DOI:10.19651/j.cnki.emt.2209046

# 基于 FPGA 加速的行为识别算法研究*

吴宇航[1] 何 军[2]
(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)

**摘 要:** 为提高行为识别算法的实时性,适用于资源有限的嵌入式设备,提出了一种行为识别算法硬件加速方法,并在 FPGA 平台实现。传统的基于可穿戴传感器的行为识别算法需要严格标记的数据进行训练分类,但传感器序列的标注过程消耗大量的人力和计算资源,针对该问题,在传统的卷积神经网络模型中引入注意力机制,用于基于弱标签数据的行为识别。算法中的卷积、池化和注意力机制等计算模块使用高层次综合设计。针对模型的运算特性,通过流水线约束、多像素多通道并行计算和数据定点化等方法,提升运算速度。在 Ultra96_V2 平台上使用弱标签数据集进行实验,实验结果表明,所设计的行为识别系统识别准确率达到了 90%的同时,计算速度达到 25.89 frams/s,相较于 ARM_A53 处理器实现了 54.15 倍的加速效果。系统的平均功耗为 2.204 W,功耗效率为 11.75 frames/J,满足了低功耗、低延时设计要求。

**关键词:** 人体行为识别;卷积神经网络;FPGA;硬件加速;可穿戴传感器
**中图分类号:** TP302 **文献标识码:** A **国家标准学科分类代码:** 510.4030

# Research on activity recognition algorithm based on FPGA acceleration

Wu Yuhang[1] He Jun[2]
(1. School of Electronics and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China;
2. School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China)

**Abstract:** In order to improve the real-time performance of the activity recognition algorithm and be suitable for embedded devices with limited resources, a hardware acceleration method of the activity recognition algorithm was proposed and implemented on the FPGA platform. Traditional wearable sensor-based behavior recognition algorithms require strictly labeled data for training and classification, but the labeling process of sensor sequences consumes a lot of manpower and computing resources. To solve this problem, an attention mechanism is introduced into the traditional convolutional neural network model., for action recognition based on weakly labeled data. Computational modules such as convolution, pooling, and attention mechanisms in the algorithm use a high-level comprehensive design. According to the operation characteristics of the model, the operation speed is improved by pipeline constraints, multi-pixel and multi-channel parallelization, and data fixed-pointization. Experiments are carried out on the Ultra96_V2 platform, and the experimental results show that the designed behavior recognition system has a recognition accuracy of 90% and a computing speed of 25.89 frams/s, which is 54.15 times faster than that of a single-core ARM_A53 processor. The average power consumption of the system is 2.204 W and the power efficiency is 11.75 frams/s, which meets the design requirements of low power consumption and low delay.

**Keywords:** human activity recognition;convolutional neural network;FPGA;hardware acceleration;wearable sensor

## 0 引 言

近年来,随着移动智能设备的发展,基于可穿戴传感器的人体行为识别[1](human activity recognition, HAR)已经成为人工智能的重要研究方向之一,在健康医疗[2]和模式识别[3]等领域有着广泛的应用。移动设备中的如加速度计、陀螺仪等嵌入式传感器可以产生一系列时间序列数据,用于识别人体行为活动。但对于传感器数据的推断平台大多基于 CPU 或 GPU,计算过程功耗大、延迟高,难以满足实时性的要求。针对上述问题,本文选用 FPGA 作为边缘

收稿日期:2022-02-20
* 基金项目:国家自然科学基金(61601230)项目资助

端推断平台，利用其低功耗和并行计算的特性[4]，对人体行为传感器数据进行快速推断。

卷积神经网络（convolutional neural network，CNN）具有出色的数据处理能力，是一种重要的深度学习模型，广泛应用于图像识别[5]、目标追踪[6]、自然语言处理[7]等领域。虽然 CNN 在 HAR 领域有出色的性能，但仍有一些挑战需要解决，其中主要挑战之一是对传感器数据序列进行严格标记费时费力。与容易标记的图像或视频不同，很难从长序列的传感器数据中准确的分割出一种特定类型的活动，但识别被标记活动是否发生在一个长时间记录的传感器数据中是相对容易的，这种数据被称为“弱标签数据”[8]。为了提升基于弱标签数据的识别准确率，将注意力机制[9]引入到本文 CNN 模型中，可以通过弱标签数据识别行为活动并确定标记活动的位置。

FPGA 是一种可定制、可重构的专用集成电路，具有灵活度高、低功耗、并行计算的特性，适合作为专用硬件用于神经网络加速运算。陈浩敏等[10]提出了一种针对 YOLOv3-tiny 网络的通用加速器，通过轻量化网络模型，满足了面向嵌入式领域的应用需求。Zhang 等[11]使用高层次综合（high level synthesis，HLS）的设计方法，对 AlexNet 网络中的每层卷积层做了最优加速设计，获得了最优的运算性能。当前基于 FPGA 的加速器仅针对卷积神经网络基础算法，但随着 CNN 模型的不断进化，模型已融合了其他优秀的算法，比如注意力机制。因此，使用有限的硬件资源设计高效的融合注意力机制的卷积神经网络硬件架构有不错的应用前景，值得进一步研究探索。

本文设计了基于 CNN 和注意力机制的行为识别模型并部署在 FPGA 上。为了提升计算效率，采用循环分块、数组分割和流水线约束等方式实现卷积层多像素、多通道并行加速。选用 16 位的定点数代替 32 位浮点数缩小模型参数[12]，减小 FPGA 计算资源和存储资源的消耗。对硬件加速后的系统以经典 UCI_HAR 数据集和弱标签传感器数据集进行试验，分别从识别精度、计算速度和功耗效率 3 个指标对设计进行综合性能分析，实验结果表明，系统的识别精度均达到 88%以上，计算速度达到 25.89 frams/s，功耗仅 2.204 W，达到了预期设计目标。

## 1　行为识别算法模型设计

### 1.1　注意力机制

如图 1 所示，通过模型中间层提取的局部特征向量和全局特征向量之间进行计算实现注意力机制。从卷积层 $s \in \{1,2,\cdots,S\}$ 提取一组特征向量 $\boldsymbol{L}^s=\{l_1^s,l_2^s,\cdots,l_n^s\}$，其中 $\boldsymbol{L}_i^s$ 表示局部特征向量 $\boldsymbol{L}^S$ 中 $n$ 个空间位置中的第 $i$ 个特征向量。如式(1)所示，输入数据经过卷积和池化计算生成全局特征向量 $\boldsymbol{G}$，通过兼容函数 $C$ 和局部特征向量 $\boldsymbol{L}^S$ 进行点积计算。

$$c_i^s=(l_i^s,\boldsymbol{G}) \tag{1}$$

其中，$c_i^s$ 表示兼容性得分。经过式(1)的计算之后，获得了序列各个时间位置的兼容性得分 $\boldsymbol{C}(\boldsymbol{L}^s,\boldsymbol{G})=\{c_1^s,c_1^s,\cdots,c_n^s\}$，通过 Softmax 函数进行标准化计算，得到向量 $\boldsymbol{A}^s=\{a_1^s,a_2^s,\cdots,a_n^s\}$，计算过程如式(2)所示。

$$a_i^s=\frac{\exp(c_i^s)}{\sum_j^n \exp(c_j^s)} \tag{2}$$

如式(3)所示，归一化后的兼容性分数 $A^s$ 和局部特征向量 $\boldsymbol{L}^S$ 的对应元素加权平均获得每个中间层 $s$ 的单个向量 $\boldsymbol{g}^s$。

$$\boldsymbol{g}^s=\sum_{i-1}^n a_i^s \cdot l_i^s \tag{3}$$

多个向量 $\boldsymbol{g}^s$ 拼接成新的向量 $\boldsymbol{g}=[g^1,g^2,\cdots,g^s]$ 取代原先的全局向量 $\boldsymbol{G}$ 作为输入进行线性分类的计算。兼容性得分 $A^s$ 对应位置的值表示被标记活动发生在该区域的概率。

### 1.2　模型设计

带注意力机制的识别网络结构如图 1 所示。该识别模型由 4 个卷积层、3 个池化层、3 个注意力子模块、2 个全连接层组成。一维度卷积核的尺寸为 3×1，步长为 1，池化单

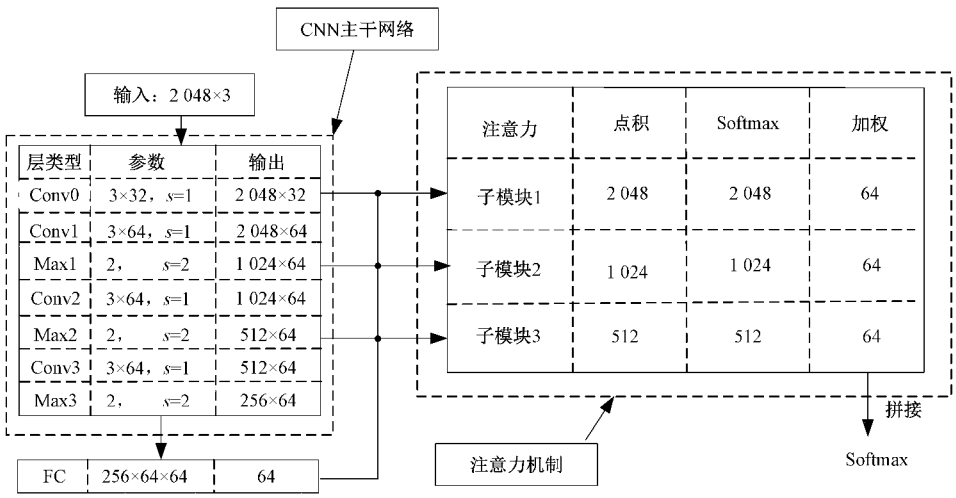



图 1　带注意力机制的 CNN 网络结构

元尺寸为 2×1,步长为 2。通过卷积层提取局部特征,池化层对特性向量降维,并输出到全连接层,最后通过 Softmax 函数输出分类结果。选用 ReLU 作为激活函数引入非线性。为了避免产生额外的计算成本,我们在局部特征向量 $\boldsymbol{L}^i$ 和全局特征向量 $\boldsymbol{G}$ 之间设定了相同的维度,均为 64。

## 2　算法的硬件加速

### 2.1　定点量化

由于模型训练通常以浮点数的数据类型进行,而 FPGA 无专用的浮点运算单元,进行浮点数运算会消耗较多 DSP 和 LUT 资源。因此需要对输入数据和权重参数进行定点化处理。如表 1 所示,选择 32 位浮点数设计单个乘法器和加法器时,需要 3 个 DSP,浮点 32 位数加法需要 2 个 DSP 资源。16 位定点数只需 1 个 DSP,加法不消耗 DSP 资源。对输入数据和权重参数进行 16 bit 定点化,占用的内存将减少 1/2。

**表 1　资源消耗比较**

| 操作 | DSP | | LUT | |
|---|---|---|---|---|
| | Float32 | Fixed16 | Float32 | Fixed16 |
| 加法器 | 2 | 0 | 231 | 52 |
| 乘法器 | 3 | 1 | 144 | 99 |

### 2.2　卷积运算单元的硬件加速

卷积运算占据了模型 90%以上的运算次数,消耗大量的计算时间,因此对卷积单元的并行加速计算是硬件设计部分的核心。卷积运算具有并行特性,本文使用同通道多像素并行计算和多通道并行计算进行加速设计。

1)同通道多像素并行计算

(1)流水线约束

如图 2(a)所示,在未进行任何优化的情况下,每次运算需经过取数据、运算和存数据 3 个周期,对输入长度为 2 048 的单通道序列,用尺寸为 3 的卷积核进行一维卷积运算,共需消耗约 2 048×3×3 个时钟周期。

如图 2(b)所示,流水化优化在第 1 个数据运算的同时对第 2 个数据进行读取,在第 2 个数据运算的同时,并行执行第 1 个数据的写入和第 3 个数据的读取。对相同的输入长度为 2 048 的单通道序列进行卷积计算,仅需要消耗约 2 048×3+2 个周期,运算效率相较于串行计算提升近 3 倍。

(2)3×1 卷积核的 3 次乘法并行计算

一般来说,执行 3 次乘法运算需要 3 个周期,通过在高层次综合中添加循环展开(UNROLL)的约束命令,可以使三次乘法并行计算。高层次综合中的循环展开会消耗 3 个 DSP 计算资源,以牺牲面积和资源的代价来提升运算速度。传统的卷积运算 1 个周期内进行一次内存读写操作,循环展开后,需要在 1 个周期内对 3 个数据进行内存读

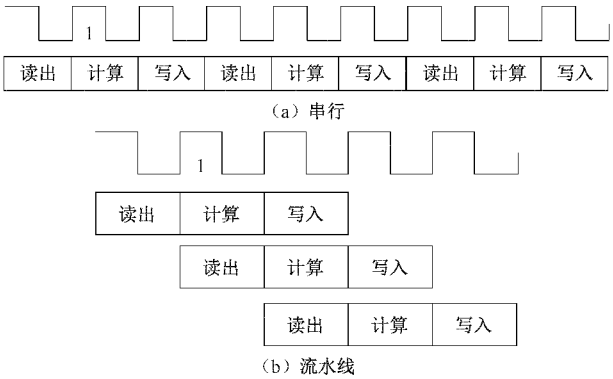



图 2　串行、流水线结构

写操作,因此对带宽的要求增高。将待计算的特征序列参数和权重参数传输到片上 BRAM 中,通过添加 ARRAY_PARTITION 约束命令将 BRAM 数组以 cyclic 的方式分割成 3 份,单个周期内支持对多个特征数据和权重参数的并行访问。对输入特征序列做卷积运算,当流水线工作后,消耗的时钟周期约为 2 048,相较于串行执行,运算周期减少了约 9 倍。

2)多通道并行计算

卷积运算在不同通道上具有并行运算的特性。如图 3 所示,$Win$ 为输入特征图长度,$Kx$ 为卷积核尺寸,$CHin$ 为输入通道数,$N$ 为通道并行度。多通道并行是指将权重参数和特征数据沿着输入通道方向切分成多个子块,同时从对应的权重子块和特征子块中取出多个数据并行计算。传感器数据的输入通道为 3 个,3 通道并行计算速度会相较于单通道计算提升 3 倍。考虑到网络每层输入通道和输出通道均为 16 的倍数,取 $N=16$,进行 16 通道并行计算。

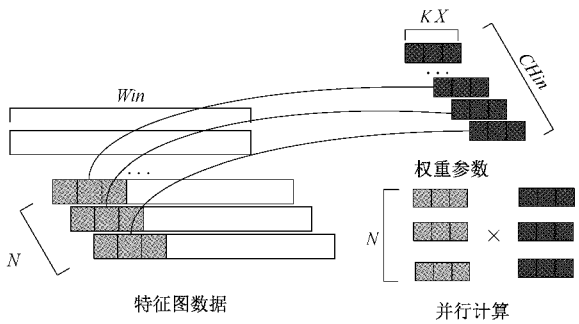



图 3　多通道并行计算

### 2.3　参数重排序

以加速度传感器采集的 $XYZ$ 三轴传感器数据为例,序列长度为 2 048,内存中的存储方式如图 4(a)所示,读取数据时,先传输“$X$”通道 2 048 个像素点,再依次传输“$Y$”通道和“$Z$”通道各 2 048 个像素点。通过这个排序传输,需等待最后一个通道传输完成才能结束计算。会导致耗费大量的缓存来储存运算的中间结果。因此,将输入序列像素在内存中的存储重新排序。重排序后的参数如图 4(b)

所示，首先传输“$X$”“$Y$”“$Z$”三轴通道的第 1 个像素点，依次传输三通道的第 2、第 3 个像素点，直至序列数据传输完成。通过这种传输方式，因为首先进行不同通道上的循环计算，可以直接叠加得到输出结果，避免消耗过多缓存来存储中间运算结果。

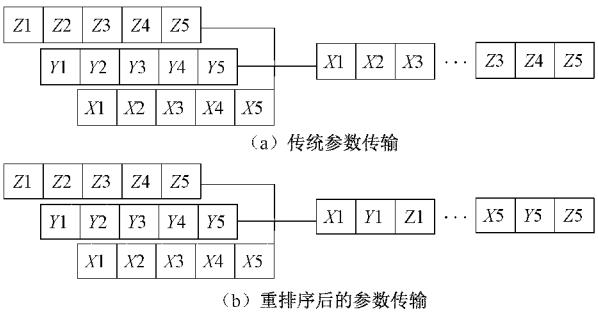



图 4　参数重排序

### 2.4　注意力机制的硬件加速

注意力机制由点积计算、Softmax 计算和加权平均计算 3 部分组成。由式(1)可知，点积运算是由全连接层输出向量和特征图特定通道的元素进行乘累加计算获得。为了提升计算效率，一方面采用多数据并行计算的方法，分别从输入特征数据和全连接层数据中一次性取出 16 个对应参数进行并行乘加计算；另一方面改进特征数据传输的方式，将待计算的特征数据分段传输而不是一次性传输到片上 BRAM，并以流水线的方式进行点积计算。第 3 部分为加权计算，具体计算由式(3)所示，该计算的优化方式和点积运算相同，使用 16 通道并行计算、分段数据传输和流水线计算等方式提升计算效率。

如式(2)所示，第 2 部分 Softmax 函数的计算包括大量的指数运算和除法运算，而 FPGA 中实现指数运算的主要方法有查表法、CORDIC 算法、泰勒级数展开和多项式拟合等。考虑到资源消耗、运算时间和精度等因素，本文采用分段函数拟合逼近的方案来实现指数运算，各区域拟合函数如表 2 所示，将指数函数在大区间内分为若干个区间，由于不同区间的拟合数值波动较大，因此每个区间的拟合函数使用不同的定点方法。

多项式拟合是利用 MATLAB 的 polyfit 函数进行的。根据注意力运算过程中进入 Softmax 层的输入值范围，将指数函数在[12，−12]区间内分成 12 个区间段。每个区间段以 0.001 步长拟合，最高阶数设为 3 次，依次求出各阶系数。除法运算在 HLS 的编译中会默认消耗大量计算资源，超出了板载资源限制，因此将除法运算转移到 PS 端进行。

### 2.5　FPGA 硬件实现

CNN 模型的不同层之间的计算相互独立，本文将不同的计算层封装成可复用 IP 核，通过时分复用调用 IP 核构建完整的带注意力机制的卷积神经网络识别算法。由于注意机制模块中的除法运算在 PS 端运算，无法将 3 个模块封装在单个 IP 核中，将点积运算和 Softmax 函数中的指数运算以层间流水的方式封装成单个 IP 核，加权运算也封装成另一个 IP 核。

表 2　不同区间的拟合函数

| 区间 | 拟合函数 |
|---|---|
| [−12，−10] | $1.670\times10^{-5}$ |
| [−10，−8] | $1.234\times10^{-3}$ |
| [−8，−6] | $1.606\times10^{-4}x^3+3.862\times10^{-3}x^2+3.137\times10^{-2}x+8.635\times10^{-2}$ |
| [−6，−4] | $1.187\times10^{-3}x^3+2.140\times10^{-2}x^2+1.139\times10^{-1}x+2.791\times10^{-1}$ |
| [−4，−2] | $8.770\times10^{-3}x^3+1.062\times10^{-1}x^2+4.461\times10^{-1}x+6.762\times10^{-1}$ |
| [−2，0] | $6.481\times10^{-2}x^3+2.919\times10^{-1}x^2+9.561\times10^{-1}x+9.959\times10^{-1}$ |
| [0，2] | $4.880\times10^{-1}x^3+2.257\times10^{-2}x^2+1.231\times10x+9.756\times10^{-1}$ |
| [2，4] | $3.538\times10x^3-2.106\times10^{1}x^2+5.088\times10^{1}x-3.862\times10^{1}$ |
| [4，6] | $2.614\times10^{1}x^3-3.125\times10^{2}x^2+1.312\times10^{3}x-1.869\times10^{3}$ |
| [6，8] | $1.932\times10^{2}x^3-3.468\times10^{3}x^2+2.125\times10^{4}x-4.369\times10^{4}$ |
| [8，10] | $1.427\times10^{3}x^3-3.419\times10^{4}x^2+2.766\times10^{5}x-7.530\times10^{5}$ |
| [10，12] | $1.055\times10^{4}x^3-3.159\times10^{5}x^2+3.181\times10^{6}x-1.075\times10^{7}$ |

FPGA 硬件系统设计如图 5 所示，为了提高数据从片外存储器 DDR 到片上存储器的传输效率，使用直接内存读取(direct memory access，DMA)进行片内外数据的传输，数据传输总线为 AXI4-Stream，AXI4-Stream 数据传输没有地址线，传输效率高。系统启动时，将 AXI4-Stream 数据通过 AXI-Switch0 进入所设计的相应计算单元内，各 IP 计算核并排排列，每次仅进行 1 个 IP 核的加速运算，计算结束后将计算结果通过 AXI-Switch1 输出到片外 DDR 中。1 个计算模块结束后，开启新一轮的加速计算。将 DMA 的数据位宽设为 256，适用于 16 bit 定点数的 16 通道并行计算。

## 3　实验结果及分析

### 3.1　数据集

1)UCI_HAR 数据集：本文使用经典的 UCI_HAR 数据集验证模型对传感器数据识别的有效性。UCI_HAR 数据集是由三星智能手机中的加速度计和陀螺仪分别以 50 Hz 的固定速率收集三轴线性加速度和三轴角速度数据制作而成。数据集分别由 3 个动态动作(步行、上楼、下

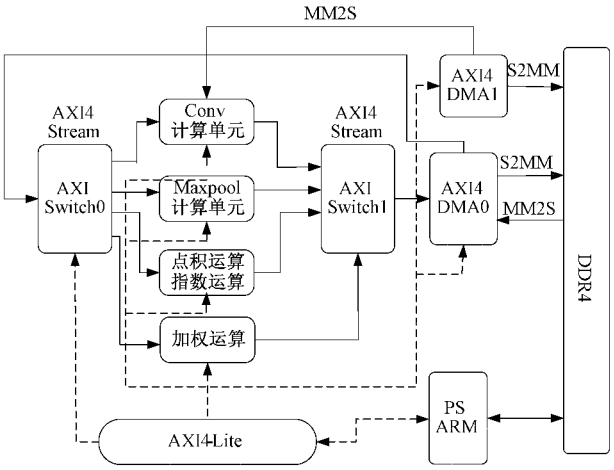



图 5 FPGA 硬件系统设计

楼),3 个静态动作(站、坐、躺)和 3 个静态活动的转换(站坐、坐站、站躺、躺站、坐躺、躺坐)组成。对传感器数据进行滤波除噪,在 2.56 s 的时间间隔和 50%重叠的固定宽度滑动窗口中进行采样(窗口宽度为 128)。本实验利用数据集的 3 个动态动作和 3 个静态动作进行实验。UCI_HAR 数据集各样本数目如表 3 所示。

**表 3 UCI_HAR 数据集**

| 活动类别 | 样本数目 |
|---|---|
| 行走 | 1 722 |
| 上楼梯 | 1 544 |
| 下楼梯 | 1 407 |
| 坐着 | 1 801 |
| 站立 | 1 979 |
| 躺下 | 1 958 |
| 总结 | 10 411 |

2)弱标签数据集:弱标签数据集包括 5 种活动:“步行”、“慢跑”、“跳跃”、“上楼”和“下楼”,其中“步行”是背景活动。该数据由 iPhone 的 3 轴加速度传感器采集,放置在 10 名志愿者的右裤袋中,智能手机加速度计的采样率为 50 Hz。通过区分不同的参与者来划分原始数据,然后使用固定宽度的 40.96 s 滑动窗口(窗口宽度为 2 048)对数据进行采样。最后,收集到的弱标记数据集共 91 266 条数据,其中 70%用于训练,30%用于测试。弱标签数据集各行为样本总数如表 4 所示。

**表 4 弱标签数据集**

| 活动类别 | 样本数目 |
|---|---|
| 上楼梯 | 21 087 |
| 下楼梯 | 23 597 |
| 跳跃 | 23 434 |
| 慢跑 | 23 148 |
| 总结 | 91 266 |

弱标签采样数据实例如图 6 所示。每条数据由 2 048 个采样点组成,纵坐标为每个点的幅值。

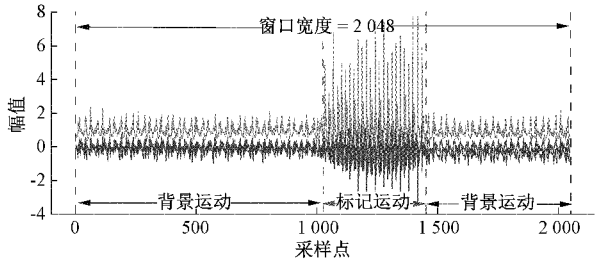



图 6 弱标签传感器数据实例

### 3.2 实验环境

实验采用的是 Xilinx 公司的 XCZU3EG 芯片作为硬件平台,工作频率为 150 MHz。芯片中包括 PS(processing system)和 PL(programmable logic)两部分,PS 端集成了四核 ARM Cotrex-A53 处理器。PL 端即 FPGA 部分,其中 BRAM_18K、DSP、FF 和 LUT 硬件资源分别为 432、360、141 120 及 70 560。

### 3.3 注意力机制效果

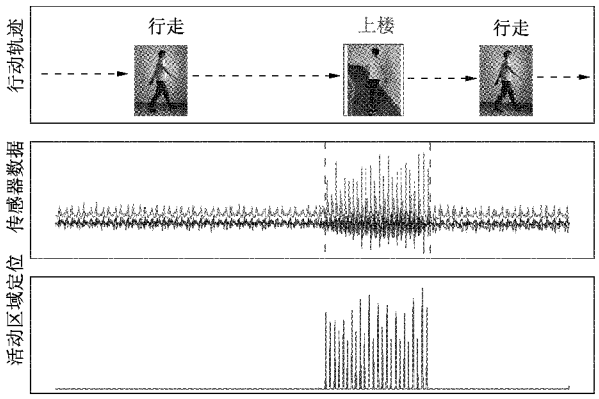
加入注意力机制后的模型准确率如表 5 所示,UCI_HAR 数据集是由标记良好的传感器数据组成,背景噪声小,因此与传统方法相比改进较小,但识别准确率仍然达到了 89.14%,验证了带注意力机制的 CNN 模型识别传感器数据的可靠性。而在弱标签数据集上,注意力机制对基础 CNN 模型有明显的性能改进,识别准确率相对于基础模型提升了 1.79%。

**表 5 不同模型在 UCI_HAR 和弱标签数据集识别准确率**

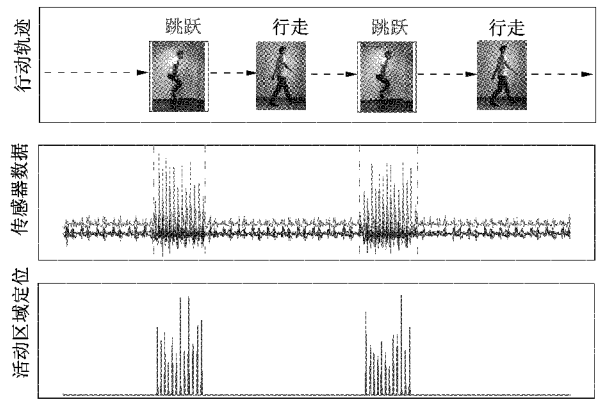
| 数据集 | 基础 CNN 模型 | 带注意力机制模型 |
|---|---|---|
| UCI_HAR 数据集/% | 89.64 | 90.21 |
| 弱标签数据集/% | 88.48 | 91.02 |

注意力机制可以对弱标签数据进行定位,如图 7 所示。图 7(a)的上两幅图分别为单采样行为数据的运动轨迹和传感器数据,背景运动(行走)中夹杂着单次标记运动(上楼梯),最下方一张图为在 FPGA 上计算的得到注意力机制中的序列权重输出,其中兼容性得分显著的区域即为活动的区域,实现了对活动区域的定位。图 7(b)的上两幅图分别为多采样行为数据的运动轨迹和传感器数据,背景运动(行走)中夹杂着多次标记运动(跳跃),最下方一张图中两段显著的区域为两次被标记活动的区域,实现了多次运动活动区域的定位。

FPGA 中各模块的硬件资源消耗如表 6 所示,卷积模块由于并行计算消耗了许多 DSP 计算资源;注意力机制中的指数运算转化为了乘法运算,也耗费了大量的 DSP 硬件计算资源。BRAM_18K 用于存储片上缓存,卷积加速运算模块中有大量的特征缓存和权重缓存,因而消耗的




(a) 单次标记运动区域定位图




(b) 多次标记运动区域定位图

图 7　注意力机制对行动区域定位示意效果图

BRAM_18K 较多。另外，共消耗了 30%的 FF 寄存器资源和 55%的 LUT 查找表资源。

**表 6　FPGA 硬件资源消耗**

| 模块 | 卷积 | 池化 | 注意力机制 |
|---|---|---|---|
| BRAM_18K | 144(33%) | 8(2%) | 24(6%) |
| DSP48E | 48(14%) | 0 | 149(41%) |
| FF | 21 318(15%) | 8 873(6%) | 12 214(9%) |
| LUT | 23 308(33%) | 6 493(9%) | 9 130(13%) |

### 3.4　性能指标

选用识别精度、计算性能和功耗效率等指标对系统性能进行分析。识别精度即测试正确的样本占总样本的比例。计算性能可用单位时间内识别的样本个数衡量，类似于每秒处理的图像帧数(frame per second，FPS)，计算方式如式(4)所示。

$$\text{Performance(frame/s)}=\frac{\text{Num(frames)}}{\text{Time(s)}} \tag{4}$$

运行功耗效率可用单位功耗处理的样本个数来表示，其计算方式如式(5)所示。

$$\text{Power Efficiency(frame/s)}=\frac{\text{Performance(frame/s)}}{\text{Power(J/s)}} \tag{5}$$

系统的设计指标如下：

1)系统的识别精度达到 85%以上。

2)计算性能高于 20 frames/s。

3)系统的运行功耗低于 5 W。

### 3.5　识别精度

1)UCI_HAR 数据集

应用于 UCI_HAR 的行为识别模型除了输入数据的尺寸不同，其余模型结构与图 1 一致。选取 UCI_HAR 数据集每种行为各 100 条数据存入 SD 卡，用于验证 FPGA 端的行为识别准确率。选取量化后的数据位宽为 16 bit 的模型进行推断计算。如表 7 所示，经过 600 次迭代，平均识别为 89.00%，识别精度略低于软件平台准确率 90.21%，这是由于量化造成的精度损失。识别准确率达到了系统的设计要求。

**表 7　FPGA 端 UCI_HAR 数据集识别准确率**

| 行为类别 | 迭代次数 | 准确率/% |
|---|---|---|
| 行走 | 100 | 92.00 |
| 上楼梯 | | 89.00 |
| 下楼梯 | | 88.00 |
| 坐着 | | 87.00 |
| 站立 | | 89.00 |
| 躺下 | | 89.00 |

2)弱标签数据集

选取弱标签每种行为各 200 条数据存入 SD 卡，用于验证 FPGA 端的弱标签数据集的识别准确率。FPGA 端各行为类别识别准确率如表 8 所示，经过 800 次迭代，FPGA 端下楼梯活动的准确率最高，跳跃活动的准确率最低，平均准确率为 90.00%，识别精度略低于软件平台准确率 91.02%。达到了预期设计目标。

**表 8　FPGA 端弱标签数据集识别准确率**

| 行为类别 | 迭代次数 | 准确率/% |
|---|---|---|
| 上楼梯 | 200 | 90.00 |
| 下楼梯 | | 91.50 |
| 跳跃 | | 88.00 |
| 慢跑 | | 90.50 |

### 3.6　计算性能与功耗评估

1)UCI_HAR 数据集

FPGA 部署 UCI_HAR 数据的识别模型时，系统对应的计算性能和运行功耗如表 9 所示。当工作频率为 150 MHz 时，在 FPGA 上识别一个样本需耗时 8.47 ms，计算性能为 118.06 frames/s，相较于单核 ARM-A53 处理器，提升了约 44.69 倍。消耗功耗为 2.204 W，换算可知功耗效率为 53.57 frams/J，符合系统计算性能高于 20 frames/s，且相

较于纯软件 ARM 运算速度提升 40 倍以上,运行功耗低于 5 W 的设计要求。

**表 9 UCI_HAR 数据集性能及功耗比较**

| 计算平台 | 数据位宽 | 计算延时/ms | 功耗/W |
|---|---|---|---|
| ARM | Float-32 | 378.52 | 1.623 |
| FPGA | Fixed-16 | 8.47 | 2.204 |

2)弱标签数据集

FPGA 部署弱标签数据集的识别模型与 ARM 平台的运算速度和功耗对比如表 10 所示,在 150 MHz 的工作频率下,在 FPGA 上处理一条弱标签数据需耗时 38.63 ms,计算性能为 25.89 frames/s,相较于单核 ARM-A53,提升了约 54.15 倍。消耗功耗也仅为 2.204 W,换算可知功耗效率为 11.75 frams/J,完全符合低功耗和低时延的设计要求,适用于嵌入式设备应用。

**表 10 弱标签数据集性能及功耗比较**

| 计算平台 | 数据位宽 | 计算延时/ms | 功耗/W |
|---|---|---|---|
| ARM | Float-32 | 2 130.44 | 1.623 |
| FPGA | Fixed-16 | 38.63 | 2.204 |

### 3.7 工作对比

由于文献[13-14]并未给出功耗和性能指标,功耗方面,根据文献给出的工作频率和资源使用率,根据 Xilinx 的功耗推算器获取运行功耗。性能指标方面,本文按照式(6)进行了计算,GFLOPS 是指每秒十亿次浮点运算的次数,与 GOPS 近似,是衡量硬件计算性能的指标,单个二维卷积层的计算公式如式(6)所示。

$$FLOPs = 2 \times H_o \times W_o \times K_x \times K_y \times Cin \times Cout \tag{6}$$

式中:$H_o$ 和 $W_o$ 分别位输出特征图的长度和宽度,$K_x$ 和 $K_y$ 分别卷积核的尺寸,$Cin$ 和 $Cout$ 分别为输入通道个数和输出通道个数。

对加速器的综合性能进行评估,与其他人的工作进行对比。如表 11 所示,由于其他人的工作中无注意力机制模块,因此仅对卷积计算模块的性能进行对比。卷积模块的功耗为 1.735 W。文献[13]数据位宽最低,但性能和能耗比均小于本文。文献[14]虽然性能和能耗比高于本文,但该设计消耗了 1 180 个 DSP,是本文设计的 8.68 倍,无法应用于资源受限的嵌入式设备。文献[15]的 DSP 资源略高于本文设计,但本文的工作频率高于文献[15],性能和能耗比也高于该设计。实验结果表明,使用 FPGA 作为边缘计算平台对基于可穿戴传感器的弱标签人体行为数据进行识别和定位的是高效可行的,并且整个推断过程达到了低功耗,低时延的要求,在边缘计算和嵌入式应用等方面有良好的应用场景。

**表 11 相关工作比较**

| 相关参数 | 文献[13] | 文献[14] | 文献[15] | 本文 |
|---|---|---|---|---|
| 实验平台 | Xc7z020 | VC707 | Xc7z020 | XCZU3EG |
| 时钟频率/MHz | 100 | 100 | 100 | 150 |
| 数据位宽/bits | 8 | 16 | 16 | 16 |
| 功率/W | 1.733 | 1.950 | 1.685 | 1.735 |
| 性能/GFLOPS | 0.265 | 6.532 | 1.730 | 1.977 |
| 能耗比/(GFLOPS·W$^{-1}$) | 0.153 | 3.354 | 1.027 | 1.139 |

## 4 结 论

本文设计了一种基于可穿戴传感器的行为识别算法并在 FPGA 实现。在高层次综合中设计卷积、最大值池化和注意力机制等计算模块并通过流水线、并行计算和定点量化等方法实现硬件加速。实验结果表明,对行为识别算法的推断过程达到了低功耗、低延时的设计要求。未来的工作有两方面改进:在 FPGA 上实现传感器信号预处理部分;最大化利用计算资源提升卷积运算模块的并行度。

## 参考文献

[1] 郑增威,杜俊杰,霍梅梅,等. 基于可穿戴传感器的人体活动识别研究综述[J]. 计算机应用,2018,38(5):1223-1229, 1238.

[2] 杨坤,程浩南,殷允杰,等. 人体健康监测用织物基智能可穿戴传感器研究进展[J]. 纺织导报,2021(1):71-76.

[3] 王震宇,张雷. 基于深度卷积和门控循环神经网络的传感器运动识别[J]. 电子测量与仪器学报,2020,34(1):1-9.

[4] 刘焰强,戚正伟,管海兵. FPGA 加速系统开发工具设计:综述与实践[J]. 软件学报,2020,31(10):3087-3099.

[5] 季玉坤,高向东,刘倩雯,等. 焊接缺陷磁光成像卷积神经网络识别方法[J]. 仪器仪表学报,2021,42(2):107-113.

[6] 刘素行,吴媛,张军军. 基于 YOLO v3 的交通场景目标检测方法[J]. 国外电子测量技术,2021,40(2):116-120.

[7] 皮乾东，邵玉斌，龙华，等. 汉语语句算式化融合句法分析[J]. 电子测量技术，2020，43(6)：123-127.

[8] HE J，ZHANG Q，WANG L，et al. Weakly supervised human activity recognition from wearable sensors by recurrent attention learning[J]. IEEE Sensors Journal，2018，19(6)：2287-2297.

[9] WANG K，HE J，ZHANG L，et al. Attention-based convolutional neural network for weakly labeled human activity recognition with wearable sensors[J]. IEEE Sensors Journal，2019，19(17)：7598-7604.

[10] 陈浩敏，姚森敬，席禹，等. YOLOv3-tiny 的硬件加速设计及 FPGA 实现[J]. 计算机工程与科学，2021，43(12)：2139-2149.

[11] ZHANG C，LI P，SUN G，et al. Optimizing FPGA-based accelerator design for deep convolutional neural networks[C]. the 2015 ACM/SIGDA International Symposium，ACM，2015.

[12] 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.

[13] BAO C，XIE T，FENG W，et al. A power-efficient optimizing framework FPGA accelerator based on winograd for YOLO[J]. IEEE Access，2020，8：94307-94317.

[14] 郑文凯，杨济民. 在 FPGA 上实现及优化加速卷积神经网络的方法[J]. 山东师范大学学报(自然科学版)，2019，34(2)：186-192.

[15] 满涛，郭子豪，曲志坚. 卷积神经网络的 FPGA 并行加速设计与实现[J]. 电讯技术，2021，61(11)：1438-1445.

## 作者简介

**吴宇航**，硕士研究生，主要研究方向为 FPGA 硬件加速器。

E-mail：18851761006@163.com

**何军**(通信作者)，博士，副教授，主要研究方向为机器学习、计算机视觉。

E-mail：jhe@nuist.edu.cn