中国科学: 信息科学 2019 年 第 49 卷 第 3 期: 256–276
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
面向智能应用的定制计算加速器技术专题 • 评述

基于通用向量 DSP 的深度学习硬件加速技术

王慧丽, 郭阳*, 屈婉霞

国防科技大学计算机学院, 长沙 410073
* 通信作者. E-mail: guoyang@nudt.edu.cn

收稿日期: 2018–12–12; 接受日期: 2019–02–26; 网络出版日期: 2019–03–19

国家自然科学基金 (批准号: 61832018, 61572025) 资助项目

摘要 随着深度学习在众多领域发挥着越来越重要的作用, 如何设计高性能、低功耗、低延迟的深度学习硬件加速器成为体系结构领域的研究热点. 本文基于深度学习算法模型的结构和优化方法, 分析了深度学习硬件实现中面临的困难和挑战, 并对比当前主流的深度学习硬件加速平台的优势和不足, 提出了基于飞腾 – 迈创通用向量 DSP 的深度学习硬件加速方案, 对其向量广播、矩阵转换等加速技术进行了阐述. 并围绕目前通用向量 DSP 硬件加速的不足, 对兼顾通用向量计算和专用深度学习计算的可重构计算阵列等优化技术进行了深入的探讨与研究.

关键词 深度学习, 体系结构, 硬件设计, 加速器, 数字信号处理器 (DSP)

1 引言

2016 年, Google DeepMind 团队研发的人工智能程序 AlphaGo 借助智能加速芯片 TPU 取得多场全球顶级 “人机大战” 的胜利, 使人工智能技术迅速进入大众的视线, 并掀起了几年来热度不减的人工智能研究热潮. 深度学习的概念最早由 University of Toronto 的 Hinton 等 [1] 于 2006 年提出, 指基于样本数据通过一定的训练方法得到包含多个层级的深度网络结构的机器学习过程. 深度学习包括训练和推理两个阶段. 训练阶段需要大量的样本, 属于计算密集型运算, 往往使用大规模的高性能处理器如 CPU 或者 GPU, 通过最大程度发掘算法的并行性来进行训练加速. 推理则是将训练好的参数加载到特定的现场设备上 (如手机、机器人、无人机等设备), 对样本之外的输入数据进行判断推理. 虽然计算量相对训练阶段小, 但是通常工作在现场终端, 对硬件的性能、功耗, 以及延迟要求较高.

深度学习算法随着应用领域不断的发展演化, 如图 1 所示, 仅 CNN 算法模型的演变, 就出现了众多形式的算法模型. 从最原始的 BP 算法模型, 到经典的 7 层 LeNet5 [2], 再到 2015 年获得 ImageNet 冠军的具有 152 层深度的微软深度残差网络 ResNet [6], 其结构和深度都发生了很大的改变. 而硬件研发具有时间周期长的特点, 这使硬件加速平台很难灵活快速地跟随算法的发展演变. 不管是云端, 还

引用格式: 王慧丽, 郭阳, 屈婉霞. 基于通用向量 DSP 的深度学习硬件加速技术. 中国科学: 信息科学, 2019, 49: 256–276, doi: 10.1360/N112018-00288
Wang H L, Guo Y, Qu W X. Deep learning hardware acceleration based on general vector DSP (in Chinese). Sci Sin Inform, 2019, 49: 256–276, doi: 10.1360/N112018-00288

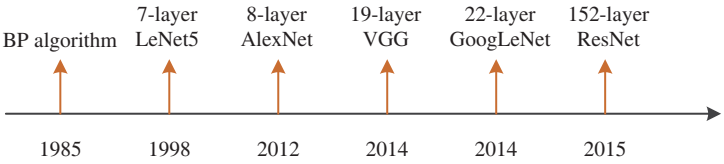

图 1 (网络版彩图) CNN 算法模型的发展演化 [2~5]

Figure 1 (Color online) Evolution of CNN algorithms [2~5]

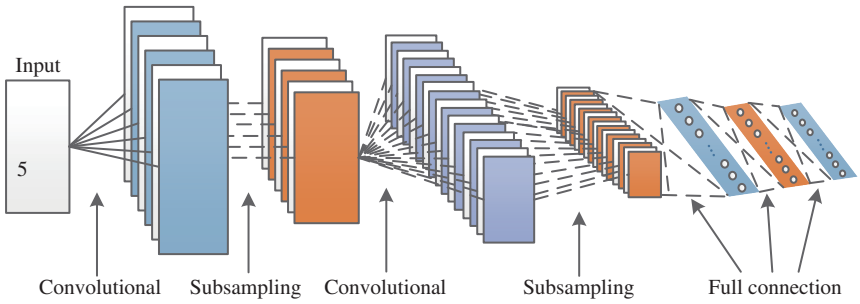

图 2 (网络版彩图) LeNet5 深度学习算法模型

Figure 2 (Color online) Algorithm model of LeNet5

是终端, 不管是传统体系结构, 还是新兴仿生物学技术, 各种硬件加速平台基于自身的竞争优势, 纷纷推出了众多高性能低功耗的硬件加速方案.

本文介绍了深度学习算法的发展和优化, 分析了当前主流的硬件加速技术和各种硬件加速平台的优势和不足; 最终, 基于以上的分析和结论, 提出了基于通用向量 DSP 的深度学习的硬件加速方案, 并针对其不足, 提出了新的改进方法.

2 深度学习硬件加速技术

2.1 深度学习核心算法分析

深度学习算法一般指层数包含 3 层以上的算法. 图 2 为经典的 LeNet5 算法模型: 一共具有 7 个层, 其中包含 2 个卷积层、2 个池化层、2 个全连接层, 以及 1 个分类层. LeNet5 对手写字符的识别精度可达 99.5%. 相比其他几个网络层, 2 个卷积层集中了整个网络 90% 以上的运算量.

在绝大多数的深度学习算法中, 卷积层的运算量占总运算量的 90% 以上, 因此卷积层算法是 CNN 算法的核心, 对其进行加速是提升 CNN 算法性能的关键. 一个三维卷积层运算过程如图 3 所示, 输入特征图像为 $(W\times H\times C)$, 卷积核为 k 组 $(R\times S\times C)$, 输出特征图像为 k 组 $(W-R+1)\times(H-S+1)$. 每一个输出特征图像都是由所有的输入特征图像和它们对应的卷积核卷积运算后的结果叠加得到, 输出特征图像的大小由输入特征图像大小以及卷积核大小共同确定. 卷积层本质是乘累加运算, 可表示如下:

$$O[(k,x,y)]=\sum_{c=0}^{C-1}\sum_{r=0}^{R-1}\sum_{s=0}^{S-1}F[(k,c,r,s)]\times I[(c,x+r,y+s)],$$
$$0\leqslant k<K,\ 0\leqslant x<W-R+1,\ 0\leqslant y<H-S+1, \tag{1}$$

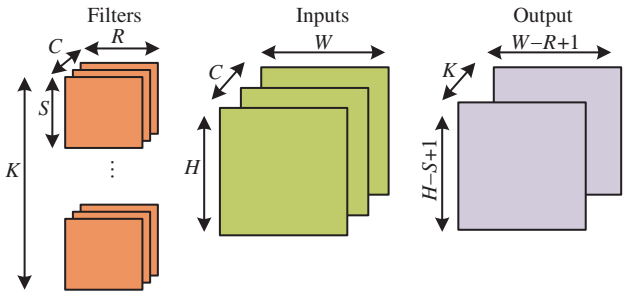

图 3 (网络版彩图) 多维卷积算法模型

Figure 3 (Color online) Multidimensional convolution algorithm

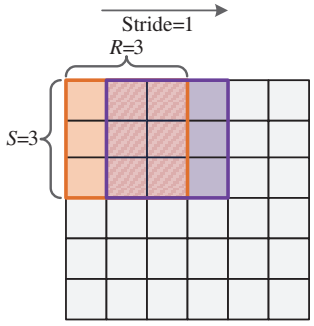

图 4 (网络版彩图) 卷积中的数据复用

Figure 4 (Color online) Data repetition in convolution

其中, O, I 和 F 分别为输出、输入和卷积核, 本级的输出对应下一级的输入.

卷积层运算过程中, 经常要反复读取运算相关数据, 而这些数据中存在大量重复, 反复读取会造成不必要的带宽浪费. 以图 4 所示卷积为例, 卷积核大小为 3×3, 跨步为 1, 相邻两次滑动中有 6 个像素被重用. 假设卷积核大小为 $S \times R$, 在跨步为 1 时, 相邻两次计算的重复数据为 $S \times (R-1)$. 因此选择合适的运算次序, 提高数据的重用性, 减少反复读取相同输入数据的次数, 可大幅降低带宽要求.

2.2 深度学习硬件加速技术

基于对深度学习核心算法的分析, 目前深度学习硬件加速的重点是对卷积算法进行加速, 主要采用软硬件联合的优化方法、运算单元的优化方法, 以及基于存储的优化方法.

2.2.1 软硬件联合的硬件加速技术

硬件的加速设计离不开算法的优化设计. 在进行硬件设计的同时, 需要结合软件的需求及算法的结构进行. 在数据进入硬件运算之前, 先进行有利于提高硬件效率的算法及软件层次的优化; 而硬件的设计, 也要适应算法结构和软件框架. 基于算法优化的软硬件联合加速技术, 主要采用降低精度、发掘稀疏性, 以及数据重排等方法.

(1) 降低精度. 在不降低正确率的前提下, 通过使用低位宽定点数代替高位宽浮点数, 可以大大减少运算和访存开销. 图 5 为 LeNet 在改变精度和位宽时, 其算法正确率的变化. 由图 5 可以看出, 由定点数代替浮点数, 位宽降至 16 位时, 正确率几乎不受影响. 硬件上, 随着位宽的降低, 其功耗也可大幅降低.

在当前主流的深度学习硬件加速器中, 位宽可以大幅降低到 16 位及以下. 另外, 由于不同算法对精度的要求不同, 不同层的计算精度也可能存在不同, 因此硬件通常支持混合精度的运算. 如 NVIDIA 的 Tensor Core, 同时支持 16 位和 32 位的精度运算, 兼顾了算法的精确度和效率 [7].

(2) 发掘稀疏性. 通过剪枝、复用、跳过 0 值等方法, 发掘算法中的稀疏性, 减少硬件中不必要的运算和访存, 可以大大提高硬件的效率 [8,9]. 目前几个比较典型的加速器中都支持这种算法压缩后的稀疏运算加速, 如 Eyeriss [10], SnaPEA [11], UCNN [12] 和 Cambricon-X [13] 等. 以上设计除了在算法上发掘稀疏性之外, 也通过相应的硬件设计, 如稀疏运算、分组运算等方法, 实现了对稀疏数据的加速. 从计算量的角度直观来看, 算法压缩能够大幅提高硬件的性能. 但有研究表明, 通过压缩的硬件设计

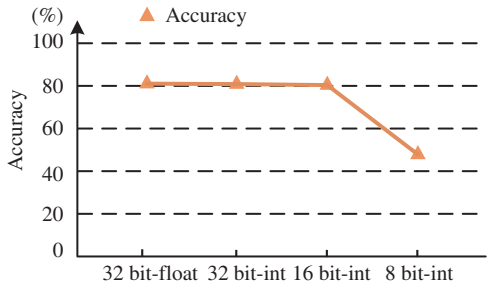

图 5 (网络版彩图) LeNet 算法的精度、位宽与正确率的关系

Figure 5 (Color online) Relationship among precision, bitwidth, and accuracy in LeNet algorithm

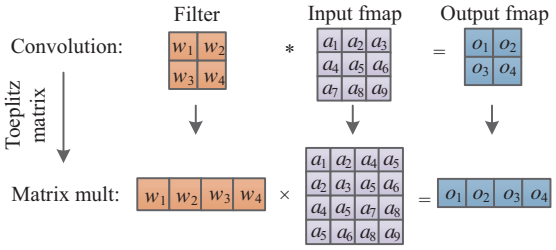

图 6 (网络版彩图) Toeplitz 矩阵方法加速卷积运算

Figure 6 (Color online) Acceleration of convolution with Toeplitz matrix

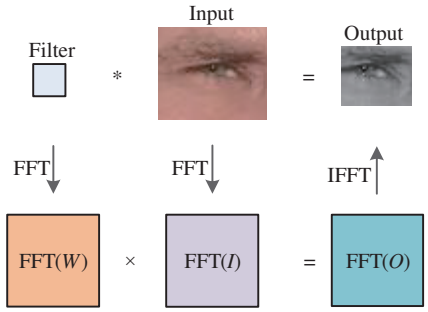

图 7 (网络版彩图) FFT 方法加速卷积运算

Figure 7 (Color online) Acceleration of convolution with FFT

有可能会出现性能的下降 [14, 15], 压缩编码和解码也增加了存储开销. Scalpel [16] 中针对不同并行度的硬件, 提出了两种解决方案: 一是基于低并行度的 SIMD 结构, 采用分组索引, 一条 SIMD 指令可以读取多个权重值; 二是针对高并行的硬件, 采用节点剪枝的方法, 既不完全去掉参数又可以将它的影响降到最小.

(3) 数据重排. 在深度学习中, 卷积运算一般占了很大比例. 卷积运算具有计算量大、数据复用率高、访存不规则的特点. 因此, 在对卷积运算进行优化时, 往往将其输入和权值进行重新组合排序. 目前主要使用的输入重排方法有 Toeplitz 矩阵方法和 FFT 方法 [17]. Toeplitz 矩阵方法如图 6 所示, 将卷积核展开为向量后, 与进行了 Toeplitz 变换的输入矩阵进行向量矩阵乘法操作. FFT 方法如图 7 所示, 通过对原始数据进行 FFT 变换, 将卷积运算转换为频域的矩阵乘运算. 矩阵乘完成后, 再将结果进行反 FFT 运算, 得到卷积结果. 这两种方法都达到了减少运算量, 规整访存模式的效果. 另外, Winograd 算法 [18] 和 Strassen 算法 [19], 以及一些边缘计算方法 [20], 也可以将卷积运算进行变换达到减少乘法操作的效果. 对于性能受限于计算资源的 FPGA, 快速 Winograd 算法可以大大降低算术复杂度, 高效实现基于 FPGA 的 CNN 加速设计 [21].

2.2.2 脉动阵列运算单元

脉动阵列在 1982 年由 Kung 提出 [22]. 脉动阵列具有访存和运算的优势, 通过使数据在相对简单的 PE 阵列之间流动, 最大限度地实现数据复用, 从而减小对存储带宽的需求, 在较小的存储带宽的情况下实现较高的运算吞吐率. 如图 8 所示, Google TPU 采用了二维的脉动架构 [23]: 它是一个 256×256

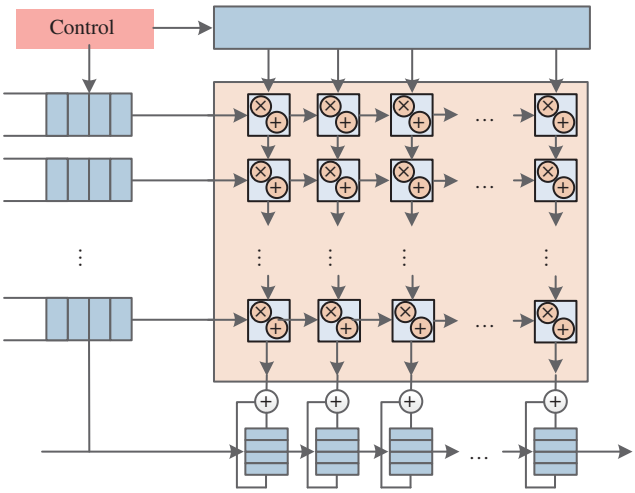

图 8　(网络版彩图) TPU 中的脉动阵列

Figure 8　(Color online) Systolic array in TPU

的脉动阵列. 而整个芯片的其他部分均围绕这个脉动阵列来进行设计. 采用脉动阵列的硬件设计还有 Eyeriss, ShiDianNao [24] 等.

2.2.3　基于存储的深度学习加速技术

在传统的 von Neumann 体系架构中, 无论计算单元有多高效, 数据都必须由存储单元 (内存或外存) 搬移到计算单元中. 大量的并行计算意味着大量的数据搬移, 当硬件的计算能力达到一定程度时, 访存开销成为限制性能的瓶颈. 因此, 新型存储结构 (如 ReRAM, STT-RAM 等非易失性存储器) 以及存储工艺 (如三维堆叠存储等) 得到研究人员的关注. ReRAM 以其特有的交叉网络结构和多比特存储性质, 可以显著提升深度学习硬件的计算效率. 但是 ReRAM 工艺尚不成熟, 存在计算与存储精度受限、ReRAM 阵列规模受限等问题. 混合立方存储器 (hybrid memory cube, HMC) 作为先进的三维堆叠的多块 DRAM 存储器, 拥有更高的性能、更低的功耗及更小的尺寸. NVIDIA 和 Google 等厂商在自己的产品中使用了高带宽存储器 (high bandwidth memory, HBM). HBM 使得片上缓存容量从 MB 级别提升到 GB 级别, 可以将整个深度学习模型放到片上, 带宽不再受限于芯片 IO 引脚的制约. 但是, 三维堆叠存储体的价格与其高性能成正比, 在设计处理器的时候, 成本开销很大.

2.2.4　其他加速技术

(1) 动态数据通道配置. STM 实现了一种支持神经网络算法的 DSP+CONV 硬件加速器, 其关键技术动态数据通道配置 (通过灵活的 DMA 设计) 的核心是一个基于 DMA 的灵活可配的数据通路 [25], 可适应多种 DCNN 的结构.

(2) 存储内计算. 这种技术一般都需要依赖新型存储器件, 也有使用现有的 Flash 技术或其他存储器件 (如 SRAM) 实现存储内计算的案例, Mythic 就是其中之一. 存内计算的目标是利用存储器件实现矩阵乘法运算, 减少数据传输所引起的带宽和功耗问题. 随着神经网络的发展, 在存储上进行创新也是适应深度学习的重要趋势.

(3) 专用加速协处理器. 将深度学习中耗时最长、计算密度最高的任务放到专用的 FPGA 或 ASIC 协处理器中, 其他部分由通用处理器完成, 从而降低通用性带来的性能损失. 专用加速协处理器与通用

处理器的协同工作, 将 RTL 开发的长周期迭代转化为指令序列的开发, 极大降低了硬件的研发周期. 如拥有比较完备指令集的寒武纪的 Cambricon 处理器 [26]、Intel Movidius 神经网络计算棒等.

(4) 新工艺. 芯片集成度的提高, 带来了功耗的不断增长. 工艺尺寸不断缩小, 同时一些新的技术如单片式三维堆叠集成 (TSV) 等 [27], 对降低功耗有重要作用.

2.3 深度学习硬件加速平台

2.3.1 通用硬件平台 CPU

作为应用最广泛的通用处理器, CPU 需要在兼顾通用性能的前提下, 在其通用架构中增加对深度学习算法的加速支持, 如, Intel 的 Knights Mill 处理器增加了对 “可变精度” 的支持 [28], 进而提高深度学习处理的性能. 另外, Intel 也通过增加协处理器来提高深度学习加速的性能. 但是, 不管是对其架构进行优化创新, 还是增加协处理器, 其功耗、面积、性能等指标会受限于其通用性能.

2.3.2 图形处理器 GPU

图形处理器 (GPU) 能够提供强大的并行计算能力特性, 使其成为神经网络硬件加速平台的首选. Oh 等 [29] 早在 2004 年就已经通过将点积转化为矩阵操作, 在 GPU 上加速神经网络. Coates 等 [30] 提出的 GPU 加速方案可以将深度学习加速 90 倍以上. NVIDIA 的 Tesla V100 内置了 5120 个 CUDA 单元, 搭载 16 GB HBM 显存, 单精度浮点性能和双精度浮点性能分别达到了 15 TFLOPS 和 7.5 TFLOPS. 虽然效率极高, 但与 CPU 类似, 由于其对图像处理的通用性能在硬件设计中所占比例大, GPU 在进行深度学习加速时, 能耗仍然高于 FPGA 和 ASIC.

2.3.3 现场可编程门阵列 FPGA

FPGA 作为一种高性能、低功耗的可编程芯片, 与通用平台 CPU 和 GPU 相比, 节省了通用功能的逻辑, 性能更高并且能耗比也更高. 与 ASIC 相比, FPGA 快速迭代的优势使其具有极短的开发周期, 灵活可扩展使其能够快速适应深度学习算法的发展与优化. 基于 FPGA 的深度学习加速研究众多, 如在 FPGA 上实现的多层感知机 [31]、可重构数据流处理器 NeuFlow [32]、结合 Winograd 算法采用行缓冲结构实现高效数据复用的 CNN 加速器 [21] 等. 在 2017 年 FPGA 会议上, Zhang 等 [33] 提出了一种在 CPU-FPGA 共享内存上, 通过快速 FFT 算法实现对卷积神经网络进行频域加速的方法. 该方法利用快速 FFT 变换将频域算法映射到 FPGA 上高度并行的基于重叠加法 (overlap-and-add, OaA) 的二维卷积结构上, 在共享内存中以一种新颖的数据布局, 实现 CPU 和 FPGA 之间高效的数据通信, 使 CNN 浮点运算次数减少 39.14%∼54.10%. 但是, FPGA 也存在一些缺点, 比如其硬件编程需要对底层硬件有一定的知识且使用硬件描述语言 (HDL) 进行开发, 具有一定的局限性, 并且因为其工作频率较低, 相对其他加速器具有较高的单位能耗和延迟.

2.3.4 专用集成电路 ASIC

相比于通用器件的低能效, 开发深度学习专用硬件已经在学术界和工业界达成共识. Pham 等 [34] 利用 IBM 的 45 nm SOI 工艺库对 FPGA 实现的 NeuFlow 进行了评估, 认为如果 NeuFlow 用 ASIC 实现, 其性能功耗比将达到 490 GOPs/W, 远大于 FPGA 实现的 14.7 GOPs/W 和 GPU 的 1.8 GOPs/W. 寒武纪系列 [35∼38]、Eyeriss [10]、TPU 系列加速器、EIE [39] 等, 均为高性能 ASIC. 尽管这些 ASIC 芯片在处理深度学习网络时, 能效高于 CPU/GPU/FPGA, 但是它们却仍然存在着一些不足: (1) 可扩展

表 1 深度学习硬件加速平台比较分析

Table 1 Comparison of deep learning hardware acceleration platform

Hardware platform	Advantages	Disadvantages	Technical solutions
CPU/GPU	Flexible programming, supporting for multiple algorithms, and have lots of technical accumulation.	For general-purpose computing, high energy consumption for deep learning acceleration.	CPU/GPU + function expansion/specialized processing module.
FPGA	Configurable, short design cycle.	Relatively high unit energy consumption and delay.	Customized for algorithm.
ASIC	Customized, performance, power consumption, and latency all have advantages.	Long development cycle, required manpower and material resources; not flexible enough.	Customized for algorithm.
Imitation biological chips	Low power consumption, consistent with neural network prototype.	Low precision, limited by current technology.	Imitate biological neural networks, using new technologies and materials.

性不足; (2) ASIC 在与其他通用芯片协同工作时, 缺乏系统级的优化考虑; (3) 工具链支持不足, 限制了 ASIC 的通用性.

2.3.5 仿生物学类脑芯片

仿生物学的脉冲神经网络芯片近些年也得到了业界的关注. 2014 年, IBM 发布了脉冲神经网络芯片 TrueNorth [40]. TrueNorth 采用 Crossbar 结构的 SRAM, 整体功耗低至 65 mW, 性能功耗比远远高于传统处理器. 高通也发布了生物启发的神经网络芯片 Zeroth 处理器 [41]. 尽管生物启发的神经网络很贴近真实的神经元细胞, 但是它在机器学习任务上的低精度以及工艺制作的复杂度, 使其难以在当前工业界得到更为广泛的应用.

2.4 小结

通过对深度学习加速技术及加速平台的对比分析, 我们总结了不同深度学习硬件加速平台之间的优势和不足, 如表 1 所示.

结合深度学习硬件加速的主要应用需求, 不难得出以下结论:

(1) 针对深度学习的硬件加速器, 在应用端仍然需要通用的硬件设计, 以计算密集型应用为主, 集控制、通信功能于一体, 并且要具有适应算法变化的灵活性.

(2) 作为整个智能处理的核心的专用加速硬件不可或缺, 这通常需要通过专用的高性能计算架构来实现.

(3) 针对以上两点, 并结合图 9 中虚线框图所示, 面向推理的应用端嵌入式硬件, 具有更高的要求: 既要兼顾通用计算和专用计算的高性能、低延迟, 又不能过多的增加其面积和功耗.

(4) 基于各种硬件平台的深度学习加速, 难以同时满足各种应用场景对深度学习算法的需求, 未来会是 CPU+GPU/TPU/ASIC/FPGA/DSP 的多芯片协同场景.

3 基于通用向量 DSP 的深度学习硬件加速

基于通用向量数字信号处理器 (DSP) 的硬件加速技术, 能够兼顾通用计算和专用加速, 以尽量小

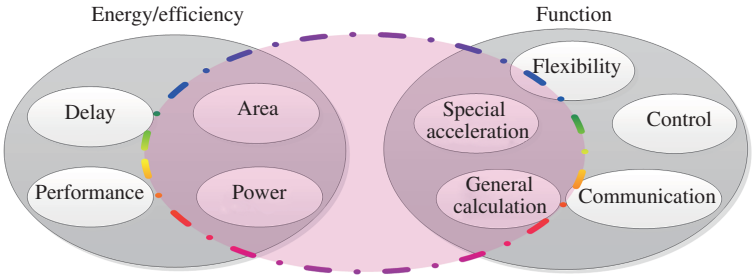

图 9 (网络版彩图) 深度学习硬件加速的应用需求

Figure 9 (Color online) Requirements of deep learning hardware acceleration

的硬件的面积和功耗开销支持深度学习. 其主要优势是:

(1) DSP 作为一种数字信号运算的专用微处理器, 在功耗和效率方面具有明显的优势. 各种复杂的数字信号处理算法都能够利用 DSP 实时实现.

(2) 通用向量 DSP 具有的丰富的硬件资源、大量的并行处理结构、强大的指令集系统, 使其具有高速灵活的数据处理能力.

(3) DSP 集信号处理、实时通信、精确控制等功能于一体, 在航空航天、4G 移动通信、雷达卫星系统, 以及医学仪器等众多领域得到了广泛的应用.

(4) DSP 通过整合自身计算资源, 通过灵活可配置可重构的设计方案, 即可在不过多增加额外功耗和延时的情况下, 实现对深度学习算法较为理想的加速性能, 同时又可以保留对通用算法的计算优势.

因此, 基于通用向量 DSP 的深度学习硬件加速符合当前深度学习硬件的发展趋势. 本节中, 我们通过对当前几款主流支持深度学习的通用向量 DSP 产品进行介绍, 对比分析几个具有影响力的 DSP 厂商的深度学习加速方案. 并基于此, 提出一种新的兼顾通用计算和专用加速的向量 DSP 设计方案.

3.1 CEVA

作为传统的 DSP IP 厂商, CEVA 的深度学习加速硬件主要是针对视觉应用: 在 XM6 图像视频处理器 [42] 的基础上, 通过 CDNN 工具包 (包括软件工具和硬件加速器) 实现嵌入式的神经网络. XM6 沿袭了 XM 系列传统的向量处理器架构, 通过协处理器接口外接深度学习硬件加速器, 兼顾通用的计算机视觉应用. 向量 DSP 一般可以一次实现两个向量 (vector) 的乘法和累加 (MAC), 以常见的 SIMD 数据并行的实现方式, 128 个 MAC 可以同时做 128 个乘累加. MAC 中乘法的位宽支持 8, 16 和 32 位的操作; 在运行深度学习算法时, 拥有 95% MAC 的利用率.

CEVA 最新推出的一款专用硬件加速系列芯片 NeuPro [43], 其结构如图 10 所示. 该系列芯片共包含 4 款芯片, 每款芯片均由 NeuPro 引擎和 NeuPro VPU 组成, 其中 NeuPro 引擎负责神经网络层的硬件实现, 包括卷积、全连通、池化和激活等; NeuPro VPU 则是高效的可编程向量 DSP, 其计算精度支持 8 位和 16 位神经网络. MAC 单元在运行时的利用率达到 90% 以上. NeuPro 的 4 款芯片峰值性能为 2~12.5 TOPS.

3.2 Synopsys

Synopsys 对深度学习中的神经网络的支持起步比较早. 它的 EV (embedded vision) 处理器架构和 CEVA 的 XM6 的架构类似, 采用了视觉处理 CPU 外接 CNN 加速器的架构. Synopsys 在其工具/IP

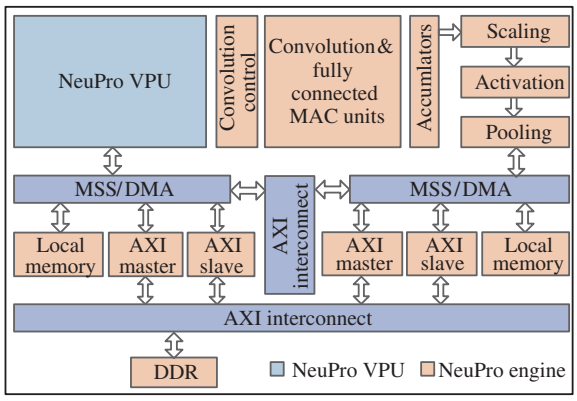

图 10 (网络版彩图) CEVA NeuPro 体系结构 [43]

Figure 10 (Color online) Architecture of CEVA NeuPro [43]

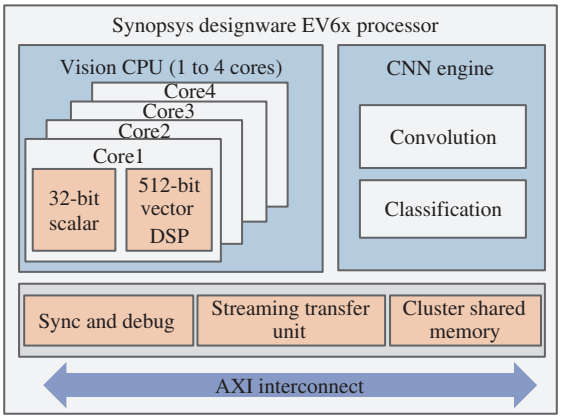

图 11 (网络版彩图) Synopsys EV6x 体系结构

Figure 11 (Color online) Architecture of Synopsys EV6x

配合方面更有优势, 如 SoC 架构层面和 EDA 的层面. 如图 11 所示, EV6x 处理器[1)]是异构多核架构, 包括 1~4 个高性能视觉 CPU. 每个视觉 CPU 都包含一个 32 位标量单元和一个 512 位宽的矢量 DSP, 支持 8/16/32 位精度运算. 此外, EV6x 处理器具有可选的 CNN 加速器, 视觉 CPU 和 CNN 加速器可以并行执行任务, 其峰值性能大于 1000 GOPS/W.

3.3 Cadence

Cadence 公司最新推出的针对车载、监控安防、无人机和移动/可穿戴设备应用的 Vision C5 DSP, 被其称为第 1 款神经网络加速 DSP[2)], 其体系结构如图 12 所示. Vision C5 DSP 为 VLIW SIMD 多核体系结构, 具有 128 路 8 位 MACs (或者 64 路 16 位 MACs)、智能 DMA, 以及 AXI4 扩展接口, 由图 12 可以看出, Vision C5 可以实现全神经网络层的计算加速 (卷积层、全连接层、池化层和归一化层). Vision C5 DSP 的功耗远低于现有的神经网络加速器, 并且, 不到 1 平方毫米的芯片面积可实现 1 TMAC/s 的计算能力 (吞吐量较 Vision P6 DSP 提高 4 倍), 基于业界知名的 AlexNet CNN Benchmark, Vision C5 DSP 的计算速度较当时的 GPU 最快提高 6 倍. 同时, Vision C5 DSP 还提供针对神经网络的单核编程模型, 并且其软件工具包与 Vision P5/P6 保持一致.

3.4 STM

STM 在 ISSCC 2017 提出了一种支持神经网络算法的 DSP+CONV 硬件加速器 [25]. 如图 13 所示, 该架构内部采用 2D DMA 以及共享 SRAM, 对卷积的支持在专用的硬件加速器中进行, 而 DSP 仅仅为神经网络提供除卷积计算以外的支持, DSP 核的运算精度为 32 位. 峰值性能达到 2.9 TOPS/W.

3.5 小结

以上介绍了几款基于 DSP 的典型深度学习加速器, 其他厂商如高通和 ARM 等, 其 DSP 加速方法与以上几种大体一致, 我们不再单独列出. 通过以上几款芯片的对比, 可以看出, 基于通用 DSP 的

1) Synopsys. DesignWare EV6x vision processors. https://www.synopsys.com/dw/ipdir.php?ds=ev6x-vision-processors.

2) Cadence. https://www.cadence.com.

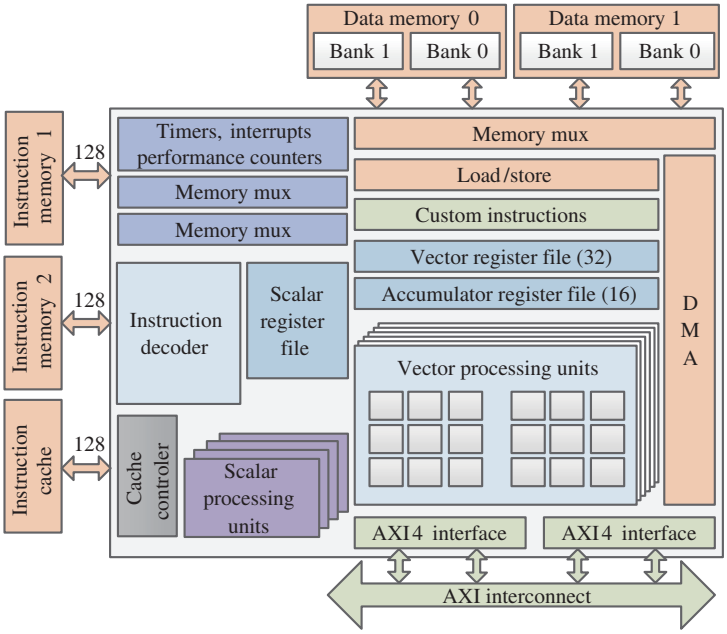

图 12　(网络版彩图) Cadence 的 Tensilica Vision C5 处理器

Figure 12　(Color online) Cadence Tensilica Vision C5

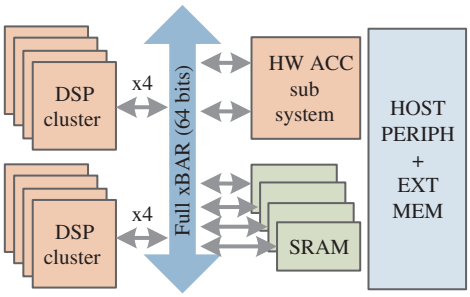

图 13　(网络版彩图) STM DSP 体系结构 [25]

Figure 13　(Color online) Architecture of STM DSP [25]

深度学习加速方法基本可以分为两大类.

(1) 通用 DSP+ 专用深度学习加速器的异构多核技术. 目前各个 DSP 厂家基本都有异构多核的 DSP, 如上文中的 CEVA XM6, Synopsys EV6x, Cadence Vision P6 和 STM. 这种加速方法可以基于已有的技术积累, 通过 DSP 的扩展接口就可以与专用深度学习加速器协同工作, 在研发周期和硬件成本上都具有优势. 但是, 由于 DSP 中本身存在大量向量运算单元, 新增加的专用处理器中也具有大量的运算单元, 虽然二者可以同时工作 (如 EV6x), 但硬件整体的开销会有所增加.

(2) 专用深度学习加速 DSP. 如上文中的 NeuPro 和 Cadence Vision C5, 专用的深度学习加速 DSP 支持深度学习的所有算法层, 但是在实际应用端, 仍然需要配合其他通用处理器平台才能发挥其高性能.

表 2 FT-Matrix DSP 与 CPU, GPU 性能比较

Table 2 Comparison among FT-Matrix DSP, CPU and GPU

Chips	Convolution: 144×5	Convolution: 16×5	Matrix×matrix: 144×144	Matrix×vector: 144
CPU	0.0013558	0.0001026	0.000802	0.0004747
GPU	0.0002323	0.0001902	0.0002809	0.0002488
Matrix	0.000218406	0.000005787	0.000026067	0.000000541
Matrix/CPU	6.207704917	17.72939347	30.76686999	877.4491682
Matrix/GPU	1.063615468	32.86677035	10.77607703	459.8890943

4 基于飞腾 – 迈创通用向量 DSP 的深度学习加速

针对 3.5 小节中对基于通用向量 DSP 的深度学习加速方法中存在的不足, 我们提出了可以兼顾通用性能和专用加速的通用向量 DSP 深度学习加速方法. 该方法基于飞腾 – 迈创通用向量 DSP 体系结构, 依托 FT-M7002 32 位通用向量 DSP 硬件平台, 结合其强大的控制、访存、向量处理、数据搬移功能, 通过精确可配置的线程控制, 复用其向量部件, 进行可重构的专用脉冲阵列的深度学习加速设计. 该设计可以在面积功耗增加较少的前提下, 实现高效的深度学习加速.

4.1 FT-Matrix 通用向量 DSP 概述

飞腾 – 迈创 (FT-Matrix) 通用向量 DSP 是由国防科技大学微电子所自主研发的面向无线通信、视频和图像处理的高性能浮点向量 DSP 核 [44]. 基于 FT-Matrix 已研制出 3 款通用 DSP 产品: FT-QMBase 浮点向量 DSP (32 位, 4 核 FT-Matrix)、FT-2000B 浮点向量 DSP (64 位, 12 核 FT-Matrix), 以及 FT-M7002 浮点向量 DSP (32 位, 2 核 FT-Matrix). 3 款芯片的不同配置和特性可适应大多数应用场景的计算需求. 本小节以 FT-M7002 DSP 为例, 介绍基于飞腾 – 迈创通用向量 DSP 的深度学习加速技术.

FT-M7002 为 VLIW SIMD 结构的双核处理器, 其体系结构如图 14 所示. 其中, 每个 DSP 核主频 1.25 Ghz, CPU 核主频 1 GHz, 具有众多可兼容的外部快速通信接口 (如 RIO, PCIE, I2C 等), 具有强大的实时通信能力. FT-M7002 DSP 能够实时快速地实现各种数字信号处理算法, 高效支持深度学习中的典型核心算法 (如卷积操作以及卷积优化后的矩阵操作). 表 2 为 I7-4790 CPU, GTX970 GPU 和 FT-Matrix 单核的部分性能简单对比, 其中, 前 3 行为处理器进行算法运算所需的时间 (单位为 s), X/Y 为 X 相对于 Y 运算速度的倍数. 从左至右分别表示 144×144 的输入、5×5 的卷积; 16×16 的输入、5×5 的卷积; 两个 144×144 的矩阵相乘; 144×144 的矩阵和 144×1 的向量相乘.

FT-M7002 DSP 具备完善的软件工具包, 提供如图 15 所示的 3 种编程模型: 标准 C 编程可以通过 C 编译器自动挖掘向量数据并生成向量指令, 提高用户程序的可移植性并缩短应用程序的开发周期; 用户也可以使用 Vector C 来获得更多的矢量性能, 实现更高效率的编程; 如果用户希望获得更高的性能, 也可以用 DSP 汇编语言编程. 利用 DSP 灵活的可编程性, 通过软硬件协同的算法优化技术与可编程方法, 可以高效支持大多数的深度学习算法.

4.2 FT-Matrix 的深度学习加速方案

FT-Matrix 核的体系结构如图 16 所示. 该 DSP 采用 11 发射超指令字 (VLIW) 结构, 其中, 指令派发负责向各部件派发指令. 标量处理单元 SPU 由标量运算单元、指令流控单元和标量数据访存单

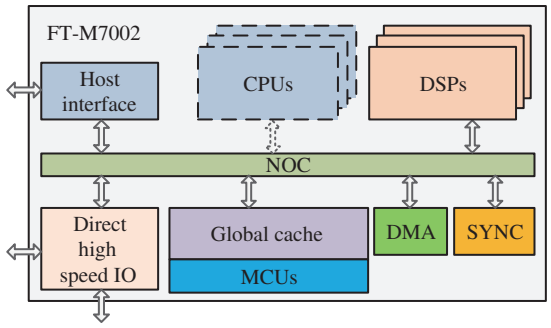

图 14 (网络版彩图) FT-M7002 DSP 多核架构

Figure 14 (Color online) Multi-core architecture of FT-M7002 DSP

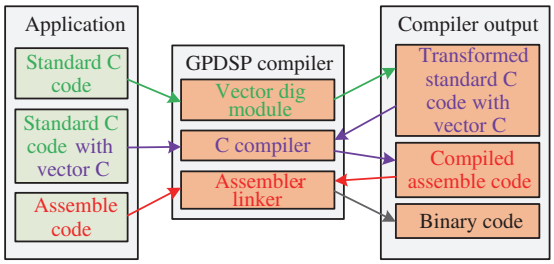

图 15 (网络版彩图) FT-M7002 DSP 的软件支持

Figure 15 (Color online) Software support for FT-M7002 DSP

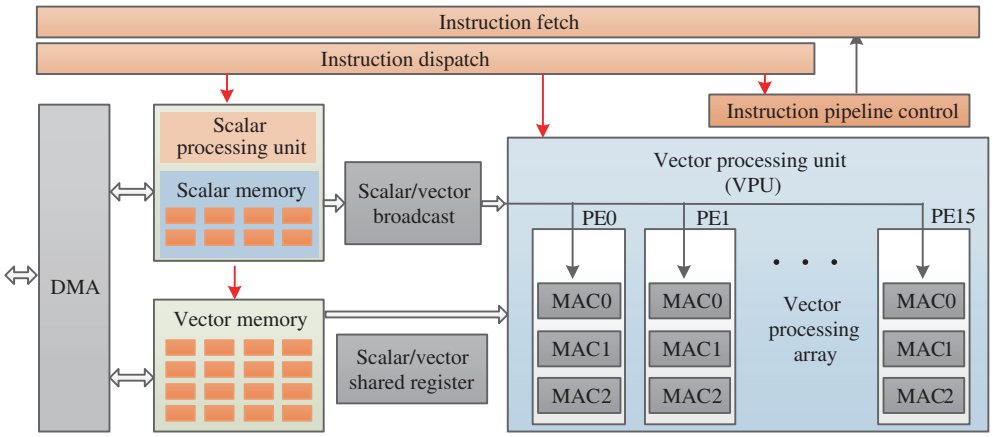

图 16 (网络版彩图) FT-Matrix 核体系结构

Figure 16 (Color online) Architecture of single core of FT-Matrix

元组成, 负责应用中串行和标量处理部分; 向量处理单元 VPU 由 16 个同构的 VPE 组成, 每个 VPE 包含 3 个乘累加 (MAC) 单元, 在通用计算模式下可并行执行 3 条向量 MAC 运算指令, 同时支持超长指令字的指令集并行开发与基于向量的数据并行开发. DSP 的片上向量存储体 AM 总容量为 512 KB, 并采用多存储体 (16 个 Bank) 的组织方式, 可同时支持两个向量数据的多粒度多模式 load/store 操作以及 SPU 和 DMA 的向量数据访问, 为 VPU 提供最大 2048 位的访存带宽, 满足 16 个 VPE 的高并行 SIMD 计算需求, 使其在进行规则的密集型计算时, 能够得到理想的峰值运算性能. 标量存储体对向量存储体读取的支持使得标量单元能够高效地处理并行处理后的数据, 或者为并行处理准备数据. 直接存储器访问 (DMA) 是 DSP 内部极其重要的数据通信方式, DMA 控制器能够后台控制完成 DSP 所有存储映射空间内的数据搬移, 支持多种模式的数据搬移.

4.2.1 适应深度学习的可变长 RISC 指令集

FT-Matrix 采用自主创新的可变长 RISC 指令集, 单周期最多可并行 11 条指令. 指令类型覆盖范围广, 如表 3 所示, 涵盖通用计算及典型深度学习运算法所需的基本 RISC 指令. 同时, FT-Matrix 支持多种精度的计算和多种粒度的访存: 支持 16/32 位定点运算以及 16/32 位浮点运算; 支持

表 3　FT-Matrix 指令集

Table 3　Instruction set of FT-Matrix

Instruction type	Main function
1. Flow control	Scalar branch, vector branch, wait, nop, etc.
2. Scalar load/stroe	Scalar load/stroe of half/one/double/quad word with linear or circular addressing.
3. Scalar MAC1 4. Scalar MAC2	Basic operation (+/−, ×, /), FMA, dot, complex multiplication, square root, elementary, functions (sine/cosine/exp/log), format conversion, floating-point logic ops etc.
5. Scalar BP	Fixed-point +/−, shift, test (==, !=, > , < , etc.), logical ops, bit ops, broadcast ops, etc.
6. Vector load/stroe 1 7. Vector load/stroe 2	(16x) vector load/store of half/one/double/quad word with linear or circular addressing.
8. Vector MAC1 9. Vector MAC2 10. Vector MAC3	(16x) vector basic operation (+/−, ×), FMA, dot, complex multiplication, data format conversion, floating-point logic ops etc.
11. Vector BP	(16x) vector fixed-point +/−, shift, test, logical ops, bit ops, shuffle, reduction, etc.

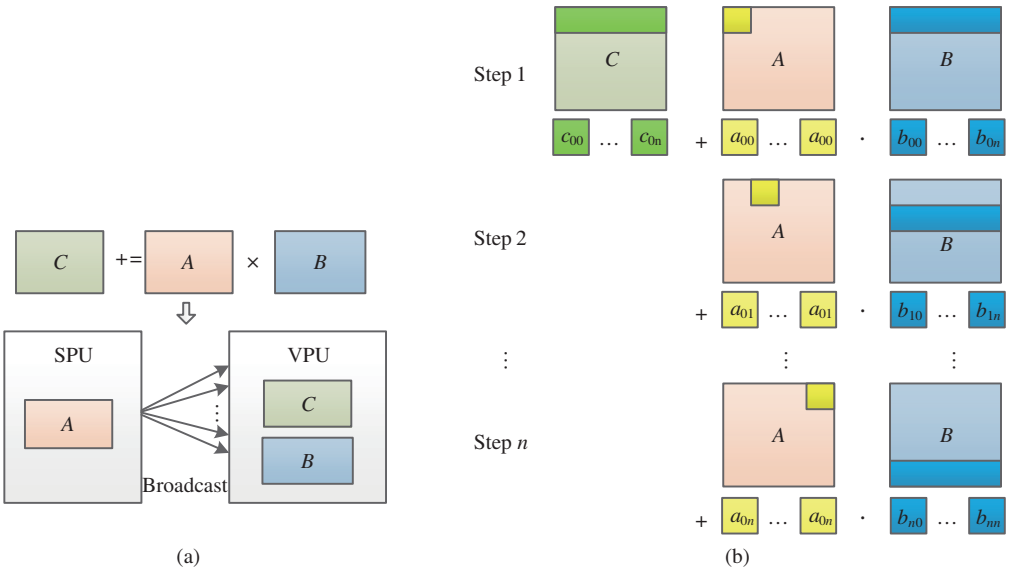

图 17　(网络版彩图) FT-Matrix 中的矩阵乘法映射

Figure 17　(Color online) Matrix multiplication on FT-Matrix DSP

8/16/32/64/128 位粒度的标量访存, 以及 32/64/128 位粒度的向量访存. FT-Matrix 在支持通用的高位宽运算同时, 也满足了对精度要求不高的深度学习推理的需求.

4.2.2　标向量数据广播功能

FT-Matrix DSP 设计了一条可以将一个标量数据复制广播到向量运算单元的指令, 该指令可以将一个标量数据复制为宽度最大为 16 的向量数据, 通过减少访存和数据复用 (一条标量访存可以得到 16 个数据), 极大地提高了深度学习的效率. 图 17 所示为通过广播指令进行矩阵算法加速的实现方法. 首先, 矩阵 A 加载在标量存储中, 矩阵 B 和结果矩阵存放在向量存储中. Step 1 中, 从矩阵 A 的第 1 行取出第 1 个标量数据, 通过标向量广播指令扩展为一组 16 宽度的向量, 并与矩阵 B 中的第 1 行进行矩阵相乘, 结果存放在 C 矩阵的第 1 行. Step 2 中, 从矩阵 A 的第 1 行中取出第 2 个标量数据, 通过

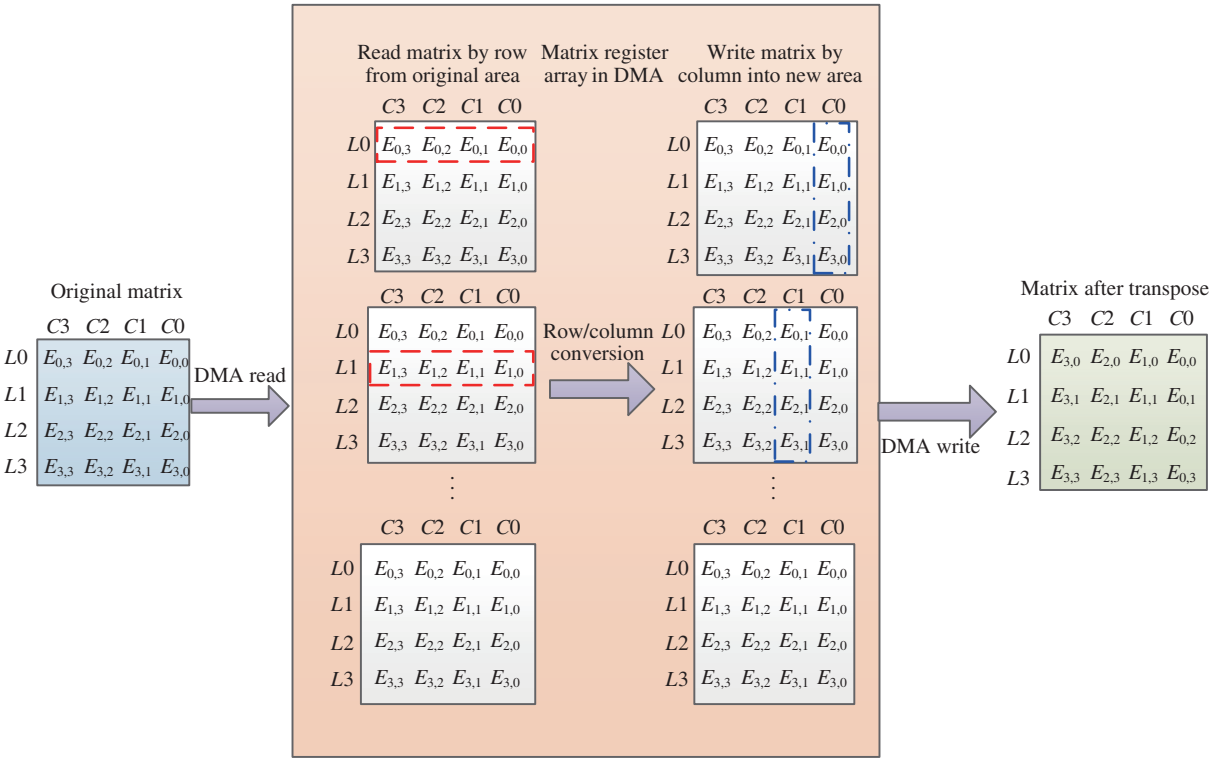

图 18　(网络版彩图) FT-Matrix 中 DMA 的矩阵转置传输

Figure 18　(Color online) Matrix transposed transmission of DMA in FT-Matrix

标向量广播指令扩展为一组 16 宽度的向量, 并与矩阵 B 中的第 2 行进行矩阵相乘, 结果与之前存放在 C 矩阵的第 1 行中间结果相加, 再次存放在 C 矩阵的第 1 行中. 依次类推, Step n 后, 即得到结果矩阵 C 的第 1 行数据. 在 Step n 之后, 按照对 A 的第 1 行的操作, 分别对 A 的每一行进行相同的操作. 得到最终的矩阵乘结果.

4.2.3　多功能 DMA 控制器

FT-M7002 中的 DMA 位于 FT-Matrix 单核外, 具有强大的数据搬移能力. 支持多核广播传输、分段传输、矩阵传输, 以及矩阵的转置传输, 这些功能在进行深度学习算法加速时, 能够迅速合理地分配数据的存储空间. 其中, 矩阵的转置传输可以对矩阵进行行列的转换, 能够灵活支持多种算法, 对深度学习中的矩阵运算具有重要意义. DMA 通过转置矩阵实现转置传输, 转置矩阵由 8 个宽度为 32 位、深度为 8 的双端口矩阵寄存器文件实现, 数据按行写入、按列读出. 其矩阵转置的过程如图 18 所示.

4.3　可配置的基于通用向量 DSP 的深度学习加速体系结构

FT-M7002 DSP 中的深度学习加速技术可以有效支持深度学习算法, 但是也存在一些不足:

(1) 对于精度要求不高的一些深度学习推理算法, 使用 8 位精度则足够保证其正确性. 而当前 DSP 使用 16/32 位的精度, 对硬件资源仍然是一种浪费.

(2) 通用 SIMD 向量 DSP 中的并行计算, 受到访存速度、并行方式的影响, 在加速不同并行度、不

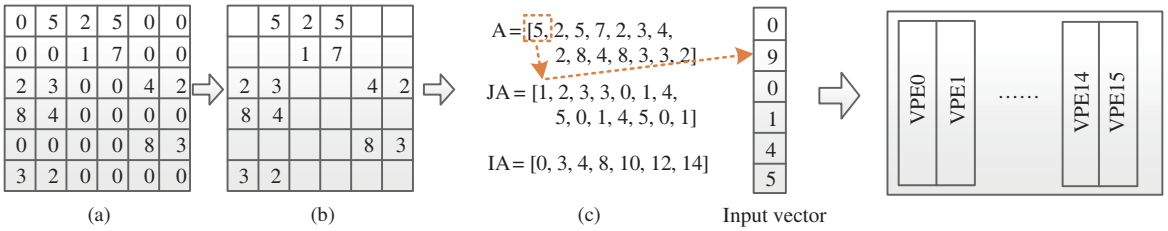

图 19 (网络版彩图) 深度学习硬件中的稀疏问题

Figure 19 (Color online) Problem of sparse matrix in deep learning hardware

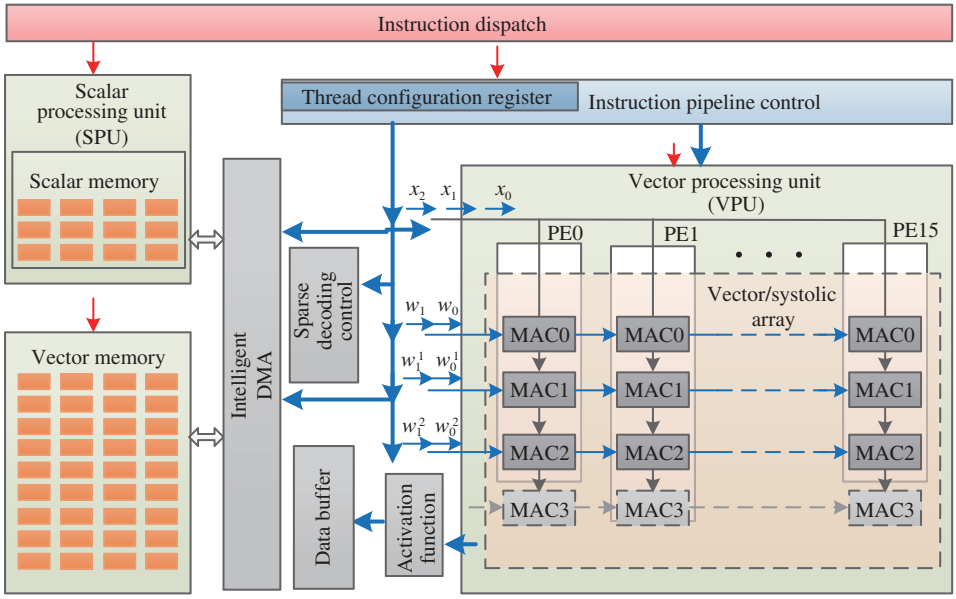

图 20 (网络版彩图) 基于 FT-Matrix 的可配置的深度学习加速体系结构

Figure 20 (Color online) Configurable deep learning acceleration based on FT-Matrix

同数据复用率, 以及对访存和数据有更高要求的深度学习算法时, 不够灵活和高效. 而如果采用 FT-M7002 DSP+ 专用协处理器的方式来完成深度学习算法运算, 虽然可以实现非常高效的加速比, 但也会导致 FT-M7002 DSP 中原有大量向量处理单元空置, 对有限的片上资源造成极大的浪费.

(3) 在深度学习算法中, 不管是软件优化还是硬件加速, 算法压缩是非常有效的优化手段. 压缩后的矩阵变得稀疏, 在硬件上会面临新的问题, 如图 19 所示, 数据在存储中是离散分布的, 不仅增加了访存的难度, 也会因硬件利用率低而限制了向量处理单元高并行计算的性能. 如果对原始矩阵的压缩率小于一个阈值, 那么这两个问题的存在会大大降低算法压缩的加速预期.

(4) 缺少一套标准的外部可扩展接口, 不利于 DSP 与 CPU 等其他处理器的协同工作.

因此, 我们针对以上不足, 在 FT-M7002 结构基础之上, 提出了如图 20 所示的可配置的基于通用向量 DSP 的深度学习加速体系结构: 指令流控、标量单元延续 FT-M7002 DSP 的结构; VPU 中的 MACs 通过可配置的方式, 在精确的线程控制下切换成深度学习专用加速模式 —— 脉冲阵列计算模式. 该模式下, 同一个 PE 间的 MACs 可以进行脉冲模式的纵向数据流动, 从而对深度学习算法进行加速. DMA 在深度学习加速模式下, 可以为脉冲计算阵列提供规整有序的数据来源. 激活函数模块在原指令集系统基础上, 提供更多类型的深度学习算法的激活函数的操作; 数据缓冲负责对脉冲阵列的

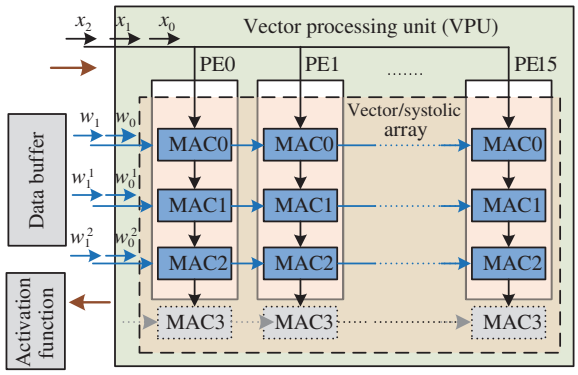

图 21　(网络版彩图) 基于 FT-Matrix VPU 的可配置计算阵列

Figure 21　(Color online) Configurable computing array based on FT-Matrix VPU

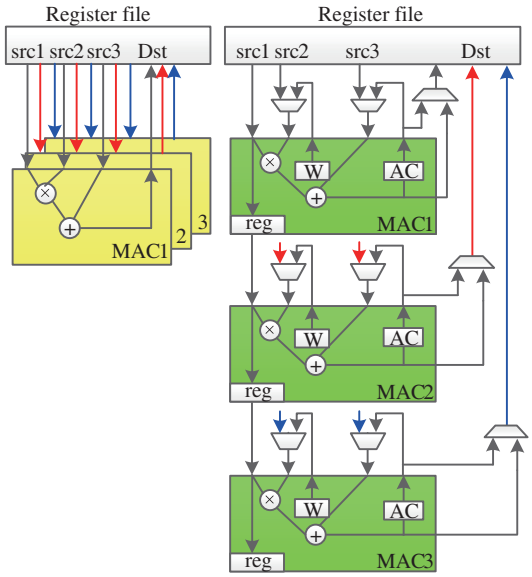

图 22　(网络版彩图) MACs 优化前后结构对比

Figure 22　(Color online) Comparison of MAC structures before and after optimization

输入/输出进行中间数据缓存及规整; 稀疏矩阵编解码则为压缩后的矩阵进行解压缩, 并生成便于访存的索引, 支持对非连续地址进行随机访存; 增加外部可扩展 AXI 标准总线协议, 支持 CPU 与 DSP 之间的通信. 具体优化方案如下:

(1) 实现更小的计算精度. 在支持 16/32 位计算的同时, 增加 8 位 SIMD 运算. 对深度学习中对精度要求不高的推理进行加速.

(2) 实现可配置的专用脉冲阵列设计. 基于单核中的并行处理单元 VPU, 进行可重构的计算阵列设计. 在进行深度学习算法加速时, 根据需要可以切换输入数据流的供给方式和 MAC 间数据流动方式, 通过复用现有的 MAC 单元, 实现兼顾通用计算和专用深度学习加速的运算单元. 如图 21, VPU 中的 MACs 可以通过线程控制, 切换成高效的深度学习专用加速模式 —— 脉冲阵列计算模式, 该模式下, 同一个 PE 间的 MACs 可以进行脉冲模式的纵向数据流动, 必要时可扩展脉动阵列的深度. 专用加速模式下的脉动阵列, 可对深度学习算法中大量存在的卷积进行加速, 极大地降低片上访存带宽.

图 22 为单个 PE 内部, 3 个 MACs 单元的优化前后结构对比. 图 22 左半部分黄色为 FT-M7002 的乘累加单元结构简图, 每个时钟周期, 3 个 MAC 单元均从寄存器文件中 (来自向量 load 操作) 获取数据, 并将结果写回到寄存器文件中 (输出给向量存储体 AM); 图 22 右半部分为增加脉冲阵列配置的 MAC 单元, 其操作数中的 src2 由向量单元 (或者标量广播指令) 将数据按行输入到其内部寄存器 W 中, src1 经由 MAC1 单元, 依次流过 MAC2, MAC3, 分别与 MAC 单元中已经加载的 W 进行乘法运算, 乘法运算的结果保存在 MAC 内部的 AC 寄存器中, 在不更新 W 的执行过程中 AC 不写会寄存器文件. 这样的流水线结构通过增加数据在流水线中的复用, 降低因频繁访存以及频繁寄存器写回所造成的功耗开销.

(3) 支持稀疏矩阵加速. 主要通过两个方面加速稀疏矩阵的运算: 一是发掘适合当前 FT-Matrix 体系结构的有效稀疏编解码技术, 实现对稀疏矩阵的快速编解码, 提高访存速度和增加计算并行度. 二是

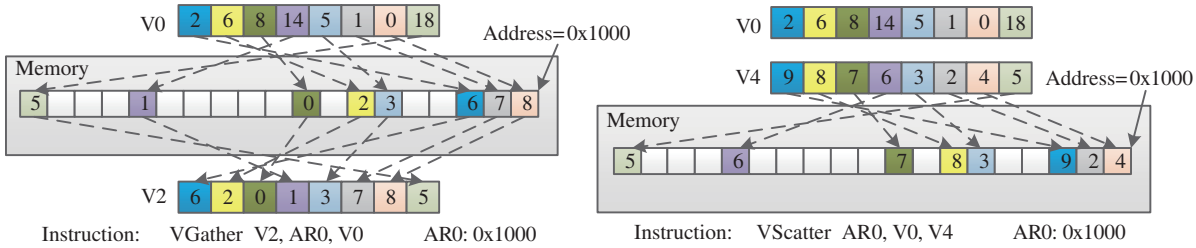

图 23 (网络版彩图) FT-Matrix 的向量 Gather/Scatter 指令

Figure 23 (Color online) VGather/VScatter instruction of FT-Matrix

表 4 不同 SIMD 宽度下的 MCPC 概率分布 (SIMD: Bank = 1:1)

Table 4 MCPC probability distribution of different SIMD widths (SIMD: Bank = 1:1)

Banks	MCPC ⩽ 2	MCPC ⩽ 3	MCPC ⩽ 4	MCPC ⩽ 5	MCPC expect
4	0.7969	0.9844	0.9999	0.9999	2.1249
8	0.5008	0.9101	0.9902	0.9993	2.5934
16	0.1948	0.7688	0.9629	0.9956	3.0515
32	0.0293	0.5456	0.9073	0.9866	3.4508
64	0.0000	0.2740	0.8041	0.9682	3.7608

通过采用 SIMD 结构的 Gather/Scatter 指令实现对非连续地址的随机访存. Gather/Scatter 通过访问由基址和一组索引指定的多个存储地址, 加载一组索引到向量寄存器, 然后, 通过这组索引执行访存, 其具体功能如图 23 所示. 对于 Gather 指令, 硬件读取这组访存地址的数据, 并且输出到向量寄存器; 对于 Scatter 指令, 则将输入向量寄存器里的一组数据, 写到该组访存地址对应的位置.

由于向量存储体 Bank 使用单端口 SRAM 实现, 针对不规则稀疏数据的 Gather/Scatter 访存方式, 存在多个 VPE 同时访问同一个 Bank 地址的情况, 从而引发 Bank 体访存冲突, 造成流水线的暂停. 为最大程度减小访存冲突造成的流水线停顿, 在流水线中增加了一组缓冲阵列. 对访存分布按照随机分布建建模后, 进行访存冲突分析, 得出如表 4 所示的不同 SIMD 宽度下, 每周期最大冲突数 (maximum conflicts per cycle, MCPC) 发生的概率 [45]. 由此可知, 16 个 VPE 同时访问 16 个 Bank 时, 最大访存冲突小于或等于 4 的情况已经基本覆盖了绝大多数访存情况, 综合硬件开销与访存效率, 最终将冲突仲裁缓冲阵列的深度设置为 4. 在 FT-M7002 中 16 向量 PE 及 16 Bank 向量存储体的体系结构下, 增加 Gather/Scatter 指令后较未增加时, 处理器稀疏矩阵相关测试集性能有 2~3 倍的提升. 因此, Gather/Scatter 指令在 FT-M7002 的优化设计中, 提高了稀疏矩阵处理的性能, 并为向量化编译提供支持, 但是高效实现稀疏矩阵的并行计算仍然面临着很大的技术困难.

(4) 智能数据搬移 DMA. 如图 20 中所示, 智能 DMA 的设计, 主要是针对在运算单元进入专用加速模式下的专用功能设计: 接收来自线程控制的配置信息, 设置自身的寻址和数据搬移模式. 实现从片上存储 (标量存储单元和向量存储单元) 按照指定的模式为片上数据缓冲、稀疏解码单元, 以及专用脉冲阵列提供规整的数据流; 并实现对专用脉冲阵列的输出数据以及激活函数的输出结果数据的存储功能.

4.4 可扩展的 AXI 接口协议

为了使 DSP 在进行专用加速时可以与其他硬件高效协同工作, FT-M7002 设计了 AXI 标准总线

协议扩展接口. AXI 协议适用于高性能、高带宽的系统, 灵活性好, 可实现读、写通道同时发送, 互不干预; AXI 总线协议支持乱序传输, 可以充分的利用总线带宽, 平衡内部系统. 从而, DSP 与其他处理器 (如 CPU) 可以通过 AXI 总线以共享内存的方式实现数据通信.

4.5 小结

本节对基于 FT-M7002 通用向量 DSP 的深度学习加速技术进行了分析, 在专用加速方面提出了几个优化方法. 这些优化在提升处理器性能的同时, 也给 DSP 本身的信号处理能力带来了一些不利的影响:

(1) 虽然兼顾了通用计算与专用加速, 但是, 这样的设计增加了硬件的设计复杂度, 给设计周期和仿真验证带来了一定的难度;

(2) 对多种精度运算的支持, 使得 Matrix DSP 单核性能由 96 GFLOPS 提升到了 384 GFLOPS (4 倍峰值性能提升), 但是也增加了实际的硬件开销, 其面积和功耗较未进行专用加速前增加了约 20%, 因此, 在实际芯片设计中, 我们对原有逻辑进行了删减, 去掉了增加的设计所能取代的向量定点运算, 以及向量规约、混洗运算, 保证芯片的面积与功耗与 FT-M7002 无太大差异;

(3) 对于稀疏运算的加速, 虽然性能有 2∼3 倍左右提升, 但是因为存在不规则访存所引起的冲突, 仍然是不能够高效发挥向量处理的优势.

5 结论与展望

本文结合深度学习的算法发展和优化, 深入分析了当前的硬件加速技术和方案, 以及各种硬件加速平台的利弊, 提出了基于通用向量 DSP 的深度学习硬件加速方案. 通过对深度学习算法及硬件加速技术的分析, 我们总结出以下 4 个结论:

(1) 随着深度学习的应用场景不断扩大, 基于各种硬件平台的加速芯片陆续出现. 同时, 不同深度学习应用对控制、通信、灵活可配置等功能的应用需求, 以及对计算效率、功耗、延迟、面积等硬件设计需求, 也在发生变化. 虽然各种硬件平台都有自己的优势, 但是单一硬件无法满足日益增长的应用需求. 而作为集通用计算、控制、通信等功能于一体的数字信号处理器 DSP, 因其强大的功能和较低的功耗、延迟、面积等优势, 早已在各个领域得到了广泛且成熟的应用. 因此, 在深度学习领域, 基于通用 DSP 的深度学习硬件加速具有广阔的研究前景.

(2) 提高数据密集计算能力是包括通用 DSP 的众多深度学习硬件加速设计的关键之一. 随着算法中并行性的提高, 通用向量 DSP 支持大规模并行计算的优势得以发挥. 因此, 通用向量 DSP 相比通用 DSP, 在深度学习加速领域, 具有更重要的应用价值.

(3) 传统的通用向量 DSP 具有较高的并行计算能力、精确的控制功能, 以及实时快速的通信能力. 但是, 并行计算能力的提升导致的高访存带宽需求, 成为限制芯片性能提升的瓶颈. 同时, 随着各种高效压缩算法的出现, 硬件加速中对稀疏运算的支持显得尤为重要. 因此, 降低硬件访存开销的技术, 如脉动阵列、高带宽三维存储等, 以及支持稀疏矩阵的随机访存, 是今后开展基于通用向量 DSP 深度学习硬件加速研究的主要方向之一.

(4) 展望深度学习硬件加速领域, 不管是云端还是终端, 多处理器异构协同加速趋势逐渐显现, 从单一平台到多平台协作的异构加速平台, 是今后深度学习硬件加速的重要发展方向. 另外, 交叉学科的优势在深度学习领域也得到发挥, 如新型存储结构以及存储工艺 (如三维堆叠存储等) 的出现、仿生

物学类脑计算等, 虽然受到当前科学技术的限制, 这些技术不能得到大规模的应用, 但是, 学科交叉所带来的科技创新, 在未来也会给深度学习硬件带来更多新的机遇.

参考文献

1 Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets. Neural Comput, 2006, 18: 1527–1554
2 Rumelhart D E. Learning internal representations by error propagation, parallel distributed processing. In: Explorations in the Microstructure of Cognition. Cambridge: MIT Press, 1986
3 Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In: Proceedings of International Conference on Neural Information Processing Systems, 2012. 1097–1105
4 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv:1409.1556
5 Szegedy C, Liu W, Jia Y Q, et al. Going deeper with convolutions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015
6 He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016
7 Park E, Kim D, Yoo S. Energy-efficient neural network accelerator based on outlier-aware low-precision computation. In: Proceedings of International Symposium on Computer Architecture, 2018. 688–698
8 Parashar A, Rhu M, Mukkara A, et al. SCNN: an accelerator for compressed-sparse convolutional neural networks. In: Proceedings of International Symposium on Computer Architecture, 2017. 27–40
9 Yu J, Lukefahr A, Palframan D, et al. Scalpel: customizing DNN pruning to the underlying hardware parallelism. Sigarch Comput Archit News, 2017, 45: 548–560
10 Chen Y H, Emer J, Sze V. Eyeriss: a spatial architecture for energy-efficient dataflow for convolutional neural networks. In: Proceedings of International Symposium on Computer Architecture, 2016. 367–379
11 Akhlaghi V, Yazdanbakhsh A, Samadi K, et al. SnaPEA: predictive early activation for reducing computation in deep convolutional neural networks. In: Proceedings of International Symposium on Computer Architecture, 2018. 662–673
12 Hegde K, Yu J, Agrawal R, et al. UCNN: exploiting computational reuse in deep neural networks via weight repetition. In: Proceedings of International Symposium on Computer Architecture, 2018. 674–687
13 Zhang S J, Du Z D, Zhang L, et al. Cambricon-X: an accelerator for sparse neural networks. In: Proceedings of International Symposium on Microarchitecture, 2016
14 Peemen M, Setio A A A, Mesman B, et al. Memory-centric accelerator design for convolutional neural networks. In: Proceedings of International Conference on Computer Design, 2013
15 Yazdani R, Riera M, Arnau J M, et al. The dark side of DNN pruning. In: Proceedings of International Symposium on Computer Architecture, 2018. 790–801
16 Yu J, Lukefahr A, Palframan D, et al. Scalpel: customizing DNN pruning to the underlying hardware parallelism. In: Proceeding of the 44th Annual International Symposium on Computer Architecture, 2017. 548–560
17 Sze V, Chen Y H, Yang T J, et al. Efficient processing of deep neural networks: a tutorial and survey. Proc IEEE, 2017, 105: 2295–2329
18 Lavin A, Gray S. Fast algorithms for convolutional neural networks. In: Proceedings of Computer Vision and Pattern Recognition, 2016. 4013–4021
19 Cong J S, Xiao B J. Minimizing computation in convolutional neural networks. In: Proceedings of International Conference on Artificial Neural Networks, 2014. 281–290
20 Zhang J Y, Guo Y, Hu X. Design and implementation of deep neural network for edge computing. IEICE Trans Inf Syst, 2018, 101: 1982–1996
21 Lu L Q, Liang Y, Xiao Q C, et al. Evaluating fast algorithms for convolutional neural networks on FPGAs. In: Proceedings of the 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), 2017. 101–108
22 Kung H T. Why systolic architectures? Computer, 1982, 15: 37–46
23 Jouppi N P, Young C, Patil N, et al. In-datacenter performance analysis of a tensor processing unit. In: Proceedings of the 44th Annual International Symposium on Computer Architecture, 2017
24 Du Z D, Fasthuber R, Chen T S, et al. ShiDianNao: shifting vision processing closer to the sensor. In: Proceedings

of the 42nd ACM/IEEE International Symposium on Computer Architecture, 2015
25 Desoli G, Chawla N, Boesch T, et al. 14.1 A 2.9TOPS/W deep convolutional neural network SoC in FD-SOI 28 nm for intelligent embedded systems. In: Proceedings of Solid-State Circuits Conference, 2017. 238–239
26 Liu S L, Du Z D, Tao J H, et al. Cambricon: an instruction set architecture for neural networks. Sigarch Comput Archit News, 2016, 44: 393–405
27 Li M, Huang R. Device and integration technologies for VLSI in post-Moore era (in Chinese). Sci Sin Inform, 2018, 48: 963–977
28 STH. Intel Xeon Phi Knights Mill for machine learning. 2017. https://www.servethehome.com/intel-knights-mill-for-machine-learning/
29 Oh K S, Jung K. GPU implementation of neural networks. Pattern Recogn, 2004, 37: 1311–1314
30 Coates A, Baumstarck P, Le Q, et al. Scalable learning for object detection with GPU hardware. In: Proceedings of International Conference on Intelligent Robots and Systems, 2009. 4287–4293
31 Yun S B, Kim Y J, Dong S S, et al. Hardware implementation of neural network with expansible and reconfigurable architecture. In: Proceedings of International Conference on Neural Information Processing, 2002. 970–975
32 Farabet C, Martini B, Corda B, et al. NeuFlow: a runtime reconfigurable dataflow processor for vision. In: Proceedings of Computer Vision and Pattern Recognition Workshops, 2011. 109–116
33 Zhang C, Prasanna V. Frequency domain acceleration of convolutional neural networks on CPU-FPGA shared memory system. In: Proceedings of ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2017. 35–44
34 Pham P H, Jelaca D, Farabet C, et al. NeuFlow: dataflow vision processing system-on-a-chip. In: Proceedings of the 55th International Midwest Symposium on Circuits and Systems, 2012. 1044–1047
35 Chen T S, Du Z D, Sun N H, et al. DianNao: a small-footprint high-throughput accelerator for ubiquitous machine-learning. ACM SIGPLAN Not, 2014, 49: 269–284
36 Chen Y J, Luo T, Liu S L, et al. DaDianNao: a machine-learning supercomputer. In: Proceedings of International Symposium on Microarchitecture, 2014. 609–622
37 Liu D F, Chen T S, Liu S L, et al. PuDianNao: a polyvalent machine learning accelerator. SIGARCH Comput Archit News, 2015, 43: 369–381
38 Du Z D, Fasthuber R, Chen T S, et al. Shidiannao: shifting vision processing closer to the sensor. In: Proceedings of the 42nd International Symposium on Computer Architecture, 2015
39 Han S, Liu X Y, Mao H Z, et al. EIE: efficient inference engine on compressed deep neural network. SIGARCH Comput Archit News, 2016, 44: 243–254
40 Merolla P A, Arthur J V, Alvarez-Icaza R, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. Science, 2014, 345: 668–673
41 Kumar S. Introducing qualcomm zeroth processors: brain-inspired computing, 2013. https://www.qualcomm.com/news/onq/2013/10/10/introducing-qualcomm-zeroth-processors-brain-inspired-computing
42 Demler M. CEVA XM6 Accelerates Neural Nets. 2016. https://www.ceva-dsp.com/wp-content/uploads/2017/02/MPR-CEVA-XM6-Accelerates-Neural-Nets.pdf
43 Demler M. CEVA NeuPro Accelerates Neural Nets. 2018. https://www.ceva-dsp.com/wp-content/uploads/2018/02/Ceva-NeuPro-Accelerates-Neural-Nets.pdf
44 Chen S M, Wang Y H, Liu S, et al. FT-Matrix: a coordination-aware architecture for signal processing. IEEE Micro, 2014, 34: 64–73
45 Tan H B, Chen H Y, Liu S, et al. Modeling and evaluation for gather/scatter operations in vector-SIMD architectures. In: Proceedings of the 28th International Conference on Application-specific Systems, Architectures and Processors, 2017

Deep learning hardware acceleration based on general vector DSP

Huili WANG, Yang GUO* & Wanxia QU

School of Computer, National University of Defense Technology, Changsha 410073*, China*
* Corresponding author. E-mail: guoyang@nudt.edu.cn

Abstract As deep learning (DL) plays an increasingly significant role in several fields, designing a high performance, low power, low-latency hardware accelerator for DL has become a topic of interest in the field of architecture. Based on the structure and optimization method of DL algorithms, this study aims to analyze the difficulties and challenges in DL hardware design. In comparison with the current mainstream DL hardware acceleration platform, advantages of the DL hardware acceleration based on general vector DSP are discussed. Besides, acceleration techniques, such as vector broadcasting and matrix conversion, are described. From the viewpoint of the shortcomings of the general vector DSP discussed herein, optimization techniques such as reconfigurable computing arrays that take into account the general vector calculations as well as specific DL acceleration are discussed in depth.

Keywords deep learning, architecture, hardware design, accelerator, digital signal processor

Huili WANG received her B.S. degree from National University of Defense Technology, China, in 2012. She is currently working toward obtaining a Ph.D. degree at the National University of Defense Technology, China. Her research interests include microprocessor design and deep learning.

Yang GUO was born in 1971. He is a Ph.D. degree holder, a professor, and has also supervised some doctorate degrees thesis. Furthermore, Prof. Guo is a senior member of China Computer Federation and a member of CADCG special committee. His primary research interests include low-power VLSI circuits, microprocessor design and verification, and electronic design automation techniques for VLSI circuits.

Wanxia QU was born in 1972; she is a Ph.D. degree holder and an associate researcher. Her research interests include high-performance computing, computer architecture, integrated circuit design, and function verification.