

引用格式:涂凯杰,朱宜炳,许达文. 一种双向脉动数据流的全卷积神经网络加速器[J]. 微电子学与计算机,2020,37(1):33-37. [TU K J, ZHU Y B, XU D W. Fully convolutional neural networks accelerator for Bi-direction systolic data flow[J]. Microelectronics & Computer,2020,37(1):33-37.]

一种双向脉动数据流的全卷积神经网络加速器

涂凯杰¹,朱宜炳²,许达文¹

(1 合肥工业大学 电子科学与应用物理学院,安徽 合肥 23009;

2 江西科技学院 信息工程学院,南昌 江西 330098)

摘 要:全卷积神经网络近年来被应用于深度学习中的多个领域,其不仅能处理简单的图像分类任务,还能应用于例如物体检测、语义/图像分割以及基于生成式对抗网络的生成型任务.典型的全卷积神经网络中不仅包括了传统的卷积层,还有反卷积层,它们都是计算密集型的.现在大多数研究者大都关注卷积层的设计优化,而反卷积的加速优化很少.本文提出了一种双向脉动数据流的全卷积神经网络加速器,可以同时高效地处理普通卷积层以及反卷积层.实验中选取了多个具有代表性的全卷积神经网络模型,例如 DCGAN, Cascaded-FCN 等.相较于以往传统的未优化的加速方案,本文所设计的加速器平均可以达到 2.8 倍的加速比,并且能耗降低了 46.3%.

关键词:全卷积;反卷积层;加速优化;双向脉动数据流

中图分类号:TN492

文献标识码:A

文章编号:1000-7180(2020)01-0033-05

Fully convolutional neural networks accelerator for Bi-direction systolic data flow

TU Kai-jie¹, ZHU Yi-bing², XU Da-wen¹

(1 School of Electronic Science & Applied Physics, Hefei University of Technology, Hefei 23009, China;

2 College of Information Engineering, JIANGXI University of Technology, Nanchang 330098, China)

Abstract: Fully Convolutional Neural Networks (FCN) have been applied in various domains of deep learning in recent years. It can be used not only for simple image classification tasks but also for such as object detection, semantic/image segmentation and generative tasks based on Generative Adversarial Networks (GAN). Typical FCN includes not only traditional convolutional layers but also deconvolutional layers, which are computationally intensive. Most researchers are now focusing on the design optimization of the convolutional layer, but deconvolutional layer receives less attention. This paper proposes a Fully Convolutional Neural Networks accelerator using bi-direction systolic data flow, which can efficiently process the general convolution layer and deconvolution layer simultaneously. Several representative FCN models, such as DCGAN, Cascaded-FCN, etc., have been selected in the experiment. Compared with the traditional unoptimized acceleration scheme, the proposed approach can achieve 2.8X speed up on average, and the energy consumption is reduced by 46.3%.

Key words: fully convolution;deconvolutional layer; accelerated optimization;bi-direction systolic data flow

1 引言

深度学习算法,尤其是深度神经网络近年来在

多个研究领域取得了令人瞩目的成果.在这些算法之中,全卷积神经网络(FCN)^[1]被广泛应用于像素预测、语义分割以及除早期简单分类任务之外的生

成型任务^[2]. 一个典型的全卷积神经网络同时由卷积层^[3]和反卷积层^[4]构成, 其中卷积层是用于提取和压缩输入数据的特征, 而反卷积层则是用于上采样(Up Sample)特征映射, 将其还原为与输入数据相同甚至更高的维度。

虽然卷积层与反卷积层都是计算密集型的, 都是全卷积神经网络的计算瓶颈, 但是卷积层已经在深度学习体系结构等相关领域被深入研究了^[5-6], 而反卷积层受到的关注程度难以与之企及. 当然, 现在也有一些反卷积加速器的设计, 然而其中有的只能实现一些固定配置的反卷积层或特殊算法^[7], 有的为了支持不同的参数配置^[8], 其特殊的架构也无法同时支持卷积层, 并且性能较差. 除此之外, 目前普遍认可的通用解决方案是考虑将反卷积层转化为一般卷积层来处理, 但是这种方法引入了大量冗余操作, 占总操作数的 70% 以上, 极大地降低了加速器的性能. 为了解决以上这些问题, 本文旨在设计一种适用于全卷积神经网络的加速器, 通过在计算阵列中加入双向脉动数据流, 其不仅对于任意反卷积层参数是可配置的, 并且能够同时支持卷积层, 运算性能也远远高于以往传统的加速方法。

2 相关的反卷积计算模式

2.1 反卷积

图 1 所示的原始反卷积算法基本可以分为以下三个步骤:

(1) 输入特征图中的每一个像素都与滤波器中的所有权重值相乘, 生成一组与滤波器尺寸大小相同的矩阵;

(2) 由这些像素与滤波器生成的矩阵用于构成输出特征图, 各个矩阵在输出中的起始坐标为 $(i_h * S, i_w * S)$, 其中 (i_h, i_w) 为相乘的像素坐标, S 为该层的反卷积步长. 需要注意的是, 这些生成的矩阵在输出图中存在叠加的部分, 这些值需要累加起来;

(3) 先前两步的处理会产生额外的数据填充在输出特征图的四周, 这一部分需要被裁剪掉以获得最终的结果. 输出特征图的大小 O_H, O_W 由式(1)和(2)给出, 其中 I_H, I_W 为输入特征图的尺寸, K_H, K_W 为滤波器的尺寸, P 代表需要裁剪的部分。

$$O_H = (I_H - 1) \times s + K_H - 2P \quad (1)$$

$$O_W = (I_W - 1) \times s + K_W - 2P \quad (2)$$

2.2 补零式反卷积

文献[7-8]是专门为某种配置或者算法定制的反卷积加速实现, 而目前最为通用并且也是最为简

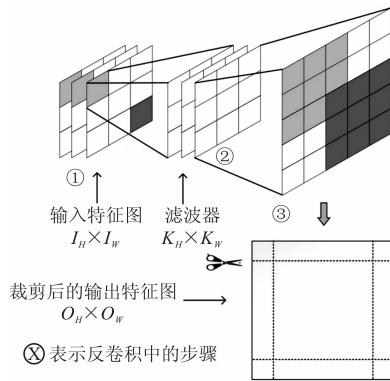


图 1 反卷积的计算过程

单的通用解决方案就是“补零式反卷积”. 该方法通过在输入特征图的边缘以及非零激活值之间填充零值, 将反卷积转换为一般卷积. 其基本思想是在非零激活值之间插入 $(S-1)$ 个零值, 其中 S 为反卷积层原始的步长. 另外, 输入特征图的边缘部分也需要补充零值, 其行数(列数)为 $(K-1)$, K 为滤波器的尺寸, 这里我们假设滤波器为正方形. 在插入零值之后, 新的步长变为 1, 计算则变为一般卷积. 图 2 的示例展示了这一转换过程(原始步长设为 2). 显而易见的是, 这种转换方法引入了大量的冗余零值, 对于这些零值的乘法以及加法操作都是无效的, 不会对输出结果造成影响, 极大地降低了加速器的运算性能, 而且还会增大加速器的存储开销。

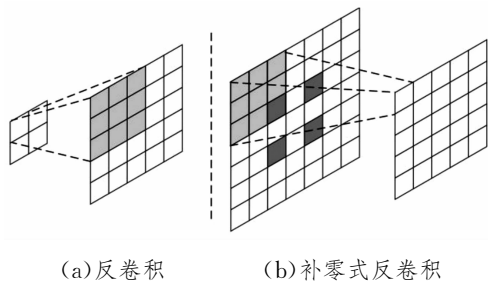


图 2 单张特征图

3 全卷积神经网络加速器

3.1 加速器总体架构

图 3 所示的是本文使用的加速器总体架构框图. 加速器的启动是由一个在主机上执行的编译器平台控制的. 该编译器将根据神经网络模型生成对应的指令集文件, 同时把训练好的权重、偏置参数以及需要处理的图片等转换成十六进制格式, 将其映射到加速器上执行. 加速器主要由以下几个模块构成:

(1) 主控单元(Controller & Decoder). 当指令

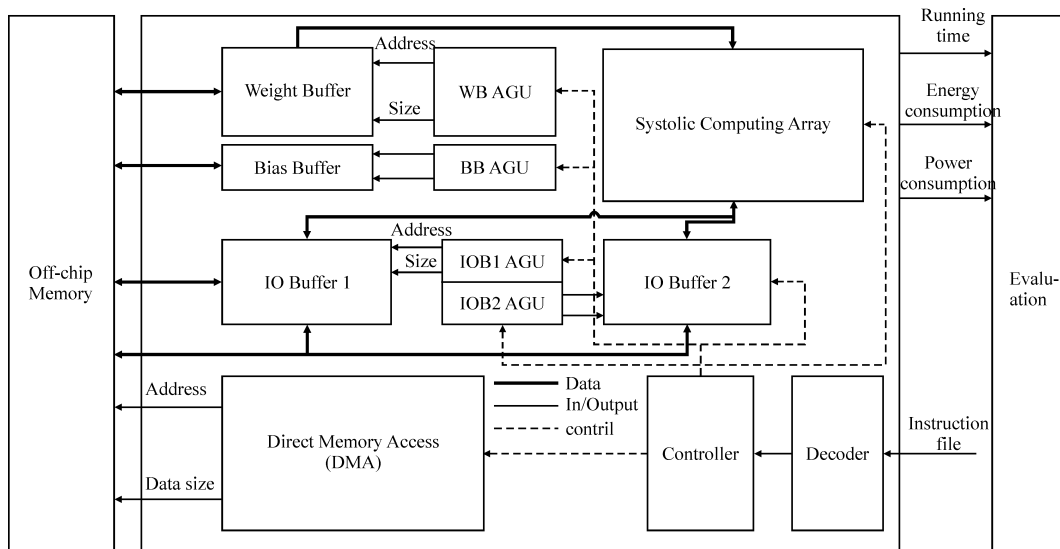


图 3 加速器总体架构

集文件准备好之后,加速器上的译码器将文件中的每一条指令进行解析,再将解析后的信号传给控制器,以调度加速器中的其他模块.

(2)直接内存存取(DMA). 已训练好的神经网络模型参数(包括权重和偏置)以及要处理的图片数据占用的存储空间较大,所以事先都存放在片外存储器(Off-chip Memory)中,DMA 则主要负责片外存储与片上缓存之间的数据交换.

(3) 片上缓存. 这其中包括了多个部分: 权重、偏置和数据的片上缓存 (Weight Buffer, Bias Buffer and IO Buffer 1&2). 卷积层和反卷积层的计算操作中有大量的数据复用, 而如果整个模型的计算都直接在片下存储中读取数据, 资源开销极大. 所以设置了多个片上缓存, 主要用于重复读取一层 (或者特征图的一部分、参数等) 的数据, 这有利于降低能耗, 提升加速器性能.

(4)地址生成单元(AGU).在图3中可以看到,每一个片上缓存都有一个对应的地址生成单元. AGU模块接收控制器传来的信号,按卷积(反卷积)的计算逻辑生成各个片上缓存的数据存取地址.

(5) 脉动计算阵列 (Systolic Computing Array). 各个片上缓存接收地址生成单元的产生的地址, 将对应地址的权重、偏置以及数据传输到计算阵列中. 计算完成的结果再传回片上缓存, 存储地址同样由对应的 AGU 产生.

指令集中的每一条指令顺序执行完毕后,加速器也完成了其任务(例如图像分类等),并且会将该次执行的评估结果,例如执行时间、功耗与能耗结果返回给编译器平台.

3.2 双向脉动数据流

如果将反卷积转换成卷积来处理,就会引入大量的冗余零值;而如果直接采用原始反卷积算法,其第一步生成的矩阵之间存在重叠的部分(Overlapped Region),普通的脉动数据流无法有效地处理这一操作.所以本文提出了双向脉动数据流用以完成这个矩阵区域的叠加操作.图4即为加速器中的计算阵列部分(Systolic Computing Array)的数据流示意图.

计算阵列的基本构成单元称为 PE (Processing Element), 图 4 中所示的阵列为 3 行 3 列, 9 个 PE. 每一个 PE 由一个乘法器以及累加寄存器构成, 其中乘法器将每个时刻接收到的权重值以及输入数据 (激活值, activation) 相乘, 再与之前存放在寄存器中的值累加 (输入特征图通常是多张, 称为多通道, 不同通道的计算结果需累加), 接下来再存回寄存器中. 阵列中的每个 PE 都执行生成一个输出激活值的所有乘法以及加法操作. 由同一列中的 PE 接收到的权重值分别属于不同的滤波器, 其生成的输出激活值是同一像素位置, 不同的输出特征图 (相同 x 、 y 坐标, 不同的 z 坐标); 而同一行中的 PE 生成的输出激活值属于同一特征图 (相同的 z 坐标). 权重值 (weight) 只从阵列中的第一列传入, 正如图中所示, 其从左侧流向右侧, 以达到权重复用的目的; 输入激活值则是直接传入同一列的所有 PE, 乘以接收的权重. 以上这部分数据流可以处理一般的卷积操作. 另外, 每一列 PE 都添加了一个小型片上缓存 (Column Buffer), 它们是另一反向数据流的主要工作模块. 在第二节的反卷积计算模式描述中提到过,

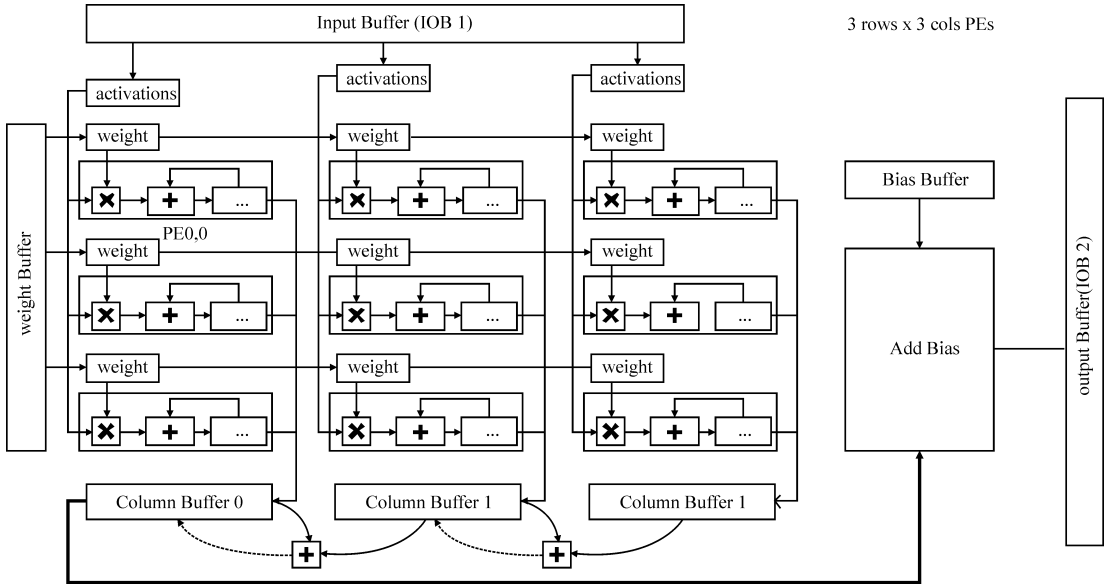


图 4 双向脉动数据流 (3 行 3 列 PE(Processing Element))

由每一个像素生成的矩阵并不是最终的结果,在处理反卷积时,这些矩阵会暂存在相应列的片上缓存中。相邻列的片上缓存之间都存在例如图 1 中所示的需要累加的值,它们由一组额外的加法器累加起来,再存回片上缓存用于下一次的累加(虚线所示)。

算法 1 给出了重构后的反卷积计算的伪代码描述,其中 o_c 和 i_c 分别代表了输出、输入通道, (i_h, i_w) 为输入特征图中的坐标, (o_h, o_w) 则是输出特征图中的坐标, (k_h, k_w) 表示权重值的坐标, S 代表步长。

算法 1: 重构的反卷积

```

for ( $o_c=0; o_c < O_c; o_c++$ )
for ( $i_h=0; i_h < I_H; i_h++$ )
for ( $i_w=0; i_w < I_w; i_w++$ )
//一个 PE 中的操作
for ( $k_h=0; k_h < K_H; k_h++$ )
    for ( $k_w=0; k_w < K_W; k_w++$ )
for ( $i_c=0; i_c < I_C; i_c++$ )
    // PE 中的累加:
     $o[i_h * S + k_h, i_w * S + k_w, o_c] \leftarrow [i_h, i_w, i_c] * w[k_h, k_w, i_c]$ 
//相邻矩阵之间的累加:  $o_h = i_h * S + k_h, o_w = i_w * S + k_w$ 
 $o[o_h, o_w, o_c] += o[i_h * S + k_h, i_w * S + k_w, o_c]$ 
Cropping( $o_h, o_w, P$ )

```

4 实验评估

4.1 实验环境配置

实验中主要有两个比较对象,其一是补零式反卷积,作为基准比较对象(Baseline),另一个就是双

向脉动数据流下重构的反卷积,所提出的设计使用 Verilog 语言实现,并且在 65 nm 工艺库下由 Design Compiler(DC)综合。计算阵列设置为 16 行 16 列,数据位宽都为 8 bit。本文选取了四组具有代表性的神经网络模型进行测试,包括 DCGAN (Large-scale CelebFaces Attributes Dataset)、GP-GAN (Transient Attributes Database)、ArtGAN (Cifar 10 Dataset) 以及 Cascaded-FCN (Biomedical Image)。

4.2 性能以及能耗对比

在这些神经网络模型中,都同时包含了卷积层和反卷积层,主要比较了其中反卷积层的运算性能。由图 5 可以明显地看出,全卷积神经网络加速器中双向脉动数据流下的反卷积运算速度比传统的补零式反卷积快,平均为 2.8 倍。加速比的差异主要来源于输入特征图冗余零值减少的比例,这些与网络参数(例如步长 S , 滤波器尺寸 K)有关,如第二节中描述的一样。同样地,运算性能的提升也能相应地减少能耗,更重要的是片下和片上存储可以不必去存放填充的冗余零值了。值得一提的是,双向脉动数据流需要给每一列 PE 添加小型片上缓存,但这部分缓存带来的能耗开销远远不及我们通过去除冗余零值而减小的能耗开销,图 6 所示的能耗对比充分说明了这一点。

4.3 硬件开销

表 1 所示的为加速器中引入了双向脉动数据流前后的硬件开销对比。在计算阵列中添加了小型片上缓存后,功耗和面积略微有所增加,分别为 6.7%

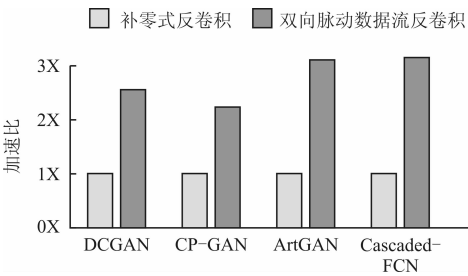


图 5 运算性能对比

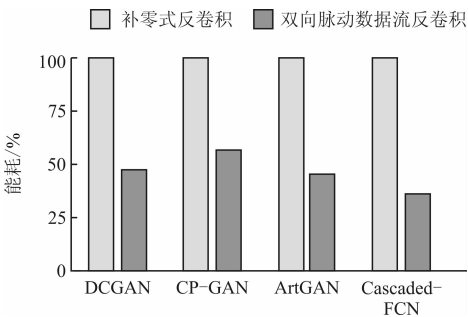


图 6 能耗对比

和 7.3%。在能耗对比中这一部分硬件开销的增大对于加速器的整体性能提升是没有影响的。

表 1 硬件开销对比

	一般卷积数据流	双向脉动数据流
片上缓存/KB	248	264
功耗/W	0.30	0.32
面积/mm ²	1.79	1.92

5 结束语

本文针对目前深度学习中关于全卷积神经网络过的加速问题,设计了一款可以同时支持传统卷积层以及反卷积层的加速器,其在阵列中添加的小型片上缓存,引入的双向脉动数据流可以有效地处理反卷积计算过程中相邻矩阵的重叠累加部分,解决了以往只能通过输入特征图填充大量的冗余零值来处理反卷积层的问题。实验结果表明,使用双向脉动数据流的全卷积神经网络加速器对于运算性能的提升十分显著,同时也有效地降低了能耗开销与存储开销。本文选取了从数据流这一方向入手来处理反卷积,未来还可以考虑在算法上,或是硬件设计上去进一步优化反卷积的加速,以获得更好的性能。

参考文献:

[1] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. Boston, Massachusetts, 2015: 3431-3440.

[2] RADFORD A, METZ L, CHINTALA S. Unsupervised representation learning with deep convolutional generative adversarial networks [J]. arXiv preprint arXiv:1511.06434, 2015.

[3] LECUN Y, BENGIO Y. Convolutional networks for images, speech, and time series [J]. The handbook of brain theory and neural networks, 1995, 3361 (10): 1995.

[4] ZEILER M D, KRISHNAN D, TAYLOR G W, et al. Deconvolutional networks [C]// 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010.

[5] WEI X, YU C H, ZHANG P, et al. Automated systolic array architecture synthesis for high throughput CNN inference on FPGAs [C]//Proceedings of the 54th Annual Design Automation Conference. Austin, TX, USA, 2017: 29.

[6] CHEN T, DU Z, SUN N, et al. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning [J]. ACM Sigplan Notices, 2014, 49(4): 269-284.

[7] BROMBERGER M, BASTIAN P, BERGEEST J P, et al. FPGA-accelerated Richardson-Lucy deconvolution for 3D image data [C]//2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI). Prague, Czech Republic, 2016: 132-135.

[8] ZHANG X, DAS S, NEOPANE O, et al. A Design Methodology for Efficient Implementation of Deconvolutional Neural Networks on an FPGA [J]. arXiv preprint arXiv:1705.02583, 2017.

作者简介:

涂凯杰 男, (1994-), 硕士研究生. 研究方向为深度学习加速. Email: tukaijie@ict.ac.cn.

朱宜炳 男, (1982-), 硕士研究生, 讲师. 研究方向为形式化方法与模型检测.

许达文 男, (1986-) 博士, 副教授. 研究方向为 GPU 通用计算与嵌入式系统.