DOI:10.19651/j.cnki.emt.1802440

基于 SoC 的卷积神经网络系统设计*

李子聪[1] 曾宇航[1,2] 熊晓明[1]
(1. 广东工业大学 自动化学院 广州 510006; 2. 佛山芯珠微电子有限公司 佛山 528200)

摘 要: 近些年,卷积神经网络(CNN)出色地完成了许多机器视觉任务。但现有的软件实施方案无法很好地在便携式设备中实现,为此设计一种基于 Xilinx 全可编程 SoC 的 CNN 系统,在固定资源的 SoC 平台下,只需较少资源即可实现快速的检测系统。系统实现多级流水线和输入数据复用的方法提高计算效率。系统硬件部分实现 CNN 计算,软件实现图片预处理及图片检测后处理,从而提高运行效率,系统可实现多种卷核的卷积操作,平均值池化,非极大值抑制抑制算法,实现图片中多人脸的准确定位。实验结果表明,在 100 MHz 的工作频率下,系统的平均计算速率为 0.19 Gops/s,功耗仅为通用 CPU 的 4.07%。

关键词: SoC;卷积神经网络;并行化;软硬件协同设计
中图分类号: TP919;TN7 **文献标识码:** A **国家标准学科分类代码:** 510.4030

Design of convolutional neural network system based on SoC

Li Zicong[1] Zeng Yuhang[2] Xiong Xiaoming[1]
(1. School of Automation, Guangdong University of Technology, Guangzhou 510006, China;
2. Chipeye Microelectronics foshan Ltd., Foshan 528200, China)

Abstract: In recent years, convolutional neural networks have done a great job in many machine vision tasks. However, existing software implementations are not well implemented in portable devices. A convolutional neural network system based on Xilinx all-programmable SoC is designed to accelerate the convolutional operation in parallel, which only need few design resource and implement fast detection system. The system uses multi-stage pipeline technology and input data reuse to improve calculation efficiency. The hardware part completes convolutional network calculation, and the software part finish the image preprocessing and post-image detection preprocessing, thereby improving operation efficiency. The system can implements the convolution operation with different size, mean pooling operation and the non-maximum suppression algorithm, which achieves accurate positioning of multiple faces in the picture. The experimental results show that the average calculation rate of the system is 0.19 Gops/s at the operating frequency of 100 MHz, and the power consumption is only 4.07% of the general purpose CPU.

Keywords: SoC; convolutional neural network; deserialize; hardware software co-designed

0 引 言

卷积神经网络(convolution neural network, CNN)是一种深度前馈人工神经网络,该网络避免了对图像复杂的前期预处理,可直接输入图像或视频至网络中,因而得到广泛应用,特别是在机器视觉和 AI。由于 CNN 的特征检测层通过训练数据集进行学习,因此不需要显示特征的提取,再者,同一特征图的神经元共用相同的权值。CNN 以局部权重共享在图像处理具有独特的优越性,共享权重降低了神经网络的复杂度,降低了网络的计算量。目前 CNN 在视频监控[1]、机器视觉、模式识别、图像分类[2]、深度图片超分辨率重建[3]等领域广泛应用。

CNN 目前主要通过 CPU 或 GPU,软件方式实现,但是都存在一定的缺点,通用处理器串行处理数据,无法利用 CNN 的并行处理优势。GPU 并行计算能力强,但是体积大,功耗高,需要散热器。文献[4-11]采用基于 FPGA 的神经网络加速器,实现手写数字字符识别,网络结构简单,数据量小,可配置性低。但最新的 CNN 模型用于大数据集的分类,具有复杂度高,计算量大的特点,网络结构复杂,需要一定的可配置性,数据及权重都存储在外部存储器,且需

收稿日期:2018-11
* 基金项目:广东省科技项目(2017B090909004)资助。

要进行片内外数据传输。文献[12]采用基于 FPGA 的神经网络加速器,通过剪枝技术生成系数的神经网络,提高计算效率,但降低了通用神经网络的可配置性。

本文分析 CNN 算法,研究网络潜在的并行性,在 SoC 的可编程逻辑(programable logic,PL)设计一种可配置的 CNN 硬件 IP,并通过 ZYNQ 的处理系统(processing system,PS)处理器控制数据流,输入数据复用技术减少片内外数据传输,利用动态定点数量化数据,软硬件协同设计,硬件上实现 CNN 算法,软件上实现后处理,最后对图片中定位人脸,并用方框显示出来。

1　CNN

CNN 主要用于识别位移、缩放及其他形式扭曲不变形的二维图形。作为一种经典的监督学习算法,CNN 采用前向过程进行识别,反向传播进行训练。在工业界的应用场景中,设计人员离线训练 CNN,并利用离线的 CNN 执行相关的识别任务。因此,我们专注于搭建完整的离线 SoC 的识别系统。

目前,CNN 已成为热门的机器视觉研究热点。典型的 CNN 由特征提取器和分类器 2 部分组成。特征提取器用于将输入图像经过卷积计算后表示为包含各种特征的特征图,经过多层的卷积层提取更高级的特征。特征提取器的输出包含这些特征的矢量,将矢量输入到分类器中确定输入属于的类别的可能性[12-13]。

一个经典的 CNN 由多层计算构成,包括卷积层,池化层和激励层。网络用卷积核提取输入图片或输入特征图的特征,第 1 层提取的是图片边缘信息,层数越多提取的特征越高级。池化层夹在连续的卷积层中间,用于压缩特征图和参数的量,减少过拟合,常用的方法有平均值池化(mean-pooling)和最大值池化(max-pooling)。激活层是将输入数据做非线性映射,常用的有 sigmoid,ReLU(the Rectified Linear Unit),tanh 函数。激活层引入非线性因素,解决线性函数不能解决的问题,且在数学上可微。

如图 1 所示,一个经典的卷积层输出特征图为 $M\times R\times C$。卷积核为 N×M×k×k,s 为卷积的步长,k 为卷积核的长宽,通过计算可得到输入特征图的大小为 $N\times(C\times s+k)\times(R\times s+k)$。$M$ 表示为输出特征图的通道数;R、C 为输出特征图的长宽;N 为输入特征图的通道数;H,L 为输入特征图的长宽。

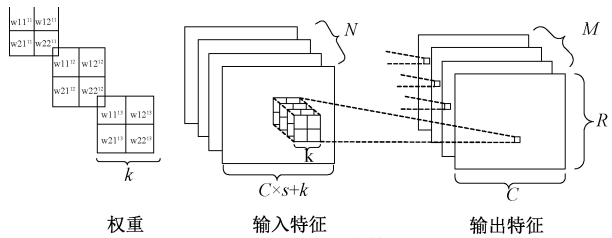

图 1　卷积计算

同样卷积计算用式(1)[15-16]表示:

$$O[m][r][c]=f\left(b[m]+\sum_{n=1}^{N}\sum_{i=1}^{k}\sum_{j=1}^{k}I[n][s\times r+i][s\times c+j]\times W[m][n][i][j]\right)$$

$$0<m\leqslant M,0<r\leqslant R,0<c\leqslant C \tag{1}$$

式中:M 表示输出特征图的数目;N 表述输入特征图的数目;R、C 表述输出特征图的长宽。

本文采用一种 7 层的 CNN 用于人脸定位检测。本文实现的一个 CNN 的整体架构如图 2 所示。

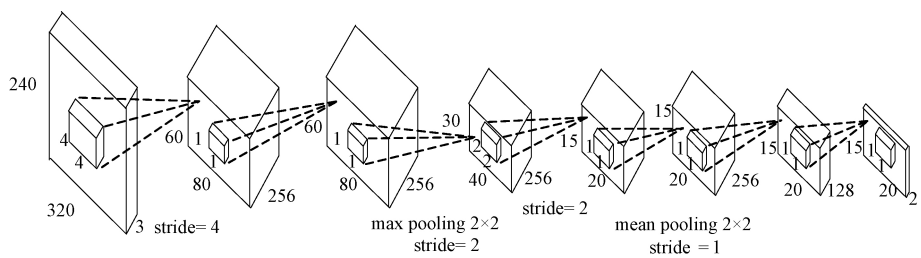

图 2　CNN 结构

整个系统由图片预处理,CNN 和分类后处理 3 个部分组成。图片预处理主要是将图片减去一个训练得到的均值,以及将像素值转化为定点数的表示。分类后处理包括将定点数表示的特征图转化为实际值,并进行 softmax 分类,以及进行非极大值抑制算法。

第 1 卷积层输入为 3×320×240 的 RGB 图像。卷积核大小为 4×4,卷积步长为 4,输出特征图大小为 80×60。第 1 层共有 256×80×60=1 228 800 个神经元,80×80×4×4×3×256=58 982 400 条连接。

第 3 卷积层输入特征图大小为 256×80×60。卷积核大小为 1×1,步长为 1,共有 256×256 个卷积核,卷积计算输出 256 个大小为 80×60 的特征图。再经过一个 2×2 的最大值下采样,步长为 2,因此第 3 层最后的输出为 256×40×30 的特征图。

第 5 卷积层输入特征图大小为 256 个大小为 20×15 的特征图,卷积核大小为 1×1,步长为 1,卷积计算输出为 256 个 20×15 的特征图,再经过一个 2×2 的均值下采样,步长为 1,得到 256 个 19×14 的特征图,为实现硬件上的复用,因

此增加第 20 列和第 15 行，并填 0。由于第 6 卷积层和第 7 卷积层的 $k=1$，因此输出不影响 19×14 的特征区域。

网络所使用的激活函数如式(2)所示。

$$f(x)=ReLu(x) \tag{2}$$

CNN 第 7 层卷积层计算完成后，经过 softmax 函数进行分类。Softmax 函数如式(3)所示。

$$\Phi_i=\frac{e^{\eta_i}}{\sum_{j=1}^{k}e^{\eta_i}} \tag{3}$$

2　系统实施方案

本文对计算架构研究重点是 CNN 计算架构，软件硬件的协同设计，数据复用，存储系统及数据位宽。系统可分为 CPU 软件编程部分，和硬件可编程逻辑部分。CPU 软件编程部分完成图像操作，以及 softmax 分类器，非极大值抑制算法。硬件可编程部分实现 CNN 计算部分。

2.1　CNN 计算架构

卷积操作在不同层之间具有高度的相似性，可复用乘加计算单元，且在同一层的卷积操作可做数据重用减少带宽及功耗。

系统硬件支持 1×1，2×2，4×4 的卷积计算，支持最大值池化，支持平均值池化，以及 ReLU 函数。乘加计算模块，最大值模块，拥有配置寄存器，可配置支持多种大小的卷积操作。平均值池化可复用卷积计算模块，只需把权重设为 1/4 即可。

数据传输如图 3 所示，分为 6 个部分，数据获取，多个 PE 卷积计算，激活函数，最大值池化，结果写入片内存储以及数据写回。数据传输在时序上采用多级流水的方式提高数据的吞吐率。

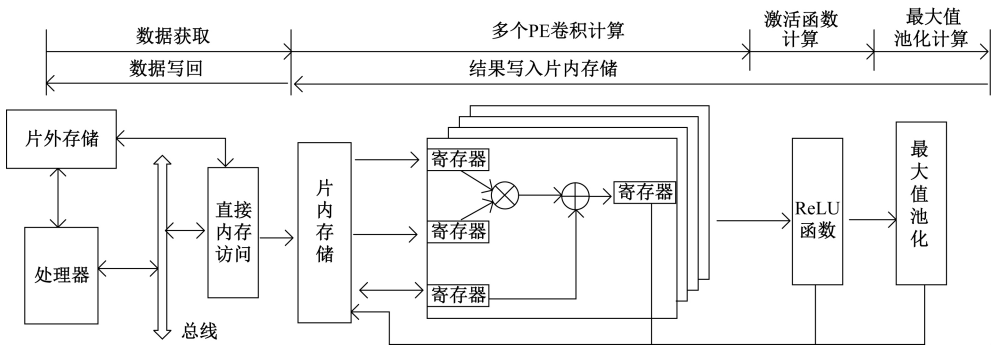

图 3　单层数据传输

在图 3 中，DDR3 SDRAM 为主要存储，存放着网络中的所有参数、所有卷积核及相应的大小、输入图片。在数据获取阶段，系统软件控制从 DDR3 SDRAM 中读取相应数据，通过 zynq PS 的 AXI HP 端口高速地将数据传输到直接内存访问(direct memory access，DMA)中，数据经过 FIFO 传输到 input feature map Block RAM、weight Block RAM 和 output feature map Block RAM。若卷积计算为某一层 $n=1$，即该层的第 1 次卷积计算，则没有数据流入 output feature map。

多个 PE 卷积计算阶段，Block RAM 传输到 REG 中，进而传输到 PE 中。PE 可同时完成 16 个乘法运算。

激活函数计算阶段或最大值池化阶段，此时 $n=N$，此时数据来自多个 PE 卷积计算的结果，网络支持多种激活函数，如 ReLU 函数，sigmoid 函数。

网络参数决定卷积计算结果是写回 Block RAM，还是乘加计算单元输出结果是否经过最大值池化，和 ReLU 函数。完成计算后写回至 Output Buffer 中，得到 AXI DMA 将数据写回到 DDR3 SDRAM 中。由于 SoC 内部资源有限，不能够将所有中间值存储在片内，因此中间值需要在下一次乘加运算时，从 DDR3 SDRAM 读出写到 Output Buffer 中，如此往复，直到单层计算结果完成，才能进入到下一层。

2.2　软硬件协同设计

系统初始化后，CPU 将存储在 SD 卡中的权重数据读取 CPU 中。读取图片信息，并存储在片外存储 DDR3 SDRAM。DDR3 SDRAM 为主要存储，存放着网络中的所有参数，所有卷积核及相应的大小，输入图片。CPU 软件控制 DDR3 SDRAM 中的数据存储位置及相应的读写。

CPU 通过 AXI 总线控制 DMA，DMA 需要完成 2 种操作：

1)从 DDR3 SDRAM 中读取输入特征图，权重，偏置和计算中间结果至 CNN 计算模块，以及读取相关的参数，配置 CNN 计算模块。完成读取数据操作产生一次中断。

2)将计算单元计算得到的中间结果写回到 DDR3 SDRAM，传输完成产生中断，准备开始下一次的读取数据的工作。

系统流程如图 4 所示，当 CPU 控制 DMA 数据传输到 CNN 中，CPU 等待 CNN 完成，并产生一个中断，进而进行数据写回及进行分类后处理。

2.3　数据复用

数据复用方式及存储系统的性能很大程度上影响运算单元的计算效率。为满足 CNN 计算单元的并行计算的

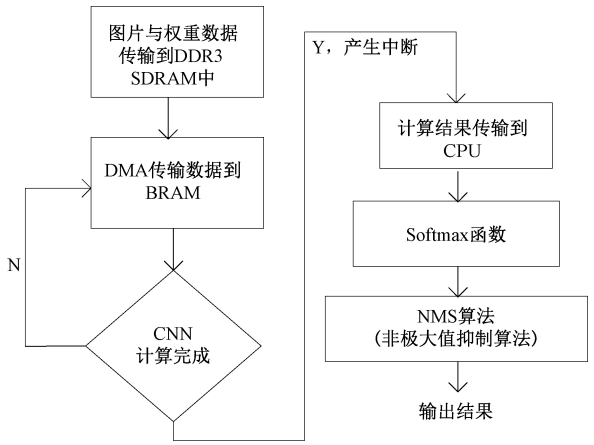

图 4 系统流程

卷积需求,减少片内外存储的数据交互,本设计采用输入特征图复用及三级存储方式。尽管设计中采用多通道并行计算,带宽仍然限制着计算效率。1)从 SDRAM 中读取输入数据至 Block RAM 需要高带宽,同时提高了功耗;2)计算中产生的大量中间值也对存储提出挑战。

由于片内资源有限,将 CNN 分片在 SoC 内运行,Th,Tl,Tn 分别表示为 SoC 内运行的输入特征块的长、宽及通道数。Tr,Tc,Tm 分别表示 SoC 内输出特征块或中间值的长、宽及通道数。$I=Th\times Tl\times Tn$,$O=Tr\times Tc\times Tm$,而 $Th=(Tr-1)s+k$,$Tl=(Tc-1)s+k$,$W=Tn\times Tm\times k\times k$,其中 s,k 为卷积操作的步长与大小。使用式(4)计算得到整个 CNN 片内外传输的数据量[12-13]:

$$Ac=I\times\beta_i+O\times\beta_o+W\times\beta_w+P\times\beta_p \tag{4}$$

式中:P 表示池化输出的结果,通常池化大小为 2×2,因此 $\beta_p=\frac{1}{4}$。

根据复用方式可将 CNN 硬件实现分为 3 种:

1)卷积核复用:由于 CNN 权重共享的属性,一部分输入特征图共享权重数据。在同一通道的输入特征图中,每一个权重复用 $C\times R$ 次。但全连接层无法使用卷积核复用模式。

$$\beta_i=\left\lceil\frac{M}{Tm}\right\rceil\beta_o=2\left(\left\lceil\frac{N}{Tn}\right\rceil-1\right)\beta_w=1 \tag{5}$$

2)输入特征图复用:在输入特征图复用中,每个输入特征图都会被完全利用后,才输入下一批的输入特征图。无论是在卷积计算还是在全连接计算中,每一个输入像素在一层内会被复用 M 次。

$$\beta_i=1\beta_o=2\left(\left\lceil\frac{N}{Tn}\right\rceil-1\right)\beta_w=\left\lceil\frac{H}{Th}\right\rceil\left\lceil\frac{L}{Tl}\right\rceil \tag{6}$$

3)输出特征图复用:输出特征图复用有 3 个特征图:(1)多个通道的输入特征图连续输入到计算单元;(2)计算得到的中间值不输出,存储在寄存器中;(3)最后的结果是经过激活函数和池化计算,否则没有其他结果输出至 SDRAM。

$$\beta_i=\left\lceil\frac{M}{Tm}\right\rceil\beta_o=0\beta_w=\left\lceil\frac{R}{Tr}\right\rceil\left\lceil\frac{C}{Tc}\right\rceil \tag{7}$$

对整 1 个 CNN 片内外传输量计算可得:输入特征图复用比输出数据复用减少 59.8%的片外存储传输量,输入特征图复用比权重复用减少了 79.9%的片外存储传输量,因此较大地减少板级功耗。

2.4 存储系统

存储系统分为三级存储,第 1 级为片外 DDR3 SDRAM,第 2 级为片内 Block RAM,第 3 级为片内寄存器,如图 5 所示。在相同的数据传输量上看,SDRAM 的功耗远大于 global buffer,array 和 RF,因此减少片内外的数据传输,可大大减少板级功耗[16]。

输入特征图采用两块双端口 RAM 存储,权重数据采用 16 块双端口 RAM 保存,不同的输出特征图分别存放在 16 块双端口 RAM。双端口 RAM 能同时读写数据,因此可以对所有的双端口 RAM 进行乒乓操作。在进行卷积计算时,双端口 RAM 分别输出输入特征图,权重数据和中间值,同时从 DDR3 SDRAM 读取数据存储在双端口 RAM 中。

为有效利用 DDR3 SDRAM 的存储,使用 2 个地址 address 1 和 address 2 分别存储输入特征图,输出特征图。如图 5 所示,输入的 RGB 图像存储放在 address0 中,address1 作为第 1 层输出特征图存放的起始地址。第 2 卷积层计算时,address1 为输入特征图存储起始地址,address2 为计算中间值和最终输出特征图存放的起始地址,第 3 卷积层计算时,address2 为输入特征图存储的起始地址,address1 为输出特征图存储的起始地址,以此类推。

2.5 数据位宽

一个定点数可以表示为$(-1)^s\times m\times 2^{-f}$,其中 s 为符号位,m 表示尾数,f 决定了小数点的位置。CNN 不同层,不同的部分有明显的动态范围的区别。通常,输出特征图是数千个结果累加的结果,而网络参数的值却很小。定点只能有限地覆盖一定的范围。而动态定点数可以根据需求改变 f 的值,这在 CNN 的数据量化上能够减少数据损失。

通常卷积后的结果小于输入特征图的数值,为提高数据精度,不同卷积层中本系统数据精度采用动态定点数表示输入特征图,输出特征图和权重。数据用 16bit 有符号表示,第 1 层输入为 320×240×3 的经过预处理的 32 bit 浮点数表示的图像数据,通过式(8)将浮点数转化为定点数,采用 Q7 格式表示,$f=7$,权重采用 Q15 格式表示,$f=15$,得到 Q6 定点表示输出结果为 16 bit,$f=6$,第 2 层输入为 Q6 定点,权重为 Q15,得到 Q7 格式表示输出结果。最后一层计算完成后,将定点数转化为浮点数,在 CPU 中进行 softmax 算法和 NMS 算法。

浮点数 y 转化为定点数 xq:

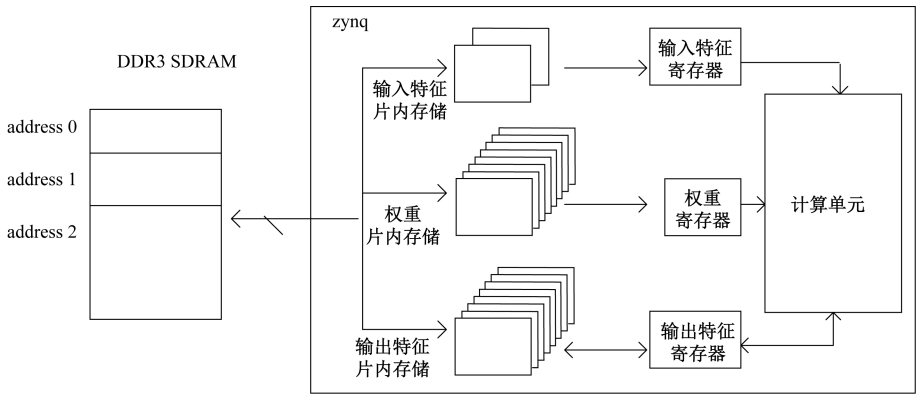

图 5　三级存储系统示意图

$$xq = round(y \times 2^Q) \tag{8}$$

定点数 xq 转化为浮点数 y：

$$y = xq/2^Q \tag{9}$$

3　实验与结果分析

3.1　实验环境

本文实现采用 Xilinx 公司的 vivado 进行硬件开发，Xilinx SDK 进行嵌入式软件开发，在 ZYNQ 系统 ZC706 评估套件上进行板级验证。

整个系统只进行 CNN 的预测过程。CNN 的相关参数需要由上位机训练得到，并传输到下位机中，只需传输 1 次即可，接下来的整个系统的计算控制都由 SoC 完成。每一幅输入的图像均为 240×320 的 RGB 图像，像素值由0～255 的整数表示。像素点在输入后需要进行减去训练得到的平均值，并送入 CNN 中进行计算，最后得到 20×15×2 的特征向量。

3.2　实验结果

SoC 的资源使用情况如表 1 所示。Soc 工作在 100 MHz 下，输入特征图和权重数据全部放在片外 DDR3 SDRAM 中，需要使用时才从 DDR3 SDRAM 传输至片内。由于片内存储着 2 个通道的输入图像，BRAM 的使用率非常高，达到 55.23%。乘法部分运算逻辑全部使用 DSP 资源，使用率为 2.67%。LUT 和 FF 的资源使用率分别为 14.32%，13.67%。显然，SoC 开发板利用率较低，因此 CNN 计算器可在资源少的 FPGA 或可硬件编程 SoC 中使用。当然可进一步提高 DSP 利用率，提高计算性能。

表 1　SoC 资源利用率

模块	占用资源	占用率/%
LUT	31 314	14.32
BRAM	301	55.23
DSP	24	2.67
FF	28 162	13.67

100 MHz 工作频率下，计算一副图像需要 7.97×10^8 次乘累加计算，平均计算速率为 0.199 Gops/s。通过 Xilinx 的 vivado 可以得到 SoC 的运行功耗大致为 2.40 W。通用的 CPU 功耗为 59 W，相比之下，FPGA 的功耗仅为 CPU 的 4.07%。

人脸检测效果如图 6 所示：

图 6　人脸检测效果

4　结　　论

本文提出了一种基于 SoC 的 CNN 系统，针对人脸检测系统，通过软硬件协同设计，不仅实现了 CNN 前向传播过程，还实现了图像预处理和相关后处理。通过输入数据复用，减少片内外数据传输量，提高通过设计调高了并行度，可在单个周期完成 16 次乘累加，通过动态定点数降低数据量化上的损失，大幅调高运行效率。实验表明，该系统的检测精度高，功耗仅为 2.40 W，该系统具有良好的设计应用前景。

参考文献

[1]　黄荷，俞亚萍，张之江.基于神经网络的密集人群视频异常检测[J].电子测量技术，2017，40(11)：103-107.

[2]　崔雪红，刘云，王传旭，等.基于卷积神经网络的轮胎缺陷 X 光图像分类[J].电子测量技术，2017，40(5)：168-173.

[3]　李伟，张旭东.基于卷积神经网络的深度图像超分辨率

重建方法[J].电子测量与仪器学报,2017,31(12):1918-1928.

[4] 余子健,马德,严晓浪,等.基于 FPGA 的卷积神经网络加速器[J].计算机工程,2017,43(1):109-114,119.

[5] 余子健.基于 FPGA 的卷积神经网络加速器[D].浙江:浙江大学,2016.

[6] 王羽.基于 FPGA 的卷积神经网络应用研究[D].广州:华南理工大学,2016.

[7] 李嘉辉,蔡述庭,陈学松,等.基于 FPGA 的卷积神经网络的实现[J].自动化与信息工程,2018,39(1):32-37.

[8] 王小雪.基于 FPGA 的卷积神经网络手写数字识别系统的实现[D].北京:北京理工大学,2016.

[9] 鲁云涛.基于 FPGA 的稀疏神经网络加速器[D].合肥:中国科学技术大学,2018.

[10] 王思阳.基于 FPGA 的卷积神经网络加速器设计[D].成都:电子科技大学,2017.

[11] 周华坤.基于 NOC 结构的卷积神经网络加速器建模[D].西安:西安理工大学,2018.

[12] 杨薇.卷积神经网络的 FPGA 并行结构研究[J].数字技术与应用,2015,(12):51.

[13] 陆志坚.基于 FPGA 的卷积神经网络并行结构研究[D].哈尔滨:哈尔滨工程大学,2013.

[14] CHEN Y H,KRISHNA T,EMER J S,et al. Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks[J]. IEEE Journal of Solid-State Circuits,2017,52(1):127-138.

[15] TU F,YIN S,OUYANG P,et al. Deep convolutional neural network architecture with reconfigurable computation patterns[J]. IEEE Transactions on Very Large Scale Integration Systems, 2017, 25 (8): 2220-2233.

[16] CHEN Y H, EMER J, SZE V. Eyeriss: a spatial architecture for energy-efficient dataflow for convolutional neural networks[J].IEEE Micro,2016,PP(99):1-1.

作者简介

李子聪,研究生,主要研究方向为软硬件协同设计技术,卷积神经网络硬件加速等。

E-mail:lzcng11@163.com