二值 VGG 卷积神经网络加速器优化设计*

张旭欣,张　嘉,李新增,金　婕
(上海工程技术大学 电子电气工程学院,上海 201600)

摘　要:基于 FPGA 的二值卷积神经网络加速器研究大多是针对小尺度的图像输入,而实际应用主要以 YOLO、VGG 等大尺度的卷积神经网络作为骨干网络。通过从网络拓扑、流水线等层面对卷积神经网络硬件进行优化设计,从而解决逻辑资源以及性能瓶颈,实现输入尺度更大、网络层次更深的二值 VGG 神经网络加速器。采用 CIFAR-10 数据集对基于 FPGA 的 VGG 卷积神经网络加速器优化设计进行验证,实验结果表明系统实现了 81%的识别准确率以及 219.9 FPS 的识别速度,验证了优化方法的有效性。

关键词:优化设计;二值卷积神经网络;FPGA 加速器
中图分类号:TN402;TP183　　**文献标识码**:A　　DOI:10.16157/j.issn.0258-7998.201207

中文引用格式:张旭欣,张嘉,李新增,等. 二值 VGG 卷积神经网络加速器优化设计[J].电子技术应用,2021,47(2):20-23.
英文引用格式:Zhang Xuxin,Zhang Jia,Li Xinzeng,et al. Optimization design of binary VGG convolutional neural network accelerator[J]. Application of Electronic Technique,2021,47(2):20-23.

Optimization design of binary VGG convolutional neural network accelerator

Zhang Xuxin,Zhang Jia,Li Xinzeng,Jin Jie
(College of Electronic and Electrical Engineering,Shanghai University of Engineering Science,Shanghai 201600,China)

Abstract: Most of the existing researches on accelerators of binary convolutional neural networks based on FPGA are aimed at small-scale image input, while the applications mainly take large-scale convolutional neural networks such as YOLO and VGG as backbone networks. The hardware of convolutional neural network is optimized and designed from the two aspects including the network topology and pipeline stage, so as to solve the bottleneck of logic resources and improve the performance of the binary VGG network accelerator. CIFAR-10 dataset resized to 224×224 was used to verify the optimized design of VGG convolutional neural network accelerator based on FPGA. Experimental results showed that the system achieved 81% recognition accuracy and 219.9 FPS recognition speed, which verified the effectiveness of the optimization method.

Key words: optimization design; binary convolutional neural network; FPGA accelerator

0 引言

深度卷积神经网络(Convolutional Neural Network,CNN)已经成为了当前计算机视觉系统中最有前景的图像分析方法之一。

近年来,随着 Binary-Net、Dorefa-Net、ABC-Net 等[1-3]低精度量化神经网络的深入研究,越来越多的研究集中于在 FPGA 硬件中构建定制的加速器结构,实现 CNN 的加速[4]。基于 FPGA 的低精度量化神经网络实现主要可分为两类:流架构[5-6]和层架构[7-8]。其中,由于流架构实现了流水线化,每个阶段都可以独立处理输入且可以针对 CNN 逐层设计并优化相应层的加速运算单元,因此拥有更高的吞吐率和更低的延迟以及内存带宽,但其逻辑资源等消耗也相当可观。因此,现有的基于流架构实现的二值神经网络加速器研究大多是针对 32×32 尺度 MNIST 数据集等小尺度的图像输入。而实际应用中更多使用如 448×448 尺度的 YOLO、224×224 尺度的 VGG 等作为骨干网络,一方面,大尺度输入的网络结构参数量往往较大(以 VGG 为例,其参数量大约 500 MB),高端 FPGA 的片上内存容量也仅 32.1 Mb 左右,这对 FPGA 实现 CNN 加速将是资源瓶颈。即使采用低精度量化策略,FPGA 有限的片上内存资源仍捉襟见肘。另一方面,虽然各层运算单元可以得到特定优化,然而由于网络拓扑结构限制,往往各层网络很难实现计算周期的匹配,从而造成推断性能难以进一步提高。针对基于流架构的二值卷积神经网络加速器设计存在的资源与性能的瓶颈,本文以 224×224 尺度的 VGG-11 网络加速器设计为例,重点研究了大尺度的二值卷积神经网络硬件加速器设计、优化及验证,主要工作如下:

(1)针对大尺度流架构的二值 VGG 卷积神经网络加速器设计存在的资源与性能瓶颈,提出了网络模型优化

*基金项目:国家自然科学基金(61701295,61801286)

和流水线优化的方法。

(2)设计并优化了 224×224 尺度的基于流架构的二值 VGG 卷积神经网络加速器。实验表明基于 FPGA 平台实现了 81%的准确率,219.9 FPS 的识别速度,相较于同类型的加速器识别速度最高提升了 33 倍。

1 二值卷积神经网络加速器

二值卷积神经网络激活与权值均采用符号函数进行二值化,如式(1)所示:

$$w^{b}=\mathrm{sign}(w)=\begin{cases}+1, w\geqslant 0\\ -1, w<0\end{cases} \tag{1}$$

其中 w 为单精度浮点权重,w^{b} 为二值权重。在硬件设计中若以逻辑 0 表示-1,逻辑 1 表示 1,则有:

(1)乘法运算可简化为 Xnor 运算和 PopCount 累加运算[6]。因此,对于特征图 r 行 c 列卷积核大小为(k_c,k_r)的卷积运算,如式(2)所示:

$$y_{\mathrm{out},r,c}=\sum_{i=0}^{k_r}\sum_{j=0}^{k_c}\mathrm{xnor}(w_{i,j}^{b},y_{r+i,c+j}^{b}) \tag{2}$$

(2)二值卷积神经网络的批归一化与符号激活函数运算过程如图 1 所示。

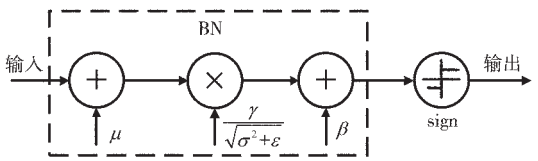

图 1 批归一化与激活

若结合归一化与符号激活函数即 $y=\mathrm{sign}\ (\mathrm{BN}(x))$,可得:

$$y=\begin{cases}+1, x\geqslant \mu-\dfrac{\beta}{\gamma}\sqrt{\sigma^{2}+\varepsilon}\\ -1, x<\mu-\dfrac{\beta}{\gamma}\sqrt{\sigma^{2}+\varepsilon}\end{cases}\quad(\gamma>0) \tag{3}$$

若以逻辑 0 表示-1,逻辑 1 表示 1,则阈值 T 可转换为:

$$T=\frac{1}{2}(\mathrm{ch_{in}}+\mu-\frac{\beta}{\gamma}\sqrt{\sigma^{2}+\varepsilon}) \tag{4}$$

其中 x 为前一层卷积层输出,μ、σ 是批量输入的均值和方差,ε、γ、β 为参数,$\mathrm{ch_{in}}$ 表示输入通道数。

综上所述,二值卷积运算单元(Processing Element,PE)计算流程如下:输入特征图与权值经过同或门与累加器进行卷积运算,再经阈值比较器实现批归一化与激活函数运算,硬件结构如图 2 所示。

卷积层包含了多通道输入与多通道输出。因此,单层计算引擎通常由 PE 阵列构成,如图 3 所示,计算引擎从缓冲区读入 SIMD 通道特征图,经 PE 阵列并行计算得到多个输出到缓冲区。

基于数据流结构的加速器,通过层间流控模块,逐层将二值卷积计算引擎连接起来,整体结构如图 4 所示,通过调节各层 SIMD 与 PE 参数,可以实现性能与逻辑资源的最优化。

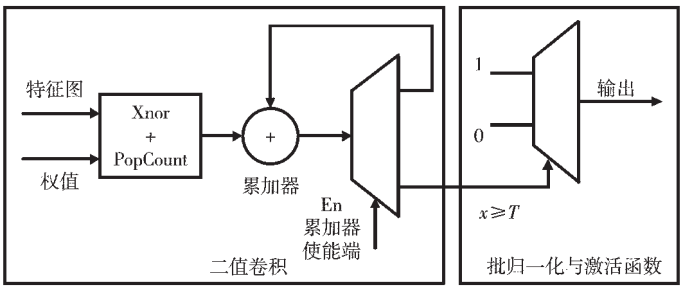

图 2 二值卷积运算单元

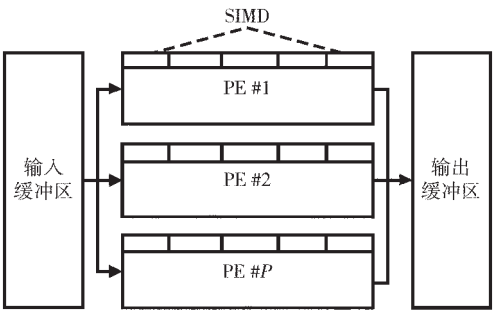

图 3 二值卷积计算引擎

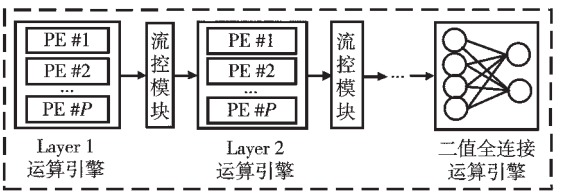

图 4 数据流架构

2 优化设计

针对二值卷积神经网络加速器存在的资源瓶颈以及性能瓶颈,需要从网络拓扑、流水线运算周期均衡等多方面进行优化设计:

(1)由于硬件资源限制、网络结构以及大量的网络参数,往往造成片上存储资源瓶颈,因此需要首先针对网络结构进行优化。

(2)由于不同网络层运算量各不相同,运算所需周期也不同,因此需要针对流水线进行逐层的运算优化,平衡每层的运算周期。

2.1 网络结构优化

原始 VGG-11 的网络拓扑中的首个全连接 Fc1 层参数量显著高于其余各层,约占网络整体参数量 79%。由于其参数量过大,既造成了片上内存资源瓶颈又导致计算量过大,与其余各层计算周期严重不均衡,使流水线阻塞造成性能瓶颈。针对上述问题,对 VGG-11 网络结构的瓶颈层进行优化:

(1)对原始浮点卷积 VGG-11 进行二值化,以有效降低内存占用以及逻辑资源数量。

(2)在卷积层与 Fc1 层之间添加全局最大池化层,将卷积层输出特征图从 7×7 池化到 1×1。

优化后的二值 VGG-11 网络拓扑如表 1 所示，添加全局最大池化层(Global Max Pool)后，Fc1 层参数量降低了约 49 倍，同时由于对网络进行了二值化，整体网络参数所占内存空间从 511.3 MB 降低到 3.66 MB，因而有效地从网络结构层面降低了内存资源瓶颈。

表 1 二值 VGG-11 网络拓扑

名称	输入特征图	核	输出特征图	参数
Input	224×224×3	/	224×224×3	/
Conv1	224×224×3	3×3	224×224×64	1728
MaxPool	224×224×64	2×2	112×112×64	/
Conv2	112×112×64	3×3	112×112×128	73728
MaxPool	112×112×128	2×2	56×56×128	/
Conv3	56×56×128	3×3	56×56×256	294912
Conv4	56×56×256	3×3	56×56×256	589824
MaxPool	56×56×256	2×2	28×28×512	/
Conv5	28×28×512	3×3	28×28×512	2359296
Conv6	28×28×512	3×3	28×28×512	2359296
MaxPool	28×28×512	2×2	14×14×512	/
Conv7	14×14×512	3×3	14×14×512	2359296
Conv8	14×14×512	3×3	14×14×512	2359296
MaxPool	14×14×512	2×2	7×7×512	/
Global MaxPool	7×7×512	7×7	512	/
Fc1	512	/	4096	2097152
Fc2	4096	/	4096	16777216
Fc3	4096	/	10	40960

2.2 流水线优化

基于数据流架构示意图如图 5 所示，Initiation Interval 为两个任务间的时间间隔，Latency 为整体任务完成的延迟。由于采用数据流架构，网络加速器的吞吐率可以采用 F_{clk}/II_{max} 来进行估算。计算延迟最慢的网络层会导致任务间的时间间隔最大为 II_{max}，从而决定了网络的吞吐率。

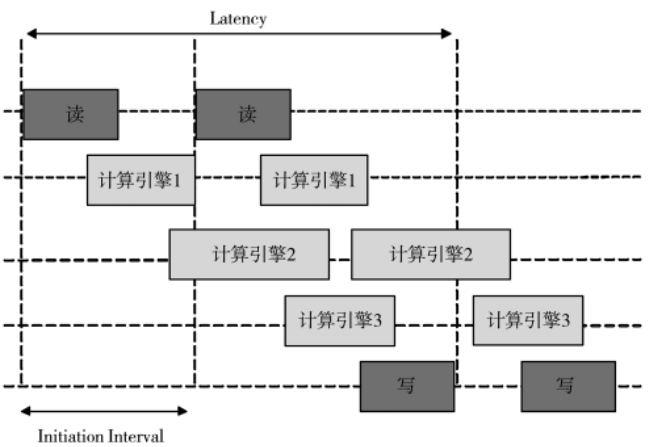

图 5 流水线时序图

根据上述分析可知，消耗时钟周期数最多的计算引擎会成为整体性能的瓶颈，从而会造成其他层资源的浪费和性能的下降。因此，针对流水线优化，需要针对不同的计算引擎之间进行整体的计算周期均衡，尽可能地保证各层的计算周期相近。

为了有效提高加速器的性能与资源利用率，本文设计了不同的 PE 阵列参数配置，以验证不同的 PE 和 SIMD 配置对分类效率的影响，表 2 中给出的计算阵列结构参数，A 是最低速的配置，B、C、D、E 依次增加了 PE 以及 SIMD，E 是根据调整得到的最好的结果。

表 2 PE 阵列配置

网络层	A	B	C	D	E
Conv1	P=8	P=16	P=32	P=64	P=64
	S=3	S=3	S=3	S=3	S=3
Conv2	P=8	P=16	P=16	P=32	P=64
	S=32	S=32	S=64	S=64	S=64
Conv3	P=8	P=16	P=16	P=32	P=64
	S=32	S=32	S=64	S=64	S=64
Conv4	P=16	P=32	P=32	P=64	P=128
	S=32	S=32	S=64	S=64	S=64
Conv5	P=8	P=16	P=32	P=32	P=64
	S=32	S=32	S=32	S=64	S=64
Conv6	P=16	P=16	P=32	P=64	P=128
	S=32	S=64	S=64	S=64	S=64
Conv7	P=4	P=8	P=8	P=16	P=32
	S=32	S=32	S=64	S=64	S=64
Conv8	P=4	P=8	P=8	P=16	P=32
	S=32	S=32	S=64	S=64	S=64
Fc1	P=1	P=1	P=1	P=1	P=1
	S=4	S=4	S=4	S=8	S=16
Fc2	P=2	P=1	P=1	P=1	P=1
	S=8	S=8	S=8	S=16	S=32
Fc3	P=1	P=1	P=1	P=1	P=1
	S=2	S=2	S=2	S=4	S=8

如表 3 所示，根据表 2 中 SIMD 及 PE 参数所对应的各网络层计算周期，通过尽可能将各网络层运算周期均衡调整，从而可以在相应的资源占用率下实现最大化加速器推断速率。

表 3 运算周期

网络层	A	B	C	D	E
Conv1	3612672	1806336	903168	451584	451584
Conv2	3612672	1806336	903168	451584	225792
Conv3	3612672	1806336	903168	451584	225792
Conv4	3612672	1806336	903168	451584	225792
Conv5	3612672	1806336	903168	451584	225792
Conv6	3612672	1806336	903168	451584	225792
Conv7	3612672	1806336	903168	451584	225792
Conv8	3612672	1806336	903168	451584	225792
Fc1	131072	131072	131072	65536	32768
Fc2	131072	131072	131072	65536	32768
Fc3	32768	32768	32768	16384	8192

3 结果

在 Ubuntu16.04 操作系统下，基于 Pytorch 深度学习框架训练二值 VGG-11 卷积神经网络，实验基于 CIFAR-10

数据集验证，将数据集图像尺寸放大到224×224作为网络输入，数据训练利用NVIDIA Quadro P2000 GPU实现加速。基于流架构二值VGG-11加速器硬件系统开发基于ZCU102开发板，最终硬件系统实现了81%的识别率，推断速率、资源占用率等如表4所示，最高实现了219.9 FPS。

表4 资源利用率对比

结构	速率/(FPS)	FPGA资源利用率评估/%			
		BRAM_18K	FF	LUT	URAM
A	18	55.70	29.50	27.50	100
B	44	60.14	30.22	30.21	100
C	96	63.65	32.32	35.69	100
D	198	74.23	35.69	46.19	100
E	219.9	93.31	41.74	64.8	100

通过实验对比可得出如下结论：

(1)逐渐增加PE或SIMD的数量能提高深度神经网络加速器的推断速率，但会占用更多逻辑资源，反之也可以通过降低推断速率来换取逻辑资源占用面积的缩减。

(2)比较方案E和方案D，除Conv1卷积层外，其余各层均提高了SIMD和PE数量以及缩减了计算周期，然而对比实现结果，可以发现逻辑资源占用率有了大幅增长，而推断速度却并没有得到大幅提升。这验证了针对于流水线结构的深度卷积神经网络加速器来说，计算周期延迟最大的计算引擎对网络整体性能有较大的影响，在设计中对各层运算单元计算周期进行均衡尤为重要。

(3)对比FPGA片上资源LUT、FF以及BRAM等资源，片上内存数量是限制进一步提高神经网络层数以及提高推断速度的资源瓶颈。

与国内外相关基于FPGA的VGG网络加速器实现进行比较，如表5所示。通过优化设计，实现了相较于其他VGG加速器最高33倍推断加速，相比基于层架构的同类型二值VGG网络加速器[8]提高了7倍。

表5 基于FPGA的VGG加速器对比

方案	平台	时钟/MHz	帧速/(FPS)	功耗/W
文献[8]	Zynq Ultrascale+	100	31.8	22
文献[9]	Stratix-V GSD8	120	3.80	19.10
文献[10]	Zynq XC7Z045	150	4.45	9.63
文献[11]	Virtex-7 VX690t	150	6.58	-
本文方案	Zynq Ultrascale+	100	219.9	4.46

4 结论

本文通过从网络结构、流水线均衡等多方面优化设计，实现了输入尺度更大的二值VGG-11卷积神经网络加速器，并验证了优化方法的有效性，为更大尺度、更深层次的卷积神经网络加速器提供了设计优化思路。

参考文献

[1] COURBARIAUX M, HUBARA I, SOUDRY D, et al. Binarized neural networks: training deep neural networks with weights and activations constrained to+1 or-1[J]. arXiv preprint arXiv: 1602.02830, 2016.

[2] ZHOU S, WU Y, NI Z, et al. Dorefa-net: training low bitwidth convolutional neural networks with low bitwidth gradients[J]. arXiv preprint arXiv: 1606.06160, 2016.

[3] LIN X, ZHAO C, PAN W. Towards accurate binary convolutional neural network[C]. Advances in Neural Information Processing Systems. 2017: 345-353.

[4] SIMONS T, LEE D J. A review of binarized neural networks[J]. Electronics, 2019, 8(6): 661.

[5] GUO P, MA H, CHEN R, et al. FBNA: a fully binarized neural network accelerator[C]. 2018 28th International Conference on Field Programmable Logic and Applications (FPL). IEEE, 2018: 51-513.

[6] UMUROGLU Y, FRASER N J, GAMBARDELLA G, et al. Finn: a framework for fast, scalable binarized neural network inference[C]. Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2017: 65-74.

[7] LIANG S, YIN S, LIU L, et al. FP-BNN: binarized neural network on FPGA[J]. Neuron Computing, 2018, 275: 1072-1086.

[8] YONEKAWA H, NAKAHARA H. On-chip memory based binarized convolutional deep neural network applying batch normalization free technique on an fpga[C]. 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, 2017: 98-105.

[9] SUDA N, CHANDRA V, DASIKA G, et al. Throughput-optimized OpenCL-based FPGA accelerator for large-scale convolutional neural networks[C]. Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2016: 16-25.

[10] QIU J, WANG J, YAO S, et al. Going deeper with embedded fpga platform for convolutional neural network[C]. Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2016: 26-35.

[11] ZHANG C, WU D, SUN J, et al. Energy-efficient CNN implementation on a deeply pipelined FPGA cluster[C]. Proceedings of the 2016 International Symposium on Low Power Electronics and Design, 2016: 326-331.

(收稿日期：2020-12-15)

作者简介：

张旭欣(1995-)，男，硕士，主要研究方向：深度神经网络、FPGA。

张嘉(1997-)，男，硕士，主要研究方向：深度神经网络、图像处理。

金婕(1978-)，通信作者，女，博士，副教授，主要研究方向：视频编解码、数字信号处理和VLSI，E-mail：jinjie_pku@126.com。

版权声明