中国科学技术大学

硕士学位论文

Android 平台的 CNN 模型能效优化问题研究

作者姓名： 王震

学科专业： 计算机系统结构

导师姓名： 周学海 教授 李曦 副教授

完成时间： 二〇一八年三月二十二日

University of Science and Technology of China

A dissertation for master's degree

Research about Energy Efficiency Optimization of CNN Models on Android Platform

Author: Zhen Wang

Speciality: Computer Architecture

Supervisors: Prof. Xuehai Zhou, Assoc. Prof. Xi Li

Finished time: March 22, 2018

中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名：_____　　签字日期：_____

中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入《中国学位论文全文数据库》等有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

□ 公开　□ 保密（____ 年）

作者签名：_____　　导师签名：_____

签字日期：_____　　签字日期：_____

摘　要

近年来，卷积神经网络（CNNs）因其具有高推断精度和强自适应性而被广泛应用于各种领域，例如：计算机视觉、语音识别等。移动手机当前已经成为了人们日常生活中的随身携带之物，其每天都产生着大量与人们相关的传感数据，因此许多工程项目也尝试着将卷积神经网络应用于移动平台来处理这些数据为人类提供更加智能的服务。然而，由于受到当前移动平台的资源限制（内存、计算能力、电池容量等），基于 CNN 模型的应用并没有在手机移动平台上成为主流。

目前，手机上基于 CNN 模型的应用绝大部分都是采用“客户端-服务器”模式。然而，这种模式不仅强依赖于网络性能而且会导致用户隐私泄露，所以于手机移动端直接进行卷积神经网络的高能效前向推断已然成为学术界和工业界所需共同面对的严峻挑战。本文直面这些挑战，采用一系列优化策略开发出一套可以高能效运行在 Android 平台的卷积神经网络推断时库，并将其运用在一个生活日志型应用中，以探索从系统层进一步降低该类场景应用的运行时功耗。论文的主要工作包括：

1. 基于 OpenCL 异构编程框架使用手机移动端 GPU 进行卷积神经网络的前向推断过程。解析 Caffe、Tensorflow 等深度学习框架训练出来的卷积神经网络模型权重，以便在手机端重构网络，并将卷积、内积等计算密集型的操作转移到手机 GPU 上进行。

2. 通过对卷积神经网络全连接层权重的剪枝、重训操作，降低网络模型的存储占用。使用稀疏矩阵向量乘代替密集矩阵的内积操作，进一步降低卷积神经网络的前向推断时间。

3. 为了能够充分利用当前以及未来移动设备 SoC 所提供的异构计算特征，本文提出了一种简单有效的方法，使得基于 CNN 模型的应用在运行时可以根据其所处运行环境中 CPU、GPU 等异构处理器的性能差异，自适应地调整 CPU、GPU 等所分配的计算任务量。

4. 针对基于 CNN 的生活日志型应用，本文从系统层分析并建模预测这类应用的运行时负载，探索使用动态电压频率调节技术（DVFS）调整 CPU/GPU 的电压、频率来进一步降低基于 CNN 模型应用的运行时功耗之可能性。

关键词：能效；异构计算；权值压缩；卷积神经网络；移动平台

ABSTRACT

Convolutional Neural Networks (CNNs) have become more and more powerful in the computer vision domain, as they achieve the state-of-the-art accuracy. Despite this, it is generally difficult to apply CNNs on mobile platforms. Client-server paradigm is a straightforward way to deploy CNNs on mobile phones, but studies have shown that it suffers serious problems, such as privacy leaks. Recently, researchers focus on using heterogeneous local processors (e.g., GPUs, CPUs) to accelerate the inference of CNNs. Utilizing all local processors available can achieve the highest performance, but it might incur energy-inefficiency. Different from previous works, this paper concerns more about energy-efficiency of CNN-based mobile applications. We present an adaptive strategy, which is able to compute the energy-efficiency of all local processors, and further to obtain the energy-efficient device processor combination to perform CNN inference in parallel. The strategy is implemented on ODROID platform, where the evaluation results show that our proposed approach provides 3.67× higher energy-efficiency with only 9.7% performance degradation on average compared with the greedy strategy which tries to use all local processors available.

Key Words: Energy-efficiency;Weights compression; Heterogeneous computing; CNNs; Mobile platform

目　录

第 1 章　绪　　论

1.1　引言

随着深度学习领域不断取得突破性的进展，基于深度学习算法的手机移动应用也如雨后春笋般发展起来。智能手机已经在逐渐改变人们的生活方式，几乎每部手机都集成着若干功能各异的传感器，这些传感器每天都在采集着与使用者相关的数据，而深度学习算法无疑是处理这些数据的最佳工具。然而，由于当前嵌入式平台的资源限制（内存、计算能力、电池容量等），基于深度学习算法的应用并没有在手机移动应用市场中成为主流。

虽然深度学习模型对硬件资源的苛刻需求阻碍了它向手机移动平台的“进军”，但是其带来的好处仍然诱惑着人们在物联网和移动硬件上采用它。目前，手机上基于深度学习模型的应用（如，图像识别、语音识别）绝大部分都是基于云端服务的，即手机端采集所需的传感数据并通过网络将数据上传到云端服务器，云端服务器在获得的数据后，运行深度学习算法进行推断，并将所得的推断结果再次通过网络传回至手机端。然而，这种处理方式带来了许多负面影响，主要总结如下：1）它可能会泄露用户的隐私数据，因为它需要将用户的一些敏感数据（如，图像、音频）发送到第三方服务端进行处理；2）它的推断执行时间将会与波动的、不可预测的网络质量（如，网络延迟、吞吐量）紧密挂钩。更糟糕的是，当网络条件很差甚至不可获取时，这种推断预测就不能正常工作；3）由于无线通信能量开销的存在，对于那些需长时期运行的应用（如，增强现实、认知助理等），使用云端处理推断任务也是不切实际的；4）当用户使用的是移动网络时，某些应用（如，需要发送视频流到云端进行处理）使用云端处理，将会消耗大量的流量，这是用户所无法接受的。

考虑到上述云端执行的负面影响，用户可能更希望那些基于深度学习模型的应用在手机本地就可以完成推断任务。另一方面，我们需要意识到高能效移动处理器的运算能力和结构复杂度一直处于不间断地发展中。例如，与 4 年前的 iPhone 5S 相比，2017 年苹果发布的 iPhone 8 在 CPU 单核处理性能上拥有着 232% 的增长, 而在多核处理性能上更是提高了 373%。许多研究学者认为，在不久的将来，即使没有远程计算的辅助，手机移动端也可以胜任许多基于深度学习模型的计算任务。通过手工改造并简化深度学习模型（如卷积神经网络），在移动端本地设备上直接运行一些深度学习模型已经被证明是可行的，但是这样做不但需要大量的设计技巧，而且对于现存的大多数深度学习模型来说都是不可行的。更为重要的，正是因为深度学习模型的复杂性才使得推断准确度和鲁棒性

取得了革命性的飞跃，而这才是智能移动应用所迫切需要的，所以手工简化模型的方式并非是我们的初衷。

与桌面、服务器端相比，手机移动端的不足不仅表现在较弱的处理能力上，还凸显在较小的内存容量上（如，华为 Mate 9 Pro 的内存容量为 4G）。然而，因为深度学习模型具有结构异常复杂的特点，所以绝大部分模型都拥有着成百万上千万甚至上亿的参数（如，VGG16 模型拥有 1.8 亿个参数）。这导致了许多深度模型并不能在手机移动端运行。而且，即使一些模型可以运行，其也会造成大量的内存能耗开销。

综上所述，采用深度学习算法处理移动传感数据并在手机端进行离线推断是未来手机移动应用发展的一种趋势。然而，为了实现这一目标，必须要对手机端深度学习模型的本地推断过程进行各种能效优化。因此，当前研究人员主要面对的挑战如下：

1. 如何利用当前移动设备处理器的特点及其未来的发展趋势，使得深度学习模型可以高能效地于手机端进行离线推断。

2. 受限于当前手机内存容量较小的不足，如何使得庞大的深度学习模型也可以正常地于手机端运行。

3. 在保证性能的条件下，如何根据应用场景的特点和系统层提供的信息，进一步降低上层基于深度学习模型应用的运行时功耗。

1.2　深度学习模型于移动端能效优化的研究现状

1.2.1.　早期的探索与尝试

Lane 等人[1] 设计了一个基于移动设备 CPU 和 DSP 的低功耗深度神经网络的原型推断系统。他们利用该推断系统研究了一些典型的移动感知任务（如，行为感知），并将该推断系统与更加通用的传统辨识技术（如，SVM、GMM 和决策树等）相比较。他们的发现表明推断系统中所使用的深度神经网络 (DNNs) 并不会给现代的移动硬件带来过度的负载压力，而且即使是一个简单的 DNN 模型（如，减少 71 倍的输入特征数），与传统的通用学习技巧相比，也可以改善推断精度。文献[2] 研究了一些深度学习算法（DNNs 和 CNNs）在按比例缩小后，于资源受限的嵌入式设备运行推断阶段时的行为和资源特征。Yanai 等人[3] 探究了适合在移动端实现的 CNN 结构，并提出了多可扩放的 network-in-networks(NIN)，即用户可以调整识别时间和识别精度的折中比。他们的研究发现 BLAS 库适合加速 IOS 系统上卷积层的快速计算，而 NEON SIMD 更适合在 Android 系统上加

速卷积层的计算。

1.2.2. 基于模型压缩的优化

文献[4] 通过使用线性压缩技巧去除 CNN 模型中的冗余参数，有效加速了大型已训练好 CNN 模型的测试时，而为此只需要付出很小的性能折中。与[4] 类似，Jaderberg 等人[5] 利用不同特征通道和卷积核间存在冗余的特征对卷积核进行低秩近似分解。文献[6] 使用非线性最小平方去计算一个四维卷积核的低秩 CP-分解，即使用一些秩为 1 的张量之和表示该四维卷积核。Wu 等人[7] 提出一个量化的卷积模型，可以在加速计算的同时降低模型的存储和内存的开销。Wang 等人[8] 提出一个基于低秩的、分组稀疏向量分解的方法对 CNN 模型的测试阶段进行加速。其将卷积核分解为一些小数量的低多线性秩张量之和，并用这些近似张量代替原始的卷积核进行标准回传以达到对模型的进一步微调。

1.2.3. 基于异构计算的优化

为了验证使用移动端 GPU 做异构计算所取得的性能，Lokhmotov 等人[9] 通过在三星 Chrome-book 2 平台上进行预实验来刻画 AlexNet 的前向传输过程。最终，他们发现带有 OpenBLAS 支持的 Caffe 要比带有 ViennaCL 支持的 Caffe 快大约 4 倍，比带有 clBLAS 支持的 Caffe 快大约 10 倍。因此，当前移动端的 GPU 性能较差，并且，支持 OpenCL 的现有并行库并没有针对移动端进行优化。DeepX 是 Lane 等人[10] 为在移动平台上运行深度学习模型设计并实现的一个软件加速器。DeepX 利用一个基于网络计算（远程处理器）和本地异构处理器的混合体（包含 CPUs，GPUs，LPUs 等）来降低资源的开销，并通过两个推断时资源控制算法来提升性能，即：（1）运行时层压缩（Runtime Layer Compression , RLC）和（2）深度结构分解（Deep Architecture Decomposition , DAD）。然而，DeepX 主要缺点有两点：（1）运行时层压缩不仅没有减少运行时内存的开销，还加重了处理器的计算量；（2）其运行时所使用的一些决策参数来源于离线计算好的值，不能适应运行环境的变化。Huynh 等人[11] 设计了一个基于移动 GPU 的深度神经网络框架 DeepSense。DeepSense 是基于 OpenCL 的，故而可以在 GPU 上运行各种不同的 CNN 模型。然而，其计算密集型的操作（如，卷积运算）主要运行在 GPU 上，没有充分利用 CPU 的多核处理能力。文献[12] 呈现了一个基于 GPU 加速的库（CNNdroid），其主要使用了 Android 官方的高性能计算框架 RenderScript 来加速已训练 CNN 模型的推断过程。然而，其并没有对一些较大的模型做压缩处理，使得一些大模型不能正常运行。

1.2.4. 其他优化方法

除了上述三类研究外，还有一些其他的相关研究。文献[13–15] 通过按比例缩小深度模型的方法，将深度学习算法缩小后运行在手机或 DSP 上。使用低功耗处理器也被证明对于连续传感类型的应用特别有效，该类系统诸如 Speakersense[16]、Dsp.ear[17] 等，它们在应用级进行优化以均衡主处理器和协处理器间的负载。Antoniou 等人[18] 将深度卷积模型应用在智能监控系统中，分别于 PC 和移动设备上实现了一款可以自动检测、智能识别的在线监控系统。最后，开发深度学习领域的专用硬件是另外一个当前较为热门的研究方向，许多研究人员对此做出了贡献，如 Diannao[19] 以及 FPGA 神经网络加速器[20] 等。

可以看出深度学习模型于手机移动端的应用是近两年刚刚兴起并快速升温的研究方向。然而，大量研究学者的关注点都是放在如何提高深度学习模型于移动端的运行速度，而本课题的研究重点更多的是能效的兼顾，并且更多的会考虑能耗问题。

1.3 论文的主要研究工作

本文主要针对当前已被工业界广泛应用的深度卷积神经网络模型（CNNs）进行 Android 平台能效优化的研究，且主要工作包含以下方面：

1. 分析 Caffe、Tensorflow 等深度学习框架进行卷积神经网络前向推断时所需要的算子，并通过 OpenCL 异构编程框架于手机 GPU 上实现计算密集型的算子。离线解析 Caffe、Tensorflow 等深度学习框架训练出来的卷积神经网络模型权重，并将这些权重保存成统一的格式，以便在手机端可以重构不同框架训练出来的 CNN 模型。

2. 在保证推断精度损失极少的条件下，使用“剪枝-重训”算法对预训练好的 CNN 模型参数进行压缩，这样不仅可以降低网络模型的存储占用，还减少了模型重构过程中的访存数量。针对剪枝得到的稀疏矩阵，使用稀疏矩阵向量乘代替内积操作，进一步降低卷积神经网络的前向推断时间。

3. 为了能够充分利用当前以及未来移动设备 SoC 所提供的异构计算特征，本文提出了一种简单有效的方法，使得基于 CNN 模型的应用在运行时可以根据其所处运行环境中 CPU、GPU 等异构处理器的性能差异，自适应地调整 CPU、GPU 等所分配的计算任务量。

4. 针对需长时期运行的 CNN 模型应用（Life-logging Apps）, 本文通过离线分析其负载特征并建立相应模型，以期在系统层预测该类应用的运行时负载。

然后，结合动态电压频率调节技术（DVFS），本文探索了从系统层进一步降低该类应用运行时功耗的可能性。

通过上述第 1、2、3 点优化策略，本课题预期开发出一套可以高能效运行在 Android 平台的 CNN 推断时库，并最终将其运用在一个生活日志型应用（如，智能监控系统）中以验证第 4 点优化策略的有效性。

1.4　论文的组织结构

本文的章节安排如下：第 1 章主要讲述了深度学习模型于手机移动端应用的现状，并分析了阻碍基于深度学习模型移动应用发展的原因以及使用手机本地离线推断的必要性。接着，通过详细介绍深度学习模型于移动端进行能效优化的研究现状，引出本文的研究内容与目标。

第 2 章首先介绍了本文的主要能效优化对象（卷积神经网络）的基本概念和组成结构。然后，探讨了当前移动端 SoC 的发展趋势，并介绍了动态电压频率调节技术（DVFS）的相关概念。最后，描述了使用 OpenCL 异构编程框架开发基于 GPU 通用计算程序的设计流程与方法。

第 3 章首先对卷积神经网络前向推断过程中所使用的基本算子进行分解，并详细描述了这些算子的作用与原理。之后，分别给出了这些基本算子在手机 CPU 和 GPU 上实现的方法。接着，本文描述了基于“剪枝-重训”的权重压缩方法，通过该方法本文对卷积神经网络中占存储量主要部分的全连接层权重进行了压缩。对于压缩后的网络，进一步使用稀疏矩阵向量乘（SpMV）代替密集矩阵的内积运算，并分别给出了 SpMV 的 CPU 和 GPU 实现。最后，本文基于 Caffe、Tensorflow 等深度学习框架训练出来的 CNN 模型权重，在手机端重构 LeNet-5 和 AlexNet, 并比较了基于手机 CPU、手机 GPU 和基于 SpMV 的实现之间的能效差异。

第 4 章首先全面分析了于手机 CPU 和 GPU 上分别执行 CNN 前向推断时的能效，并阐述了利用所有可获得的手机本地处理器去执行 CNN 前向推断并非是一种高能效的方式。本文提出一种简单有效的方法，其可以自适应地计算特定移动平台上所有可获得的本地处理器的能效，并进一步通过计算得到一个高能效的组合去执行 CNN 的前向推断过程。接着，本文提出一种基于不同处理器计算性能的方法为所选组合中每一个设备处理器分配计算任务。

第 5 章基于第 2、3、4 章设计的 CNN 运行时库开发了一款生活日志型 Android 应用——智能监控系统。针对该应用，本文从系统层分析了其负载特征并建立模型对其负载进行预测，探索了利用 DVFS 技术进一步降低该类应用功耗的可能性。

第 6 章对全文进行了总结，并对论文中尚未解决的问题提供研究线索，以期在未来的工作中加以解决并完善。

第 2 章　相关研究技术

2.1　卷积神经网络

2.2　移动端 SoC 的发展趋势

多核异构 CPUs（如，ARM 的 big.LITTLE[21]）已然成为当前移动设备处理器的主流架构，而 GPUs 也已集成在大部分的移动设备中。GPUs 与生俱来的并行计算能力很适合用来处理深度模型中的常见计算类型。然而，处理能力较强的 GPUs 也会以惊人的速度消耗着移动设备电池电量。事实上，移动 GPUs 的设计过程中更加重视的是低功耗而不是高性能，所以当前商业上应用的大多数移动 GPUs 的计算能力并不是强大，如 Mali™-T628 MP6 的频率仅为 600MHz、核心数仅为 6。因此，单独的 GPUs 解决方案也不能够满足移动平台的深度学习模型的运行条件。除 GPUs 外，我们还应该注意到，移动设备中也集成了一些低功耗处理器，如 DSPs、LPUs、NPUs 等。高通骁龙系列的 SoC 集成了 Hexagon DSP；英伟达的 Tegra K1 SoC 除了提供了高性能的 GPU（192 核）、2.3Ghz 的 4 核 CPU 外，还提供了一个第五代低功耗核 LPC；华为的海思麒麟 970 还内置了神经网络处理单元（NPU），使用 NPU 可进行高效的 AI 相关计算。每一种处理器都拥有着其自己的资源特征。根据层的类型和其他方面的特征，使用不同的处理器组合执行不同深度学习模型，这样便可以带来不同的性能-功耗折中。因此，如何高效地利用这些异构处理器将是移除深度学习被嵌入式平台所广泛采用之屏障的关键[22]。

2.3　OpenCL 异构编程框架

第 3 章　基于移动 GPU 加速和离线层压缩的能效优化

本章

3.1　一级节标题

3.1.1.　二级节标题

1.　三级节标题

第 4 章　基于异构计算的自适应计算任务分配

4.1　一级节标题

4.1.1.　二级节标题

1.　三级节标题

(1)　四级节标题

①　五级节标题

第 5 章　基于应用场景特点的系统层优化

5.1　一级节标题

5.1.1.　二级节标题

1.　三级节标题

(1)　四级节标题

①　五级节标题

第 6 章　总结与展望

6.1　本文工作总结

6.2　未来工作展望

参 考 文 献

[1] LANE N D, GEORGIEV P. Can deep learning revolutionize mobile sensing?[C]//Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications. [S.l.]: ACM, 2015: 117–122.

[2] LANE N D, BHATTACHARYA S, GEORGIEV P, et al. An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices[C]//Proceedings of the 2015 International Workshop on Internet of Things towards Applications. [S.l.]: ACM, 2015: 7–12.

[3] YANAI K, TANNO R, OKAMOTO K. Efficient mobile implementation of a cnn-based object recognition system[C]//Proceedings of the 2016 ACM on Multimedia Conference. [S.l.]: ACM, 2016: 362–366.

[4] DENTON E L, ZAREMBA W, BRUNA J, et al. Exploiting linear structure within convolutional networks for efficient evaluation[C]//Advances in neural information processing systems. 2014: 1269–1277.

[5] JADERBERG M, VEDALDI A, ZISSERMAN A. Speeding up convolutional neural networks with low rank expansions[J]. arXiv preprint arXiv:1405.3866, 2014.

[6] LEBEDEV V, GANIN Y, RAKHUBA M, et al. Speeding-up convolutional neural networks using fine-tuned cp-decomposition[J]. arXiv preprint arXiv:1412.6553, 2014.

[7] WU J, LENG C, WANG Y, et al. Quantized convolutional neural networks for mobile devices [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 4820–4828.

[8] WANG P, CHENG J. Accelerating convolutional neural networks for mobile applications[C]// Proceedings of the 2016 ACM on Multimedia Conference. [S.l.]: ACM, 2016: 541–545.

[9] LOKHMOTOV A, FURSIN G. Optimizing convolutional neural networks on embedded platforms with opencl[C]//Proceedings of the 4th International Workshop on OpenCL. [S.l.]: ACM, 2016: 10.

[10] LANE N D, BHATTACHARYA S, GEORGIEV P, et al. Deepx: A software accelerator for low-power deep learning inference on mobile devices[C]//Information Processing in Sensor Networks (IPSN), 2016 15th ACM/IEEE International Conference on. [S.l.]: IEEE, 2016: 1–12.

[11] HUYNH L N, BALAN R K, LEE Y. Deepsense: A gpu-based deep convolutional neural network framework on commodity mobile devices[C]//Proceedings of the 2016 Workshop on Wearable Systems and Applications. [S.l.]: ACM, 2016: 25–30.

[12] LATIFI OSKOUEI S S, GOLESTANI H, HASHEMI M, et al. Cnndroid: Gpu-accelerated execution of trained deep convolutional neural networks on android[C]//Proceedings of the 2016 ACM on Multimedia Conference. [S.l.]: ACM, 2016: 1201–1205.

[13] LANE N D, GEORGIEV P, QENDRO L. Deepear: robust smartphone audio sensing in unconstrained acoustic environments using deep learning[C]//Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. [S.l.]: ACM, 2015: 283–294.

[14] CHEN G, PARADA C, HEIGOLD G. Small-footprint keyword spotting using deep neural networks[C]//Acoustics, speech and signal processing (icassp), 2014 ieee international conference on. [S.l.]: IEEE, 2014: 4087–4091.

[15] VARIANI E, LEI X, MCDERMOTT E, et al. Deep neural networks for small footprint text-dependent speaker verification[C]//Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. [S.l.]: IEEE, 2014: 4052–4056.

[16] LU H, BRUSH A B, PRIYANTHA B, et al. Speakersense: Energy efficient unobtrusive speaker identification on mobile phones[C]//International conference on pervasive computing. [S.l.]: Springer, 2011: 188–205.

[17] GEORGIEV P, LANE N D, RACHURI K K, et al. Dsp. ear: Leveraging co-processor support for continuous audio sensing on smartphones[C]//Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems. [S.l.]: ACM, 2014: 295–309.

[18] ANTONIOU A, ANGELOV P. A general purpose intelligent surveillance system for mobile devices using deep learning[C]//Neural Networks (IJCNN), 2016 International Joint Conference on. [S.l.]: IEEE, 2016: 2879–2886.

[19] CHEN T, DU Z, SUN N, et al. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning[J]. ACM Sigplan Notices, 2014, 49(4): 269–284.

[20] ZHANG C, LI P, SUN G, et al. Optimizing fpga-based accelerator design for deep convolutional neural networks[C]//Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. [S.l.]: ACM, 2015: 161–170.

[21] CHUNG H, KANG M, CHO H D. Heterogeneous multi-processing solution of exynos 5 octa with arm® big. little™ technology[J]. Samsung White Paper, 2012.

[22] ATTIA K M, EL-HOSSEINI M A, ALI H A. Dynamic power management techniques in multi-core architectures: A survey study[J]. Ain Shams Engineering Journal, 2015.

致　谢

在研究学习期间，我有幸得到了三位老师的教导，他们是：我的导师，中国科大 XXX 研究员，中科院 X 昆明动物所马老师以及美国犹他大学的 XXX 老师。三位深厚的学术功底，严谨的工作态度和敏锐的科学洞察力使我受益良多。衷心感谢他们多年来给予我的悉心教导和热情帮助。

感谢 XXX 老师在实验方面的指导以及教授的帮助。科大的 XXX 同学和 XXX 同学参与了部分试验工作，在此深表谢意。

在读期间发表的学术论文与取得的研究成果

已发表论文

1. Zhen Wang, Xi Li, Chao Wang, Zhinan Cheng, Jiachen Song, and Xuehai Zhou. Rethinking Energy-Efficiency of Heterogeneous Computing for CNN-Based Mobile Applications[C]// 15th IEEE International Symposium on Parallel and Distributed Processing with Applications(IEEE ISPA 2017)

2. Zhen Wang, Zhinan Cheng, Xi Li, Chao Wang, Xianglan Chen, and Xuehai Zhou. Building A Game Benchmark for Cooperative CPU-GPU with Pseudo User-interaction[C]// 15th IEEE International Symposium on Parallel and Distributed Processing with Applications(IEEE ISPA 2017)

参与的主要项目

1. 寒武纪智能芯片系统