哈爾濱工業大學

毕业设计（论文）开题报告

题　目：题目

专　　业　数据科学与大数据技术

学　　生　王翰坤

学　　号　**1183710106**

指导教师　苗东菁; 俞凯

日　　期　**2021** 年 **11** 月 **17** 日

哈尔滨工业大学教务处制

1. 课题来源及研究的目的和意义

在包括图像识别、自然语言处理、机器人、自动驾驶等许多应用场景中，深度神经网络 (Deep Neural Network, DNN) 已成为解决复杂问题的重要手段。在不同的需求下，人们对深度前馈神经网络 (或称 DNN 推理) 的需求可能有很大区别。例如，数据中心的图像识别系统往往要求较高的吞吐量但愿意牺牲一定的准确度，而自动驾驶系统中则对功耗和延时极为敏感。相比于通用处理器和 FPGA，定制化的 DNN 推理加速器能够灵活地提供设计方案，使之更贴合各类应用场景的性能和功耗需要。当前，已有若干关于定制化加速器架构的工作，它们的共同点是集成了大量的乘-加 (Multiply-and-Accumulate, MAC) 单元，但在计算单元网络连接、内存层级和数据流等方面有相当大的差异。这表明，DNN 推理加速器有着庞大的硬件设计空间 (Hardware Design Space)。然而，ASIC 的工程成本高昂且设计周期漫长，在 DNN ASIC 需求巨大且差异显著的当下，若人工逐个尝试每个设计点并加以验证，时间和经济成本都是无法接受的。因此，如何实现 DNN 加速器设计的自动化成为 DNN ASIC 领域的重大课题[1]。

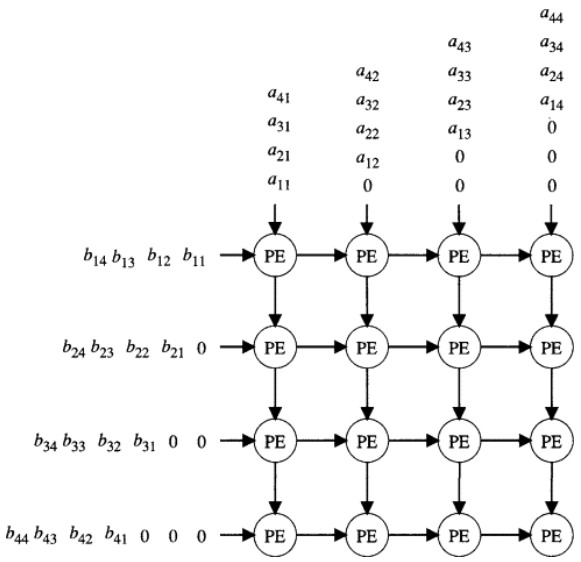


图 1: 脉动阵列

脉动阵列 (Systolic Array, SA) 是一种由相同的互连的计算单元 (Processing Element, PE) 组成的架构，如图1所示。它可用于许多应用程序，如线性代数[2]、机器学习[3] 和动态规划[4] 等，是资源利用率最高、可伸展性最强、功耗最优的架构之一，被 NVIDIA TPU 等许多卓有成效的深度学习硬件采用。通过局部连接和模块化 PE，这种设计可以轻松扩展到整个高频芯片。多面体模型 (Polyhedral Model) 是循环编译优化技术的重要数学模型[5,6]，也常用作脉动阵列结构的刻画工具，广泛

应用于脉动阵列的自动生成中。

本课题中，我们拟设计一种基于多面体模型和脉动阵列的深度前馈网络加速器自动生成系统。该系统的输入是 (深度前馈网络, 资源约束, 设计指标约束)，输出是经过自动调优的、针对该网络的、满足资源约束和设计指标约束的加速器架构设计和数据流映射。只要给定设计参数，该系统的生成和调优过程完全黑盒，预期将大大缩短 DNN ASIC 设计环节的耗时，降低相关工业环节的技术门槛。

2. 国内外在该方向的研究现状及分析

神经网络是一类蕴含着极高并行化潜能的算法，DNN 推理加速的核心手段就在于提高数据运算和传输的并行性。例如，全连接 (Full Connected, FC) 和卷积 (Convoltion) 层具有拓扑并行性 (topological parallelism)，因为它们进行的 MAC 操作不存在数据依赖，可以并行执行。迎合这一特点，许多并行计算相关的理论和技术在神经网络上得以发扬光大[7]。

另一方面，有关研究也指出，神经网络的性能瓶颈并非来自于 MAC 运算，而是来自于数据存取的过程[8]。从 DRAM 中进行一次读取数据的能耗比进行一次 MAC 运算高约 1~2 个数量级。因此，如何针对神经网络特点，采用合适的数据流模式、最大化数据复用，也成为了 DNN ASIC 设计中的焦点之一[9]。

DNN 推理加速器设计空间过于庞大的问题，不同的工作有着各自的的设计空间探索 (Design Space Exploration, DSE) 方案，但大体上可以按照循环优化方法、数据流模式和硬件资源约束等三个维度进行划分[10]。

Chen *et al.*[11] 提出一种高能效、可配置的神经网络加速器 Eyeriss。针对神经网络运算过程中数据搬运时间能耗开销大的问题，该加速器采用行静止 (Row Stationary, RS) 和数据压缩两种办法，将 AlexNet 的卷积层能耗效率提高了 2.5 倍。该文中提出的新型数据复用及其映射的方法成为为后来诸多加速器生成器借鉴。Eyeriss 基本架构如图 2 所示，它包含四个基本要素：片外内存 (如 DRAM)、片上全局缓存、计算单元 PE 及其构成的片上网络 (Network-on-Chip, NoC)。后续工作的 ASIC 设计也基本在这一框架之内。其作者还在此基础上提出 Eyeriss v2[12]，在其中设计了一种在低数据复用情况下依然提供高带宽和较高 PE 利用率的 NoC，使之更适应紧凑型和稀疏型的 DNN。

Cong *et al.*[13] 提出一种编译框架 PolySA，它利用多面体模型实现了 FPGA 上脉动阵列的端到端编译。其前端支持的算子为静态边界且循环体中仅含一条仿射表达式的循环，包括矩阵乘法、CNN 卷积等。以矩阵乘法为例，给定约束和指标，PolySA 自动生成其所有可行的脉动阵列仅需 26 分钟。

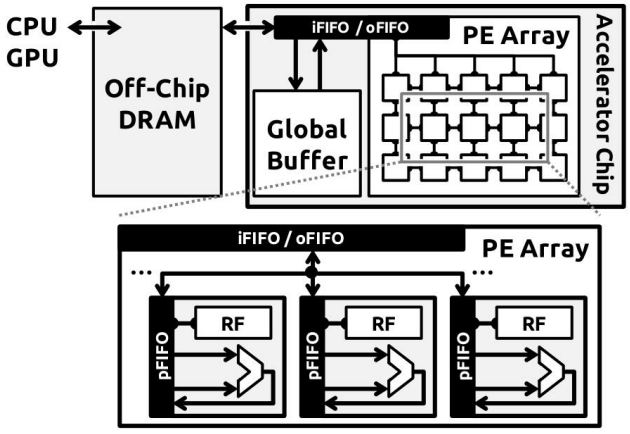


图 2: 典型的 DNN ASIC 架构

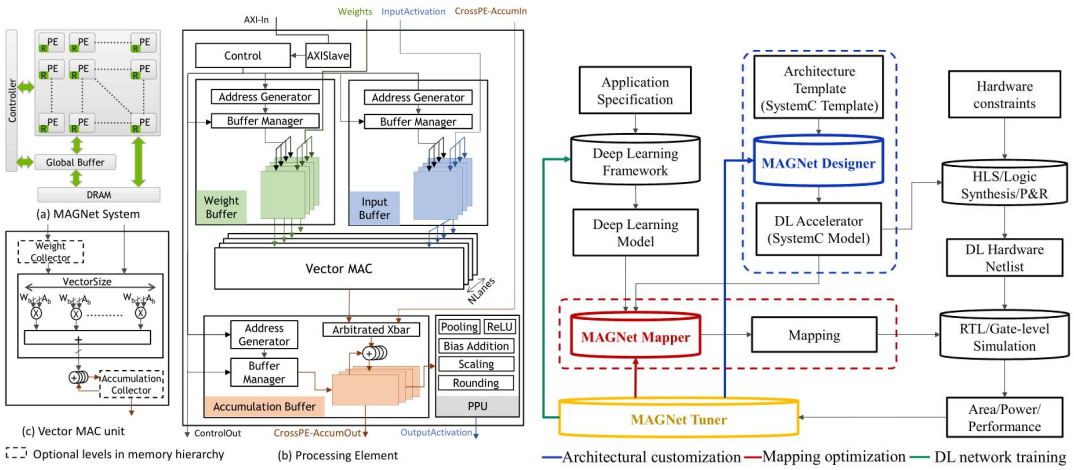
Wang *et al.*[14] 提出的 AutoSA 同样以多面体模型为手段，相比 PolySA 支持了包括 LU 分解在内的更多算子，并对 SIMD 向量化等提供了支持。但是，由于 FPGA 平台定制化程度有限，PolySA 和 AutoSA 均未能将内存层级、数据流模式等要素纳入设计。

NVIDIA 早前推出过开源的深度学习加速器架构 NVDLA[15]，旨在简化集成和可移植性。硬件支持各种 IoT 设备，促进设计深度学习推理加速器的设计的标准化和模块化。它的 PE 仅支持向量级的 MAC 运算。

Venkatesan *et al.*[16] 提出了一种模板化的神经网络加速器生成器 MAGNet，其架构模板和全局 DSE 框架如图3所示。它以神经网络组成的目标应用程序以及硬件约束作为输入，产生用于神经网络加速器 ASIC 的可综合 RTL，以及用于在所生成的硬件上运行目标网络的有效映射。对于模板化后的设计空间，MAGNet 用随机采样和贝叶斯方法进行优化。不过，MAGNet 的 PE 结构同样仅支持向量级 MAC 运算。

Xi *et al.*[17] 提出了可模拟端到端深度学习应用程序的 DSE 框架 SMAUG，旨在使 DNN 研究人员能够迅速评估不同的加速器和 SoC 设计。但 SMAUG 的硬件设计空间是固定的。

以上 DNN 加速器及其生成器的设计和评估之间往往是孤立的，未能将现实环境中的跨栈和系统级效应纳入考虑。针对这一问题，Genc *et al.*[18] 提出了一种全栈 DNN 加速器生成器 Gemmini。Gemmini 同样是用架构模板生成 ASIC 加速器，支持灵活的编程堆栈和完整的 SoC，具有可捕获系统级效果的共享资源。它生成的加速器在高性能 CPU 上可提供最高达三个数量级的加速。但是，Gemmini 并不支持针对设计参数的自动调优。



(a) Picture a title (b) Picture b title

图 3: MAGNet 架构

Mei *et al.*[19] 提出的 DSE 框架 ZigZag 支持非均匀映射和更多映射搜索策略，大大拓展了设计空间，且提升了代价预估分析模型和对分级内存支持的的精度。

综上所述，将多面体-脉动阵列架构应用于 DNN ASIC 设计是有迹可循且具备较高研究价值的。

3. 主要研究内容

本课题将探索一种新型的 DSE 框架，我们期望把多面体-脉动阵列架构和分级内存结合起来，并分析各种数据流模式、循环分解方法和搜索策略在这套框架下的性能。

4. 研究方案

5. 进度安排，预期达到的目标

进度安排如下：

- 2021.11
-

6. 课题已具备和所需的条件、经费

7. 研究过程中可能遇到的困难和问题，解决的措施

8. 主要参考文献

[1] Capra M, Bussolino B, Marchisio A, et al. Hardware and Software Optimizations for Accelerating Deep Neural Networks: Survey of Current Trends, Challenges, and the Road Ahead[J/OL]. IEEE Access, 2020, 8: 225134-225180. https://ieeexplore.ieee.org/document/9269334/.

[2] Moss D J, Krishnan S, Nurvitadhi E, et al. A Customizable Matrix Multiplication Framework for the Intel HARPv2 Xeon+FPGA Platform: A Deep Learning Case Study[C/OL] // Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. Monterey CALIFORNIA USA: ACM, 2018: 107-116. https://dl.acm.org/doi/10.1145/3174243.3174258.

[3] Jouppi N P, Young C, Patil N, et al. In-Datacenter Performance Analysis of a Tensor Processing Unit[C/OL] // Proceedings of the 44th Annual International Symposium on Computer Architecture. Toronto ON Canada: ACM, 2017: 1-12. https://dl.acm.org/doi/10.1145/3079856.3080246.

[4] Khailany B, Ren H, Dai S, et al. Accelerating Chip Design With Machine Learning[J/OL]. IEEE Micro, 2020, 40(6): 23-32. https://ieeexplore.ieee.org/document/9205654/.

[5] Benabderrahmane M-W, Pouchet L-N, Cohen A, et al. The Polyhedral Model Is More Widely Applicable Than You Think[G/OL] // Hutchison D, Kanade T, Kittler J, et al. Compiler Construction: Vol 6011. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010: 283-303. http://link.springer.com/10.1007/978-3-642-11970-5_16.

[6] Bastoul C, Cohen A, Girbal S, et al. Putting Polyhedral Loop Transformations to Work[G/OL] // Rauchwerger L. Languages and Compilers for Parallel Computing: Vol 2958. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004: 209-225. http://link.springer.com/10.1007/978-3-540-24644-2_14.

[7] Sze V, Chen Y-H, Yang T-J, et al. Efficient Processing of Deep Neural Networks: A Tutorial and Survey[J/OL]. Proceedings of the IEEE, 2017, 105(12): 2295-2329. http://ieeexplore.ieee.org/document/8114708/.

[8] Chen T, Du Z, Sun N, et al. DianNao: a small-footprint high-throughput accelerator for ubiquitous machine-learning[J/OL]. ACM SIGARCH Computer Architecture

News, 2014, 42(1): 269-284. https://dl.acm.org/doi/10.1145/2654822.2541967.

[9] Chen Y-H, Emer J, Sze V. Using Dataflow to Optimize Energy Efficiency of Deep Neural Network Accelerators[J/OL]. IEEE Micro, 2017, 37(3): 12-21. http://ieeexplore.ieee.org/document/7948671/.

[10] Yang X, Gao M, Liu Q, et al. Interstellar: Using Halide's Scheduling Language to Analyze DNN Accelerators[J/OL]. Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, 2020: 369-383. http://arxiv.org/abs/1809.04070.

[11] Chen Y-H, Emer J, Sze V. Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks[C/OL] // 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA). Seoul, South Korea: IEEE, 2016: 367-379. http://ieeexplore.ieee.org/document/7551407/.

[12] Chen Y-H, Yang T-J, Emer J S, et al. Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices[J/OL]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2019, 9(2): 292-308. https://ieeexplore.ieee.org/document/8686088/.

[13] Cong J, Wang J. PolySA: polyhedral-based systolic array auto-compilation[C/OL] // Proceedings of the International Conference on Computer-Aided Design. San Diego California: ACM, 2018: 1-8. https://dl.acm.org/doi/10.1145/3240765.3240838.

[14] Wang J, Guo L, Cong J. AutoSA: A Polyhedral Compiler for High-Performance Systolic Arrays on FPGA[C/OL] // The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. Virtual Event USA: ACM, 2021: 93-104. https://dl.acm.org/doi/10.1145/3431920.3439292.

[15] NVIDIA. NVDLA[EB/OL]. 2018. http://nvdla.org/.

[16] Venkatesan R, Raina P, Zhang Y, et al. MAGNet: A Modular Accelerator Generator for Neural Networks[C/OL] // 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). Westminster, CO, USA: IEEE, 2019: 1-8. https://ieeexplore.ieee.org/document/8942127/.

[17] Xi S L, Yao Y, Bhardwaj K, et al. SMAUG: End-to-End Full-Stack Simulation Infrastructure for Deep Learning Workloads[J/OL]. ACM Transactions on Architecture and Code Optimization, 2020, 17(4): 1-26. https://dl.acm.org/doi/10.1145/3424669.

[18] Genc H, Kim S, Amid A, et al. Gemmini: Enabling Systematic Deep-Learning

Architecture Evaluation via Full-Stack Integration[J/OL]. arXiv:1911.09925 [cs], 2021. http://arxiv.org/abs/1911.09925.

[19] Mei L, Houshmand P, Jain V, et al. ZigZag: Enlarging Joint Architecture-Mapping Design Space Exploration for DNN Accelerators[J/OL]. IEEE Transactions on Computers, 2021, 70(8): 1160-1174. https://ieeexplore.ieee.org/document/9360462/.