DOI:10.13878/j.cnki.jnuist.2022.03.009

吴宇航[1]　何军[2]

基于 FPGA 的人体行为识别系统的设计

摘要

为实现边缘端人体行为识别需满足低功耗、低延时的目标,本文设计了一种以卷积神经网络(CNN)为基础、基于可穿戴传感器的快速识别系统.首先通过传感器采集数据,制作人体行为识别数据集,在 PC 端预训练基于 CNN 的行为识别模型,在测试集达到 93.61%的准确率.然后,通过数据定点化、卷积核复用、并行处理数据和流水线等方法实现硬件加速.最后在 FPGA 上部署识别模型,并将采集到的传感器数据输入到系统中,实现边缘端的人体行为识别.整个系统基于 Ultra96-V2 进行软硬件联合开发,实验结果表明,输入时钟为 200 M 的情况下,系统在 FPGA 上运行准确率达到 91.80%的同时,识别速度高于 CPU,功耗仅为 CPU 的 1/10,能耗比相对于 GPU 提升了 91%,达到了低功耗、低延时的设计要求.

关键词

人体行为识别(HAR);边缘端;可穿戴传感器;卷积神经网络(CNN);现场可编程门阵列(FPGA);硬件加速

中图分类号 TP274

文献标志码 A

收稿日期 2021-04-06

资助项目 国家自然科学基金(61601230)

作者简介

吴宇航,男,硕士生,研究方向为 FPGA 硬件加速器.18851761006@163.com

何军(通信作者),男,博士,副教授,研究方向为机器学习、计算机视觉.jhe@nuist.edu.cn

1 南京信息工程大学 电子与信息工程学院,南京,210044

2 南京信息工程大学 人工智能学院,南京,210044

0 引言

人体行为识别(Human Activity Recognition,HAR)[1]是人工智能和模式识别[2]的重点研究方向之一,广泛应用于智能家居[3]、老人护理[4]和轨迹追踪[5]等领域.当前基于人体行为识别的研究方法主要分为基于计算机视觉[6]和基于可穿戴传感器[7]两大类.基于计算机视觉的研究方法是通过外部设备采集的图像、视频等信息进行检测识别,该方法存在功耗高、可持续性差、无法应用于非固定场景等不足.基于可穿戴传感器的研究方法是通过收集智能移动设备上的传感器数据实现行为识别[8],虽然相比于前一种解决方案,该方法使用便捷、抗干扰能力强,可以应对不同的使用场景,但推断计算过程大多基于云端或者 CPU,功耗高、延时大.而 FPGA 以其强大的并行计算能力、灵活的可配置性和超低功耗,是边缘端理想的计算平台.

可穿戴传感器采集的大多为序列数据,当前,主要以基于循环神经网络(RNN)[9]和卷积神经网络(CNN)的模型对序列数据进行预测和识别.虽然基于 RNN 的识别模型的精度略高于基于 CNN 的模型精度,但是 RNN 模型的循环性质和数据依赖特性,使得该模型的运算难以在硬件上实现高度并行化,从而导致在边缘端硬件平台计算效率低,而 CNN 的优势在于可以实现更高的并行度,计算性能高,适合部署在 FPGA 上,因此本文选用基于 CNN 的模型作为系统的识别模型.

目前已经有许多研究人员开展关于使用 FPGA 实现 CNN 加速的研究[10-11].文献[12-13]使用了流水线和循环展开等方法优化卷积计算过程;文献[14]提出了参数化架构设计并在卷积运算中四个维度方向实现了并行化计算;文献[15]提出了层间计算融合的模式以减少片内外的数据传输带来的时间消耗;文献[16]设计出基于 CPU-FPGA 的软硬件协同系统来加速 CNN 网络计算.上述方法通过提高计算单元的利用率、提升卷积计算并行度和优化系统与内存之间数据传输等途径,实现加速器计算性能的提升.本文使用直接内存读取(Direct Memory Access,DMA)进行高速数据传输,同时使用定点化[17-18]、多通道并行和流水线等方法提升计算效率,最后在 PYNQ 框架下进行软硬件协同处理并输出识别结果.实验结果表明,在识别准确率达到 91.80%的情况下,系统的计算性能高于 CPU,功耗仅为 CPU 的 1/10,能耗比相对于 GPU 提升了 91%.

1 HAR-CNN 模型的搭建和训练

1.1 HAR-CNN 模型的搭建与训练

人体行为识别模型如图 1 所示.HAR-CNN 识别模型由 1 个输入层、4 个卷积层、2 个池化层、2 个全连接层和 1 个 Softmax 层组成,Softmax 层用于输出识别结果.选用 RELU 作为激活函数引入非线性.表 1 给出了各级卷积层和全连接层的权重参数个数以及各层的计算次数.从表 1 中可知模型中绝大部分计算都集中于卷积层,因此对于卷积计算的加速很有必要.

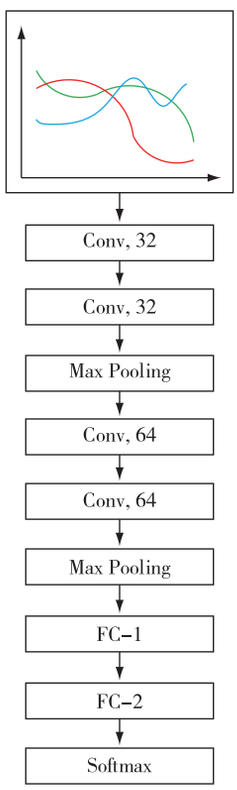


图 1 CNN 网络模型

Fig. 1 CNN model

1.2 定点化处理

通常,在 CPU 或是 GPU 上训练的模型参数类型均为浮点数,而 FPGA 仅支持定点数计算,因此本文首先对权重数据和特征数据进行定点化处理:一方面使用低精度的定点数(16 bit、8 bit)来取代浮点数,模型准确率的下降可以控制在预期范围之内;另一方面,定点数计算会节约大量的 BRAM 存储资源和 DSP 计算资源,减少数据的传输时间从而提升计算效率.

卷积计算的核心是乘累加操作,假设输入数据为一个单通道的二维矩阵,卷积核尺寸为 3×3,则第 2 层中第 i 行第 j 列的特征值可以用式(1)计算:

$$f_{i,j} = b + w_{00}p_{i-1,j-1} + w_{01}p_{i-1,j} + w_{02}p_{i-1,j+1} + w_{10}p_{i,j-1} + w_{11}p_{i,j} + w_{12}p_{i,j+1} + w_{20}p_{i+1,j-1} + w_{21}p_{i+1,j} + w_{22}p_{i+1,j+1}. \quad (1)$$

权重 W 和偏置项 b 是已知的,可以计算得出每一层特征图和权重参数的最大值(h_x)和最小值(h_n),取 $h = \max(|h_x|, |h_n|)$,取整数点位为大于以 2 为底,h 的对数的最小整数加 1,其中第 1 位为符号位.假设权重参数的最大值和最小值分别为 7.8、-8.1,则 h 为 8.1,大于以 2 为底,h 的对数的最小整数为 4,加上一位符号位,整数点位数为第 5 位.取 n 位定点数取代 32 位浮点数计算,n 减去整数点数为小数点数位,浮点数剩余(32-n)的位数由四舍五入法舍去.

由于不同卷积层的权重参数处于动态范围,故本文采用动态指数量化的方式对每层权重参数进行定点化.每个网络层的输入特征矩阵参数、输出特征矩阵参数和权重参数分别根据上文方法确定整数位和小数位.

卷积计算采用定点化后的 CNN 模型,对每层的输入特征矩阵和权重参数进行乘累加操作,计算结果根据每层的量化尺度进行量化,然后进行下一层的运算,以此延续完成整个卷积计算过程.

由于使用 DMA 搬运的数据位宽必须为 8 的整数倍,考虑到模型识别精度和定点资源消耗的情况,选用 8 bit、16 bit 和 32 bit 的数据位宽作为模型运算的数据精度,并对比不同数据位宽的模型精度和计算性能.

表 1 各级参数量

Table 1 Parameters at all levels

网络层	输出尺寸($H \times W \times C$)	权重参数数量/bias	各层计算量/flops
Input	16×8×6		
Conv1	16×8×32	1 728/32	221 184
Conv2	16×8×32	9 216/32	1 179 648
Pool1	8×4×32		
Conv3	8×4×64	18 432/64	589 824
Conv4	8×4×64	36 864/64	1 179 648
Pool2	4×2×64		
FC1	128	65 536/128	65 536
FC2	5	640/5	640
Softmax	5		
总计		132 416/325	3 236 480

2 HAR-CNN 硬件设计

2.1 系统架构

人体行为识别软硬件协同加速系统整体框架如图 2 所示,整个系统设计由采集并制作数据集、模型训练及优化、IP 核设计、FPGA 验证以及终端显示等模块组成.模型开发阶段,对识别模型进行训练和优化,并将量化后的权重参数和传感器数据存入 SD 卡.模型验证阶段,首先设计出卷积和池化 IP 核,并采取流水线和并行化处理等方法进行加速计算,最后在 PYNQ 框架中的 PS(Processing System)端使用高级语言映射 CNN 模型,对输入的传感器数据进行识别.

传感器采集到的数据首先传入 PS 端数据预处理模块,经过降噪、标准化处理后将数据传入 PL(Programmable Logic)端硬件加速模块,PS 端 CPU 负责根据模型运算的过程分别调用卷积核和池化核进行计算.经过模型计算后的识别结果通过 HDMI-out 接口输出到终端显示.

2.2 硬件设计

CNN 加速器的运行架构如图 3 所示.卷积模块和池化模块分别负责运算模型中的卷积计算和池化计算.PS 端的 CPU 控制端负责控制调用计算模块并配置计算参数.DMA 模块负责数据在内部存储器和外部双倍速率同步动态随机存储器(Double Data Rate,DDR)之间的传输.CPU 通过控制信号调用卷积模块和池化模块再进行不同的排列组合可以实现不同网络的卷积神经网络模型.本文设计的 IP 核都是复用的,通过对计算参数的配置,可以支持不同尺寸、不同模式下的卷积运算.卷积启动信号启动卷积计算之后,此时特征图数据和权重数据存入到片上存储中,经过卷积和 RELU 计算后,将计算结果通过 DMA 传回外部存储器.池化启动信号负责将经过卷积和 RELU 计算后的特征信号进行池化计算,计算结果最终存入外部空间.

加速计算架构的运算流程如下:

1)PYNQ 框架下,CPU 在外部存储器 DDR 上开辟空间,将量化后的权重参数从 SD 卡存入到 FPGA 中的 DDR 中.

2)PS 端 CPU 启动加速器之后,首先通过 S-AXI 接口将计算参数和控制信号写入随机存取存储器 RAM 中,然后 DMA 将特征数据搬运到片上 BRAM 中,将权重数据搬运到权重 buffer.

3)卷积运算模块得到指令后,同时获取权重数据和特征图数据,并且进行卷积和 RELU 计算,计算后的结果存入片上 BRAM 中,最后通过 DMA 存入 DDR 中.

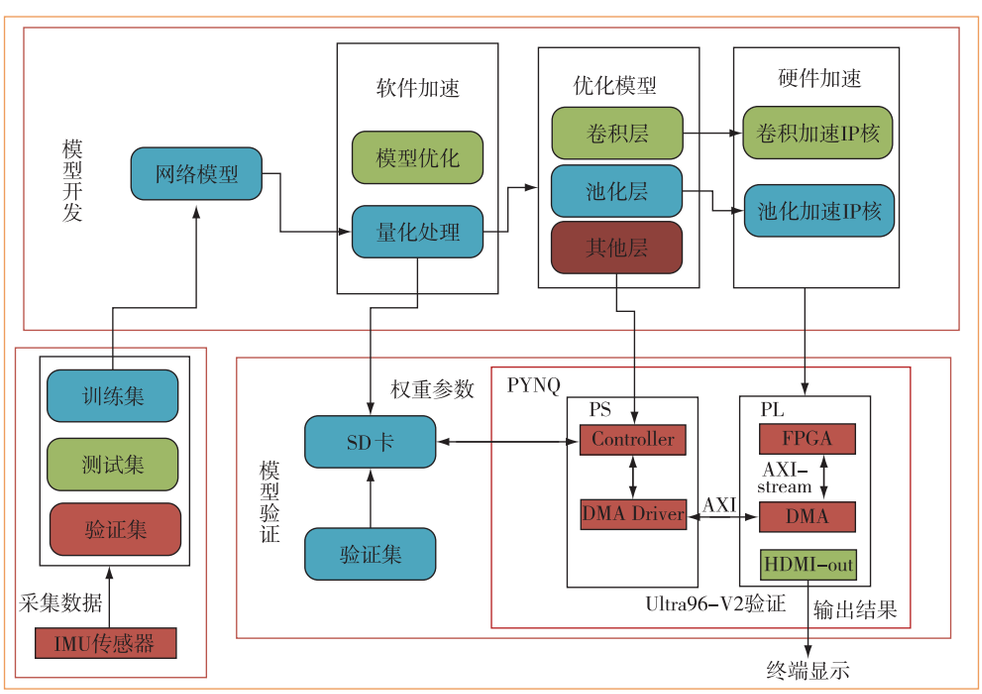


图 2 软硬件协同加速系统设计框图

Fig. 2 System architecture of hardware & software co-acceleration

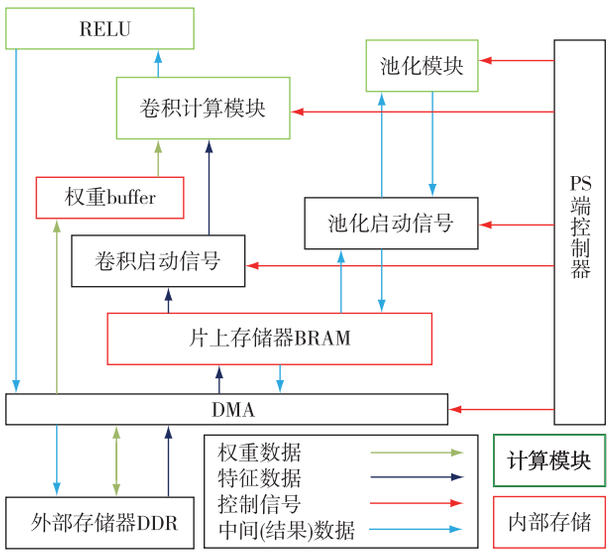


图 3　加速计算架构

Fig. 3　Accelerator's computing architecture

4)池化运算模块得到指令后,将经过卷积模块计算后的特征图数据根据 CPU 配置的池化参数进行计算,将计算中间数据存入 BRAM 中,将最终结果直接通过 DMA 传入到 DDR 中.

5)根据模型设计,重复执行上述 3)和 4),直到完成整个推断过程,将最终的识别结果通过 HDMI 输出到终端显示.

2.3　卷积加速设计

卷积运算模块如图 4 所示.受限于硬件资源,本文使用 HLS 设计了 2 个处理单元(PE),并使用 Dataflow 约束条件使 2 个 PE 模块并行计算.PE 模块主要负责卷积的乘加运算,并将计算结果储存到 BRAM 中.

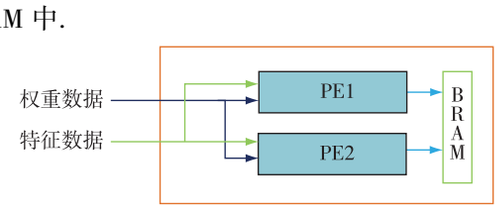


图 4　卷积运算模块

Fig. 4　Convolution operation module

PE 模块核心计算单元的设计如图 5 所示,PE 模块内部都开辟了缓存用来存储权重数据,模块内具体的计算流程如下:取 N 个对应的位宽为 16 bit 的特征数据和权重参数,将对应的 N 个数据相乘,再将相乘的 N 个结果累加.

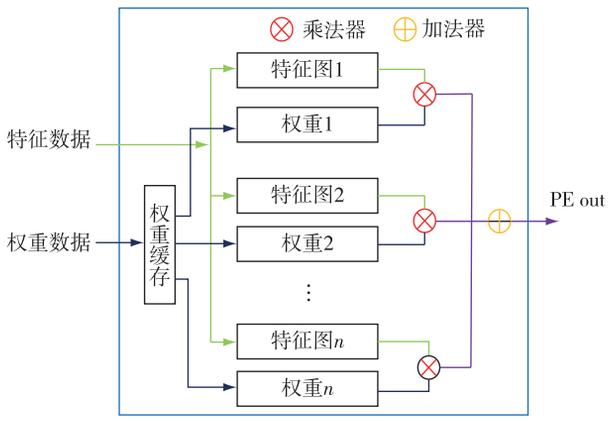


图 5　PE 模块核心计算单元

Fig. 5　PE module core computing unit

2.4　卷积并行计算

卷积计算是 CNN 模型中最大的计算部分,为了提升卷积计算的运算效率,本设计采取了多 channel 和多 filter 并行的方法进行加速计算.具体过程如图 6 所示.图 6 中输入特征图尺寸为 $H_{in}\times W_{in}$,卷积核尺寸为 $K_x\times K_y$,输入通道为 CHin,卷积核个数为 K,N 为通道并行度.

多 channel 并行是将特征图和每个权重数据沿着输入方向切成多个子块,每个子块的输入通道维度均为 N,分别从特征图子块和权重子块中取出 N 个对应的数据进行计算.多 filter 并行是将输入权重和多个卷积核同时进行计算.图 6 为子块进行卷积运算的流程,具体的计算流程如下:

1)如图 6a 所示,2 个 PE 单元同时取出特征图和 2 个卷积核的 N 个通道方向上的第 1 个数据,将分别取出的对应数据相乘,然后在通道方向上累加,输出结果.

2)如图 6b 所示,PE 模块中的权重不变,滑动特征图,使得权重 N 个通道上第 1 个数据与部分特征图数据相乘并在通道方向上累加.

3)如图 6c 所示,改变 PE 中的权重数据,使得计算可以覆盖子块的全部特征数据和权重数据,得到 N 个通道上 3×3 卷积核所有数据与整幅特征图计算中,完成与 3×3 卷积核 9 个权重数据相对应的特征图相乘并在通道方向累加的结果.

4)沿通道方向取出下一个 N 通道子块重复上述计算,将最终得到的结果累加输出特征图.

为了验证卷积加速设计的有效性,选取 N 分别为 16、32、64 时进行多通道加速设计,并与未加速的设计进行比较.本文选取数据位宽为 16 bit,尺寸

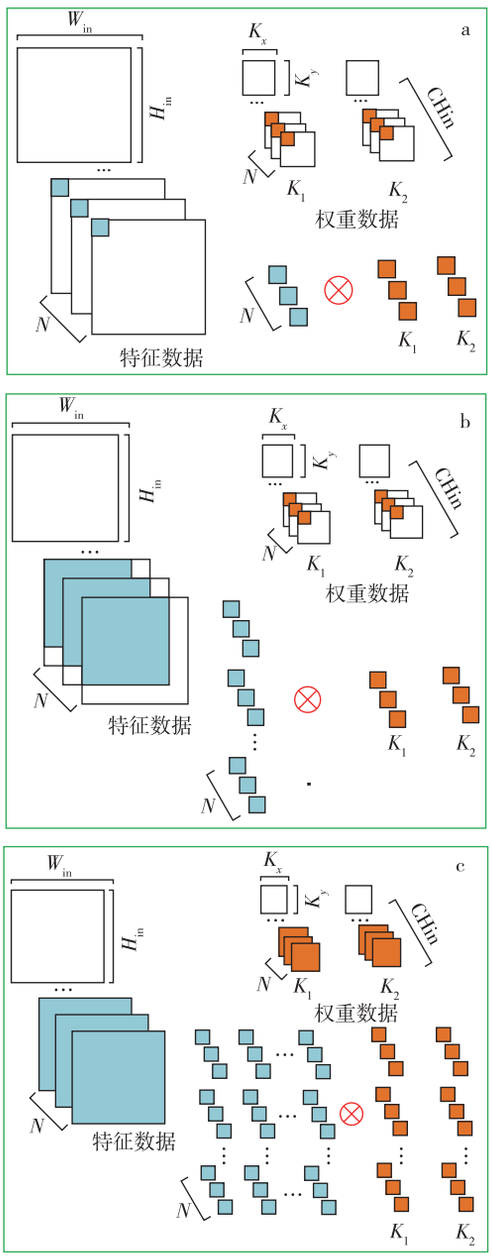


图 6 卷积计算流程

Fig. 6 Convolution operation process

为 32×32×64 输入特征数据和尺寸为 64×3×3×64 的权重数据进行卷积计算,加速对比如表 2 所示.表 2 中的实验都进行了定点化和流水线优化,但对比实验未进行多 channel 并行和多 filter 并行设计加速.实验结果表明,最优设计(本设计)的计算性能是未进行多 channel 并行和多 filter 并行加速设计的 95.46 倍.

为了验证量化设计对计算性能加速的有效性,选取数据位宽分别为 8 bit、16 bit 和 32 bit,在 PE=2,N=32 的条件下进行卷积加速设计.选取和表 2 尺寸相同的特征数据和权重数据进行卷积计算,加速对比如表 3 所示.实验结果表明,数据位宽为 8 bit、16 bit 量化设计的计算性能分别是数据位宽为 32 bit 的 1.38 倍和 1.21 倍.

表 2 不同卷积加速方案的性能对比

Table 2 Performance of different convolution acceleration schemes

加速方案	运行时间/ms	加速倍数
FPGA (PE=1,N=0)	336.02	
FPGA (PE=2,N=16)	11.87	28.31
FPGA (PE=2,N=32)	6.41	52.44
FPGA (PE=2,N=64)	3.52	95.46

表 3 不同数据位宽的卷积性能对比

Table 3 Performance of different data bit widths

数据位宽	运行时间/ms	加速倍数
32-bit fixed	7.76	
16-bit fixed	6.41	1.21
8-bit fixed	5.62	1.38

2.5 池化加速设计

最大值池化计算如图 7 所示.将特征图沿着长度和宽度分成多个 $n \times n$ 的子块,在每个子块内取最大值,输出的特征图的长度和宽度分别为原特征图长度和宽度的 $1/n$.

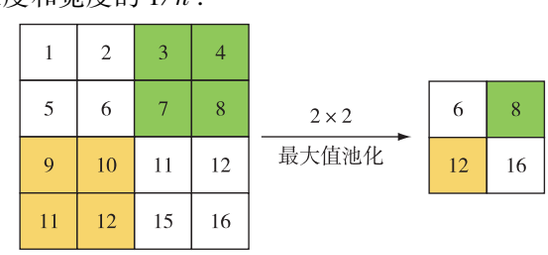


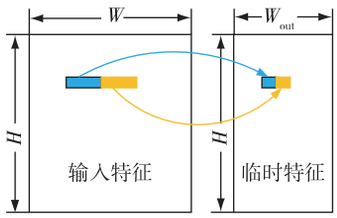
图 7 最大值池化计算

Fig. 7 Maxpooling calculation

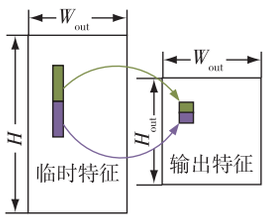
如图 8 所示,为了提升池化模块的计算效率,本文设计了横向池化和纵向池化两个子函数,并利用流水线并行计算得到输出结果,同时对每个子函数进行多通道并行加速.

假设对输入特征数据进行 $n \times n$ 的最大值池化,步长也为 n ,其具体计算流程如下:

1)如图 8a 所示,输入特征进入横向池化模块,第 1 个数据寄存在寄存器中,与后续输入的 $n-1$ 个数据相互比较,将最大值存在寄存器中.以输入的 n 个数据为一组,反复执行上述的操作,得到横向池化的结果,此时的中间数据高不变,宽度为原先宽度的



a. 横向池化



b. 纵向池化

图 8　池化流程

Fig. 8　Pooling flowchart

$1/n$.

2)如图 8b 所示,首先将中间特征输入的前 W_{out} 个数据存入 buffer 中,后面输入的 $n-1$ 组 W_{out} 个数据与第 1 组数据在对应的位置上相互比较得到最大值,输出结果.以输入的 n 组 W_{out} 个数据为一组,反复执行上述的操作,得到纵向池化的结果,此时输出特征的高度、宽度均分别为原先高度和宽度的 $1/n$.

为了验证池化加速设计的有效性,选取尺寸为 100×100×64 输入特征数据进行最大值池化运算,在 $N=16$ 的条件下进行多通道加速设计,同时应用定点化和流水线优化加速.不同的是方案 1 使用的传统的设计方案,方案 2 使用的是横向池化和纵向池化并行加速的设计方案.加速对比结果如表 4 所示,本文设计为传统的方案计算性能的 1.91 倍.

为了验证量化设计对池化计算性能加速的有效性,选取数据位宽为 8 bit、16 bit 和 32 bit 使用方案 2 进行卷积加速设计.选取和表 4 尺寸相同的特征数据和权重数据进行卷积计算,加速对比如表 5 所示.

表 4　不同池化加速方案的性能对比

Table 4　Performance of different pooling schemes

方案	运行时间/ms	加速倍数
1	0.644	
2	0.337	1.91

表 5　不同数据位宽的池化性能对比

Table 5　Performance of different data bit widths

数据位宽	运行时间/ms	加速倍数
32-bit fixed	0.377	
16-bit fixed	0.337	1.12
8-bit fixed	0.310	1.22

实验结果表明,数据位宽为 8 bit、16 bit 量化设计的计算性能分别是数据位宽为 32 bit 的 1.22 倍和 1.12 倍.

2.6　系统实现

CPU 通过不断调用卷积和池化两个子函数对输入特征进行卷积计算和池化计算,最终在 FPGA 上搭建行为识别的 CNN 网络结构.利用时分复用技术,节省硬件资源开销,时分复用结构如图 9 所示.当人体行为数据传输到神经网络之后,卷积层 L1、L2、L3、L4、L5 和 L6 由卷积核计算,池化层分别为 P1 和 P2,由池化核进行最大值池化计算. 另外,全连接层也使用卷积核进行计算.

3　实验以及结果分析

3.1　数据集

本文使用 UCI_HAR 数据集和自制 HAR 数据集进行实验.UCI_HAR 数据集可在 https://ucihar-data-analysis.readthedocs.io/en/latest/处进行下载,自建 HAR 数据集可在 https://github.com/nuisyaya/HAR_Dataset 处下载,也可向本文通信作者(jhe@nuist.edu.cn)申请使用.

1)UCI_HAR 数据集:为了验证模型对传感器数

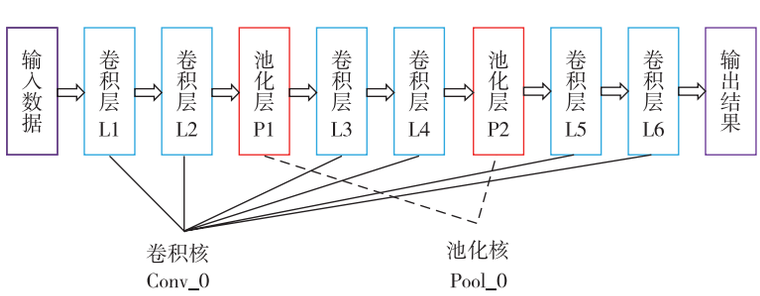


图 9　时分复用技术

Fig. 9　Time division multiplexing technology

据进行识别的有效性,本文使用已公开的 UCI_HAR 标准数据集进行验证.该数据集包括 12 个日常活动,即 3 个静态活动(站立、坐、躺)、3 个动态活动(步行、上楼、下楼)和 3 个静态活动的转换(站坐、坐站、站躺、躺站、坐躺、躺坐).这些数据由三星智能手机记录,该智能手机使用嵌入式加速度计和陀螺仪,以 50 Hz 的恒定速度收集三轴线性加速度和三轴角速度.采集者利用录制视频手动标记数据,然后随机分为两组,选用 70%作为训练集,30%作为测试集.应用噪声滤波器对传感器数据进行预处理之后,在 2.56 s 和 50%重叠的固定宽度滑动窗口中采样(窗口宽度为 128).如表 6 所示,利用这个数据集的 6 个活动,包括 3 个静态活动和 3 个动态活动进行实验,每条传感器数据的尺寸为 128×6.

表 6　UCI_HAR 数据集

Table 6　UCI_HAR dataset

活动类别	标签	数量
行走	0	1 722
上楼梯	1	1 544
下楼梯	2	1 407
坐着	3	1 801
站立	4	1 979
躺下	5	1 958
总计		10 411

2)自制 HAR 数据集:选取 SparkFun 公司的惯性测量单元(Inertial Measurement Unit,IMU)传感器采集人体行为数据制作数据集.传感器实物图如图 10 所示.采集的数据包括陀螺仪测量的 X、Y、Z 三轴的角速度,加速度计测量的 X、Y、Z 三轴的加速度,共 6 个输入通道,设置采样率为 50 Hz,要求 10 个志愿者分别采集两组行为,共计 20 组,每组行为包括行走、上楼梯、下楼梯、跑步和跳跃,分别对应标签 0 到 4.然后在 2.56 s 和 50%重叠的固定宽度滑动窗口中采样,每条样本包括 128 个采样点,每条数据尺寸为 128×6.表 7 展示 5 种行为样本个数统计结果.自制的 HAR 数据集按照 7 比 3 的比例划分为训练集和测试集.在测试集选取 500 个样本存入 SD 卡中用于 FPGA 端验证.

由于传感器在数据采集过程中会由于志愿者的主观能动性产生额外的动作,使原始数据中夹杂背景噪声,因此本文在数据预处理阶段使用中值滤波对数据进行降噪处理.

模型开发阶段,使用 TensorFlow 深度学习框架

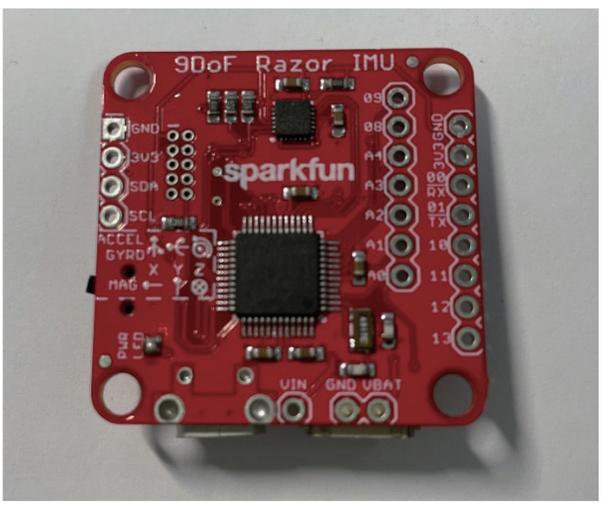


图 10　IMU 传感器实物

Fig. 10　Picture of IMU sensor

表 7　各行为样本数据统计

Table 7　Sample data of each behavior

行为类别	标签	样本数量
行走	0	3 934
上楼梯	1	3 172
下楼梯	2	3 358
跑步	3	3 752
跳跃	4	2 504
总计		16 700

进行网络训练,设置学习率为 0.000 1,使用 Adam 优化器进行优化,使用如图 1 所示的 CNN 模型架构对传统的 UCI_HAR 数据集和自制的数据集进行训练优化.如图 11 所示,本文设计的模型在传统 UCI 数据集上准确率为 91.23%,验证了模型的可靠性.在自制的 HAR 数据集上准确率达到 93.61%.

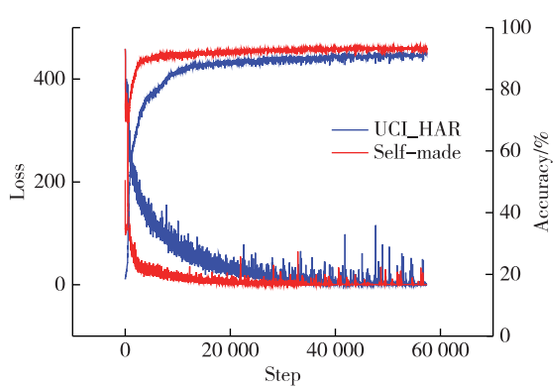


图 11　模型性能曲线

Fig. 11　Model performance curve

3.2 实验环境

本文使用 Xilinx 公司的 Vivado HLS 进行开发.FPGA 选用的是 Avnet 公司的 Ultra96-V2,工作频率为 200 MHz.实验数据类型为 16 位定点数.CPU 采用主频为 3 GHz 的 i7-9700 处理器,GPU 采用英伟达公司的 GTX1080Ti 显卡.

3.3 实验结果

随机选取每种行为各 100 条数据存入 SD 卡,用于验证部署在 FPGA 端系统的准确率.如表 8 所示,选取量化后数据位宽为 8 bit、16 bit 和 32 bit 的模型在 FPGA 端进行模型精度的验证,经过 500 次迭代,数据位宽为 8 bit、16 bit 和 32 bit 的模型验证准确率分别为 84.40%、91.80% 和 93.20%.数据量化导致的精度损失,模型精度均低于软件平台准确率 93.61%.考虑到资源消耗和模型精度的情况,最终选用 16 bit 的数据位宽作为模型运算的数据精度,同时也将其他两种精度的模型部署在 FPGA 上用于测试计算性能和功耗.

表 8 不同数据位宽模型准确率对比

Table 8 Recognition accuracy of different bit widths

数据位宽	迭代次数	准确率/%
32-bit fixed		93.20
16-bit fixed	500	91.80
8-bit fixed		84.40

本文选取数据位宽为 16 bit,N 分别为 16、32、64 三个通道并行参数进行架构设计.FPGA 硬件资源消耗如表 9 所示.

表 9 FPGA 硬件资源消耗

Table 9 FPGA hardware resource consumption

硬件资源	$N=16$	$N=32$	$N=64$
BRAM_18K	346	346	346
DSP48E	138	168	214
FF	53 486	81 572	10 450
LUT	37 514	49 574	70 548

FPGA 的资源占用率如图 12 所示.在数据位宽为 16 bit,N 分别为 16、32、64 的条件下,BRAM 的资源占用率均为 80%,DSP 资源占用率分别为 39%、47% 和 60%,触发器(Flip-Flop,FF)的占用率分别为 40%、58% 和 78%,查找表(Look-Up-Table,LUT)占用率分别为 53%、70% 和 99%.

表 10 中列举了不同计算平台的计算性能和功耗的对比.不同设计方案之间功耗对比可由各方案

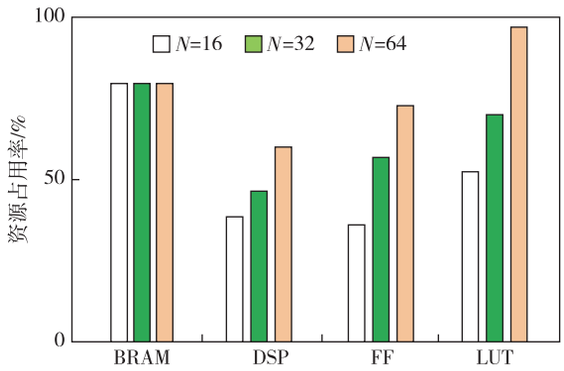


图 12 FPGA 资源占用率

Fig. 12 FPGA resource utilization

的迭代一次平均耗时与功耗的乘积的倒数相除获得.从表 10 中可以看出,本文设计的数据位宽为 16 bit、通道并行参数 $N=64$ 的人体识别系统识别速度高于 CPU,而功耗仅为 CPU 的 1/10,能耗比是 GPU 的 1.91 倍.除此之外,本文还设计数据位宽为 8 bit 和 32 bit 的模型并部署在 FPGA 上,数据位宽为 8 bit、通道并行参数 $N=64$ 的系统识别速度是 CPU 的 1.34 倍,能耗比是 GPU 的 3.02 倍.实验结果表明,使用 FPGA 作为边缘计算平台搭建人体行为识别系统,达到了低功耗、低延时的设计要求.

表 10 性能及功耗对比表

Table 10 Performance & power consumption comparison

硬件平台	数据位宽	迭代一次平均耗时/ms	功耗/W
CPU	float	11.87	38
GPU	float	0.29	250
FPGA($N=16$)	32-bit fixed	47.21	2.63
FPGA($N=16$)	16-bit fixed	27.58	2.51
FPGA($N=16$)	8-bit fixed	24.20	2.21
FPGA($N=32$)	16-bit fixed	17.42	2.95
FPGA($N=32$)	8-bit fixed	15.15	2.42
FPGA($N=64$)	16-bit fixed	10.43	3.64
FPGA($N=64$)	8-bit fixed	8.86	2.71

4 结束语

本文设计了基于 FPGA 和 CNN 的人体行为快速识别系统.通过数据定点化,并行处理数据和流水线等方法提升计算速度.使用 UCI_HAR 数据集和自制的 HAR 数据集进行了实验,并与 CPU 和 GPU 在计算性能和功耗方面进行对比,实验结果表明,本设计在识别准确率达到 91.80% 的情况下,计算速度优

于CPU,能耗比相比于GPU提升91%,达到了低功耗、低延时的设计要求,验证了FPGA作为边缘计算平台进行人体行为系统识别的可行性和优越性.在未来工作中,我们将进一步优化该系统,例如增加人体行为种类、识别不同行为之间的切换和改进神经网络模型等.

参考文献

References

[1] 朱煜,赵江坤,王逸宁,等.基于深度学习的人体行为识别算法综述[J].自动化学报,2016,42(6):848-857
ZHU Yu,ZHAO Jiangkun,WANG Yining,et al.A review of human action recognition based on deep learning[J].Acta Automatica Sinica,2016,42(6):848-857

[2] 庞拂飞,刘奂奂,王廷云.相位敏感光时域反射光纤传感技术的研究综述[J].南京信息工程大学学报(自然科学版),2017,9(2):130-136
PANG Fufei,LIU Huanhuan,WANG Tingyun.A review of distributed fiber sensors based on phase-sensitive optical time domain reflectometer[J].Journal of Nanjing University of Information Science & Technology(Natural Science Edition),2017,9(2):130-136

[3] Rashidi P,Cook D J.Keeping the resident in the loop:adapting the smart home to the user[J].IEEE Transactions on Systems,Man,and Cybernetics:Systems and Humans,2009,39(5):949-959

[4] Magherini T,Fantechi A,Nugent C D,et al.Using temporal logic and model checking in automated recognition of human activities for ambient-assisted living[J].IEEE Transactions on Human-Machine Systems,2013,43(6):509-521

[5] Yang J B,Nguyen M N,San P P,et al.Deep convolutional neural networks on multichannel time series for human activity recognition[C]//Proceedings of the 24th International Conference on Artificial Intelligence,2015:3995-4001

[6] Ordóñez F J,Roggen D.Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition[J].Sensors (Basel,Switzerland),2016,16(1):115

[7] Wang K,He J,Zhang L.Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors[J].IEEE Sensors Journal,2019,19(17):7598-7604

[8] Sainath T N,Kingsbury B,Saon G,et al.Deep convolutional neural networks for large-scale speech tasks[J].Neural Networks,2015,64:39-48

[9] 王朋,孙永辉,翟苏巍,等.基于小波长短期记忆网络的风电功率超短期概率预测[J].南京信息工程大学学报(自然科学版),2019,11(4):460-466
WANG Peng,SUN Yonghui,ZHAI Suwei,et al.Ultra-short-term probability prediction of wind power based on wavelet decomposition and long short-term memory network[J].Journal of Nanjing University of Information Science & Technology(Natural Science Edition),2019,11(4):460-466

[10] 李炳辰,黄鲁.一种移动卷积神经网络的FPGA实现[J].微电子学与计算机,2019,36(9):7-11
LI Bingchen,HUANG Lu.Hardware implementation of a convolutional neural network for mobile terminal based on FPGA[J].Microelectronics & Computer,2019,36(9):7-11

[11] Lu L Q,Liang Y,Xiao Q C,et al.Evaluating fast algorithms for convolutional neural networks on FPGAs[C]//2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines(FCCM),2017:101-108

[12] 陈辰,柴志雷,夏珺.基于Zynq7000 FPGA异构平台的YOLOv2加速器设计与实现[J].计算机科学与探索,2019,13(10):1677-1693
CHEN Chen,CHAI Zhilei,XIA Jun.Design and implementation of YOLOv2 accelerator based on Zynq7000 FPGA heterogeneous platform[J].Journal of Frontiers of Computer Science & Technology,2019,13(10):1677-1693

[13] Feng G,Hu Z Y,Chen S,et al.Energy-efficient and high-throughput FPGA-based accelerator for convolutional neural networks[C]//2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology(ICSICT),2016:624-626

[14] 李炳剑,秦国轩,朱少杰,等.面向卷积神经网络的FPGA加速器架构设计[J].计算机科学与探索,2020,14(3):437-448
LI Bingjian,QIN Guoxuan,ZHU Shaojie,et al.Design of FPGA accelerator architecture for convolutional neural network[J].Journal of Frontiers of Computer Science & Technology,2020,14(3):437-448

[15] Alwani M,Chen H,Ferdman M,et al.Fused-layer CNN accelerators[C]//2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO),2016:1-12

[16] Meloni P,Capotondi A,Deriu G,et al.NEURAghe:exploiting CPU-FPGA synergies for efficient and flexible CNN inference acceleration on Zynq SoCs[J].ACM Transactions on Reconfigurable Technology and Systems,2018,11(3):1-24

[17] 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 and Learning Systems,2019,30(3):644-656

[18] 雷小康,尹志刚,赵瑞莲.基于FPGA的卷积神经网络定点加速[J].计算机应用,2020,40(10):2811-2816
LEI Xiaokang,YIN Zhigang,ZHAO Ruilian.FPGA-based convolutional neural network fixed-point acceleration[J].Journal of Computer Applications,2020,40(10):2811-2816

Design of human activity recognition system based on FPGA

WU Yuhang[1] HE Jun[2]

1 School of Electronics & Information Engineering,Nanjing University of Information Science & Technology,Nanjing 210044

2 School of Artificial Intelligence,Nanjing University of Information Science & Technology,Nanjing 210044

Abstract In order to achieve the goal of low power consumption and low latency for edge-end human activity recognition,this paper designs a fast recognition system based on wearable sensors and Convolutional Neural Networks (CNNs).First,the system collects data through sensors to make a human activity recognition dataset,and pre-trains a CNN-based behavior recognition model on the PC side,which achieves an accuracy of 93.61% on the test set. Then,hardware acceleration is realized through methods such as data fixed point,convolution kernel multiplexing, parallel processing of data,and pipeline.Finally,the recognition model is deployed on the FPGA,and the collected sensor data are input into the system to realize the recognition of human activity at the edge.The whole system is developed jointly with hardware and software based on Ultra96-V2.The experimental results show that when the input clock is 200 M,the system runs on FPGA with an accuracy of 91.80%;the proposed system is superior to CPU in recognition speed as well as power consumption,specifically,the power consumption is only one-tenth of CPU consumed,and energy consumption ratio is 91% higher than that of GPU.It can be concluded that the FPGA-based human activity recognition system meets the design requirements of low power consumption and low delay.

Key words human activity recognition (HAR); edge-end; wearable sensor; convolutional neural networks (CNNs);field programmable gate array (FPGA);hardware acceleration