**密级： 保密期限：**

xm 拷贝

**硕士学位论文**



**题目：基于FPGA的Alexnet前向网络加速**

**学 号： 2015110001**

**姓 名： 邱宇**

**专 业： 信息与通信工程**

**导 师： 别红霞**

**学 院： 信息与通信工程学院**

**2017年 12 月 1 日**



**A Thesis for Master Degree**

**TITLE: TO ACCELERATE THE FORWARD NETWORKS OF ALEXNET BASED ON FPGA**

|  |  |
| --- | --- |
| **Student No.:** | **2015110001** |
| **Author:** | **Yu Qiu** |
| **Major:** | **Information and Communi-**  **cation Engineering** |
| **Supervisor:** | **Hongxia Bie** |
| **School:** | **School of Information and Communication Engineering** |

**Dec. 1st, 2017**

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：日期：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在年解密后适用本授权书。

非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名：日期：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

导师签名：日期：\_\_\_\_\_\_\_\_\_\_ \_\_\_

基于FPGA的Alexnet前向网络加速

**摘 要**

随着芯片制造工艺和协处理器的不断发展，FPGA芯片内部集成了越来越多的DSP以及片内存储资源，使得FPGA在计算密集型硬件加速方面具有巨大优势。作为典型的计算密集型应用，深度神经网络中的卷积神经网络在人脸识别、图像分割等领域都具有重要的指导意义和应用价值，一直备受学术界和工业界的青睐。然而，目前通用处理器实现卷积神经网络的方法，无法充分挖掘网络模型内部的并行性。在目前对实时性和低功耗的应用需求越来越旺盛的前提下，越来越多的研究人员开始采用FPGA对基于卷积神经网络的应用进行开发。

Alexnet作为卷积神经网络领域内具有重要历史意义的一个网络模型，不仅证明了卷积神经网络在复杂模型下的有效性，并使用GPU使大数据训练在可接受的时间范围内得到了结果。因此，研究Alexnet模型的计算加速对复杂模型下卷积神经网络加速的研究具有重要意义。

论文以此为研究背景，在充分研究和分析了当前研究成果的基础之上，设计实现了一种基于FPGA的Alexnet前向网络加速器，通过优化模型结构、流水式层间处理以及提高网络并行性来提高网络的整体识别速度。本文的主要研究工作包括：

1、对Alexnet前向识别网络性能的主要影响因素进行了研究并对网络模型中激活函数和池化模块进行了优化改进。论文首先对网络模型中存在的并行性计算进行了研究。其次，给出了前向计算过程的分析以及乘法计算量的分析。最后，分析激活函数和池化模块的计算量，并结合激活函数和池化模块比较计算输出最大值的特点以及FPGA的特性，对网络模型进行了优化。在保证输出结果不变的前提下，减少了397428次比较运算，占激活函数和池化模块优化前计算量的76.4%。

2、完成了基于FPGA的Alexnet前向网络关键模块的优化设计。本文首先完成了卷积神经网络内部基本单元的设计及其二维并行加速的设计，分析了二维并行加速的加速比。其次，根据每层输出的缓存结果的数据量大小，对芯片内部的块RAM资源进行合理的分配，从而确定了每层网络的并行度，完成了网络内每一层结构的设计。

3、完成了基于FPGA的Alexnet前向网络实现及性能分析。基于FPGA开发平台，完成Alexnet前向网络整体框架设计及实现，编写仿真文件进行验证，并完成对资源使用情况以及设计性能进行分析。给出了前向识别网络中每一层具体所需的计算时间，列出了加速器设计整体的资源使用情况，完成了与GPU、CPU识别速率的对比，并对识别结果的正确性进行了验证。

**关键词：**卷积神经网络 Alexnet 加速器 FPGA 并行计算

**TO ACCELERATE THE FORWARD NETWORKS OF ALEXNET BASED ON FPGA**

**ABSTRACT**

With the continuous development of chip manufacturing process and coprocessor, more and more DSPs and on-chip memory resources are integrated into FPGA chip, which makes FPGA have great advantages in computing intensive hardware acceleration. As a typical computing intensive application, convolutional neural networks in deep neural networks have important guiding significance and application value in the fields of face recognition, image segmentation, and so on, and have been favored by the academic and industrial circles. However, at present, the implementation of convolutional neural networks by general processors can not fully exploit the parallelism within the network model. With the increasing demand for real-time and low power applications, more and more researchers begin to use FPGA to develop applications based on convolutional neural networks.

As a network model with important historical significance in the field of convolutional neural networks, Alexnet not only proves the effectiveness of the convolutional neural network in complex model, but also uses GPU to make the training of large data within the acceptable time range. Therefore, it is of great significance to study the acceleration of Alexnet model for accelerating the convolutional neural network under complex model.

Based on this background, in the full research and Analysis on the basis of current research, design and implementation of a Alexnet forward network accelerator based on FPGA, by optimizing the model structure, flow layer processing and improve network parallelism to improve the overall recognition rate of network. The main research work of this thesis includes:

1. The main factors affecting the performance of Alexnet forward recognition network are studied, and the activation function and the pooling module in the network model are optimized and improved. The parallel computation in the network model is studied in this thesis. Secondly, the analysis of forward calculation process and multiplication calculation are given. Finally, the activation function and the calculation of the pool module are analyzed, and the network model is optimized by combining the activation function and the pooling module to calculate the maximum output value and the characteristics of FPGA. On the premise of ensuring the output unchanged, 397428 comparative operations are reduced, accounting for 76.4% of the calculation before the optimization of the activation function and the pooling module.

2. The optimization design of key modules of Alexnet forward network based on FPGA is completed. In this thesis, the design of the basic unit of the convolutional neural network and the two dimensional parallel acceleration are completed, and the speedup ratio of the two dimensional parallel acceleration is analyzed. Secondly, according to the amount of data output from each layer, the BRAM resources in the chip are allocated reasonably, thus the parallelism of each layer network is determined, and the design of each layer structure in the network is completed.

3. The implementation and performance analysis of Alexnet forward network based on FPGA are completed. Based on the FPGA development platform, the overall framework design and implementation of the Alexnet forward network are completed, and the simulation files are verified, and the resource usage and design performance are analyzed. The specific computing time of each layer in the forward recognition network is given, the resource usage of the accelerator design is listed, and the recognition rate of the GPU and CPU is compared, and the correctness of the recognition results is verified.

**KEY WORDS:** Convolutional Neural Network Alexnet Accelerator FPGA Parallel Computing

目录

[第一章 绪论 1](#_Toc508912430)

[1.1 课题背景 1](#_Toc508912431)

[1.2 国内外研究现状 2](#_Toc508912432)

[1.3 本文主要工作 4](#_Toc508912433)

[1.4 论文的组织结构 5](#_Toc508912434)

[第二章 相关技术研究 6](#_Toc508912435)

[2.1 卷积神经网络 6](#_Toc508912436)

[2.1.1 卷积神经网络前向传播网络结构 6](#_Toc508912437)

[2.1.2 Alexnet前向网络模型 11](#_Toc508912438)

[2.2 硬件平台介绍 12](#_Toc508912439)

[2.2.1 FPGA介绍 12](#_Toc508912440)

[2.2.2 集成开发环境简介 14](#_Toc508912441)

[2.2.3 开发流程 15](#_Toc508912442)

[2.3 FPGA实现Alexnet网络的优势 18](#_Toc508912443)

[2.4 本章小结 19](#_Toc508912444)

[第三章 影响Alexnet前向网络性能的研究及优化 20](#_Toc508912445)

[3.1 Alexnet前向网络计算特征分析 20](#_Toc508912446)

[3.1.1 并行性分析 20](#_Toc508912447)

[3.1.2 计算过程分析 22](#_Toc508912448)

[3.1.3 乘法运算量分析 26](#_Toc508912449)

[3.2 激活函数和池化模块优化 27](#_Toc508912450)

[3.2.1 激活函数 27](#_Toc508912451)

[3.2.2 池化模块 27](#_Toc508912452)

[3.2.3 优化分析 28](#_Toc508912453)

[3.3 本章小结 30](#_Toc508912454)

[第四章 基于FPGA的Alexnet前向网络关键模块优化设计 31](#_Toc508912455)

[4.1 加速器总体结构 31](#_Toc508912456)

[4.2 基本运算单元设计 32](#_Toc508912457)

[4.2.1 卷积核并行性设计 32](#_Toc508912458)

[4.2.2 池化并行性设计 34](#_Toc508912459)

[4.2.3 激活函数并行性设计 35](#_Toc508912460)

[4.2.4 全连接层并行性设计 35](#_Toc508912461)

[4.3 层结构设计 36](#_Toc508912462)

[4.4 本章小结 39](#_Toc508912463)

[第五章 基于FPGA的Alexnet前向网络实现及性能分析 40](#_Toc508912464)

[5.1 实现平台及开发环境 40](#_Toc508912465)

[5.2 整体框架设计 40](#_Toc508912466)

[5.3 性能与分析 41](#_Toc508912467)

[5.3.1 每层加速效果分析 41](#_Toc508912468)

[5.3.2 总体性能分析 43](#_Toc508912469)

[5.4 本章小结 46](#_Toc508912470)

[第六章 总结与展望 47](#_Toc508912471)

[6.1 工作总结 47](#_Toc508912472)

[6.2 展望 48](#_Toc508912473)

[参考文献 49](#_Toc508912474)

[致谢 52](#_Toc508912475)

[攻读学位期间取得的研究成果 53](#_Toc508912476)

1. 绪论
   1. 课题背景

近年来，深度学习在语音、自然语言处理、计算机视觉等领域取得非常显著的成果，使得机器学习研究迈上了一个新的台阶。作为当前最引人注目的技术热点之一，其不仅在相关的行业领域应用广泛，且具有很高的学术研究价值。因此，无论是在学术界还是工业界，深度学习都备受青睐[1]。

卷积神经网络（Convolutional Neutral Network，CNN），作为深度学习范畴内的一种模型结构，包含特征检测层和分类层[2]。其中，特征检测层采用类似于人脑的分层结构，可以从原始数据中直接提取出由具体到抽象、由低层到高层的特征。该结构通过自主学习训练时的输入数据，避免了显式提取输入数据特征参数的过程；此外，由于处于相同特征映射面上的节点具有一样的权重，因此可以并行的对模型结构进行学习，这也是CNN相对于全连接无法比拟的地方。其权值共享和局部感受野的特性，使得网络模型中需要训练的参数的数量大大减少，同时，对网络模型的复杂度也有一定的降低。特别是当网络模型的输入数据为包含多维输入向量的图像数据时，多维向量直接输入的特点，可以有效的减少提取图像特征和归类过程中数据重建的复杂度[3]。

这些优点使得卷积神经网络已经成为绝大多数计算机视觉系统的核心技术。无人驾驶3D感知与物体检测、Google街景中的人脸识别以及Facebook的图像自动标签都用到了卷积神经网络[4]。同时在语音识别、自然语言处理等领域中的应用也越来越广泛。

现阶段卷积神经网络的前向识别过程主要由通用处理器完成，考虑到网络内部大量并行计算的存在，通用处理器的实现方式计算效率不高，所以很难满足当前实时性和低功耗的性能需求。在新的芯片材料以及制造工艺等基础技术未能提高的前提下，一种有效的解决方案就是采用异构计算方式，即在通用处理器的基础上添加专用协处理器，来提升处理性能。现有的协处理器主要有专用集成电路（Application Specific Integrated Circuit，ASIC）、图形处理器（Graphics Processing Unit，GPU）和现场可编程门阵列(Field Programmable Gate Array，FPGA)。由于独特的处理架构，FPGA拥有其他处理器无法比拟的优势，可以通过软件重新配置芯片内部的资源形成不同功能硬件，因此，不仅有了软件设计灵活的特性，同时又有ASIC吞吐率高以及延时性低的特点。作为一种计算密集型加速部件，相对于GPU只能对数据做并行计算，FPGA可以很容易对数据进行并行计算以及流水式并行处理，且很容易在两者之间找到平衡。

由于设计需求中要求硬件平台为中低端FPGA，而现在其他论文中提到的模型实现大多是高端FPGA，在资源有限的前提下很难做到并行性的提高。相较于其他网络模型的数据量大模型复杂的特点，Alexnet很适合在中低端FPGA上实现。其次，Alexnet作为卷积神经网络领域内具有重要历史意义的一个网络模型，不仅证明了卷积神经网络在复杂模型下的有效性，并使用GPU使大数据训练在可接受的时间范围内得到了结果[5]。因此，研究Alexnet模型的计算加速对复杂模型下卷积神经网络加速的研究具有重要意义。本文在已有的相关理论和研究方法的基础之上，结合Alexnet模型内存在大量并行计算的特点以及FPGA对于密集计算的加速特性，提供了一种较为通用的基于FPGA的Alexnet前向识别网络加速设计方案，对于研究基于硬件实现的复杂卷积神经网络模型的识别加速，具有重要的参考价值。

* 1. 国内外研究现状

人工神经网络（Artificial Neural Network，ANN）又称类神经网络，是基于人体中神经系统的基本原理，在充分分析和抽象了人脑对外界激励的反应过程后，以模型拓扑结构为研究基础，效仿大脑神经处理系统对外界数据的处置机理的一种数学建模[6]。该模型可以并行的对数据进行处理，具有很强的自主学习能力以及容错性，并能够很好的协调数据的处理和缓存。由于其独特的特征表达以及完全自主的适应性，使得对人工神经网络的研究越来越广泛。它实际上是一个具有很好的非线性性的网络，网络模型中包含大量的相互连接的神经元节点，并能够进行复杂的数据处理和信息分类。

卷积神经网络是多层前馈网络中最具代表性的一种模型结构，能够自主的对大量有标签或者无标签的数据进行学习并从中提取数据特征[7]。卷积神经网络只需要对输入的图像数据进行少量的预操作就可以自动的提取出输入图像的特征，对包含多个特征的处理对象也有非常好的识别效果，并且图片的弯曲、平移或者轻微旋转对识别效果不会造成太大的影响[8]。这些优势使得卷积神经网络成为多层人工神经网络研究的一个重要方向，多年来一直是研究的热点，近些年，更是在国内外知名的高等院校、科研院所、互联网公司获得了大量的项目和资金支持，成为一个极其热门的研究课题[9]。

Wiesel和Hubel在1962年对猫大脑主要视觉皮层的研究促进了卷积神经网络思想的提出，该研究定义了简单细胞，其具有局部感知域内定向选择的功能，该功能与卷积神经网络中滤波器的功能类似；同时，还定义了与卷积神经网络中的聚集层功能相似的复杂细胞 [10]。随后一种基于生物视觉的人工神经网络模型Neocognitron被Fukushima提出，并完成了该类网络模型在计算机上的首次模拟。复杂细胞层和简单细胞层的串行构成了该类网络的基本架构，多个复杂细胞平面组成了一个复杂细胞层，其中，每个平面由一组神经元组成，其通过与前面复杂细胞层的卷积计算来提取特征[11]；简单细胞层是由多个简单细胞平面组成，其作用是通过对前面复杂层池化区域进行取最大值或取均值处理来提高网络模型对输入视觉信息形状、位置变化的容忍能力，同时，简单细胞层还具有降低后继层神经元节点数量的功能；最后一层是用一个已标记的神经元节点表示每个被识别的类的输出层。网络模型的训练采用的是无监督的竞争学习算法，输出层是经过单独有监督训练的线性分类器；在对网络模型进一步的简化后[12]，模型的训练采用反向传播的方式，该方式是在有监督的情况下进行的。

近年来，卷积神经网络获得了大量的实际应用。法国电信在视频会议以及其他系统中部署的人脸检测系统即为基于卷积神经网络的[13]；Google使用卷积神经网络进行街景图片中车牌和人脸识别，用来保护个人隐私[14]；微软使用卷积神经网络来进行手写识别和光学字符识别[15]，其中包括中文字符和阿拉伯文识别系统；基于计算机视觉的非公路机器人[16]采用了有监督的卷积神经网络的算法来躲避障碍物，而基于视觉导航的非公路机器人开发人员在DARPA的赞助下利用卷积神经网络来实现对长距离障碍物的躲避[17]；美国电话电报公司使用卷积神经网络对交易过程中存在的支票进行识别[18]；在文献[19]中，系统首先对网络模型进行离线训练，然后采用带标签样本的短距离立体视觉系统 对机器人的运行进行在线修正，其中，训练采用的是有监督学习和无监督学习相结合的方式；其他研究类型的网络模型应用有手写数字识别，人体手势识别以及标识和汉字识别，最近的应用还包括特征点检测，图像分割，尤其是生物医学领域中应用的图像的分割。

作为非常经典的密集计算型应用，利用软件来实现卷积神经网络的方式在其刚出现之时，就与并行计算紧密结合，充分利用了GPU非常适合乘法计算的特点，对于多层网络而言，只有在GPU上计算才能确保良好的性能指标。文献[20]中设计的用来人脸识别的多层网络模型的训练和前向识别过程就是在两块NVIDIA GTX 580 3G GPU上完成的。然而，该型号的GPU一块就有不低于200W的功率消耗，并且该类型的平台灵活性非常差，且体积相对较大。

早在20世纪90年代就有人开始研究基于硬件的卷积神经网络实现，当时贝尔实验室开发了人工神经网络芯片ANNA，该芯片采用模拟和数字相结合的结构，实现了64个8×8卷积并行计算的处理[21]。这时基于硬件完成的网络模型设计性能十分低下，同时数据精度也处于较低水准，其权值数据用6比特的定点数据量化，状态表示用4比特的定点数据量化[22]。Cloutier等人[23]在FPGA完成了针对图像处理的神经网络处理器的设计，采用二维的网络拓扑作为处理器的架构设计[24]，拓扑结构中的每一个节点作为一个处理单元，并通过与其所处位置周围的四个节点相互连接，构成一个处理器阵列，并由单指令多数据（Single Instruction Multi Data，SIMD）控制器控制。该设计模型的规模较小，因而可以处理的信息量相对较少，用于性能测试的输入图片的像素值只能用3比特的数据进行量化。Antony等人[25]完成了基于FPGA的多层感知器的实现，并详细分析了并行实现和串行实现以及浮点数格式表示和定点数格式表示对硬件实现的网络模型的性能的影响。但是，Antony等人没有给出具体的使用FPGA来实现网络模型的设计方案。Murugan等人在文献[26]给出了在大量使用FPGA芯片内部的数字信号处理单元以及DDR存储器资源的基础上实现卷积网络协处理器的设计方案，该设计方案对4个5×5的二维卷积运算并行处理。该方案在充分使用DDR带宽的基础之上，对较低位宽的数据进行组合，使得模型中的中间处理结果可以放入外部 DDR中。然而，这种设计方案需要对中间处理结果进行复杂的逻辑控制，因此，仍然需要PC机对逻辑模块进行控制，同时，可以将网络模型中的部分计算放入到PC机中。若要将该设计中提出的实现方案应用到低功耗、小尺寸平台或者其他智能移动终端，仍需要进一步的优化改进。文献[27]提出基于机器学习的定制化多芯片处理架构，设计了专门用于处理深度神经网络、卷积神经网络的专用芯片，该芯片将中间缓存数据存储到芯片内部，因此，对带宽的要求没有那么严格，且可以做到很高的数据处理并行度，能够达到很好的并行加速效果。然而由于其设计时功能模块相对固定，灵活性相对较低。文献[28] 采用多级流水线结构完成了一种基于FPGA的浅层卷积神经网络加速系统的设计，该设计对LeNet-5的模型结构进行了优化，根据网络模型内部层与层之间流水线式计算的特点设计完成，并得到了很好的加速效果。然而作者着重介绍了系统实现，忽略了功能单元设计模块化的原则，使得虽然在FPGA上完成了相应的识别功能，却使得系统设计的灵活性很差。就其他诸如数字信号处理芯片和嵌入式处理器等计算平台而言，由于器件内部架构属于串行结构，对于卷积神经网络的前向加速并不适用，因此，很少看到在类似平台上的计算加速。

* 1. 本文主要工作

本文旨在提供一种高实时性、低功耗、高性能的基于FPGA的Alexnet前向网络识别加速的硬件解决方案，其主要工作包括：

（1）、影响Alexnet前向网络性能的研究及优化

卷积神经网络内部存在的大量的并行计算是基于FPGA实现整个加速框架的基础和核心。因此，论文首先分析了影响Alexnet前向网络性能的主要因素，研究网络模型中存在的并行性计算，并对计算量进行分析。其次，研究了激活函数和池化函数的特点，结合FPGA的特性对网络模型进行优化。

（2）、基于FPGA的Alexnet前向网络关键模块优化设计

在FPGA芯片内部资源有限的前提下，如何选择满足前向网络加速器性能设计要求的并行性结构是当前研究卷积神经网络并行结构的关键问题。因此，本文结合FPGA的特性以及资源情况，分别对Alexnet前向网络中关键的模块及运算单元进行设计和优化，其中包括：卷积核的并行性优化设计、激活函数及池化并行性优化设计、全连接层并行性优化设计。

（3）、基于FPGA的Alexnet前向网络实现及性能分析

基于FPGA开发平台，完成Alexnet前向网络整体框架设计及实现，编写仿真文件进行验证，并对资源使用情况以及设计性能进行分析。

* 1. 论文的组织结构

本文首先介绍Alexnet网络模型基于FPGA进行加速的研究背景及其发展现状，提出本文基于FPGA的Alexnet前向网络加速的研究目标并给出了论文的组织架构。

第二章，首先介绍了Alexnet前向网络的整体框架，然后对网络拓扑结构中关键的处理函数进行了分析，最后对硬件开发平台进行了介绍，包括FPGA介绍、集成开发环境介绍以及开发流程介绍等。

第三章，对Alexnet前向网络中存在的并行计算进行研究，并对网络模型的计算量进行了分析，并根据每层计算量的大小对硬件资源进行分配。同时，根据激活函数和池化函数的特点，对网络模型结构进行了优化，并对优化前后计算量的大小进行了分析。

第四章，详细介绍了基于FPGA的Alexnet前向网络关键模块优化设计，包括：卷积核并行性设计、激活函数及池化并行性设计以及全连接层并行性设计。

第五章，总体介绍基于FPGA的Alexnet前向网络实现的整体框图，对基于FPGA的网络具体实现进行仿真验证，并对仿真结果及加速器的整体性能进行分析。

第六章，对论文所做工作进行总结，分析论文研究工作中存在的不足，并对未来的研究工作进行了展望。

1. 相关技术研究
   1. 卷积神经网络

卷积神经网络（Convolutional Neutral Network，CNN）作为深层神经网络模型的一种，已成为当前图像分类和计算机视觉处理领域的热门研究内容。其权值数据共享的网络结构特点，有效降低了网络模型中的权值数据量，同时进一步的优化了网络模型的复杂性。另外，其模拟人脑的数据提取过程，使之可以对训练数据的特征参数进行自动提取。这些优点使得卷积神经网络在处理多维图像数据时更具优势。将图像的多维向量数据直接作为模型的输入时，避免了传统分类算法中复杂的数据重建和特征提取[29]过程。

* + 1. 卷积神经网络前向传播网络结构

卷积神经网络是一种前馈神经网络，包含多个隐藏层，如图2-1所示，每层由多个独立地神经元节点组成的二维平面组成。



图2-1 一般卷积神经网络结构

首先对输入的图像数据进行卷积处理，并对卷积后的数据加偏置，得到特征映射图谱，如C1层所示。然后对特征映射图中的像素值进行池化操作，其中池化可分为最大池化和均值池化。对池化的数据再进行加权值，加偏置，并通过一个激活函数得到S1层所示的特征图谱。这些特征图谱再经过卷积运算得到C2层。这个层级的特征图谱再和S1一样产生S2。另外，如果网络模型用于分类，一般会在网络的末端加上全连接层用于网络输出。

卷积神经网络的前向识别过程主要有四种操作：卷积、下采样、光栅化、多层感知器预测，分别对应着卷积层、下采样层、激活函数和全连接层[30]。下面分别对每一层进行介绍。

* + - 1. 卷积层

卷积层（Convolution layers）的作用主要在于学习输入的特征表示，它由一定数目的特征图谱（feature maps）构成[31]。特征图谱上的每一个节点与上一层的邻近节点相连，这些相邻的节点构成的区域就组成了该节点在上一层的局部感知野[32]。如图2-2所示，为了得到下一层的特征图谱，输入特征图谱首先与训练好的权值数据做卷积运算，这些权值数据组成的矩阵又称为滤波器或特征预测器，然后将得到的特征图谱传输至激活函数。通过应用不同的特征检测器得到新的特征图谱。卷积运算三维卷积公式定义为：

 (2-1)

其中，是第m帧在位置处输出的特征图谱的值；是第n帧输入图像在位置处的值；是连接第n帧输入图像和第m帧输出图像的卷积核在位置处的权重；是在第m帧输出图像处添加的偏置。



图2-2 卷积运算

* + - 1. 激活函数

为解决卷积神经网络中线性模型表达能力不足的问题，会在卷积操作后加入激活函数，用来增强模型的非线性因素，使得神经网络对复杂问题的处理能力进一步提升[33]。常用的激活函数的形式有：Sigmoid函数、tanh函数、ReLU函数等。

1）、Sigmoid函数

Sigmoid非线性函数[34]对输入的数值在（0,1）之间映射，它的数学公式为：

 （2-2）

其图形如图2-3所示：



图2-3 Sigmoid函数

Sigmoid函数曾被广泛地应用，因为它的输出在(0,1)之间，且具有单调连续的性质，输出范围有限，能够很好的表达“激活”的意思，未激活就是0，完全饱和的激活则是1。另外对卷积后的特征图谱优化稳定，可以用作输出层。然而现在却很少使用Sigmoid函数，主要是因为以下两个缺点：

1、函数容易饱和。当输入非常大或者非常小的时候，神经元的梯度就接近于0，导致最终权重基本没有更新，无法递归地对输入数据进行学习。

2、函数输出的平均值不为零。该特性会传输至下一层网络，以至于下一层的输入平均值也不为零，从而会对梯度下降的运作产生影响。

2）、tanh函数

tanh 函数实际上是一个放大的 sigmoid 函数[35]，数学公式为：

 （2-3）

函数的取值范围在 [-1, 1]区间上，如图2-4所示为tanh 函数的图像：



图2-4 tanh函数

tanh函数同样存在着饱和问题，但它输出值的均值大小为零，且相对于Sigmoid函数具有更快的收敛速度，因此网络模型中 tanh函数比 sigmoid函数使用更加广泛。

3）、ReLU函数

ReLU函数近些年应用最为普遍的激活函数，它的数学公式为：

 （2-4）

对应的图像如图2-5所示：



图2-5 ReLU函数

ReLU函数的主要优点有：

1.相对于Sigmoid函数和tanh函数，ReLU函数[36]在随机梯度下降中能够快速收敛。

2.Sigmoid函数和tanh函数中包含了很多复杂的计算，比如指数运算，ReLU函数则更易于在硬件上进行实现。

3.有效缓解了梯度消失的问题。

4.在无监督学习时有更好的学习能力。

5.使卷积神经网络的稀疏表达能力进一步增强。

但是，随着训练的进行，ReLU激活函数可能会导致网络模型中出现神经元死亡，权重无法更新的情况。一旦权值数据无法更新，流经节点的梯度从该点开始讲始终为零。换句话说，即ReLU节点在学习过程中不可逆地死亡了。

* + - 1. 池化层

在卷积神经网络中，连续卷积层之间往往会插入池化层。池化操作的主要作用是为卷积神经网络提供了很强的鲁棒性，并且减少了参数的数量，防止过拟合现象的发生。池化操作在每个深度切片上进行，池化层的参数一般有三个，分别为：

kernel\_size：表示池化区域的大小；

pad: 和卷积层的pad的一样，进行边沿扩充，一般大小为0；

Stride：步长，即池化窗口每次上下或前后移动的距离，一般与kernel\_size的大小一致，该类池化为不重叠方式。移动距离也可以小于kernel\_size的大小，即为重叠池化，在Alexnet网络模型中使用的即为重叠池化。

池化操作通常由最大值池化和平均值池化两种不同的方式。

最大池化操作即为选择区域内所有神经元的最大值（max-pooling）作为输出。以下图2-6为例，输入数据为4×4，采样核size为2，stride为2。输入数据大小类似卷积层的计算方法，（input\_width+2×pad- kernel\_size）/stride+1。在训练过程中不仅需要求得pool窗口内的最大值，还需要记录该最大值在输入数据中对应的位置，目的是为了在反向更新时，需要把梯度值传输至最大值对应的位置。



图2-6 最大池化

均值池化即对池化窗口内所有神经元节点求和取均值，并以此作为该区域池化后的值。以下图2-7为例，输入数据为4×4，采样核size为2，stride为2。首先对每一个2×2的区域元素求和，再除以4，得到主要特征输出。



图2-7 均值池化

* + - 1. 全连接层

如图2-8所示，全连接层（fully connected layers，FC）的每一个结点都与上一层的所有结点相连，用来把前边提取到的特征综合起来。由于其节点全部相互连接的特点，使得全连接层的权值数据一般也是最多的。在整个卷积神经网络中，全连接层起到“分类器”的作用。



图2-8 全连接层结构

* + 1. Alexnet前向网络模型

作为卷积神经网络在计算机视觉应用中的经典模型，Alexnet模型也越来越引起人们的关注。模型包含八层网络[37]，其网络结构如图2-9所示：

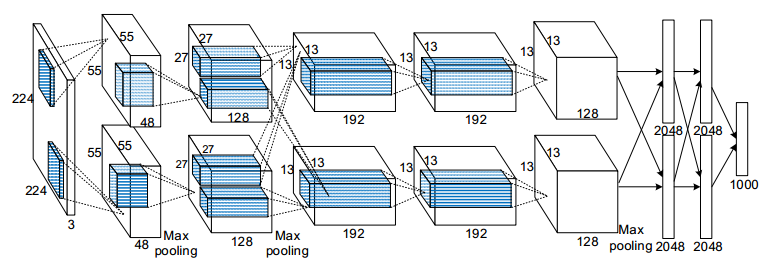


图2-9 Alexnet网络结构

其中前五层为卷积层，后面三层为全连接层，网络模型的的输出为一个包含1000个分类标签的分类器。

第二层的卷积核只连接到前一层并处于同一块GPU中的那些卷积核映射上，第四和第五层的连接方式与第二层相似。第三层的卷积核被连接到第二层中的所有卷积核映射上。全连接层中的节点与前一层中的所有节点连接。网络第一层处理结束后会跟有局部响应归一化层，第二层类似。在局部响应归一化层以及第五层卷积核之后会跟有最大池化层。ReLU激活函数在网络模型的全部八层中都有应用。

第一个卷积层利用96个大小为11×11×3、步长为4个像素的核，来对大小为224×224×3的输入图像进行滤波。第一个卷积层的输出将直接作为第二个卷积层的输入，在第二个卷积核中，特征检测器中会有256个大小为5×5×48的核。第三层、第四层以及第五层之间没有池化层与局部响应归一化层，它们彼此互相连接。第三层中共有384个大小为3×3×256的卷积核与第二层的输出结果相连。第四层拥有384个大小为3×3×192的卷积核，第五层拥有256个大小为3×3×192的卷积核。全连接层都各有4096个神经元。

* 1. 硬件平台介绍

本文选择了Xilinx公司的Kintex-7 FPGA开发板作为加速器设计的硬件平台，核心芯片的具体型号是XC72K325T，它内部包含18070个触发器，203800个查找表，840个乘法器资源，445个36kb的Block RAM资源。下面对FPGA的基本组成单元、集成开发环境以及FPGA设计流程进行介绍。

* + 1. FPGA介绍

现场可编程门阵列（Field－Programmable Gate Array，FPGA）是在可编程阵列逻辑、通用阵列逻辑、可编程逻辑器件等可编程器件的基础上进一步发展起来的。作为一种半定制电路，它既弥补了专用集成电路无法现场修改的不足，又克服了通用阵列逻辑门电路数量受限的缺陷。

FPGA芯片主要由可配置逻辑块CLB（Configurable Logic Block）、完整的时钟管理单元、内嵌的底层功能单元、嵌入式块RAM、可编程输入输出单元IOB（Input Output Block）、丰富的布线资源六部分组成。每部分具体介绍如下：

⑴、可编程输入输出单元（IOB）

外部电路信号与芯片内部信号交互的接口即为可编程输入输出单元，它的主要作用是完成对输入、输出信号在不同电气特性下的匹配和驱动。FPGA内部按组对基本输入输出单元进行分类，每组对不同I/O口的电平特性和物理特性的支持是独立地并可以通过软件编程完成不同特性的灵活配置。同时，可以通过改变上、下拉电阻的阻值大小调整驱动电流的大小。目前，对可编程输入输出单元的收发频率的要求越来越高。通过DDR寄存器技术，一些高端FPGA的数据速率可以达到2Gbps。

通过可编程输入输出单元的存储模块可以将外部信号输入到芯片内部，也可以直接将外部信号输入到芯片内部。当通过可编程输入输出单元的存储模块对外部信号输入时，可以降低对其保持时间（Hold Time）的要求，通常情况下默认为0。

为了适应不同的电气标准以及便于I/O口的管理，FPGA的可编程输入输出单元被划分为若干个组（bank），其接口电压的大小决定每个组的接口标准。其中，同一个组内只能有一种电气标准，但不同组的电气标准可以不同。电压相同是接口标准的基本条件，只有电气标准相同的端口才能连接在一起。

⑵、可配置逻辑块（CLB）

作为FPGA内的基本逻辑单元，可配置逻辑块是由寄存器(Register) 和查找表(LUT)组成的，其中，组合逻辑功能由查找表完成，同步时序逻辑功能的设计则需要依赖寄存器来完成。可以将FPGA内部的寄存器配置成锁存器，也可以配置为带置位、时钟使能以及异步/同步复位的触发器。

Xilinx公司定义了Slice作为芯片内部的基本逻辑单位。其中，两个4输入的函数、函数复用器、存储逻辑、进位逻辑和算术逻辑组成一个Slice。函数复用器（MUXC）以及专用进位信号组成了一个进位逻辑，用于快速的算术加减法操作的实现，每个进位逻辑有两条快速进位链，可以提高可配置逻辑块的处理速度；算术逻辑则由一个专用与门（MULTAND）和一个异或门（XORG），专用与门可以提升乘法器的效率，一个异或门可以使一个Slice实现 2比特数据的全加操作； 4输入的函数发生器可以用来实现分布式RAM、4输入查找表或16比特的移位寄存器。

⑶、数字时钟管理模块（DCM）

数字时钟管理模块（digital clock manager）是FPGA芯片内部专门用于消除时钟偏斜、进行时钟相位调整和时钟综合的固件资源，利用数字时钟管理模块可以很方便的完成对时钟信号的分频、相移、倍频，提高了系统时钟的可靠性，同时也给系统时钟的设计带来了方便。

⑷、嵌入式块RAM（BRAM）

FPGA内部的BRAM资源使得芯片的灵活性以及应用范围得到了很大的拓展。常用的存储器结构，如双端口RAM、FIFO、单端口RAM以及内容地址存储器 （CAM）都可以由块RAM配置而成。除了块RAM，FIFO、ROM和RAM等结构还可以通过 FPGA中的查找表灵活地配置而来。在实际选型过程中，芯片内部嵌入式块RAM的数量也是一个重要选择指标。

单片嵌入式块RAM的大小为18kb，即深度为1024、位宽为18bit，可以依据具体的设计需要对其深度和位宽进行改变，但是要在满足两个原则的基础之上：首先，修改后块RAM的大小不能超过18kb；其次，位宽的大小不能超过36bit。此外，可以通过多片块RAM组合的方式形成更大的块RAM，此时上述两条原则将不再受用，其深度和位宽的大小只受限于FPGA内块RAM的数量。

⑸、丰富的布线资源

FPGA芯片内部的所有单元由布线资源连通，而连线的工艺和线路的布局决定着信号的传输速率和驱动能力。FPGA内部的布局布线资源非常丰富，依据不同的分布位置、工艺、宽度和长度可以将其划分为四种不同的类别。第一类为用于FPGA内部全局复位/置位和全局时钟布线的全局布线资源；第二类为用于复位、专有时钟等控制信号线的分布式布线资源；第三类为用于实现基本逻辑单元之间的逻辑互连和布线的短线资源；第四类为用以实现FPGA内部不同Bank间的高速信号布线的长线资源。

⑹、内嵌的底层功能单元

DSP、CPU、延迟锁相环（Delay Locked Loop，DLL）和锁相环回路（Phase Locked Loop，PLL）等软处理核构成了FPGA内部主要的内嵌底层功能单元。由于芯片内部内嵌底层功能单元越来越丰富，使得单个FPGA芯片可以进行系统级的设计，并且有能力进行软硬件的联合设计，FPGA逐步向SOC平台过渡。

* + 1. 集成开发环境简介

目前Xilinx公司比较流行的基于可编程器件的电子设计自动化软件工具主要有ISE(Integrated System Configuration)和Vivado。2012年，Xilinx公司发布了Vivado这一款集成设计开发环境。该设计套件在通用调试环境以及共享可扩展数据模型的基础上，高度集成了从系统级开发到IC级开发的工具以及设计环境。该集成开发环境基于AMBA AXI4 互联规范，包含了很多有助于根据用户需求量身定制的设计流程，包含Synopsys 系统约束(SDC)、工具命令语言(TCL)以及IP-XACT IP封装元数据等，这些开发环境均符合当前开放式环境的业界标准。Vivado这一集成开发环境有效的结合了各类可编程技术，可以设计的等效专用集成电路门数高达1亿个。

ISE作为Xilinx公司中一款不断发展和演进的领域专用集成开发环境，可以为系统级设计、嵌入式开发、数字信号处理以及逻辑开发提供工具配置和互操作性的设计流程。作为一款专用设计套件，ISE具有非常强大功能，可以完成设计输入、功能仿真、布局布线以及综合实现等FPGA设计开发的全部流程。在设计流程中的各个阶段，用户都可以使用命令行界面、图形用户界面以及EDA工具界面进行设计开发。用户也可以根据自己的开发习惯灵活的选择只使用以上界面的一种，也可以针对处于设计流程的阶段选择任意一种界面进行开发。

由于现阶段ISE的应用更为普遍且满足整体项目需求，本文中的硬件设计部分使用ISE设计套件进行实现，后续的系统优化将在Vivado设计套件中完成。

* + 1. 开发流程

状态图、原理图以及HDL（Hardware description language，硬件描述语言）是当前主流的三种FPGA设计输入方式。由于三种设计输入方式具有不同的特点，因此可以针对不同的场合选择合适的设计输入方式。如表2-1所示为三种不同设计输入方式特性的比照表。

表2-1 设计输入方式特性比照表

|  |  |  |
| --- | --- | --- |
| 设计方式 | 性能特点 | 应用场合 |
| 硬件描述语言 | 易于实现代码的移植以及模块的重用，与开发环境基本无关，具有较高的设计效率 | 控制逻辑、复杂状态机 |
| 原理图 | 具有清晰的电路结构，设计效率对综合工具性能的依赖性低 | 顶层设计、需手工优化的逻辑 |
| 状态图 | 简化状态机的设计，对其工作方式一目了然 | 简单状态机 |

由于硬件描述语言设计法具有通用性、重用性与模块划分以及更好的可移植性的特点，使其成为当前主流的设计输入方式。下面对FPGA设计熟悉电路时的开发流程是基于硬件描述语言的，主要包括系统功能设计、寄存器传输级硬件描述语言设计、寄存器传输级仿真、综合、门级仿真、布局布线、时序仿真和FPGA板级调试等几个步骤。如图2-10所示。



图2-10 FPGA开发流程

1）系统功能设计

FPGA芯片选择、系统设计以及总体方案论证是要在系统设计之前完成的准备工作。系统设计师需要依据系统的复杂度以及设计指标等具体的设计需求，对芯片内部的各种资源情况、芯片成本以及工作速度等方面进行权衡，选择合适的器件类型以及合理的设计方案。系统一般采用自顶向下的方式进行设计，首先将系统划分为多个基本设计单元，然后对每个基本单元进行更深层次的划分，并一直划分下去直到可以直接使用元件库为止。

2）RTL级HDL设计

寄存器传输级（Register Transfer Level，RTL）是通过描述寄存器之间数据的流向、数据的处理以及数据控制的硬件描述语言设计方法，不关注寄存器和使用逻辑门的数目以及逻辑门之间的拓扑结构等具体的组合逻辑的细节。寄存器传输级设计相对而言更加的简单和高效，但是相对于门级设计而言更加抽象。其最大特点是门级网表可以由综合工具直接综合形成，寄存器传输级设计对系统的功能和效率起决定性的作用。

3）RTL级仿真

寄存器传输级仿真又称综合前仿真，或称为功能（行为）仿真，是对设计输入的逻辑功能的初步检验，此时是在加入延迟信息之前的功能验证。仿真前，需要先建立所关心的输入信号组合成序列的测试向量或波形文件，建立的方式可以分为硬件描述语言和波形编辑器。最后将会以输出波形信号或者生成报告文件的形式给出仿真结果，以便更好的观察各个信号的状态变化。如果发现状态变化与设计需求不符，则返回设计输入对逻辑设计进行修改。常用的仿真工具有ISE自带的Isim、新思科技公司的VCS以及ModelTech公司的ModelSim等软件。虽然在FPGA的设计流程中功能仿真不是必需步骤，但却是整体功能设计中最为关键的一步。

4）综合

将抽象的、较高层次的描述转换为较低层次描述的过程即为综合。对综合过程中生成的逻辑连接进行优化即为综合优化，优化的目的是使系统设计的层次更加平面化，方便实现FPGA内部的布局布线。就当前的层次来看，综合优化（Synthesis）并非是将设计输入综合成真实的门级电路，而是编译成逻辑连接网表，该逻辑网表由触发器、RAM、或门、与门、非门等基本逻辑单元组成。门级电路的具体实现需要根据综合后形成的逻辑连接网表来产生，该部分将由布局布线功能来完成。为了能够生成符合设计要求的门级结构网表，硬件描述语言的设计输入必须满足特定综合器的风格要求。由于对于门级结构、寄存器传输级的硬件描述语言综合的技术已经很成熟，所有厂家的综合工具都已经做到对这一级别综合的支持。经常使用到的综合开发工具有各个FPGA厂家推出的综合工具以及Synplicity公司的Synplify/SynplifyPro软件。

5）门级仿真

门级仿真又称综合后仿真，主要用于检查综合结果与设计需求是否一致。在进行门级仿真时，需要在综合仿真模型中加入综合过程中生成的标准延时文件，以此来估计门延时对整个系统设计带来的影响。由于在仿真过程中没有对线延时进行估计，因此仿真结果不是十分精确，与布线后的真实情况仍然存在一定的差距。就当前相对成熟的综合工具而言，一般的设计过程可以省略这部分仿真，但如果发现经过布局布线后生成的电路结构与设计目标不相符时，则需要返回该部分以进一步确认具体的问题点。一般而言，功能仿真中使用的仿真工具均对综合后仿真提供支持。

6）布局布线

将综合后生成的连接网表映射到具体的FPGA芯片中，并将系统设计的时序和逻辑与芯片内部的可用资源相匹配的过程称为布局布线，它是系统设计中最重要的步骤。其中，合理地将连接网表中的底层单元和硬件原语配置到FPGA内部的可用资源上的过程称为布局，在布局的过程中往往需要在速度和面积之间进行互换，以达到设计需求。布线需要建立在布局过程中生成的拓扑结构的基础之上，合理的运用FPGA内部的不同种类的连线资源，完成各个原件之间的正确连接。布局布线也可以简单的理解为合理配置芯片内部寄存器和查找表资源的过程。布局是挑选最合适的资源使其组合实现逻辑网表，而布线就是以最好的方式将寄存器和查找表资源连接起来。

由于FPGA的内部结构十分复杂，特别是在对设计输入进行时序约束时，完成系统设计的布局布线需要借助时序驱动的引擎。在完成系统布线后，开发工具会自动生成设计报告，对系统设计中资源使用情况进行详细说明。由于不同FPGA芯片厂家生产的芯片的内部结构互不相同，因此，对系统设计进行布局布线时需要选择相应开发商提供的布局布线工具。

7）时序仿真

时序仿真是指在逻辑网表中加入延时信息来检测系统设计中是否存在时序违规的现象，即检查是否满足时序约束条件，是否满足芯片内部触发器的建立和保持时间。时序仿真能够真实的反映FPGA内部的实际工作情况。由于不同芯片触发器的建立和保持时间不同，不同的布局布线选择造成的内部时延也不尽相同，因此在完成系统的布局布线后，需要检查和消除设计中存在的竞争冒险现象。同时，需要对各个模块以及整体设计进行时序仿真，用来评估系统设计的整体性能以及和信号之间的时序关系。

8）FPGA板级调试

通过烧写器将系统设计的配置文件下载到芯片中，对其资源进行分配映射的过程即为FPGA板级调试。其中，配置文件的格式一般可分为.sof或.pof两种不同的格式，下载的方式包括边界扫描、主动、被动等方式。

FPGA开发的主要调试工具是逻辑分析仪（Logic Analyzer，LA），但是使用这种器件进行测试需要对大量的测试信号引出，且逻辑分析仪的价格非常昂贵。为了解决上述问题，大部分的FPGA生产厂家在自己的集成开发环境中集成了在线逻辑分析仪，如Xilinx公司在ISE中嵌入了ChipScope。由于其占用的逻辑资源的数目有限，且具有很好的调试效果，因此在实际应用中具有很高的价值。

* 1. FPGA实现Alexnet网络的优势

FPGA对于实现Alexnet前向网络加速的优势[38]主要在于其高度并行、流水线处理、低功耗、低时延以及架构灵活的特点。

（1）高度并行主要是指对板上资源的复用，使得多个功能单元之间可以相互独立并行的完成计算任务。这一点与现在的多核和SIMD技术相似。但是与SIMD技术相比，FPGA的并行计算可以同时完成多个不同的任务，而不仅仅只是完成相同功能的计算。以简单的逻辑计算为例，SIMD技术使得处理单元可以并行的完成多个加法的操作，而FPGA可以在计算多个加法操作的同时并行的计算多个乘法运算。

（2）流水线处理是将单个任务拆分为多个不同的阶段，每个阶段相互独立且并行执行。其实这一点和CPU相似，只是CPU是指令间的流水而FPGA是任务间流水或者我们可以说是线程间流水。

（3）FPGA芯片内部包含大量的存储资源，可以降低多外部存储器带宽的要求，提高网络模型在芯片上的计算效率。

（4）FPGA架构灵活，可以根据需求的调整改变其内部的逻辑，减少开发成本。

* 1. 本章小结

本章首先对卷积神经网络的概念以及其局部感受野和权值共享特点的优势进行了介绍，对一般的网络模型结构进行了分析，并详细介绍了卷积神经网络前向识别过程中主要的四种操作：卷积、下采样、光栅化、多层感知器分类及其分别对应着的卷积层、下采样层、激活函数和全连接层，重点分析了卷积神经网络中的经典模型——Alexnet网络结构。同时，对硬件加速的核心器件FPGA进行了介绍，包括FPGA芯片组成、集成开发环境以及整体开发流程。最后，对FPGA实现Alexnet前向识别网络的优势进行了分析。

1. 影响Alexnet前向网络性能的研究及优化

本章对影响Alexnet前向识别网络性能的因素进行了研究，并针对在FPGA上的实现对网络模型进行了优化。首先在3.1节分析了Alexnet前向网络中主要的计算特征，包括并行性特征、计算过程以及卷积模块中存在的乘法计算量；其次，在3.2节对模型中使用的激活函数和池化模块进行了研究，并结合FPGA的计算特性，对网络模型中的计算量进行了优化。

* 1. Alexnet前向网络计算特征分析

Alexnet前向识别网络计算特征是基于FPGA实现整个加速框架的基础和核心，因此，本节重点对网络内部的计算特征进行了分析，包括网络模型内并行性计算的分析、计算过程的分析以及卷积模块乘法计算量的分析。

* + 1. 并行性分析

在对上一章卷积神经网络以及Alexnet前向识别网络模型分析的基础之上，本节将对Alexnet前向识别网络中涉及的并行性特征进行研究和分类。

Alexnet前向识别网络中的并行性特征主要包括神经元节点内的并行、神经元结点间的并行、卷积核并行以及层间并行等多种类型。一般来说，除非卷积神经网络的模型规模很小，否则在硬件上实现完全并行卷积神经网络模型是不可行的，因此，基于硬件实现的卷积神经网络在一定程度上都会包含串行计算，下面将对Alexnet前向识别网络中不同类型的并行性进行详细的分析：

（1）神经元节点内的并行

神经元节点内的并行是指计算单个神经元节点时，神经元节点内部的操作并行执行。

如图3-1所示为一个典型的神经元节点内的并行计算单元结构，计算输出神经元。在计算过程中，可以利用FPGA内部丰富的乘法器资源对乘法进行并行计算，然后利用流水加操作实现累加运算，最后把求得输出神经元的值。



图3-1 神经元节点内的并行

（2）神经元结点间的并行

神经元节点间的并行是指计算单个神经元结点的输出值时，多个神经元节点之间并行计算。

如图3-2所示为一个典型的神经元节点间的并行计算单元结构，在利用FPGA内部的乘累加单元计算单个神经元节点的输出值的同时，多个神经元节点之间并行操作。



图3-2 神经元结点间的并行

神经元结点间的并行性是基于FPGA实现Alexnet前向网络识别加速中最重要的并行性，如果神经元结点间的并行性被充分利用，更高层次的并行性自然被发现。由于Alexnet前向网络中神经元节点的数目众多且FPGA内部资源有限，使得神经元结点间的并行性并不能被完全的开发，但是单个神经元节点内的并行性计算很适合FPGA去实现，因为FPGA内部有大量的乘累加资源，其运算方式与神经元节点的计算方式十分匹配。

（3）卷积核并行

卷积核并行是指Alexnet前向识别网络内部多个卷积核之间并行计算。由于网络模型中有大量的卷积核，且不同卷积核之间的计算结果互不影响，因此，在计算单个卷积核的同时，可以对其他卷积核进行计算，以此提高整个网络模型的并行度。与神经元节点间的并行性计算相似，由于FPGA内部资源的限制，卷积核之间完全的并行计算并不能实现，因此，卷积核之间的并行性设计也是整个网络模型加速的研究重点。

（4）层间并行

在多层神经网络中，不同层之间可以通过建立流水线的方式进行并行计算，由于Alexnet网络模型前向传播的数据传递特点，通过数据控制单元控制图片信息以及权值数据的输入。当该层计算完毕之后，发出使能信号，控制下一帧图片的输入，以此来达到层间并行计算。

* + 1. 计算过程分析

Alexnet前向识别网络共有八层，前五层为卷积层，后三层为全连接层，最后一个全连接层输出一个具有1000类的网络识别结果。

网络模型的输入为227×227×3的RGB三维图像，该输入与96个11×11×3的特征检测器进行卷积计算，特征检测器与输入图像的卷积生成新的特征图谱。特征检测器沿输入图像的x轴和y轴进行移动，每次移动的长度为4个像素点。因此，在特征检测器沿一个方向移动卷积时会产生(227-11)/4+1=55个像素点（输入图像每行的像素点数减去特征检测器每行的像素点数再除以步长，即生成54个像素点，再加上特征检测器一行的像素点数，即为最后输出的每行像素点的个数），每列像素点个数的计算过程与行相似，因此最终生成的每幅特征图谱的大小为55×55。由于特征检测器的个数为96，因此最终会生成的像素层为55×55×96。由于该层为两个GPU训练得到，因此得到的96个特征图谱会分为两部分，每一部分包含48个特征图谱。这两部分的特征图谱会经过激活函数的处理，生成激活后的特征图谱，大小仍为两组55×55×48。

激活后的特征图谱会再经过池化模块的处理，池化窗口的大小为3×3，窗口每次的移动距离为2，则经过池化处理后得到的特征图谱每行有(55-3)/2+1=27个像素点。因此，经过第一层特征检测器的处理后特征图谱的大小为27×27×96。



图3-3 conv1数据流图

网络模型第一层输出的27×27×96的特征图谱会作为第二层的数据输入，为了简化后面的计算处理，需要在每一层特征图谱的上下左右分别做包零处理，即为添加两个像素值。96个27×27的特征图谱会被分成两组，每一组有48个。每组的特征图谱分别与5×5×48的特征检测器进行卷积计算，经过与特征检测器的卷积计算后会生成新的特征图谱。特征检测器沿输入图像的x轴和y轴进行移动，每次移动的长度为1个像素点。因此，在特征检测器沿一个方向移动卷积时会产生(27-5+2×2)/1+1=27个像素点，每列像素点个数的计算过程与行相似，因此最终生成的每幅特征图谱的大小为27×27。由于特征检测器的个数为256个，因此新生成的特征图谱的个数也为256个。这些特征图谱会经过relu2激活函数的处理，生成激活后的特征图谱，大小仍为两组27×27×128。

激活后的特征图谱会再经过池化模块的处理，池化窗口的大小仍为3×3，窗口每次的移动距离为2，则经过池化处理后得到的特征图谱每行有(27-3)/2+1=13个像素点。因此，经过第二层特征检测器的处理后得到两组大小均为13×13×128的特征图谱。



图3-4 conv2数据流图

网络模型第二层输出的2组13×13×128的特征图谱会作为第三层的数据输入，为了简化后面的计算处理，需要在每一层特征图谱的上下左右分别添加1个像素值。两组特征图谱同时与特征检测器进行卷积计算，每组均有192个特征检测器，每个检测器的大小为3×3×192，经过与特征检测器的卷积计算后会生成新的特征图谱。特征检测器沿输入图像的x轴和y轴进行移动，每次移动的长度为1个像素点。因此，在特征检测器沿一个方向移动卷积时会产生(13-3+1×2)/1+1=13个像素点，每列像素点个数的计算过程与行相似，因此最终生成的每幅特征图谱的大小为13×13。特征检测器的个数为384个，最后输出的特征图谱的大小为13×13×384。



图3-5 conv3数据流图

第三层输出的2组13×13×192的特征图谱会作为第四层的数据输入，为了简化后面的计算处理，需要在每一层特征图谱的上下左右分别添加1个像素值。两组特征图谱同时与特征检测器进行卷积计算，每组均有192个特征检测器，每个检测器的大小为3×3×192，经过与特征检测器的卷积计算后会生成新的特征图谱。特征检测器沿输入图像的x轴和y轴进行移动，每次移动的长度为1个像素点。因此，在特征检测器沿一个方向移动卷积时会产生(13-3+1×2)/1+1=13个像素点，每列像素点个数的计算过程与行相似，因此最终生成的每幅特征图谱的大小为13×13。特征检测器的个数为384个，最后输出的特征图谱的大小为13×13×384。



图3-6 conv4数据流图

第四层输出的2组13×13×192的特征图谱会作为第五层的数据输入，为了简化后面的计算处理，需要在每一层特征图谱的上下左右分别添加1个像素值。两组特征图谱同时与特征检测器进行卷积计算，每组均有192个特征检测器，每个检测器的大小为3×3×192，经过与特征检测器的卷积计算后会生成新的特征图谱。特征检测器沿输入图像的x轴和y轴进行移动，每次移动的长度为1个像素点。因此，在特征检测器沿一个方向移动卷积时会产生(13-3+1×2)/1+1=13个像素点，每列像素点个数的计算过程与行相似，因此最终生成的每幅特征图谱的大小为13×13。特征检测器的个数为256个，这些特征图谱会经过relu5激活函数的处理，生成激活后的特征图谱，大小仍为两组13×13×128。最后输出的特征图谱的大小为13×13×256。

激活后的特征图谱会再经过池化模块的处理，池化窗口的大小仍为3×3，窗口每次的移动距离为2，则经过池化处理后得到的特征图谱每行有(13-3)/2+1=6个像素点。因此，经过第二层特征检测器的处理后得到两组大小均为6×6×256的特征图谱。



图3-7 conv5数据流图

第六层输入的特征图谱的大小为6×6×256，并用大小为6×6×256的特征检测器对输入的特征图谱进行卷积计算。每个大小为6×6×256的特征检测器对输入的特征图谱进行卷积计算后都会得到一个神经元节点值，并将该值作为输出结果。由于第六层共有4096个6×6×256大小的特征检测器，因此，共有4096个神经元节点的输出。将这4096个输出值进行激活操作，并通过最后的drop操作得到第六层的最终输出值。

由于第六层的计算过程中，采用的特征检测器的大小为(6×6×256)与待处理的特征图谱的大小(6×6×256)相同，即特征检测器中的每个权值只与特征图谱中的一个像素点值进行乘法运算，而在前五层网络中，每个特征检测器的权值都会与多个特征图谱中像素点值进行乘法运算。因此，将第六层称为全连接层。

第五层输出的大小为6×6×256的特征图谱与第六层的4096个节点进行全连接，然后经由激活函数处理后得到4096个输出数据，最后通过dropout6处理后得到最终的4096个数据。



图3-8 fc6数据流图

第六层输出的4096个数据与第七层的4096个节点进行全连接，然后经由relu7进行处理后生成4096个数据，再经过dropout7处理后输出4096个数据。



图3-9 fc7数据流图

第七层输出的4096个数据与第八层的1000个神经元进行全连接，经过计算后得到最后的分类结果。



图3-10 fc8数据流图

根据以上分析可知，在八层网络模型中，各层输出数据量的总和为261448，其中每层具体数据量个数如图3-11所示：

图3-11 Alexnet网络每层数据输出量

从图中可以看出，网络中每层的数据输出量主要集中在前四层，因此在后期BRAM资源进行分配时，前四层将占用大部分的BRAM资源。

* + 1. 乘法运算量分析

在整个Alexnet前向识别网络中，全部八层需要进行724406816次乘法计算。其中，每一层乘法运算次数=该层卷积核个数×每个核进行卷积次数×每个核一次卷积乘法计算量。每个核进行卷积次数=卷积核的大小×输入特征图谱的个数，每个核一次卷积乘法计算量=该层特征图谱的大小。

以第一层为例，模型的输入是227×227×3大小的RGB图像，第一层的卷积核大小为11×11×96，经过卷积之后得到的特征图谱大小为55×55，因此，在第一层网络中，每个核进行一次卷积计算量的大小为11×11×3=363，每个卷积核进行卷积的次数为55×55=3025，共有96个卷积核，故第一层所需要的乘法运算次数为96×3025×363=105415200。

其具体每层所需的乘法计算量如表3-1所示：

表3-1 每层所需乘法计算量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 层数 | 卷积核个数 | 每个核进行卷积次数 | 每个核一次卷积乘法计算量 | 乘法运算次数 |
| conv1 | 96 | 3025 | 363 | 105415200 |
| conv2 | 256 | 729 | 1200 | 223948800 |
| conv3 | 384 | 169 | 2304 | 149520384 |
| conv4 | 384 | 169 | 1728 | 112140288 |
| conv5 | 256 | 169 | 1728 | 74760192 |
| fc6 | 1 | 4096 | 9216 | 37748736 |
| fc7 | 1 | 4096 | 4096 | 16777216 |
| fc8 | 1 | 1000 | 4096 | 4096000 |
| sum |  |  |  | 724406816 |

* 1. 激活函数和池化模块优化

本节主要是针对Alexnet前向识别网络模型在FPGA实现过程中的优化。首先对激活函数进行了简单的介绍，分析了网络模型中各层激活函数所包含的计算量。其次对网络模型中各层池化模块所包含的计算量进行了分析。最后，依据上文对网络模型前向计算过程的分析，结合FPGA的计算特点，对网络模型进行了优化。

* + 1. 激活函数

在Alexnet前向识别网络中，激活函数采用的是ReLU激活函数，当输入值小于零时，输出为零；当输入值大于等于零时，输出数据本身。因此，只需要一个阈值就可以得到激活值，而不用去算一大堆复杂的运算。

在基于FPGA的硬件实现中，实现ReLU激活函数的本质是进行一次比较运算。由网络模型可知，ReLU激活函数存在于前五层卷积层中，全部五层激活函数共需要进行650080次比较运算。其中，每一层所需的比较运算=经过卷积操作后输出的卷积核的大小×经过卷积操作后输出的特征图谱的个数。

以第一层为例，经过卷积操作后输出的卷积核的大小为55×55，经过卷积操作后输出的特征图谱的个数为96个，因此，实现第一层激活函数需要进行55×55×96=290400次比较运算。

其具体每层激活函数所需的比较运算如表3-2所示：

表3-2 每层激活函数所需比较运算

|  |  |  |  |
| --- | --- | --- | --- |
| 层数 | 卷积核大小 | 特征图谱个数 | 比较运算次数 |
| conv1 | 55×55 | 96 | 290400 |
| conv2 | 27×27 | 256 | 186624 |
| conv3 | 13×13 | 384 | 64896 |
| conv4 | 13×13 | 384 | 64896 |
| conv5 | 13×13 | 256 | 43264 |
| sum |  |  | 650080 |

* + 1. 池化模块

为了保持图形旋转、平移、伸缩时特征图谱的不变性，卷积神经网络模型中会在卷积层添加池化模块。在Alexnet前向识别网络中，池化模块采用的是最大池化，即取操作区域内神经元的最大值作为输出，在保证特征图谱不变形的同时，减少了网络模型中的特征参数。

取操作区域内神经元的最大值作为输出即对区域内的特征数据进行比较运算。而在Alexnet网络模型中，池化模块包含于第一层、第二层以及第五层中，其所需要的比较运算共979712次。其中，每一层池化模块所需的比较运算=池化运算后卷积核的大小×池化运算后输出的特征图谱的个数×（采样核的大小-1）。

以第一层为例，池化运算后卷积核的大小为27×27，池化运算后输出的特征图谱的个数为96，采样核的大小为2×2，因此，第一层池化模块所需的比较运算次数为27×27×96×（2×2-1）=559872。

其具体每层池化模块所需的比较运算如表3-3所示：

表3-3 每层池化模块所需比较运算

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 层数 | 池化运算后卷积核的大小 | 池化运算后输出的特征图谱的个数 | 采样核的大小 | 比较运算次数 |
| conv1 | 27×27 | 96 | 3×3 | 559872 |
| conv2 | 13×13 | 256 | 3×3 | 346112 |
| conv5 | 6×6 | 256 | 3×3 | 73728 |
| sum |  |  |  | 979712 |

* + 1. 优化分析

在Alexnet网络中，激活函数和池化模块分别采用的是ReLU激活函数和最大池化。在前向网络第一层、第二层以及第五层中，激活函数ReLU的输出特征图谱直接作为最大池化操作的输入。其中ReLU激活函数公式为：

 （3-1）

最大池化是输出操作区域内最大的神经元节点，即为激活函数输出的神经元，即：

 （3-2）

为了减少模型实现过程的计算量，对两个模块进行优化分析，由于ReLU激活函数和最大池化均为比较输出，且池化模块可以减少网络模型的特征参数，因此考虑改变两个模块在模型中的计算顺序。首先对池化操作后输出神经元节点分析，如公式3-3所示

 （3-3）

可以看出，改变两个模块在模型中的计算顺序对于结果输出没有影响。下面对改变模型计算顺序前后的计算量进行分析。

以第一层为例，根据上文分析可知，模型改变前第一层激活函数模块需要290400次比较运算，池化模块需要559872次比较运算。改变后的网络模型如图3-12所示：

图3-12 模型优化后结构

模型改变后，输入数据仍为原始的227×227×3的图像，图像通过11×11×3的卷积核后输出55×55×96的特征图谱。该特征图谱经过pool运算(池化运算)的处理后形成的像素层的规模为27×27×96，这些像素层经过relu1单元的处理，生成激活像素层，尺寸仍为27×27×96的像素层数据。

根据上节公式总结可知，模型改变后池化运算单元所需的比较运算次数仍为27×27×96×（2×2-1）=559872次，激活函数模块所需的比较运算次数为27×27×96=69984，相较于模型改变前，激活函数所需的比较运算次数减少了220416次。

其具体每层的运算优化如图3-13所示：

图3-13 优化前后计算量对比

由以上分析可知，在保证模型输出值不变的前提下，改变模型中池化操作和激活函数的执行顺序，可以减少模型中比较运算的计算量。其中第一层减少了220416次运算，第二层减少了143360次运算，第五层减少了33652次运算，减少的比较计算的运算量占激活函数和池化模块优化前计算量的76.4%。

* 1. 本章小结

本章主要对影响Alexnet前向网络性能因素进行了研究，并结合FPGA以及网络模型前向计算的对网络模型进行了优化。

首先对Alexnet前向网络计算过程中存在的并行性进行分析，包括神经元节点内的并行、神经元节点间的并行、卷积核的并行以及层间并行，并分别对其进行了介绍。

其次分析了前向网络的整体计算过程，并给出了每层输出特征图谱的数据量大小。然后对网络模型中每层所需乘法计算量进行了分析。

最后，重点研究了网络模型中的激活函数和池化模块，分析了各个模块所需要的比较计算量，并针对Alexnet网络中采用的ReLU激活函数和最大池化的特点，对网络模型进行了优化，改变网络模型中池化操作与激活函数的执行顺序。在保证输出结果不变的前提下，减少了397428次比较运算，占激活函数和池化模块优化前计算量的76.4%。

1. 基于FPGA的Alexnet前向网络关键模块优化设计

本章首先介绍了基于FPGA的Alexnet前向识别网络加速器的总体结构。其次，介绍了网络模型中基本计算模块的并行性设计，包括：卷积核并行性设计、池化模块的并行性设计、激活函数的设计以及全连接层的并行性设计。最后，在芯片内部资源有限的情况下，讨论如何找到每层的优化并行策略，并完成对FPGA内部资源的分配以及网络内每一层结构的设计。

* 1. 加速器总体结构

在现有的基于FPGA的Alexnet前向识别网络实现中，仍然有很大的空间来提高整体识别性能。为了进一步提高性能和提高吞吐量，可以在Alexnet模型上采用流水线结构，并针对不同层次采用不同的并行策略。图4-1为加速器的整体结构。



图4-1 加速器总体架构

在该架构中，5个卷积层和3个全连接层以流水线方式同时运行在8个模块中。因此，需要大量的输入数据和权值数据，这导致了高的数据访问工作量。为了解决这个问题，将每一层计算好的中间结果存储在片上缓冲器中。这样可以大大减少数据访问工作量。由于权值数据量太大，无法将其存储到块RAM中，因此，权值数据存储在外部存储器中。此外，片上缓冲器也有助于数据重用。每个阶段有两个缓冲区，前一层可以写入或读取一个缓冲区，而下一层从另一个缓冲区读取数据。双端口内存块被用作缓冲区可以有效的避免数据重叠并提高整体设计性能。

此外，为了避免前后两幅输入图像的处理时发生冲突，在层与层之间添加握手信号。在当前层数据处理完成以后，检测下一层的工作状态。只有当下一层处理完成，且下一层所需的权值数据准备就绪后，下一层才开始读取前一层缓冲器的数据，开始本层的计算。

* 1. 基本运算单元设计

Alexnet前向识别网络中，主要包括卷积层和全连接层，其中卷积模块内的主要运算包括卷积运算、池化运算以及激活函数。本节结合上文中对网络中并行性特点的分析以及各运算单元的特点，完成各个基本模块的设计，并对其加速比进行分析。

* + 1. 卷积核并行性设计

在Alexnet前向识别网络中，卷积核的大小共有11×11、5×5以及3×3三种不同的尺寸，步长分为4、2、1三种不同的大小。由于卷积核并行性设计的通用型，本文以3×3内核大小和步幅1为例，特征图谱的大小为5×5。

如图4-2所示，输入数据按行输入到处理单元（PE）中，如输入映射中的箭头所示。一维处理单元主要由三个累加器和三个乘法器以及三个多路复用器组成，以选择所需的权重。计算过程如下。



图4-2 一维处理单元处理过程

首先，输入数据通过三个移位寄存器。当第一个输入值到达最右边乘法器的输入时，3个乘法器开始计算。在计算开始3个时钟周期之后，最右边的乘法器产生第一个卷积窗的临时结果，其值为，而另外两个乘法器产生两个临时结果和。当前三行完成计算时，会生成最终结果、、。如图4-3所示。



图4-3 卷积过程

输出映射中的其他结果以相同的方式生成。由于滑动窗口之间存在重叠，所以除了输入图像顶部和底部的2行之外，所有行都应该读取3次。如果要求更高的加速比，则一维处理单元可以扩展到二维处理单元，如图4-4所示。一个二维处理单元由多个一维处理单元组成。在计算过程中，输入数据被广播给这些一维处理单元。因此，可以通过应用不同的权重在多个一维处理单元中同时计算输入映射的每一行。这样，每行只需要被读取一次。因此，在这种情况下，与一维处理单元相比，加速比可以增加3倍。



图4-4 二维处理单元架构

这种结构通常适用于任何卷积操作。使用的乘法器，累加器和多路复用器的数量随着内核大小和步长的变化而变化。给定内核大小kernel\_size和相应的移位步长stride后，一维和二维处理单元所需的乘法器，累加器和多路复用器的数量分别为（4-1）和（4-2）。

 （4-1）

 （4-2）

* + 1. 池化并行性设计

在Alexnet前向识别网络中，只有第一层、第二层以及第五层网络用到池化模块，池化核的大小均为3×3，步长为2。由于池化核并行性设计的通用型，本文以3×3内核大小和步幅2为例，特征图谱的大小为5×5。

如图4-5所示，输入数据按行输入到一维池化模块中，如输入映射中的箭头所示。一维池化模块主要由三个寄存器和两个最大比较器组成，以选择池化区域内的最大值。计算过程如下。

首先，输入数据通过三个移位寄存器。当第一个输入值到达最右边比较器的输入时，两个比较器开始进行比较。在计算开始2个时钟周期之后，最右边的比较器产生第一个池化窗的临时结果，其值为，并作为一维池化模块的输出。



图4-5 一维池化模块

输出映射中的其他结果以相同的方式生成。由于滑动窗口之间存在重叠，所以除了输入图像顶部和底部的2行之外，所有行都应该读取2次。如果要求更高的加速比，则一维池化模块可以扩展到二维池化模块，如图4-6所示。一个二维池化模块由多个一维池化模块组成。在计算过程中，输入数据被广播给这些一维池化模块。因此，可以通过多个一维池化模块同时计算输入映射的每一行。这样，每行只需要被读取一次。因此，在这种情况下，与一维池化模块相比，加速比可以增加3倍。



图4-6 二维池化模块

* + 1. 激活函数并行性设计

对于激活函数，可以将其看作内核大小为1×1和步幅1为的池化操作。由于上文对激活函数和池化模块进行了优化，因此，直接对二维池化模块的输出进行激活运算，如图4-7所示。激活运算即完成神经元节点值与0比较，将最大值输出。因此，其加速比与二维池化模块的并行度有关。



图4-7 池化及激活操作

* + 1. 全连接层并行性设计

全连接层可以看作是卷积核为1×1的矩阵乘法。以Alexnet前向识别网络中的第六层为例详细说明计算模式，其中有9216维输入特征向量和4096维输出特征向量。图4-8中的权重表示为9216×4096矩阵。为了在有限的硬件资源下实现这种大规模的矩阵乘法，我们把它分成多个小规模的矩阵乘法，如图4-9所示。 首先，到作为输入向量，通过应用m×n权重生成到的临时结果。然后我们把换成另一个m×n权重作为输入，将临时结果更新到。在9216输入数据和权重矩阵的前n行之后，可以生成到的最终结果。其他输出结果可以用相同的方式生成。



图4-8 全连接层计算模型



图4-9 全连接层小规模矩阵乘法

* 1. 层结构设计

在基于FPGA的前向识别网络的具体实现过程中，数据的大小采用定点数据表示，其中权值数据以及层间的中间缓存结果用16位的定点数据表示，图片的输入数据用8位的定点数据表示。另外，18Kb BRAM被用作基本内存块，并可扩展为36Kb大小。

由于Alexnet前向识别网络的计算加速性能的重要影响因素为网络计算并行度，因此，在乘累加单元以及块RAM资源有限的情况下，如何提高网络模型计算的并行度将成为层结构设计的重点研究内容。本文首先对每一层的结构设计进行分析，然后对每一层的计算并行度进行介绍。

如图4-10所示为卷积层第一层的设计，首先输入图像为227×227×3的RGB三维图像，与核大小为11×11的卷积核进行卷积，卷积核的权值数据通过读取外部存储器取得。卷积后的数据通过流水加计算得到卷积后的特征图谱并存储至缓存RAM中。由于前向识别网络加速器的设计是基于层间并行计算，为了防止前后两幅输入图像在处理时发生冲突，在设计模块中添加了两个缓存RAM，用于缓存不同图像的中间处理结果，缓存RAM的选择则由双RAM控制单元进行控制选择。取缓存RAM中的中间处理结果进行激活函数和池化操作，由于设计已经对激活函数和池化操作进行了与优化，因此，先对中间结果进行池化操作，再将池化输出结果进行激活。对于激活函数和池化模块的输出，同样采用两个块RAM进行缓存，缓存器的选择同样由双RAM控制单元进行控制选择。当本层处理结束后，会将缓存RAM中的数据进行输出，以供下一层进行计算。



图4-10 conv1设计

由于卷积层第二层在GPU上训练时分为上下两部分进行训练，因此，基于FPGA的第二层设计也将分为两个部分进行设计。如图4-11所示，将第一层输出块RAM分为上下两个部分，两个部分分开同时进行层内的处理，每部分的处理步骤大致与卷积层第一层设计相同。



图4-11 conv2设计

由于第三层的训练没有进行分批处理，且第三层中没有池化操作，因此，卷积层第三层的设计如图4-12所示。在经过乘累加器计算后，无需对中间结果进行缓存，直接对乘累加后的数值进行激活操作即可。其他设计均与第一层设计相同。



图4-12 conv3设计

在确定加速器的每层结构之后，对每层的并行度以及资源使用情况进行详细分析。

在FPGA芯片内部，单片块RAM的容量为18k比特，即位宽为18比特、深度为1024，可以根据需要改变其位宽和深度，但要满足两个原则：首先，修改后的容量不能大于18k比特；其次，位宽最大不能超过36比特。当然，可以将多片块RAM级联起来形成更大的RAM，此时只受限于芯片内部块RAM的数量。此外，由于第一层输出的特征图谱会作为第二层的输入进行并行处理，比如第一层输出为96个27×27大小的特征图谱，将该96幅特征图谱分批进行计算，因此选择如何对中间数据缓存不仅会影响加速器整体使用块RAM的数量，还会对前向识别网络计算的并行对产生影响。

以第一层为例，由于输出的特征图谱的个数为96，每层特征图谱的大小为27×27，数据位宽大小为16比特，因此，第一层输出的每个特征图谱的数据量大小为27×27×16，总的数据输出量大小为27×27×16×96比特。对于96个特征图谱分24批进行处理，即每批的数据量大小为27×27×16×4，需要3个18Kb的块RAM进行存储，整体需要48×3+4×3个18Kb的块RAM进行缓存。其中4为流水加后缓存RAM的个数。以36Kb为基本内存块，故第一层所需的块RAM个数为72个。

由于第一层的输入图像为三维图像，故其二维处理单元的并行度为3，一维并行度选择卷积核的列的大小的两倍为22，因此，第一层卷积层所需的DSP资源为3×22=66个。

第二层的二维处理单元并行度由第一层的输出特征图谱的批处理确定，为24路并行操作。同时第二层中一维卷积的并行度为卷积核的列的大小，因此，第二层卷积核的整体并行度为24×5=120，所需的DSP资源即为120个。

其具体每一层所需的块RAM以及DSP资源分别如图4-13、图4-14所示。

图4-13 每层所需块RAM资源

图4-14 每层所需DSP资源

* 1. 本章小结

本章完成了基于FPGA的Alexnet前向网络关键模块优化设计。在FPGA芯片内部资源有限的前提下，如何选择满足前向网络加速器性能设计要求的并行性结构是当前研究卷积神经网络并行结构的关键问题。本文首先完成了卷积神经网络内部基本模块单元的设计及其二维并行加速的设计，其中包括：卷积核的并行性优化设计、激活函数及池化并行性优化设计、全连接层并行性优化设计，并分析了其二维并行加速的加速比。其次，根据每层输出的缓存结果的数据量大小，对芯片内部的块RAM资源进行合理的分配，从而确定了每层网络的并行度，完成了网络内每一层结构的设计。

1. 基于FPGA的Alexnet前向网络实现及性能分析

本章将以图像分类的加速识别作为应用背景，详细介绍加速器识别加速性能。首先在5.1节对加速器的实现平台及开发环境进行了介绍。然后，在5.2节详细介绍了加速器整体框架的设计。最后，在5.3节对加速器的性能进行分析，其中包括每层加速效果的分析以及总体性能的分析。

* 1. 实现平台及开发环境

在本文研究中，使用了Xilinx公司开发的ISE 14.7环境进行硬件开发，并使用ISE自带的仿真软件Isim进行仿真验证。以硬件描述语言Verilog作为设计工具进行模型设计输入，使用的硬件开发平台为Xilinx kintex-7 XC72K325T FPGA。此外，时钟频率为100Mhz。

卷积神经网络模型采用的是Alexnet的图像分类模型，其中输入图像为227×227的三通道的RGB图像，每个像素点的取值范围为0~255。由于在FPGA上仅仅实现网络的前向识别过程，网络的训练部分将在GPU平台上完成，模型的训练采用当前普遍认可的caffe框架，并将训练好的模型参数与图片数据一起外部存储器中。

* 1. 整体框架设计

基于FPGA的Alexnet前向识别网络加速器的整体实现框图如图5-1所示。其中，外部存储器负责存放训练完成的权值数据和图像数据，Input Controller单元负责控制权值数据以及图像数据到Computation Kernels的数据读入。BRAM负责中间特征图谱数据的缓存。Output Controller负责将识别分类结果进行输出。



图5-1 整体框架设计

加速器的整体控制由Computation Kernels控制，当外部存储器中的权值数据和图像数据准备就绪后，Computation Kernels会根据前向计算过程中的计算状态，控制Input Controller单元将相应的数据输入到Computation Kernels中的缓存单元。同时将计算过程中的中间缓存结果存储到BRAM中，同时会读取BRAM中缓存的特征图谱共下一层卷积计算。当第八层网络计算计算后，将最后的识别结果进行输出。

* 1. 性能与分析

卷积神经网络硬件实现的性能分析主要体现在识别精度和识别速率两个方面。本节将就这两个方面对基于FPGA的加速器设计进行验证和分析，其中包括每层加速效果的分析以及总体性能的分析，并将识别速率与基于CPU以及基于GPU的识别效果进行对比，验证其加速效果。

* + 1. 每层加速效果分析

如图5-2至5-9所示分别为卷积神经网络第一层至第八层的状态信号变化图。以第二层状态信号变化为例，如图5-3所示，当第二层所需的权值数据准备完毕后，layer2\_data\_ready信号会置高，此时layer\_state信号同样置高，表示第二层开始计算。此时，buffer\_choose信号为零，表示选择第一个buffer群作为缓存进行数据存储，至于具体选择第一个buffer群中的哪一个buffer，则由choose\_for\_one\_ram信号进行控制。当该层计算结束后，layer2\_finish信号会拉高，同时layer\_state信号置低。当下一帧图像进入第二层处理模块时，buffer\_choose信号置1，表示选择第二个buffer群作为缓存进行数据存储。其他层状态信号变化与第二层类似。

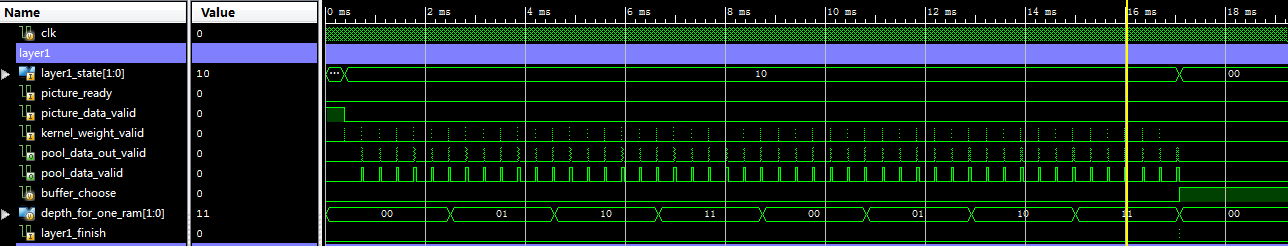


图5-2 第一层状态信号变化图

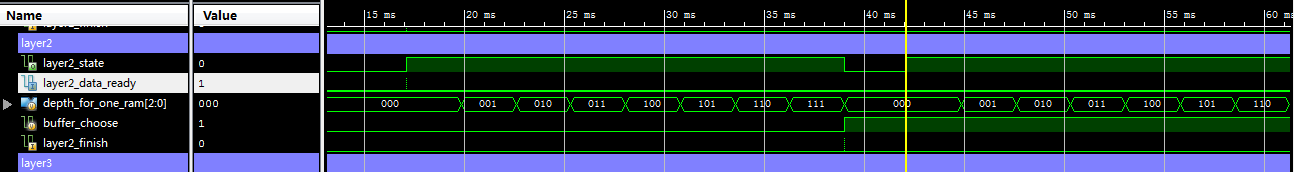


图5-3 第二层状态信号变化图

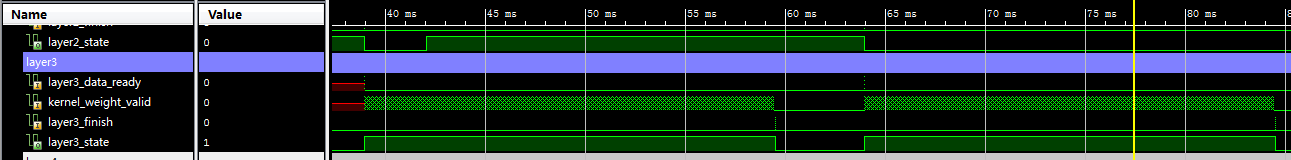


图5-4 第三层状态信号变化图

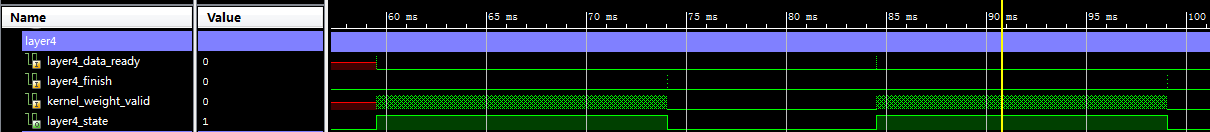


图5-5 第四层状态信号变化图

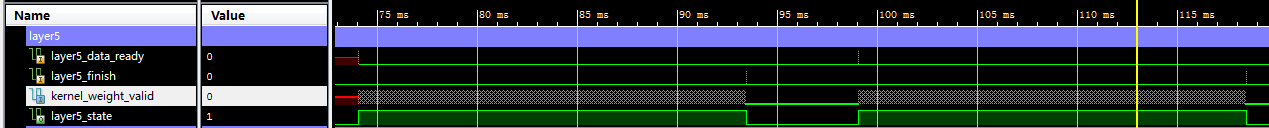


图5-6 第五层状态信号变化图

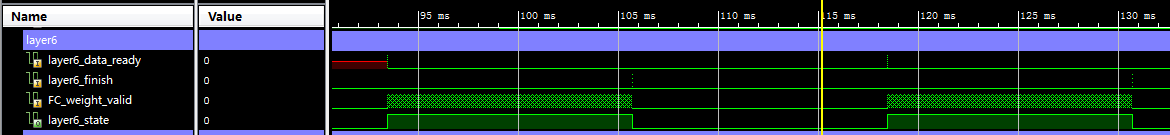


图5-7 第六层状态信号变化图

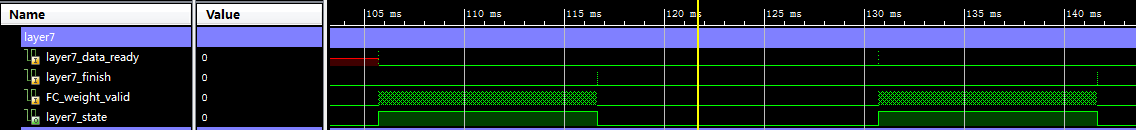


图5-8 第七层状态图

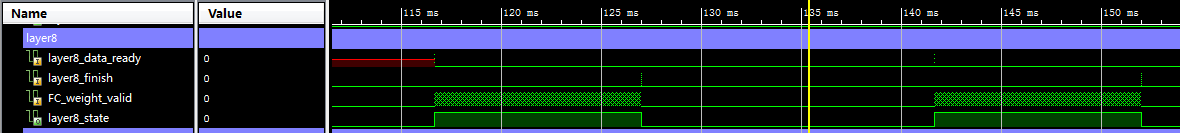


图5-9 第八层状态信号变化图

根据以上状态变量变化图，可以得出加速器每层处理所需要的时间，据图每层计算时间如图5-10所示。

图5-10 每层所需的计算时间

由上图可知，加速器设计中耗时最长的为第二层卷积层，计算时间为22毫秒。因此，加速器处理过程中可以每隔22毫秒输入一副图像进行计算。为了不出现处理数据的重叠，本文按每隔23毫秒输入一帧待处理的图像。

* + 1. 总体性能分析

如图5-11所示为前向识别网络加速器顶层封装包含的模块，主要包含八层网络分别对应的模块设计。图5-12为其中某一层网络的封装包含的模块，主要包含卷积计算模块、池化模块以及缓存块RAM设计模块。

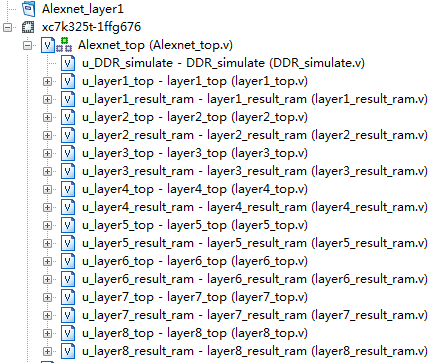


图5-11 顶层封装包含的模块

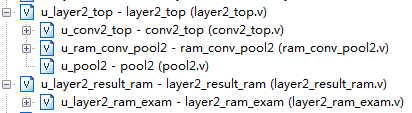


图5-12 第二层封装包含的模块

如图5-13所示为前向网络加速器每层握手信号状态变化。当picture\_data\_valid信号置高时，第一层开始计算，计算过程中，layer1\_state信号为高电平，计算结束后将layer1\_state信号拉低，同时layer1\_finish信号置高，一个时钟周期后再拉低，表示网络模型第一层计算完成。

在第一层处理结束，且layer2\_data\_ready信号置高时开始第二层的计算。加速器其他层的状态信号变化与第一层相同，具体变化如下图所示。



图5-13 加速器每层握手信号状态变化图

本文设计的基于FPGA的Alexnet前向识别网络加速器的整体硬件资源消耗如表5-1所示。其中，触发器资源使用9809个，占总触发器资源的54%；查找表资源使用了12855个，占总的查找表个数的6%；DSP48资源使用了478个，占总的DSP48资源的56%；36Kb大小的块RAM资源使用了364个，占总的块RAM资源的81%。

表5-1 加速器整体资源使用情况

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Resource Utilization on Xilinx K7 | | | |
| FF | LUT | DSP48 | BRAM(36kb) |
| Utilization | 9809 | 12855 | 478 | 364 |
| Available | 18070 | 203800 | 840 | 445 |
| Percent(%) | 54 | 6 | 56 | 81 |

表5-2为不同平台下处理速度的对比。在GPU平台下，识别一副图像需要28.34毫秒的时间，即每秒钟处理35.29帧图像。其中，GPU的型号为GTX1080TI。在型号为Intel CoreTM i7-7700 CPU @3.60GHz x8的CPU处理器上，完成一副图像的识别需要817.34毫秒的时间，即每秒钟只能处理1.22帧图像。在本设计使用Xilinx kintex-7 XC72K325T FPGA开发平台上，识别一副图像需要23毫秒的时间，即每秒钟处理43.48帧图像。对比发现，本文设计的基于FPGA的Alexnet前向识别网络加速器具有很好的加速效果。

表5-2 不同平台识别效果对比

|  |  |  |  |
| --- | --- | --- | --- |
|  | Platform | Processing time/image(ms) | FPS |
| GPU | GTX1080TI | 28.34 | 35.29 |
| Caffe CPU | Intel CoreTM i7-7700 CPU @3.60GHz x8 | 817.34 | 1.22 |
| FPGA | K7-325T | 23 | 43.48 |

为了验证加速器设计的正确性，本文选择人脸图片作为加速器图像分类应用的输入，同时在GPU上进行识别。如表5-3所示为GPU平台与FPGA平台识别输出概率对比。

表5-3 GPU与FPGA平台识别输出概率对比。

|  |  |  |  |
| --- | --- | --- | --- |
| 幅数 | GPU输出概率 | FPGA输出概率 | 绝对误差 |
| 1 | 0.652914 | 0.627621 | 4.03% |
| 2 | 0.694324 | 0.668584 | 3.85% |
| 3 | 0.685607 | 0.662038 | 3.56% |
| 4 | 0.704653 | 0.679249 | 3.74% |
| 5 | 0.678925 | 0.652122 | 4.11% |
| 6 | 0.659943 | 0.634805 | 3.96% |
| 7 | 0.701465 | 0.675067 | 3.91% |
| 8 | 0.674529 | 0.649522 | 3.85% |
| 平均值 | 0.681545 | 0.656126 | 3.87% |

从表中可以看出，八幅输入图像在不同平台上最大的绝对误差为4.11%，平均误差值为3.87%，误差在相对较小的范围内，因此证明基于FPGA前向网络加速器设计的正确性。

* 1. 本章小结

本章以图像分类的加速识别作为应用背景，对基于Xilinx kintex-7 XC72K325T硬件开发平台的Alexnet前向识别网络的加速设计进行性能的分析和验证。

首先，对加速器的实现平台及开发环境进行了介绍。其中，FPGA开发平台主要负责前向识别模块的设计实现，GPU平台负责网络模型的训练，训练框架采用当前普遍认可的caffe框架。

其次，介绍了基于FPGA实现的网络前向识别加速器的整体框架结构，并对结构设计中的模块控制以及数据流向进行了分析。

最后对加速器的硬件实现进行了验证及性能分析，其中包括每层的加速效果分析以及总体性能分析。给出了前向识别网络中每一层具体所需的计算时间，列出了加速器设计整体的资源使用情况，完成了与GPU、CPU识别速率的对比，并对识别结果的正确性进行了验证。

1. 总结与展望
   1. 工作总结

随着芯片制造工艺和协处理器的不断发展，FPGA芯片内部集成的硬件乘累加(Multiply-add Accumulation，MAC)单元及片内块RAM资源不断增加，功能也越来越强大。这些特点使得FPGA在计算密集型硬件加速方面具有巨大优势。作为典型的计算密集型应用，多层神经网络中的卷积神经网络在图像分类、模式识别、图像分割以及计算机视觉领域都具有重要的指导意义和应用价值，一直备受学术界和工业界的青睐。然而，目前通用处理器实现卷积神经网络的方法，无法充分挖掘网络模型内部的并行性。在目前对实时性和低功耗的应用需求越来越旺盛的前提下，越来越多的研究人员开始采用FPGA对基于卷积神经网络的应用进行开发。

Alexnet作为卷积神经网络领域内具有重要历史意义的一个网络模型，不仅证明了卷积神经网络在复杂模型下的有效性，并使用GPU使大数据训练在可接受的时间范围内得到了结果。因此，研究Alexnet模型的计算加速对复杂模型下卷积神经网络加速的研究具有重要意义。

本论文以此为背景，对当前研究成果做了充分的研究和分析，设计实现了一种基于FPGA的Alexnet前向识别网络加速器，通过优化模型结构、流水式层间处理以及提高网络并行性来提高网络的整体识别速度。本文的主要研究工作包括：

（1）对Alexnet前向识别网络性能的主要影响因素进行了研究并对网络模型中激活函数和池化模块进行了优化改进。卷积神经网络内部存在的大量的并行计算是基于FPGA实现整个加速框架的基础和核心，因此，论文首先对网络模型中存在的并行性计算进行了研究，并对神经元节点内的并行、神经元节点间的并行、卷积核的并行以及层间并行进行了详细介绍。其次，给出了前向计算过程的分析以及乘法计算量的分析。最后，分析激活函数和池化模块的计算量，并结合激活函数和池化模块比较计算输出最大值的特点以及FPGA的特性，优化了基于FPGA实现的网络模型结构。在保证输出一致性的前提下，减少了397428次比较运算，占激活函数和池化模块优化前计算量的76.4%。

（2）完成了基于FPGA的Alexnet前向网络关键模块优化设计。在FPGA芯片内部资源有限的前提下，如何选择满足前向网络加速器性能设计要求的并行性结构是当前研究卷积神经网络并行结构的关键问题。本文首先完成了卷积神经网络内部基本模块单元的设计及其二维并行加速的设计，其中包括：卷积核的并行性优化设计、激活函数及池化并行性优化设计、全连接层并行性优化设计，并分析了其二维并行加速的加速比。其次，根据每层输出的缓存结果的数据量大小，对芯片内嵌的块存储器进行合理的分配，从而确定了网络模型中每层计算的并行度，完成了网络内每一层结构的设计。

（3）完成了基于FPGA的Alexnet前向网络实现及性能分析。基于FPGA开发平台，完成Alexnet前向网络整体框架设计及实现，编写仿真文件进行验证，并完成对资源使用情况以及设计性能进行分析。其中包括每层的加速效果分析以及总体性能分析。给出了前向识别网络中每一层具体所需的计算时间，列出了加速器设计整体的资源使用情况，完成了与GPU、CPU识别速率的对比，并对识别结果的正确性进行了验证。

* 1. 展望

在完成上述工作成果的同时，由于时间的关系，本文的研究仍然存在很多地方需要进一步完善，希望在接下来的工作中对以下提出的问题进行更深层次的研究和讨论：

1、数据精度与性能的权衡

通过对卷积神经网络算法的相关研究发现，卷积神经网络中的基本运算比如池化单元、激励函数以及浮点数乘法或加法运算等等对数据精度的要求并不高。此外，局部数据准确度的部分损失并不会对系统整体的准确度产生太多的影响。因此，基于FPGA的Alexnet前向网络加速器的设计，可以通过采用更小的数据位宽，比如将权值数据或者层之间的缓存数据用8比特的数据位宽代替16比特，来减少FPGA芯片内部的块RAM资源的消耗。另外可以通过或门和与门来近似实现加法和乘法操作，通用在适度削减数据精度的情况下大帧减少乘累加器资源的使用。在片内块RAM资源和乘累加器资源使用量减少的前提下，相同资源下网络模型的计算并行度便可以提高，从而使得整个前向网络识别加速器的性能得到提升。

2、通用IP核设计

卷积神经网络中卷积核的大小有不同的尺寸，对于不同尺寸卷积核的重复设计不仅浪费设计时间，同时不能满足很好的适用性。因此，设计一个通用的卷积器IP核，不同大小的卷积只需配置相应的核大小参数，对于网络模型具有很好的可移植性。

参考文献

1. Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553):436-444.
2. Wang P, Li W, Liu S, et al. Large-scale Continuous Gesture Recognition Using Convolutional Neutral Networks[J]. 2016.
3. Wang P, Li W, Liu S, et al. Large-scale Continuous Gesture Recognition Using Convolutional Neural Networks[C]// International Conference on Pattern Recognition. IEEE, 2017.
4. Schmidhuber J. Deep learning in neural networks: an overview[J]. Neural Networks the Official Journal of the International Neural Network Society, 2015, 61:85.
5. Yuan Z W, Zhang J. Feature extraction and image retrieval based on AlexNet[C]// Eighth International Conference on Digital Image Processing. 2016:100330E.
6. Downing K. Evolving Artificial Neural Networks[C]// MIT Press, 1999:1423-1447.
7. 陈先昌. 基于卷积神经网络的深度学习算法与应用研究[D]. 浙江工商大学, 2013.
8. 楚敏南. 基于卷积神经网络的图像分类技术研究[D]. 湘潭大学, 2015.
9. Delakis M, Garcia C. text Detection with Convolutional Neural Networks[C]// Visapp 2008: Proceedings of the Third International Conference on Computer Vision Theory and Applications, Funchal, Madeira, Portugal, January. DBLP, 2015:290-294.
10. 侯宇昆. 卷积神经网络概述[J]. 中国新通信, 2017, 19(9):45-45.
11. Fukushima K. Neocognitron: A hierarchical neural network capable of visual pattern recognition[J]. Neural Networks, 1988, 1(2):119-130.
12. Lecun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural Computation, 2014, 1(4):541-551.
13. Garcia C, Delakis M. Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2004, 26(11):1408-23.
14. Frome A, Cheung G, Abdulkader A, et al. Large-scale privacy protection in Google Street View[C]// IEEE, International Conference on Computer Vision. IEEE, 2010:2373-2380.
15. Simard P Y, Steinkraus D, Platt J C. Best practices for convolutional neural networks applied to visual document analysis[C]// International Conference on Document Analysis and Recognition. IEEE Computer Society, 2003:958.
16. Lecun Y, Muller U, Ben J, et al. Off-road obstacle avoidance through end-to-end learning[C]// International Conference on Neural Information Processing Systems. MIT Press, 2005:739-746.
17. Happold M, Ollis M. Using Learned Features from 3D Data for Robot Navigation[M]// Autonomous Robots and Agents. Springer Berlin Heidelberg, 2007:61-69.
18. Lécun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 2001, 86(11):2278-2324.
19. Hadsell R, Sermanet P, Ben J, et al. Learning long-range vision for autonomous off-road driving[J]. Journal of Field Robotics, 2010, 26(2):120-144.
20. Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(2):2012.
21. Neveu J N P, Kumar M J. Implementation of the neocognitron on a SIMD architecture[C]// Intelligent Information Systems,1994. Proceedings of the 1994 Second Australian and New Zealand Conference on. IEEE, 1994:179-183.
22. Boser B E, Sackinger E, Bromley J, et al. An analog neural network processor with programmable topology[J]. IEEE Journal of Solid-State Circuits, 1991, 26(12):2017-2025.
23. Cloutier J, Pigeon S, Boyer F R, et al. VIP: An FPGA-based Processor for Image Processing and Neural Networks[C]// International Conference on Microelectronics for Neural Networks and Fuzzy Systems. IEEE Computer Society, 1996:330.
24. Ariyadoost H, Kavian Y S, Ansari-Asl K. Two dimensional systolic adaptive DLMS FIR filters for image processing on FPGA[C]// Electrical Engineering. IEEE, 2012:243-248.
25. Savich A W, Moussa M, Areibi S. The Impact of Arithmetic Representation on Implementing MLP-BP on FPGAs: A Study[J]. IEEE Transactions on Neural Networks, 2007, 18(1):240-52.
26. Sankaradas M, Jakkula V, Cadambi S, et al. A Massively Parallel Coprocessor for Convolutional Neural Networks[C]// IEEE International Conference on Application-Specific Systems, Architectures and Processors. IEEE, 2009:53-60.
27. Chen Y, Sun N, Temam O, et al. DaDianNao: A Machine-Learning Supercomputer[C]// Ieee/acm International Symposium on Microarchitecture. IEEE Computer Society, 2014:609-622.
28. 方睿, 刘加贺, 薛志辉,等. 卷积神经网络的FPGA并行加速方案设计[J]. 计算机工程与应用, 2015, 51(8):32-36.
29. 吴正文. 卷积神经网络在图像分类中的应用研究[D]. 电子科技大学, 2015.
30. Ketkar N. Convolutional Neural Networks[J]. 2017.
31. 彭玉炳. 基于异构计算的CNN并行框架的设计与实现[D]. 电子科技大学, 2016.
32. 赵爽. 基于卷积神经网络的遥感图像分类方法研究[D]. 中国地质大学(北京), 2015.
33. Surhone L M, Tennoe M T, Henssonow S F. Activation Function[M]. 2010.
34. Surhone L M, Timpledon M T, Marseken S F, et al. Sigmoid Function[J]. 2010.
35. Kapanova K G, Dimov I, Sellier J M. On randomization of neural networks as a form of post-learning strategy[J]. Soft Computing, 2015:1-9.
36. Zhang C, Woodland P C. DNN speaker adaptation using parameterised sigmoid and ReLU hidden activation functions[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2016:5300-5304.
37. Sun J, Cai X, Sun F, et al. Scene image classification method based on Alex-Net model[C]// International Conference on Informative and Cybernetics for Computational Social Systems. IEEE, 2016:363-367.
38. Cesur E, Yildiz N, Tavsanoglu V. On an Improved FPGA Implementation of CNN-Based Gabor-Type Filters[J]. IEEE Transactions on Circuits & Systems II Express Briefs, 2013, 59(11):815-819.

致谢

转眼间两年半的研究生生活即将结束，回首这段学习生涯，我收获了很多。从最开始接受任务的不知所错，到学会查找资料收集文献为解决问题做积累；从最开始遇到bug的无从下手，到最后学会如何独立地思考问题解决bug。在实验室的两年半时间，使得我的专业水平和综合素质都得到了很大的提升，这些都离不开我的导师、同学以及家人的支持和帮助，再次向他们表示由衷的感谢。

首先要感谢我的导师别红霞教授。别老师是一个对待工作和学习非常严谨的人，对项目和论文都以高标准来要求我们。她做事一丝不苟的态度也一直感染着我们。这对我们形成良好的科研和工作习惯打下了良好的基础。

其次要感谢我的家人，他们一直在背后默默支持着我去追寻自己的目标，为我提供了坚强的后盾。在我遇到挫折时他们给了不断前进的动力以及解决问题的勇气。

最后还要感谢舍友以及实验室的同学，与你们一起度过了一个非常丰富、充实和快乐的研究生生活。感谢别志松老师、张大旺师弟在我学习和研究生生涯里给予的帮助和鼓励。

攻读学位期间取得的研究成果

[1] 邱宇,别红霞. 基于FPGA的Alexnet局部响应归一化函数实现.中国科技论文在线 [2017-12-04].http://www.paper.edu.cn/releasepaper/content/201712-32.