使用多目标神经结构搜索对FPGA加速器进行自动化硬件和神经网络结构协同设计

1st Philip Colangelo
英特尔PSG
美国圣何塞，philip.colangelo@intel.com

2nd Oren Segal
Hofstra University
Hempstead, USA
oren.segal@hofstra.edu

3rd Alex Speicher
美国霍夫斯特拉大学汉普斯特德分校
aspeicher1@pride.hofstra.edu

4th Martin Margala
马萨诸塞州大学洛厄尔分校
美国洛厄尔，马丁
Margala@uml.edu

2020 IEEE第十届国际消费电子会议（ICCE-Berlin）| 978-1-7281-5885-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICCE-Berlin50680.2020.9352153

摘要--
***最先进的*神经网络架构（NNAs）在硬件设计和有效实施方面具有挑战性。在过去的几年里，这导致了自动神经结构搜索（NAS）工具的研究和发展的爆炸性增长。AutoML工具现在被用来实现最先进的NNA设计，并试图对硬件使用和设计进行优化。最近在自动设计NNA方面的许多研究都集中在卷积网络和图像识别上，忽略了数据中心的工作负载中有很大一部分是通用的深度神经网络。在这项工作中，我们开发并测试了一个通用的多层感知器（MLP）流程，它可以将任意的数据集作为输入，并自动产生优化的NNA和硬件设计。我们在六个基准上测试了该流程。我们的结果显示，我们超过了目前公布的MLP准确性结果的性能，并且与基于非MLP的结果有竞争力。我们将一般和常见的GPU架构与我们可扩展的FPGA设计进行了比较，并显示我们可以为大多数数据集实现更高的效率和更高的吞吐量（每秒的输出）。对精确网络和高性能硬件的设计空间的进一步深入了解表明，通过将精度与吞吐量、网络大小与精度以及扩展到高性能设备之间的关系，共同设计的力量。**

***Index Terms-进化*算法，机器学习，FPGA，自动设计**

I. 简介和动机

优化NNAs是一个困难的过程，部分原因是存在大量的超参数组合。在组合不理想的情况下，性能会受到影响。研究表明，一个网络的参数可以直接影响该模型在部署中的准确性、吞吐量和能耗[1]。设计高性能神经网络的困难使人们最近对神经网络的自动设计和优化产生了浓厚的兴趣。现有研究的重点是优化NNA的准确性[2][3][4]，但缺乏针对网络的硬件优化。我们的重点是通过使用进化算法来搜索整个设计空间，包括NNA和可重新配置的硬件，来弥补这一差距。这种协同设计方法提供了一种解决方案，为特定的NNA模型实现了定制的硬件解决方案。

许多文章都集中在用卷积神经网络进行图像分类，部分原因是由于引入了的AlexNet在2012年和ImageNet数据库。传统的神经网络，如多层感知器（MLP）已经从聚光灯下退居幕后。然而，他们的应用仍然很有意义，大型的以数据为中心的公司，如Facebook[5][6]和Google[7]已经公布的数据显示，MLP的工作负载是其应用基础的大多数。Facebook引用MLP的任务，如确定显示哪些广告，在新闻提要中看到哪些重要的故事，以及从搜索中呈现哪些结果。Park等人在[8]中强调了这些网络的重要性和目前对标准硬件的限制，并呼吁本研究旨在解决的问题，即软件和硬件共同设计。谷歌最近公布了他们的TPU的性能分析，显示MLP占TPU工作负载的61%。我们的发现与他们的论文中提出的结果一致，指出MLP工作负载是有内存限制的。我们的协同设计过程能够设计出平衡计算和内存需求的最佳硬件结构。第四节提供了关于内存带宽考虑的见解。

MLP的核心是一般的矩阵乘法（GEMM），多年来一直存在的库，如基本线性代数子程序（BLAS），提供高度调整的特定硬件GEMM实现。标准硬件如CPU和GPU提供这些子程序，使实现高性能的MLP变得容易。但是，在这种典型的流程中仍然存在两个问题。1.在创建MLP的过程中通常不考虑目标硬件，而且
2.标准架构是通用的。通用架构不一定提供有关ANN描述的性能扩展（见第四节）。

我们的贡献是利用了FPGA器件的可重构架构，它能够适应特定的工作负载和神经网络结构。利用进化算法同时搜索MLP和目标硬件的整个设计空间，我们找到了独特的解决方案，实现了最高的精度和最佳的硬件性能。第四节提供了从多个数据集的一系列测试中得出的MLP网络和硬件配置的顶级结果。传统的

我们讨论了各种方法和我们的方法，以提供关于我们如何取得顶级结果的见解。

II. 相关工作

在过去的几十年里，自动搜索人工神经网络一直是一个持续的努力，但由于设计复杂度不断增加的深度网络的难度，自动搜索正在成为人工神经网络研究界的一个焦点[3][4][9]。自动人工神经网络架构搜索（NAS）可以使用不同的策略，如Random搜索、进化算法、强化学习（RL）、贝叶斯优化和基于梯度的方法[9]。多年来，使用进化算法（EA）来搜索高性能的架构已经得到了广泛的研究[10][11]。最近的一些结果表明，进化算法提供了比随机搜索和强化学习更好的结果[4]。最近，人们对专门从事图像识别的深度神经网络的NAS越来越感兴趣[2][3][12]。这里有一份关于可用工具流的最新调查[13]。关于NAS的工作集中在准确性上，作为性能的主要衡量标准，尽管优化NAS可以导致更简化的NNA，反过来可以简化和优化硬件设计[4][9]。另一方面，对硬件性能参数（延迟/吞吐量/功率）的优化通常是在现有的NNA设计上进行的，没有试图修改NNA（层/神经元等）[13]。

III. 进化的细胞辅助设计流程

进化单元辅助设计（ECAD）的流程，预先在[14]中描述，如图1所示，从一个一般的工业/研究问题开始，（a）有足够的数据，（b）有明确定义的输入/输出和(c)
它是一个可以从软件/硬件加速中受益的问题。一旦确定了这样的问题，数据集将被导出为逗号分隔值（CSV）的表格数据格式，此外还将创建一个配置文件，其中将包含以下信息：（a）一般的NNA结构，包括输入和输出大小，初始层数和神经元。
(b)
硬件目标，包括可重新配置的硬件设备类型、DSP数量、内存大小和块数，(c)
光学目标，如精度、吞吐量、延迟和浮点运算。注意，配置文件可以根据现有的模板配置文件和数据集自动生成。

A. 进化过程

基于稳态模型的ECAD进化过程[15]，产生了一个NNA/硬件共同设计的候选群体，每个候选群体都有一套完整的参数，对准确性和硬件性能都有影响。我们在搜索过程中考虑的参数包括层数、层大小、激活函数和偏置。群体中的每个候选者都根据可配置的和潜在的多个标准进行评估，例如，仅准确性或准确性与吞吐量。*原始*评估

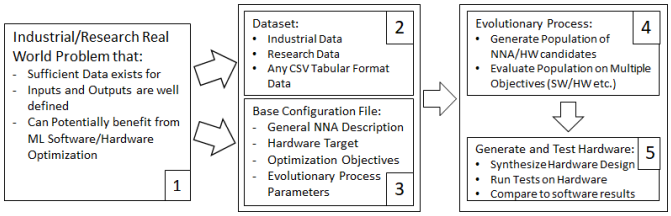


图1：ECAD流程。

测量是在一个软件工件上进行的，在ECAD术语中被称为"*工作器*"。工作者将*原始*评估信息返回给*主*进程。*主*程序通过分配协同设计群体和评估结果来管理评估过程。结果评估是使用用户定义的适配函数完成的。例如，准确度适配函数可以简单地返回一个模拟工作者获得的准确度值（关于工作者类型的更多信息，见下一节）。但它也可以对该值进行缩放或加权，或者指定最小化或最大化该值。简单的评估函数可以在配置文件中指定，而更复杂的函数则写在代码中，通过在框架中注册来添加。

B. ECAD硬件

进化搜索有三个工作器可用于评估各种硬件平台的适用性。仿真器对评估基于指令集的架构（如CPU和GPU）很有用，而物理和硬件数据库器对需要设计和综合的硬件很有用。

仿真工作者从进化引擎中获取查询，并在目标硬件上运行它们。例如，如果目标硬件是GPU，进化引擎向物理工作者发送一个ANN描述，物理工作者将其转换为GPU的可读格式。一旦GPU返回结果，物理工作者负责记录所有必要的指标，提供给引擎以评估适配性。指标包括吞吐量、延迟和耗电量。

硬件数据库工作者为易于模拟或建模的硬件平台提供了一种手段。例如，在我们的实验中，我们利用硬件数据库工作者提供了一种接受ANN描述和硬件配置的方法，它们一起通过一个模型运行，以获得适用性评估的指标。一旦引擎通过跟踪每个配置的适配性找到高性能的设计，该模型就可以被实现为硬件，并通过物理工作者运行。事实证明，FPGA是一个适合硬件数据库工作者的架构。FPGA的可重新配置的特性，加上建模的叠加架构，使工作者能够以相对快速的方式评估许多配置，而不是通过综合工具运行。

物理工作者可用于合成和评估硬件设计。虽然硬件数据库工作者亲

物理工作者通过吞吐量等指标提供整体应用的适用性，物理工作者旨在通过功率、逻辑利用率和操作频率等指标提供硬件设计本身的适用性。在英特尔FPGA的情况下，物理工作者通过ALM、M20K和DSP的利用率、功率估计和时钟频率（Fmax）作出反应。

除了这些工具提供对架构性能的洞察力外，解析进化设计空间后产生的帕累托边界定义了什么是最佳解决方案。每个部署ANN解决方案的行业都可能有不同的意见和对其使用案例的要求。一刀切，或者说一个解决方案是不现实的，也不是最优的。拥有基于权衡的决定的数据是非常有价值的。由于一些ANN描述是近似的，即包括冗余（尽管我们注意到较新的研究开始限制冗余量）和有根据的猜测，为特定硬件平台手工优化ANN描述是不合理的期望。

C. FPGA设计模型和硬件描述

工人可以为任何能够有效建模的硬件提供输入。鉴于FPGA的可重构结构和现代HLS工具能够描述与软件程序很好地衔接的覆盖设计，我们选择了英特尔FPGA和OpenCL作为我们的硬件平台和开发。任何类型的FPGA都可以使用我们开发的覆盖式架构，改变设计搜索空间所需要的只是硬件数据库工作者使用的硬件配置，例如在Arria 10和Stratix 10之间。

硬件数据库工作者收到的配置包含了足够的信息来构建模型并得出必要的性能结果。这个配置的一部分是一个定义目标加速器的特定硬件配置文件。这些信息包括FPGA的名称，相关的原始逻辑细节，如DSP和SRAM的数量，目标时钟频率，要使用的全局存储器（DRAM）的类型，以及其速度和速率。所有这些信息都是用来估计性能的。配置的另一部分是ANN描述。低级别的硬件适用部分从ANN描述中分解出来，并定义加速器必须完成的工作。

我们的模型返回我们认为最基本的数值，包括潜在的和有效的性能、总时间、每秒的产出和延迟。每秒千兆操作数（GOP/s）是描述潜在和有效性能的单位。潜在性能通常是定义配置顶线的市场性能，而有效性能定义了配置在工作负载下的实际或真实性能。它们之间的差值提供了一个衡量问题与模型映射程度的指标。我们的模型的总时间是由所有必要的数据持续存在于加速器DRAM中的时间戳和我们排队等待OpenCL内核的时间戳之间的差异来定义的，一旦所有的结果持续存在于DRAM和最后一个OpenCL内核返回。我们称这为一次运行。每秒输出量是对数据类型的概括，它提供了我们在一秒钟内可以实现的总结果。例如，如果数据类型是图像，那么这个指标可以解释为每秒钟的图像。最后，延迟是指从运行开始到将一个结果存储到DRAM中所需的时间。

在模型中计算这些结果是通过从配置的基线性能开始计算的。所有的数据都是32位浮点，并映射到Arria 10设备的硬化浮点DSP块。从[16]中，我们可以通过确定有多少DSP块在工作来计算出基线性能。DSP的利用率是网格尺寸和矢量宽度的乘积。这个数字是潜在的性能，但在考虑带宽之前。使用配置中的DRAM规格，我们可以确定有多少带宽可用与我们需要多少带宽的比例。每块数据的循环数除以以字节为单位的块的大小，用来计算带宽需求。这种计算就是潜在的性能。接下来，网格配置被用来将ANN分解成一系列封锁的矩阵乘法。有了现在被封锁的数据，我们可以得出所有其他的结果。

IV. 实验

在本节中，我们展示了在六个不同数据集上运行一系列进化搜索的结果。MNIST [17], Fashion MNIST [18], Credit-g, Har, Phishing, and Biore- sponse [19].选择MNIST和Fashion MNIST是为了让我们能够与广泛使用的研究基准进行比较。选择Credit-g/Har/Phishing/Bioresponse是因为它们代表了潜在的真实世界数据集。

FPGA硬件结果假定了III-B节中描述的架构。在两个不同的FPGA上进行了搜索，一个是时钟频率为250 MHz的Arria 10 1150 GX器件，一个是时钟频率为400 MHz的Stratix 10 2800器件。经过多次硬件编译，平均而言，250 MHz是Arria 10的OpenCL设计达到的频率。在250 MHz下运行，可提供759 GFLOP/s FP32单精度性能的峰值吞吐量。Fmax对设计的影响被视为性能的线性扩展。更高的时钟速度确实需要每秒更多的数据，导致可用的全局存储器可能饱和。加速器卡是英特尔的开发套件，它包含一个单组DDR4内存，提供19.2GB/s的峰值带宽。在许多情况下，进化算法要求的配置导致了带宽受限的设计。我们确实提供了一些结果，其中包括有2个和4个DDR组的设计空间，提供38.4和4个DDR组。76.8 GB/s的带宽。所有Stratix 10型号都使用4组DDR运行。

GPU的结果是使用三种不同的设备获得的。一台带有8 GB DDR5的NVIDIA Quadro M5000，能够以211 GB/s的内存带宽实现4.3 TFLOP/s的FP32单精度性能，一台能够实现12 TFLOP/s的FP32单精度性能的Titan X和Radeon VII 16GB HBM2。13.44 FP32 TFLOPS和1 TB/s内存带宽。剖析

GPU的时序报告是使用从Tensor Flow生成的跟踪文件完成的。时序报告考虑了矩阵乘法、交流和向量加法程序，但似乎没有考虑到DRAM传输。FPGA时序报告确实考虑了DRAM，因为内存缓冲是设计中的一个积极组成部分。总的来说，结论没有受到时序报告之间的差异的影响，这种差异可能使FPGA和GPU的直接比较发生偏差（有利于GPU）。

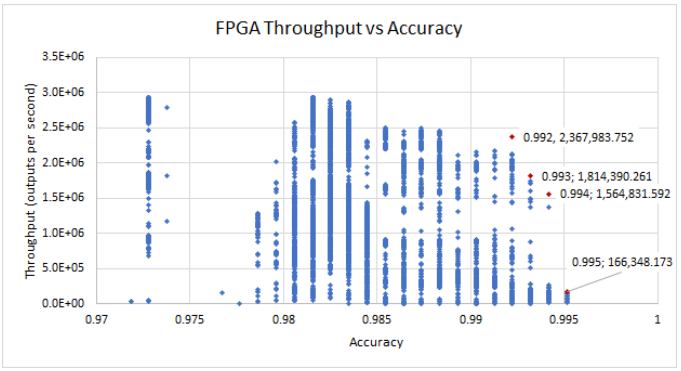
A. 总体表现结果

我们首先检查了我们使用进化流程所能达到的准确性结果。表一介绍了使用10折评价方法[20]从进化算法搜索准确率获得的最高结果。这种方法将数据集分成10个相等的训练/测试褶皱，并在每个褶皱上测量性能。可以看出，ECAD MLP（我们的方法）能够取得比所有基于MLP的分类器更好的准确性结果。此外，ECAD的进化过程设法在credit-g和网络钓鱼数据集方面超过了基于MLP的方法。mnist和fashion-mnist数据集是在OpenML[19]之外获得的，使用传统的1倍训练/测试数据集，我们将它们与文献[18][17]中发表的结果进行比较。可以看出，我们的mnist和fashion-mnist的准确率超过了顶级报告的结果。此外，我们的自动MLP网络有第二好的报告结果，比SVC方法的记录保持者少了0.0047。表三显示了表一和表二中报告结果的ECAD运行时间统计。它报告了由ECAD系统自动生成和评估的不同NNA/HW组合的数量，每次评估的平均时间和所有候选架构的总评估时间。请注意，为了优化系统的搜索和运行时间，首先分析潜在的NNA/HW候选人与以前的评估的相似性，重复的不会被评估两次。

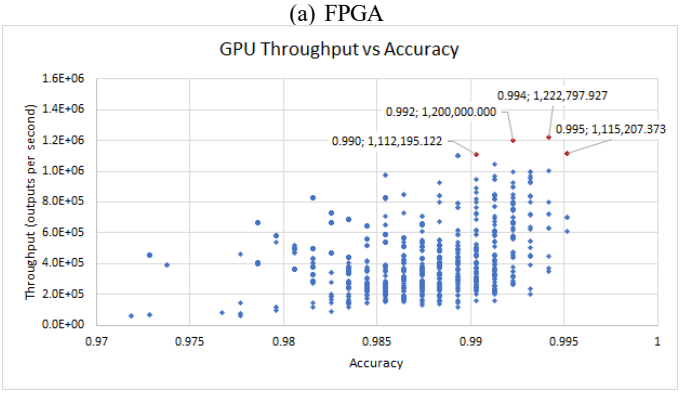
表四显示了每个数据集的两个顶级帕累托前沿解决方案的结果。请注意，这个搜索是在OpenML规范之外进行的，即运行一个单一的折叠。这些解决方案提供了Stratix 10 (S10)FPGA和TitanX (TX)GPU的精度和吞吐量。在大多数情况下，FPGA取得了比GPU更高的性能。例如，credit-g有利于GPU获得更高的精度，但看一下credit-g的第二行，通过牺牲仅仅一个点的精度，FPGA看到了吞吐量的非常显著的改善。

B. 准确度与吞吐量的关系

这项研究的部分动机是为了展示可重构硬件对神经网络的适应性。这种灵活性和对一系列GEMM调用的塑造产生了特殊的结果，旨在适应一个最佳的网络描述，即最高的准确性和有效的吞吐量。我们开始在HAR数据集上运行进化搜索，观察GPU和FPGA对每个进化步骤的反应。图2a提供了Arria 10的结果，图2b提供了Quadro M5000的结果，其中一个单一的



(a) FPGA



(b) GPU

图2：FPGA和GPU在不同精度水平上对har数据集的性能。

蓝点代表进化过程的结果，最佳结果显示为红点。

进化过程在精度方面提供了许多高性能的结果，最高为0.995，并为吞吐量提供了许多不同的结果。吞吐量结果的广泛波动（特别是对于GPU，作为一个固定的架构）表明，存在许多MLP解决方案可以达到最高的精度。GPU以相同的方式加速每个解决方案，因此不同的吞吐量水平意味着MLP结构正在发生变化。显示出GPU在0.995的精度下实现的最高吞吐量约为每秒1E6个结果，而最低吞吐量的结果约为每秒6E5个结果。对于同样的推理，当我们向下移动一个精度点时，GPU的性能几乎没有变化，这是因为神经元的数量大致保持不变，是层之间的分布造成了对精度的影响。对于GPU来说，神经元的数量和吞吐量之间基本没有关系。FPGA则有不同的关联性。图2a显示，MLP中神经元的分布对性能有很大影响。与GPU不同，每个数据点都有（潜在的）不同的硬件配置。虽然最高精度只达到每秒1.6E5个输出，但向下移动精度仅0.1%就会导致每秒1.56E6个输出的巨大飞跃，高出一个数量级。此外，再降低0.2%，性能又提高了约1.5倍。每个数据集

表一:与以往作品相比，所有数据集的前10倍精度（Acc）。

数据集	OpenML任务标识	顶尖高手(任意)	顶级方法	最受欢迎的投资(MLP)	MLP类型	ECAD MLP
信用-g	31	0.7860	mlr.classif.ranger	0.7470	*MLPClassifier	**0.7880**
晴天	14970	0.9957	*DecisionTreeClassifier	0.1888	*MLPClassifier	0.9909
钓鱼网站	34537	0.9753	*SVC	0.9733	*MLPClassifier	**0.9756**
生物反应	14966	0.8160	mlr.classif.ranger	0.5423	*MLPClassifier	0.8038

*注：*OpenML数据集/结果可以在openml.org找到。

表二：与以前的作品相比，所有数据集的最高1倍精度（Acc）。

数据集	顶尖高手(任意)	顶级方法	最受欢迎的投资(MLP)	MLP类型	ECAD MLP
MNIST	0.9979	手册	0.9840	手动（无失真）。	0.9852
时尚MNIST	0.8970	SVC	0.8770	编码	0.8923

*注意*MNIST和Fashion MNIST是独立的预拆分（1倍）数据集。

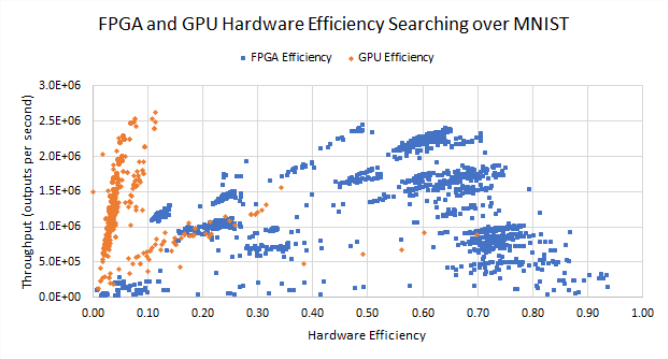
表三：最高准确度的运行时间统计

数据集	评估的模型总数	AVG模型评估时间(s)	总评估时间（秒）
MNIST	553	71.23	39388.6
时尚MNIST	481	82.55	39708.7
信用-g	10480	2.24	23495.2
晴天	3229	10.20	33069.4
钓鱼网站	3534	9.24	32661.3
生物反应	5309	5.89	31285.0

注意
每个生成的模型都是NNA特质和硬件特质的全功能组合，在任何测量指标上都要进行性能评估。ECAD系统*缓存了*类似的配置，避免了重新评估。

表四：搜索准确率和吞吐量的最佳帕累托前沿结果

数据集	准确度	S10（输出/秒）	TX（输出/秒）
MNIST	0.9841	7.97E5	7.73E5
MNIST	0.9763	2.45E6	1.97E6
时尚MNIST	0.893	4.8E5	8.1E5
时尚MNIST	0.8850	1.92E6	2.3E6
晴天	0.996	1.16E6	9.59E5
晴天	0.985	4.74E6	2.46E6
信用-g	0.83	8.19E3	1.59E6
信用-g	0.82	1.40E7	1.23E6
生物反应	0.798	4.64E5	1.34E6
生物反应	0.7952	1.36E6	1.66E6
钓鱼网站	0.9675	6.81E6	2.27E6
钓鱼网站	0.9656	1.16E7	2.27E6



测试显示了这一相同的趋势。

C. 硬件效率和扩展到更大的设备

我们修改了硬件数据库工作者，以返回基于具有4组DDR的Stratix 10 2800设备的结果。这个设备比我们在以前的实验中使用的Arria 10设备提供了高达10倍的性能扩展。虽然Stratix 10设备的性能高达10 TFLOP/s，但我们使用了与Arria 10设备相同的方法，在400MHz的时钟频率下搜索Stratix 10设备，将屋顶线缩小到4.6 TFLOP/s。为了更好地匹配Stratix 10器件的性能，我们使用了一个能够达到12 TFLOP/s的Nvidia Titan X器件。

我们发现，总的来说，可重构架构有更大的潜力来执行更高的水平；然而，对于某些数据集，Titan

图3：Stratix 10 2800和Titan X搜索MNIST数据集的硬件效率结果

能够实现最高的每秒产出。吞吐量是一个需要考虑的流行指标，但我们也发现效率是非常重要的。较大的FPGA设备处理通过MLP运行这些数据集的原因是由于资源的分配。当进化算法选择一个硬件配置时，在FPGA上有一个分配的空间，其中包含了潜在的性能。然后在映射MLP之后，我们得到一个有效的性能（关于潜在和有效性能的细节，见第三部分C）。有效性能与潜在性能的比率给了我们硬件效率。效率可以产生更少的FPGA资源，同时保持吞吐量，使逻辑可用于其他任务，如预处理或后处理。

图3显示了不同的FPGA和GPU的效率。

在MNIST数据集上搜索出的解决方案。这一特定运行的最高精度为0.9845，FPGA和GPU的吞吐量分别为每秒796,611和773,162个输出，几乎相同。如果我们考虑这个结果的效率，FPGA利用了41.5%的分配逻辑，而GPU只利用了0.3%。我们将GPU的效率计算为从设备每秒总潜在操作中获得的每秒操作数。结论是，在MLP开发过程中，如果没有考虑到目标硬件，很有可能会失去效率。在可重新配置的硬件上运行进化搜索可以平衡吞吐量和准确性之间的适配性。

V. 结论

我们通过利用进化搜索算法来解决设计高性能神经网络的困难，该算法能够为分类精度和硬件吞吐量找到最合适的解决方案。与传统的方法相比，这个过程显示出高度的效率和效果。我们提出了我们对最先进的神经网络配置的结果，超过了目前已发表的工作。我们通过讨论实验结果来解释协同设计的力量，实验结果显示了准确率与吞吐量、性能与带宽的比例以及与大型设备的比例设计。

参考文献

[1] A.Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," *arXiv preprint arXiv: 1605.07678*, 2016.

[2] H.Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu, "Hierarchical representations for efficient architecture search," *arXiv preprint arXiv: 1711.00436*, 2017.

[3] E.Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, Tan, Q. V. Le, and A. Kurakin, "Large-scale evolu- tion of image classifiers," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp.2902-2911.

[4] E.Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," *arXiv preprint arXiv:1802.01548*, 2018.

[5] K.Hazelwood, S. Bird, D. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, *et al.*, "Facebook的应用机器学习。A datacenter infrastructure perspective," in *2018 IEEE International Symposium on High Performance Com- puter Architecture (HPCA)*, IEEE, 2018, pp.620-629.

[6] C.-J. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M.Dukhan, K. Hazelwood, E. Isaac, Y. Jia, B. Jia, *et al.*, "Machine learning at facebook:Understanding inference at the edge," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, IEEE, 2019, pp.331-344.

[7] N.Jouppi, C. Young, N. Patil, and D. Patterson, "Motivation for and evaluation of the first tensor processing unit, " *IEEE Micro*, vol. 38, no.3, pp. 10-19, 2018.

[8] J.Park, M. Naumov, P. Basu, S. Deng, A. Kalaiah, D. Khudia, J. Law, P. Malani, A. Malevich, S. Nadathur, *et al.*, "Facebook数据中心的深度学习推理。表征、性能优化和硬设备影响，"*arXiv预印本arXiv:1811.09886*，2018。

[9] T.Elsken, J. H. Metzen, and F. Hutter, "Neu- ral architecture search:A survey," *arXiv preprint arXiv:1808.05377*, 2018.

[10] G. F. Miller, P. M. Todd, and S. U. Hegde, "Designing neural networks using genetic algorithms.," in *ICGA*, vol. 89, 1989, pp.379-384.

[11] K.O. Stanley和R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation*, vol. 10, no. 2, pp.99-127, 2002.

[12] B.Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.8697-8710, 2018.

[13] S.I. Venieris, A. Kouris, and C. -S.Bouganis, "在FPGA上映射卷积神经网络的工具流程。A survey and future directions," *ACM Com- puting Surveys (CSUR)*, vol. 51, no.3, p. 56, 2018.

[14] P.Colangelo, O. Segal, A. Speicher, and M. Margala, "Artificial neural network and accelerator co-design using evolutionary algorithms," in *2019 IEEE High Performance Extreme Computing Conference（HPEC）*, 2019, pp.

[15] D.E. Goldberg和K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of genetic algorithms*, vol. 1, Elsevier, 1991, pp.69-93.

[16] A.Vishwanath等人，"启用高性能浮点设计"，*英特尔，白皮书*，2016。

[17] Y.LeCun和C. Cortes，"MNIST手写数字数据库"，2010。[在线]。Available: http://yann.lecun. com/exdb/mnist/.

[18] H.Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist:A novel image dataset for benchmarking machine learning algorithms," *CoRR*, vol. abs/1708.07747, 2017. arXiv: 1708.07747.[在线]。Available: http://arxiv.org/abs/ 1708.07747.

[19] J.Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "Openml:机器学习中的网络化科学"，*SIGKDD探索*，第15卷，第2期，第49-60页，2013。DOI: 10 .1145 / 2641190 .2641198.[在线].可用。http://doi.acm.org/10.1145/2641190.2641198.

[20] R.Kohavi等人，"交叉验证和引导带的准确性估计和模型选择的研究，"在*Ijcai*，蒙特利尔，加拿大，第14卷，1995年，第1137-1145页。