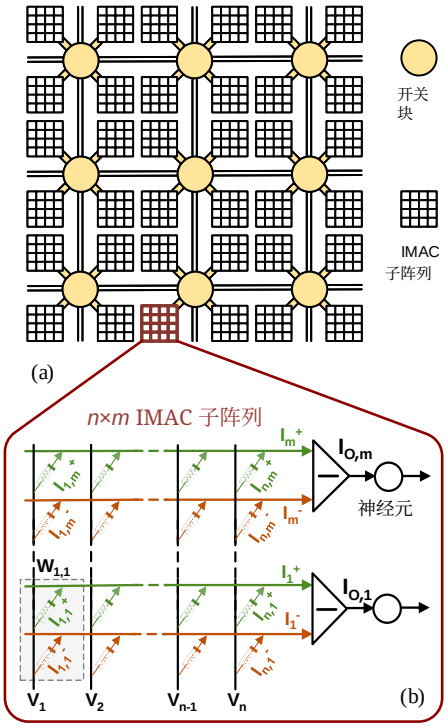


图1：(a) IMAC架构，(b) 一个包含忆阻器交叉阵列和模拟神经元的 $n \times m$ 子阵列。

以计算激活函数。对于给定层，该架构在每个子阵列中同时执行矩阵向量乘法运算和神经元激活函数计算，然后将结果传递给下一个IMAC子阵列以计算下一层。本文采用模拟*Sigmoid*函数神经元，它由两个电阻器件和一个基于CMOS的反相器组成 [11]。神经元电路中的电阻器件构成一个分压器，可降低反相器线性工作区的斜率，从而实现平滑的高电压到低电压转换，形成Sigmoid函数。

3 所提出的 TPU-IMAC架构。

通常，TPU 架构包含大量处理单元，其中包括负责执行矩阵-矩阵、向量-向量和矩阵-向量乘法的乘累加（MAC）单元[20]。TPU 利用脉动阵列，通过重复使用从内存和寄存器中读取的值来提升性能 [17], ，从而最大限度地减少对缓冲区的读写。输入数据并行馈入阵列，通常以对角波前模式传播 [16, 21]。乘加单元的基本架构决定了脉动阵列中的数据流，而不同的数据流架构会影响功耗、硬件利用率百分比以及整体性能。

神经网络处理中，脉动阵列的数据流经过精心安排，以确定性顺序提取数据并生成输出结果，从而优化深度学习算法中的主要运算——乘加运算。人们提出了多种数据流算法

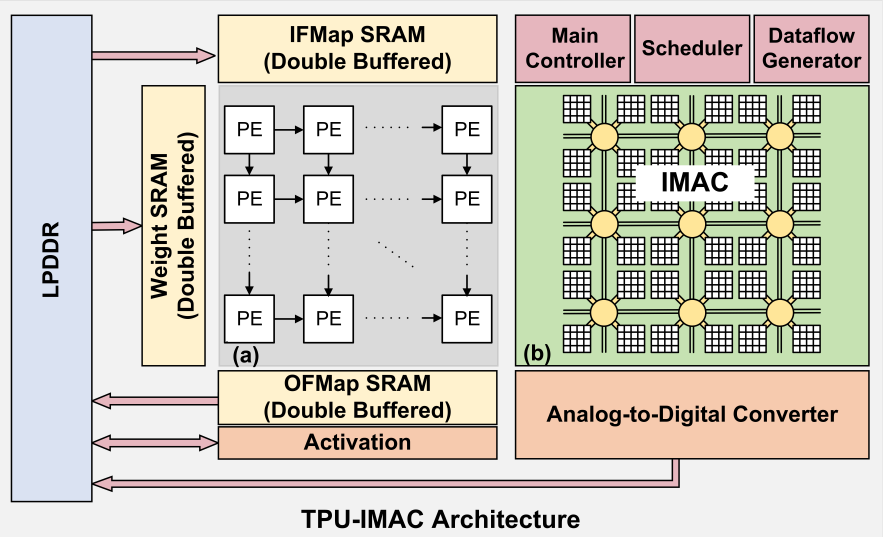


图 2：异构 TPU-IMAC 架构。(a) 输出驻留脉动阵列，包含由处理单元 (PE) 构成的深度流水线网络；(b) 由交换盒互连的紧密耦合忆阻器子阵列网络。

，大致可分为输入驻留数据流（IS） [22], 权重驻留（WS） [16], 输出驻留（OS） [23], 行驻留（RS） [24], 和无本地复用（NLR） [25, 26]。“驻留”一词指的是在执行乘加运算时，数据保留在处理单元中，不经过寄存器传输 [19]。在 WS 方法中，每个权重都会预先加载到乘加单元上。每个时钟周期，待与固定权重相乘的输入元素会广播到阵列中的各个乘加单元，从而在每个时钟周期产生部分和。这一过程沿列方向分布，以从脉动阵列底部生成结果 [16]。输入驻留数据流的过程几乎相同，只是输入在矩阵中保持不变，而权重则沿水平方向传递。对于输出驻留数据流，输出作为输入附加到乘加单元上，权重则在各单元之间循环传递。图 2(a) 展示了输出驻留脉动阵列的架构，其中权重从阵列左侧输入，输入特征图（IFMap）则从顶部流入。每个处理单元负责生成一个输出特征图（OFMap）。

我们提出一种混合精度、混合信号的异构技术系统，将基于电阻式 RRAM 的 IMAC 与传统的基于 CMOS 的 TPU 架构集成在一起。如图 2 所示，所提出的 TPU-IMAC架构保留了传统 TPU 的功能，同时将 IMAC 电路直接连接至 TPU 脉动阵列中的处理单元，从而提升整体性能。我们采用脉动阵列中的 OS 架构，利用其将 OFMap 数据固定在相应处理单元中的能力。这样便可将每个 OFMap 的最高有效位（*sign* 位）直接连接至 IMAC 输入端，从而根据神经网络拓扑，将卷积层的结果直接从 TPU 传输至 IMAC，以执行后续的全连接层。通过反相器连接 *sign* 位，即可完成数据量化，无需专用硬件或软件功能；此操作将正 OFMap（≥ 0）转换为高逻辑位 ’1’，将负 OFMap（高 *sign* 位）转换为低逻辑位 ’0’。在 IMAC 执行全连接层期间，通过由主控制器 控制的三态缓冲器，将这种单比特精度数据连接至 IMAC 输入端。

在所提出的架构中，我们采用低功耗双倍数据速率 (LPDDR) 内存。由于工作电压较低且支持节能模式，LPDDR 适用于边缘设备。LPDDR 根据数据流生成器和主控制器的指令，负责存储和读取神经网络的输入特征图、权重和输出特征图数据。通常，数据会预先加载到 LPDDR 中。当 TPU-IMAC 开始执行工作负载时，数据流生成器会根据输出驻留数据流算法生成读取地址轨迹，从 LPDDR 中读取输入特征图和权重，并分别将其发送至输入特征图 SRAM 和权重 SRAM。主控制器会根据调度器的请求，协助完成这一数据传输。

执行完第一个卷积层后，输出特征图数据会根据数据流生成器生成的数据流和写入轨迹（地址），从处理单元转发到输出特征图 SRAM，然后传输到 LPDDR。调度器 负责调度 CNN 工作负载的每一层，而数据流生成器 和主控制器 负责管理 CNN 工作负载执行的整体流程。根据 CNN 工作负载的不同，调度器 可能需要执行一个或多个全连接层。在这种情况下，调度器 会通知主控制器，由其激活中间的三态缓冲器，从而在*sign* 信号与 IMAC 输入之间传输各输出特征图（存储在脉动阵列的各处理单元中）的数据。全连接层的输入数据采用低或高逻辑形式，而 IMAC 权重采用三值逻辑（取值为 1、0 和 -1）。

我们在所提出的架构上训练 CNN 工作负载，以缓解低精度模拟计算可能造成的准确率损失。当数据从脉动阵列传输到 IMAC 后，IMAC 根据调度器的请求执行所需的全连接层，每个全连接层均在单个时钟周期内执行，从而提升性能。需要注意的是，如第 2 节所述，在配置阶段，全连接层的权重会预先加载到 IMAC 电路中的忆阻器件上。在

在 IMAC 上完成全连接层的执行后，结果通过 IMAC 架构中配备的模数转换器（ADC）转换为数字信号，然后写回 LPDDR，供用户访问。

值得注意的是，我们的架构不需要数模转换器（DAC），因为我们的IMAC接受直接来自脉动阵列中每个PE的 *sign*位的二值化输入，从而降低了功耗。如果需要激活层或归一化层，则会在TPU的脉动阵列外部实现专用硬件单元，以相应地执行这些操作。图2还使用箭头简化展示了架构组件之间的数据流。

总之，调度器控制每一层的执行，并根据CNN拓扑结构进行编程。数据流生成器根据OS数据流算法生成轨迹（地址），供LPDDR读取数据并将其发送至输入特征图存储器和权重存储器，或将输出特征图存储器或ADC中的结果写回LPDDR。主控制器管理各组件的使能信号，以及TPU脉动阵列与IMAC 电路之间的三态缓冲器。值得注意的是，TPU脉动阵列中的每个PE都包含一个全精度32位浮点（FP）乘累加单元，而IMAC则使用三值权重和二值输入，如前文所述。所提出的TPU-IMAC架构将这种独特的精度组合与混合信号技术相结合，为提升CNN推理性能提供了一种创新方法。

4 架构感知型CNN模型开发

我们开发了一种硬件感知学习算法，在保持准确率的同时，充分发挥 TPU-IMAC架构的优势。我们的混合精度、混合信号TPU-IMAC架构具有计算约束和独特特性，算法对此均加以考虑。学习过程分为两个步骤。第一步，在网络的全连接层部分之前插入一个 *tanh* 激活函数，以确保激活值处于{*-1, 1*}范围内。随后，我们使用反向传播算法训练 CNN。第二步，冻结已训练的卷积层，并使用三值权重重新训练全连接层。在此过程中，我们将 *tanh* 激活函数替换为 *sign* 函数，使全连接层的输入值为*-1*和*1*。这一点很重要，因为将全连接层的输入限制为*-1*和*1*后，我们只需将最后一个卷积层输出特征图的符号位传输至IMAC，无需使用任何数模转换单元。此外，我们根据 IMAC 的特性调整全连接层，采用三值突触和 *sigmoid* 激活函数；如第 2 节所述，这些结构可通过基于RRAM的突触和神经元实现。

在表1中，我们给出了前文所述每个训练步骤中卷积层和全连接层的激活函数及权重精度。在训练第二步的前向传播中，全连接层使用三值权重进行训练，而在反向传播中使用浮点权重。使用三值权重重新训练模型后，仅保留三值权重。值得注意的是，大多数现有CNN模型使用ReLU函数来实现非饱和非线性，这是因为与数字实现的{v1}和{v2}激活函数相比，ReLU函数实现简单且具有性能优势

表1：所提出的 TPU-IMAC 感知学习算法在不同阶段的权重和激活值。

Step	层	组件	前向 Pass	反向 Pass
1	All	权重 神经元	$\mathbf{w}_i \in R$ ReLU	$\mathbf{w}_i \in R$ ReLU
2	Conv	权重 Act.	冻结 ReLU	
	FC	权重 神经元	$\mathbf{W}_i \in \{-1,0,+1\}$ *Sigmoid*函数	$\mathbf{w}_i \in R$ *Sigmoid*函数

然而，在我们提出的IMAC架构中，模拟神经元能够实现高性能的Sigmoid函数激活，从而以极小的性能开销带来准确率方面的优势。尽管在TPU上实现的卷积层中仍使用ReLU，但在IMAC架构中，我们采用模拟Sigmoid函数激活。为了充分利用具有 32×32 脉动阵列规模的TPU-IMAC架构，我们对CNN模型进行了修改，使最后一个卷积层的OFMap展平后输入全连接层的线性向量恰好包含1024个元素。这样，我们便可以将脉动阵列中计算并存储的最后一个卷积层OFMap直接传输至IMAC，而无需与主内存之间传输数据。对于VGG9和ResNet，这是通过增加最终卷积层中的通道数量并减小最大池化层的步幅实现的；对于MobileNetV1和MobileNetV2，则是通过增加最终卷积层中的通道数量实现的。

5 结果与讨论

5.1 准确率

在本研究中，我们针对七种不同的 CNN 架构进行了实验，包括用于 MNIST 数据集的 LeNet [27] 、用于 CIFAR-10 [31] 数据集的 VGG-9[28]、MobileNet V1 和 V2 [29], 以及 ResNet-18 [30]，还有用于 CIFAR-100数据集的 MobileNet V1 和 V2，以评估 TPU-IMAC架构相较于张量处理器架构的优势。为张量处理器架构训练的模型采用 FP32精度，而 TPU-IMAC 模型则采用混合精度，包含 FP32 卷积层和三值稠密层。两种架构的准确率如表 2 所示。仿真结果表明，TPU-IMAC 实现用于 CIFAR-10数据集时，准确率下降幅度很小，不到 1%。具体而言，VGG-9 和 ResNet-18 模型的准确率下降幅度分别为最大值 0.59% 和最小值 0.12%。对于 LeNet 数据集，准确率下降了 1.13%，这可归因于其全连接层与卷积层比例较高。最后，针对 CIFAR-100数据集部署在 TPU-IMAC 上的混合精度模型，准确率下降幅度接近 3%，这可能是由于该数据集较为复杂，且其全连接层规模大于用于 CIFAR-10数据集的卷积神经网络模型。

表 2：不同卷积神经网络模型的准确率、内存利用率和执行时间。

模型	数据集	准确率 (%)		内存 (MB)				周期数 ($\times 10^3$)	
		张量处理器	TPU-IMAC	TPU	TPU-IMAC			张量处理器	TPU-IMAC
				SRAM	SRAM	阻变随机存取存储器	总计		
LeNet	MNIST	98.95	97.82	0.177	0.01	0.01	0.02	2.475	0.956
VGG9	CIFAR-10	90.9	90.31	38.747	34.512	0.265	34.776	331	297.18
MobileNetV1		92.89	92.7	16.976	12.74	0.265	13.005	214.9	181.1
MobileNetV2		93.73	93.43	12.904	8.668	0.265	8.933	338.7	304.9
ResNet-18		94.96	94.84	48.872	44.637	0.265	44.902	681.7	647.8
MobileNetV1	CIFAR-100	66.21	63.07	17.344	12.74	0.288	13.028	218	181.1
MobileNetV2		73.06	70.14	13.272	8.668	0.288	8.956	356	319.1

表 3：不同卷积神经网络模型中，TPU-IMAC 相较于 TPU 的准确率、性能和内存削减。

模型	数据集	准确率差异	内存减少	加速比
LeNet	MNIST	-1.13%	88.34%	2.59
VGG9	CIFAR-10	-0.59%	10.25%	1.11
MobileNetV1		-0.19%	23.39%	1.19
MobileNetV2		-0.3%	30.77%	1.11
ResNet-18		-0.12%	8.12%	1.05
MobileNetV1	CIFAR-100	-3.14%	24.89%	1.2
MobileNetV2		-2.92%	32.52%	1.12

5.2 内存利用率

在资源有限的边缘设备上部署机器学习工作负载时，内存占用是至关重要的考量因素 [32]。卷积神经网络模型中的稠密全连接层通常会显著增加内存使用量。为此，采用极低精度表示（例如仅用 2 位表示的三值权重），可以大幅降低卷积神经网络模型的内存使用量。表 2 和表 3 对比了部署在张量处理器上的单精度浮点模型与部署在 TPU-IMAC 上的混合精度模型的内存利用率。仿真结果表明，得益于将传统 SRAM 单元与新兴电阻式存储技术（如阻变随机存取存储器）集成在一起的混合内存架构，TPU-IMAC架构能够有效降低所研究卷积神经网络模型的内存使用量。具体而言，与张量处理器相比，TPU-IMAC架构针对 LeNet、CIFAR-10 和 CIFAR-100 数据集构建的卷积神经网络模型，平均所需存储空间分别减少了 88.34%、18.13% 和 28.7%。

5.3 性能

在本研究中，我们使用 Scale-Sim 仿真器对所提出的 TPU-IMAC架构进行了性能分析。Scale-Sim 是一种周期精确、架构级的仿真工具，专为执行卷积神经网络的基于脉动阵列的加速器而设计 [19]。该工具提供灵活的仿真选项，包括调整脉动阵列架构参数（如尺寸、数据流规范（IS、WS 和 OS）），以及动态随机存取存储器和 SRAM 的大小，还有输入特征图、滤波器和输出特征图的偏移量。借助 Scale-Sim，我们能够在各种配置和场景下评估 TPU-IMAC架构的性能，从而提供有关其在移动设备上执行 CNN 工作负载的潜在优势和局限性的见解

我们向 Scale-Sim 提供了有关 CNN 工作负载的详细信息，包括每一层的维度、输入特征图的维度、滤波器维度以及每一层的通道数。Scale-Sim 利用这些信息报告了执行每一层所需的时钟周期、硬件利用率、内存带宽，以及整个 CNN 执行过程的动态随机存取存储器索引轨迹。随后，我们汇总了执行每一层所需的时钟周期，以确定完成整个 CNN 工作负载所需的总时钟周期数。我们提出的 TPU-IMAC架构使每个全连接层仅需一个周期即可执行，因此，整体性能提升可通过以下方式计算：将仅在张量处理器上执行整个工作负载所需的总时钟周期数，除以在张量处理器上执行卷积层所需的时钟周期数与在集成式 IMAC 上运行全连接层所需的时钟周期数之和。需要强调的是，由于脉动阵列中的处理单元与 IMAC 直接相连，在脉动阵列与 IMAC 之间传输数据不会浪费任何周期数。

表 2 展示了在张量处理器和 TPU-IMAC架构上运行 CNN 时的周期数执行时间。表 3 中的结果表明，采用所提出的 TPU-IMAC架构可显著提升性能，尤其是在使用 MNIST 数据集的 LeNet 模型上，性能提升了 $2.59\times$。此外，在运行 VGG、MobileNetV1、MobileNetV2 和 ResNet-18 等其他模型时，性能提升幅度介于 $1.05\times - 1.2\times$。这些性能提升幅度的差异可归因于各模型在 IMAC 上执行的全连接层的大小和数量。具体而言，CNN 模型中的全连接层规模越大、数量越多，采用我们的 TPU-IMAC架构所带来的性能提升通常也越显著。这些发现表明，我们提出的架构在移动设备上执行 CNN 工作负载具有潜在优势，尤其适用于包含大量全连接层的模型。

6 结论

在本文中，我们提出了一种新型架构，即 TPU-IMAC。该架构结合了异构混合精度技术和混合信号技术的优势，以提升移动设备上 CNN推理的能效和性能。具体而言，我们提出将模拟IMAC单元与数字

张量处理器分别结合，用于实现卷积神经网络模型的全连接层和卷积层。我们研究了高效实现 TPU-IMAC架构所需的架构级要求，并通过对多种卷积神经网络模型（例如 LeNet、VGG9、MobileNetV1、MobileNetV2 和 ResNet）进行架构级仿真，评估了其潜在的性能和能耗优势。仿真结果表明，根据 CNN 工作负载中全连接层的数量和规模，TPU-IMAC架构可实现 5% – 158% 的性能提升。此外，我们的分析显示，根据模型的不同，所提出的 TPU-IMAC架构可将内存需求显著降低 8%–88%。这些改进符合阿姆达尔定律，并与全连接层和卷积层的比例成正比。此外，由于 IMAC 使用三值权重，我们开发了一种架构感知的混合精度卷积神经网络模型训练方法，以缓解潜在的准确率下降。结果表明，对于部署在TPU-IMAC架构上的卷积神经网络模型，该方法仅造成极小的准确率下降。具体而言，我们报告了 CIFAR-10 上 0.12% 至 0.59% 的准确率下降、MNIST 上 1.13% 的准确率下降，以及 CIFAR-100数据集上近 3% 的准确率下降。我们的研究结果凸显了采用 TPU-IMAC架构进行移动设备上 CNN推理的潜在优势，并为未来的研究提供了多种机会。

参考文献

[1] Tzoof Hemed、Nitai Lavie 和 R. Kaplan。2018 年。精简型智能手机节点上的分布式深度学习。*2018* 年以色列电气工程科学国际会议（*ICSEE*），1–5。[2] Zhenxing Zhou 等。2020 年。基于深度学习和 Jetson Nano 的便携式香港手语翻译平台。第 *22* 届 *ACM* 计算机与无障碍国际会议。[3] A. Shrivastava 等。2019 年。用于环境性肠病和乳糜泻视觉识别的深度学习。*2019* 年 *IEEE EMBS* 生物医学与健康信息学国际会议（*BHI*），1–4。[4] R. Miotto 等。2018 年。医疗保健领域的深度学习：综述、机遇与挑战。生物信息学简报，19 6，1236–1246。[5] A. Sebastian 等。2020 年。用于存内计算的内存器件及其应用。自然·纳米技术，15，529–544。[6] Noha Shaarawy 等。2018 年。2t2m 混合 CMOS-忆阻器型阻变随机存取存储器的设计与分析。微电子学杂志，73，75–85。[7] Shihui Yin 等。2020 年。采用单片集成阻变随机存取存储器和 90 纳米 CMOS、面向二值深度神经网络的高吞吐量存内计算。*IEEE* 电子器件学报，67，4185–4192。[8] S. Angizi 等。2020 年。Mrima：一种基于 MRAM 的存内加速器。*IEEE* 集成电路与系统计算机辅助设计学报，39，5，1123 1136 10 1109 2019 2907886 –。doi：。/TCAD。。[9] Aayush Ankit 等。2019 年。Puma：一种面向机器学习推理的可编程超高效忆阻器加速器。载于第二十四届编程语言与操作系统架构支持国际会议论文集（ASPLOS ’19）。计算机协会，美国罗德岛州普罗维登斯，715–731。ISBN：9781450362405。doi：10.1145/3297858.3304049。https://doi.org/10.1145/3297858.3304049。[10] Miao Hu 等。2016 年。神经形态计算点积引擎：对 1*2016 53*1 1 6 10 1145 t m 交叉阵列进行编程，以加速矩阵向量乘法。载于第 *nd* 届 *ACM/EDAC/IEEE* 设计自动化会议（*DAC*），–。doi：。/ 2897937.2898010。[11] Md Hasibul Amin 等。2022 年。用于存内计算的基于 MRAM 的模拟 Sigmoid函数。*2022* 年大湖区 *VLSI* 研讨会论文集。[12] Mohammed E. Elbtity 等。2021 年。面向移动设备高能效 CNN推理的存内模拟计算协处理器。*2021* 年 *IEEE* 计算机学会 *VLSI* 年度研讨会（*ISVLSI*），188–193。[13] Fan Zhang 等。2022 年。Xbm：一种用于高效多任务适配的交叉阵列逐列二值掩码学习方法。载于*2022* 年第 *27* 届亚太设计自动化会议（*ASP-DAC*），610–615。doi：10.1109/ASP-DAC52403.2022.9712508。[14] Md Hasibul Amin、Mohammed E. Elbtity 和 Ramtin Zand。2022 年。Xbar-partitionin g:
模拟存内计算电路中应对寄生效应和噪声容限的一种实用方法。*IEEE*《电路与系统新兴及精选主题期刊》，12，867–877。
[15] Md Hasibul Amin、Mohammed Elbtity 和 Ramtin Zand。2022。全模拟存内计算架构中的互连寄生效应与分区。载于 *2022*年*IEEE*国际电路与系统研讨会（*ISCAS*），389–393。doi：10.1109/ISCAS48785.2022.9937884。[16] Norman P Jouppi 等。2017。张量处理器的数据中心内性能分析。载于第*44*届年度国际计算机体系结构研讨会论文集，1–12。[17] Norman Jouppi 等。2018。首款张量处理器的动机与评估。*IEEE Micro*，38，3，10–19。doi：10.1109/MM.2018.032271057。[18] Aswathy Ravikumar 等。2022。神经网络结构对使用 GPU/TPU 进行图像分析的卷积神经网络性能和准确率的影响。*PeerJ Computer Science*，8。[19] Ananda Samajdar 等。2018。Scale-sim：脉动式CNN加速器模拟器。*arXiv*预印本 *arXiv:1811.02883*。[20] Mohammed E. Elbtity 等。2022。AptPU：基于近似计算的张量处理器。*IEEE*《电路与系统汇刊*I*：正规论文》，69，5135–5146。[21] Jing Shen 等。2019。专用于CNN的高性能脉动阵列加速器。载于 *2019 IEEE*第*19*届国际通信技术会议（*ICCT*）。IEEE，1200–1204。[22] Sangkug Lym 和 Mattan Erez。2020。FlexSA：用于高效剪枝DNN模型训练的灵活脉动阵列架构。*arXiv*预印本 *arXiv:2004.13027*。[23] Shasha Guo 等。2019。脉动式SNN推理加速器及其协同优化软件框架。载于 *2019*年大湖区*VLSI*研讨会论文集，63–68。[24] Yu-Hsin Chen 等。2016。Eyeriss：用于深度卷积神经网络的高能效可重构加速器。*IEEE*《固态电路期刊》，52，1，127–138。[25] Eric Qin 等。2020。Sigma：用于DNN训练的稀疏非规则GEMM加速器，具有灵活互连。载于 *2020 IEEE*国际高性能计算机体系结构研讨会（*HPCA*）。IEEE，58–70。[26] Hyesook Lim、Vincenzo Piuri 和 Earl E Swartzlander。2000。用于二维离散余弦变换和逆离散余弦变换的串并行架构。*IEEE*《计算机汇刊》，49，12，1297–1309。[27] Y. Lecun 等。1998。应用于文档识别的基于梯度的学习。*IEEE*论文集，86，11，2278–2324。[28] S. Liu 和 W. Deng。2015。使用小规模训练样本的基于超深度卷积神经网络的图像分类。载于 *2015*年第*3*届*IAPR*亚洲模式识别会议（*ACPR*），730–734。doi：10.1109/ACPR.2015.7486599。[29] Andrew G Howard 等。2017。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*。[30] Kaiming He 等。2016。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，770–778。[31] Alex Krizhevsky、Geoffrey Hinton 等。2009。从微小图像中学习多层特征。[32] Soumyalatha Naveen 和 Manjunath R Kounte。2022。分布式卷积神经网络的边缘内存优化。《新兴电信技术汇刊》，33，12，e4648。