魏行健, 孙泽宇, 王正斌. 一种基于 PYNQ 的神经网络模型加速设计[J]. 智能计算机与应用,2025,15(1):69-74. DOI:10.20169/j.issn.2095-2163.250111

一种基于 PYNQ 的神经网络模型加速设计

魏行健[1], 孙泽宇[1], 王正斌[1,2]
(1 南京邮电大学 电子与光学工程学院、柔性电子(未来技术学院), 南京 210023;
2 射频集成与微组装技术国家地方联合工程实验室, 南京 210023)

摘 要: 针对卷积神经网络存在运算量大、资源要求高的问题,本文提出一种易于在移动端低功耗嵌入式设备上布置的二值化神经网络(Binary Neural Network,BNN)图像分类模型,并提供了其在 ARM(Advanced RISC Machines)+FPGA(Field Programmable Gate Array)异构系统上的硬件加速设计。通过将卷积的累乘加运算转化为简单的同或运算(Exclusive NOR, XNOR)和位计数运算(population count,popcount),降低了运算复杂度和片上资源要求;利用数据复用、流水线设计和并行计算提升整体运算速度;针对 CIFAR-10 数据集进行图像分类识别,利用 Vivado HLS 工具在 FPGA 平台上完成该网络模型的部署。在 PYNQ-Z2 平台上进行测试的实验结果显示,在 100 MHz 工作频率下,部署在 FPGA 端的网络模型对任意尺寸的图像输入经过 PS(Processing System)端裁剪后整体处理速度可达约 631 FPS,运行总时间仅约 1.58 ms。

关键词: FPGA; 图像分类; 神经网络; 硬件加速设计
中图分类号: TP274+.2 **文献标志码**: A **文章编号**: 2095-2163(2025)01-0069-06

A neural network model acceleration design based on PYNQ

WEI Xingjian[1], SUN Zeyu[1], WANG Zhengbin[1,2]
(1 College of Electronic and Optical Engineering, College of Flexible Electronics (Future Technology), Nanjing University of Posts and Telecommunications, Nanjing 210023, China; 2 National Joint Engineering Laboratory of RF Integration and Microassembly Technology, Nanjing 210023, China)

Abstract: Aiming at the problems of large computational complexity, time-consuming, and high resource requirements of convolutional neural network (CNN), this paper proposes a design scheme of binary neural network (BNN) image classification model running on embedded platforms with limited resources and power consumption in mobile terminals and designs a hardware acceleration design for its implementation on an ARM + FPGA platform. By converting the convolution multiply-accumulate operation into XNOR logic and popcount operations, the computational complexity and on-chip resource requirements are reduced. Data multiplexing, pipeline design, and parallel calculation were utilized to increase overall computation speed. Taking image recognition under the CIFAR-10 data set as an example, We use VIVADO HLS tool to complete the deployment of convolutional neural network model on FPGA platform. The test results on the PYNQ-Z2 platform show that the network model deployed on the FPGA side achieves a processing speed of approximately 631 FPS at a working frequency of 100 MHz, total runtime is only about 1.58 ms for image inputs of any size, after cropping on the processing system (PS) side.

Key words: FPGA; Image classification; neural network; hard-ware accelerator

0 引 言

近年来,卷积神经网络(Convolutional Neural Network, CNN)作为人工智能的重要成果,已被广泛应用在图像识别、目标检测、图像分割等任务上[1-3]。常用网络模型的布置需要大量计算和存储资源,对计算机硬件性能要求较高。

目前的主流硬件平台中, CPU(Central Processing Unit,CPU)受自身架构局限,不适合此类密集运算; GPU(Graphics Processing Unit, GPU)拥有较多计算单元,可以提供较高的并行度、浮点运算能力,但价格昂贵且功耗较大,难以应用在移动嵌入式平台; ASIC(Application-Specific Integrated Circuit)芯片通用性较差,且价格贵、研发周期长[4-5]。而 FPGA

作者简介: 魏行健(1997—),男,硕士研究生,主要研究方向:智能信号处理。
通信作者: 王正斌(1978—),男,博士,教授,主要研究方向:射频通信系统。Email: wangzb@njupt.edu.cn。
收稿日期: 2023-08-03

(Field Programmable Gate Array,FPGA)相对于 CPU 或 GPU,计算能力强、功耗低、具有可编程的硬件映射能力,能提供高度定制的硬件解决方案,具有独特且明显的优势。

目前,基于 FPGA 的神经网络加速器研究主要集中在卷积层计算、数据类型、模型结构等方面。李钦祚等[6]采用网络权重定点数优化、通道优化等方法在较低资源消耗的情况下实现了 YOLO 网络的部署;Courbariaux M 等[7]提出一种二值化的神经网络(Binary Neural Network,BNN),使用 1 bit 量化权重和激活函数,占用较小内存,几乎完全使用二进制运算的方法非常适合使用在 FPGA 上;Kim 和 Smaragdis 等[8]使用完全二值化的神经网络,实现了 MNIST 数据集上 98.7%的准确率;Li 等[9]在 BNN 模型上实现了 ImageNet 较大尺寸数据集上较高的精度,进一步提高了资源有限情况下 BNN 网络的竞争力;李佳骏等[10]提出一套二值化网络的设计及训练方案,通过模拟溢出的矩阵乘法、细化设计位移批标准化层、降低访存等方式,较好保证网络准确率无明显损失的情况下,减少了片上计算规模,有效提升了加速器性能。

基于以上背景,本文提出一种适合在移动嵌入式设备上部署的二值神经网络,有效控制模型尺寸并设计了其并行加速系统,利用流水线设计、并行设计等优化网络计算,实现了约 630 FPS 的识别速度和 86%的准确率。

1 网络模型整体架构

1.1 网络结构

本文采用的二值化神经网络是一种对 CNN 模型进行二值化压缩的网络,将网络中的特征图和卷积核权重参数进行二值化,最终压缩成为“+1”和“-1”,以实现 1 bit 位宽数据的存储,并用 XNOR 和 popcount 进行累乘加计算以减少参数,降低复杂度,与 FPGA 内部硬件的逻辑设计非常贴合[11]。

为利于硬件实现,使用确定性二值化方法如下式所示:

$$x^b = \mathrm{sign}(x) = \begin{cases} +1, & \text{if } x \geqslant 0 \\ -1, & \text{otherwise} \end{cases} \tag{1}$$

其中,x^b 表示二值化之后的权重和特征图数据,x 表示原始的浮点数值。

为了减少在网络传输过程中由二值化运算带来的分布信息丢失问题,还添加了批归一化层(Batch Normalization,BN),如下式所示。通过将输入数据进行线性位移和缩放,使其具有零均值和单位方差,减少了激活过程中的信息丢失,在保证原始数据分布信息稳定的同时也可以加速训练过程。

$$y = \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} \gamma + \beta \tag{2}$$

其中,x、y 表示输入和输出;μ 为平均值;σ^2 是批的方差;ε 为增加数值稳定而添加的一个很小的数,保证分母不为 0。

那么判断输出 y 的二值可以表示如下:

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

下式表示第 m 张输出特征图中第 p 行,第 q 列关于 N_{in} 个 $K \times K$ 大小的卷积核累乘加计算:

$$x^m_{(p,q)} = \sum_{n=1}^{N_{\text{in}}} \sum_{i=1}^{K} \sum_{j=1}^{K} x_{\text{NOR}}(I^n_{p+i,q+j}, W^{n,m}_{(i,j)}) \tag{4}$$

1.2 网络整体架构及训练过程

本文模型中,神经网络模型包括 6 个卷积层、3 个池化层和 3 个全连接层,卷积层和全连接层后均有 BN 层和激活层,池化层后添加 BN 层,卷积神经网络结构见表 1。

输入任意图像尺寸由 PS 端裁剪为 32×32 像素的 3 通道 RGB 图像进入卷积层;卷积层使用较小尺寸的 3×3 卷积核,保证感受野不变的情况下减少存储和计算上的负担;池化层选用最大池化,池化窗口大小为 2×2,步长为 2。

本文训练采用 CIFAR-10(Canadian Institute for Advanced Research)数据集,这是一个用于图像识别的经典数据集,包含 10 个类别的 60 000 张 32×32 像素的彩色图片,每个类别有 6 000 张图片,50 000 张图片用于训练,10 000 张用于测试。

神经网络的搭建和参数的训练在 Pytorch 中进行,SGD(Stochastic Gradient Descent)优化器设置反向传播函数,Batch Size 设置为 128,学习率设置为 0.001,训练 200 个 epoch 后,配合自适应学习率算法,每 50 个 epoch 学习率下降 50%,500 个 epoch 后在 CIFAR-10 数据集上识别准确率达到了 86%。

ARM + FPGA 上定制化的神经网络整体架构如图 1 所示,神经卷积网络各计算模块实现后封装为 IP 调用,PS 与 PL(Progarmmable Logic) 之间的数据传输通过 AXI DMA 协议发送。PS 端对任意尺寸的图像裁剪为 32×32 图像后发送至 PL 端运算后,输出结果由 AXI4-Lite 接口发送给 PS 端。

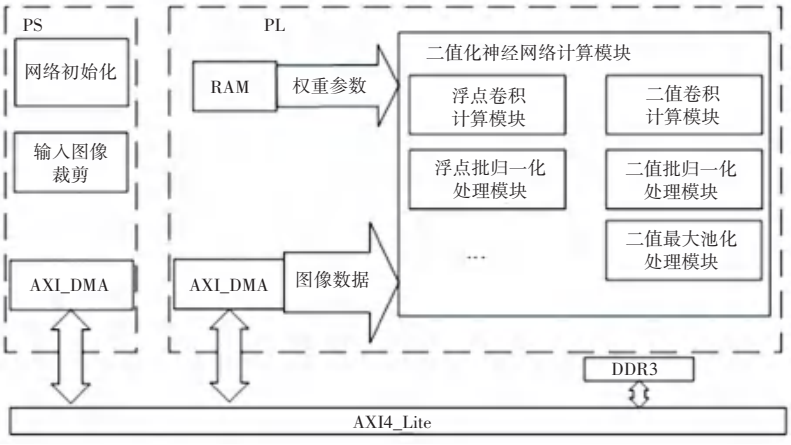

图 1　整体架构

Fig. 1　Overall architecture

表 1　卷积神经网络结构

Table 1　Architecture of convolutional neural network

模型结构	输入图像尺寸	填充	输出图像尺寸	步长	卷积核大小	输入通道数	输出通道数
conv1	32×32	1	32×32	1	3×3	3	128
conv2	32×32	1	32×32	1	3×3	128	128
maxpool1	32×32	0	16×16	2	2×2	128	128
conv3	16×16	1	16×16	1	3×3	256	256
conv4	16×16	1	16×16	1	3×3	256	256
maxpool2	16×16	0	8×8	2	2×2	256	256
Conv5	8×8	1	8×8	1	3×3	256	512
Conv6	8×8	1	8×8	1	3×3	512	512
maxpool3	8×8	0	4×4	2	2×2	512	512
全连接层 FC1	1		1			8 196	1 024
全连接层 FC2	1		1			1 024	1 024
全连接层 FC3	1		1			1 024	10

模型在 PYNQ 平台中实现时，卷积神经网络的权重和偏置以及输入特征存储在 FPGA 内部的块寄存器中，系统初始化读取网络参数、卷积核信息，并放入缓存中，等待各神经网络 IP 使用。同时将图像信息通过 AXI DMA 协议以数据流的方式存储到片外存储器（DDR3），再发送给各 IP，PS 端 AXI_DMA 模块控制片外存储器中数据的存储位置以及相应的读写操作。

工作时，PS 端将各层的权重和偏置以及输入特征由 DDR3 加载给 PL 端进行各层运算，输出结果由 AXI4-Lite 接口传输回 PS 端。PYNQ 可连接 PC 端通过 Jupyter Notebook 进行系统测试，直接在 Jupyter Notebook 上控制 DMA 发送图像数据和接收最后的分类结果。

2　硬件架构设计

2.1　二值卷积计算模块硬件加速设计

卷积层对输入特征图进行特征提取，卷积计算也是卷积神经网络中计算量的主要来源[12]。由于第一层卷积层的输入为二值化的定点数，之后的卷积计算均为二值化卷积计算，所以分为定点卷积和二值卷积两类模块。特征图按行顺序输入，配合 3×3 卷积核步长为 1，依次进行计算。

定点数卷积接收到 PS 端裁剪后发送的 3 通道 32×32 像素尺寸的图片，与二值化的权重进行卷积，之后进行批归一化和二值化。定点卷积和二值卷积均采取完全并行的计算方法，利用 HLS（High-Level Synthesis）工具分别对输入特征图像数据与权重进

行预读取数组分割指令进行优化,图像数据与卷积核数据先缓存后并行相乘相加。卷积层并行加速设计优化方案如图 2 所示,将模型中卷积核的内部运算在 FPGA 中进行完全展开,使卷积核内的计算并行,以 3×3 卷积核为例,输入通道展开,每个时钟提供卷积核大小数量的也就是 9 个输入通道的数据,在一个卷积核内的 9 个乘法运算也完全展开,在一个周期内完成,并配合并行加法器实现卷积核内各元素计算的并行。多输入情况下卷积核内运算的多通道并行计算设计如图 3 所示。

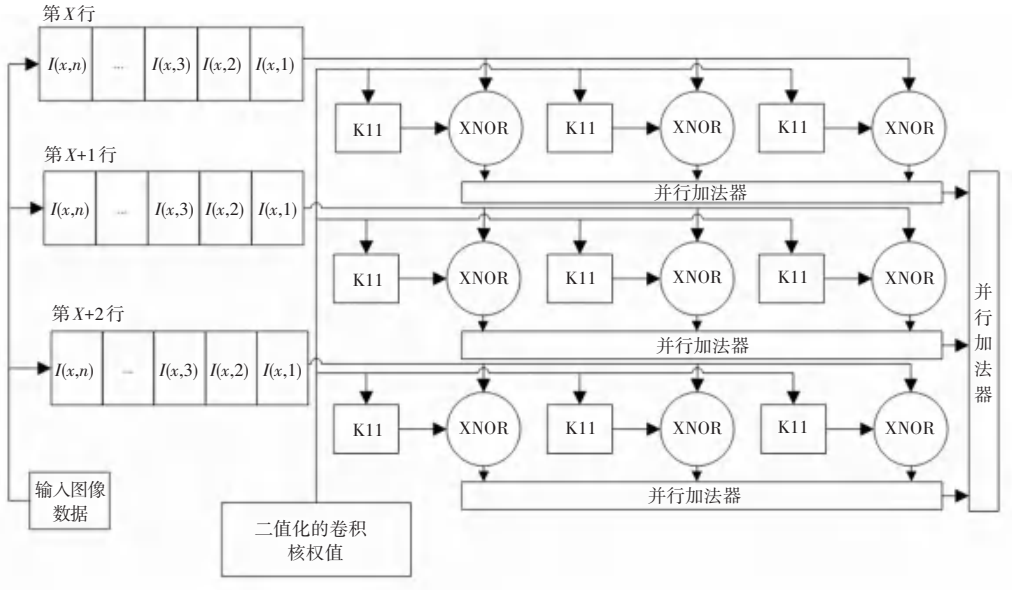

图 2 卷积层并行加速设计

Fig. 2 Convolution layer parallel acceleration design

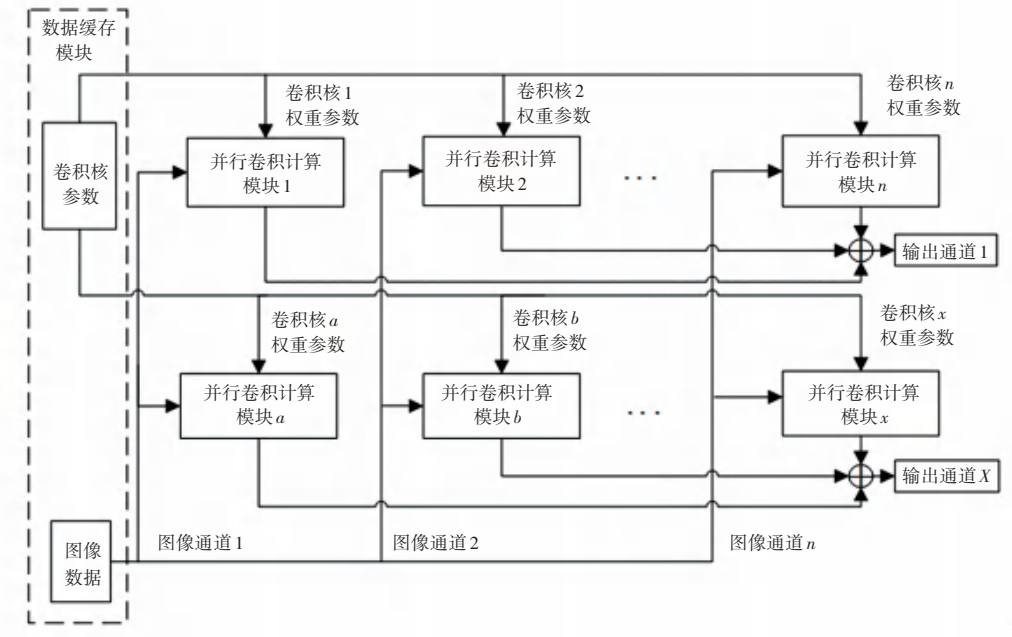

图 3 多通道并行计算设计

Fig. 3 Multi-channel parallel computing design

2.2 池化层加速设计

池化层对输入特征图降采样,保留图片特征的同时减低特征图尺寸,加速运算过程,选用最大池化,2×2 最大池化可用下式表示:

$$P_{\text{out}}(w,h,j)=\max_{0\leqslant p\leqslant 2}\{P_{\text{in}}(2w+p,2h+P,j)\} \quad (5)$$

其中,w 是输出特征图的列元素;h 是输出特征图的行元素;j 为输出特征图的通道元素。

随着窗口的移动,比较器更新最大值作为输出,池化层流水线化设计如图 4 所示。

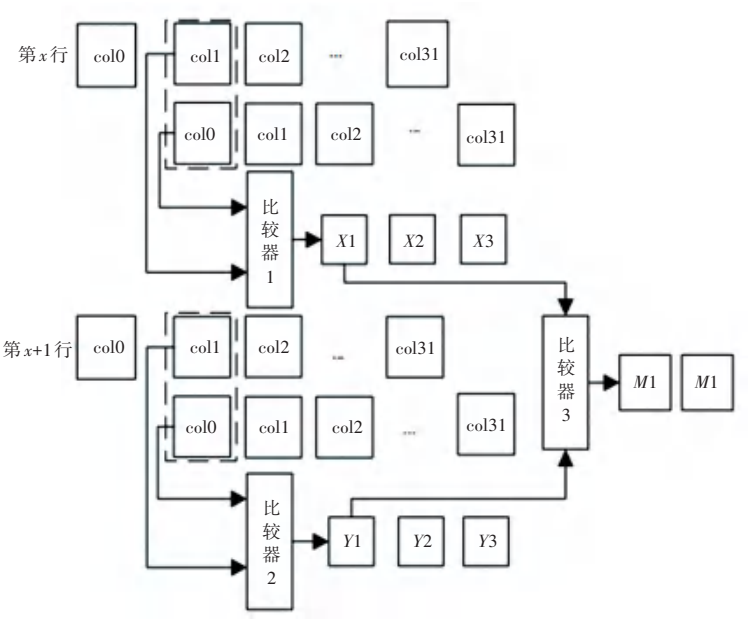

图 4　池化层流水线设计

Fig. 4　Pooling layer pipeline design

2×2 最大池化中，将一行的输入数据（col0）在第一个时钟暂时寄存在一个新寄存器中，第二个时钟取出寄存器中数据（col0）和输入数据（col1）进入比较器比较，获得该列最大值 $X1$，同时相邻的另外一行并行一个相似原理的两列元素进入比较器，比较得到列最大值（$M1$），两个列最大值再继续进入第三个比较器取得该 2×2 池化的最大值输出，形成完整的最大池化计算的流水线结构，能够有效加快比较速度。

3　实验结果与分析

本系统硬件平台选择 Xilinx PYNQ-Z2 开发板，采用了 Xilinx 的 Zynq-7000 系列 Zynq SoC 芯片 XC7Z020-CLG400，ARM Cortex-A9 处理器作为 PS 端，工作频率 100 MHz。使用 Python 编程语言和 Jupyter Notebook 交互式编程进行硬件设计和嵌入式应用程序开发。使用 Vivado HLS 以及 Vivado 2018.2 工具进行综合、布局布线、生成比特流。

对电路各端口添加约束，进行综合和实现，测试使用任意尺寸的图片作为输入，记录从图片输入、裁剪、传送至 PL 端完成识别分类全部运算的整体时延。

3.1　片上资源使用情况

根据 Vivado 2018.2 提供的综合报告对 FPGA 片上资源利用情况进行分析。本文图像分类加速器设计在 PYNQ-Z2 开发板的片上资源利用情况见表 2，可以看出乘法运算逻辑使用 DSP（Digital Signal Processor）资源使其利用率较高，达到近 91%。在片上存储的权重数据较多，使得 BRAM（Block Random Access Memory）利用率也达到了 72.1%；此外 FF（Flip-Flop）利用率为 23%，LUT（Look Up Table）也达到了 60.6%的利用率，本文设计片上资源整体利用率较高。

表 2　FPGA 资源使用情况

Table 2　FPGA resource consumption

资源类型	总量	使用数量	使用率/%
DSP	220	200	90.9
FF	106 400	24 472	23.0
LUT	53 200	32 266	60.6
BRAM	140	101	72.1

3.2　性能测试与分析

通过板级测试，得到系统的性能在软件端（PS）和本文设计系统中运行性能对比见表 3，可见本文的二值化网络在 ARM 端进行计算时延较长，而在加速硬件系统端则实现了低时延，实现了 631 FPS 的高帧率。

表 3　运行性能对比

Table 3　Comparison of running performance

性能	PS 端运行方案	本文运行方案
一帧图像处理时间/ms	3 212.56	1.58
每秒处理图像帧数/FPS	0.31	631.70

由于不同文献使用不同 FPGA 器件和不同的网

络结构,本文选取测试用数据集、平台频率、运算时延、DSP 数量和识别率作为比对标准,对各类现有的图像识别硬件加速器进行比较,结果见表 4,可以看出本文方案在实现 86%正确率的情况下获得 631.7 FPS 的高帧率,相较于其他方案和优化设计,有较好的低时延优势。

表 4 本文方案与其他方案的实验对比

Table 4 Experimental comparison between this scheme and other schemes

参数	实验平台	数据集	频率/MHz	运算时延/ms	DSP 总量	识别率/%
文献[13]	XILINX ZC702	MNIST	143	4.10	220	98
文献文献[14]	PYNQ-Z2	MNIST	100	31.00	220	98
文献文献[15]	Zynq7020	CIFAR-10	100	15.50	220	80
本文	PYNQ-Z2	CIFAR-10	100	1.58	220	86

4 结束语

本文针对在移动嵌入式平台部署可靠、低时延的图像分类神经网络算法的问题,提出了一种准确率较高,且计算时延低的二值神经网络,搭建并训练了分类模型。采用软硬协同的设计方法,在 PYNQ-Z2 平台上布置了整体网络,从网络结构、流水线设计、并行多通道计算等方面进行优化,精简网络模型的同时达到分类正确率与运算速度的平衡,并实现了其网络加速器的设计与验证。

参考文献

[1] KALA S, NALESH S. Efficient CNN accelerator on FPGA[J]. IEEE Journal of Research, 2020, 66(6): 733-740.

[2] 易啸, 马胜, 肖侬. 深度学习加速器在不同剪枝策略下的运行优化[J]. 计算机工程与科学, 2023,45(7): 1141-1148.

[3] MO Y, WU Y, YANG X, et al. Review the state-of-the-art technologies of semantic segmentation based on deep learning[J]. Neurocomputing, 2022, 493: 626-646.

[4] 刘卫明, 罗全成, 毛伊敏,等. 基于 Spark 和 AMPSO 的并行深度卷积神经网络优化算法[J/OL]. 计算机应用研究, 1-11 (2023-07-14). DOI:10.19734/j.issn.1001-3695.2023.03.0083

[5] BADAR M, HARIS M, FATIMA A. Application of deep learning for retinal image analysis: A review[J]. Computer Science Review, 2020, 35: 100203.

[6] 李钦祚, 肖灯军. 基于 FPGA 的低功耗 YOLO 加速器设计[J]. 电子设计工程, 2022, 30(20): 6-12.

[7] 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.

[8] SMARAGDIS P, KIM M. Bitwise neural networks[J]. arXiv preprint arXiv, 1601.06071, 2016.

[9] LI Z, NI B, ZHANG W, et al. Performance guaranteed network acceleration via high-order residual quantization[C]//Proceedings of the IEEE International Conference on Computer Vision. Piscataway,NJ:IEEE,2017: 2584-2592.

[10] 李佳骏, 许浩博, 王郁杰,等. 面向高能效加速器的二值化神经网络设计和训练方法[J/OL]. 计算机辅助设计与图形学报, 1-10(2023-07-14). http://kns.cnki.net/kcms/detail/11.2925.TP.20230705.1123.004.html

[11] KRIZHEVSKY A, HINTON G. The Cifar10 Dataset[EB/OL]. (2020-06-15). http://www.cs.toronto.edu/kriz/cifar.html

[12] CONG J, XIAO B. Minimizing computation in convolutional neural networks[C]//Proceedings of International Conference on Artificial Neural Networks. Cham:Springer,2014: 281-290.

[13] 王玉雷, 谢凯亮, 陈思赟, 等. 卷积神经网络硬件加速的通用性设计[J]. 计算机工程与科学, 2023, 45(4): 577-581.

[14] 肖望勇. 基于 FPGA 的神经网络设计与实现研究[D]. 株洲: 湖南工业大学, 2021.

[15] 许杰, 张子恒, 王新宇,等. 一种基于 Zynq 的 CNN 加速器设计与实现[J]. 计算机技术与发展, 2021, 31(11): 108-113.