# 第一章 绪论

## 1.1研究背景和意义

人工神经网络，特别是卷积神经网络，是一种受到生物视觉神经网络启发而发展起来的计算机视觉模型，它通过模仿生物（人类）大脑通过视觉识别物体的方法，设计多层神经元模型，让计算机视觉识别系统能够提高识别率，并达到较强的抗位移和形变干扰的能力[1]。卷积神经网络通过对大量有标签样本数据的学习，而实现对图像中各种物体的识别。比较成熟的卷积神经网络根据规模可以大致分为浅层网络和深层两类，深层网络广泛用于识别较为庞大的图片数据库中图片的内容，和人脸识别工作。浅层网络，如LeNet-5有更为明确的应用意义，专门用于识别手写数字图片。较高的正确率使得这种网络可以实际应用在现实生活中，完成某些实际任务。

卷积神经网络的每层是一张或者若干张特征映射图组成，每层特征映射图的一个像素由上一层一个局部区域的像素乘以一组权值，这是一个图像与核函数卷积的过程，因此被成为卷积神经网络。由于每张特征映射图用同一个核函数做卷积，因此网络权值是共享的。最后，特征映射图会经过一个下采样的过程，将分辨率降低。由于这样的特征，网络参数数量较少，训练和学习更容易，并且能摒弃图像中的冗余信息，从而在识别图像的时候具有对识别对象位移、缩放和形变的抗干扰能力。

由于卷积神经网络具备上述优势，该算法被广泛应用，比如一种在ImageNet数据库中做识别的网络能够识别数据库中1000类共一百二十万张图片，其前五选错误率约为17%[2] 。而这种深层网络的运算量依然非常庞大，常常需要使用服务器甚至多个GPU组成的高端运算平台，才能满足这种深层网络的运算需求，否则这种网络的识别运算会非常耗时，其训练甚至会因为迭代不能快速收敛而无法完成。鉴于卷积神经网络对运算设备的要求如此高，这将极大地限制这种具备巨大性能优势的计算机视觉算法的广泛应用，比如在一些具有小尺寸、低功耗要求的场合，使用软件在图形处理器（Graphics Processing Unit，GPU）等高端平台上实现的卷积神经网络将完全无法发挥其优势。

目前，常用于小尺寸、低功耗场合的硬件设备中，嵌入式处理器由于运算架构与通用处理器类似，仍然以串行方式实现运算，无法充分应对卷积神经网络中大量独立、重复的乘法、加法运算。而专用的数字信号处理器（Digital Signal Processor，DSP）芯片虽然有针对乘、加运算的优化，但是仍然难以胜任以大规模图像卷积运算为主的卷积神经网络算法。面对这种情况，具有大量阵列形式逻辑、运算单元的现场可编程门阵列（Field Programmable Gate Array， FPGA）芯片则在尺寸、功耗和并行运算方面有突出的优势。借助FPGA中丰富的逻辑资源和专用乘法器、数字信号处理单元等资源实现的卷积神经网络，能高度并行地执行算法中大量重复、独立的乘法、加法运算，以较低的功耗，实现卷积神经网络中密集的运算。

## 1.2研究现状

本节将介绍卷积神经网络的应用、软件实现以及其硬件实现方面的研究现状。

自从1998年的LeNet-5[3]被成功应用在交易支票读取系统中以来，越来越多的卷积神经网络被应用到实际问题的解决中来。微软的手写识别系统和光学字符识别系统中应用了Simard和Chillapilla等人的三种卷积神经网络[4][5]。谷歌街景中的人脸检测和车牌识别应用了Frome等人提出的神经网络[6]。而法国电信视频会议中的人脸识别系统应用到了Garcia等人提出的神经网络[7]。可见，卷积神经网络已经成为计算机视觉和图像识别领域重要的研究方向，其研究成果也已经可以被广泛地应用在现实生活中的方方面面。

作为典型的运算密集型算法，卷积神经网络的软件实现方式自其产生之初，就与并行计算紧密结合，利用GPU对乘法运算的先天优势【】，深层卷积神经网络大多都需要在GPU上运算，才能保证优良的性能。文献[2]中提出的用于图像识别的深层网络就是在两块NVIDIA GTX 580 3GB GPU上完成训练和识别的。然而这样的GPU一块就有超过200W的功耗，并且这样的平台往往是服务器，其体积非常大，非常不灵活。

卷积神经网络硬件实现[8][9]的工作早在20世纪90年代就已经有人涉足，当时贝尔实验室开发了人工神经网络芯片ANNA，该芯片通过数-模混合架构，实现了同时完成64个8\*8卷积的任务。这时对神经网络的硬件实现相对初级，运算能力较低，精度也无法达到较高水平，其状态表示用4bit量化，权重用6bit量化[10]。Cloutier等人[11]设计了基于FPGA的图像和神经网络处理器，该处理器架构采用与脉动阵列[12]相似的二维网格结构，网格中每个节点是一个处理单元，每个单元通过与其上下左右四个单元循环互联，构成处理器矩阵，由单指令多数据（Single Instruction Multi Data，SIMD）控制器控制。该处理器规模较小，处理的图像数据量很少，用于完成识别任务的图像每个像素只有3比特。Antony等人[13]通过FPGA实现了多层感知器网络，并详细分析了定点数格式表示和浮点数格式表示[14]以及串行实现和并行实现对硬件实现的神经网络的性能的影响。然而，Antony等人并未给出具体的卷积神经网络FPGA设计方案。Murugan等人在文献[15]设计了大量使用FPGA中DPS单元和片外DDR存储器的卷积神经网络协处理器，该协处理器同时完成4个二维卷积操作。其特点在于组合较低精度的数据，充分利用片外存储器带宽，让算法中最关键的中间数据缓存器可以放在分布式的片外存储器中。然而，为了实现复杂的运算模块控制，该设计仍然需要使用PC作为上位机，同时，上位机也参与了卷积神经网络的部分运算。若要将这种卷积神经网络协处理器应用于小尺寸、低功耗平台或者其他移动智能平台，尚需要较多的改动。文献[16]提出定制化多芯片机器学习芯片架构，设计了卷积神经网络、深度神经网络专用芯片，该芯片使用片内存储模式，内部带宽高很大而外部带宽较小，且并行程度很高，能够达到很好的并行加速效果。然而由于其专用芯片的设计模式，灵活性较低。文献[17]在FPGA上设计了深度流水线结构的浅层卷积神经网络运算系统，其设计简化了LeNet-5的网络结构，通过挖掘卷积神经网络层间流水线运算的特点设计系统，得到了良好的加速效果。然而其作者侧重系统设计，忽略了模块化设计要求，使得设计本身虽然在FPGA上实现，却仍然缺乏应有的灵活特性。而其他的运算平台，诸如嵌入式处理器和DSP芯片等，由于其固有的串行特性并不适合对卷积神经网络算法实现加速，因此很少见到相关的实现案例。

## 1.3本文的研究方向

本文充分挖掘卷积神经网络本身的并行运算特征，提出“从特征映射图到特征映射图”的运算层次，并结FPGA阵列形式的逻辑、运算单元的架构特点，针对小尺寸、低功耗应用场合，探索基于FPGA的卷积神经网络应用方案，详细介绍FPGA卷积神经网络关键电路模块的设计方案，以及如何逐步分模块在FPGA芯片上实现完整的卷积神经网络系统。完成设计之后，本文以静态手势识别这一实际任务为例，验证基于FPGA的卷积神经网络在应用方面的优势和不足。最后，本文总结了基于FPGA实现的卷积神经网络在小尺寸、低功耗平台上的实际意义，并提出对卷积神经网络芯片未来发展的展望

## 1.4本文的组织架构

本文的结构按照如下思路组织：第一章，绪论：介绍卷积神经网络及其FPGA实现的背景，并引出本文设计的卷积神经网络FPGA实现方案。第二章，详细介绍卷积神经网络的运算模型，并举例详细介绍这种运算结构。第三章，详细介绍FPGA平台和FPGA实现的卷积神经网络的架构优势。第四章，详细介绍基于FPGA的卷积神经网络各个模块的设计方案，设计思路和设计过程。第五章，探讨了完整卷积神经网络FPGA系统所涉及的层内并行、串行运算子系统和完整的多层卷积网络FPGA系统。第六章，对本文设计的模块和系统进行实验和优化，并以静态手势识别网络作为完整的卷积神经网络FPGA系统，验证各模块的性能。结论：总结本文设计的基于FPGA的卷积神经网络，同时提出对该网络的进一步设想。图1- 1为本文组织架构图。

|  |
| --- |
|  |

图1- 1论文组织架构图

# 第二章 卷积神经网络

## 2.1卷积神经网络的产生和发展

1962年Hubel和Wiesel研究了猫视觉皮层细胞，并且提出了感受野(receptive field)的概念，1984年日本学者福岛在设计手写阿拉伯数字识别系统的时候，根据感受野的概念，提出了神经认知机(neocognitron)这个概念[1]。这种神经认知机甚至能够识别形变比较严重的手写数字。福岛提出的神经认知机是一个层级结构的多层神经元网络，输入信息逐层通过网络被处理。网络中每个神经元选择性地对特定的激励模式做出反应。福岛的这个神经认知机是卷积神经网络的第一个现实网络，同时他也首次将感受野的概念应用在了人工神经网络中，之后，“局部感受”、“权值共享”和“下采样”成为被大家公认的卷积神经网络基本特点。近几年来卷积神经网络在许多领域不断取得进展，在语音识别、人脸识别、通用物体识别、运动分析、自然语言处理等方面均有突破。

## 2.2卷积神经网络的基本运算模型

卷积神经网络以待识别的图像直接作为输入，经过多个卷积层、子采样层和全链接层处理，得出识别结果。其中卷积层将输入的图像（这些图像可能是输入层的原始图像，也可能是前一个卷积层或子采样层处理后的特征映射图）与若干个卷积核分别做卷积，输出相应数量该层的特征映射图。图2- 1表示1-to-n和m-to-n两种卷积示意图。子采样层的操作也一般被称为池化（pooling），该层对卷积层输出的若干个特征映射图做减采样操作。一般情况下是对输入图像分割为不重叠的2\*2像素小块，每块对应的输出图像素为这一小块四个像素的和或最大值，根据取值的方法不同，分别被称为“均值子采样”或“最大值子采样”，其运算示意图如图2- 2所示。卷积层或子采样层输出的每个特征映射图的每个像素，一般会加上一个与生成该特征映射图的卷积核相对应的的偏置，并且将每个像素值带入一个激活函数，函数的输出作为该层最终输出的特征映射图的像素值。这里激活函数的选择可以是多种多样的，它们分别有各自的优缺点。Sigmoid函数曾被认为是神经网络的核心所在，从数学上来看，非线性的Sigmoid函数对输入为0附近的信号有较大增益，对输入远离0的区域的信号增益小，在信号的特征空间映射上，有很好的效果。从神经科学上来看，0附近的区域酷似神经元的兴奋态，远离0的区域酷似神经元的抑制态，因而在神经网络学习方面，可以将重点特征推向0附近的区域，将非重点特征推向0附近的区域【】。双曲正切函数形状与Sigmoid函数基本一致，只是小于零的部分会趋近于-1，适用于数据标签是-1到1的情况。ReLU函数对比Sigmoid之类的函数，主要变化有三点：单侧抑制；相对宽阔的兴奋边界；稀疏激活性。另外，相比于普通的Sigmoid函数，采用ReLU激活函数的神经网络能更快地学习特征。上述几种激活函数的图像在图2- 3中均有展示。经过多个卷积层和子采样层的处理，每个特征映射图的尺寸最终会很小，而其数量会很多，这就是原始输入图像经过特征提取后得到的抽象特征。最后这些特征将被送入一个或几个全连接的传统神经网络进行分类，得到最终的识别结果。

|  |
| --- |
|  |

图2- 1 1-to-n和m-to-n两种卷积示意图

|  |
| --- |
| 图2- 2子采样运算示意图 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 图2- 3 Sigmoid函数图像、双曲正切函数图像和ReLU函数图像（右图红线） | | |

## 2.3举例说明卷积神经网络的运算模型

以图像本身直接作为神经网络的输入，由于图像像素数量大，因此全连接的传统神经网络就需要数量非常巨大的权值参数。以100\*100的图像（共1万个像素）为例，全连接网络需要1亿个参数，使得每个神经元与输入图像的每个像素相连接。然而我们知道，图像的空间联系往往是局部的，每个神经元并不需要对图像全局做出反应，而通常只需要对图像的一小块做出反应。结合感受野的概念，假设我们只需要对图像中一个5\*5的小块做出反应，而且，每个特征映射图对输入图像做出相同的反应，也就是一个特征映射图共享一组25个权值，这样多个特征映射图（这里假设有6个）就只需要150个参数。这也体现了卷积神经网络权值共享的巨大优势——大大减小了网络的规模。总结下来，结合局部感受野、权值共享以及子采样这三大主要思路，就获得了卷积神经网络的图像识别能力，并且，这种识别能力在某种程度上达到了位移、尺度、形变不变性。

正因为网络规模的缩小和对位移、尺度变化与形变的一定程度的抵抗能力，使得卷积神经网络成为目前应用最广泛的深度神经网络之一。作为最经典的手写数字识别网络之一，于20世纪九十年代由LeCun提出的LeNet-5，其识别准确率已经达到相当高的水平，已经在美国银行系统内用于识别支票上的手写数字。该网络结构示意图如下，不包括输入层的情况下，共有7层，输入32\*32像素的图像。每个卷积层卷积核大小为5\*5像素；每个子采样层做加和子采样，各个子采样窗口之间没有重叠部分，子采样结果乘以系数并加偏置；激活函数采用Sigmoid函数，在子采样层之后，卷积层不设置激活函数。网络第一个卷积层输出6个特征映射图，第二个卷积层输出16个特征映射图，然而，第二个卷积层的输入是与前一层输出的6个特征映射图中的部分连接的，其连接情况可以用表2- 1来表示。

例如，第二个卷积层的第一个卷积核就是将前一层输出的前三个特征映射图分别卷积一次，并且将卷积结果叠加起来得到本层的第一个特征映射图。网络最后的传统神经网络是两个全连接层。这种全连接层的输入是最后的卷积层输出的特征映射图，一般情况是将最后所有的特征映射图的每个像素顺序排列成一列，输入给全连接层。而LeNet-5最后一层卷积层稍微特殊，其最后一个卷积层输出的特征映射图都只有一个像素。网络的输出是一个10元向量，通过选取10个元素中最大或最小的一个值来确定网络识别的结果。

表2- 1特征映射图连接方式

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | √ |  |  |  | √ | √ | √ |  |  | √ | √ | √ | √ |  | √ | √ |
| 1 | √ | √ |  |  |  | √ | √ | √ |  |  | √ | √ | √ | √ |  | √ |
| 2 | √ | √ | √ |  |  |  | √ | √ | √ |  |  | √ |  | √ | √ | √ |
| 3 |  | √ | √ | √ |  |  | √ | √ | √ | √ |  |  | √ |  | √ | √ |
| 4 |  |  | √ | √ | √ |  |  | √ | √ | √ | √ |  | √ | √ |  | √ |
| 5 |  |  |  | √ | √ | √ |  |  | √ | √ | √ | √ |  | √ | √ | √ |

《ImageNet Classification with deep convolutional neural networks》一文中设计的卷积神经网络在imagenet数据库上训练了一个卷积神经网络[2]。该网络有六千万个参数，六十五万个神经元，网络包含5个卷积层，其中的3个卷积层之后设置了最大值子采样层，网络最后有三个全连接层其中一个是一千类的softmax层。网络中使用的激活函数是ReLU函数。表格2-2详细地说明了这个网络各层的配置情况。下面结合表2- 2所示的各个卷积神经网络层，说明该网络的连接情况。

表2- 2卷积神经网络各层配置情况

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 层序号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 层名称 | 卷积1 | 卷积2 | 卷积3 | 卷积4 | 卷积5 | 全连接6 | 全连接7 | 全连接8 |
| 输入数据 | 224\*224像素\*3（RGB）图像 | 48\*2个55\*55尺寸的特征映射图 | 128\*2个27\*27尺寸的特征映射图 | 192\*2个13\*13尺寸的特征映射图 | 192\*2个13\*13尺寸的特征映射图 | 4096个数据 | 4096个数据 | 4096个数据 |
| 操作1 | 卷积（核尺寸：11\*11\*3，步长：4，共96个） | 卷积（核尺寸：5\*5\*48，步长：1，共256个） | 卷积（核尺寸：3\*3\*128，步长：2，共384个） | 卷积（核尺寸：3\*3\*192，步长：1，共384个） | 卷积（核尺寸：3\*3\*192，步长：1，共256个） | 点乘 | 点乘 | 点乘（softmax） |
| 操作2 | ReLU | ReLU | ReLU | ReLU | ReLU | ReLU | ReLU | ReLU |
| 操作3 | 最大值子采样 | 最大值子采样 | — | — | 最大值子采样 | — | — | — |
| 操作4 | 归一化 | 归一化 | — | — | — | — | — | — |
| 中间数据 | 48\*2个55\*55尺寸的特征映射图 | 128\*2个27\*27尺寸的特征映射图 | 192\*2个13\*13尺寸的特征映射图 | 192\*2个13\*13尺寸的特征映射图 | 4096个数据 | 4096个数据 | 4096个数据 | 1000个识别结果 |

第一层224\*224像素\*3（RGB）图像通过卷积1层（该层包含最大值子采样层）之后分为两组，每组输出48个特征映射图。第二层将前层的两组输出分别通过卷积2层（该层包含最大值子采样层），各组之内的卷积是全连接的，组之间无连接，每组输出128个特征映射图。第三层将前两组输出全连接地通过卷积3层，无视分组，卷积全连接，输出则仍然分为两组，每组192个特征映射图。第四层和第五层都分组做卷积，组内全连接，组间无连接，在第五层最后通过一个最大值子采样层，并将两组输出合并，重新排列成4096个元素的列向量。第六层和第七层是全连接层，不分组，每层都输出4096个元素的列向量。第八层是softmax全连接层，输入前一层的4096个元素，输出1000个元素的分类识别结果。这个网络在LSVRC-2010 contest测试集上得到了前五选错误率17%的结果，即网络输出1000个标签数据中最大的5个数值的标签包含了数据真实的标签，则为前五选正确。这个测试集包含了一百二十万张图片，共分为一千类。

## 2.4本章小结

本章以简要介绍卷积神经网络的产生与发展为开端，从卷积神经网络的三个基本特点入手，详细介绍与之对应的三个基本运算：“二维卷积”、“非线性激活函数”和“子采样运算”，最后本章通过浅层网络和深层网络两个实例，具体介绍了三种基本运算在卷积神经网络算法中是如何协同工作的。本章对卷积神经网络算法做了有针对性的介绍，直观地展现出了该算法密集的运算量，为本文后续介绍使用FPGA实现这种计算密集型算法做了充分铺垫。

# 第三章 使用FPGA实现卷积神经网络

## 3.1 FPGA的基本情况

FPGA是现场可编程门阵列的英文缩写，是最常用的可编程逻辑器件之一。随着集成电路工艺技术的飞速发展，FPGA芯片在资源，架构、性能、使用方法和应用场景方面都日益成熟，近年来，FPGA厂商又将处理器软核[18]、ARM硬核[19]等通用嵌入式处理器融入FPGA架构，这使得FPGA芯片在当今芯片制造，系统设计和算法加速等领域有着巨大的实用价值。

理论上讲，FPGA可以用于任何计算领域，其内部可以集成或直接实现微处理器就是最直接的证明。而FPGA的真正优势在于，针对某些特定的应用场合，它们的运算速度会非常快，因为它们本身具有大量逻辑门，以及其并行性的本质。具体的FPGA应用场合有： ASIC原型设计[20][21]，计算机视觉[22][23]，医学影像[24][25]等。FPGA的另一重要应用趋势就是对算法的硬件加速。这种加速是指设计者可以使用FPGA来加速某一算法的某些具体步骤，或者是将这种高性能的运算模块在FPGA和通用处理器之间实现共享。特别是在图像特诊提取算法方面，使用FPGA加速HoG特征[26][27]和SURF特征[28][29]都能取得较好的性能。

从另一个角度讲，FPGA还较多地适用于小产量的纵向应用。因为对于小产量应用，每个可编程芯片的单位硬件成本是比生产专用芯片却无法形成市场规模时的单位硬件成本更低。而且，如今的成本与性能优势，正在不断地拓宽FPGA的应用领域，一些针对低端应用市场的低成本FPGA产品也不断涌现。

## 3.2 FPGA实现卷积神经网络的优势

### 3.3.1卷积神经网络计算的并行特点

从以上的介绍我们可以总结出，卷积神经网络的计算由大量的二维图像卷积构成。分析整个网络的数据流和运算特点可以将卷积神经网络涉及的并行运算分为一下四类：位计算并行，卷积运算内的并行，同一层中各卷积运算之间的并行，各层之间的并行。文献[30]中，提出了五层并行层次结构，详细区分了神经元节点之间的并行结构与层内并行结构。其神经元节点之间的并行结构主要针对网络输出层的传统全连接层中的各个节点，提出这类数量众多的节点对网络并行性有至关重要的影响。然而本文认为，网络输出层的传统全连接网络运算方式与卷积层的卷积运算有极大的相似之处，完全可以通过控制卷积器的输入，几乎不加修改地将卷积器移植到全连接层，完成全连接层的各种运算。因此在FPGA实现过程中，并不需要专门区分全连接层的并行结构，该层的并行性，可以归纳到卷积运算内的并行结构这一层次。由于卷积神经网络运算规模都相对较大，在现有的软硬件条件下，是不可能完全并行实现，选择合适的并行层次结构和并行规模，设计并行运算单元，才能在FPGA等嵌入式平台上实现卷积神经网络运算加速，并且得到比较优秀的性能。下面就从上述四类并行层次入手，详细讨论各类并行层次的计算特点。

（一）位计算的并行。由于FPGA是数字化的信号处理方式，运算中各个操作数都是二进制表示，通过分析各操作数本身的二进制表示，可以从位运算的角度实现并行计算。具体来讲，就是设计性能更高的数字信号处理单元，如多位全加器、乘法器等。也就是说，这个层面的并行化属于数字电路底层设计的范畴。由于现代化的FPGA中大量使用嵌入式DSP单元实现高性能的多位全加和乘法，该层面的并行化不是本文讨论的重点。

（二）卷积运算内的并行。一个典型的二维卷积运算由图3- 1所示，卷积核扫过被卷积的图片，3\*3大小的卷积核内的每一个点，与图像中被卷积核覆盖的3\*3大小的范围内的每一个像素分别做乘法，并将9个乘法结果加起来得到一个卷积输出，图中*x1=1*，*x0=0*。由此可以看出，这就个乘法相互独立，可以同时进行，这是卷积运算内部的并行实现依据。

|  |
| --- |
|  |

图3- 1二维卷积运算示意图

（三）同一层中各卷积运算之间的并行。由前文所述，在同一卷积层内，可能存在着两种卷积情况：1）同一个输入图片被不同的卷积核卷积得到不同的输出特征映射图；2）不同的输入图片（前一层输出的多个特征映射图）被相同的卷积核卷积，并将结果相叠加。这样两类层内卷积运算都是相互独立的，这里将得到一个特征映射图的整个过程看作一个卷积运算，则每个卷积运算顺序的改变，只影响中间数据缓存的形式，不影响最终结果。因此，这些卷积运算也是可以同时进行的，这便是同一层中各卷积运算之间的并行计算依据。

（四）层间的并行。一般来说，卷积神经网络是一种多层的网络结构，各层在结构上也有较大的相似性。但是，由于整个网络中的数据是层层递进地传播和处理，前后层之间的运算有很强的相关性，因此层间计算的并行性很低，基本不存在前后层并行处理的可能。然而，由于训练好的网络参数在各层之间基本没有任何相关性，层间的并行主要可以体现在系统初始化时装载网络参数的阶段，网络各层的各种参数可以同时并行加载。

### 3.3.2FPGA与卷积神经网络基本运算相似的阵列结构

目前FPGA基本逻辑单元的结构主要由一个多输入（四输入或六输入）查找表，一个可编程寄存器和一些可编程互连线组成，用于实现简单的组合逻辑或者比特的寄存。Xilinx公司的FPGA中的Slice比上述基本逻辑单元稍加复杂，由两个4输入的函数、进位逻辑、算术逻辑、存储逻辑和函数复用器组成。算术逻辑包括一个异或门（XORGate）和一个专用与门（AND Gate），一个异或门可以使一个Slice实现2bit全加操作，专用与门用于提高乘法器的效率；进位逻辑由专用进位链和函数复用器组成，用于实现快速的算术加减法操作ALTERA公司的FPGA在基本逻辑单元的基础上，用16个逻辑单元配合以内部控制信号和互联同路，组成逻辑阵列，以实现稍微复杂的逻辑功能。逻辑单元结构图如图3- 2所示。

|  |
| --- |
|  |

图3- 2逻辑单元结构图

通过对卷积神经网络运算结构的进一步分析，可以清楚地发现，卷积神经网络的运算由大量以“乘-累加”为基本单元的神经元运算组成，每个单元所做的运算操作高度重复，且同一个卷积层内的各个“乘-累加”运算相互独立，可以同时进行，不分先后。这种由大量重复基本单元组成复杂系统的结构，与FPGA中以逻辑单元阵列和可编程互连线组成复杂数字系统的结构高度相似，因此可以合理地推断，FPGA芯片适合于实现卷积神经网络，由FPGA实现的卷积神经网络能够充分结合两者的并行特性，发挥并行优势。

从上述对FPGA基本逻辑单元结构和规模的描述中，可以得出结论，一般情况下，为了保证较高的运算精度，一个卷积神经网络神经元运算单元是难以用单个逻辑单元甚至Slice或者逻辑阵列实现的。而当今的FPGA厂商为了提高其产品对大规模乘法、乘-加运算的性能，以便在数字信号处理领域有更高的竞争力，普遍将一定规模的多位乘法器以及专用的DSP单元以硬核的方式，嵌入到FPGA芯片中，构成数字信号处理单元。以ALTERA的Stratix IV为例，该系列FPGA内拥有多达1360个18\*18乘法器，其DSP单元可以完成高分辨率视频处理，数字上、下变频和多采样率滤波等众多数字信号处理功能。FPGA内嵌DSP单元的结构图如图3- 3所示。而这种高性能乘法、乘-加功能的实现，正是人工神经网络特别是卷积神经网络运算最需要的。从FPGA的内部结构看，这些内嵌的硬件乘法器仍然是以整列方式，排列在整个芯片中，能够方便地实现与片内其他逻辑资源的互联。因此，围绕着FPGA中的硬件乘法器来设计卷积神经网络的神经元和网络层次结构，规避了当前FPGA产品中单个逻辑单元或Slice资源不足以完成卷积神经网络神经元运算的瓶颈，在运算精度和网络实现的并行度上取得了很好的折衷。

|  |
| --- |
|  |

图3- 3 FPGA内嵌DSP单元结构示意图

## 3.3 本章小结

本章详细介绍了FPGA的产生、发展以及FPGA芯片的基本架构，内部逻辑、运算单元的基本形式。接着具体分析了卷积神经网络运算的并行特性，并根据这一特性，结合FPGA芯片中阵列排布的DSP单元，提出围绕DSP单元设计卷积神经网络基本模块的构思。同时，考虑到设计要求中规模、性能、功耗等限制因素，本文的卷积神经网络各层模块设计中，均讨论多种设计方案，并比较各方案在性能、规模等方面的优劣，根据具体设计要求，折衷挑选设计方案。

# 第四章 基于FPGA的卷积神经网络关键电路模块的设计

本章介绍基于FPGA的卷积神经网络中，关键电路模块的设计细节。在前两章的分析中，本文已经阐述了卷积神经网络中存在大量可以并行执行的重复、独立的基本运算。其中，同层内的各个卷积运算是可以并行执行的，具体来讲，就是一层的运算中，从各个输入特征映射图到每个输出特征映射图的运算都是独立、可并行的。在这个层次，本文抽象出“从特征映射图到特征映射图”的运算，即“从一个输入特征映射图到一个输出特征映射图”的运算是独立重复的，改变卷积核的参数，即可完成同一层内其他的“从特征映射图到特征映射图”的运算；而若将输入输出特征映射图的尺寸参数也做改变，就是其他层中“从特征映射图到特征映射图”的运算。因此，这个层次的运算独立性和重复性相对其他层次运算更强，更适合设计为FPGA电路的基本运算单元模块。

进一步地，实现“从特征映射图到特征映射图”的运算，就是需要实现卷积神经网络的三个最基本运算：卷积、非线性激活函数和子采样，其中每个基本运算作为一个子模块，其内部也有相应层次的并行特性，本章下面的部分，就详细分析基本运算内的并行特性，并设计子模块的具体FPGA电路。

## 4.1卷积器

卷积神经网络涉及的一个核心运算是图像的卷积运算，图像卷积运算将图像处理和传统神经网络巧妙地结合起来。卷积神经网络正是借助卷积核和卷积运算实现了感受野的概念。同时，也正是借助卷积核和卷积运算，实现了卷积神经网络权值共享，减少了网络参数数量的。如公式（4-1）所示的图像卷积运算即二维卷积运算，在数字图像处理领域有着非常广泛和成熟的应用。其为计算第*i*行第*j*列的卷积结果，*x*为输入图像，卷积核尺寸为*K­­2*大小。

|  |  |
| --- | --- |
|  | 公式（4-1） |

图像的空间域滤波处理就大量地使用到了二维卷积运算，通过调整卷积核的内容，二维卷积运算可以实现图像的高、低通滤波，边缘检测，形态学滤波等多种实用的处理。而在卷积神经网络中，多种卷积运算的本质也是对输入图像的各种空间域滤波和边缘检测，从而实现对图像数据中提取多种特征，以便网络最后的全连接层实现对图像的分类识别。

在卷积神经网络中，大量的二维卷积运算成为整个识别算法中运算最为密集的部分。二维卷积运算虽然在数学上非常简单，但是对于硬件的实现，则有非常多的内容需要考虑，才能以较高的效率完成这种简单的运算。这些需要考虑的内容包括前端数据流馈入的形式（这种形式包括数据流量，数据格式和数据流时序等），数据处理对运算时间和硬件资源的要求等。根据实际情况，具体分析上述条件，是设计高效硬件卷积器的必然要求。下面将结合本文提出的卷积神经网络应用芯片，从数据流的缓存和卷积运算两个角度详细地讨论卷积器的设计与实现。

如前文公式所示，在计算一个像素点的卷积结果*Ym,n*时，计算单元需要获得输入像素点*Xm,n*以及它周围*K2-1*邻域的*K2*个数据，然后与卷积核中的*K2*个权值对应相乘再累加。首先考虑系统前端输入数据是以数据流形式给进的，即像素数据从前端缓存中只能每个时钟一个像素按照从左到右从上到下的顺序逐个流入，要同时获得一个卷积窗口中的*K2*个数据，必须对输入数据进行缓存，通过如图4- 1的行缓存结构，在寄存器id0~id8上面，可以获得一个大小为3\*3的卷积窗口的图像数据。不考虑图像边缘的情况下，在初始等待*(K-1)\*M+K*个时钟之后，卷积窗口可以每个时钟向后（从左到右，从上到下）移动一个像素的位置，从而实现每个时钟都能得到一个新的卷积窗口的全部数据。这里*M*为输入图像的列数。这种情况下，后续的卷积运算单元（乘-累加单元）若能在一个时钟内完成*K2*个乘法和一个*K2*路并行加法，则可以实现每个时钟一个卷积结果的数据吞吐量，同时通过简单的数据使能标志，这个卷积层模块便可以高效地完成一张特征映射图的运算。通过对权值寄存器的更新或者对该卷积层模块的复制，即可串行或者并行地计算一个输入图像的多个输出特征映射图。这样的卷积层架构对输入数据的带宽要求最低，但有着在此限制下最高的并行性能。其主要缺点在于对硬件资源的大量需求，随着卷积核尺寸的增大，资源消耗呈平方关系增长，同时，大量的并行数据流对芯片内时序同步的要求大幅提高，极大地限制了系统时钟的频率，不利于硬件并行优势的充分发挥，反而凸显了并行数据流所带来的缺陷。

|  |
| --- |
|  |

图4- 1行缓存结构

为了避免上述并行数据流时序同步问题，本节提出串行-并行结合的设计思路。在计算卷积核尺寸为*K2*的卷积运算时，如图4- 2将乘法运算分成等量的*K*组，每组的*K*个乘法和加法运算通过乘-累加（MAC）单元串行完成，最后将*K*组乘-累加结果并列相加。该模块数据时序的具体设计如图4- 3所示。次方案通过牺牲一定的并行程度，换取了硬件资源的节省和时序要求的降低，进而提升了系统时钟的频率限制，在不考虑其他要求的条件下，可以通过提高系统时钟，达到更高的性能。在实现该串行-并行结合方案的时候，通过对卷积运算模块时钟频率的*K*倍频，即可保证输入、输出数据流的速度不变。而实际情况下，由于完成卷积运算后需要将乘-累加结果代入激活函数，实际倍频频率会略大于*K*倍。

|  |
| --- |
|  |

图4- 2串行、并行相结合的卷积模块核心架构

|  |
| --- |
|  |
|  |

图4- 3串行-并行相结合的卷积模块数据时序设计

在实际应用中，由于资源成本的更严重限制等原因，也可能需要进一步牺牲并行特性，将*K2*个乘法和加法运算用如图4- 4所示的一个乘-累加单元架构，完全串行地实现，这种情况下，就需要对卷积运算模块的时钟做大于等于*K2*倍的倍频。而多个输出特征映射图的计算，仍然可以通过更新权值寄存器或复制该卷积层模块来按需求实现。根据面积-速度的折衷规律，灵活地组合完全并行卷积模块、串行-并行结合卷积模块和完全串行卷积模块，便可以设计出最符合实际需求的卷积层模块，灵活按需地解决嵌入式平台上卷积神经网络芯片的设计难题。

|  |
| --- |
|  |

图4- 4乘累加（MAC）模块

另外，文献[31]中提出的流水线全并行卷积运算单元如图4- 5所示，则巧妙地将缓存结构和乘-累加运算单元相结合，避免了对*K2*路数据做并列加法，使用*K2*个乘法器和*K2*个加法器，规避了大量并行数据流时序同步难的问题。在如今FPGA内部计算单元数量较多的普遍现实下，这成为一个非常高效的并行卷积方案。

|  |
| --- |
|  |

图4- 5流水线全并行卷积运算模块

最后，如果前端输入的数据流可以同时*K*路流入，则可以将二维卷积运算转化为*K*个一维卷积，该情况类似与上述串行-并行结合的卷积架构，只是此时前端同时流入*K*路数据，不需要缓存*(K-1)\*M+K*个像素，既可完成一个窗口的卷积运算，然而这种方案对数据流带宽要求很高，而且带框上还存在浪费。另外，文献[32]还提出多窗口部分缓存技术，通过对缓存数据的重用，缓解带宽压力，对大尺寸图像的卷积运算有较大意义。

## 4.2激活函数

非线性的激活函数在各类人工神经网络中被普遍应用，将非线性因素引入人工神经网络，使得人工神经网络能够对被识别物体在输入图像中位置的变化进行一定程度的适应，以便网络能够适用于更多的场合，更好地解决各种实际问题，其中Sigmoid激活函数被广泛应用于各种神经网络中。Sigmoid函数是一个单极性激活函数，其表达式为公式（4-2）。

|  |  |
| --- | --- |
|  | 公式（4-2） |

函数图像如图4- 6所示。作为一个典型的非线性函数，Sigmoid函数在数字电路中的实现成本是很高的[33]，因为函数运算中涉及一个指数运算和一个除法运算。在数字电路硬件上实现这些非线性函数的方法主要有如下几种方案：查找表实现、比特映射实现、分段线性拟合和泰勒展开近似，以及上述方法的混合实现。本节将详细说明Sigmoid函数的两种数字电路设计实现方式，它们分别采用分段线性拟合与查找表相结合的方案和复用式分段线性拟合方案。

|  |
| --- |
|  |

图4- 6 Sigmoid函数图像

观察Sigmoid函数图像我们发现如下两个事实：1）函数图像关于点(0,0.5)对称，即*f(x)=1-f(-x)*。2）函数图像在自变量*x*接近0的位置比较直，近似于一条过点(0,0.5)，斜率为0.25的直线，而在自变量稍微远离0的位置弧度比较明显，当自变量再大一些的时候，函数图像比较平，接近于*f(x)=1*或*f(x)=0*。由此我们可以得出结论，可以只计算自变量大于零的部分，小于零的部分可以很容易通过对称关系得到；在函数输入不同自变量的时候，应该采取不同的实现方案，从而达到最优的性能。这就是分段线性拟合与查找表相结合设计方案的理论基础。

在函数图像中线性度较好的区域采用线性拟合，在线性度相对较低的区域采用查找表实现，可以充分节省逻辑资源，提高精度。根据图像形状，这里将自变量大于零的部分分为三段：0≤x<1和x≥8，这两段图像接近直线，用线性拟合可以直接将自变量简单地与固定的斜率相乘得到输出或者直接得到输出；1≤x<8，这一段图像弧度明显，采用查找表实现可以避免计算多段拟合直线，耗费过多逻辑资源用于乘、加运算和比较运算。具体设计思路如下：

1. 将输入的二进制补码表示的数据转换成原码，即求其绝对值。
2. 用两个比较器，将输入分为三类，即上述三个分段。
3. 将绝对值数据乘以固定斜率，同时也将绝对值数据输入查找表，得到两个候选结果。
4. 根据前一步的分类（分段）结果，选择候选结果作为中间结果。
5. 用公式*f(x)=1-f(-x)*换算中间结果，得到第二步候选结果。
6. 根据原始数据的符号选择最终的输出结果。
7. 该原理的伪代码和设计框图则如图4- 7所示。

对于复用式分段线性拟合，因为段拟合都是使用二维直线，二维直线可以用一个乘法和一个有符号加法来实现，若在每段拟合运算中使用同一个硬件和相应不同的斜率、截距参数，则可以只用一个乘法器、一个加法器、一个多路选择器、一个编码器和若干比较器实现，实现精度要求从两方面影响资源使用情况：一，输入、输出的量化位数影响每一个运算器的规模；二，分段数量只影响多路选择器、编码器的规模和比较器的数量。分段数和各运算器数量关系如表4- 1所示。该设计模块框图如图4- 8所示。

表4- 1复用式分段线性拟合方案中分段数和各运算器数量关系

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 分段数 | 多路选择器输入个数 | 多路选择器选择端位数个数 | 编码器输入比特数 | 编码器输出比特数 | 比较器个数 |
| N | N | log2(N) | N | log2(N) | N |

**输入**

|  |
| --- |
| **斜率k（第一段线性拟合部分）**  **（不可能）**  **符号位**  **符号位**  **结果**  **转成负数**  **查找表** |

图4- 7分段线性拟合与查找表相结合的模块设计

|  |
| --- |
| *载入查找表**;*  *求输入P的绝对值;*  *if(输入的绝对值< 1)*  *中间结果 = 0.24 \* 输入的绝对值 + 0.5;*  *else*  *if(输入的绝对值< 8)*  *中间结果 = 非线性查找表（输入的绝对值）;*  *else*  *中间结果 = 1;*  *end*  *end*  *if(输入为负) X=1-Xi;*  *else X=Xi;*  *end* |

图4- 8分段线性拟合与查找表相结合设计方案的伪代码

## 4.3子采样器

本文设计的卷积神经网络的子采样层采用加和子采样，采样区域为2\*2像素，区域不重叠，即在横、纵方向的步长都为2。加和子采样的实际意义与平均值子采样类似，就是对感兴趣点的邻域计算均值，经过子采样后，降低了输出特征映射图的分辨率，也就降低了卷积神经网络对待识别的物体在输入图片中位置变化的敏感度，让卷积神经网络具有了一定程度的抗输入图形畸变的能力。

子采样的核心在于获取图像中一个2\*2的图像区域，由于整个运算过程中，像素数据是以数据流的方式流入，获取这个2\*2图像区域的像素数据时就遇到了与卷积运算相同的难题。当然，这里也可以采用行缓存的方案予以解决，要获得一个2\*2的图像区域，只需要缓存一行的图像数据即可，但是由于在子采样过程中，这个2\*2的图像区域，即窗口，移动的步进长度不再是1，为了不重叠地获取2\*2的图像区域，这个步进长度是2，这时通过行缓存取得的运算窗口还需要后续的使能模块，抽样选取图像窗口运算得到的结果，这就不免增加了系统的复杂性，延长设计周期。

本文将采用一次换位全缓存的方法，实现子采样的运算。由于子采样的输入特征映射图尺寸最大的情况也只有24\*24像素，全部缓存这576个数据也不会花费过多时间和存储资源，那么，将这些数据按照4个一组，每组恰好是待采样的2\*2图像窗口，按照窗口的顺序缓存，之后再顺序读取这些按照窗口顺序存储的数据，则恰好每4个数据就可以运算得到一个输入特征映射图中2\*2窗口的数据，再进行子采样操作就非常便捷。这种换位缓存方案的示意图如图4- 9所示。

|  |
| --- |
|  |

图4- 9换位缓存原理示意图

由于我们知道输入的数据是按照从左到右从上到下的顺序，逐个像素依次流入的数据流，要实现换位缓存，则需要将按照上述顺序流入的数据放在缓存单元不连续的相应位置，这里只需要将每个数据存入的位置存入一张地址查找表，按照表中的地址，存入缓存单元，再按照地址递增顺序从缓存单元中读出缓存数据，此时的数据便已经是按照窗口顺序，每组4个的待采样数据。这时，只需要对读出的数据每4个累加便可得到加和子采样，累加结果舍去末尾两个比特，即右移两位，便得到均值子采样，在这4个值中选取一个最大值，便得到最大值子采样。该电路的具体模块框图如图4- 10所示。

|  |
| --- |
| **Reg**  **均值（加和）子采样**  **最大值子采样**  **寄存器控制信号**  **Reg**  **换位缓存地址**  **输入像素流**  **换位地址**  **查找表**  **换位**  **缓存**  **单元**  **输入像素**  **计数器** |

图4- 10换位缓存子采样模块框图

## 4.4本章小结

本章内容对本文提出并设计的卷积神经网络的关键FPGA模块设计进行了详细的叙述。根据未来应用时，实际采用的硬件系统对串行、并行程度的不同需求，本章所讨论的每个模块都介绍了多种设计方案，并分析了每种方案的硬件资源消耗，运算用时，模块的时序约束要求和实际综合结果以及设计难度等方面的内容，同时也简要介绍了这些因素对整个系统的影响。

本章所叙述的详细设计方案，都以模块设计为核心思路。这是考虑到，在卷积神经网络的某一层中，从输入特征映射图到输出特征映射图的运算具有完整性和独立性，整个卷积神经网络中，从输入的图像抽象到待分类特征这个主要过程就可以分解为若干个通过输入特征映射图求输出特征映射图的运算。这种从特征映射图到特征映射图的基本运算的并列（层内）、级联（层间）组合便构成了整个卷积神经网络的最主要部分。

# 第五章 基于FPGA卷积神经网络的系统设计

如上文所介绍的，整个卷积神经网络就是由若干个“从输入特征映射图到输出特征映射图”的运算构成，联合本文第四章中设计的三个关键模块，组成基本运算模块，将这些基本运算模块的并列（层内）、重用或级联（层间），即可组成整个卷积神经网络。

## 5.1 层内运算并行子系统

卷积神经网络层内运算是将上一层的输出特征映射图作为本层的输入特征映射图，经过若干个卷积、激活、子采样等运算得到本层的相应数量的输出特征映射图。这些层内运算是相互独立的，因此可以通过复制基本运算模块，并行地完成层内的运算。在输入层这种，并行卷积运算少，图像尺寸较大，执行一次运算耗时较长的层次中，复制较少的基本运算模块并行完成计算，并不会过多地增大硬件资源使用量，却能够显著缩短运算耗时。此类并行系统架构如图5- 1所示。

|  |
| --- |
| 输入像素流  结果输出  控制逻辑  结果缓存  权值配置逻辑  基本运算模块  基本运算模块  基本运算模块  基本运算模块 |

图5- 1层内并行运算系统架构

## 5.2 层内运算串行子系统

卷积神经网络中间若干层输出的特征映射图数量较多，而此时输入特征映射图经过之前子采样层的处理，其尺寸已经较小，经过为数不多的时钟周期就能完成运算。虽然这些层的运算依然相互独立，此时需要大量复制基本运算单元才能完全实现并行运算，然而，由于此时执行一次输出特征映射图运算所需的时间已经较短，完全并行运算取得的加速效果并不显著。因此，在这个层次的运算中，采取基本运算模块复用，并通过时序控制模块协调模块间的时序实现系统是合理的选择。此串行系统架构如图5- 2。

|  |
| --- |
| 输入像素流  输出数据  输出缓存  时序控制逻辑  权值配置逻辑  基本运算模块 |

图5- 2层内串行运算系统架构

## 5.3 层间运算串行系统

卷积神经网络层间运算前后相关，无法并行执行，必须在得到前一层的特征映射图全部得到之后才能计算本层的特征映射图。因此要完成整个卷积神经网络的运算，需要级联各层的基本运算模块，统一协调各模块时序。基于FPGA的卷积神经网络系统架构如图5- 3所示。

|  |
| --- |
| 串行子系统  并行  子系统  输入图片  输出特征 |

图5- 3 FPGA卷积神经网络系统架构

## 5.4 本章小结

本章从层内并行和串行运算以及层间串行运算的特点出发，设计了相应的并行、串行子系统，最后结合各个子系统，设计了完整网络的多层并行、串行结合架构。

# 第六章 关键模块和系统的验证与测试

模块和系统的验证和测试是基于FPGA芯片设计中重要部分，本章内容将对所设计的几个关键模块和完整FPGA卷积神经网络系统的功能和性能进行验证与测试。以静态手势识别网络为例，验证与测试基于FPGA的卷积神经网络设计结果，说明围绕本文提出的“从特征映射图到特征映射图”的基本运算单元设计具备良好的运算精度和并行加速能力，且功耗较低，同时可以灵活地重用于卷积神经网络的各个运算层中。

本章中，用于验证和测试本文所设计的基于FPGA的卷积神经网络基本运算模块的实例是一个用于静态手势识别的三层卷积神经网络。在静态手势识别问题中，静态手势图像经过简单的预处理，如肤色提取、尺度缩放和动态范围归一化等，得到的图像形式类似MNIST手写数字训练库中的手写数据图片。本文选择静态手势识别问题作为切入点测试基于FPGA的卷积神经网络，是因为静态手势识别在非接触式人机交互系统[34]中有广泛的应用，而且非接触式人机交互的典型应用场景就是以小尺寸、低功耗平台为主的智能终端等。因此本文设计的网络类似与LeNet-5。为了适应硬件串行、并行组合实现，本文将静态手势识别网络的输入设计为28\*28像素的静态手势图片，第一层通过核尺寸为5\*5，步进为1的卷积得到6个特征映射图，经过2\*2尺寸、步进为2的加和减采样后，送入网络的下一个卷积层。第二个卷积层接受上一层的6个12\*12像素的特征映射图，通过核尺寸为5\*5，步进为1的卷积，得到12个特征映射图，这里6个输入到12个输出是全连接的，其组合方式可以从图6- 1的伪代码中直观理解。第二个卷积层输出的特征映射图经过2\*2尺寸、步进为2的加和减采样后得到12个4\*4尺寸的特征向量，将这些特征向量中的元素全部送入全连接的softmax网络，进行分类，得到最终的识别结果。

|  |
| --- |
| for j = 1 : 输出图个数  初始化中间变量 z  for i = 1 : 输入图个数  z = z + 第i个输入图（卷积）第j个卷积核;  end  第j个输出图 = Sigmoid（z + 第j个偏置）  end |

图6- 1中间卷积层输入特征映射图到输出特征映射图的伪代码

## 6.1 卷积器的验证与测试

本文上一章中对卷积运算单元的详细分析，已经将多种硬件架构展现出来，本节根据上述讨论，验证并测试串行-并行结合的卷积运算单元。该单元以一路数据流形式输入每像素8bit的图像数据，卷积核尺寸为5\*5，权值数据用16bit定点数表示，1位符号位，15bit小数位，用于量化(-1,1)的权值数据。输出数据为29比特定点数，1位符号位，5比特整数，23bit小数位。设计中关键的乘-累加运算单元均使用FPGA中内嵌的DSP模块完成。使用Block RAM组成行缓存，缓存输入数据流，实现卷积窗口的提取，当存在同一数据源，需要1-to-n卷积时，该行缓存单元不需要复制，是共用的，只需要增加得到窗口数据的寄存器扇出。根据综合结果，模块使用硬件资源情况如表6- 1所示。根据时序分析软件给出的结果，该模块最大时钟频率约为100MHz。对该模块时序约束后，时序分析工具根据布局布线结果，分析得到模块实际可以以最高频率107.19MHz运行。

表6- 1卷积运算模块资源使用量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | DSPs | 组合逻辑资源 | 寄存器资源 | Block RAM |
| 一个卷积运算单元 | 2 | 1177 | 170 | 672 bits |

本节还将介绍根据实际应用需求，对上述设计的卷积运算单元的实际验证情况。本文设计的卷积神经网络输入层需要完成对28\*28像素的图片做5\*5的卷积，这是该网络中输入图像尺寸最大的卷积运算，在此框架下设计的测试单元可以按数据流形式将图像数据送入待验证的卷积运算单元，在FPGA开发板上实际验证卷积运算单元的设计结果。验证输入的是如图6- 2的静态手势测试图像数据，经过117个时钟初试等待，将会得到每个时钟一个像素的卷积输出结果。初始等待时间和卷积核尺寸、输入图像尺寸的关系如表6- 2所示。初始等待时长和卷积核尺寸和输入图像列数有关，具体关系由公式（6-1）所示。公式中*K*为卷积核宽度或高度（正方形卷积核），*N*为图像列数，*D*为初始等待时钟数。图6- 3显示了FPGA运算结果和MATLAB运算结果的对比，经计算，两者的归一化误差不到1%，带来这一误差的主要原因是对图像数据和权值数据的定点量化。

表6- 2初始等待时钟数与卷积核、图像尺寸的关系

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 卷积核尺寸3\*3 | 卷积核尺寸5\*5 | 卷积核尺寸7\*7 | | |
| 图像尺寸28\*28 | 59 | 117 | 155 | | |
| 图像尺寸12\*12 | 27 | 53 | 79 | | |
|  | | | | 公式（6-1） | | |
|  | | | | |

图6- 2输入卷积器的静态手势图像数据

|  |  |  |
| --- | --- | --- |
| a） | b） | c） |
| d） | e） | f） |

图6- 3 FPGA卷积结果与MATLAB卷积结果对比以及这两者之间的残差，a）为输入图像与1号卷积核在FPGA上的运算结果，b）为输入图像与1号卷积核在MATLAB上的运算结果，c）为a）和b）之间的残差；d）为输入图像与2号卷积核在FPGA上的运算结果，e）为输入图像与2号卷积核在MATLAB上的运算结果，f）为d）和e）之间的残差

## 6.2 激活函数的验证与测试

### 6.2.1 激活函数模块误差来源的理论分析

数字电路硬件实现非线性函数时需要考虑的一个非常重要的问题就是实现精度。对于卷积神经网络的应用来说，实现精度关系到识别正确性问题，也就是网络性能。精度太低必然影响网络的识别正确率，而过高的精度会增大硬件资源的耗费，同时也会提高系统的功耗。因此，在设计硬件电路之前，对实现误差做理论分析就非常必要。

在激活函数的实现中，若不考虑具体的实现细节，可以把该硬件系统看成对函数自变量和函数值的定点数量化。直观地可以将这个量化过程理解为将函数图像画入如图6- 4的网格中，用网格中离图像最接近的网格交点替代真实的函数值，且根据函数定义，要保证网格的每条纵格线上只取一个网格交点作为函数值的近似，这个过程即为自变量和函数值的量化。

|  |
| --- |
|  |

图6- 4非线性函数量化

前文所述的非线性函数的多种不同实现方案，便是用不同的方法来寻找网格中邻近真实函数曲线的网格交点来近似表达函数值，以实现量化的过程。如Sigmoid函数局部细节图6- 5所示，我们可以直观地看到这种量化过程是如何带来误差的，由于用网格交点近似实际函数值，我们可以看到，在横轴[2.48，2.59)区间上，真是的函数值随着自变量的增大而略微增大，但是由于使用了网格交点来近似真实的函数值，在整个输入区间，无论自变量具体是多少，都被量化为数值*2.48*，而输出都被量化为0.92。因此这个区间里就有两方面误差：一），输入无论是多少，都被当成输入是*2.48*；二），输入即使恰好就是2.48时，输出也不是真实的*f(2.48)*，而是其近似值*0.92*。

通过上述分析，我们了解了误差的具体来源。现在，我们对分段线性拟合与查找表结合的设计方案进行MATLAB仿真和FPGA开发板的实际验证，分析在实际实现的过程中，误差的具体大小和表现形式。对分段线性拟合与查找表相结合的方案，MATLAB仿真从理论上揭示了误差的表现形式：在线性拟合段，误差由真实函数值和拟合直线之间的位置差异带来；在查找表段，误差由查找表对函数自变量和函数值的量化误差带来。在FPGA实现中，本文采用非对称量化方式，以输出数据为基准，反向定位输入数据，这样量化的好处是，在函数图像非线性程度高的地方，对输入量化较细；在函数图像非线性程度较低的地方，对输入量化较粗。这种方案能在有限的量化位数的条件下得到更低的量化误差。这里输入数据和输出都是16位定点数量化，输入数据格式为S3Q12，即1比特符号位、3比特整数位、12比特小数位，表示-8到8的数值，输出数据格式为Q16，即16比特全用于表示小数位，表示0到1的数值。

|  |
| --- |
|  |

图6- 5 Sigmoid函数量化局部细节图

### 6.2.2 激活函数模块的实现情况和性能

将分段线性拟合与查找表结合的设计方案以verilogHDL建模，综合后资源使用情况如表6- 3所示，该设计使用极少的逻辑资源，存储资源与精度需求相关。同时，该模块组合逻辑运算量较少，数据位数最高为16位，则按照主时钟200MHz运行要求，进行时序约束。时序分析结果，系统最高时钟频率215.19MHz，关键路径时序余量0.353ns。为了预估模块在芯片上真实运行的效果，FPGA开发软件为开发者提供了一个非常有用的逻辑分析工具：片上逻辑分析仪。利用FPGA芯片中的存储器单元，开发软件可以将逻辑分析仪嵌入FPGA芯片，采集模块在芯片中实际运行时的数据流。借助这个工具，开发者可以得到比仿真数据更真实的运行数据，极大地方便了开发工作。然而，由于需要模块实际运行，布局布线工作必不可少，因此使用片上逻辑分析仪进行调试会耗费更多开发时间，设计的仿真工作仍然是不可省略的。通过FPGA开发板对上述设计进行验证，片上逻辑分析仪采集到的Sigmoid模块对-8到8的输入的运算结果如图6- 6所示。将运算结果导入MATLAB与PC软件采用双精度浮点数运算结果相对比，以后者为基准，FPGA运算结果在各种输入情况下的相对误差均小于1%。其最大误差出现在线性拟合与查找表拟合的分界处。此处函数曲线非线性程度较高，在有限的量化位数的条件下，是误差最大的部分。这种随输入变化的误差情况，可以从图6- 7中直观理解。其中的公式中，表示向“零”取整。表6- 4表示了每段误差的理论值计算方式。

表6- 3分段线性拟合与查找表结合的设计方案资源使用情况

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | DSPs | 组合逻辑资源 | 寄存器资源 | Block RAM |
| 一个卷积运算单元 | 2 | 56 | 32 | 4096 bits |

|  |
| --- |
|  |

图6- 6片上逻辑分析仪采集的Sigmoid函数运算模块运算结果

|  |
| --- |
|  |

图6- 7随输入变化的误差

表6- 4误差理论值

|  |  |  |
| --- | --- | --- |
|  | 线性拟合段 | 查找表段 |
| 理论误差 |  |  |

### 6.2.3 激活函数与卷积器级联后的误差分析

在大多数卷积神经网络中，包括本文设计的静态手势识别网络，输入特征映射图与卷积核卷积后，一般都需要经过激活函数激活，也就是说，激活函数模块是级联在卷积器之后的。由于相互级联的两个模块都存在数据量化误差，级联之后有可能出现误差的积累，使得级联输出误差较大。而在卷积与激活函数级联的具体情况下，由于激活函数是将从-8到8的输入值映射到0到1的函数值，是一个数据动态范围压缩的过程。由于在量化位数一定的情况下，被量化的数据动态范围越小，量化精度越高，因此通过激活函数压缩数据动态范围，限制了误差的积累。经过对上述两个级联模块的FPGA运算结果与MATLAB运算结果相对比，实验得到的误差依然小于1%，没有因为模块级联、多次量化误差积累造成误差如蝴蝶效应般扩散。

## 6.3 子采样器的验证与测试

本文设计的换位缓存结构主要是对存储器的操作，从某种程度上讲，是随机的地址和数据操作，这就对整个模块的时序性能提出了较高的要求，时序约束也将更加复杂和严格。本模块主要占用存储器资源，按照系统时钟100MHz约束，综合后结果消耗存储器22144比特，模块中的逻辑控制单元会用到71个组合逻辑单元，输入输出寄存器和数据计数器等单元用到92个寄存器。最后运算最大值或累加的模块会用到少量的组合逻辑单元或者DSP单元。时序分析得到最大运行时钟277.32MHz，关键路径的建立时间余量为6.39ns。

该模块对存储器主要为随机读写操作，这对数据时序要求很高，其代码中不合理的内容，在综合时结果非常容易与预期效果有很大差异，其设计难度较高。因此，本节就从该模块的验证入手，介绍在FPGA设计中非常重要的部分——仿真。由于FPGA设计最终是设计数字逻辑电路，其关键步骤综合就是从硬件描述语言得到电路网表的过程。若需要对这个网表在FPGA芯片中的性能做验证，就需要进行布局布线。布局布线过程借助开发软件完成，但布局布线非常复杂，需要耗费大量的时间。为节省设计时间，优化设计流程，仿真就成了关键的一步。本节讨论的子采样器，以存储器操作为主，其测试向量分两部分：测试数据生成和子采样结果读取。测试向量采用verilogHDL语言编写，其中主要是不可综合的语句，这些语句对开发者来说直观以理解，同时仿真软件不需要将测试向量综合成电路网表，不可综合语句的使用是非常方便的。在子采样器仿真的过程中，测试数据以两种方式产生：顺序产生和随机产生，前者用于测试功能正确性，后者测试模块的鲁棒性。图6- 8的截图展示了子采样模块在测试软件中，对输入数据的响应，这与设计之初预期的时序关系一致，因此可以证明设计的正确性，设计工作可以正常推进到后面的步骤。

|  |
| --- |
|  |

图6- 8子采样器仿真时序图

## 6.4 系统验证与测试

作为一个面向小尺寸、低功耗平台的完整卷积神经网络芯片，在保证运算时间和硬件资源使用量合理的前提下，更关心芯片运算结果的正确性和系统功耗。表6- 5展示了一个“从特征映射图到特征映射图”的基本运算模块硬件资源使用量的情况。通过功耗分析工具，我们可以直观地得到各个模块和整个系统的功耗表现，单个模块1.05W的总功耗已经是一个可以接受的表现，而该功耗随着系统规模的增长而线性增加，实验得到的数据为每增加一个基本运算单元，功耗增加约7mW，因此若完全并行完成静态手势识别的78个“从特征映射图到特征映射图”的运算，系统功耗小于1.6W。而通常会在第二层卷积运算上采用串行-并行相结合的架构，因此功耗会更低。而由于功耗随系统规模的增加较少，采用更高的并行程度，会节约运算时间，进而节省整体运算能耗。

|  |
| --- |
| 图6- 4 |

表6- 5基本运算模块资源使用情况

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | DSPs | 组合逻辑资源 | 寄存器资源 | Block RAM |
| 一个基本运算单元 | 4 | 1233 | 202 | 19744bits |

## 6.5 本章小结

本章对前文所设计的各个关键电路模块和卷积神经网络FPGA完整系统进行了验证和测试，验证和测试工作采用FPGA设计常用的软件验证、仿真以及开发板验证测试等方法，得出了各个模块功能正确、性能良好的结果，证明了本文的设计工作的合理性和具体优势。

# 结 论

在集成电路制造工艺飞速发展的今天，以FPGA为代表的大规模可重构芯片也不断发展，其中的逻辑单元密度已经非常客观，同时还集成了大量的专用数字信号处理单元和通用处理器核，这些特点使得FPGA芯片在算法加速和系统级设计方面有这极大的潜力，成为芯片设计的重要辅助工具，甚至也可以直接作为成熟产品面向市场。卷积神经网络作为深度学习的最成功实际应用之一，始终处于学术研究的核心地位。对卷积神经网络的应用，使得机器学习和计算机视觉领域得到了长足的进步。越来越多的学术机构开发出卷积神经网络的实际应用。

学习了解目前卷积神经网络在计算机视觉领域的实际应用情况，我们发现，在实际应用中，深层的卷积神经网络通常被大型科研机构或互联网公司部署，用以完成复杂的或者数量庞大的识别、分类任务。这些应用场合对卷积神经网络的性能提出了较高的要求，而实现设备的体积和功耗则往往没有严格的限制。另一方面，浅层的网络如果想要得到更广泛的应用，结合智能电视、智能机顶盒以及智能手机等家用、个人智能终端是必不可少的。然而这些智能终端对设备尺寸和功耗都有着比较苛刻的限制，因此，分离卷积神经网络密集的运算，引入卷积神经网络协处理芯片的方案在浅层卷积神经网络的广泛应用方面则有着必然的需求和优势。

观察传统神经网络和卷积神经网络的运算结构特点，我们发现其中有比较强的独立性和重复性，因此结合FPGA可重构阵列架构的特点，自然将二者联系在一起，提出使用FPGA实现和加速卷积神经网络运算的基本思路。通过对卷积神经网络运算的具体分析，可以将其中最主要的部分分解为“从特征映射图到特征映射图”的基本运算，整个神经网络特征提取的部分就是对上述基本运算的并列和顺序组合。因此，设计实现这个基本运算的FPGA模块，并行或串行地组合这些实现基本运算功能的模块，能够充分发挥出FPGA的并行优势，完成在FPGA芯片上实现完整卷积神经网络的应用需求。

本文在对卷积神经网络基本运算模块的FPGA实现中充分继承前人研究成果，提出并设计了卷积神经网络中，卷积器、非线性激活函数和子采样器等关键模块的FPGA电路。在卷积器的设计中，本文讨论了多种并行程度的设计方案，其中串行-并行相结合的设计方案能灵活地适应不同的并行程度，使得设计方案能够满足不同的硬件条件，避免了单一设计方案对硬件要求过于固定的弊端，拓宽了该卷积器和卷积神经网络的应用场合。在非线性激活函数的设计中，本文从神经网络中最常见的Sigmoid函数入手，分析总结了多种常用的非线性函数FPGA拟合实现方案，针对Sigmoid函数本身特性，扬长避短，结合多种方案的优势之处，设计了线性拟合与查找表相结合的Sigmoid函数实现方案。同时，还针对硬件资源紧张的场景，提出复用式分段直线拟合方案，并且对该方案资源使用情况做了理论分析，为其实际应用提供了理论支持。在子采样器的设计中，本文设计了换位缓存结构，可以方便地从缓存输出端读取数据并进行多种子采样操作。相比于抽样读取方式或者卷积方式，换位缓存的方案灵活性高，通过改变读取端之后的运算器，可以方便地改变子采样方式。

在整体系统设计方面，一个实现“从特征映射图到特征映射图”基本运算的基本运算模块将上述三个关键模块组合，构成基本运算模块，该模块的串行、并行组合构成完整的卷积神经网络硬件系统。由于围绕FPGA中DSP单元进行设计，一个基本运算模块需要的逻辑资源很少，DSP资源也在合理范围内。整个神经网络的运算与上述实现的基本运算单元的运算是一致重复的，随着基本运算单元的复制增加，各种资源的消耗均呈线性增加。另外，其功耗控制表现良好，总功耗约为1.05W，随着基本运算模块数量的增加，功耗呈线性增加。每增加一个基本运算模块，总功耗增加约为7mW。对于本文提出的静态手势识别网络，完整实现的功耗控制在1.6W以内。该功耗表现相比于显卡、PC等平台有着非常明显的优势。因此，本文探索并设计的卷积神经网络芯片功耗较小，可以应用于小尺寸、低功耗应用平台。

本文在设计完成上述系统的同时，限于时间等因素，对某些问题和设计细节未能进行深入讨论，希望能在未来的工作中能完善卷积神经网络芯片在时序和精度方面的性能表现，并对以下问题进行更深入的研究：一，卷积神经网络以及传统的深度网络，作为运算重复性较高的一种算法，能否设计以可编程乘法器为基本单元的大规模可编程阵列器件，真正实现神经网络的灵活硬件化。二，以神经网络的“层”为单位，该层次的运算重复性依然比较强，由于每层一般都有多个神经元节点，能否将这些节点的运算模块组合，组成层次更高、功能比较完善的可编程运算单元，对这个层次的可编程单元阵列化，从而设计卷积神经网络硬件系统。三，在如今高层综合逐步普及的背景下，使用相关的设计工具，是否高效地设计性能可靠的卷积神经网络硬件系统，使用此类工具设计出来的系统与传统的RTL级设计在性能方面的优势和劣势。

# 参考文献

1. Kunihiko F., Sei M., Takayuki I. Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition[J]. IEEE Transactions on systems, man, and cybernetics, vol. SMC-13, No.5, September/October 1983: 826-834
2. Alex Krizhevsky., IlyaSutskever., Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks [C]. Advances in Neural Information Processing Systems 25 (NIPS 2012)
3. Yann L., Leon B., Yoshua B., et al. Gradient-Based Learning Applied to Document Recognition. in Proceedings of the IEEE , vol.86, no.11, pp.2278-2324, Nov 1998
4. Simard P.Y., Steinkraus D., Platt J.C. Best practices for convolutional neural networks applied to visual document analysis[C]. Proceedings of the seventh international conference on document analysis and recognition, 2003:958-962
5. Chellapilla K., Shilman M., Simard P. Optimally combining a cascade of classifiers[C]. Electronic Imaging, 2006:207-214
6. Frome A., Cheung G., Abdulkader A., et al. Large-scale privacy protection in google street view[C]. IEEE International Conference on Computer Vision, 2009:2373-2380
7. Garcia C., Delakis M. Convolutional face finder: A neural architecture for fast and robust face detection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(11): 1408-1423
8. Dawwd, S.A.; Mahmood, B.S., "A reconfigurable interconnected filter for face recognition based on convolution neural network," in Design and Test Workshop (IDT), 2009 4th International , vol., no., pp.1-6, 15-17 Nov. 2009
9. Neveu, J.N.P.; Kumar, M.J., "Implementation of the neocognitron on a SIMD architecture," in Intelligent Information Systems,1994. Proceedings of the 1994 Second Australian and New Zealand Conference on , vol., no., pp.179-183, 29 Nov-2 Dec 1994
10. 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
11. Cloutier, J.; Cosatto, E.; Pigeon, S.; Boyer, F.-R.; Simard, P.Y., "VIP: an FPGA-based processor for image processing and neural networks," in Microelectronics for Neural Networks, 1996., Proceedings of Fifth International Conference on , vol., no., pp.330-336, 12-14 Feb 1996
12. Ariyadoost, H.; Kavian, Y.S.; Ansari-Asl, K., "Two dimensional systolic adaptive DLMS FIR filters for image processing on FPGA," in Electrical Engineering (ICEE), 2012 20th Iranian Conference on , vol., no., pp.243-248, 15-17 May 2012
13. Antony W. Savich.,MedhatMoussa., ShawkiAreibi. The Impact of Arithmetic Representation on Implementing MLP-BP on FPGAs: A Study [J]. IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 18, NO. 1, JANUARY 2007: 240-252
14. Holt, J.L.; Baker, T.E., "Back propagation simulations using limited precision calculations," in Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on , vol.ii, no., pp.121-126 vol.2, 8-14 Jul 1991
15. Sankaradas, M.; Jakkula, V.; Cadambi, S.; Chakradhar, S.; Durdanovic, I.; Cosatto, E.; Graf, H.P., "A Massively Parallel Coprocessor for Convolutional Neural Networks," in Application-specific Systems, Architectures and Processors, 2009. ASAP 2009. 20th IEEE International Conference on , vol., no., pp.53-60, 7-9 July 2009
16. Yunji Chen; Tao Luo; Shaoli Liu; Shijin Zhang; Liqiang He; Jia Wang; Ling Li; Tianshi Chen; ZhiweiXu; Ninghui Sun; Temam, O., "DaDianNao: A Machine-Learning Supercomputer," in Microarchitecture (MICRO), 2014 47th Annual IEEE/ACM International Symposium on , vol., no., pp.609-622, 13-17 Dec. 2014
17. 方睿,刘加贺,薛志辉,杨广文. 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,2015,08:32-36
18. ALTERA NIOS处理器。URL：https://www.altera.com/products/processors/overview.html
19. 赛灵思ZYNQ系列器件。URL：http://www.origin.xilinx-china.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html
20. Xiaoyin Shao; Dong Sun, "Development of an FPGA-Based Motion Control ASIC for Robotic Manipulators," in Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on , vol.2, no., pp.8221-8225
21. Omar, A.M.; Rahim, N.A.; Mekhilef, S., "Three-phase synchronous PWM for flyback converter with power-factor correction using FPGA ASIC design," in Industrial Electronics, IEEE Transactions on , vol.51, no.1, pp.96-106, Feb. 2004
22. Honegger, D.; Oleynikova, H.; Pollefeys, M., "Real-time and low latency embedded computer vision hardware based on a combination of FPGA and mobile CPU," inIntelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on , vol., no., pp.4930-4935, 14-18 Sept. 2014
23. Maggiani, L.; Salvadori, C.; Petracca, M.; Pagano, P.; Saletti, R., "Reconfigurable FPGA architecture for computer vision applications in Smart Camera Networks," in Distributed Smart Cameras (ICDSC), 2013 Seventh International Conference on , vol., no., pp.1-6
24. Chiuchisan, I., "A new FPGA-based real-time configurable system for medical image processing," in E-Health and Bioengineering Conference (EHB), 2013 , vol., no., pp.1-4, 21-23 Nov. 2013
25. CaiXuesen; Han Liguo; Dai Jinbo, "Design and implementation of FPGA-base diagnosis of medical image data acquisition equipment," in Electronic Measurement & Instruments (ICEMI), 2011 10th International Conference on , vol.3, no., pp.51-55, 16-19 Aug. 2011
26. Blair, C.; Robertson, N.M.; Hume, D., "Characterizing a Heterogeneous System for Person Detection in Video Using Histograms of Oriented Gradients: Power Versus Speed Versus Accuracy," in Emerging and Selected Topics in Circuits and Systems, IEEE Journal on , vol.3, no.2, pp.236-247, June 2013
27. Xiaoyin Ma; Najjar, W.A.; Roy-Chowdhury, A.K., "Evaluation and Acceleration of High-Throughput Fixed-Point Object Detection on FPGAs," in Circuits and Systems for Video Technology, IEEE Transactions on , vol.25, no.6, pp.1051-1062, June 2015
28. Sledevic, T.; Serackis, A., "SURF algorithm implementation on FPGA," in Electronics Conference (BEC), 2012 13th Biennial Baltic , vol., no., pp.291-294, 3-5 Oct. 2012
29. Bouris, D.; Nikitakis, A.; Papaefstathiou, I., "Fast and Efficient FPGA-Based Feature Detection Employing the SURF Algorithm," in Field-Programmable Custom Computing Machines (FCCM), 2010 18th IEEE Annual International Symposium on , vol., no., pp.3-10, 2-4 May 2010
30. 陆志坚. 基于FPGA的卷积神经网络并行结构研究[D].哈尔滨工程大学,2013
31. Farabet, C.; Poulet, C.; Han, J.Y.; LeCun, Y., "CNP: An FPGA-based processor for Convolutional Networks," in Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on , vol., no., pp.32-37, Aug. 31 2009-Sept. 2 2009
32. Hui Zhang; Mingxin Xia; Guangshu Hu, "A Multiwindow Partial Buffering Scheme for FPGA-Based 2-D Convolvers," in Circuits and Systems II: Express Briefs, IEEE Transactions on , vol.54, no.2, pp.200-204, Feb. 2007
33. Del Campo, I.; Finker, R.; Echanobe, J.; Basterretxea, K., "Controlled accuracy approximation of sigmoid function for efficient FPGA-based implementation of artificial neurons," in Electronics Letters , vol.49, no.25, pp.1598-1600, December 5 2013
34. Varkonyi-Koczy, A.R.; Tusor, B., "Human–Computer Interaction for Smart Environment Applications Using Fuzzy Hand Posture and Gesture Models," inInstrumentation and Measurement, IEEE Transactions on , vol.60, no.5, pp.1505-1514, May 2011
35. Coric, S.; Latinovic, I.; Pavasovic, A., "A neural network FPGA implementation," Neural Network Applications in Electrical Engineering, 2000. NEUREL 2000. Proceedings of the 5th Seminar on, pp.117-120, 2000.
36. Gironés, R. G.; Palero, R. C.; Boluda, J. C.; Cortés, A. S., “FPGA Implementation of a Pipelined On-Line Backpropagation,” J. VLSI Signal Process. Syst.,vol. 40, no. 2, pp.189-213., Jun 2005.
37. Ciresan, D.; Meier, U.; Schmidhuber, J., "Multi-column deep neural networks for image classification," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on , vol., no., pp.3642-3649, 16-21 June 2012
38. Ciresan, D.; Meier, U., "Multi-Column Deep Neural Networks for offline handwritten Chinese character classification," in Neural Networks (IJCNN), 2015 International Joint Conference on , vol., no., pp.1-6, 12-17 July 2015
39. Auviz Systems加速卷积神经网络性能。URL：http://xilinx.eetrend.com/article/8769
40. 未来深度学习硬件前瞻。URL：http://www.nextplatform.com/2015/08/25/a-glimpse-into-the-future-of-deep-learning-hardware/

# 攻读学位期间获得的学术成果

[1]Yu Wang; XiangminXu; Xiaoshi Chen; Xiaobo Jiang, "A Look-Up-Table based skin color segmentation method," in *Consumer Electronics - China, 2014 IEEE International Conference on* , vol., no., pp.1-3, 9-13 April 2014

[2] Yu Wang; XiangminXu; Xiaobo Jiang; Ling, B.W.-K., "A parallel VLSI architecture of contactless HCI system," in *Picture Coding Symposium (PCS), 2015* , vol., no., pp.267-271, May 31 2015-June 3 2015

# 致谢

首先，感谢我的导师徐向民教授。在我攻读硕士的过程中，论文的每一步工作都倾注着老师的心血。徐老师无论是在学习、工作、生活等方面都给予我无私的教诲、悉心的指导和无微不至的关心。徐老师严谨求实的治学态度、踏实的工作作风、敏锐的思维方式和平易近人的生活态度潜移默化地影响着我如何去做人、做事、做学问，是我终身学习的榜样。

感谢姜小波老师对我科研工作的悉心指导和大力支持，姜老师勤奋的工作态度、出色的科研能力和的刻苦的钻研精神深深的影响了我，是我学习的楷模。

感谢邢晓芬老师对我攻读研究生期间，对我学术论文方面的悉心指导和大力支持，对我的教导让我终生受益。

感谢新型人机交互创新实验室的苗捷博士、裘索博士、刘芳博士等几位博士。在我攻读硕士期间，几位博士在科研管理、科学研究方法和学习生活上均给予我精心的指导，使我受益匪浅。

感谢同实验室的李叙琼、冯永鹏、杨锋超、蔡博伦、廖国铭、黄彬等同学、学长，感谢你们在实验室共同营造浓厚的学术氛围，和你们的讨论使我受益匪浅。

感谢所有曾经给予我帮助和支持的老师、朋友和同学。

最后，感谢我的父母和家人对我硕士论文工作的巨大支持和生活中无微不至的关爱，我的每一点进步都与你们的无私奉献分不开，没有你们的支持我将无法完成此篇论文。