基于 FPGA 的卷积神经网络并行加速器设计

王　婷，陈斌岳，张福海
（南开大学 电子信息与光学工程学院，天津 300350）

摘　要： 近年来，卷积神经网络在许多领域中发挥着越来越重要的作用，然而功耗和速度是限制其应用的主要因素。为了克服其限制因素，设计一种基于 FPGA 平台的卷积神经网络并行加速器，以 Ultra96-V2 为实验开发平台，而且卷积神经网络计算 IP 核的设计实现采用了高级设计综合工具，使用 Vivado 开发工具完成了基于 FPGA 的卷积神经网络加速器系统设计实现。通过对 GPU 和 CPU 识别率的对比实验，基于 FPGA 优化设计的卷积神经网络处理一张图片的时间比 CPU 要少得多，相比 GPU 功耗减少 30 倍以上，显示了基于 FPGA 加速器设计的性能和功耗优势，验证了该方法的有效性。

关键词： 并行计算；卷积神经网络；加速器；流水线
中图分类号： TN402　　**文献标识码：** A　　DOI：10.16157/j.issn.0258-7998.200858

中文引用格式： 王婷，陈斌岳，张福海. 基于 FPGA 的卷积神经网络并行加速器设计[J]. 电子技术应用，2021，47(2)：81-84.
英文引用格式： Wang Ting，Chen Binyue，Zhang Fuhai. Parallel accelerator design for convolutional neural networks based on FPGA[J]. Application of Electronic Technique，2021，47(2)：81-84.

Parallel accelerator design for convolutional neural networks based on FPGA

Wang Ting，Chen Binyue，Zhang Fuhai
（College of Electronic Information and Optical Engineering，Nankai University，Tianjin 300350，China）

Abstract： In recent years，convolutional neural network plays an increasingly important role in many fields. However，power consumption and speed are the main factors limiting its application. In order to overcome its limitations，a convolutional neural network parallel accelerator based on FPGA platform is designed. Ultra96-v2 is used as the experimental development platform，and the design and implementation of convolutional neural network computing IP core adopts advanced design synthesis tools. The design and implementation of convolutional neural network accelerator system based on FPGA is completed by using vivado development tools. By comparing the recognition rate of GPU and CPU，the convolutional neural network based on FPGA optimized design takes much less time to process a picture than CPU，and reduces the power consumption of GPU by more than 30 times. It shows the performance and power consumption advantages of FPGA accelerator design，and verifies the effectiveness of this method.

Key words： parallel computing；convolutional neural network；accelerator；pipeline

0 引言

随着人工智能的快速发展，卷积神经网络越来越受到人们的关注。由于它的高适应性和出色的识别能力，它已被广泛应用于分类和识别、目标检测、目标跟踪等领域[1]。与传统算法相比，CNN 的计算复杂度要高得多，并且通用 CPU 不再能够满足计算需求。目前，主要解决方案是使用 GPU 进行 CNN 计算。尽管 GPU 在并行计算中具有自然优势，但在成本和功耗方面存在很大的缺点。卷积神经网络推理过程的实现占用空间大，计算能耗大[2]，无法满足终端系统的 CNN 计算要求。FPGA 具有强大的并行处理功能，灵活的可配置功能以及超低功耗，使其成为 CNN 实现平台的理想选择。FPGA 的可重配置特性适合于变化的神经网络网络结构。因此，许多研究人员已经研究了使用 FPGA 实现 CNN 加速的方法[3]。本文参考了 Google 提出的轻量级网络 MobileNet 结构[4]，并通过并行处理和流水线结构在 FPGA 上设计了高速 CNN 系统，并将其与 CPU 和 GPU 的实现进行了比较。

1 卷积神经网络加速器的设计研究

1.1 卷积神经网络的介绍

在深度学习领域中，卷积神经网络占有着非常重要的地位，它的图像识别准确率接近甚至高于人类的识别水平。卷积神经网络是同时具有层次结构性和局部连通性的人工神经网络[5]。卷积神经网络的结构都是类似的，它们采用前向网络模型结构，节点使用神经元来实现分层连接。并且，相邻层之间的节点是在局部区域内相连接，同一层中的一些神经元节点之间是共享连接权

重的。传统的卷积神经网络结构如图 1 所示，卷积神经网络是直接将准备识别的原始图像作为输入，然后依次通过多个隐藏层连接到各层，得到识别结果。

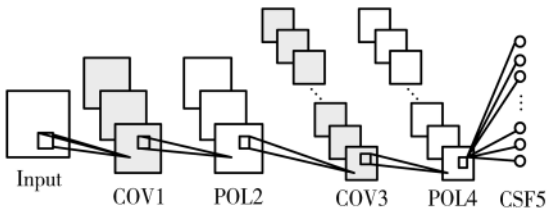

图 1 卷积神经网络典型结构

1.2 CNN 的结构框架

MobileNet 是用于移动和嵌入式设备的有效模型。MobileNet 基于简化的架构，并使用深度可分离卷积来构建轻型深度神经网络。为了进一步减少参数数量并促进在 FPGA 上的部署，本文中使用了经过修改的 CNN 网络结构，如图 2 所示。共有 9 个卷积层和 3 个池化层。

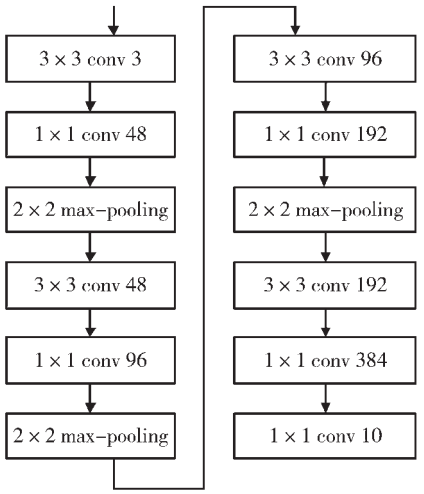

图 2 卷积神经网络结构

1.3 卷积模块的设计

在卷积神经网络中，卷积运算占据了大部分计算量。传统的卷积分为两个步骤，即每个卷积核与每张特征图片进行按位乘法然后相加，此时的计算量为 DF * DF * DK * DK * M * N，DF 是输入特征图的尺寸，DK 是卷积核的尺寸，M、N 分别是输入通道数和输出通道数。本文采用的卷积方式不同于传统卷积，首先按照输入通道进行按位相乘，得到的结果通道数是没有变化的，接下来使用 1 * 1 的卷积核再进行计算，以改变通道数。这种方法的计算量为 DK * DK * M * DF * DF+1 * 1 * M * N * DF * DF，第 1 项表示的是卷积核为 3 时的计算量，第 2 项表示卷积核为 1 时的计算量，当 DK=3 时，这种卷积方式比传统卷积减少了 8 倍多的计算量，计算量的大幅度减少更有利于部署在资源有限的 FPGA 端。运算一个卷积层需要 6 个循环嵌套来实现，循环顺序按照输出通道>输入通道>高度>宽度>卷积内核元素依次来排列计算。对于每一卷积层来说，最外面的循环按照顺序遍历所有像素。上述循环结构的优化实现可以使用循环展开，循环拆分以及循环合并的指令方法，以设计加速器的 IP 核。

1.4 资源占用优化

在训练了卷积神经网络之后，参数数据是一个 32 位浮点数。相关实验已经证实，精度降低一定程度对 CNN 识别精度的影响非常微弱[6]。因此，本文设计中经过尝试不同量化位数后，在保证了精度的情况下选择输入的图像数据和权重数据使用 9 位定点数。这种设计大大降低了 FPGA 资源的利用率，并提高了网络运行速度。

卷积神经网络的计算成本主要有卷积层的大量乘法运算，在 FPGA 中通常使用 DSP 资源进行乘法运算，而通常不足的 DSP 资源会成为卷积神经网络部署在 FPGA 端的瓶颈。BOOTH 算法实现的乘法器可有效地代替使用 DSP 资源的传统乘法。在 Vivado HLS 中，数据都是以十六位二进制带符号的补码表示，原码乘法器的移位相加方法并不能直接推广用于补码的乘法运算中。普通的移位相加运算量比较大，乘数的每一位都产生部分积，乘数中值为 1 的位数决定着累加的次数。BOOTH 算法的思想是将乘数近似为一个较大的整数值，利用这个整数值与被乘数相乘的结果减去这个整数值的补数与被乘数相乘的结果，对于具有连续的 1 和 0 的乘数来说产生的部分积较少。具体运算步骤如下：

(1)被乘数 X 与乘数 Y 均为有符号数补码，运算结果也是补码。

(2)初始部分积为 0，乘数 Y 末尾添加附加位，初始值为 0。

(3)判断乘数 Y 后两位：若是 01 则部分积加被乘数 X 再右移一位，若是 10 则部分积减被乘数 X 再右移一位，若是 00 以及 11 则只进行右移一位操作。

(4)累加 n+1 次(n 表示数据数值位数)，右移 n 次。

2 基于 FPGA 的加速器系统设计

2.1 卷积神经网络层融合策略

卷积层之间的运算有两种实现模式，分为层串行模式和层并行模式[7]。本文在设计基于 FPGA 的 CNN 加速器时，选择了高度的灵活性和实现难度低的层串行模式。

在层串行模式中，FPGA 中的所有 PE 单元都只用于实现卷积神经网络中一层的功能。并且通过重复调用存在的 PE 单元，即使用时分复用 PE 单元的策略来实现整个神经网络的运算[8]。根据卷积神经网络单层操作的类似性原理，因此考虑由单层实现的层串行模式是确实可行的。并且，在这种操作模式下，从 DDR 中读取数据传输给 PE 单元，PE 单元计算得到结果后将其写回到 DDR，数据控制比较简单。然而，对于中间数据的存储，

层串行模式是通过 AXI 总线协议将每一层的中间运算结果都再传输到外部存储器 DDR 中，因此这种方法对 IO 带宽的要求非常高[9]。

为了增大吞吐量并解决因带宽瓶颈而造成的传输时间过长，可以减少每一层的数据访问以及存储空间，以实现最大程度的数据和模块复用。因此，本文将每三层合并为一组，然后将结果输出到 DDR，从而将 12 层 CNN 结构减少为 5 层，这将节省一部分传输步骤。此操作将多层融合在一起而形成局部组合的方法，将从 DRAM 接收的输入数据和操作的中间结果缓存都存储在片上 BRAM 存储器中。

2.2 缓存结构

在带宽瓶颈的影响下，整个硬件平台的加速性能主要受到数据的访存效率限制。为了有效控制数据流的访存将使用缓冲技术，以增加带宽利用率[10]。乒乓操作的缓冲方式是使用两个数据存储器，先将数据存储在第一个数据缓存中，当第一个数据缓存存满时，数据将转换到第二个数据缓存中存储，并在相同时刻读取第一个数据缓存中的数据。这种方式使得单通道的数据传输有效地变化为双通道的数据流传输，数据流经过缓冲后，不断地传递到数据处理模块，这将使数据传输时间与数据运算时间重叠，以抵消大部分的时间[11]。

为了提高加速器系统的吞吐效率，在片内的输入缓存设置了图像输入缓存和权值输入缓存，以及结果输出缓存。输入缓存的作用是从外部存储器 DDR 中载入所需数据以及所需参数，输出缓存的作用是将存储运算结果输出至外部存储器 DDR 中或者是再应用于计算单元中。缓存结构根据 DMA 的方式来进行数据交互。本文的输入图像、权值以及输出的计算结果都采用如图 3 所示的乒乓缓冲方式。两个数据缓冲模块通过二选一复用器相互配合使用，使数据可以没有停顿地依次加载到计算单元中，计算单元可以时时刻刻处于计算状态，以此充分利用了有限的计算资源。

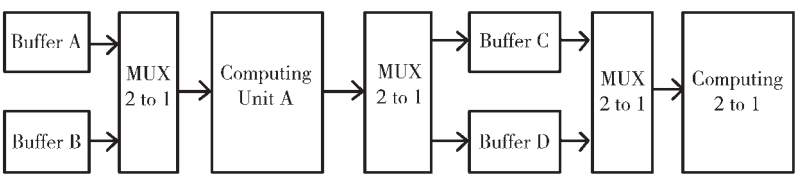

图 3 乒乓缓存数据流

2.3 加速器整体架构

加速器的总体设计如图 4 所示，由 PS 和 PL 组成。其中 PS 主要负责图像数据预处理，权重数据预处理和特征定位的任务，而 PL 负责整个 CNN 计算部分。加速器系统通过 AXI 总线将 CPU 和外部存储器 DDR 中的卷积神经网络参数权重，以及要识别的输入图像像素数据传递给 PL 部分。当操作控制指令传递到 PL 端时，PL 端启动系统主程序，并通过输入缓冲区的乒乓操作将参数和像素数据传输到运算操作逻辑单元。在完成整个卷积神经网络的计算后，输出数据通过 AXI 总线通过输出缓冲区传输到 DDR 存储器，并输出最终结果。

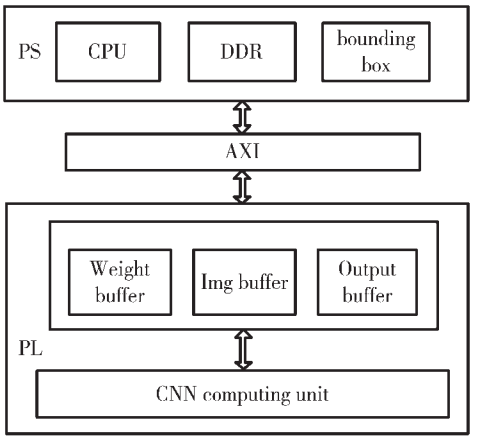

图 4 加速器系统的整体设计

3 基于 FPGA 的加速器系统设计

3.1 实验环境

实验采用 Xilinx Zynq UltraScale+MPSoC ZU3EG A484 Development Board 对本文目标检测定位算法进行加速。片内由 ARM 处理器与可重构 FPGA 构成，片上资源主要由 432 个 BRAM 和 360 个 DSP 组成。CPU 采用 Intel Core i5 2500K 处理器，GPU 是 NVIDIA UeForce UTX 960。所用到的软件开发工具为赛灵思公司开发的 Vivado 设计套件 Vivado IDE 和 Vivado HLS。

传统的 FPGA 设计流程复杂且繁琐，为了简化开发流程，加速器系统采用高级综合方式来进行优化设计[12]。首先采用 Vivado HLS 开发工具将 CNN 计算过程的高级编程语言 C++转化为硬件描述语言，再封装为 Vivado 的 IP 核输出。Vivado HLS 工具具体的设计流程如图 5 所示。然后利用 Vivado IDE 开发工具，导入封装好的 CNN 运算 IP 核、主控单元 zynq_ultra_ps、时钟单元以及 AXI 传输模块。通过综合、设定约束、布局布线来实现完成整个加速器系统的设计。

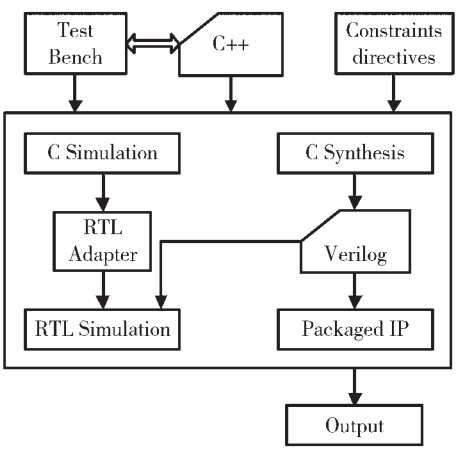

图 5 Vivado HLS 工具设计流程

3.2 实验环境

表 1 列出了默认乘法的 FPGA 的资源使用情况，表 2 列出了部分乘法用 BOOTH 算法代替的资源使用情况，由于开发板的 LUT 资源使用率已经很高，因此部分乘法还是采用了 DSP 资源。BRAM 用于图像数据、网络权重及输出数据的缓存，DSP 以及 LUT 用于卷积模块的乘加运算，该设计高效地利用了 FPGA 的内部资源。

表 1 默认乘法 FPGA 内部资源的利用率

类目	占资源数	总资源数	使用率/%
DSP	329	360	91
BRAM	386	432	89
LUT	43 666	70 560	62
FF	46 458	141 120	33

表 2 BOOTH 乘法 FPGA 内部资源的利用率

类目	占资源数	总资源数	使用率/%
DSP	265	360	74
BRAM	386	432	89
LUT	65 739	70 560	93
FF	49 866	141 120	35

表 3 中显示了将 FPGA 中 CNN 的性能与 Intel Core i5 CPU 和 NVIDIA UeForce UTX 960 UPU 进行比较的结果。基于 FPGA 优化设计的卷积神经网络处理单个图像所需的时间比 CPU 要少得多，相当于 GPU 的速度。GPU 功耗是本文设计的 30 倍以上。

表 3 不同硬件平台的性能评估

平台	CPU	GPU	FPGA
时间/ms	286.9	55.3	62.5
功耗/W	28	130	3.9

4 结论

本文提出了一种基于 FPGA 有限资源的卷积神经网络加速器。利用 BOOTH 算法实现乘法，有效降低了DSP 资源占用量。通过流水线结构和卷积运算的并行性提高了卷积运算的速度。网络加速器的内部结构在资源有限的开发板上实现 12 层 CNN 网络，并将其与 CPU 和 GPU 进行比较。实验结果表明，嵌入式 FPGA 的功耗和性能具有很大的优势，更适合于移动端的部署。

参考文献

[1] YU K, JIA L, CHFN Y, et al. Deep learning: yesterday, today, and tomorrow[J]. Journal of Computer Research and Development, 2013, 50(9): 1799-1804.

[2] FUKAGAI T, MAEDA K, TANABE S, et al. Speed-up of object detection neural network with GPU[C]. Proceedings of the 25th IEEE International Conference on Image Processing. LosAlamitos: IEEE Computer Society Press, 2018: 301-305.

[3] 吴艳霞，梁楷，刘颖，等. 深度学习 FPGA 加速器的进展与趋势[J]. 计算机学报，2019(11): 2461-2480.

[4] HOWARD A G, ZHU M, CHEN B, et al. Mobile net: efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv: 1704, 04861, 2017.

[5] Nielsen, Michael A. Neural networks and deep learning[M]. USA: Determination Press, 2015.

[6] 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. New York: ACM, 2016: 26-35.

[7] ZHANG J, LI J. Improving the performance of OpenCL-based FPGA accelerator for convolutional neural network[C]. Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York: ACM, 2017: 25-34.

[8] GSCHWEND D. Zynqnet: An fpga-accelerated embedded convolutional neural network[D]. Master ETH-Zurich: Swiss Federal Institute of Technology Zurich, 2016.

[9] ZHANG C, LI P, SUN G Y, et al. Optimizing FPGA-based accelerator design for deep convolutional neural networks[C]. Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York: ACM Press, 2015: 161-170.

[10] LIU S L, FAN H X, NIU X Y, et al. Optimizing CNN-based segmentation with deeply customized convolutional and deconvolutional architectures on FPGA[J]. ACM Transactions on Reconfigurable Technology and Systems, 2018, 11(3): Article No. 19.

[11] SHEN Y M, FERDMAN M, MILDER P. Escher: a CNN accelerator with flexible buffering to minimize off-Chip transfer[C]. Proceedings of the 25th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines. Los Alamitos: IEEE Computer Society Press, 2017: 93-100.

[12] GUO K Y, SUI L Z, QIU J T, et al. Angel-Eye: a complete design flow for mapping CNN onto embedded FPGA[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(1): 35-47.

（收稿日期：2020-08-21）

作者简介：

王婷(1995-)，女，硕士研究生，主要研究方向：集成电路设计。

陈斌岳(1991-)，男，硕士研究生，主要研究方向：集成电路设计。

张福海(1963-)，男，副教授，主要研究方向：集成电路设计。

版权声明