

面向混合量化 CNNs 的可重构处理器设计

常立博^{1,2}, 张盛兵¹

(1.西北工业大学 计算机学院, 陕西 西安 710072; 2.西安邮电大学 电子工程学院, 陕西 西安 710121)

摘要:为了解决已有卷积神经网络(convolution neural networks, CNNs)加速器,因无法适应混合量化 CNN 模型的计算模式和访存特性而引起加速器效率低的问题,设计了可适应混合量化模型的可重构计算单元、弹性片上缓存单元和宏数据流指令集。其中,采用了可根据 CNN 模型结构的重构多核结构以提高计算资源利用率,采用弹性存储结构以及基于 Tile 的动态缓存划分策略以提高片上数据复用率,采用可有效表达混合精度 CNN 模型计算和可重构处理器特性的宏数据流指令集以降低映射策略的复杂度。在 Ultra96-V2 平台上实现 VGG-16 和 ResNet-50 的计算性能达到 216.6 和 214 GOPS,计算效率达到 0.63 和 0.64 GOPS/DSP。同时,在 ZCU102 平台上实现 ResNet-50 的计算性能可达 931.8 GOPS,计算效率可达 0.40 GOPS/DSP,相较于其他类似 CNN 加速器,计算性能和计算效率分别提高了 55.4% 和 100%。

关键词:混合精度量化;卷积神经网络加速器;可重构计算

中图分类号:TP391.41

文献标志码:A

文章编号:1000-2758(2022)02-0344-08

基于卷积神经网络 CNNs 的智能算法已广泛应用到自动驾驶、智能监控和移动虚拟现实等领域。然而,此类智能算法在获得较高精度的同时,也具有极高的计算复杂度和巨大的参数量,导致采用对功耗和计算资源敏感的边缘计算设备实现此类算法时,通常无法满足实时性和低功耗的应用需求。因此本文将探索可适应于终端场景的高能效 CNN 加速器设计方法。

采用基于参数量化 CNNs 的轻量化方法可以极大降低神经网络的参数数量、访存量和计算复杂度^[1],从而为将 CNNs 的计算任务映射到边缘计算设备上提供了可能。因为 CNNs 中不同层的冗余度存在很大差异,而基于混合精度量化策略可针对不同层的参数变化范围调整数据位宽,得到在保证量化精度操作情况下表示参数的最小数据位宽^[2],从而确定最优的量化位宽。所以如果 CNNs 中各卷积层的参数均采用最优的量化位宽时,混合精度量化算法可在精度损失和压缩率之间达到最佳平衡。然而混合量化 CNNs 可能引入不规则的算法运算操作

(如奇数位宽的乘法运算),而已有仅支持特定统一参数位宽或特定几种参数位宽的 CNN 处理器,由于未能最大限度地利用混合量化提供的计算并行度,阻碍了处理器性能进一步提升。因此需要设计可灵活且高效支持混合量化 CNNs 的运算操作,才能真正提高加速器针对混合精度 CNNs 的计算效率。

与此同时,由于 CNNs 具有丰富的计算并行性,可以增加计算单元的数量以提高计算并行度。然而混合量化 CNNs 中不同层的参数位宽会导致不同层之间的访存特性具有较大差异,如果采用固定的数据划分和访问模式^[3],则会造成巨大的片上缓存资源浪费并增加数据移动量。如果将缓存不同类型参数的片上缓存区统一划分,并根据不同类型参数的存储需求来划分缓存区尺寸,则会提高片上存储器的利用率^[4]。同时,由于 CNNs 邻近层之间的参数存在一定的关联性,如上一层卷积的输出特征图可能是下一层卷积的输入特征图,邻近层之间同类型的参数规模变化较小;同时,基于残差结构的 CNNs 通过跨层信息融合以提高精度,因此可通过设计灵

收稿日期:2021-07-15 **基金项目:**国家重点研发计划(2019YFB1803600)与中国民航适航中心开放基金(SH2021111903)资助

作者简介:常立博(1985—),西北工业大学博士研究生,主要从事微处理器体系结构及深度学习加速器设计研究。

通信作者:张盛兵(1968—),西北工业大学教授,主要从事微处理器体系结构研究。e-mail:zhangsb@nwpu.edu.cn

活的参数访问模式以提高参数的复用性并减少数据移动量,从而提高计算效率。

不同 CNNs 的模型结构和计算模式均存在较大差异,并且不同应用场景可提供的计算资源及对处理器的性能和功耗等要求也不同。采用面向特定应用领域体系结构(domain-specific architectures, DSA)^[5]和可重构计算技术方法,既可获得很高的效能又具备一定的灵活性,可以兼顾嵌入式智能终端系统对高性能、低功耗以及高灵活性的要求,从而满足应用场景对 CNN 处理器的可定制性和可扩展性要求。然而,随着可重构加速器灵活度的提高,其设计空间探索(design space exploration, DSE)的范围将变得十分巨大,因此需要一种可高层次表达 CNNs 和可重构平台特性的算法映射表达方法,以降低映射复杂度,提高可重构效率和处理器的通用性,从而可加速不同类型的 CNN 模型。

针对上述问题,本文采用软硬件协同优化和可重构计算方法,针对混合精度 CNNs 计算和访存特性,设计可支持混合位宽的可重构运算单元、可支持多种数据复用以及减少数据移动数量的缓存器,以及设计了一种表达混合 CNNs 计算、访存以及计算模式的宏指令集和可重构处理器架构。本文主要贡献如下:

1) 提出具有可定制性和可扩展性的 CNN 处理器架构,以及可有效表达混合精度 CNNs 模型计算的宏指令集(macro instruction set architecture, mISA)。通过高层描述被加速的 CNNs 模型计算、访存和控制等数据流特征,并设计对应的可重构处理器结构以提高计算效率。

2) 提出可支持混合位宽并行乘加运算的可重构微处理单元(reconfigurable micro-processing element, RmPE)和可重构多核计算引擎架构(compute engine, CE)。根据被加速的混合精度 CNNs 模型的结构特点和目标平台的资源限制,计算引擎通过重构阵列结构和数据流模式以提高计算资源利用率。

3) 提出可适应可重构 CNNs 计算特性的弹性片上数据缓存策略。通过动态配置地址及片上互联模式减少非必要数据移动的延时和功耗开销;通过基于 Tile 的动态缓存划分策略提高片上存储资源利用率。

1 相关工作

为了满足不同 CNN 模型中各卷积层对于运算位宽的多样化需求,提高 CNN 加速器的计算效率,目前针对混合精度 CNNs 计算模式以及可重构 CNN 处理器体系结构方面开展了大量研究。

1.1 混合精度 CNNs 计算模式

目前已有大量基于 ASIC 或可重构平台(如 FPGA)的面向混合精度 CNN 加速器设计。Judd 等^[6]提出的 Stripes 处理器支持可变精度的激活数据, Lee 等^[7]提出的 UNPU 处理器支持可变精度的权重数据,但是 2 种架构都只支持一种运算数据位宽的改变,因此没有最大限度提高混合精度 CNNs 的计算效率;Sharify 等^[8]提出的 Loom 处理器采用串行乘法单元以支持多精度的卷积运算,然而由于需要并串转换电路,需要消耗大量芯片面积和功耗;Sharna 等^[9]提出的基于位级融合的 Bit Fusion 处理器,利用 2 bit 数据运算单元的融合和分解支持不同运算数据位宽,但是由于基础运算单元处理位宽的限制,对于存在较多非二次幂位宽参数的混合精度 CNNs,该架构的加速效率会受到限制。

虽然基于 ASIC 的方法可通过定制化运算单元实现混合精度 CNNs 模型加速,但是此类方法无法应用到基本运算单元固定的可重构平台,如基于 DSP 的 FPGA 等。已有基于 FPGA 的 CNNs 加速器通常采用匹配 CNNs 模型中最大的数据位宽的计算模式,所以对于量化后存在大量低数据位宽的 CNNs 则无法充分利用 FPGA 平台中的 DSP 等计算资源,从而降低了加速器的计算效率,因此需要一种自适应多精度计算的高效处理单元。为了提高 FPGA 平台针对低数据位宽运算的并行度,可将多个低位宽数据合并为一个高位宽数据,通过复用 DSP 以提高计算并行度^[10],其能够根据不同运算位宽调整运算并行度,但是运算单元的并行度受到符号位的限制,无法充分复用 DSP 资源。

1.2 可重构 CNNs 处理器体系结构

虽然 CNNs 具有计算类型较少并且计算流程固定的特点,但是不同 CNNs 或是同一模型中的不同卷积层的数据流和数据访存等方面具有不同的特性。已有学者提出基于模板或自动化设计方法^[11-14],此类方法首先对算法和可重构平台的资源进行抽象描述,然后通过优化算法确定加速器体系

结构和具体可重构配置的参数。如 Ma 等^[11]首先设计了可实现 CNNs 运算的基本操作单元库,然后通过自动化地组合不同运算单元以适应不同的 CNNs 模型;Wei 和 Guo 等^[12-13]分别提出了基于性能和计算资源利用率约束的自动化设计优化方法,从而提出针对不同 CNNs 的计算效率。然而此类方法的优化策略复杂度将随着搜索空间的增加而快速提高,因此当 CNNs 模型复杂度较高时,通常很难得到最优的处理器体系结构和映射策略。同时, Azizimazreah 等^[14]通过配置和重构“物理”基本单元,构建“逻辑”运算单元和缓存单元,从而降低重构延时并提高加速器的可定制性和可扩展性,然而其直接将各卷积运算转换为可重构加速器的配置信息,因此映射复杂度较高。

2 面向混合 CNNs 的可重构处理器

2.1 可重构 CNNs 处理器结构

为了提高 CNNs 的运算效率,本文设计了一种可支持混合精度运算的多核处理器,结构如图 1 所示。由支持可变精度的卷积、激活、池化等操作的计算引擎,可支持弹性划分与动态重组的缓存单元以及可支持乱序发射的控制单元组成,而计算数据和控制数据通过片上总线接口单元与外部完成数据交换。

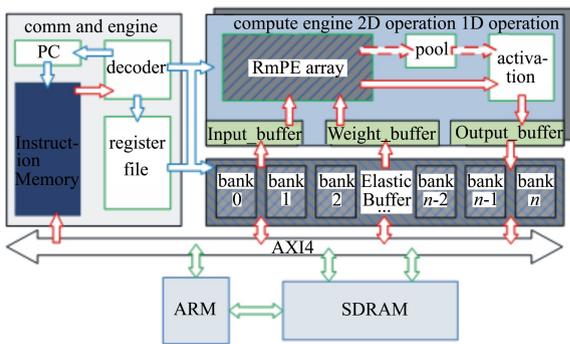


图 1 可重构 CNN 处理器架构

其中,控制单元(command engine, CMDE)包含指令存储、PC 寻址单元、寄存器堆以及译码单元,通过 PC 访问指令存储,经译码模块产生相应的控制信号,进而控制片上数据通路和数据缓存。计算引擎由用于完成特定特征图 tile 块卷积运算的二维 RmPE 阵列,以及用于池化和激活等标量运算的专用计算单元组成。片上弹性缓存包含多个相互独立

的存储块(bank),通过专用 Crossbar 总线将弹性缓存与计算引擎中对应的数据缓存单元连接起来。

2.2 宏指令集设计

为了将不同结构混合精度 CNNs 高效地映射到可重构计算平台,根据 CNNs 中计算类型较少并且计算流程固定的特点,本文提出一种控制通路简单、可提高数据级并行度的宏指令集。该指令集由 10 条 32 bit 指令组成,指令字段划分方式如图 2 所示,根据 32 bit 数据表示的含义将其细分为 3 个字段:功能,寄存器和参数。其中,由 4 bit 数据组成的 function 表示指令类型,由 10 bit 数据组成的 registers 表示用于控制计算过程的变量或索引,而由 18 bit 数据组成的 parameters 表示如量化的 CNN 模型的特征等信息。指令集分为 4 类:配置、运算、访存和循环控制。配置指令(config_*)用于实现在每个卷积层运算前对该层的存储划分以及计算特性描述;运算指令 compute 用于启动 tile 级运算;访存指令 load/store 控制计算引擎与弹性存储之间的数据交互,循环控制指令 add、beq、jmp 负责控制计算任务的循环过程。

Function	Registers		Parameters					
conv_config	r_1 (Tif)		K size	Padding	Stride	W_width	I_width	
act_pool_config	r_1 (Tof)		P_size	P_stride	Pool	A_method	O_width	
bank_config			B_col	B_size	Padding	Tif/Tof	W/I/O	
tile_config	r_2 (Tif/Tif)	r_1 (K/T)	Tile_channel		Tile_row		W/F	
load	r_2 (k/l)	r_1 (m/n)				P_M	W/F	
store	r_2 (l)	r_1 (m)				P_M	W/F	
compute	r_2 (n)	r_1 (m)				P_M	C/A/P	
add	r_2	r_1				Imm		
beq	r_2	r_1				PC		
jmp						PC		

图 2 宏指令集格式

该宏指令包含 2 个 32 bit 通用寄存器,用于缓存特征图块的索引,并暂存一些标量数据,例如用于循环控制的变量值。如图 2 所示, $r_1(a)$ 表示参数 a 存储在寄存器 r_1 中, $r_1(a/b)$ 表示参数 a 或 b 存储在寄存器“ r_1 ”中,寄存器 r_2 以类似的方式起作用。宏指令的参数(parameters)用于配置或控制可重新配置的计算引擎和弹性片上缓冲区。例如,根据存储单元阵列的行和列中的 RAM 数量,特征图的通道数,填充的大小以及相关的缓冲区类型,使用“bank_config”来配置划分缓冲区方法等。表 1 列出了宏指令集中各参数缩写的含义。

表 1 指令集中参数缩写的含义

参数缩写	含义
K_size	卷积核的大小
Padding	Tile 化过程中填充的数据量
Stride	卷积步长
I_width/O_width	输入特征值/输出特征值的数据位宽
W_width	权重的数据位宽
P_size	池化核的大小
P_stride	池化步长
Pool	池化操作的类型
A_method	激活操作的类型
B_row/B_col	缓存阵列的宽/高
Tif/Tof	在通道维度分块后的输入特征图/输出特征块数量
Tiy	一个 Tile 块的高度
W/L/O	缓存区用于缓存权重/输入特征图/输出特征图
Tile_channel	tile_config 中配置 r_1 寄存器的立即数
Tile_row	tile_config 中配置 r_2 寄存器的立即数
K/T	在通道/行维度划分后特征图数量
W/F	缓冲区用于缓存权重/特征图
m/n	输出/输入特征图的块索引号
PM	输出通道方向的计算并行度
C/A/P	计算类型为卷积/激活/池化
k/t	权重块/特征块的块索引号
Imm	立即数
PC	程序存储器地址

3 混合精度 CNNs 计算引擎

3.1 可重构微处理单元

混合精数量化算法通过调整不同卷积层中特征值和权重的位宽,以实现 CNNs 的精度和压缩率之间最优平衡。其量化后的 CNNs 各层的参数位宽量为确定值,因此根据乘法分割原理,将多个低位宽数据拼接为一个高位宽数据运算的模型可表达如(1)式所示,其中 x 为乘数位宽, y 为被乘数位宽, p 为并行度, b 为乘法器的最大运算位宽。在确定可重构平台中的乘法器位宽后,可利用(1)式确定特定运算数据位宽和乘法操作的最大并行度。

$$b = px + (p - 1)y - 2p + 1 \quad (1)$$

根据上述并行乘法的分析,结合已有可重构计算资源的特性(如 FPGA 平台中的 DSP),本文设计可支持 2~8 bit 内任意精度并行乘加运算的可重构微处理单元(reconfigurable micro-processing element, RmPE),其架构如图 3 所示。RmPE 内部采用权重

复用的计算模式,可根据输入特征和权重的位宽确定运算并行度,并且通过控制编码与解码操作动态可重构 RmPE 单元。并行输入特征与权重在运算并行度的控制下分离符号位与数据位,数据位输入乘法运算单元进行并行乘法运算,符号位输入异或门进行符号计算。不同运算并行度下解码出的乘法运算结果输入对应累加器,对卷积过程中的部分和进行累加,而计算出的符号位则会作为累加器执行加法或减法运算的标志。卷积运算结束后,累加结果统一截断至 8 bit 以降低激活运算的复杂度。

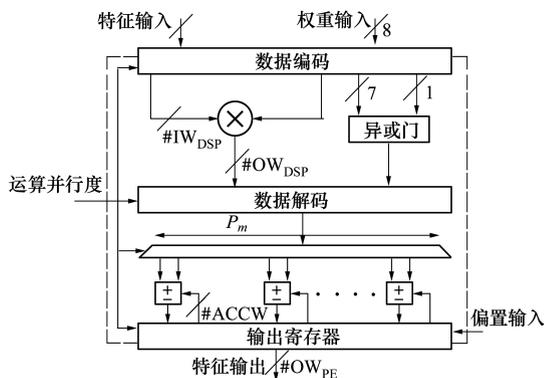


图 3 RmPE 架构

为了适应不同类型的混合精度 CNNs 的数据位宽,本文通过定义 RmPE 中对应的可重构参数,从而可根据不同混合精数量化网络模型定制计算单元的规模,在提升计算性能的并提高计算资源利用率,表 2 列举了 RmPE 中的可重构参数及其含义。

表 2 RmPE 参数描述

参数名	描述
#IW _{PE}	RmPE 的输入宽度,其大小可表示为 DSP 最大运算并行度与对应输入特征数据位宽的乘积
#IW _{DSP}	DSP 输入运算数据位宽
#OW _{DSP}	DSP 输出运算结果位宽
#P _m	DSP 乘法运算并行度最大值
#OW _{PE}	RmPE 的输出宽度,其大小为 8 倍的计算并行度

3.2 计算引擎

本文采用文献[11]提出的计算单元组织方式,根据被加速的混合精度 CNNs 模型的结构特点和目标平台的资源限制,将多个 RmPE 组成的二维阵列构成计算引擎(compute engine, CE)。计算引擎的整体设计如图 4 所示,包括采用阵列结构的卷积运算单元和串行处理卷积结果的池化和激活单元。计算引擎采用流模式处理运算数据,计算流程中的数

据通路选择由控制单元管理,卷积运算单元从输入缓存和权重缓存中读取数据进行卷积运算,产生的并行卷积结果通过重组,以串行数据流模式输入激活或者池化单元,最终运算结果以并行方式存入输出缓存。

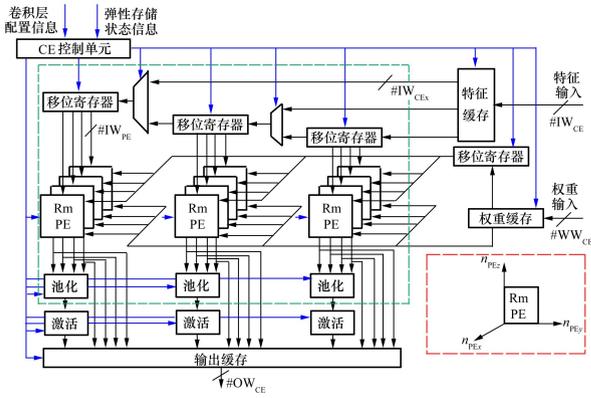


图 4 计算引擎硬件结构

由于 RmPE 在卷积处理单元中按照阵列方式进行排列,因此可通过调整阵列结构以适应不同混合精度 CNNs 模型的计算特性。如图 4 的虚线框中给出的 n_{PE_x}, n_{PE_y} 以及 n_{PE_z} 分别表示计算阵列的 3 个维度。可根据被加速的混合精度 CNNs 的结构特点和目标平台的资源限制,通过重构阵列结构和数据流模式以提高计算资源利用率。

4 面向弹性缓存的重构与划分机制

为了解决卷积运算不同类型数据在混合精度 CNNs 各层间存储量差异巨大的问题,本文采用文献[14]提出的弹性存储结构(elastic buffer, EB),将片上存储单元划分为多个独立的 bank,不再设有特定的输入缓存、权重缓存以及输出缓存,只提供对应的数据输入输出接口,接口数目根据加速系统中计算引擎的数目定制,以保证数据访问并行度。在每层运算开始前,控制器根据各 bank 的状态和 bank 划分策略,以动态重组的方式将 bank 指定为特定的计算引擎缓存单元。计算引擎通过待运算数据的索引(如特征图的 tile 编号等)访问对应的 bank。其中卷积计算单元中的输入、输出、权重数据按照图 5 所示的方式进行 tile 划分。在图 5a)中,输入和输出特征数据的 tile 在行方向上的尺寸等于特征图宽度,在列方向上的尺寸决定了 tile 的组数,每组包含

全部输入或输出通道,计算过程中, tile 的划分方式会随着运算量的改变而改变,因此弹性存储内部以特征图尺寸对特征数据进行存储,以支持不同 tile 尺寸的重构。图 5b)中权重块在行方向上的数据量对应该层单个卷积核的数据量,列方向的尺寸对应输入通道的分组,每组包含全部输出通道的卷积核。

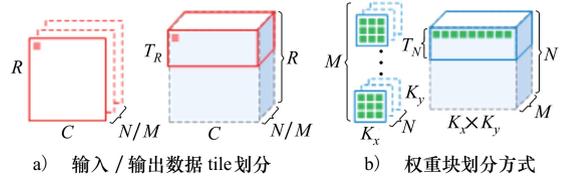


图 5 tile 划分方法

为了简化存储单元的重构逻辑,片上数据存储位宽采用 4 bit 和 8 bit 模式,通过多个 bank 的组合即可实现不同 tile 尺寸的重构,以输入特征数据为例,行和列方向上的 bank 数目可分别表示为 $N_{bank}^r = C \times IW_f / W_{bank}$ 和 $N_{bank}^c = R \times N / H_{bank}$,因此在每层运算前,通过 N_{bank}^r 和 N_{bank}^c 即可确定该层的输入特征数据的存储划分方式。

弹性存储需要根据计算引擎中 RmPE 阵列的 n_{PE_x}, n_{PE_y} 以及 n_{PE_z} 3 个维度进行重构,以减少计算引擎因等待数据引起停顿,从而提高计算单元的利用率。因此,构建输入特征存储区域的存储块在位面方向的数目需要匹配 n_{PE_z} ,从而为 RmPE 阵列提供 n_{PE_y} 行的并行输入;构成权重以及输出特征存储区域的存储块在位面方向的数目需要匹配 n_{PE_z} ,从而满足 RmPE 阵列在输出通道方向上的并行度。

5 验证与分析

5.1 建立验证系统

本文选择经典 CNNs-VGG-16 和包含残差结构并且结构复杂的 ResNet-50,验证本文提出的混合精度 CNNs 可重构处理器性能。采用文献[2]提出的混合精度量化算法量化 VGG-16 和 ResNet-50 模型,量化参数位宽变化范围及准确率如表 3 所示,其中准确率为针对 ImageNet 数据集的图像分类结果。从表 3 可以看出量化后网络模型的精度损失均小于 1%,2 种网络量化后的权重位宽分布在 2~8 bit 之间。同时,根据文献[15]对于激活数据量化敏感度分析,激活值量化位宽的改变会对网络准确率造成

很大影响,因此在混合精度量化后的 ResNet-50 和 VGG-16 网络中,激活数据的位为特定几种数据位宽,其中针对 VGG-16 模型选择 4 或 8 bit,而针对 ResNet-50 模型选择 4,6 或 8 bit。

表 3 混合精度量化结果

模型	权重位宽/ bit	激活值 位宽/bit	原始模型 准确率/%	量化后模 型准确率/%
VGG-16	2~8	4,8	71.7	71.6
ResNet-50	2~8	4,6,8	76.1	75.2

5.2 不同阵列结构对计算效率的影响分析

根据 n_{PE_x} 方向上对 RmPE 数目的分析,卷积处理单元在 3 个方向上的尺寸共有 4 种可能性,按照 $(n_{PE_x}, n_{PE_y}, n_{PE_z})$ 的方式标记,分别为 $(4, 7, 12)$, $(5, 7, 10)$, $(6, 7, 8)$, $(7, 7, 7)$, 因此需要为 2 种混合精度网络各设计 4 种卷积处理单元进行性能比较,为了方便描述,依照 CNNs 模型以及卷积处理单元在 n_{PE_x} 方向上的尺寸对加速器分别标记为: ResNet₄, ResNet₅, ResNet₆, ResNet₇。表 4 展示了针对混合精度 ResNet-50 模型,本文设计的 4 种结构的混合精度 CNN 加速器,在 Ultra96-V2 开发板上的计算吞吐量、计算效率、推理延时以及资源使用量等。其中计算性能最优的是 ResNet₄,其计算性能和计算效率分别达到了 219.56 GOPS 和 0.653 GOPS/DSP,其吞吐量甚至超过了使用更多计算单元的 ResNet₅,并且在 DSP 数量相同的情况下,ResNet₄ 的计算性能要优于 ResNet₆,这是由于混合精度 ResNet-50 模型在 n_{PE_x} 方向上的计算并行度处于饱和状态,不需要很多 RmPE 即可在特征的行方向上获得较高的并行计算能力。然而,针对运算并行度不高的模型(如 VGG-16),不同类型计算阵列在特征行方上的并行

计算能力差别不大,因此可通过减少 n_{PE_x} 并增加 n_{PE_z} 来提高计算效率。可以看出,针对特定的混合精度 CNN 模型,可通过改变计算阵列在不同维度的尺寸和对应的片上缓存划分策略,达到最优的计算效率。然而,针对不同类型的混合精度 CNN 模型,则需要不同的重构方法才能获得最优性能。

5.3 与相关研究的对比

本文设计的混合精度 CNNs 处理器与其他 FPGA 平台的加速器^[10-14]的对比结果如表 5 所示。其中存储资源部分的 Xilinx FPGA 采用 36 kB 的存储块,而 Intel FPGA 采用 20 kB 的存储块;表中的 DSP 资源的 Xilinx FPGA 的 DSP 运算位宽为 27×18 bit, Intel FPGA 的 DSP 运算位宽为 18×18 bit;表中逻辑资源的 Xilinx FPGA 为 LUTs, Intel FPGA 为 ALMs。为了公平地对比性能,表 5 中所有的计算性能均在 batch size 为 1 时测得。本文的混合精度 CNN 处理器在计算性能方面,比使用了更多 DSP 资源的固定位宽加速器^[13]提高了 111.4 GOPS,同时可获得更高的准确率。文献[11-12]使用与本文类似的脉动阵列结构加速卷积运算,当 FPGA 上的 DSP 资源用量处于饱和状态时,虽然本文的混合精度 CNN 处理器在片上资源较少的 Ultra96-V2 平台上的计算性能较低,但是在计算效率方面分别提高了 3.2 倍和 1.6 倍,说明本文提出的并行计算单元 RmPE 对计算效能的提升作用。同时,将本文提出的可重构 CNNs 处理器映射到片上资源较多的 ZCU102 平台上,针对 ResNet-50 网络的计算性能可达 913.8 GOPS,计算效率达到 0.40,其性能和计算效率均高于文献[11],说明了本文提出的处理器具有较好的可扩展性和可定制性。文献[10]使用了与本文类似的并行

表 4 针对混合精度 ResNet-50,采用不同阵列结构的各种资源使用情况和性能对比

RmPE 阵列结构	ResNet ₄	ResNet ₅	ResNet ₆	ResNet ₇
DSP	336(93%)	350(97%)	336(93%)	343(95%)
LUTs/kB	51.25(74%)	53.04(77%)	51.28(74%)	52.01(75%)
FFs/kB	78.01(57%)	81.22(58%)	78.02(56%)	79.56(57%)
BRAMs/36kB	210(97%)	212(98%)	212(98%)	202(93%)
运算次数/GOP	8.24	8.24	8.24	8.24
DSP 数量	336	350	336	343
推理延时/ms	37.53	38.28	42.15	44.79
计算性能	219.56	215.26	195.49	183.97
计算效率	0.653	0.615	0.582	0.536

表5 与其他基于FPGA平台CNN加速器对比

加速器	[10]	[11]	[12]	[13]	[14]	本文		
FPGA	Virtex7 VC709	Aria 10 GX 1150	Aria 10 GT 1150	Zynq XC7Z030	Virtex-7 485T	Ultra96-V2	Ultra96-V2	ZCU102
频率/MHz	200	240	232	150	150	150	150	200
CNNs	VGG-16	ResNet-50	VGG-16	VGG-16	VGGNet-D	VGG-16	ResNet-50	ResNet-50
检测精度	-	-	-	67.7%	-	71.6%	75.2%	75.2%
DSP	2877(80%)	3036(100%)	3000(99%)	400(100%)	2800(100%)	343(95%)	336(93%)	2315(92%)
BRAMs	1765(60%)	2356(87%)	1668(61%)	203(77%)	7275(89%)	54.5(40%)	204(97%)	743(82%)
计算性能	1713.0	599.6	1171.3	105.2	809.0	216.6	214.0	931.8
计算效率	0.59	0.20	0.39	0.26	0.29	0.63	0.64	0.40

乘法单元加速混合精度 CNNs,在吞吐率和计算效率方面均优于其他设计,而本文利用分离符号位与使得本文的混合精度 CNN 处理器在 DSP 的使用效率方面相较于文献[6]提高了 6.7%。

6 结 论

为了解决将基于混合精度 CNNs 的智能算法在已有通用计算平台上实现,无法满足终端设备对实时性和低功耗的应用需求的问题。本文设计了支持

多精度并行乘加运算的可重构微处理单元,可根据混合 CNNs 模型结构重构多核处理器。根据不同混合精度 CNNs 定制片上资源,在计算过程中重构计算单元并行度和片上缓存单元的划分方式,提高处理器的计算效率。本文设计的 CNN 处理器在 Ultra96-V2 上推理 VGG-16 和 ResNet-50 时计算性能分别达到 216.6 和 214 GOPS,计算效率为 0.63 和 0.64 GOPS/DSP,实现了对嵌入式硬件平台上计算资源的高效利用。

参考文献:

- [1] ZHAO R, HU Y, DOTZEL J, et al. Improving neural network quantization without retraining using outlier channel splitting[C] //International Conference on Machine Learning, 2019
- [2] WANG K, LIU Z, Lin Y, et al. HAQ: hardware-aware automated quantization with mixed precision[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019
- [3] MA Y, CAO Y, VRUDHULA S, et al. Performance modeling for CNN inference accelerators on FPGA[J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2019, 39(4): 843-856
- [4] AZIZIMAZREAH A, CHEN L. Shortcut mining: exploiting cross-layer shortcut reuse in DCNN accelerators[C] //2019 IEEE International Symposium on High Performance Computer Architecture, 2019
- [5] HENNESSY J, PATTERSON D. A new golden age for computer architecture: domain-specific hardware/software co-design, enhanced[C] //ACM/IEEE 45th Annual International Symposium on Computer Architecture, 2018
- [6] JUDD P, ALBERICIO J, HETHERINGTON T, et al. Stripes: bit-serial deep neural network computing[C] //2016 49th Annual IEEE/ACM International Symposium on Microarchitecture, 2016
- [7] LEE J, KIM C, KANG S, et al. UNPU: an energy-efficient deep neural network accelerator with fully variable weight bit precision[J]. IEEE Journal of Solid-State Circuits, 2018, 54(1): 173-185
- [8] SHARIFY S, LASCORZ A D, SIU K, et al. Loom: exploiting weight and activation precisions to accelerate convolutional neural networks[C] //2018 55th ACM/ESDA/IEEE Design Automation Conference, 2018: 1-6
- [9] SHARMA H, PARK J, SUDA N, et al. Bit fusion: bit-level dynamically composable architecture for accelerating deep neural network[C] //2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture, 2018: 764-775
- [10] YIN S, TANG S, LIN X, et al. A high throughput acceleration for hybrid neural networks with efficient resource management on FPGA[J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2018, 38(4): 678-691
- [11] MA Y, CAO Y, VRUDHULA S, et al. Automatic compilation of diverse CNNs onto high-performance FPGA accelerators[J].

IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 39(2): 424-437

- [12] WEI X, YU C H, ZHANG P, et al. Automated systolic array architecture synthesis for high throughput CNN inference on FPGAs [C]//Proceedings of the 54th Annual Design Automation Conference, 2017
- [13] GUO K, SUI L, QIU J, et al. Angel-eye: a complete design flow for mapping CNN onto embedded FPGA[J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2017, 37(1): 35-47
- [14] AZIZIMAZREAH A, CHEN L. Polymorphic accelerators for deep neural networks[J]. IEEE Trans on Computers, 2022, 71(3): 534-546
- [15] DONG Z, YAO Z, ARFEEN D, et al. HAWQ-v2: hessian aware trace-weighted quantization of neural networks[J]. Advances in Neural Information Processing Systems, 2020, 33: 18518-18529

A reconfigurable processor for mix-precision CNNs on FPGA

CHANG Libo^{1,2}, ZHANG Shengbing¹

(1.School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China;
2.School of Electronic Engineering, Xi'an University of Posts and Telecommunication, Xi'an, 710121 China)

Abstract: To solve the problem of low computing efficiency of existing accelerators for convolutional neural network (CNNs), which caused by the inability to adapt to the characteristics of computing mode and caching for the mixed-precision quantized CNNs model, we propose a reconfigurable CNN processor in this paper, which consists of the reconfigurable adaptable computing unit, flexible on-chip cache unit and macro-instruction set. The multi-core CNN processor can be reconstructed according to the structure of CNN models and constraints of reconfigurable resources, to improve the utilization of computing resources. The elastic on-chip buffer and the data access approach by dynamically configuring an address to better utilization of on-chip memory. Then, the macroinstruction set architecture (mISA) can fully express the characteristics of the mixed-precision CNN models and reconfigurable processors, to reduce the complexity of mapping CNNs with different network structures and computing modes to reconfigurable the CNNs processors. For the well-known CNNs-VGG16 and ResNet-50, the proposed CNN processor has been implemented using Ultra96-V2 and ZCU102 FPGA, showing the throughput of 216.6 GOPS, and 214 GOPS, the computing efficiency of 0.63 GOPS/DSP and 0.64 GOPS/DSP on Ultra96-V2, respectively, achieving a better efficiency than the CNN accelerator based on fixed bit-width. Meanwhile, for ResNet-50, the throughput and the computing efficiency are up to 931.8 GOPS, 0.40 GOPS/DSP on ZCU102, respectively. In addition, these achieve up to 55.4% higher throughput than state-of-the-art CNN accelerators.

Keywords: mixed-precision quantization; convolutional neural network accelerator; reconfigurable computing

引用格式: 常立博, 张盛兵. 面向混合量化 CNNs 的可重构处理器设计[J]. 西北工业大学学报, 2022, 40(2): 344-351

CHANG Libo, ZHANG Shengbing. A reconfigurable processor for mix-precision CNNs on FPGA[J]. Journal of Northwestern Polytechnical University, 2022, 40(2): 344-351 (in Chinese)