◎理论研究、研发设计◎

卷积神经网络的FPGA并行加速方案设计

方 睿,刘加贺,薛志辉,杨广文
FANG Rui, LIU Jiahe, XUE Zhihui, YANG Guangwen

清华大学 计算机科学与技术系,北京 100084
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

FANG Rui, LIU Jiahe, XUE Zhihui, et al. FPGA-based design for convolution neural network. Computer Engineering and Applications, 2015, 51(8):32-36.

Abstract: According to the characteristics of the Convolution Neural Network(CNN), a FPGA-based acceleration program which uses deep-pipeline architecture is proposed for the MNIST data set. In this program, theoretically 28×28 clock cycles can finish the whole calculation and get the output of the CNN. For the propagation stage of the training process, and in the same network structure and the same data set, this FPGA program with 50 MHz frequency can achieve nearly five times speedup compared to GPU version(Caffe), achieve eight times speedup compared to 12 CPU cores. While the FPGA program just costs 26.7% power which GPU version costs.

Key words: convolution neural network; Field Programmable Gate Array(FPGA); deep-pipeline; acceleration

摘 要:根据卷积神经网络的特点,提出了深度流水的FPGA加速方案,设计了卷积层的通用卷积电路。该卷积电路可以在一个时钟周期内获得一个计算结果。理论上,该方案对于MNIST数据集,在28×28个时钟周期内可以获得一幅图片的运算结果。针对网络训练过程的前向传播阶段,在网络结构和数据集相同的情况下,对GPU,FPGA,CPU进行了在计算效率和能耗之间的比较。其中在计算效率方面,50 MHz频率的FPGA就可以相较于GPU实现近5倍的加速,相较于12核的CPU实现8倍的加速。而在功耗方面,该FPGA的实现方案只有GPU版本的26.7%。

关键词:卷积神经网络;现场可编程门阵列(FPGA);深度流水;加速
文献标志码:A **中图分类号:**TP391 doi:10.3778/j.issn.1002-8331.1405-0335

1 引言

卷积神经网络(Convolutional Neutral Network)是人工神经网络的一种。CNN是第一个真正成功训练多层网络结构的学习算法。它利用空间关系,采用权值共享网络结构,使之更类似于生物神经网络,降低了网络模型的复杂度并减少了权值的数量,以提高一般前向BP算法的训练性能。该优点在网络的输入是多维图像时表现得更明显。另一方面,在CNN中,图像可以直接作为网络的底层输入,信息再依次传输到不同的层,每层通过一个数字滤波器去获得观测数据的最显著的特征,避免了传统识别算法中复杂的特征提取和数据重建。这个方法能够获取对平移、缩放和旋转不变的观测数据的显著特征,因为图像的局部感受区域允许神经元或者处理单元可以访问到最基础的特征,例如定向边缘或者角点。

CNN的这些优点使得它如今已成为当前语音分析和图像识别领域的研究热点。AT&T的支票读取系统[1]、微软的OCR[2-4]和手写识别系统、Google街景中的人脸

基金项目:国家高技术研究发展计划(863)(No.2010AA012302,No.2013AA01A208);国家自然科学基金(No.61040048,No.61303003,No.41374113)。

作者简介:方睿(1978—),男,博士,研究领域为高性能计算、数据挖掘;刘加贺(1992—),男,硕士,研究领域为高性能计算、数据挖掘;薛志辉(1988—),男,硕士,研究领域为高性能计算、数据挖掘;杨广文(1963—),男,博士,教授,研究领域为并行处理,、高性能计算、高等计算机系统结构。E-mail:liujiahe14@mails.tsinghua.edu.cn

收稿日期:2014-05-27 **修回日期:**2014-09-19 **文章编号:**1002-8331(2015)08-0032-05
CNKI**网络优先出版:**2014-10-29, http://www.cnki.net/kcms/doi/10.3778/j.issn.1002-8331.1405-0335.html

识别和车牌识别[5]以及法国电信视频会议系统中的人脸识别[6]都用到了CNN。

现有的大部分CNN实现主要是基于通用处理器CPU实现的。在CNN网络结构[7]中，层内计算是独立不相关的，而层间结构可以理解为是一个流水结构。CPU由于其自身特点无法充分地挖掘CNN内部的并行性。FPGA（Field-Programmable Gate Array），即现场可编程门阵列，作为一种计算密集型加速部件，通过将算法映射到FPGA上的并行硬件进行加速。FPGA上所设计的各个硬件模块可以并行执行。各个硬件模块输入输出的相互连接以及FPGA所提供的流水结构可以很好地和CNN算法相匹配，充分利用算法网络结构内部的并行性，在提高运算速度的同时缩小了能耗。之前已经有学者在FPGA上实现了不同结构的CNN来做简单的实时图像识别或分类[8-12]，但并没有完全发挥出FPGA的计算潜能，并具有较差的扩展性。

本文首先简单地介绍卷积神经网络CNN，然后根据CNN的特点设计了高效的FPGA的硬件加速方案，并且和已有GPU的实现进行了比较，并对进一步研究工作进行了展望。

2　卷积神经网络CNN

2.1　CNN的网络结构

CNN是深度神经网络的一种。图1所示为一种典型的卷积神经网络的层次化体系结构，卷积神经网络的每层通常由两个子层构成：卷积层和子抽样层，此外在卷积计算和子抽样之后还需要对输出数据进行非线性变换处理，典型的卷积神经网络通常包含1~3个具有两个子层结构的层，最后再连接一个分类层，如Softmax。

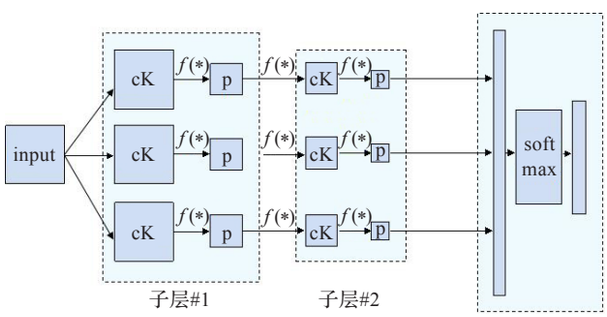

图1　卷积神经网络网络结构示意图

（1）卷积层中的局部接受域和权值共享

二维空间上的局部接受域使得神经网络从输入图像中提取初级视觉特征，如特定角度的边缘，端点等，后续各层可以组合这些初级特征，从而得到更高层的特征。局部接受域从客观上减少了需要训练的权值数目。

局值共享使那些共享同一组权值的神经元在输入的不同位置可以检测同一种特征。卷积神经网络把每层共享相同权值的神经元组织成了一个二维平面-特征图，这使得输入中的平移变化，会以同样的方向和距离出现在特征图的输出中，但是不引起其他形式的变化。权值共享大幅减少需要训练的权值数目，从而可以大大降低对训练样本的需求。

在卷积层，上一层的特征图会和当前卷积层的卷积核进行卷积，卷积出来的结果加权求和后经过非线性函数处理后得到这一层的特征图。一般地，卷积层可以用以下的表达式来表示：

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right) \tag{1}$$

其中 x_j^l 表示第 l 层卷积层的第 j 个卷积核对应的特征图，M_i 表示当前卷积对前输入特征图的选择，k_{ij}^l 表示第 l 层的第 j 个卷积核的第 i 个加权系数，b_j^l 表示第 l 层卷积层的第 j 个卷积核对应的偏置系数，而 f 为非线性函数，如 *relu* 函数，如下：

$$relu(x) = \begin{cases} x, x > 0 \\ 0, x \leqslant 0 \end{cases} \tag{2}$$

（2）抽样层-子采样

抽样层中的采样操作无重叠地采集上一层特征图同一大小的子区域，取均值或者最大值作为输出结果，从而降低输入图像分辨率，使得卷积神经网络对输入的局部变换具有一定的不变性。这模拟初级视皮层中的复杂细胞。抽样层的一般形式如下：

$$x_j^l = f(\beta_j^l down(x_j^{l-1}) + b_j^l) \tag{3}$$

其中 $down(\cdot)$ 是下采样函数。每一个抽样层都有自己的加权系数 β 和偏置系数 b 。

2.2　训练过程

卷积网络是一种有监督的网络。在卷积网络训练的过程中，所输入的样本集是由（输入向量，给定标签）这样的向量所组成。卷积网络从本质上来看是一种输入到输出的映射，它通过用大量的样本进行网络训练，构造输入和输出之间的映射关系，而不需要对输入样本与标签进行精确的数学表述。在开始训练前，在网络中所有的权重及偏置系数需要随机初始化。随机数不能太大，以此来保证网络不会因权值过大进入饱和状态。

训练算法与传统的BP算法差不多。主要分为两个阶段：

第一阶段，前向传播阶段：

（1）从样本集中取一个样本 (x, y_p)，将 x 输入网络。

（2）计算相应的实际输出 O_p 。

此阶段，信息从输入层经过逐级的变换，传送到输出层。在这个过程中，输入的图像信息需要经过多层的卷积操作和非线性操作。

第二阶段，向后传播阶段：

（1）算实际输出 O_p 与相应的理想输出 y_p 的差。

（2）按极小化误差的方法反向传播进行权矩阵调整。

3　FPGA设计方案

本章将先介绍本次实验中采用的模型结构,接着介绍整体的硬件架构设计,最后介绍本实验提出的卷积运算单元模块及其组合应用。

3.1　CNN模型结构

如图2所示,本文所采用的CNN模型结构是由1个输入层input,1个输出层output,2个卷积层,2个池化和一个全连接网络Softmax组成。在本文实验中,输入的图像集为手写数字图像集MNIST。每幅图像的大小为28×28像素点。具体的网络结构如下:

Input layer:28×28;

C1 Conv layer:3 kernels,each with size 5×5,*stride*=1;

S1 Max-pooling layer:each with size 2×2,*stride*=2,β=1.0,*b*=0.0;

C2 Conv layer:6 kernels,each with size 5×5,*stride*=1;

S2 Max-pooling layer:each with size 2×2,*stride*=2,β=1.0,*b*=0.0;

Softmax:output:10 vector(classification result).

其中kernel是卷积核,都采用大小为5×5的卷积核对特征图像进行卷积操作;kernel的 *stride* 表示卷积核在特征图像上做卷积的平移步数。

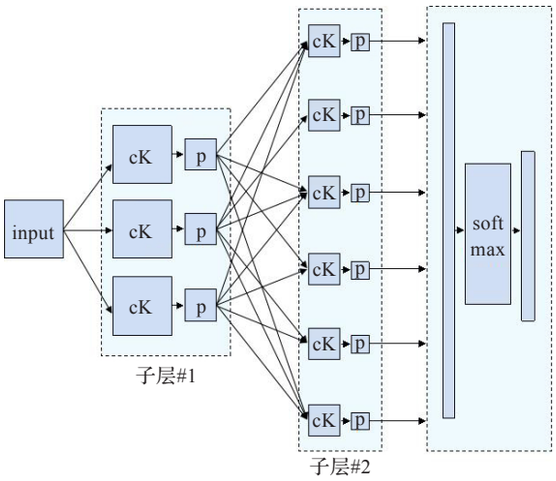

图2　数字图像识别的CNN模型结构

在上述模型中涉及的权重系数以及偏置系数个数如下。

*w*参数个数计算公式:*w*参数个数=卷积核个数×单个卷积核的*w*参数个数

*b*参数个数计算公式:*b*参数个数=卷积核个数

(1)第一层:*w*参数个数为3×25=75个,*b*参数个数为3个。

(2)第二层:*w*参数个数为(3×6)×25=450个,*b*参数个数为18个。

(3)Softmax层:*w*参数个数为6×4×4×10=960个,*b*参数个数为1个。

三层所有参数个数的总数为1 507个,如果采用float来存储参数的话,每个参数占32位,则总共消耗的存储约为6 kB,非常得小,PCIE的贷款为8 GB/s,因此参数的传输时间占得比重很低。而且这些权重以及偏执参数在较长的时间内是不会发生改变,需要多次的前向操作后才会更新。考虑到程序的扩展性,这些参数通过DRAM传入。

3.2　硬件架构设计

只考虑前向部分,通过上述的模型以及计算过程,可以看出CNN算法中每一层算完之后需要把该层所有的输出信息作为下一层的输入,此时该层将会停止运作,直至当前的图片被完全地处理完。FPGA不同于CPU,单核CPU每一时刻只能执行一条指令,而FPGA上的硬件模块可以同时进行运算。因此如果CNN在FPGA上的实现采用流水结构可以大大提高资源的使用效率。

图像数据的输入是以数据流的方式进行流入,在每个时钟周期内传入一个图像的像素点。因此在最理想的情况下,每28×28个时钟周期便可以完成对一幅图像的运算。为了达到这种效果,网络中的每一层需要每时钟输入一个值并且输出一个值。

如图3所示,在CPU端,读取了输入的训练数据图像集,对输入图像做归一化预处理之后,通过PCIE总线传输给Dataflow Engine;CNN中用到的参数*w*和参数*b*是在CPU中初始化的,也通过PCIE传输到Dataflow Engine的Fast Memory当中,这样将加快卷积运算取参数的过程;Dataflow Engine中实现了2个卷积层和1个softmax全连接层的硬件电路,数据流过Dataflow Engine之后将会产生图像的分类结果并将结果传回CPU。

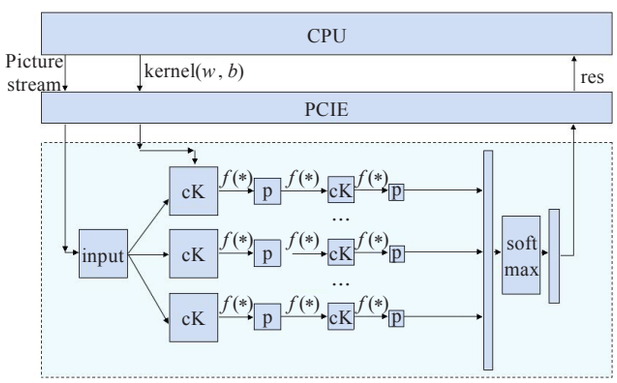

图3　CNN-FPGA硬件架构设计

3.3　卷积运算单元设计

在网络的各层中,卷积层需要的乘加操作最多,也是最耗时的操作。为了保证卷积层能够在每个时钟周期内输出特征图的一个点,需要对卷积层做以下几点:

(1)将单个卷积运算继续拆分实现并行化,使得单个卷积运算能够在每个时钟周期输出一个点。

(2)将同一层内的多个卷积运算进行并行化，使得在给定的28×28个时钟周期内能够同时流出完该层的多个特征图。

针对第一点，首先根据硬件电路的数据流的特点设计了卷积运算单元(CCU)，使得在单个FPGA周期就可以完成整个5×5大小卷积运算，一个5×5大小的卷积运算相当于25个乘法和24个加法，而FPGA在一个周期就可以完成CPU49个周期才能完成的任务，在相当于单个卷积运算的基础上就实现了49倍的加速。

FPGA其实是使用资源(Resource)或者空间(Space)换取时间的一种实现方式。在单个周期(Cycle)下，由于数据是流过电路的，因此电路中的每一个计算单元都进行了一次运算，当流水线越深，计算单元的数量越多的时候，相当于单个周期内的运算量越大，加速效果越好。

实现单个卷积运算的加速其实就是采用了5×5个乘法器和加法器设计而成的卷积运算单元。为了方便显示，3×3大小的卷积电路设计如图4所示。

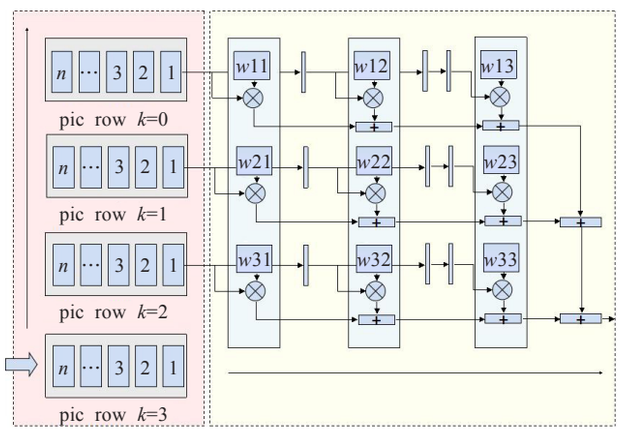

图4　3×3卷积运算单元(CCU)电路简图

在图4中左侧区域是输入数据预处理区域，每当28个周期之后，将会填满一个row，此时该row的数据会整体向上推移。而在这个28个周期中，计算部分的row将会往前推移一步。也就是说每隔一个时钟周期将会进行一个3×3的卷积操作。在上面的电路中，它用到了9个乘法器和8个加法器。

针对第二点，对于本文模型的第一个卷积层，有3个kernel，因此需要对同一张输入图像的同一个5×5的部分做3次卷积运算。对于第一层，由于只有一个输入，各个kernel可以共享图4中左侧的电路，而每个kernel采用图4右侧的类似电路，只是权重的值不同。而对于第二层，和第一层类似，只是输入的数量变成3个，kernel的数量变成6个，因此需要18个卷积运算电路。

通过以上的设计，保证了在每个时钟周期内，每一层可以输出相关特征图中的某一个像素点上的值，也就是每28×28个时钟周期可以获得一幅图像在该层的输出结果。

3.4　计算精度

为了有效利用FPGA上的资源，本文减少了数据精度。通过实验比较发现，对于权值数据采用了20 bit的定点数，而其他的输入参数则采用了16 bit的定点数，可以获得与单精度浮点数相当的最为接近的运算结果。

4　实验结果

4.1　实验环境

在本文研究中，使用了Maxeler公司开发的Maxide环境[13-14]进行硬件开发和测试，使用的FPGA电路板为Xilinx Virtex-6（SX475T）FPGA的Max3加速板卡，FPGA卡通过带宽为8 GB/s的PCI Express 2.0 x8与CPU进行连接，在FPGA板上有最大带宽为38 GB/s的24 GB的DDR3 DRAM。

目前采用的用于数字识别的图像是采用MNIST数据集，其中包含60 000张训练图像和10 000张测试图像，每张图像为灰度图像，大小为28像素×28像素，每个像素的取值在0~255的整数范围内，对该数据进行了初始化，使得输入网络的数值范围变为[−0.5，0.5]。

同时还在GPU和CPU上进行了同样算法的同样数据集的测试。软件上，GPU和CPU上的CNN代码采用的是目前得到普遍认可的Caffe库。通过调整CNN模型的参数使得和本文FPGA上采用的CNN模型一致。硬件上，GPU本文采用的是NVIDIA tesla K20c，峰值双精度浮点性能为1.17 Tflop，峰值单精度浮点性能为3.53 Tflop，存储器带宽为208 GB/s，存储器容量为5 GB，CUDA核心数量为2 496。而CPU本文采用的是12核Inter® Xeon®主频为2.40 GHz的CPU。

稍后会给出在这三种平台下的相关测试的实验结果。

4.2　实验结果

目前实现了图像数字识别训练过程的前向传播过程在FPGA中，并在GPU，CPU和FPGA中测试其训练过程的前向传播过程。目前将FPGA的频率设置为50 MHz，单块FPGA电路板上的资源消耗情况如表1所示。

表1　单块FPGA电路板上的资源消耗情况

	资源	比例/%
LUTS	10 078/297 600	3.39
FFS	64 364/595 200	10.81
BRAMS	23/1 064	2.16
DSPS	1 485/2 016	73.66

可以看出，目前DSP资源消耗比例很大，主要原因是在算法中所需要的乘法器比较多。

表2是在三种不同平台下的相同的CNN模型在MNIST数据集下做图像识别训练过程的前向传播过程的计算效率对比。算法耗时包括了图片数据传输的时间和卷积神经网络的处理时间。由此可以看出FPGA识别的效率相较于12核Inter® Xeon®主频为2.40 GHz

的CPU提高了8倍,相较NVIDIA tesla K20c GPU提高了近5倍,由于FPGA上硬件电路的复杂度高,本文对电路进行了相应的延时优化从而在牺牲了部分性能的基础上保证了电路的实时要求。

表2 三种不同平台下实现的CNN算法应用于图像识别的效率对比

	图像/迭代	时间/s
12-core Xeon CPU	60 000	34.02
NVIDIA tesla K20c GPU	60 000	20.43
Xilinx Virtex-6 FPGA	60 000	4.33

而对于能耗,本文使用Xilinx公司提供了Xilinx Power Estimator(XPE)来估测FPGA芯片的耗能情况,结果如表3所示。

表3 GPU和FPGA的功耗对比

平台	功耗/W
NVIDIA tesla K20c GPU	225
Xilinx Virtex-6 FPGA	60

综上可以看到FPGA在获得比GPU近5倍的加速比的同时,在功耗上只有GPU所消耗的26.7%。

5 结语

本文根据卷积神经网络层内独立层间高度流水的特点,设计了深度流水的FPGA加速方案,根据权值及相关参数的取值范围确定了权值及参数的定点数精度大小,并且设计了卷积操作电路,使得卷积操作能够在一个时钟周期内输出当前层输出特征图的一个值,大大提高了计算的效率和资源使用效率。该方案对于MNIST数据集,在28×28个时钟周期内可以获得一幅图片的运算结果。并且本文对网络训练过程的前向传播阶段,在网络结构和数据集相同的情况下,对GPU,FPGA,CPU进行了在计算效率和能耗之间的比较。其中在计算效率方面,50 MHz频率的FPGA就可以相较于GPU实现近5倍的加速,相较于12核的CPU实现8倍的加速。而在功耗方面,本文FPGA的实现方案只有GPU版本的26.7%。

参考文献:

[1] LeCun Y,Bottou L,Bengio Y,et al.Gradient-based learning applied to document recognition[J].Proceedings of the IEEE,1998,86(11):2278-2324.

[2] Simard P Y,Steinkraus D,Platt J C.Best practices for convolutional neural networks applied to visual document analysis[C]//Proceedings of the 7th International Conference on Document Analysis and Recognition,2003:958-962.

[3] Chellapilla K,Puri S,Simard P.High performance convolutional neural networks for document processing[C]//10th International Workshop on Frontiers in Handwriting Recognition,2006.

[4] Chellapilla K,Shilman M,Simard P.Optimally combining a cascade of classifiers[C]//Electronic Imaging,2006:207-214.

[5] 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.

[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] Bouvrie J.Notes on convolutional neural networks[Z].2006.

[8] Sankaradas M,Jakkula V,Cadambi S,et al.A massively parallel coprocessor for convolutional neural networks[C]//20th IEEE International Conference on Application-specific Systems,Architectures and Processors,2009:53-60.

[9] Cadambi S,Majumdar A,Becchi M,et al.A programmable parallel accelerator for learning and classification[C]//Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques,2010:273-284.

[10] Farabet C,Poulet C,LeCun Y.An FPGA-based stream processor for embedded real-time vision with convolutional networks[C]//2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops),2009:878-885.

[11] Farabet C,Poulet C,Han J Y,et al.CNP:an FPGA-based processor for convolutional networks[C]//International Conference on Field Programmable Logic and Applications,2009:32-37.

[12] Perko M,Fajfar I,Tuma T,et al.Low-cost,high-performance CNN simulator implemented in FPGA[C]//Proceedings of the 2000 6th IEEE International Workshop on Cellular Neural Networks and Their Applications,2000:277-282.

[13] Dataflow programming with MaxCompiler.version 2012.1[Z].2012.

[14] MaxCompiler manager compiler tutorial.version 2012.1[Z].2012.