Journal of Computer Applications
计算机应用, 2020, 40(10): 2811 - 2816
ISSN 1001-9081
CODEN JYIIDU
2020-10-10
http://www.joca.cn

文章编号:1001-9081(2020)10-2811-06
DOI:10.11772/j.issn.1001-9081.2020020256

基于FPGA的卷积神经网络定点加速

雷小康[1,2],尹志刚[2*],赵瑞莲[1]
(1. 北京化工大学 信息科学与技术学院,北京 100029; 2. 中国科学院 自动化研究所,北京 100190)
(*通信作者电子邮箱zhigang.yin@ia.ac.cn)

摘 要:针对卷积神经网络(CNN)在资源受限的硬件设备上运行功耗高及运行慢的问题,提出一种基于现场可编程门阵列(FPGA)的CNN定点计算加速方法。首先提出一种定点化方法,并且每层卷积设计不同的尺度参数,使用相对散度确定位宽的长度,以减小CNN参数的存储空间,而且研究不同量化区间对CNN精度的影响;其次,设计参数复用方法及流水线计算方法来加速卷积计算。为验证CNN定点化后的加速效果,采用了人脸和船舶两个数据集进行验证。结果表明,相较于传统的浮点卷积计算,所提方法在保证CNN精度损失很小的前提下,当权值参数和输入特征图参数量化到7-bit时,在人脸识别CNN模型上的压缩后的权重参数文件大小约为原来的22%,卷积计算加速比为18.69,同时使FPGA中的乘加器的利用率达94.5%。实验结果表明了该方法可以提高卷积计算速度,并且能够高效利用FPGA硬件资源。

关键词:卷积神经网络;定点量化;现场可编程门阵列;模型压缩;YOLO模型
中图分类号:TP391.4 **文献标志码**:A

FPGA-based convolutional neural network fixed-point acceleration

LEI Xiaokang[1,2], YIN Zhigang[2*], ZHAO Ruilian[1]
(1. *School of Information Science and Technology, Beijing University of Chemical Technology, Beijing* 100029, *China*;
2. *Institute of Automation, Chinese Academy of Sciences, Beijing* 100190, *China*)

Abstract: Aiming at the problem of high running power consumption and slow operation of Convolutional Neural Network (CNN) on resource-constrained hardware devices, a method for accelerating fixed-point computation of CNN based on Field Programmable Gate Array (FPGA) was proposed. First, a fixed-point processing method was proposed. In order to reduce the storage space of the CNN parameters, different scale parameters were designed for different convolution layers and the relative divergence was used to determine the bit width length. The effect of different quantization intervals on the accuracy of CNN was studied. Then, the parameter multiplexing method and the pipeline calculation method were designed to accelerate the convolution calculation. In order to verify the acceleration effect of CNN after fixed-point processing, two datasets of face and ship were used for verification. Compared with the traditional floating-point convolution computation, on the premise of ensuring that the accuracy loss of the CNN is small, when the weight parameters and the input feature map parameters are quantized to 7-bit, on the face recognition CNN model, the proposed method has the compressed weight parameter file size of about 22% of the origin, and the convolution calculation speedup is 18.69. At the same time, the method makes the utilization rate of the multiplier-accumulator in FPGA reach 94.5%. Experimental results show that the proposed method can improve the speed of convolution calculation, and efficiently use FPGA hardware resources.

Key words: Convolutional Neural Network (CNN); fixed-point quantization; Field Programmable Gate Array (FPGA); model compression; YOLO model

0 引言

近年来,人工神经网络在日常生活中的应用越来越广泛,卷积神经网络(Convolutional Neural Network, CNN)是一种源自人工神经网络的机器学习算法,它简化了传统识别算法中复杂的特征提取和数据重建的过程,在视频监控、机器视觉、图像搜索、模式识别等领域得到越来越广泛的应用。

随着深度学习的普及和Caffe、Tensorflow、Torch等深度学习框架的成熟,卷积神经网络模型的识别精度越来越高。比较有名的有:LeNet-5[1]手写数字识别卷积神经网络,精度达99%以上;AlexNet[2]模型和VGG-16[3]模型的提出突破了传统图像识别的精度;GooLeNet[4]和ResNet[5]推动了卷积神经网络的应用,但是卷积神经网络参数也随之越来越多,庞大的计算量导致CNN模型很难移植到手机端或嵌入式芯片中。因此,如何在保证卷积神经网络模型精度的前提下,对深度卷积神经网络进行压缩和加速,并在嵌入式设备上部署,已成为一个重要的研究课题。

收稿日期:2020-03-16;**修回日期**:2020-04-22;**录用日期**:2020-05-07。 **基金项目**:国家自然科学基金资助项目(61672085)。
作者简介:雷小康(1994—),男,河南周口人,硕士研究生,主要研究方向:深度学习、卷积神经网络模型压缩与加速; 尹志刚(1976—),男,湖北天门人,研究员,博士,主要研究方向:人工智能、处理器芯片架构; 赵瑞莲(1964—),女,山西忻州人,教授,博士,主要研究方向:软件测试、软件可靠性。

早期有学者提出了一些CNN模型压缩方法，如：Denil等[6]使用低秩分解的方法减少了深层网络模型的动态参数数量；Sainath等[7]研究了深度神经网络中最终加权层的低阶矩阵分解方法以进行声学建模。但是低秩分解方法计算成本高昂，并且需要大量的重新训练来达到收敛。自2016年以来Han等[8-10]提出通过设定阈值来修剪权值参数，结合*K*-Means聚类和霍夫曼编码进一步压缩网络，达到网络稀疏化的目的；Iandola等[11]提出Fire Module模型结构，用更小的卷积核代替较大的卷积核，通过减少参数数量进行模型压缩。这两种方法虽然减少了网络参数，但是卷积计算仍然采用浮点数卷积运算，计算复杂度并没有降低。

Gysel[12]提出一种对卷积神经网络定点化仿真的工具Ristretto，将浮点数仿真表示为定点数，采用一系列CNN模型定点化方法，研究不同定点化模型与精度损失之间的关系；Rajasegaran等[13]提出胶囊网络，采用3D动态路由卷积算法，虽然一定程度上减少了参数数量，但是CNN模型参数依然使用浮点数存储，卷积计算采用浮点数卷积运算，模型参数的存储大小不变，无法在嵌入式硬件上实现。Zhao等[14-16]提出基于现场可编程门阵列（Field Programmable Gate Array，FPGA）的卷积神经网络加速方法，将CNN参数定点量化，但是没有考虑成本较低的硬件资源的限制。

FPGA是一种集成电路，包含大量的定点计算单元，与图形处理器（Graphics Processing Unit，GPU）相比，FPGA具有低功耗低成本的特点，并且在大多数情况下能达到GPU相近的加速效果。

综合上述研究，本文针对传统浮点数卷积计算复杂度高、浮点模型占用存储空间大以及运行速度慢的问题，提出一种基于FPGA的优化定点卷积计算方法。本文的主要工作包括以下几个方面：

1）本文通过设计动态定点量化方法，将浮点CNN模型的权值参数和各层特征图参数动态量化为定点模型；

2）考虑数据的存储方式及卷积核的特点，将参数量化为7-bit，设计参数复用和流水线卷积计算方法，在精度损失很小的情况下，卷积计算速度加速比达到18.69。

1 卷积神经网络模型参数预处理

为提高训练结果的准确性，卷积神经网络在训练过程中一般会使用多种优化方法，但是，在模型训练完成之后的前向推理过程，预先对卷积神经网络模型参数进行处理，可以减少前向推理的计算量，有利于对模型定点量化，提高运算速度。

1.1 卷积神经网络

卷积神经网络一般包含输入层、卷积层、激活层和全连接层。本文采用的CNN模型包括输入和权值参数的卷积计算、批量归一化、激活层和池化层，最后一层使用YOLO[17]层检测目标的坐标位置。

卷积计算本质上是大量数据的乘累加操作，公式如下：

$$\boldsymbol{x}_j^l = \sum_{i=1}^{m} \boldsymbol{x}_i^{l-1} \otimes \boldsymbol{w}_{ij};\ j = 1, 2, \cdots, n \tag{1}$$

其中：m为输入特征图个数；n为卷积计算后输出特征图个数；$\boldsymbol{x}_j^l$为第l层的第j个特征图；$\boldsymbol{w}_{ij}$为第i通道的第j个权值参数矩阵。

目前大多数CNN模型在训练时都会在每个卷积层后面增加批量归一化（Batch Normalization，BN）层[18]。BN层用于将数据归一化，可以有效解决梯度爆炸问题，加速网络收敛，并且可以解决过拟合的问题，一般放在卷积层之后，计算公式如下：

$$\boldsymbol{x}_j^{l_{\text{bn}}} = \frac{\gamma\left(\boldsymbol{x}_j^l - \mu\right)}{\sqrt{\sigma^2 + \phi}} + \beta;\ j = 1, 2, \cdots, n \tag{2}$$

其中：γ为缩放因子；μ为均值；σ^2为方差；ϕ为很小的正数，本文取值为10^{-6}；β为偏置；$\boldsymbol{x}_j^l$为式(1)中卷积计算结果；$\boldsymbol{x}_j^{l_{\text{bn}}}$为经过BN计算后的结果。

激活层本文采用ReLU函数作为激活函数，计算公式如下：

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leqslant 0 \end{cases} \tag{3}$$

本文池化层使用最大值池化。

1.2 权值参数预处理

BN层在训练时起到了积极作用，但是会导致在网络前向推理时多了一层运算，占用了更多的计算资源，在一定程度上会降低运算速度。因此，本文将BN层的参数合并到卷积层的权值参数中，来提升模型前向推理的计算速度。依据式(1)~(2)可得：

$$\boldsymbol{x}_j^l = \frac{\gamma\left(\sum_{i=1}^{m} \boldsymbol{x}_i^{l-1} \otimes \boldsymbol{w}_{ij} - \mu\right)}{\sqrt{\sigma^2 + \phi}} + \beta_j \tag{4}$$

展开后得到：

$$\boldsymbol{x}_j^{l_{\text{bn}}} = \sum_{i=1}^{m}\left(\boldsymbol{x}_i^{l-1} \otimes \frac{\gamma^* \boldsymbol{w}_{ij}}{\sqrt{\sigma^2 + \phi}}\right) - \frac{\gamma^* \mu}{\sqrt{\sigma^2 + \phi}} + \beta_j \tag{5}$$

由式(5)、(1)可得：权值参数变为$\boldsymbol{w}'_{ij} = \dfrac{\gamma^* \boldsymbol{w}_{ij}}{\sqrt{\sigma^2 + \phi}}$，偏置变为$\beta'_j = -\dfrac{\gamma^* \mu}{\sqrt{\sigma^2 + \phi}} + \beta$。合并后卷积计算变为：

$$\boldsymbol{x}_j^l = \sum_{i=1}^{m}\left(\boldsymbol{x}_i^{l-1} \otimes \boldsymbol{w}'_{ij}\right) + \beta'_j;\ j = 1, 2, \cdots, n \tag{6}$$

经过变换，卷积计算和BN计算由式(1)~(2)转化为式(6)，每一次的卷积计算都减少了开方和除法操作，一定程度上可以加速卷积计算。

2 CNN模型参数定点化优化方法

参数预处理将卷积神经网络简化为只包含卷积层、池化层和激活层，主要计算量在卷积层，本文针通过对卷积层的权值和输入特征参数定点量化，将卷积层的浮点卷积计算转换为高效的定点卷积计算，提高运算速度。

2.1 权值参数定点化

在FPGA中浮点运算相较于定点运算要耗费数倍的资源和时间，统计不同网络模型权值参数的分布可以发现，不同CNN模型的权值大致对称分布在零值两侧，且不同卷积层的权值参数具有显著的动态范围，在对其定点量化[12]时，若将所有层的参数量化为同一个范围，会造成较大的精度损失。

本文设计动态指数定点量化的方式对权值参数进行量化，在对每一层参数进行量化时，将每层参数分组为具有指数为常数fl的组中，分配给小数部分的位数在该组内是恒定的，但与其他组相比是不同的。每个网络层分为三组，分别用于层输入、权重、层输出，可以更好地覆盖每层输入参数和权重

参数的动态范围。

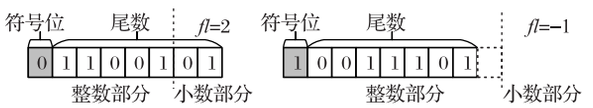


图1　权值参数动态定点量化

Fig. 1　Dynamic fixed-point quantization of weight parameters

动态定点量化计算公式如式(7)所示:

$$y=(-1)^{s}*2^{-fl}*\left(\sum_{i=0}^{B-2}2^{i}*\boldsymbol{x}_{i}\right) \tag{7}$$

其中:B是量化的位宽长度;s是符号位;fl是不同卷积层的量化指数位长度;$\sum_{i=0}^{B-2}2^{i}*\boldsymbol{x}_{i}$是定点数的尾数部分。

卷积计算时使用定点化后的CNN模型,卷积计算转化为定点数的尾数部分进行乘累加运算,将运算的结果使用每层的量化尺度再进行量化,之后参与下一层的卷积计算,依此类推,直到完成所有的卷积计算。

2.2　输入参数定点化

在对输入层数据量化时,由于每次传入的图片数据不一样,每一层计算的输出差异很大,因此每次的输入不能直接确定量化范围。卷积计算时,每层的输入都计算量化位宽会增加前向推理运算时间。为了减少计算量并且保证精度损失不大,本文采用Kullback-Leibler(KL)散度来计算输入参数定点化的尺度[19]。

首先构建一种由32-bit数据向n-bit定点数的映射关系,该映射中的边界并不是两种数据类型的最大值(图2(a)),而是设置一个阈值T(图2(b)),将这个阈值与n-bit定点数的最大值(例如8-bit定点数最大为127)构建映射关系,计算输入尺度fl_in。

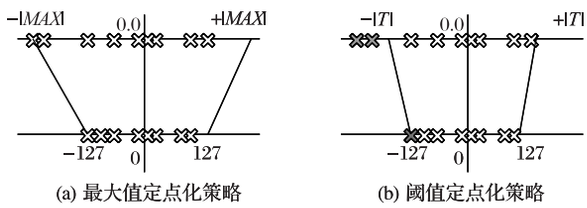
确定这种映射关系的阈值T和尺度采用KL散度。不同的网络阈值T和每层的尺度是不同的,32-bit浮点数映射到n-bit定点数相当于重新编码信息,在选择阈值T和尺度时应尽量保证减少信息的丢失,设置一个矫正数据集来进行输入尺度的选取,计算最小化KL散度来确定最佳尺度,如式(8)所示:

$$KL(P,Q)=\sum_{x\in X}\left(P[x]*\log\left(\frac{P[x]}{Q[x]}\right)\right) \tag{8}$$

其中:$P[x]$和$Q[x]$是两个离散概率分布;x为量化到不同位宽长度的参数个数。

当式(8)中KL取最小值时,得到阈值T,通过式(9)计算得出每层参数的尺度fl_in:

$$fl\_in=(-1)^{s}\frac{T}{\sum_{i=0}^{B-2}2^{i}*x_{i}} \tag{9}$$



(a) 最大值定点化策略　　(b) 阈值定点化策略

图2　输入定点化阈值选择策略

Fig. 2　Fixed-point threshold selection strategy of inputs

3　基于FPGA的定点化CNN加速设计

针对第2章提出的CNN定点化方法,考虑Arm处理器与FPGA之间的交互,设计了基于FPGA的CNN加速计算模型,如图3所示。本文提出的定点计算模型主要分为五个模块,分别为参数量化模块、参数加载模块、输入模块、卷积计算模块和输出模块。

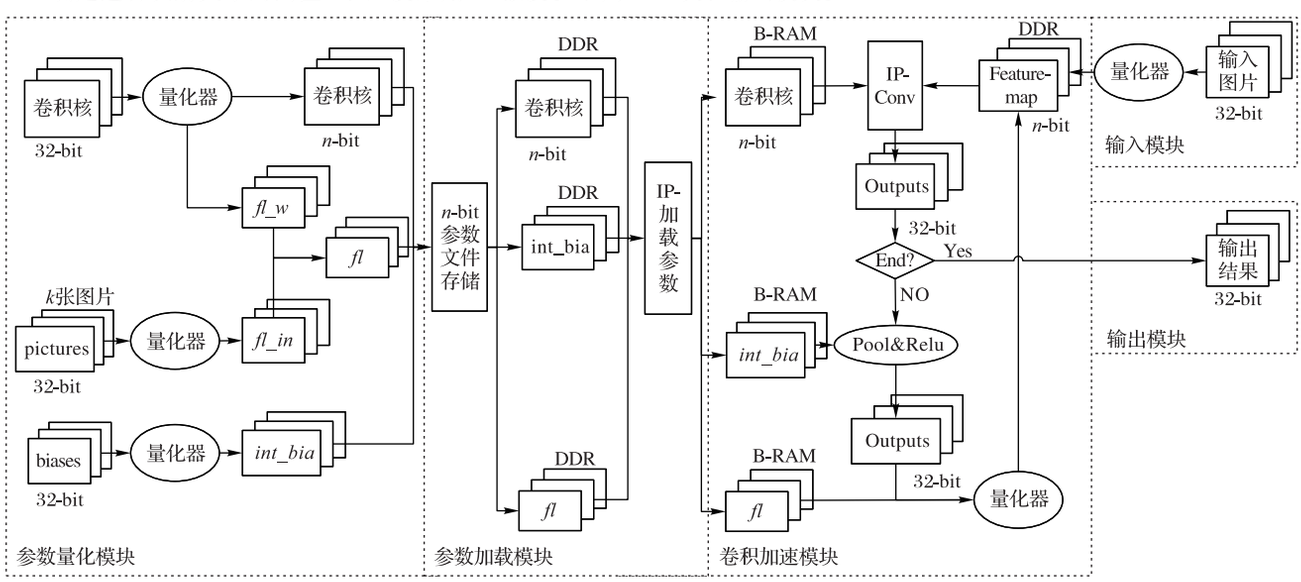


图3　定点计算模型

Fig. 3　Fixed-point computation model

3.1　参数量化

参数量化模块是预先计算定点CNN模型,将定点后的参数存储在文件中,卷积计算时,直接使用定点化后的参数文件。该模块由三部分构成:第一部分是卷积核(权值参数)经过量化器进行量化,量化方法使用2.1节中的方法,计算出n-bit权值参数以及每层参数的尺度fl_w;第二部分是使用2.2节中的方法确定输入量化尺度fl_in,经过矫正数据集(k张图片)计算出fl_in;第三部分使用2.1节中的方法对偏置进行int-32量化,得到定点化后的偏置。根据fl_w和fl_in计算每层整体尺度fl,计算方法如式(10)所示:

$$fl_{l+1} = fl\_w_l + fl\_in_l - fl\_in_{l+1} \tag{10}$$

其中：l指第l层卷积。

最后将定点化后的权值参数、尺度fl以及偏置存储在文件中，得到定点化后的CNN模型。

3.2 参数加载

在卷积运算时，需要将定点化后的CNN模型参数文件加载到FPGA内部存储器Block-RAM中。本文使用vivado HLS工具设计读取参数文件的接口，使用AXI（Advanced eXtensible Interface）高速总线接口，如图4所示。

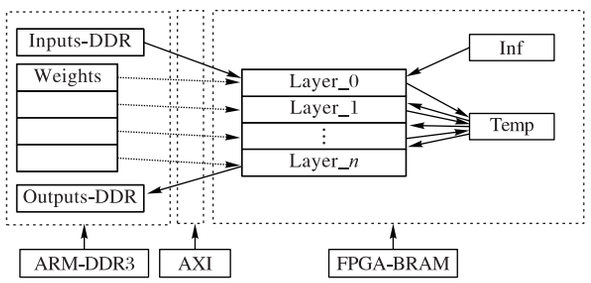


图4 数据加载接口设计

Fig. 4 Data loading interface design

首先ARM处理器将参数文件中的参数读取到双倍数据率同步动态随机存取存储器（Double Data Rate Synchronous Dynamic Random Access Memory，DDR）中，再将输入参数、权值参数和输出参数的地址通过AXI总线配置到FPGA中，FPGA将数据读取到Block RAM中，一些接口配置信息存储在Inf中，中间层计算结果存储在Temp中。

3.3 输入模块

输入模块是对输入图像像素值的预处理，RGB三通道的图像作为输入，将图像像素值利用2.2节介绍的方法进行定点化，得到定点化后的输入数据。

3.4 卷积加速

本文使用vivado HLS高层次综合工具完成FPGA硬件编程部分，根据卷积计算特性，设计了一种卷积加速计算的方法。

首先考虑数据的复用。由于FPGA内部存储资源有限，每个卷积核（权重参数）会多次与特征图进行卷积，所以在卷积计算前，将特征图矩阵进行拆分，分批加载数据，将大矩阵拆分为小的矩阵，再将小矩阵依次加载到FPGA的Block-RAM中缓存。本文使用的模型每一层的特征图的长和宽均为16的倍数，因此将每一层的特征图拆分为边长为16的小矩阵，分批对16×16的特征图与3×3的卷积核进行卷积计算，最终将计算结果合并在一起。

其次考虑卷积计算的流水线操作。加速卷积计算采用将3×3卷积核及拆分后的16×16小特征图矩阵，循环展开及流水线卷积计算，伪代码如下所示：

```
for(i=0; i<R; i++){
  for(j=0; j<C; j++){
    for(k=0; k<M; k+=Tm){
      for(l=0; l<N; l+=Tn){
        for(kk=0; kk<Tm;kk++){
          #pragma HLS pipeline
          for(ll=0; ll<Tn; ll++){
            #pragma HLS UNROLL
            for(m=0; m<K; m++){
              #pragma HLS UNROLL
              for(n=0; n<K; n++){
                outputs[kk][i][j]+=
                  weights[kk][ll][m][n]*
                  inputs[l][S*i+m][S*j+j];
}}}}}}}}
```

3.5 输出模块

该模块将3.4节中卷积计算的结果合并到一个大矩阵中，作为下一层的输入参数，循环计算得到最后一层的计算结果，最后一层计算的结果返回到ARM中参与YOLO层的分类和检测计算。

4 实验与结果分析

4.1 实验设计

考虑本文方法需要在硬件FPGA上实现加速计算，因此选择C语言编写的基于YOLO-V3深度学习框架Darknet，利用Xilinx的HLS工具将C语言综合为硬件描述语言，缩短硬件开发周期。

本文在人脸数据集和船舶数据集两个数据集上验证，两个数据集为Pascal VOC格式的标准数据集。人脸数据集包含1 665张1 280×720的高清图像，分别为5个人不同角度的图片，其中随机选取1 278张作为训练集，剩余387张作为测试集；船舶数据集包含7 618张图片，分为帆船、航母、货船、游轮、游艇和战舰六类，随机选取5 484张图片作为训练集，2 134张图片作为测试集。本文设计的CNN模型为15层的卷积网络，模型大小为1 545 KB。

实验分为两部分：

第一部分使用不同量化位宽对参数定点化，分析不同量化位宽对精度的影响，计算模型的精度、召回率和平均准确率（mean Average Precision，mAP）。mAP评价的是目标预测位置的准确率，mAP越大，预测的坐标位置越接近真实位置。

第二部分，针对定点化后的CNN模型使用FPGA加速前向推理计算，分析加速效果。该部分使用第一部分定点化效果较好的模型，使用vivado SDK工具测试CNN模型在ARM上的运行速度，使用vivado HLS综合工具，设计FPGA加速方法，测试CNN模型的加速效果。

4.2 实验结果与分析

针对实验的第一部分，考虑不同量化位宽对精度、召回率和mAP的影响，针对人脸数据集进行验证，结果如表1所示。

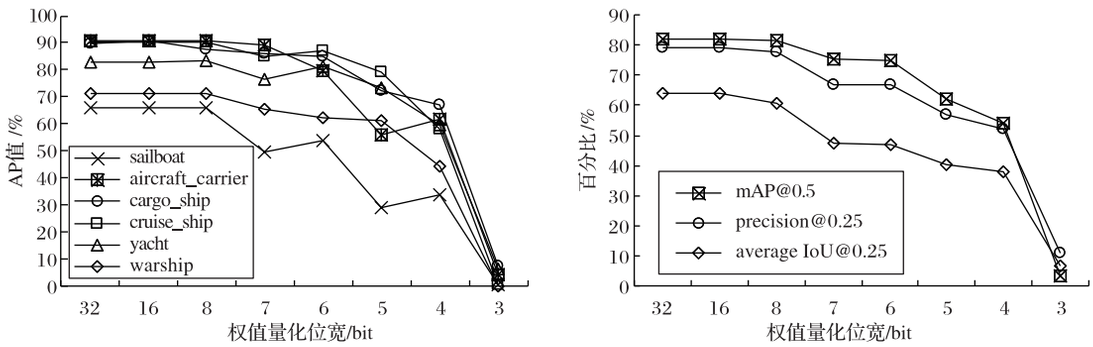
表1 不同量化位宽下的人脸数据集识别结果精度

Tab. 1 Recognition accuracy results under different quantization bit widths on face dataset

权值量化位宽/bit	输入量化位宽/bit	Face-detection		
		精度/%	召回率/%	mAP/%
32	32	97.23	97	89.95
16	16	97.21	97	89.95
8	8	95.67	95	88.08
7	7	95.52	81	87.26
7	6	94.13	80	76.53
7	5	92.71	77	75.27
7	4	90.67	74	73.06

在船舶数据集上的验证结果如图5~6所示。图5仅对权值进行定点化，图5(a)为保持每层的输入特征值原始位宽

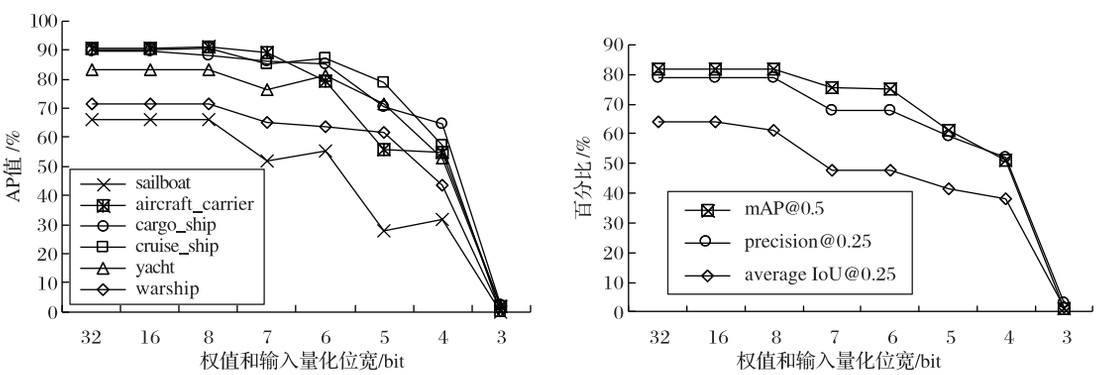
(32-bit)不变,每类平均准确率(Average Precision, AP)值随量化位宽的变化情况;图5(b)为mAP值、精度和召回率随量化位宽的变化情况。图6对输入特征值和权值都进行定点化,图6(a)为每层的输入特征值和权值同时定点化,每类AP值随量化位宽的变化情况;图6(b)为每层的输入特征值和权值同时定点化,mAP值、精度和召回率随量化位宽的变化情况。



(a) 不同类别AP值变化趋势　　(b) mAP、精度和平均IoU变化趋势

图5　权值定点化结果

Fig. 5　Results of fixed-point processing of weights



(a) 不同类别AP值变化趋势　　(b) mAP、精度和平均IoU变化趋势

图6　输入特征值和权值定点化结果

Fig. 6　Results of fixed-point processing of input feature values and weights

由图5~6可看出:本文方法在仅对权值定点化和对权值、输入特征值同时定点化两种方式下差异很小,可以满足两种不同的量化方式,具有较好的鲁棒性。综合表1还可看出:权值和输入参数量化位宽大于7-bit时,精度和mAP影响不大;量化位宽小于7-bit时,精度和mAP影响较大。

当权值参数压缩到8-bit或者7-bit时,压缩比率较大,且精度损失很小。考虑到使用的卷积核大小为3×3,一个卷积核占用63-bit,可以用一个64-bit的数据类型一次性读取一个完整的卷积核的数据进行运算,充分利用FPGA资源,高效读取数据。因此本文最终采用7-bit的定点模型作为FPGA的加速目标,针对人脸检测的CNN模型定点量化后的CNN模型大小由1 545 KB压缩为344 KB,压缩后的权重参数文件大小约为原来的22%。

使用2.3节数据复用和流水线计算方法,将代码综合成电路,硬件资源利用情况与文献[20-21]方法对比如表2所示,LUT(Look-Up-Table)利用率为54.7%,Block RAM利用率为65.3%,DSP(Digital Signal Processor)资源中的乘加器利用率高达94.5%,可以看出本文方法较为充分地利用了FPGA的资源。

表2　资源利用情况　　单位:%

Tab. 2　Resource utilization conditions　　unit:%

资源类型	文献[20]方法	文献[21]方法	本文方法
LUT	26	66	54.7
Flip-flop	11	16	13.7
DSP	68	77	94.5
Block RAM	97	60	65.3

针对人脸检测模型,与文献[20-21]方法相比,采用本文方法,在功耗增加很小的情况下,FPGA具有更高的运算峰值,加速效果更好,实验结果如表3所示。

针对不同平台人脸检测模型量化前后CNN前向推理计算耗时如表4所示。将模型量化前、量化后和FPGA加速的结果进行对比,可以看出本文提出的方法对CNN计算速度有较大提升,加速比为18.69。

表3　FPGA加速效果

Tab. 3　FPGA acceleration effect

方法	运算平台	时钟频率/MHz	数据精度	运算峰值/GMACS	功耗/W
文献[20]方法	Vertex-5	75	32-bit float	0.676	1.584
文献[21]方法	XC7A200TSBG484	—	16-bit fixed	0.598	1.225
本文方法	Zynq-7000 ZC702	100	7-bit fixed	0.712	1.831

表4 定点量化卷积加速效果

Tab. 4 Fixed-point quantization convolution acceleration effect

平台	卷积计算耗时/ms	加速比
ARM-Float	13 290	—
ARM-7bit	5 289	2.51
FPGA-7bit	711	18.69

5 结语

本文提出了一种基于FPGA的卷积神经网络定点化加速计算方法,设计了卷积计算策略,并在FPGA上验证本文提出的方法的有效性。通过与现有的两个工作进行比较,本文的方法有更高的运算峰值,同时,FPGA资源利用率较高。最后通过在人脸数据集和船舶数据集上进行精度和计算速度测试,精度损失在可接受的范围内,模型压缩为原来的22%,速度提升了17.69倍,一定程度上可以满足嵌入式设备的需求。本文提出的方法是基于特定大小的卷积核(3×3)对CNN模型进行定点化加速研究,针对卷积核大小不是3×3的CNN模型并不适用,后续的研究将考虑更通用的CNN模型,针对更加通用的CNN模型定点化进行研究。

参考文献(References)

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

[2] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks [C]// Proceedings of the 25th International Conference on Neural Information Processing Systems. Red hook, NY: Curran Associates Inc., 2012: 1097-1105.

[3] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[EB/OL]. [2020-01-20]. https://arxiv.org/pdf/1409.1556.pdf.

[4] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions [C]// Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2015: 1-9.

[5] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C]// Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2016: 770-778.

[6] DENIL M, SHAKIBI B, DINH L, et al. Predicting parameters in deep learning [C]// Proceedings of the 26th International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates Inc., 2013: 2148-2156.

[7] SAINATH T N, KINGSBURY B, SINDHWANI V, et al. Low-rank matrix factorization for deep neural network training with high-dimensional output targets [C]// Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway: IEEE, 2013: 6655-6659.

[8] HAN S, MAO H, DALLY W J. Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding[EB/OL]. [2019-05-20]. https://arxiv.org/pdf/1510.00149.pdf.

[9] HAN S, LIU X, MAO H, et al. EIE: efficient inference engine on compressed deep neural network [C]// Proceedings of the 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture. Piscataway: IEEE, 2016: 243-254.

[10] HARTIGAN J A, WONG M A. A *K*-means clustering algorithm [J]. Journal of the Royal Statistical Society, Series C (Applied Statistics), 1979, 28(1): 100-108.

[11] IANDOLA FORREST, HAN S, MOSKEWICZ M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size[EB/OL]. [2019-05-20]. https://arxiv.org/pdf/1602.07360.pdf.

[12] GYSEL P M. Ristretto: hardware-oriented approximation of convolutional neural networks[EB/OL]. [2019-05-20]. https://arxiv.org/pdf/1605.06402.pdf.

[13] RAJASEGARAN J, JAYASUNDARA V, JAYASEKARA S, et al. DeepCaps: going deeper with capsule networks [C]// Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2019: 10717-10725.

[14] ZHAO R, SONG W, ZHANG W, et al. Accelerating binarized convolutional neural networks with software-programmable FPGAs [C]// Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York: ACM, 2017: 15-24.

[15] 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. New York: ACM, 2017: No. 29.

[16] AIMAR A, MOSTAFA H, CALABRESE E, et al. NullHop: a flexible convolutional neural network accelerator based on sparse representations of feature maps[J]. IEEE Transactions on Neural Networks, 2019, 30(3): 644-656.

[17] REDMON J, FARHADI A. YOLOv3: an incremental improvement [EB/OL]. [2019-04-08]. https://arxiv.org/pdf/1804.02767.pdf.

[18] IOFFE S, SZEGEDY C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C]// Proceedings of the 32nd International Conference on Machine Learning. New York: JMLR.org, 2015: 448-456.

[19] 施一飞. 对使用TensorRT加速AI深度学习推断效率的探索[J]. 科技视界, 2017(31): 26-27. (SHI Y F. Exploring the use of TensorRT to accelerate AI deep learning inference efficiency [J]. Science and Technology Vision, 2017(31): 26-27.)

[20] 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. (YU Z J, MA D, YAN X L, et al. FPGA-based accelerator for convolutional neural network[J]. Computer Engineering, 2017, 43(1): 109-114, 119.)

[21] 魏浚峰,王东,山丹. 基于FPGA的卷积神经网络加速器设计与实现[J]. 中国集成电路, 2019, 28(7): 18-22, 67. (WEI J F, WANG D, SHAN D. Design and implementation of convolutional neural network accelerator based on FPGA[J]. China Integrated Circuit, 2019, 28(7): 18-22, 67.)

This work is partially supported by the National Natural Science Foundation of China (61672085).

LEI Xiaokang, born in 1994, M. S. candidate. His research interests include deep learning, convolutional neural network model compression and acceleration.

YIN Zhigang, born in 1976, Ph. D., research fellow. His research interests include artificial intelligence, processor chip architecture.

ZHAO Ruilian, born in 1964, Ph. D., professor. Her research interests include software testing, software reliability.