嵌入式系统设备上深度网络推理的隐性成本

Chankyu Lee江陵原州国立大学电子工程系韩国江陵
cksrb0710@gwnu.ac.kr

Woohyun Choi高级人工智能系 *LG*电子韩国首尔
settler12@gmail.com

Sangwook Park*江陵原州国立大学电子工程系韩国江陵
spark2@gwnu.ac.kr*通讯作者

arXiv:2601.01698v1 [cs.CC] 5 Jan 2026

摘要—**本研究评估了各种深度学习模型在嵌入式系统环境中的推理性能。以往的研究通常使用乘累加运算来衡量深度模型的计算负载。然而，本研究表明，这一指标在估算嵌入式设备上的推理时间时存在局限。本文探讨了以乘累加运算来表示时，哪些方面被忽略了。在实验中，使用 CIFAR-100 数据集在嵌入式系统设备上执行图像分类任务，并将十种深度模型的推理时间与各模型理论计算得出的乘累加运算量进行比较和分析。结果强调，在针对嵌入式系统中的实时运行优化深度学习模型时，必须考虑张量之间的额外计算量。**

索引词—**实现，卷积神经网络，推理时间，嵌入式系统，实时运行**

I. 引言

消费者在日常生活中会接触到各种基于深度学习的服务，涵盖移动设备、家用电器和汽车。这些服务由制造商提供，他们不断致力于开发先进的深度学习技术，以提高准确率并确保客户满意度。自深度学习兴起以来，准确率一直是技术发展的首要目标。随着时间推移，人们推出了越来越庞大、复杂的模型，并整合了注意力机制、残差连接以及特征维度压缩与扩展等多种模块 [1] [2] [3]。然而，这些模型在训练和推理阶段都需要大量的计算、内存存储和功耗。鉴于这一挑战，近期深度模型的发展不仅关注准确率，也审慎考量模型大小（即参数数量）和计算效率[4]。

通常，减少模型参数数量与提高模型准确率之间存在权衡。因此，在内存有限的设备上部署深度模型时，目标是在保持准确率的同时尽可能减少参数数量。直观而言，人们可能会认为，随着模型大小减小，推理阶段的计算量也会随之减少。然而，根据模型架构的不同，这种预期可能会产生误导。表 I 总结了在嵌入式系统中使用 MNIST 数据集进行手写数字分类的示例。在这项测试中，我们评估了三个模型。模型 A 是一种标准模型，由两层堆叠的常规卷积层（使用大尺寸卷积核）和一个全连接层组成。模型 B 则将模型 A 中的每个卷积替换为两层堆叠的常规卷积（使用小尺寸卷积核），以减少模型参数数量，同时保持感受野大小 [5]。模型 C 的设计方式是将模型 A 中的每个卷积替换为可分离卷积模块，即深度卷积和逐点卷积 [3]。请注意，为了确保计算时间的比较公平，由于可分离卷积模块中未包含批量归一化，准确率略有下降。有趣的是，这个示例表明，即便模型参数和乘加运算（MAC）更少，计算时间仍可能较长。这一现象给准确评估实现基于深度学习服务的系统所需资源带来了挑战。

在以往的研究中，模型参数数量和乘加运算（MACs）均被用于评估模型复杂度和计算效率 [3]。另一方面，如表 I 所示，这些指标与计算时间之间存在差异。这种不一致性使得嵌入式系统资源需求的预测变得复杂。本文提供了 10 个具有代表性的图像分类模型的推理时间，并探讨了模型组件与计算时间之间的关系。实验聚焦于图像分类任务，并基于 CIFAR100 数据集评估 10 个具有代表性的模型。计算时间在配备 1.8 GHz 处理器和 8 GB 内存的嵌入式系统硬件上进行测量，并将结果与每个模型的准确率、参数数量和计算负载进行比较，以提出高效的设计策略。

本研究获得韩国教育部资助的韩国国家研究基金会（NRF）基础科学研究计划支持（RS-2024-00358953）

表 I 手写数字分类样例测试

		模型 A	模型 B	模型 C
架构-tecure	卷积层 1	5x5，10通道	3x3，5通道 3x3，10通道	5x5，深度卷积，1通道 1x1，10通道
	Act.	ReLU		
	Pool	最大池化 步长=2		
	二维卷积	5x5，20通道	3x3，15通道 3x3，20通道	5x5，深度卷积，10通道 1x1，20通道
	Act.	ReLU		
	Pool	最大池化 步幅=2		
	fc	线性层(980, 10)		
	Out	Softmax		
Comp-复杂度	# 参数	15,090	12,650	10,336
	MACs	1,185,800	616,616	125,440
	推理时间[a]	3.298 毫秒	4.003 毫秒	6.033 毫秒

a 嵌入式系统上 100 个样本的平均推理时间。

II. 材料与方法

A. 数据库与硬件规格

本文聚焦于图像分类任务，研究配备四核64位@1.5GHz处理器和 8 GB 内存的嵌入式设备上的推理时间。采用 CIFAR-100 数据集对每个深度模型进行训练，并评估其模型准确率和推理时间。

B. 模型

本研究使用 10 种模型执行图像分类任务：*VGG16*、*InceptionV3*、*InceptionV4*、*ResNet50*、*SeResNet50*、*Xception*、*MobileNet*、*MobileNetV2*、*ShuffleNet*和*Shuf-fleNetV2*。简而言之，*VGG16* 由 16 层组成，采用 3x3 卷积层和 2x2 最大池化层。该模型提出使用多个小尺寸卷积核进行卷积，以较少的模型参数扩大感受野 [5]。*Inception* 采用并行结构，使用不同尺寸的卷积滤波器有效提取特征。本文考虑了*Inception*的两个版本：版本 3 由 94 层组成，版本 4 则由 149 层组成 [6]，[7]。*ResNet50* 由 50 个残差块组成，通过跳跃连接跨越上层，以实现稳定学习 [2]。*SeResNet50* 融合了挤压与激励模块，通过对*ResNet*中各通道的重要信息加权来提升性能*50* [8]。*Xception*是*Inception* 模型的扩展版本，通过采用深度可分离卷积，以更少的参数实现更高的性能 [9]。*MobileNet* 是一种专为在移动设备上推理而设计的轻量级模型，采用深度可分离卷积 [3]。*MobileNetV2*是后续推出的模型，通过采用倒置残差结构和瓶颈结构，进一步提升了*MobileNet* 的性能 [10]。*ShuffleNet* 通过应用通道混洗，成为一种效率极高的*MobileNet* 变体 [11]。此外，*ShuffleNetV2* 在轻量级网络的效率与准确性之间实现了更好的平衡，是 *ShuffleNet* 的进阶版本 [12]。

上述十种模型的训练和推理均在 Python环境中使用 PyTorch库实现。

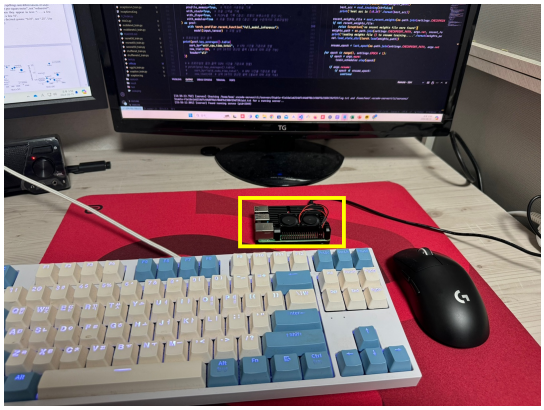


图 1. 在嵌入式系统上进行的实验环境

C. 推理时间和乘加运算次数调查

本研究使用两种库来测量推理时间和乘加运算次数。使用 *torch.profile* 测量各模型在嵌入式系统设备上的推理时间，而*thop.profile*则用于根据输入计算乘加运算次数。计算方法如下 [3]:

卷积层:

$$\text{MACs} = 2 \times (C_{\text{in}} \times H_{\text{out}} \times W_{\text{out}} \times K_H \times K_W \times C_{\text{out}}) \quad (1)$$

全连接层:

$$\text{MACs} = 2 \times (N_{\text{in}} \times N_{\text{out}}) \quad (2)$$

其中，C_{in} 表示输入通道数，H_{out} 和 W_{out} 分别表示输出特征图的高度和宽度。K_H 和 K_W 分别表示卷积核的高度和宽度，C_{out} 表示输出通道数。对于全连接层，N_{in} 表示输入神经元数，N_{out} 表示输出神经元数。

III. 结果

模型参数数量、乘加运算次数、类别平均准确率和推理时间汇总于图 2 中，并给出了皮尔逊相关系数 ρ。请注意，圆圈的大小表示模型大小。模型准确率与模型大小之间的关系，以及模型大小与计算量之间的关系，均符合深度学习范式的直觉。模型准确率与推理时间之间没有相关性，这一点可以理解，因为计算量增加并不总能带来性能提升。然而，推理时间与计算量之间缺乏相关性，值得进一步研究。

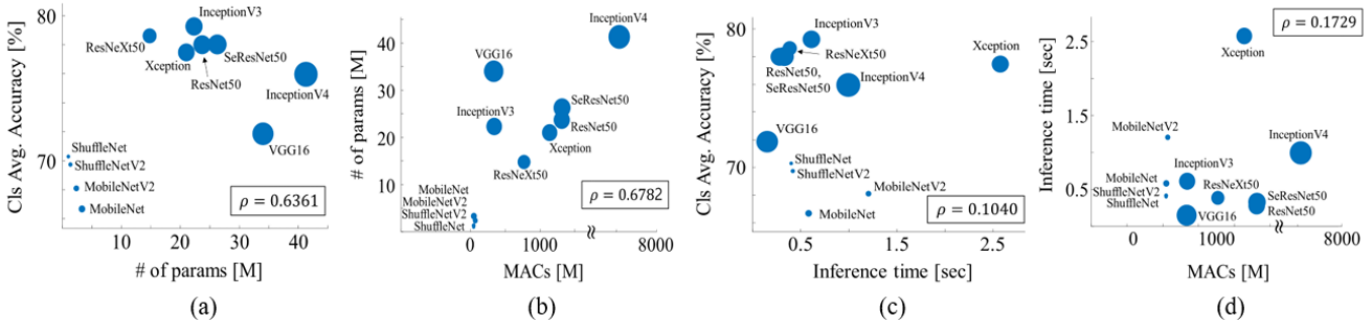


图 2. 10 个深度模型的模型准确率与计算速度分布及皮尔逊相关系数：(a) 准确率与大小，(b) 大小与计算量，(c) 准确率与时间，(d) 时间与计算量

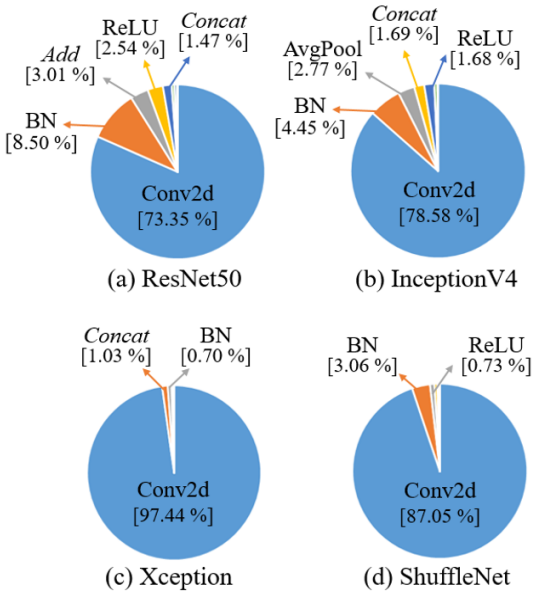


图 3. 模型推理过程中计算组件的占比。

为进行更详细的分析，我们选取了四种模型：ResNet50、InceptionV4、Xception 和 ShuffleNet。它们的乘加运算次数和推理时间存在显著差异。随后，我们将推理时间分解为各项操作，汇总结果如图 3 所示。四种模型的推理时间中，Conv2D 操作均占 70% 以上，其次是批量归一化（BN）、ReLU 激活、拼接（Concat）等。需要注意的是，拼接 和 加法 操作都是在张量之间而非模型组件之间执行的，因此不计入乘加运算次数计算。

为进行更详细的分析，表 II 总结了 Conv2d 操作期间调用的各函数的计算时间。Conv2d 操作包含多种函数，包括卷积、切片、窄化、空和视图。具体而言，切片和窄化用于从多维张量中提取或排除特定维度的数据。空函数为卷积运算分配所需的内存存储，而视图则在不增加额外内存存储的情况下修改张量的形状。所有这些操作都针对张量，因此不计入乘加运算次数计算。

就计算时间而言，卷积运算占 ResNet50 和 InceptionV4 总运算时间的 97% 以上，而其他模型中卷积运算所占比例相对较低。值得注意的是，对于 Xception，只有约 50% 的计算量发生在卷积层中。这一差异凸显了 Xception 模型的乘加运算量与其实际计算时间之间的差别。对于服务器等高性能设备，此类张量运算通常可以在不产生明显延迟的情况下完成（请参见表 *III*）。然而，必须认识到，在硬件资源有限的设备上，此类张量运算所带来的时间延迟或功耗可能会产生重大影响。

表 II 二维卷积运算中各组件的计算时间 [毫秒]

	Ops.	ResNet50	InceptionV4	Xception	ShuffleNet
	卷积	200.67	766.48	1269	190.65
	切片*	0	0.99	389.73	53.53
	窄化*	0	1.38	602.67	83.70
	空*	3.05	10.17	182.43	15.60
	视图*	0.66	1.53	47.24	5.27
	etc	2.17	1.44	18.9	0.78
总计		206.55	781.99	2510	349.53

* 张量运算

IV. 结论

本研究考察了各种深度学习模型中推理时间与计算复杂度之间的关系，揭示了乘加运算次数（MACs，乘加运算）作为性能指标的局限性。虽然乘加运算次数主要关注模型组件中的运算，例如卷积、矩阵乘法和激活函数，但往往无法计入额外的计算成本，例如张量切片操作和内存分配，而这些成本可能会显著影响实际推理时间。这些辅助运算未纳入乘加运算次数指标，但仍可能影响

表 III 附录 A. 不同设备上二维卷积运算的计算组件：在 24 核 CPU @ 3.0 GHz、125 GB 内存或 GPU @ 8192 个处理单元、24 GB GPU 显存上执行。

	[ms]	InceptionV4		Xception	
	Ops.	**本地 PC**	**本地 PC-GPU**	**本地 PC**	**本地 PC-GPU**
	卷积	34.17	7.4	12.83	2.23
	CPU：切片*	0.07	0.81	0	0.40
	CPU：窄化*	0.13	0.06	0	0.005
	空*	0.24	0.82	0.08	0.33
	视图*	0.007	0.01	0.009	0.007
	etc	1.25	0.3	0.93	0.22
总计		36.14	9.4	13.85	3.19

* 张量运算

计算过程中的显著延迟。例如，尽管 Xception 的乘加运算次数相对少于 InceptionV4，但切片操作等运算所占比例较高，可能导致推理时间更长。在资源受限的嵌入式系统中，这种低效问题会更加突出，对性能的影响也更为显著。因此，为这类系统设计深度学习模型时，仅考虑层配置是不够的；还必须将其他会产生计算开销的运算纳入考量，才能在嵌入式系统设备上实现最佳性能。

参考文献

[1] Ashish Vaswani，《注意力就是一切》，*arXiv* 预印本 *arXiv:1706.03762*，2017年。[2] Kaiming He、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun，《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，2016年，第770–778页。[3] Andrew G. Howard、Menglong Zhu、Bo Chen、Dmitry Kalenichenko、Weijun Wang、Tobias Weyand、Marco Andreetto 和 Hartwig Adam，《MobileNets：面向移动视觉应用的高效卷积神经网络》，2017年4月。[4] Mingxing Tan 和 Quoc Le，《EfficientNet：重新思考卷积神经网络的模型缩放》，载于国际机器学习会议。PMLR，2019年，第6105–6114页。[5] K. Simonyan 和 A. Zisserman，《用于大规模图像识别的超深卷积网络》，*arXiv* 预印本 *arXiv:1409.1556*，2014年。[6] Christian Szegedy、Vincent Vanhoucke、Sergey Ioffe、Jon Shlens 和 Zbigniew Wojna，《重新思考用于计算机视觉的 Inception 架构》，载于*IEEE* 计算机视觉与模式识别会议论文集，2016年，第2818–2826页。[7] Christian Szegedy、Sergey Ioffe、Vincent Vanhoucke 和 Alexander Alemi，《Inception-v4、Inception-ResNet 及残差连接对学习的影响》，载于人工智能促进协会人工智能会议论文集，2017年，第31卷。[8] Jie Hu、Li Shen 和 Gang Sun，《压缩与激励网络》，载于*IEEE* 计算机视觉与模式识别会议论文集，2018年，第7132–7141页。[9] Franc. ois Chollet，《Xception：采用深度可分离卷积的深度学习》，载于 *IEEE* 计算机视觉与模式识别会议论文集，2017年，第1251–1258页。

[10] Mark Sandler、Andrew Howard、Menglong Zhu、Andrey Zhmoginov 和 Liang-Chieh Chen，《MobileNetV2：倒残差与线性瓶颈》，载于*IEEE* 计算机视觉与模式识别会议论文集，2018年，第4510–4520页。

[11] Xiangyu Zhang、周新宇、林梦晓和Jian Sun，《ShuffleNet：一种面向移动设备的极高效卷积神经网络》，载于*IEEE*计算机视觉与模式识别会议论文集，2018年，第6848–6856页。

[12] 马宁宁、Xiangyu Zhang、Jian Sun、Hai《ShuffleNet V2：高效CNN架构设计的实用指南》，载于欧洲计算机视觉会议（*ECCV*）论文集，2018年，第116–131页。