

基于FPGA的高效卷积神经网络设计

潘坤榕¹, 夏福源², 李瑞民¹, 刘子嫣¹, 唐珂¹, 孙科学^{1,3*}

(1. 南京邮电大学 电子与光学工程学院、微电子学院, 江苏 南京 210023;

2. 南京邮电大学 贝尔英才学院, 江苏 南京 210023;

3. 射频集成与微组装技术国家地方联合工程实验室, 江苏 南京 210023)

摘要:作为深度学习的代表算法之一,卷积神经网络因为拥有良好的特征提取能力而被广泛应用于计算机视觉、自然语言处理等领域。然而,因为卷积神经网络拥有庞大的计算量,主流的硬件平台往往不能满足模型的各种需求。例如,CPU受限于自身架构无法提供高效的算力;GPU因功耗太高而无法满足不同设备需求;ASIC开发周期较长,成本较高,难以实现设计的复用。现场可编程逻辑门阵列是一种半定制电路,拥有计算力强、功耗低等特点,其并行化的结构特点正适用于卷积神经网络模型的搭建。针对MINST数据集,该文提出了一种卷积神经网络模型的设计思路及优化方法,并利用VIVADO HLS工具在FPGA平台上完成卷积神经网络模型的部署,探讨了卷积层IP核的通用性设计。经实验验证,卷积层的时钟周期经优化后大大缩短,卷积层的设计可通过参数调整实现复用。部署于FPGA的卷积神经网络模型性能良好,能通过参数传输的方式实现针对不同数据的通用。

关键词:人工智能;卷积神经网络;现场可编程逻辑门阵列;数字识别;TensorFlow

中图分类号:TP391.4

文献标识码:A

文章编号:1673-629X(2022)07-0105-06

doi:10.3969/j.issn.1673-629X.2022.07.018

Design of Efficient Convolutional Neural Network Based on FPGA

PAN Kun-rong¹, XIA Fu-yuan², LI Rui-min¹, LIU Zi-yan¹, TANG Ke¹, SUN Ke-xue^{1,3*}

(1. School of Electronic and Optical Engineering, Nanjing University of Posts and Telecommunications,

Nanjing 210023, China;

2. Bell Honors School, Nanjing University of Post and Telecommunications, Nanjing 210023, China;

3. Nation-Local Joint Project Engineering Lab of RF Integration & Micropackage, Nanjing 210023, China)

Abstract: As one of the typical algorithms of deep learning, convolutional neural network algorithm is widely used in computer vision, natural language processing and other fields because of its excellent feature extraction capabilities. However, because the convolutional neural network has a huge amount of calculation, mainstream hardware platforms often cannot meet the various needs of the model. For example, CPU is limited by its own architecture and cannot provide efficient computing power; GPU cannot meet the needs of mobile devices due to high power consumption; ASIC has a long development cycle, high cost, and non-reusability. Field programmable gate array is a kind of semi-custom circuit, which has the characteristics of strong computing power and low power consumption. Its parallel structure is suitable for the construction of convolutional neural network models. We present a design idea and optimization method of convolutional neural network model for MINST data set, use VIVADO HLS tool to complete the deployment of convolutional neural network model on FPGA platform, and discuss the universal design of IP core in convolution layer. Experiments have verified that the clock cycle of the convolutional layer is greatly shortened after optimization, and the design of the convolutional layer can be reused through parameter adjustment. It is showed that the convolutional neural network model deployed on the FPGA has good performance and can be universal for different data through parameter transmission.

Key words: artificial intelligence; CNN; field programmable gate array/FPGA; digit recognition; TensorFlow

0 引言

人工智能(Artificial Intelligence, AI)作为计算机

科学技术领域的一项重要分支,得益于近年来高速发展的计算硬件,越来越受到关注并逐渐发展为最重要

收稿日期:2021-05-06

修回日期:2021-09-09

基金项目:江苏省大学生创新训练计划(SYB2020012);南京邮电大学自孵化项目(NY220013)

作者简介:潘坤榕(2000-),男(壮族),研究方向为电子技术与智能信号处理;通信作者:孙科学,博士,教授,硕导,研究方向为智能信号处理与通信软件设计。

的研究领域之一,其发展成果渗透到了各个领域,对现代世界的方方面面产生了深远的影响。作为人工智能的一个重要研究领域,图像识别技术在探查资源、公安刑侦、生物医学等领域有着广泛的应用^[1]。举例来说,图像识别技术能够提取遥感图像的信息以探查森林、水利、海洋、农业等资源;能够完成指纹、手印、人像等类型数据的辨别以向公安机关提供有效的线索;能够处理病患的各类图像数据而为医生提供各种辅助诊断的信息。受限于硬件和算法,图像识别技术的发展一度停滞。90 年代,支持向量机和人工神经网络的结合促进了图像识别技术的发展^[2-3]。人工神经网络需要人为参与预处理,这导致了图像识别准确率的下降。为摆脱人工神经网络的限制,人们提出了循环神经网络(Recurrent Neural Network, RNN)^[4]、深度信念网络(Deep Belief Network, DBN)^[5-6]、卷积神经网络(Convolutional Neural Network, CNN)等深层次网络结构^[7]。

其中,卷积神经网络以其良好的特征提取能力和泛化能力,在图像处理、目标跟踪与检测、自然语言处理、场景分类、人脸识别等诸多领域获得了巨大的成功^[8]。^{[5][6]}卷积神经网络模型性能优异,但计算量异常庞大^[9],因此对计算机硬件设备的计算能力有着比较严苛的要求。现行的主流硬件平台包括 CPU、GPU、FPGA、ASIC 等,其中, CPU 因自身架构的局限而难以支持并行运算,处理效率不高; GPU 因价格昂贵,功耗太高无法应用于嵌入式移动终端;专用集成电路(Application-Specific Integrated Circuit, ASIC)计算强、功耗低,但其通用性差、成本高昂且可迁移性低。现场可编程逻辑门阵列(Field Programmable Gate Array, FPGA)器件在 1985 年由 Xilinx 公司发明,它的出现在很大程度上弥补了复杂可编程逻辑器件(Complex Programmable Logic Device, CPLD)和 ASIC 之间的空白,这种半定制化的解决方案既克服了 ASIC 开发成本巨大的弊端,又有着通用可编程逻辑器件无法比拟的运算效率。此外, FPGA 配置了众多逻辑单元可用于深度学习算法的并行计算,其计算力强、功耗低,并行化的结构特点正适用于卷积神经网络模型的部署^[10-11]。

于是卷积神经网络模型的研究者和工程师们将目光转向了 FPGA 平台。FPGA 加速深度学习算法,往往面临如下挑战:有限的 FPGA 资源难以满足庞大的计算和数据需求;开发周期长,学习成本高,设计复用性差^[12]。深度学习算法经过长足的发展,其算法复杂度不断提高,其数据量也大幅增长。近年来, FPGA 片上资源越来越丰富,但尚未能满足不断增长的算法配置需求。设计者必须关注如何使得资源得到高效的利

用,否则其精心设计的算法将因资源的局限而无法配置成功。特别地,对于具有商业意义的设计而言,设计者更需要合理地降低产品生产成本及人工成本。大规模的 FPGA 芯片,往往价格高昂,资源与成本两者间存在矛盾;相较于程序设计人员, FPGA 开发者需要丰富的硬件知识及硬件开发经验; FPGA 系统的开发流程,也比软件设计更繁琐、冗长。此外, FPGA 加速深度学习算法的设计往往只针对特定的应用场景,成本高昂的设计不能得到很好的重用,这造成了资源的浪费。

为解决以上难题,该文以 MNIST 数据集为例,在 TensorFlow 上搭建及训练卷积神经网络模型,探讨了基于 FPGA 平台的卷积神经网络设计与优化方法,通过传输模型参数的方法实现了卷积层模块的可重用,最后,完成驱动及系统上位机的程序设计,并对系统进行功能和性能测试。

1 卷积神经网络

1.1 卷积神经网络定义

卷积神经网络是一种“端到端”的学习方法,它将图像像素信息作为输入,通过卷积操作进行图像像素特征的提取,对原始图像进行高度抽象,既能够在最大程度上确保输入图像信息的完整性,又使得模型的输出直接是图像识别的结果,因此在图像处理领域获得了优异的性能表现,并得到了广泛的实际应用。一个完整的卷积神经网络模型的组成结构为:若干个卷积层(Convolutional Layer)、池化层(Pooling Layer)和全连接层(Fully Connected),处理单元包括:卷积核、池化、激活函数、分类器等^[13],其中卷积层和全连接层的功能主要是完成原始图像特征提取,而池化层的主要目的是防止过拟合现象的出现。当待处理的数据输入卷积神经网络模型后,通过卷积层进行数据的特征提取,池化层分为最大池化层、均值池化层等类型,完成数据的采样,全连接层处理输入数据和输出数据的线性映射。如图 1 所示,该模型为一个手写数字识别的卷积神经网络算法结构。

卷积层利用多个不同的卷积核过滤提取不同的特征信息,多个不同的卷积核可以视作不同的滤波器。图像的基本特征在浅层卷积层中处理,图像的抽象特征在深层卷积层中处理。在算法模型中,卷积层的运算量是最大的,并且可以视为是全并行的计算,故卷积层是优化加速的关键。卷积层的计算公式如公式(1)所示:

$$z_i^l = f(\sum_{j \in a_i} z_j^{l-1} * k_{ji}^l + b_i^l) \quad (1)$$

式中, l 为卷积层数, k 为卷积核, a_i 为输入特征谱的一个选择, b 为偏执参数。

池化层主要用于合并各层中相似的特征信息。常

用的池化结构分为平均池化层和最大池化层两种,在神经网络结构中,池化层取最大池化,即计算一个特征

映射中部分区域的最大值,最大池化可使得失真的概率降低,最大池化如图 2 所示。

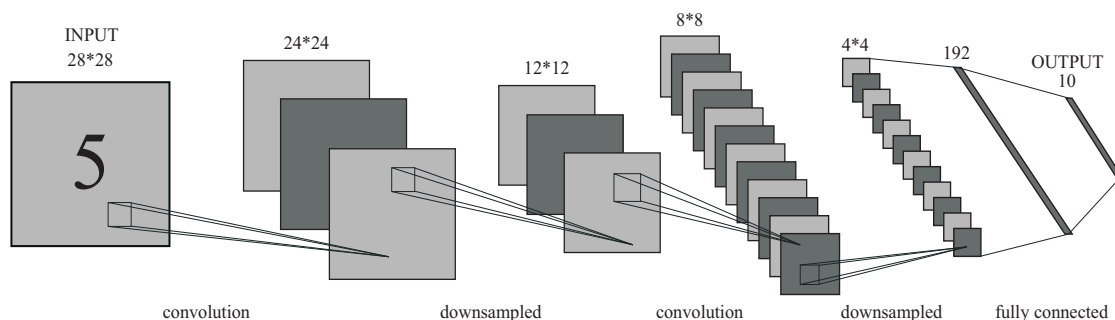


图 1 手写识别模型结构

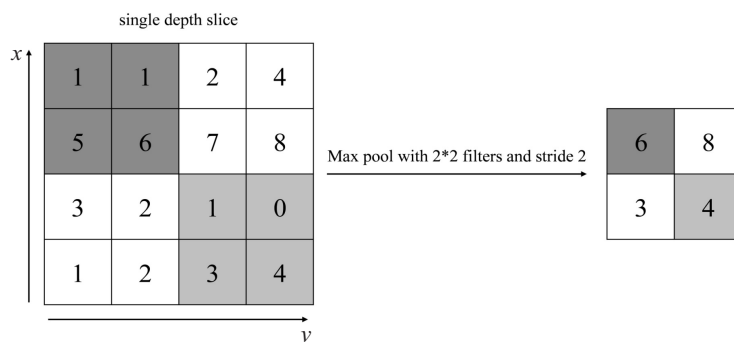


图 2 最大池化层计算模型

使用激活函数可以让神经网络具有非线性识别功能,极大提升了神经网络的表达能力。在激活函数的选择上,sigmoid 激活函数在梯度下降中容易出现过饱和、造成终止梯度传递,且没有 0 中心化。为了解决这个问题,在神经网络中使用另外一个激活函数:ReLU。

ReLU 函数的全称为 Rectified Linear Units。表达式如下:

$$f(x) = \max(0, x) \quad (2)$$

因为没有时间开销巨大的幂运算,运用 ReLU 函数作为激活函数的神经网络具有收敛速度快、求梯度简单的优点。

1.2 网络搭建及训练

针对 MNIST 数据集的卷积神经网络搭建是在 TensorFlow 中进行^[14]。

MNIST 数据集是一手写字数字数据集,包括了训练样本、训练样本标签、测试样本、测试样本标签四部分数据。MNIST 数据集常用于各种模型的训练及测试,该文的卷积神经网络模型正是基于 MNIST 数据集。

TensorFlow 是一种流行的深度学习框架,主要包括了 TensorFlow 核心库以及配置、部署等软件。TensorFlow 方案方便快捷易用性强,被广泛地应用于机器学习领域。TensorFlow 提供了数据整合、数据处理、搭建及训练机器模型、模型存储其参数量化等服务。该文的卷积神经网络搭建,基于 TensorFlow 并遵

循如下设计。

卷积层选择使用步长为 1,边距为 0 的模板,输入经处理得到的输出保证大小一致,池化层选择步长为 2 的 2x2 大小的最大池化模板。

搭建的第一层卷积层由一个卷积接一个最大池化层完成。卷积在每个 3x3 的 patch 中计算得到 16 个特征,权重张量为 [3, 3, 1, 16],对每一个输出通道都有一个对应的片质量,将偏执量设定为 32。接着处理图片,重塑为 28x28 的格式,将卷积和与重塑后的图片进行卷积处理,经过 ReLU 激活函数处理后进入最大池化层。

第二层相比第一层将得到 32 个特征,权重张量为 [3, 3, 16, 32]。在前述步骤中,图片的格式为 28x28,池化层为 2x2 的参数设定,因此经过两次池化层后的图片尺寸变为 7x7 像素。现在经一个参数为 128 神经元的全连接层处理,生成一个向量,将向量 reshape 为一维数组,与权重 W_{fc1} 相乘后与偏执 b_{fc1} 相加,通过 ReLU 激活函数处理。为了尽量减小过拟合问题造成的不良影响,在输出层后添加 Dropout^[15]。

最后使用全连接层,通过 Softmax Regression 得到识别结果。

在训练模型前需要定义损失函数 (Loss Function),选择交叉熵函数 (Cross-Entropy),交叉熵函数如公式(3)所示。

$$H_y(y) = - \sum_i y_i' \log(y_i) \quad (3)$$

计算交叉熵后,就可以使用梯度下降来优化参数。由于已部署好网络结构, TensorFlow 可以使用反向传播算法计算梯度,自动地优化参数,直到交叉熵最小。TensorFlow 提供了多种优化器,该文选择 Adam 优化器来做梯度最速下降,学习率 0.000 1。每次训练随机选择 50 个样本,加快训练速度,每轮训练结束后,计算预测准确度。实验得出,预测正确率可以达到 98% 以上。

由于需要在 PYNQ 开发板上部署模型,故需要记录卷积层、池化层、全连接层等详细参数,其中数据的先后读取顺序必须保持顺序相同,针对不同维度的数据,由维度从高到低存储,以便在硬件层运行正常。

2 设计与优化

2.1 HLS 设计

卷积神经网络性能优异,但计算量庞大。图像识别技术往往对数据处理的实时性有严格的要求。FPGA 属于专用集成电路中的一种半定制电路,是可编程的逻辑序列。其并行化的结构特点十分适合应用于图像处理,极强的灵活性方便系统的维护、移植、升级及扩展,较低的功耗满足于移动设备的耗能限制。但对于不了解硬件设计的工程师来说,传统的 FPGA 设计需要熟练掌握 VerilogHDL 硬件编程语言。因此,赛灵思公司推出了 Vivado 软件的对应套件 Vivado HLS 开发工具, HLS 是高层次综合 (High - Level - Synthesis) 的缩写, HLS 的主要功能是将 C 语言、C++ 等 C 规范语言转换为可以由 Xilinx 旗下 FPGA 实现。HLS 的出现大大降低了非专业硬件工程师的设计难度和开发周期,也减少了工程师进行 FPGA 开发的学习成本,并提高了软件设计人员所设计系统的性能表现。

软件开发人员可以在 FPGA 上加速他们算法中计算密集的部分。与传统设计模式相比,利用 Vivado HLS 开发套件可以更快速地验证硬件描述语言设计

的功能正确性。与 VerilogHDL 和 VHDL 这样的硬件描述语言相比, C/C++ 这样的高级编程语言拥有更好的可读性和使用基数。HLS 套件可以根据默认的行为、约束条件以及开发人员自定义的优化设置来对高级语言所编写的程序进行硬件层面的综合。可以在程序相同的情况下综合生成不同的硬件实现方案,通过优化指令自定义修改控制电路内部逻辑和 I/O 端口。为了确定 HLS 套件综合生成的设计是否满足性能要求和资源限制, HLS 会在综合结束后自动生成性能资源报告,可查看综合得到的 RTL 级别模块的响应性能及占用资源量。

如果采用传统的软件开发思路来设计 HLS 程序,会造成大量冗余, HLS 套件会针对每一个函数都生成一份电路,在层数较多的网络中就会造成大量电路出现。与此同时,如果后期进行删减添加网络层或者实现其他功能,都需要重新生成综合电路,不仅低效而且不具备通用性,这与现在的开发模式是背道而驰的。

通过对 MNIST 的代码实现分析可知,在 MNIST 网络中,存在卷积运算、ReLU 运算、池化运算和全连接运算。Softmax 可以视为全连接运算的一种特殊形式,其中 ReLU 函数原理较为简单,而全连接层也可以看作是卷积层的一种特殊形式,它的核大小正好等于输入的空间尺寸。因此,实现一个具备灵活性的通用卷积电路只需要实现两种通路,即两种分别是通用卷积运算的电路和通用池化运算的电路。PYNQ 的 PS 端 (ARM 端) 所搭载的 CPU 作为主控来调度电路,卷积与 ReLU 运算和池化运算为两条数据通路,通过 CPU 可以配置通路参数。

常见的 CNN 中,各个网络函数的数据均为三维数据,即多个矩阵叠加,为了实现各层间共享加速模块的设计要求,需要将各层的输入输出数据以及 kernel 的步长以变量形式进行传递。卷积层处理过程如图 3 所示。

```

define Cnn1(
    float In[5][32][32],
    float Out[5][28][28],
    float W[5][5][3][3] )
for kr = 0 to 2 do
    for kc = 0 to 2 do
        for r = 0 to 27 do
            for c = 0 to 27 do
                for cho = 0 to 4 do
                    for chi = 0 to 4 do
                        float tmp = In[chi][r + kr][c + kc] * W[cho][chi][kr][kc]
                        Out[cho][r][c] += tmp

```

图 3 卷积层伪代码

经过 HLS 综合后得出的性能参数如图 4 所示。

Loop Name	Latency		Iteration Latency	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
-Loop 1	2378400	2378400	792800	-	-	3	no
+Loop 1.1	792798	792798	264266	-	-	3	no
++Loop 1.1.1	264264	264264	9438	-	-	28	no
+++Loop 1.1.1.1	9436	9436	337	-	-	28	no
++++Loop 1.1.1.1.1	335	335	67	-	-	5	no
+++++Loop 1.1.1.1.1.1	65	65	13	-	-	5	no

图 4 性能报告

由图 4 可知,上述设计仿真得到的时钟周期高达 2 378 400,对于 HLS 设计来说,需要有针对性地进行优化加速设计。

2.2 HLS 设计优化

软件层面的设计若未经优化则直接综合生成串行电路,即同一时间只能进行单路运算,无法发挥并行计算的优势。未经优化的 HLS 设计等效于串行结构,对于实例的卷积层电路来说,需要六次轮巡计算,理论上优化后可以最高提升六倍的性能。但 HLS 的

UNROLL 优化方法对循环的展开能力是有限的,且性能提升受工作频率及数据耦合的限制,故难以达到理论值。

进行优化设计,可在 Directive 中添加指引,通过 UNROLL 将循环运算展开为并行运算,进行综合后的参数得到了提升。进一步地,为了提高数据输入时的读取速度,再对输入数据进行优化,将输入数据多维度展开,综合生成的电路性能参数如图 5 所示。

Loop Name	Latency		Iteration Latency	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
-Loop 1	226320	226320	75440	-	-	3	no
+Loop 1.1	75438	75438	25146	-	-	3	no
++Loop 1.1.1	25144	25144	898	-	-	28	no
+++Loop 1.1.1.1	896	896	32	-	-	28	no

图 5 性能参数

HLS 的默认形式是不进行流水化处理的,默认综合生成的电路为最节省资源的配置,使用 pragma PIPELINE 约束将卷积层进行流水化处理,PIPELINE 将自动将循环内部的所有子循环展开,因此可以取代上

述 UNROLL 的设置。经过流水优化后的性能报告如图 6 所示。由图 6 可知,经过上述步骤进行优化后,卷积层的时钟周期从 2 378 400 缩短至 7 088,即加速了 335.55 倍。

Loop Name	Latency		Iteration Latency	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
-Loop 1	7088	7088	34	1	1	7056	yes

图 6 性能参数

上述的卷积层 HLS 是在卷积模型结构参数固定的情况下设计的,所使用的优化加速方法,如循环展开、多维展开、流水化处理等操作都需要在模型参数确定的条件下才可以使用。当在 PYNQ 上部署的是针对某种特定边缘计算场景的系统时,确实可以设计参数固定的卷积模型,但实际应用中,只能实现某一种单一功能的终端设备是很少见的,其设计成本高且不具备通用性。

为了设计一个通用性的卷积层 IP,需要将原先设计中的各个权重参数作为变量来进行编程。VIVADO HLS 对 C 语言以及 C++都做了相应的扩展,允许工程师在设计中自定义任意精度的浮点数据类型和定点数据类型,在卷积层的设计中使用 ap_uint 定义位数精度

即可。在适用不同场景时,通过参数调整便可实现卷积层 IP 的复用。

为了实现泛用性,实现在 PYNQ 的 linux 系统直接传参,可以进行端口类型约束指引,通过 INTERFACE 控制电路的启停,选择 s_axilite 总线方式,生成 AXI 总线接口。在此基础上,可以将数据的输入和读取同样采用 AXI 总线传输方式处理,同时添加 INTERFACE 指引,选择 m_axi 方式实现数据的自动读取。

3 结束语

针对主流深度学习算法 CNN 中计算量最大的卷积层,提出了在赛灵思高层次综合开发工具环境下的

设计思路及优化方法,利用 HLS 的特点及优势设计了参数固定的卷积模块以及具备通用性的卷积模块,讨论了 FPGA 实现深度学习算法部署的设计思路和加速方法。经验证,部署于 FPGA 的卷积神经网络模型性能良好,能通过参数传输的方式实现针对不同数据的通用。该方法为图像识别技术提供了一种卷积神经网络系统的设计思路,实现了卷积层模块的通用性设计。

参考文献:

- [1] 张琦,张荣梅,陈彬. 基于深度学习的图像识别技术研究综述[J]. 河北省科学院学报,2019,36(3):28-36.
- [2] 杨雨诺,张国林,孙科学,等. 基于深度学习网络的心音智能分析平台构建[J]. 计算机技术与发展,2019,29(7):130-134.
- [3] 郑远攀,李广阳,李晔. 深度学习在图像识别中的应用研究综述[J]. 计算机工程与应用,2019,55(12):20-36.
- [4] SHIN D, LEE J, LEE J, et al. 14.2 DNPU: an 8.1TOPS/W reconfigurable CNN-RNN processor for general-purpose deep neural networks[C]//2017 IEEE international solid-state circuits conference (ISSCC). San Francisco, CA, USA: IEEE, 2017: 240-241.
- [5] ZHANG Y, LI P, WANG X. Intrusion detection for IoT based on improved genetic algorithm and deep belief network[J]. IEEE Access, 2019, 7: 31711-31722.
- [6] DENG W, LIU H, XU J, et al. An improved quantum-inspired differential evolution algorithm for deep belief network[J]. IEEE Transactions on Instrumentation and Measurement, 2020, 69(10): 7319-7327.
- [7] ZHAO Z, ZHENG P, XU S, et al. Object detection with deep learning: a review[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(11): 3212-3232.
- [8] 渠吉庆,陈禹,刘玉琪,等. 基于语音识别智能家居系统的设计与实现[J]. 计算机技术与发展, 2020, 30(12): 148-152.
- [9] ZHANG C, SUN G, FANG Z, et al. Caffeine: toward uniformed representation and acceleration for deep convolutional neural networks[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2019, 38(11): 2072-2085.
- [10] MEDUS L D, IAKYMCHUK T, FRANCES-VILLORA J V, et al. A novel systolic parallel hardware architecture for the fpga acceleration of feedforward neural networks[J]. IEEE Access, 2019, 7: 76084-76103.
- [11] FUJITA N, KOBAYASHI R, YAMAGUCHI Y, et al. Parallel processing on FPGA combining computation and communication in OpenCL programming[C]//2019 IEEE international parallel and distributed processing symposium workshops (IPDPSW). Rio de Janeiro, Brazil: IEEE, 2019: 479-488.
- [12] 吴艳霞,梁楷,刘颖,等. 深度学习 FPGA 加速器的进展与趋势[J]. 计算机学报, 2019, 42(11): 2461-2480.
- [13] 陈超,齐峰. 卷积神经网络的发展及其在计算机视觉领域中的应用综述[J]. 计算机科学, 2019, 46(3): 63-73.
- [14] 邢艳芳,段红秀,何光威. TensorFlow 在图像识别系统中的应用[J]. 计算机技术与发展, 2019, 29(5): 192-196.
- [15] 宋晓茹,吴雪,高嵩,等. 基于深度神经网络的手写数字识别模拟研究[J]. 科学技术与工程, 2019, 19(5): 193-196.