arXiv:2301.10999v1 [cs.LG] 26 Jan 2023

PERFSAGE：面向边缘设备上任意深度学习模型的广义推理性能预测器

Yuji Chai [1] **Devashree Tripathy** [1] **Chuteng Zhou** [2] **Dibakar Gope** [2] **Igor Fedorov** [2] **Ramon Matas** [2] **David Brooks** [1] **Gu-Yeon Wei** [1] **Paul Whatmough** [1 2]

摘要

对于目标硬件平台上的任意深度神经网络（DNN），准确预测其推理性能指标（如延迟、功耗和内存占用）的能力，对于深度神经网络模型的设计至关重要。这种能力对于面向特定硬件部署平台的实用深度神经网络的（手动或自动）设计、优化和部署也至关重要。遗憾的是，使用模拟器（若有）评估这些指标耗时较长，而且通常需要在目标硬件上进行测量。

本文介绍了`PerfSAGE`，这是一种新颖的图神经网络（GNN），可预测任意深度神经网络 TFLite 图（TFL，2017年）上的推理延迟、能耗和内存占用。相比之下，以往发表的性能预测器只能预测延迟，且仅适用于预定义的构建规则或搜索空间。本文还介绍了 `EdgeDLPerf` 数据集，其中包含从四个任务搜索空间中随机抽取的 134,912 个深度神经网络，并标注了来自三个边缘硬件平台的推理性能指标。我们利用该数据集训练 `PerfSAGE`，并给出实验结果，表明其预测准确率达到最先进水平，在所有目标平台和模型搜索空间上的平均绝对百分比误差为 <5%。这些结果表明：(1) 性能优于以往最先进的基于图神经网络的预测器（Dudziak 等人，2020年）；(2) 能够准确预测加速器上的性能（弥补了非基于图神经网络的预测器的不足（Zhang 等人，2021年））；(3) 无需修改特征提取器，即可对任意输入图进行预测。

1 引言

深度神经网络（DNN）（Schmidhuber，2015）是计算机视觉、自然语言处理等领域众多任务的最先进方法。随着其日益普及，深度神经网络推理工作负载已常规部署于各种硬件平台，范围从数据中心的仓库规模计算机到消费级边缘设备中的嵌入式微控制器。这些不同的硬件平台各自带来各种实际部署限制，但边缘计算平台上的限制往往最为严格，因为推理延迟、能耗和内存通常都受到严苛约束（Fedorov 等人，2019b；Banbury 等人，2021年）。因此，面向边缘平台的应用开发者不仅需要考虑深度神经网络准确率，还必须满足推理延迟、能耗和内存方面的限制。

深度神经网络架构通常通过反复试验来开发，以找到能够在目标任务上实现高准确率的网络（以及训练方案）。当推理性能也至关重要时，开发过程会变得更加复杂，此时需要在准确率、延迟、能耗和内存占用之间进行复杂的权衡。这些指标在概念上存在取舍；例如，较大的模型通常能实现更高的准确率，但需要更多内存，推理延迟也更长。然而在实际中，要满足所有硬件约束，同时最大限度地提高准确率，无疑是一项极具挑战性的超参数优化任务。为应对这一挑战，人们广泛提出了所谓的神经架构搜索（NAS）技术，通过搜索算法自动发现能够满足这些相互交叠约束的深度神经网络（Zoph 和 Le，2016年；Cai 等人，2019年；Liu 等人，2018年；White 等人，2021年；Fedorov 等人，2019a）。

无论采用人工方法还是神经架构搜索（NAS）方法来开发深度神经网络工作负载，我们都需要能够在目标硬件平台上快速评估推理性能指标。虽然可以直接在目标硬件平台上测量深度神经网络推理运行时指标，但这存在诸多限制。首先，这要求能够方便地访问硬件，而这可能无法实现。即使可以使用硬件，有些指标（例如

[1]约翰·A·保尔森工程与应用科学学院，哈佛大学，美国马萨诸塞州波士顿 [2]ARM 公司，美国马萨诸塞州波士顿。联系作者：Yuji Chai <yuc927@g.harvard.edu>。

预印本，作者版权所有，2022年。

能耗，通常也很难测量。其次，在神经架构搜索（NAS）场景中，通常需要推理模型本身具备可微分性（Cai 等人，2018年）。NAS 还可能要求大幅提升性能测量的吞吐量，这或许需要数量多到难以承受的物理硬件板卡。最后，在典型的边缘 SoC 平台上，通常需要考虑多个异构计算目标，例如 CPU、图形处理器（Huang 等人，2008）和 NPU（Esmaeilzadeh 等人，2012）。要找到最适合部署的目标，可能需要在多个目标上进行基准测试，这会变得缓慢而费力。基于基准测试数据训练的学习型推理性能预测器能够克服这些限制，快速且可扩展地估算性能。

理想的推理性能预测器应具备以下特征：它应能很好地泛化到各种输入深度神经网络图，以应对网络连接结构和算子不可避免的演变。它不仅应预测延迟，还应预测能耗和内存占用[1]。单一的预测器模型架构在使用来自多种硬件目标的数据进行训练时，应能达到较高的准确率。迄今为止，基于学习的深度神经网络推理性能预测器日益受到研究者关注。已发表的研究（Zhang 等人，2021年；Wen 等人，2020年；Dudziak 等人，2020年；Kaufman 等人，2021年；Gao 等人，2021年）证明了这种方法的可行性，但尚未实现所有目标。

本文介绍了PerfSAGE，一种新颖的、基于图神经网络（GNN）的深度神经网络推理性能预测模型（图1）。图神经网络特征提取器的主要优势在于能够直接处理任意深度神经网络图，并很好地泛化到新的图连接结构。算子级聚合函数可通过训练获得，无需采用以往工作中使用的任何手动调优聚合方法（Wen 等人，2020年；Dong 等人，2018年），这些方法开发耗时且会限制泛化能力。PerfSAGEPerfSAGE 中的图神经网络模型甚至能够准确刻画复杂的图级交互，而这些交互决定了加速器性能。

学习型性能预测器需要数据 来训练并评估准确率。我们收集了一个名为 EdgeDLPerf的开源数据集，使我们能够在不同的深度神经网络搜索空间和硬件目标平台上训练和评估性能预测器。这与先前使用 NASBench201（Dong 和 Yang，2020年）的工作形成对比；后者仅针对图像分类任务考虑卷积神经网络（CNN）（Krizhevsky 等人，2012a）。EdgeDLPerf 包含从四种不同深度神经网络设计空间（第 3 节）中随机采样的模型，用于典型的边缘感知任务；每个模型都标注了在三种常见硬件平台上基准测试得到的推理性能指标。在扩展多样性方面，除

[1]内存占用是嵌入式微控制器上微型机器学习应用的一项重要考量（Fedorov 等人，2019b）

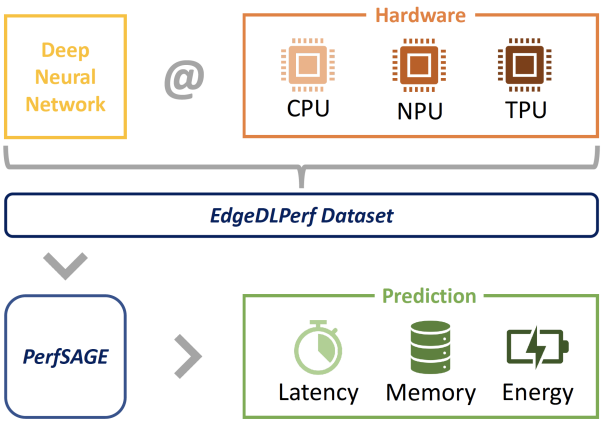


图 *1.* PerfSAGE 和 EdgeDLP 概览 erf.

NASBench201 外，EdgeDLPerf 中的模型规模从 7 个算子到 1.2K 个算子不等，层数、算子类型和连接关系也各不相同。我们还纳入了近期的视觉 Transformer（Xiao 等人，2021年）模型。

我们的实验结果（第 5 节）表明，PerfSAGE不仅在针对四项任务分别训练时能够达到最先进的准确率（<4.73% 平均绝对百分比误差），还可以通过训练单个模型，在全部四个设计空间中均实现较高准确率（<5.90% 平均绝对百分比误差）。其泛化能力源于类别特征编码架构、混合损失和数据集上采样技术（第 4 节）。相比之下，我们表明，先前的性能预测器在我们更多样化的数据集上的表现并不理想。我们在下文进一步总结本文的贡献：

1.**无需特征提取器调优，即可准确预测任意深度神经网络的性能**。输入的深度神经网络不受特定深度神经网络架构构建规则的限制（参见 Eagle（Dudziak 等人，2020年）），也无需调整预测器模型设计（参见nn-Meter（Gao 等人，2021年）。

2. **同一架构无需任何修改，即可训练用于预测推理延迟、能耗和内存**。尽管先前工作仅预测延迟，但能耗和内存对于边缘设备部署同样至关重要。

3. **在来自各种硬件平台的所有预测目标上实现最先进的准确率（<5% 平均绝对百分比误差）**。我们表明，PerfSAGE 在所有预测目标上的表现均优于 Eagle（Dudziak 等人，2020年）和 nn-Meter（Gao 等人，2021年），同时还解决了 nn-Meter 的不足：其对从硬件加速器采集的推理性能目标预测效果不佳。

4. **一个用于评估边缘平台上深度神经网络推理性能预测器的开源数据集**。

EdgeDLPerf 包含来自四种不同深度神经网络骨干网络的 ∼135K 个模型，并标注了它们在三种不同硬件平台上的基准性能。

2 背景与相关工作

2.1 图神经网络

图神经网络（GNN）（Zhou 等人，2020年）擅长处理需要从图结构中提取上下文信息的任务，例如交通预测、物理仿真和推荐系统。图上的预测任务分为三类：图级、节点级和边级。图级任务预测整个图的属性；节点级任务预测图中每个节点的属性；边级任务则预测图中边的属性或是否存在。图神经网络能够解决这三个层级的预测问题。

其他深度神经网络模型，例如卷积神经网络和循环神经网络（RNN）（You 等人，2018年），也可以应用于图。然而，它们在处理节点邻居分布不均所导致的不规则图时效率低下。图卷积网络（GCN）（Kipf 和 Welling，2016年）和图注意力网络（GAT）（Velickovic 等人，2017年；Goodfellow 等人，2014）是图神经网络家族中常见的模型，过去曾用于运行时性能预测。为避免局限于特定图层，我们采用 Spektral GNN（You 等人，2020年），它提供模块化的图神经网络实现和标准化评估，便于使用不同的图神经网络层。

DNN 推理性能预测是一种广义的图级预测。它首先对每个节点进行嵌入，然后聚合这些嵌入以生成最终的通用预测。在深度神经网络模型中，隐藏层或算子表示节点，而两个层或算子之间的中间张量表示边。为降低预测器的复杂度，PerfSAGE 将边特征编码为每个节点的附加特征。如第4节所述，我们的边聚合器在同一个深度神经网络模型中使用一组相同的参数，将相邻边组合成节点嵌入。通过聚合所有节点嵌入，生成最终的图级嵌入。

2.2 推理性能预测

DNN 推理性能预测是一个活跃的研究领域。研究人员近期重点关注学习型性能预测器的应用；与传统方法相比，这些预测器具有更高的准确率和灵活性。这些研究不仅预测深度学习模型的运行时性能，也预测其他类型软件的运行时性能。例如，DeepPerf（Ha 和 Zhang，2019年）主要用于分析 C 程序的运行时性能，以便提升性能。已有一些与本文相似的研究。我们将重点讨论深度学习模型运行时性能预测方面的相关工作。表1 列出了密切相关的先前研究。

Gao 等人（2021）提出了 DNNPerf，这是一种基于自定义 GAT 节点—边编码器的预测器，用于预测图形处理器的训练时间和内存消耗。它支持多种框架和设备选项。该预测器使用专有数据集进行训练，该数据集包括 LeNet（LeCun 等人，1998）、ResNet-V1（He 等人，2016年）、Inception-V3（Szegedy 等人，2016年）、基础循环神经网络（Rumelhart 等人，1986）、长短期记忆网络（Graves，2012），并在 AlexNet（Krizhevsky 等人，2012b）、VGG（Simonyan 和 Zisserman，2014）、OverFeat（Sermanet 等人，2013）、ResNet-V2（He 等人，2016年）和门控循环单元（Cho 等人，2014）上进行测试。DNNPerf 模型主要关注云端图形处理器上的训练。我们的工作则侧重于预测多种边缘设备上的推理性能。由于篇幅限制，表 1 中未列出 DNNPerf。

LPM-TPU Kaufman 等人（2021年）提出了一种学习型性能模型，用于预测 TPU 上 XLA 张量程序的性能（XLA，2021年）。XLA 程序的计算图被分解为更小的子图，并针对不同的优化任务（如分块大小选择和算子融合）分别训练模型实例。对于与训练集具有一定相似性的程序，该模型具有良好的泛化能力。不过，该模型需要以编译后的 XLA 格式模型作为输入，而各种硬件平台的用户难以获取这种输入。相比之下，PerfSAGE直接以保存的模型为输入，并且在多个平台上都表现稳定。

NP-NAS Wen 等人（2020年）介绍了一种使用 top-K 策略寻找准确率较高的网络候选项的方法。他们提出的神经预测器能够很好地泛化到 NASBench-101（Ying 等人，2019年）中的模型。不过，他们的预测器仍依赖于一种针对明确定义的设计空间手工调优的编码方法。

BRP-NAS（Eagle） Dudziak 等人（2020年）提出了一种神经架构搜索方法，用于在延迟约束下寻找性能最佳的模型。通过利用基于 GCN 的预测器，他们缩小了搜索空间并加快了搜索过程。BRP-NAS 中使用的 Eagle 预测器（Dudziak 等人，2020年）仅支持由 NASBench201 架构构成的模型。它利用 NASBench201 的构造规则来降低输入模型架构的复杂度。如果模型架构的构造规则事先未知，准确率就会大幅下降，如表 6所示。另一方面，PerfSAGE即使不将构造规则作为输入，也能取得良好的表现。

表 1.比 PerfSAGE 与其他深度学习模型推理性能预测相关工作的比较 .

	LPM-TPU (Kaufman 等人，2021年)	NP-NAS (Wen 等人，2020年)	BRP-NAS (Eagle) (Dudziak 等人，2020年)	nn-Meter (Zhang 等人，2021年)	PerfSAGE (本研究)
硬件目标平台	TPU	–	CPU、图形处理器、边缘GPU	CPU、图形处理器、VPU	CPU、NPU、边缘TPU
预测性能指标	延迟	准确率	延迟、准确率排名	延迟	延迟、能耗、内存
数据集	专有	NAS-Bench	NAS-Bench	专有	EdgeDLPerf
深度神经网络架构	CNN, LSTM, RNN	CNN	CNN	CNN	CNN, Transformer
任意 DNN 输入，无需更改预测器	✓			✓ （部分）	✓
在多种硬件目标平台上准确率良好					✓
预测不同指标，且／相同的预测器设计			✓		✓

nn-Meter Zhang 等人（2021年）和**NetAdapt** Yang 等人（2018年）都通过估算 DNN 在边缘设备上各层的延迟，再将所有层的延迟相加，来预测其总延迟。NetAdapt 仅支持具有明确定义架构规则的模型。虽然 nn-Meter 声称支持任意卷积神经网络模型，但根据我们的测试结果，其融合规则仍对输入模型架构有所限制。此外，这些方法不易移植到不同的硬件平台。例如，对于从深度学习加速器收集的数据，nn-Meter 预测器的性能会大幅下降。正如第 PerfSAGE 节所讨论的，5.3已克服了这一缺陷。

3 EdgeDLPerf 数据集

为了涵盖现代深度学习模型的多样性，我们收集了一个用于边缘部署的基准数据集（EdgeDLPerf）。该数据集包含一系列深度学习模型，以及这些模型在各种边缘硬件平台上测得的运行时性能指标。这些模型主要在两个方面各不相同。首先，EdgeDLPerf 涵盖了许多不同应用领域的模型。第 3.1节介绍了 EdgeDLPerf 如何通过采样过程生成，并进一步说明了如何为不同应用构建设计空间（第 3.2节）。其次，EdgeDLPerf包含来自各种硬件平台基准测试运行的多样化预测标签。硬件平台和运行时性能指标的多样性将在第 3.3节进一步讨论，体现了PerfSAGE的泛化能力。基准测试策略将在第 3.4节讨论。因此，与基于 NASBench-101 （ Ying 等人，2019 年）和 NASBench- 201（Dong 和 Yang，2020年）构建的数据集相比，EdgeDLPerf 涵盖了更多应用领域（例如关键词识别）和新颖的模型架构（例如视觉 Transformer）。它还包含作为标签的多目标硬件性能指标，并采用可用于推理的 TFLite 模型格式。

3.1 从设计空间中采样 RandomModels

我们利用可微神经架构搜索（NAS）（Fedorov 等人，2022）中的搜索空间构建和采样机制，对构成 EdgeDLPerf的模型进行采样。在 NAS 中，会定义一个超网模型，其中包含许多可搜索的网络架构选项。每个选项都关联一个由其底层概率定义的分类随机变量。从这些选项对应的随机变量中采样，可得到超网的一个子图，即从搜索空间中采样得到的一个模型。在 NAS 中，通常使用 Gumbel-Softmax 类型的可微采样器进行采样（Jang 等人，2016年）。在本研究中，我们的目标是从搜索空间中生成大量不同的模型。因此，我们采用均匀采样策略，直接从均匀分类随机变量中采样。此外，我们假设所有选项彼此独立。采样得到的模型图会通过 TFLite 转换器转换为 TFLite 模型格式。由于 INT8 是最适用于边缘部署的格式，因此我们使用 TFLite 训练后量化工具，将所有模型的权重和激活值都量化为 INT8。模型权重未经训练，采用随机初始化。采样得到的 TFLite 模型被划分为80% 训练数据和20% 测试数据。

3.2 边缘深度学习的设计空间

近年来，深度学习不断发展，涌现出许多不同的模型架构。架构差异可能源于应用领域、准确率要求、延迟要求等方面的不同。为了涵盖广泛的应用及其相关模型，我们选取了四种不同的模型设计空间，分别对应四种不同的任务，并使用第3.1节中所述的采样方法，从每种设计空间中采样了一组模型。这些模型共同构成了EdgeDLPerf 数据集。

- **CNN-CIFAR10** 卷积神经网络 de-

表 2. EdgeDLPerf 数据集特征及各设计空间中所用算子类型概述。

	CNN-CIFAR10	*CNN-ImageNet*	*CNN-KWS*	*ViT-CIFAR100*
特征				
样本数	39,997	79,998	4,917	10,000
#算子	13-70	16-77	7-21	59-994
#张量	28-145	39-182	16-51	97-1264
大小（MB）	0.09-18.25	1.65-20.54	0.04-5.05	1.86-53.98
算子				
全连接层				✓
二维卷积	✓	✓	✓	
深度卷积	✓	✓	✓	
二维平均池化	✓	✓	✓	
PAD	✓	✓		
ADD	✓	✓		✓
MUL				✓
POW				✓
TANH				✓
Softmax				✓
重塑	✓	✓	✓	✓
转置				✓
字符串切片				✓
拼接				✓

该模型面向图像分类任务，使用 CIFAR10 数据集。CNN-CIFAR10 的设计空间采用与 C NN-ImageNet 相同的构建模块，但使用 3 个模块而非 6 个。可搜索的通道数也统一缩减至原来的三分之一，以体现 CIFAR10 是一个规模较小的数据集。因此，加上 CIFAR10 数据集的分辨率较低，该设计空间中的模型规模更小，运行速度也更快。

• **CNN-ImageNet**针对 ImageNet 数据集图像分类任务设计的卷积神经网络。对于此应用，我们采用了受 EfficientNet-EdgeTPU 启发的搜索空间（Gupta 和 Akin，2020年）。该设计空间由 6 个独立模块组成，每个模块包含 5 个节点，可从倒置瓶颈、融合 MBConv 和恒等函数中进行选择。设计空间还搜索每个节点的通道数（6 种不同取值）和卷积核尺寸（3×3 和 5×5），以及模块级别的可选残差连接。

• **CNN-KWS** 专为音频关键词检测设计、面向 Google Speech Commands 数据集的卷积神经网络。这些模型的设计空间源自Banbury 等人(2021年年使用的设计空间，并通过增加内核尺寸选项进行了扩展；该设计空间针对在商用微控制器上的边缘部署场景进行了优化。因此，这些模型的推理性能指标数值最小，也更容易出现预测错误。该设计空间还包含卷积滤波器形状非正方形的模型。

表 3. 硬件平台和预测目标。

	硬件平台	**预测目标**
CPU	四核 Cortex-A57	延迟、能耗
NPU	Arm Ethos-U NPU	延迟、SRAM 使用量
TPU	Google EdgeTPU	延迟†

† EdgeTPU 不支持能耗或内存测量。

• **ViT-CIFAR100** 用于在 CIFAR100 数据集上进行图像分类的视觉 Transformer 模型。由于该设计空间中的所有模型都采用基于 Transformer 的模型架构，其架构与其他设计空间有显著差异。这些模型的大型搜索空间涵盖了视觉 Transformer 的主要可变维度，例如嵌入维度、头数、查询/键/值维度、MLP 比率和网络深度。该空间包含大量结构各异、模型复杂度不同的 Transformer。因此，*ViT-CIFAR100* 通常是最难搜索的空间。

为了更好地表示来自不同设计空间的模型，我们用图表示每个模型。每个层或操作数都视为一个图节点，而操作数之间的每个张量都是连接两个节点的一条边。我们可以用图级拓扑结构和特征表示每个深度学习模型。表 2展示了每个设计空间中采样的模型数量、操作数数量范围、张量数量范围和参数数量范围，也展示了各设计空间中包含的不同类型操作数。尽管三个卷积神经网络设计空间（*CNN-CIFAR10*、*CNN-ImageNet*和*CNN-KWS*）具有相似的操作数类型，但其模型规模差异很大。 *ViT-CIFAR100* 展现了截然不同的操作数组合。由于采用基于 Transformer 的结构，其操作数数量也显著更多。如表 2所示，`EdgeDLPerf` 由大量面向不同应用构建的深度学习模型组成，模型图特征和操作数类型具有广泛的多样性。

3.3 边缘硬件性能标签

不同的硬件目标平台对模型连接性和算子变化的性能响应可能存在巨大差异。例如，在 CPU 上运行的模型，其延迟和能耗通常与参数数量及乘加运算次数呈现更直接的相关性。这种相关性源于 CPU 上相对有限的并行性。相反，诸如 NPU（Esmaeilzadeh 等人，2012）和 TPU（Jouppi 等人，2017年）等加速器则利用大规模

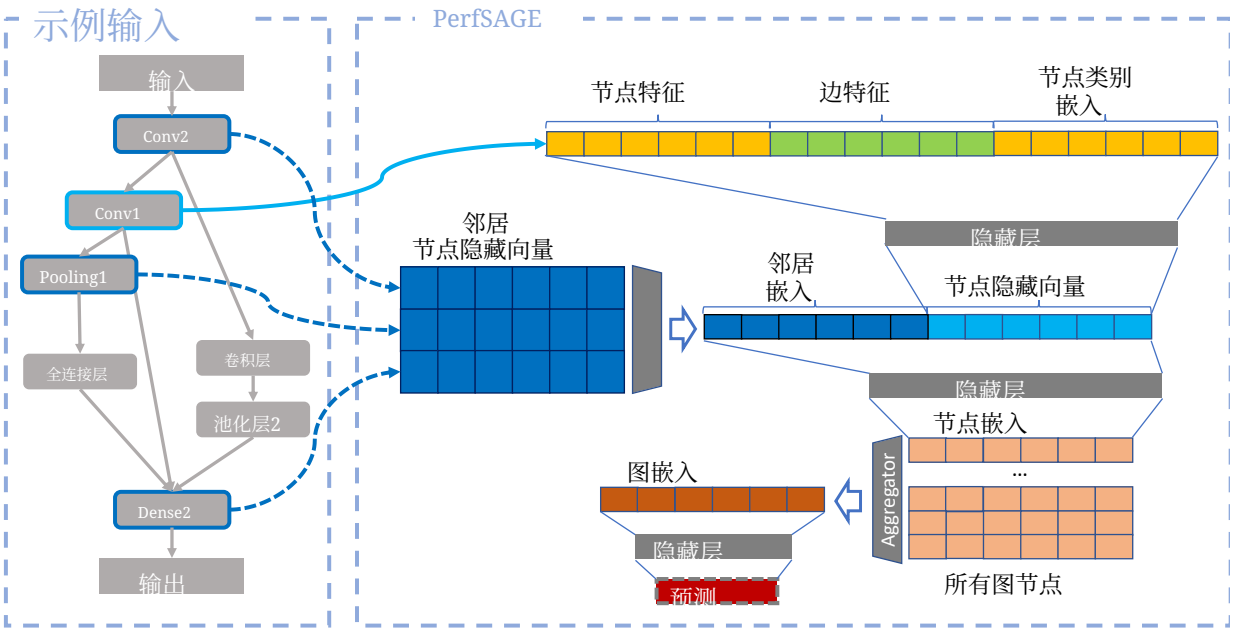


图2. PerfSAGE 的模型架构。

并行性，尤其适用于卷积等操作数。因此，NPU 和 TPU 的结果通常不会随模型的一阶维度直接扩展。为了捕捉这种多样性，我们使用三种不同的硬件目标平台为数据集中的每个模型添加标签，如表 3所示。对于 CPU 和 TPU 基准测试，我们能够将每个模型直接部署到硬件上，以测量性能数据。对于 NPU，我们从支持 Arm Ethos-N NPU 的开源模拟器 Vela[2], 中收集性能数据。使用 Vela，我们还可以测量推理期间的内存（SRAM）使用量，而通过直接部署并不总能轻松获得该数据。不过，由于算子支持有限，Vela 目前无法估算 *ViT-CIFAR100* 的延迟。

3.4 硬件基准测试方法

生成采样模型后，接下来我们为每个模型添加实测性能标签，即预测目标。标签通过在各硬件平台上测量每个模型生成（NVIDIA TX2 开发板上的 CPU（nvi，2022）、Arm Ethos-U NPU（有限公司）以及 Coral 开发板上的 Edge TPU（cor）），结果如表 3 所示。每次基准测试分为两个部分：预热阶段，循环执行推理以预热缓存；测量阶段，采集性能指标。DNN 输入（批大小为 1）和权重张量均使用随机数据。为减少测量期间的随机噪声，我们所做的所有测量结果均为多次推理运行的平均值，以降低测量期间的随机噪声。功耗测量基于推理循环运行期间多次功耗读数的平均值。所有硬件平台均以最高性能模式运行；如果观察到热节流，则会提供额外冷却。

[2]Vela 可在以下地址获取：
https://git.mlplatform.org/ml/ethos-u/ethos-u-vela.git/about/

4 PerfSAGE 设计

`PerfSAGE`（图 2），旨在从多种深度神经网络中提取特征。为此，它使用图神经网络对任意定义的输入模型图进行编码。正如第 2.1 节所述，图神经网络能够对未见过的图结构进行归纳推理。

4.1 模型架构

图 2（左）展示了一个假设的 CNN 输入图。输入图中的节点 i 表示一个算子（灰色方框，例如 *Conv1*、*Pooling1* 等），连接节点的有向边表示算子之间的张量。以算子 *Conv1*（输入图中的灰色节点）为节点 i 的示例，其对应的节点类型 c_i为二维卷积（图中未显示节点类型）。节点的输入特征由三部分组成：节点特征（黄色方框）、边特征（绿色方框）和节点类别嵌入（黄色方框）。节点特征F_{node}（i）是对应层的超参数，例如填充大小或卷积核维度。该节点的输出维度被视为边特征 F_{edge}（i）。节点类别嵌入 E_{cat}（c_i）来自一个可训练嵌入表，其中存储了

不同节点类型对应的一组参数向量。只有与当前节点类型对应的参数向量会输入节点类别嵌入向量。然后，我们拼接这三部分特征向量，并使用一个隐藏层生成下面公式（1）所描述的节点隐藏向量（浅蓝色方框）h_i^0

$$h_i^0 = \text{relu}(w_{c_i}^0 * (F_{node}(i)||F_{edge}(i)||E_{cat}(c_i)) + b_{c_i}^0). \quad (1)$$

仅考虑节点 i（即*Conv1* 节点）的特征输入是不够的，因为整个图的拓扑结构对于预测输入深度神经网络的性能同样至关重要。因此，我们的方法还会纳入直接连接的节点邻居（*Conv2*、*Pooling1*和*Dense2*）的特征信息。

以图 2 的示例为例，为实现这一点，我们通过求和核聚合节点隐藏向量（深蓝色方框）来收集邻居节点信息，从而生成邻居嵌入。将节点隐藏向量与邻居嵌入拼接后，经过八个图神经网络层处理，随后转换为节点 i 的最终节点嵌入。图 k^{th} 中的图神经网络层（如下方公式 (2) 所示）接收节点 h_i^{k-1} 的上一层隐藏向量以及 i 其邻居的隐藏向量 $\{h_j^{k-1}, j \in N(i)\}$，并生成隐藏向量 h_i^k，因此：

$$h_i^k = \text{relu}((h_i^{k-1}||\text{SUM}(\{h_j^{k-1}, j \in N(i)\})) * w^k + b^k). \quad (2)$$

为了预测整个图的属性，而不只是单个节点的属性，我们使用最大聚合核汇总所有节点的信息，从而获得最终的图嵌入向量。该向量包含每个节点及其邻居的信息。基于此图嵌入，通过 4 个全连接层生成最终预测，隐藏维度为 1024。

4.2 类别特征编码

每个算子节点最重要的特征之一是其类别或类型（例如，二维卷积、全连接、加法等）。类别不同的两个节点不仅可能具有完全不同类型的特征，同一特征对它们也可能具有截然不同的含义。例如，考虑节点的输出维度。具有相同输出维度的二维卷积层与加法算子，其层属性会大相径庭。它们的计算操作数或硬件资源使用情况也会完全不同。如果它们仍然共享同一组参数，那么针对不同节点类型进行训练，对于节点输出维度参数而言将适得其反。随着节点类型变化，其训练梯度的方向也会截然不同。因此，我们提出类别特征编码（CFE），其中包含一个根据每个节点类别建立的参数查找表，用于节点。正如公式（1）所示，节点 i 的权重 $w_{c_i}^0$ 和偏置 $b_{c_i}^0$ 取决于其类型 c_i。每个节点特征由算子的超参数及其算子类型组成。借助 CFE，二维卷积和加法的节点输出维度会被视为两种不同类型的参数，而非同一类型。此基于查找表的方法确保不同类型的节点可以为其输入特征使用完全不同的权重集，即便它们共享某些特征。总之，CFE 使 PerfSAGE 能够采用不同数量的特征，也能够针对预测目标采用同一特征但赋予其不同作用。

4.3 混合损失函数

如表 2 所示，来自`EdgeDLPerf` 的模型种类繁多（从极小到极大），导致运行时性能指标回归问题的数值范围很广。最小预测目标的数值比最大预测目标高出一个数量级，例如，延迟范围为 2ms 至 34ms（图3）。由于大多数指标的数值范围很广，百分比误差是大多数潜在用户首要关注的问题。如果始终将绝对误差作为优化目标，那么对于 30ms 的目标，1ms 的绝对误差仅意味着 3% 的误差率；但对于 3ms 的目标，则意味着 33% 的误差率。因此，采用平均绝对百分比误差（MAPE）作为通用的优化和比较目标更为合理，并且能在整个预测目标范围内实现相对较低的误差率。然而，当预测目标的数值较大时，仅使用 MAPE 不足以限制绝对误差。因此，我们还使用均方误差（MSE）作为补充损失函数，在预测值较大时限制误差。混合损失函数如公式（4）所示。它使用一个额外的超参数 α_{MSE}，经过调优后，可确保 MAPE 和 MSE 两部分的损失贡献相等，从而获得最佳结果。混合损失定义如下：

$$\text{MSE}(X,Y) = \sqrt{(X-Y)^2}, \text{MAPE}(X,Y) = \frac{|X-Y|}{Y}, \quad (3)$$

$$\text{Loss}(X,Y) = \text{MAPE}(X,Y) + \alpha_{MSE} * \text{MSE}(X,Y). \quad (4)$$

4.4 数据集上采样

不同设计空间的数据点数量分布不均是很常见的。数据点较多的设计空间可能会对泛化模型和其他设计空间产生较大影响，而数据点较少的设计空间则会因数据点不足而受到影响，例如 *CNN-KWS*。为了解决这种有偏分布，我们引入了基于数据集的上采样。根据数据集的大小，我们确保较小数据集中的模型在训练过程中出现得更频繁，同时较大数据集中的模型至少出现一次。上采样比例由以下方式确定：

表4. 类别特征编码（CFE）对不同设计空间的延迟（CPU）预测平均绝对百分比误差的影响。

	CNN-CIFAR10	*CNN-ImageNet*	*CNN-KWS*	*ViT-CIFAR100*
	PerfSAGE（专用）			
不含 CFE	4.800%	1.439%	7.043%	11.471%
使用 CFE	**2.314%**	**1.098%**	**3.722%**	**4.451%**
	PerfSAGE（广义）			
不含 CFE	4.136%	4.342%	5.969%	6.505%
使用 CFE	**2.617%**	**1.263%**	**5.847%**	**4.798%**

通过对较小的数据集进行上采样，使其至少占训练集中所有数据点的 10%。这确保了来自较小设计空间的模型在训练过程中产生更大的影响。因此，较大设计空间中的模型不会因其占主导地位而被忽略。

5 评估结果

本节介绍了评估 `PerfSAGE` 的实验。我们训练了多个模型，并通过消融实验评估不同训练方法的效果（第 5.2 节）。我们还与 Eagle（Dudziak 等人，2020年）和 nn-Meter（Zhang 等人，2021年）进行了定量比较。所有对比实验均使用 `EdgeDLPerf` 数据集（第 3.2 节）。

5.1 图神经网络训练

训练时，我们随机选取 80% 的数据点，其余 20% 专门用于测试。所有实验均使用相同的训练集和测试集划分。所有结果均采用平均绝对百分比误差（MAPE）进行比较，该指标可用于在所有数值范围内进行总体比较。在比较中，我们采用两种不同的方法训练模型：专用和泛化。在专用模式下，预测器模型每次针对单个设计空间进行训练，最终得到四个不同的模型。在泛化模式下，使用单个模型同时预测任意设计空间。泛化误差率由同一个泛化模型在各个设计空间上的测试结果计算得出。每个模型均在 NVIDIA A100 上训练，时间预算为 8 小时。我们所有的模型都能在该时间预算内收敛。

5.2 消融研究

本节通过消融实验展示所提出的独特技术的有效性，包括类别特征编码、混合损失函数和数据集上采样。

5.2.1 类别特征编码（*CFE*）

CFE 有助于为具有相似节点特征的不同类型节点提供不同的参数。不同的参数能够区分不同节点，并产生不同的预测结果，即使它们的节点特征相似。我们针对专用模型和以延迟（CPU）为预测目标的泛化模型测试了 CFE。表4展示了两种场景下的结果，可以看出 CFE 对专用模型和泛化模型都至关重要。对于专用模型，*ViT-CIFAR100* 因 CFE 而取得了最显著的提升。CFE 对于理解不同操作数即使具有相似的超参数，也可能拥有完全不同的属性至关重要。对于其他三个 CNN 设计空间，延迟通常由其卷积层主导。因此，即使参数无法区分不同类型的算子，性能也不会完全下降。*ViT-CIFAR100* 中的模型没有占主导地位的算子类型，因此提升幅度更大。在泛化模型的对比中，未使用 CFE 的测试表明，很难理解不同层类型之间的差异。这使得*CNN-ImageNet*上的性能进一步下降，尽管它在专用测试中表现良好。专用模型与泛化模型的这一对比表明，CFE 至关重要，尤其是在模型设计空间日益多样化的情况下。

5.2.2 混合损失函数

为了了解引入混合损失函数如何改善最终测试误差，我们比较了在 *CNN-ImageNet* 设计空间中使用 CPU 进行延迟预测时的测试 MAPE。在图 3中，我们针对整个数据集的不同延迟范围生成了预测错误率。图分为两部分。左图显示仅使用简单 MAPE损失函数时的测试 MAPE，右图则显示使用混合损失函数的结果。图的第一行展示所有模型延迟的直方图，接下来两行展示 MAPE 以及整个延迟范围内的 MAPE。若不使用混合损失函数，延迟范围较大时误差会明显上升，而且在数据点较少的延迟区间中，MAPE 的表现也不理想。较高延迟范围内的这类误差源于 MAPE：当延迟值较大时，MAPE 难以轻易限制误差。混合损失可以通过对绝对误差而非百分比误差施加额外限制来轻松解决这一问题。如右图所示，MSE损失成功限制了较大延迟范围内较高的错误率。它还有助于降低模型密度低得多的延迟范围内的损失。在图 4中，我们还研究了权重因子 α_{MSE}对 MSE 损失的影响。随着权重增加，MAPE损失会下降，随后趋于平稳。如果α_{MSE}过高，则会导致一些不必要的训练

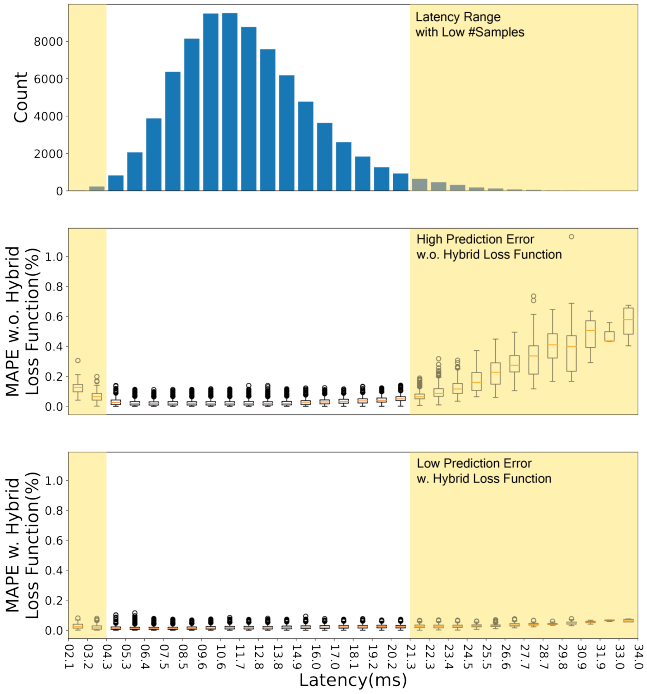


图 *3*。按预测目标数值排序的不同区间中平均绝对百分比误差（MAPE）的详细箱线图。所报告的结果来自对 *CNN-ImageNet*的 CPU 推理延迟预测。

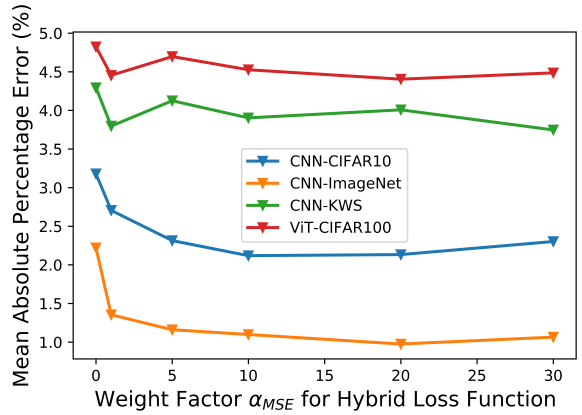


图*4*。 混合损失函数的权重因子对 EdgeDLPerf 数据集上 CPU 延迟预测 MAPE 的影响。

噪声。通过实验，我们将 α_{MSE} 设为 10，以获得最优训练配置。

5.2.3 数据集上采样

数据集上采样对于PerfSAGE 在模型密度有偏的数据集上的泛化能力至关重要。我们比较了上采样前后的模型性能，其他所有超参数和训练方法均保持不变。我们将 *CNN-KWS* 上采样 3×，并将*ViT-CIFAR10* 上采样 2×，使其达到全部训练样本的 10%。从表5可以看出，上采样能够显著提升

表 *5*。数据集上采样（DUS）对不同设计空间中 CPU 延迟预测 MAPE 的影响。

	PerfSAGE（广义）			
	CNN-CIFAR10	***CNN-ImageNet***	***CNN-KWS***	***ViT-CIFAR100***
不使用 DUS	3.080%	1.592%	8.475%	6.710%
使用 DUS	**2.617%**	**1.263%**	**5.847%**	**4.798%**

在有偏数据集分布下的性能。数据点数量较少的数据集上的提升最为显著。同时，上采样不会损害模型数量较多的数据集上的性能。

5.3 总体结果与相关工作比较

表6 展示了PerfSAGE 在我们的EdgeDLPerf数据集上的结果。表格上半部分展示了针对各个设计空间的专用模型的性能，下半部分则展示了广义模型的性能。广义模型的表现略逊于专用模型。我们的模型在不同硬件平台上的所有可用预测目标上均表现稳定。与 nn-Meter 上的研究发现相反（Zhang 等人，2021年），PerfSAGE 的预测性能并未明显下降，即使使用专用硬件加速器（如 NPU 或 TPU）也是如此。nn-Meter 的预测性能下降很可能是由于其聚合函数假设整个模型的延迟等于各个内核延迟之和。这一假设对于不支持更高级别并行性的 CPU 等硬件或许适用。然而，在使用深度学习硬件加速器进行推理时，这一假设并不成立，因此其聚合方法会导致预测准确率下降。

我们还将 nn-Meter 方法应用于我们的EdgeDLPerf CPU 数据集（见表 7）。nn-Meter 主要用于延迟预测，不适用于其他推理性能指标。它也无法完全支持来自 *ViT-CIFAR100* 设计空间的模型，因为这些模型需要为新型 Transformer 层实现预测器。来自*CNN-CIFAR10*、*CNN-ImageNet* 和*CNN-KWS* 的部分模型不符合 nn-Meter 的融合规则。此外，由于要求卷积核形状对称，*CNN-KWS*的预测误差远不理想。相比之下，PerfSAGE 支持EdgeDLPerf 数据集中的所有模型，且错误率更低。为了进行更全面的比较，我们调整了 Eagle 预测器（Dudziak 等人，2020年），使其能够在我们的数据集上训练。尽管 Eagle 预测器能够预测 NASBench201 设计空间中的所有模型，但随着模型设计空间和硬件平台不断增多，它的表现会受到影响

表 *6.* PerfSAGE 在预测 EdgeDLPerf 数据集中各种目标时的均方百分比误差（MAPE）。结果展示了专用模型和广义模型的表现。作为比较，我们还展示了 Eagle（Dudziak 等人，2020年）用于预测我们的 EdgeDLPerf 数据集中各种目标时的 MAPE 结果。Eagle 主要针对 NASBench201 设计，该设计空间定义明确，并具有详细的架构规则。因此，面对更加多样化的模型集合时，其表现不佳。由于泛化能力有限，此处仅展示专用模型的结果。由于 ARM Ethos-U NPU 不支持 Transformer，故未列出 NPU 上 *ViT-CIFAR100* 的延迟结果。

	延迟（CPU）	能耗（CPU）	延迟（NPU）	内存（NPU）	延迟（TPU）
PerfSAGE（专用版）					
CNN-CIFAR10	**2.314%**	**3.164%**	**2.103%**	**2.758%**	**2.757%**
CNN-ImageNet	**1.098%**	**1.131%**	**1.352%**	3.599%	2.360%
CNN-KWS	**3.722%**	4.455%	3.574%	**3.248%**	**3.055%**
ViT-CIFAR100	**4.451%**	**4.633%**	–†	**0.701%**	**4.730%**
PerfSAGE（广义）					
CNN-CIFAR10	2.617%	3.694%	3.157%	3.625%	2.816%
CNN-ImageNet	1.263%	1.474%	2.115%	**3.423%**	**1.345%**
CNN-KWS	5.847%	**4.259%**	6.390%	4.952%	5.115%
ViT-CIFAR100	4.798%	5.901%	–†	1.580%	4.830%
Eagle（专用）					
CNN-CIFAR10	41.138%	38.482%	36.388%	39.625%	85.079%
CNN-ImageNet	18.080%	16.745%	15.412%	21.908%	96.921%
CNN-KWS	36.142%	39.464%	35.131%	24.575%	57.823%
ViT-CIFAR100	46.203%	52.256%	–†	50.503%	91.550%

† Vela 缺少对 ViT 模型的算子支持，因此无法进行特征表征。

表*7*。使用 nn-Meter 预测 EdgeDLPerf 数据集中的各类目标时的均方百分比误差（平均绝对百分比误差）（Zhang 等人，2021年）。同时列出 PerfSAGE 的结果以供比较。nn-Meter 的设计主要用于延迟预测，不适用于其他推理性能指标。此外，它的灵活性不足，无法支持我们数据集中的所有模型。

	延迟（CPU）		
	nn-Meter	PerfSAGE（专用）	PerfSAGE（广义）
CNN-CIFAR10	4.371%	**2.314%**	2.617%
CNN-ImageNet	3.219%	**1.098%**	1.263%
CNN-KWS	452.694%	**3.722%**	5.847%
ViT-CIFAR100	–†	**4.451%**	4.798%

† nn-Meter 缺少对 ViT 模型的算子支持，因此无法进行表征。

进行了评估，如表 6所示。Eagle 在 NAS-Bench201 上表现良好，部分归功于其对模型架构构建规则的先验知识。一旦预测器无法获取这些知识（如我们的基准测试中所示），其准确率便会显著下降。 PerfSAGE 无需这些信息，却仍能在多样化的模型空间中很好地泛化。与最先进的方法相比，PerfSAGE 在更多样化的设计空间和预测目标上提供了更高的预测准确率。

5.4 未来工作

学习型性能模型是计算机体系结构建模领域一个非常令人振奋的方向，而 DNN 推理性能只是其中一个狭窄的方面。学习型性能模型是计算机体系结构建模领域一个很有前景的方向。我们希望，开源的 EdgeDLPerf 数据集能够为未来改进我们自己的 PerfSAGE 预测器提供一个平台。

具体而言，我们设想可以在三个主要方面进行改进。首先，我们的模型仍依赖于生成包含大量模型的基准测试数据集进行训练。可将自监督训练应用于我们的训练过程，以减少需要进行基准测试的模型数量。其次，我们仅针对批量大小为一的推理进行预测。在实际应用中，边缘推理可能采用更大的批大小。因此，我们应研究如何将批大小作为运行时性能预测的额外信息。第三，对于编译器优化和神经网络架构搜索等用例，数据学习过程中的任何错误都会给优化过程带来一系列问题。因此，为每项预测附加一个不确定性分数会很有帮助，这样便可通过应用贝叶斯深度学习方法，避免错误预测带来的隐患。

6 结论

PerfSAGE 朝着开发理想的运行时性能预测器迈出了重要一步，在三个硬件平台上的所有预测目标上实现了$<$ 5% 的平均绝对百分比误差。与 EdgeDLPerff 数据集上的其他最先进的基于图神经网络的方法相比，其性能显著更优。

PerfSAGE 展现出卓越的泛化能力。不同于其他未采用基于图神经网络的模型的预测器，PerfSAGE 无需手动调优或实现变更，便可轻松适应不同的硬件平台和预测目标。对于配备优化内核级调度器的硬件平台，它同样具有很强的适应性。因此，PerfSAGE 克服了以往工作在预测深度学习加速器上的深度神经网络推理性能时准确率下降的缺点。

参考文献

开发板。网址 https://coral.ai/products/dev-board/。TFLite：TensorFlow Lite。https://www.tensorflow.org/lite，2017。XLA：面向机器学习的优化编译器。https://www.tensorflow.org/xla，2021。借助 Jetson TX2 开发套件在边缘端驾驭 AI，2022年8月。网址 https://developer.nvidia.com/embedded/jetson-tx2-developer-kit。Banbury, C., Zhou, C., Fedorov, I., Matas, R., Thakker, U., Gope, D., Janapa Reddi, V., Mattina, M. 和 Whatmough, P. Micronets：用于在商用微控制器上部署微型机器学习应用的神经网络架构。《机器学习与系统》论文集，3，2021。Cai, H., Zhu, L. 和 Han, S. ProxylessNAS：在目标任务和硬件上直接进行神经架构搜索。*arXiv* 预印本 *arXiv:1812.00332*，2018。Cai, H., Zhu, L. 和 Han, S. ProxylessNAS：在目标任务和硬件上直接进行神经架构搜索。载于*第7届国际学习表征会议，ICLR 2019*，美国路易斯安那州新奥尔良，*2019*年*5*月*6*日至*9*日。OpenReview.net，2019。网址 https://openreview.net/forum?id=HylVB3AqYm。Cho, K., Van Merri¨enboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. 和 Bengio, Y. 使用循环神经网络编码器-解码器学习用于统计机器翻译的短语表征。*arXiv* 预印本 *arXiv:1406.1078*，2014。Dong, J.-D., Cheng, A.-C., Juan, D.-C., Wei, W. 和 Sun, M. PPP-Net：面向帕累托最优神经网络架构的平台感知渐进式搜索。2018。Dong, X. 和 Yang, Y. NAS-Bench-201：扩展可复现神经架构搜索的范围。*arXiv* 预印本 *arXiv:2001.00326*，2020。Dudziak, L., Chau, T., Abdelfattah, M., Lee, R., Kim, H. 和 Lane, N. BRP-NAS：使用图卷积网络进行基于预测的神经架构搜索。神经信息处理系统进展，33：10480–10490，2020。Esmaeilzadeh, H., Sampson, A., Ceze, L. 和 Burger, D. 面向通用近似程序的神经网络加速。载于*2012*年第*45*届 *IEEE/ACM* 国际微体系结构年会，第449–460页。IEEE，2012年。

Fedorov, I., Adams, R. P., Mattina, M. 和 Whatmough, P. N. 面向资源受限微控制器上卷积神经网络的 *SpArSe*：稀疏架构搜索。Curran Associates Inc.，美国纽约州 Red Hook，2019a。

Fedorov, I., Adams, R. P., Mattina, M. 和 Whatmough, P. N. SpArSe：面向资源受限微控制器上卷积神经网络的稀疏架构搜索。神经信息处理系统进展，32，2019b。

Fedorov, I., Matas, R., Tann, H., Zhou, C., Mattina, M. 和 Whatmough, P. Udc：面向可压缩微型机器学习模型的统一深度神经架构搜索。*arXiv* 预印本 *arXiv:2201.05842*，2022。

Gao, Y., Gu, X., Zhang, H., Lin, H. 和 Yang, M. 使用图神经网络预测深度学习模型的运行时性能。技术报告，技术报告 MSR-TR-2021-3。微软，2021年。

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. 和 Bengio, Y. 生成对抗网络。神经信息处理系统进展，27，2014。

Graves, A. 长短期记忆。使用循环神经网络进行监督式序列标注，第 37–45 页，2012。

Gupta, S. 和 Akin, B. 使用自动机器学习进行加速器感知神经网络设计。*arXiv* 预印本 *arXiv:2003.02838*，2020年。

Ha, H. 和 Zhang, H. DeepPerf：使用深度稀疏神经网络预测可配置软件的性能。发表于*2019*年*IEEE/ACM*第*41*届国际软件工程会议（*ICSE*），第 1095–1106 页。IEEE，2019年。

He, K.、Zhang, X.、Ren, S. 和 Sun, J. 用于图像识别的深度残差学习。发表于《*IEEE*计算机视觉与模式识别会议论文集》，第 770–778 页，2016年。

Huang, Q.、Huang, Z.、Werstein, P. 和 Purvis, M. GPUASA：通用计算资源。发表于*2008*年第九届并行与分布式计算、应用与技术国际会议，第 151–158 页，2008。doi：10.1109/PDCAT.2008.38。

Jang, E., Gu, S. 和 Poole, B. 使用 Gumbel-Softmax 进行分类重参数化。*arXiv* 预印本 *arXiv:1611.01144*，2016年。

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., 等。《数据中心内张量处理单元的性能分析》。载于第*44*届年度国际计算机体系结构研讨会论文集，第1–12页，2017年。

Kaufman, S., Phothilimthana, P., Zhou, Y., Mendis, C., Roy, S., Sabne, A., 和 Burrows, M. 一种面向张量处理单元的学习型性能模型。机器学习与系统论文集，3:387–400，2021年。Kipf, T. N. 和 Welling, M. 基于图卷积网络的半监督分类。*arXiv*预印本 *arXiv:1609.02907*，2016年。Krizhevsky, A., Sutskever, I., 和 Hinton, G. E. 使用深度卷积神经网络进行 ImageNet 分类。载于 Pereira, F., Burges, C., Bottou, L., 和 Weinberger, K.（编），神经信息处理系统进展，第25卷。Curran Associates, Inc., 2012a。网址：
h
t
t
p
s://
proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper
pdf.

Krizhevsky, A., Sutskever, I. 和 Hinton, G. E. 使用深度卷积神经网络进行 ImageNet 分类。神经信息处理系统进展，25，2012b。

LeCun, Y., Bottou, L., Bengio, Y. 和 Haffner, P. 将基于梯度的学习应用于文档识别。*IEEE* 会议录，86(11):2278–2324，1998年。

Liu, H., Simonyan, K. 和 Yang, Y. DARTS：可微架构搜索。*arXiv* 预印本 *arXiv:1806.09055*，2018年。

Arm 有限公司。Ethos-U65 机器学习处理器（NPU）。网址 https://www.arm.com/products/silicon-ip-cpu/ethos/ethos-u65。

Rumelhart, D. E., Hinton, G. E. 和 Williams, R. J. 通过反向传播误差学习表征。《自然》，323(6088):533–536, 1986。

Schmidhuber, J. 神经网络中的深度学习：综述。《神经网络》，61:85–117, 2015。

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. 和 LeCun, Y. Overfeat：利用卷积网络实现识别、定位和检测。《arXiv 预印本 arXiv:1312.6229》，2013。

Simonyan, K. 和 Zisserman, A. 用于大规模图像识别的超深卷积网络。《arXiv 预印本 arXiv:1409.1556》，2014。

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., 和 Wojna, Z. 重新思考用于计算机视觉的 Inception架构。发表于*IEEE*计算机视觉与模式识别会议论文集，第 2818–2826 页，2016年。

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., 和 Bengio, Y. 图注意力网络。统计，1050：20，2017年。

Wen, W.、Liu, H.、Chen, Y.、Li, H.、Bender, G. 和 Kin- dermans, P.-J. 用于神经架构搜索的神经预测器。载于欧洲计算机视觉会议，第660–676页。施普林格，2020年。

White, C.、Neiswanger, W. 和 Savani, Y. Bananas：利用神经架构进行贝叶斯优化，以实现神经架构搜索。载于*AAAI*人工智能会议论文集，2021年。

Xiao, T.、Singh, M.、Mintun, E.、Darrell, T.、Doll´ar, P. 和 Girshick, R. 早期卷积有助于Transformer实现更好的视觉感知。神经信息处理系统进展，34:30392–30400，2021年。Yang, T.-J.、Howard, A.、Chen, B.、Zhang, X.、Go, A.、San- dler, M.、Sze, V. 和 Adam, H. Netadapt：面向平台的神经网络适配，用于移动应用。载于欧洲计算机视觉会议（*ECCV*）论文集，2018年9月。Ying, C.、Klein, A.、Christiansen, E.、Real, E.、Murphy, K. 和 Hutter, F. Nas-bench-101：迈向可复现的神经架构搜索。载于国际机器学习会议，第7105–7114页。PMLR，2019年。You, J.、Ying, R.、Ren, X.、Hamilton, W. 和 Leskovec, J. GraphRNN：利用深度自回归模型生成逼真图。载于国际机器学习会议，第5708–5717页。PMLR，2018年。You, J.、Ying, Z. 和 Leskovec, J. 图神经网络的设计空间。神经信息处理系统进展，33:17009–17021，2020年。Zhang, L. L.、Han, S.、Wei, J.、Zheng, N.、Cao, T.、Yang, Y. 和 Liu, Y. nn-Meter：面向多样边缘设备的深度学习模型推理精确延迟预测。载于第*19*届年度移动系统、应用与服务国际会议论文集，第81–93页，2021年。Zhou, J.、Cui, G.、Hu, S.、Zhang, Z.、Yang, C.、Liu, Z.、Wang, L.、Li, C. 和 Sun, M. 图神经网络：方法与应用综述。*AI Open*，1:57–81，2020年。ISSN 2666-6510。doi：https://doi.org/10.1016/j.aiopen.2021.01.001。URL https://www.sciencedirect.com/science/article/pii/S2666651021000012。

Zoph, B. 和 Le, Q. V. 通过强化学习进行神经架构搜索，2016年。URLhttp://arxiv.org/abs/1611.01578。引用arxiv:1611.01578。