

电子科技大学
UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

专业学位硕士学位论文

MASTER THESIS FOR PROFESSIONAL DEGREE



论文题目 基于移动终端的人体姿态检测技术
研究与实现

专业学位类别	<u>工程硕士</u>
学 号	<u>201922010310</u>
作者姓名	<u>袁环宇</u>
指导教师	<u>武 畅 副教授</u>
学 院	<u>信息与通信工程学院</u>

分类号 _____ 密级 _____ 公开 _____
UDC^{注1} _____

学位论文

基于移动终端的人体姿态检测技术研究与实践

(题名和副题名)

袁环宇

(作者姓名)

指导教师 武畅 副教授
电子科技大学 成都
(姓名、职称、单位名称)

申请学位级别 硕士 专业学位类别 工程硕士
专业学位领域 电子与通信工程
提交论文日期 2022年3月20日 论文答辩日期 2022年5月20日
学位授予单位和日期 电子科技大学 2022年6月
答辩委员会主席 文光俊
评阅人 肖卓凌、杨帆

注1: 注明《国际十进分类法UDC》的类号。

Research and Implementation of Human Posture Detection Technology Based on Mobile Terminal

A Master Thesis Submitted to
University of Electronic Science and Technology of China

Discipline Master of Engineering

Student ID 201922010310

Author Yuan Huanyu

Supervisor Wu Chang

School School of Information & Communication
Engineering

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：袁环宇

日期：2022年 5月 19日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：袁环宇

导师签名：武畅

日期：2022年 5月 19日

摘要

近年来,随着人们对身体健康的重视,基于传感器数据的人体姿态检测成为了热门研究领域,这种方案在隐私保护上的表现优秀。现在大部分用户使用的手机都集成了多种常用传感器,使得基于手机传感器的人体姿态检测技术具有巨大的应用价值。

目前基于手机传感器的人体姿态检测研究以机器学习为主,这些方法的缺陷在于人工提取特征的效果不好。针对传统方法的缺陷,本文主要采用深度学习方法进行人体姿态检测。本文使用卷积神经网络对传感器数据分类,通过实验设置网络结构和超参数以获得性能最佳的网络。基于独立提取不同传感器数据特征的思想,提出一种多通道卷积神经网络,该网络使用多个权值不同的卷积头并行提取特征,在公开数据集上的检测准确率达到 94.231%。然后使用长短时记忆网络对传感器数据分类,并将卷积神经网络和长短时记忆网络在提取特征上的优势结合起来,能同时提取传感器数据的时空特征。提出一种 ConvLSTM 网络,该网络将长短时记忆网络中的门控单元的矩阵乘法换成卷积运算,在公开数据集上的检测准确率达到 94.639%。通过对比,本文提出的两种用于人体姿态检测的网络性能均优于传统的机器学习模型。

本文将性能最优的 ConvLSTM 模型部署到移动端,以此为基础设计并实现了人体姿态检测系统,主要包括手机 APP 和后台管理系统两部分。在日常应用中受到噪声和随机性的影响,该模型的检测准确率为 85.2%。对于静态活动和动态活动的区分效果依然很优秀,二分类准确率为 97.667%。

关键词: 人体姿态检测, 手机传感器, 深度学习, 卷积神经网络, 长短时记忆网络

ABSTRACT

In recent years, because of people's attention to health, human posture detection based on sensor data has become a hot research field, which performance in privacy protection is excellent. Most users use mobile phone which integrates a variety of sensors, making the technology of human posture detection based on mobile phone sensor have great value in application.

The research of human posture detection based on mobile phone sensor mainly uses machine learning methods which have poor performance in manual feature extraction. In order to solve the defect of traditional method, deep learning method is used to detect human attitude in this thesis. In this thesis, CNN is used to classify the sensor data, and the network structure and super parameters are set through experiments to obtain the network with the best performance. Based on the idea of extracting data features from different sensors independently, a multi-channel CNN is proposed. The network uses multiple convolution heads with different weights to extract features in parallel, the detection accuracy on the public data set is 94.231%. Then, LSTM is used to classify the sensor data, and the advantages of CNN and LSTM in feature extraction are combined to extract the temporal and spatial features of sensor data at the same time. A ConvLSTM network is proposed, which replaces the matrix multiplication of the gating unit in LSTM with convolution operation, the detection accuracy on the public data set is 94.639%. Through comparison, the performance of the two networks proposed in this thesis for human posture detection is better than the traditional machine learning model.

Then the ConvLSTM model with the best performance is deployed to the mobile terminal. Based on the model, a human posture detection system is designed and implemented, mainly including mobile app and background management system. Affected by noise and randomness in daily application, the detection accuracy of the model is 85.2%. The performance of secondary classification is excellent, the accuracy of classifying static activities and dynamic activities is 97.667%.

Keywords: Human Attitude Recognition, Mobile Phone Sensor, Deep Learning, Convolutional Neural Network, Long Short-term Memory Network

目 录

第一章 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	2
1.2.1 国内研究现状	3
1.2.2 国外研究现状	3
1.3 论文主要研究内容	4
1.4 论文结构	4
第二章 人体姿态检测相关技术研究	6
2.1 手机传感器在人体姿态检测中的应用	7
2.2 基于机器学习的人体姿态检测	8
2.2.1 支持向量机 (SVM)	9
2.2.2 朴素贝叶斯 (NB)	10
2.2.3 决策树 (DT)	10
2.3 基于深度学习的人体姿态检测	11
2.3.1 卷积神经网络 (CNN)	13
2.3.2 循环神经网络 (RNN)	15
2.3.3 长短时记忆网络 (LSTM)	16
2.4 本章小结	18
第三章 基于 Multi-channel CNN 的人体姿态检测	19
3.1 数据集构建	19
3.1.1 经典数据集分析	19
3.1.2 数据采集	22
3.1.3 数据预处理	23
3.2 一维卷积神经网络 (1D CNN)	24
3.2.1 1D CNN 模型搭建	25
3.2.2 超参数调整	27
3.2.3 评价指标	31
3.3 多通道卷积神经网络 (Multi-channel CNN)	35
3.4 实验结果对比	37
3.5 本章小结	39

第四章 基于 ConvLSTM 的人体姿态检测.....	40
4.1 循环神经网络（RNN）及其缺陷.....	40
4.2 长短时记忆网络（LSTM）.....	41
4.2.1 LSTM 模型搭建.....	41
4.2.2 超参数调整.....	42
4.3 1D CNN-LSTM 网络设计.....	46
4.4 卷积循环神经网络（ConvLSTM）.....	50
4.5 实验结果对比.....	54
4.6 本章小结.....	56
第五章 人体姿态检测系统设计与实现.....	57
5.1 人体姿态检测系统概述.....	57
5.2 移动端应用实现.....	58
5.2.1 神经网络模型的移动端转化与部署.....	58
5.2.2 数据采集模块和检测模块.....	59
5.2.3 APP 功能设计.....	62
5.3 后台管理平台实现.....	62
5.4 实验及结果分析.....	64
5.5 本章小结.....	66
第六章 总结与展望.....	67
6.1 总结.....	67
6.2 展望.....	68
致 谢.....	69
参考文献.....	70
攻读硕士学位期间取得的成果.....	73

第一章 绪论

1.1 研究背景及意义

人体姿态检测（Human Attitude Recognition, HAR）技术近年来被提出，主要通过一些摄像头以及传感器等设备对人体特定动作进行检测^[1]。它在医疗健康、人机交互等领域都有重要作用，特别是在健康监测方面，通过远程监测患者使残疾和老年患者得到更好的诊断治疗。

伴随着社会发展，人们的生活压力也在增加。上班族长期久坐缺乏运动导致腰肌劳损，过度摄入食物导致身体肥胖，所以人们对与健康状况的关注日益增加。而人体姿态检测技术的发展为医疗健康领域注入了新的活力，通过姿态检测设备可以记录自己的运动状况和活动时间，通过数据分析自己的健康状况。目前许多厂商都有运动健康相关的设备，通过数据监测，运动提醒等软件服务，提醒用户适当锻炼保持健康^[2]。

对于专业运动来说，也同样有运动情况监测方面的需求。通过设备记录运动速度，心跳频率等指标，针对训练表现制定下一步的训练计划，合理调整训练强度，科学指导运动员进行训练^[3]。同时对于一些过度训练的情况，及时发出警报，以免运动员身体受到不可逆的损伤。

最近各国的人口数据报告表明，世界各国的人口结构正在迅速老龄化。例如，对欧洲主要年龄组人口结构变化的预测表明，到 2060 年，老年人（即 65 岁以上的人）将接近其人口的 30%。我国 2020 年进行的第七次全国人口普查数据显示^[4]，在过去的十年间 60 岁以上的人口占比上升 5.44%，65 岁以上占比 13.5%。数据表明我国人口老龄化正在加速，预计到 2060 年老年人口占比将达到 35%。为应对中国老龄化社会的到来，党中央以及国务院对此也做出了积极应对，党的十九大工作报告提到要着力发展医疗，养老服务业等养老产业。

在中国的社会结构中，大量的年轻人外出务工，异地就业，对于老年人的健康状况变化无法及时了解，在家中出现突发疾病也无法及时开展救助，往往会错失最佳救助机会，导致情况恶化。所以，在老年人能够在全天进行常规活动的同时，对他们的健康和福祉进行持续监督是非常有必要的。对于这一问题，使用人体姿态检测设备是一个很好的解决方案^[5]。通过检测设备实时检测老人状态，老人以及子女可以及时的获取设备信息，对身体状况有所把控。同时如果发生摔倒等异常身体姿态，也能通过网络及时做出报警，让老人得到及时的救助，降低突发事件造成的风险。

随着智能手机的研发进步和价格下降，包括许多老年人都能使用智能手机进行一些通信、聊天等基本操作。各种手机内嵌传感器尺度越来越小，而测量精度却在不断提升，越来越多的新型传感器都被嵌入到手机当中，例如加速度计、陀螺仪、地磁传感器等。手机可以通过传感器采集各种环境信息，对于外界的感知更加多维。由此衍生出来的人机交互方式也更加新颖，比如体感游戏，手势控制。同时手机处理器也由最初单核升级为双核、四核甚至八核，手机的计算能力不断提高，对于采集到的传感器数据可以快速的处理分析。以前需要大量算力支持的机器学习、深度学习等方法也可以用在移动端处理数据，使得在人体姿态检测领域有更多的研究方法得以提出。

综上所述，利用手机多传感器数据对人体姿态进行检测具有重要研究价值及意义。通过采集人体日常活动的数据，并用机器学习、深度学习等方法对数据进行处理，提高人体姿态检测的准确率，是本文研究的主要目标。

1.2 国内外研究现状

近年来，人体姿态检测的研究愈发受到重视，这个方向的研究在各个领域都展现出了巨大的研究和商业价值，医疗健康、体感游戏等领域的公司推出了许多相关产品。在体感游戏领域，微软公司推出的 XBOX 体感手柄，任天堂在 2017 年推出的 Switch 掌上游戏机以及 2019 年推出的健身环上，都推出了大量结合设备游玩的体感游戏。对于非常细微的手势以及身体动作，设备都能有效捕捉并与主机交互，从而丰富玩家游戏体验。在医疗健康领域，华为、小米等公司都相继推出了运动手环产品。运动手环上的传感器设备会监测心率、血压、运动状况等数据，并连接到用户的手机，通过软件分析显示用户的每日活动以及健康状况。

根据使用数据类型的不同，目前的人体姿态检测研究主要有两类识别方法^[6]：一种基于计算机视觉，通过摄像装置采集视频获取人体姿态信息来进行分类识别。另一种基于传感器数据，通过可穿戴的传感器设备采集人体活动数据来进行分类识别。使用传感器数据用于人体姿态识别有着其得天独厚的优势。通过在研究者身上佩戴传感器设备，采集人体运动数据，更加贴合用户的日常生活，并且不受外界环境的影响，使用者到处移动也依然能保持检测状态。同时传感器数据量比视频数据量小了很多，在数据处理的过程也更加简单，无需处理二维的连续数据，只需要处理一维的时间序列，所以检测结果的时间延迟也比较低。最重要的是，这种方法能很好的保护用户的个人隐私，因为它采集数据的过程中只涉及到用户的运动数据，这些传感器数据相比于图像数据没有那么大的私密性^[7]。

众多研究者在基于传感器数据的人体姿态检测领域投入了许多精力，研究的

方法也有许多转变。传统的方法主要有支持向量机、贝叶斯方法、决策树以及随机森林等等。这些传统方法主要依赖从运动过程中提取传感器数据的特征作为输入对模型进行训练^[8]。比如一些窗口化的特征值，包括最值、均值、标准差、幅度以及功率谱密度等时频域特征。这种方法的局限性在于，需要研究人员有信号处理相关领域的知识，并且要对原始数据有足够的分析认识，所以有时分类效果并不好。随着设备计算能力的发展，也有研究人员提出利用深度学习的方法来做姿态检测。将特征提取的工作交给神经网络模型来处理，它们能够从原始数据学习自动提取特征，并能获得比传统方法更好的识别性能。

1.2.1 国内研究现状

随着智能手机使用的普及以及手机制作工艺的进步，越来越多的传感器被嵌入到手机当中，包括加速度计、陀螺仪、光传感器等等。并且手机的计算能力不断提高，使得处理数据的方式不只限于手机端采集数据上传至服务器进行处理了，越来越多的研究者开始把手机当成客户端-服务端融合平台，在手机上直接完成数据采集-数据处理-结果显示的闭环工作了。最早的基于手机传感器识别的研究，一般只采集加速度计数据，并人工提取时间窗口特征，将特征加入到机器学习分类算法进行识别分类。

重庆大学的何理通过提取加速度传感器数据和传感器倾角方向进行摔倒检测算法研究，提出了一种基于时域特征的阈值判定算法。还通过快速傅里叶变换和离散余弦变换将问题转到频域进行分析，并使用支持向量机算法分析频域特征，提出了一种基于频域特征的阈值判定算法，摔倒检测准确率达到 95%^[9]。天津大学的罗薇通过提取加速度传感器的频域特征，结合图论的邻接矩阵和图嵌入理论，提出了一种基于多图理论的人体姿态识别算法，在开源数据集 SCUT-NAA 上的准确率达到了 83%^[10]。东南大学的冯增涛使用身体多部位加速度计和陀螺仪数据，融合集成学习和随机森林的算法原理，提出了一种基于随机森林的集成学习算法，在公开数据集 PAMAP 上取得了 85%的检测准确率^[11]。厦门大学的熊富海使用深度学习方法处理传感器数据，融合 CNN 和 RNN，提出了 CRNN 模型，并通过改变卷积维度，提出了变维卷积模型，在公开数据集上取得了非常好的效果^[12]。

1.2.2 国外研究现状

WISDM 实验室的 Jennifer R. Kwapisz 等人最早开始做使用三轴加速度传感器数据进行人体姿态检测的研究。他们在论文中详细介绍了一个使用基于手机三轴加速度计执行姿态检测的系统，详细介绍了通过实验采集加速度计数据集的过程，

并公布了他们的实验数据集——WISDM 数据集供研究者下载研究。在实验数据集上采用决策树、逻辑回归以及多层神经网络等分类算法来验证数据集的正确性，得到了很好的表现^[13]。Davide Anguita 等人构建了名为 UCI-HAR 的数据集^[14]，该数据集采集了加速度计、陀螺仪等三轴传感器的共九维数据。并提出了一种硬件友好的多分类支持向量机^[15]。该方法在躺下等静态活动的识别效果比较好，但对于上下楼活动的分类效果就略显一般。Moiz Ahmed 等人建立了一个数据集——SMotion 用于老人摔倒检测，将六轴传感器数据分为了年龄组和体重组，用于不同的分类算法需求。还开发了一个使用三种分类器的特征选择和分类系统，即支持向量机（SVM）、K-近邻（KNN）和神经网络（NN）。研究的结果表明 KNN 是最准确的分类器，年龄组的最大准确率为 96%，体重组的最大准确率为 93%^[16]。

总的来说，在基于传感器数据实现人体姿态检测这个领域上，国内外很多学者都提出了值得后人研究的创新方法。我相信伴随着手机、传感器等硬件的升级，这一方面的研究也会越来越多。

1.3 论文主要研究内容

本文研究了大量基于传感器数据的人体姿态检测相关论文，对使用深度学习模型进行人体姿态检测方向进行深入研究，并使用其中性能最好的模型搭建人体姿态检测系统。本文主要研究内容如下：

（1）通过分析经典数据集 UCI-HAR，说明了使用传感器数据进行人体姿态检测的可行性。针对 UCI-HAR 静态活动数据可分性差的缺点，改进了数据采集方式，制作了自建数据集。

（2）使用 1D CNN 模型进行人体姿态检测，对 1D CNN 的网络结构和超参数对性能的影响进行实验。并针对 1D CNN 模型提取特征的缺陷，提出一种 Multi-channel CNN 网络，使用不同的卷积头对不同传感器数据独立提取特征。

（3）结合 CNN 网络和 LSTM 网络在特征提取上的优势，提出了用卷积头作为前端的 CNN-LSTM 网络和门控单元的矩阵乘法换成卷积运算的 ConvLSTM 网络。

（4）搭建人体姿态检测系统，实现人体姿态在移动端离线检测，活动记录在线上上传后台管理系统，实现姿态检测、用户管理、健康监控等功能。

1.4 论文结构

本文共分为六章，各章内容如下：

第一章绪论。首先介绍了人体姿态检测的研究背景及意义，然后介绍了基于传

感器数据的人体姿态检测领域的国内外研究现状，最后介绍了本文的主要研究内容以及论文结构。

第二章人体姿态检测相关技术研究。首先介绍了基于传感器数据的人体姿态检测技术的主要流程，并介绍了使用手机传感器的优势。然后介绍了基于机器学习的人体姿态检测技术。最后介绍了基于深度学习的人体姿态检测技术。

第三章基于 Multi-channel CNN 的人体姿态检测。首先分析了经典数据集 UCI-HAR 的分类可行性和缺点，并介绍了自建数据集 MyData 的优势和建立过程。然后介绍了 1D CNN 的网络结构和超参数调整过程，以及有关神经网络性能的评价指标。最后提出了一种 Multi-channel CNN 网络，该网络使用多个参数设置不同的卷积头并行提取特征。

第四章基于 ConvLSTM 的人体姿态检测。首先证明了 RNN 在处理长时间依赖数据存在梯度爆炸和梯度消失的原因。然后介绍了 LSTM 的网络结构和超参数调整过程。然后将 CNN 和 LSTM 在提取特征上的优势结合起来，同时提取传感器数据的时空特征，提出了用卷积头作为前端的 CNN-LSTM 网络和门控单元的矩阵乘法换成卷积运算的 ConvLSTM 网络。最后对比了各网络的性能。

第五章人体姿态检测系统设计与实现。首先介绍了将 ConvLSTM 模型移植到移动端的过程。然后设计并实现了基于传感器数据的人体姿态检测系统，分为手机 APP 和后台管理系统两个部分。最后通过实验验证了移动端模型的识别准确率。

第六章总结和展望。总结全文工作，并对存在的问题，提出下一阶段研究的建议。

第二章 人体姿态检测相关技术研究

本章主要介绍基于手机传感器数据的人体姿态检测的流程以及相关的技术。本质上，人体姿态检测是一个数据分类问题，这类问题的研究主要分为训练过程和预测过程两个大部分，其具体步骤如图 2-1 所示。

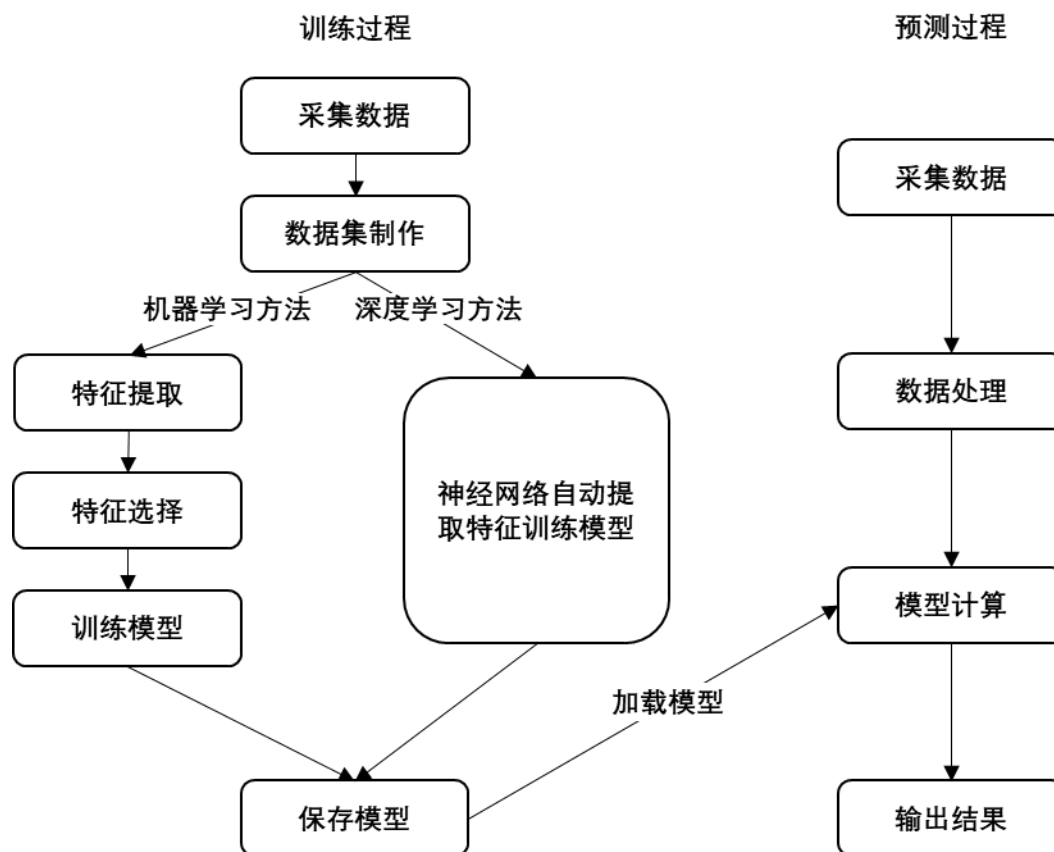


图 2-1 人体姿态检测流程

训练过程的主要步骤包括：采集传感器数据，数据集制作，训练分类模型，保存模型。数据集制作包括两步：首先要数据预处理，因为采集到的传感器数据会有大量的噪声和误差，需要进行滤波等操作，然后还需要对传感器数据打上对应的分类标签，这样就能告诉计算机每组数据分别对应什么类别。训练分类模型根据使用的方法不同，又可以分为两类：机器学习方法和深度学习方法^[17]。机器学习方法是指人工对数据集特征进行提取，并选择其中对分类结果影响比较大的特征作为训练模型的输入。深度学习方法则是通过神经网络模型自动学习，使模型自动提取数据集表现好的特征。众所周知，特征选取的好坏直接影响分类性能的好坏，大部分情况下自动提取的特征比人工提取的特征表现好。最后则需要把训练好的模型的

结构以及参数固化输出进行保存，供预测过程进行使用。预测过程则是，采集传感器数据并进行同样的预处理操作，将处理好的数据输入到已经训练好的模型中，模型会通过计算返回给我们它预测的分类结果。

2.1 手机传感器在人体姿态检测中的应用

上一章提到了，随着智能手机制作的进步，手机已经成为了集合多种传感器的软硬件平台，这些传感器可以帮助我们采集环境信息以及手机的运动信息。本文选择的 Android 开发平台，目前大部分安卓手机中都内置了以下 11 种传感器^[18]，传感器类型及调用代码如表 2-1 所示，下面对在姿态检测中应用较多的传感器进行介绍。

表 2-1 手机传感器类型

传感器类型	对应常量
加速度感应器	Sensor.TYPE_ACCELEROMETER
陀螺仪传感器	Sensor.TYPE_GYROSCOPE
方向传感器	Sensor.TYPE_ORIENTATION
磁场传感器	Sensor.TYPE_MAGNETIC_FIELD
距离传感器	Sensor.TYPE_PROXIMITY
光线传感器	Sensor.TYPE_LIGHT
气压传感器	Sensor.TYPE_PRESSURE
温度传感器	Sensor.TYPE_TEMPERATURE
线性加速度传感器	Sensor.TYPE_LINEAR_ACCELERATION
旋转矢量传感器	Sensor.TYPE_ROTATION_VECTOR
重力传感器	Sensor.TYPE_GRAVITY

加速度传感器是对手机在设备坐标系中 XYZ 三个方向上的加速度进行测量的传感器，设备坐标系是跟随手机姿态变化的坐标系，当手机水平向前放在桌上时，X 和 Y 轴平行于电话屏幕，分别指向右侧和前方，Z 轴由 X 轴和 Y 轴的叉积给出，该叉积指向屏幕正上方。与之对应的是全球坐标系，该坐标系通常用于描述智能手机的运动^[19]，它的 X 轴指向正东方向，Y 轴指向正北方向，Z 轴指向重力矢量相反方向。这两个坐标系如图 2-2 所示。三轴加速度计的主要作用是将空间加速度在 X、Y、Z 三个轴上进行分解，当手机在空间上发生运动时，相应方向产生的作用力会对传感单元产生挤压，传感单元通过将形变量转换成电压输出，从而得到该方向上的单轴加速度。

陀螺仪又称角速度传感器，它主要测量手机设备坐标系三个轴在手机运动时

产生的转动角速度。陀螺仪是基于角动量守恒定律实现的，在陀螺仪旋转的时候，根据角动量守恒定律，会产生抗拒方向改变的趋势。在手机中对角速度进行积分运算，就可以通过陀螺仪得到手机旋转角度的变化。陀螺仪在智能手机上用途非常广泛，除了用于姿态检测，可以配合摄像头拍照防抖，在各类体感游戏中也可以加强操作准确性。

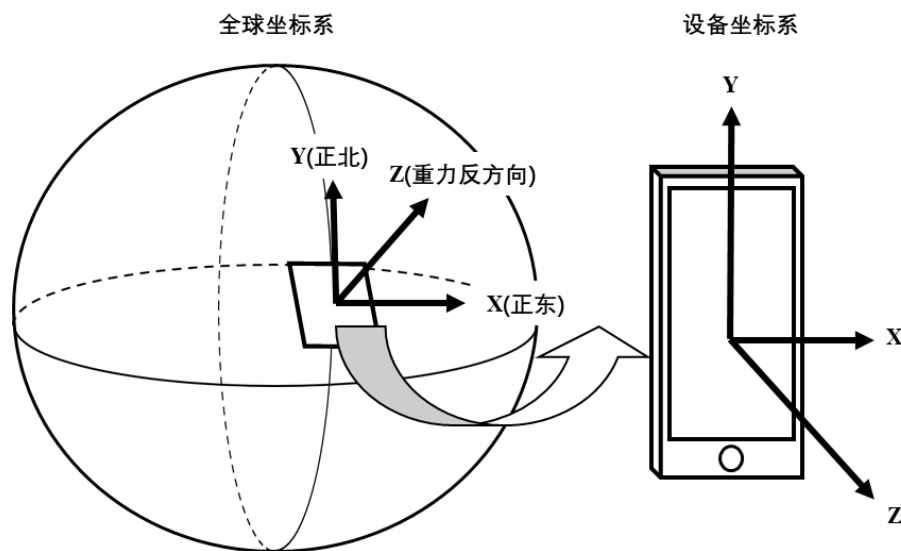


图 2-2 全球坐标系和设备坐标系

基于可穿戴传感器的人体姿态检测应用中，无论是在采集数据还是识别检测阶段，都需要实验对象在四肢以及躯干绑定相应的传感器原件。这种方法可以采集更多的运动姿态数据，理论上会使检测结果更加准确。但同时这种方法也存在很多问题，例如传感器穿脱费时、穿戴传感器导致行动不便以及产生异物感、各种传感器电量管理困难以及分设备采集数据不方便处理等问题。

使用手机传感器则可以避免上述问题，手机把所有传感器集成在一起，采集数据时通常只需要编写一个采样程序就能同时采集多个传感器的数据。并且手机电量充足，不会因为长时间使用传感器而导致电量不足，可以保证持续检测实验对象的日常活动。使用手机检测时通常将手机放在腰部位置，腰部是人体的重心所在位置，重心的活动状态很大程度上代表着整个人体的运动状态，能在保证检测准确性的同时，还不会使实验对象有异物感，也不会影响实验对象的日常活动。

2.2 基于机器学习的人体姿态检测

机器学习是融合了数学、计算机等基础学科的一个大学科领域，其中涉及概率论与数理统计、最优化分析等经典数学理论。科学家亚瑟·塞缪尔创造了这一术语，

并解释它为：使计算机在没有明确人工编程指导的情况下进行自我学习。让计算机使用算法分析已发生事件的数据，然后对未发生事件做出预测^[20]。

机器学习算法根据训练的数据是否拥有标签，可以分为两大类：监督学习和无监督学习。监督学习是对已发生事件的数据打上标签，告诉计算机这些数据对应的类别，计算机使用这些数据训练出一个模型，当未标注的新数据输入后，模型会返回一个数据类别的预测结果。无监督学习是事先无法知道数据的类别，需要计算机自己根据各个样本的相似性和差异性进行区分，使各类类内的相似性最大，类间的差异性最大^[21]。

人体姿态检测中经常用到的是监督学习方法，通过对每个时间序列活动打上对应的标签的方式，对模型进行训练。获得打好标签的数据集之后，传统的机器学习方法通常会需要人工提取活动序列的时频域特征，这个过程被称为特征工程^[22]。在特征工程中，提取的主要时域特征有均值、方差、标准差、最值等等，主要频域特征包括傅里叶变换、功率谱密度等等。获得特征后，再通过一些机器分类算法进行模型训练。人体姿态检测作为一个典型的序列分类问题，很多研究者将一些传统机器学习方法应用在其中都取得了良好的性能，下面选取了一些具有代表性的主要方法进行简单介绍。

2.2.1 支持向量机 (SVM)

支持向量机 (Support Vector Machine, SVM) 最早由 Vapnik 提出，最初用来解决二分类问题^[23]。最初提出的方法是，对于一些线性可分的二分类数据，希望找到一个超平面把这些数据分为两类，并且希望这个超平面到两类数据最近的样本数据之间距离最大。距离超平面最近的这些样本数据就称为支持向量，顾名思义，支持向量才是对寻找最优超平面影响最大的样本，它反映了支持向量机的一个重要特性：训练完成后，大部分训练样本都不需保留，最终模型仅与支持向量有关。支持向量机的训练过程，就是求解最大超平面间隔的最优化问题，能直接用现成的优化计算包求解，但通过构造拉格朗日函数，求解对偶问题的方法更加高效。

后来经过不断研究，对于线性不可分的数据，支持向量机加入核函数之后，可以通过将低维空间的原始特征映射到高维空间进行分类，并且分类性能还有所提高。对于实际应用中像人体姿态检测这样的多分类问题，使用支持向量机的解决方法目前有两条，一是将多分类问题分解成几个二分类问题，训练出多个支持向量机之后再将其超平面组合在一起实现多分类；二是将多个超平面的参数最优化过程合并为一个最优化问题，通过解决该问题实现多分类^[24]。

2.2.2 朴素贝叶斯 (NB)

朴素贝叶斯 (Naive Bayes, NB) 是贝叶斯分类算法中最常见的一种^[25], 这类方法的理论基础就是贝叶斯定理如(2-1)所示, 其中 A 、 B 为两个事件, $P(B|A)$ 是已知事件 A 发生, 事件 B 发生的概率。获得后验概率 $P(B|A)$, 在现实任务中通常难以直接获得。所要实现的是基于有限的训练样本集尽可能准确地估计出后验概率 $P(B|A)$, 将待求的量转化为其它可求的值, 从而解决分类问题。

$$P(B|A) = \frac{P(AB)}{P(A)} = \frac{P(A|B)P(B)}{P(A)} \quad (2-1)$$

朴素贝叶斯算法基于贝叶斯定理进行简化改进, 假设给定的各个类别之间在概率上是相互独立的, 将每个特征对分类结果的影响看作一致的。这种参数化的方法相对简单。但预测结果的准确性严重依赖于数据的概率分布是否和其真实分布一致。

虽然朴素贝叶斯方法的逻辑非常简单, 易于实现, 在分类过程中的时空开销小。理论上比其它分类方法更小的误差率, 但也存在一些缺点。它假定给定的各个类别之间在概率上是相互独立的, 但有些数据的特征之间往往有一些关联性, 和这个假设相矛盾, 导致分类效果不好。

2.2.3 决策树 (DT)

决策树 (Decision Tree, DT) 是一类常见的机器学习方法, 顾名思义, 决策树是基于树型结构来进行分类的^[26]。一般的, 一棵决策树包含一个根节点、任意中间节点和任意叶子节点。叶子节点代表分类结果, 所以有多少个类别就有多少个叶子节点。其它节点代表一个属性测试, 每个节点包含的样本集合会根据属性测试结果划分到对应的子节点。每条从包含所有样本的根节点出发到叶子节点的路径都对应了一个判定序列。当决策树碰到以下三种情况时, 会停止判别: 一是当前节点样本为同一类无需划分; 二是当前节点所有属性值都相同无法划分, 此时分类结果为当前节点包含最多样本的类别; 三是当前节点没有样本, 此时分类结果为父节点包含最多样本的类别。

在决策树学习的过程中, 为了使分类效果更加准确, 需要不断地重复地划分节点, 导致决策树的分支过多, 会使决策树把训练数据才有一些特征当成所有数据的共性而产生“过拟合”现象, 因此需要通过剪枝处理来降低过拟合的风险^[27]。剪枝包括后剪枝和预剪枝。后剪枝是在决策树第一遍学习完之后, 从中间节点开始自底向上判断, 如果将子树替换成叶子节点, 分类性能提升, 则子树替换为叶子节点; 预剪枝是在决策树生成的过程中, 判断当前节点在划分后能不能给整棵树的性能

带来提升，如果不能则停止生长。

决策树的树型结构直观易理解，符合人认知事物的过程，并且应用范围广，分类回归均可以。缺点就是容易造成过拟合。一般都需要采取剪枝或者限制树的高度等方式防止产生过拟合。

2.3 基于深度学习的人体姿态检测

相比于机器学习，深度学习是一个于 2006 年才被提出的新概念。它本来被看作是机器学习的一个分支领域，其本身的学习目的也是通过有标签数据或无标签数据来训练深度神经网络。深度学习最初被提出的目的是用来解决机器学习中需要人工提取特征、选择特征的缺点^[28]，同时随着研究者对其的深入，相继提出了一些特有的学习方法，因此越来越多的研究者将其视为一个单独的研究领域。

神经网络比起普通的机器学习方法来说，计算量大得多，受到计算能力的限制，它的研究工作发展缓慢。直到 GPU 被发明出来，深度学习的研究开始迅速发展，许多公司都提出了开源框架，如 Google 的 tensorflow、Facebook 的 PyTorch 和百度的 PaddlePaddle 等。

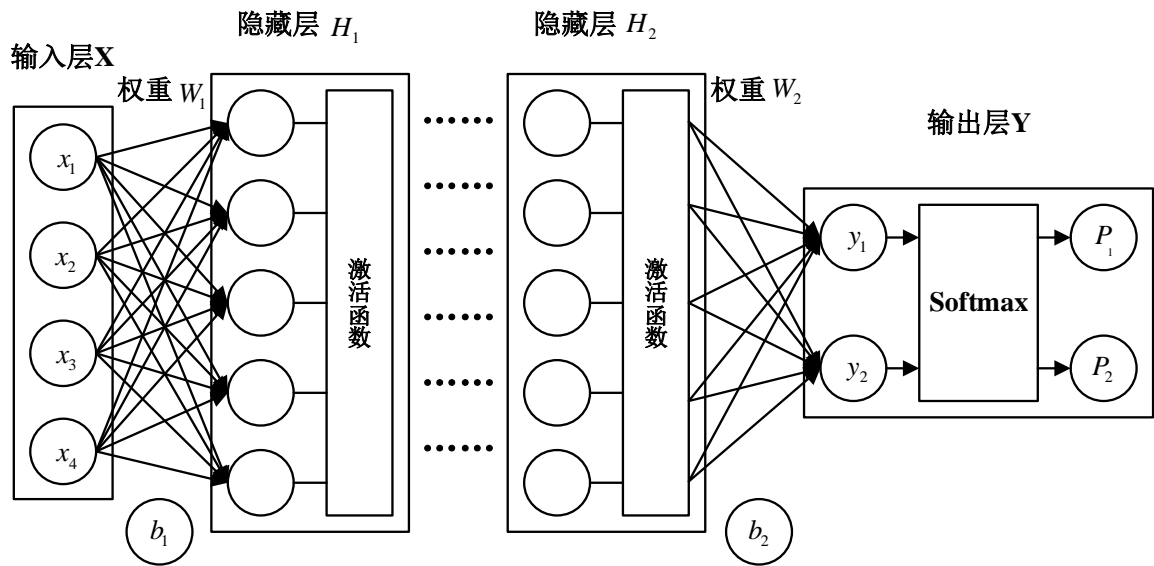


图 2-3 神经网络结构

图 2-3 所示为一个全连接神经网络结构，其中每一个圆圈被称为神经元，从中可以发现神经网络有如下特点^[29]。神经网络的结构为层次型的结构，输入数据从左向右传输进行前向计算，最左边一层为输入层，负责输入数据；最右边一层为输出层，负责输出神经网络的计算结果，图中最后输出的概率值为神经网络对各类的预测概率，概率最大的类即为预测结果；中间的所有层称为隐藏层，它们对外是不

可见的。同一层的神经元之间没有关系，每一层神经元的输入是根据上一层的输出矩阵乘以权重矩阵再加上偏置向量获得，如式(2-2)所示。其中， H_1 为隐藏层神经元输出， W_1 为输入层到隐藏层的权重矩阵， b 为偏置项。

$$H_1 = W_1 \cdot X + b \quad (2-2)$$

如果对矩阵运算有一定了解，可知级联的矩阵运算可以简化为一个线性方程来表示。也就是说如果神经网络只包含简单的线性运算，那么中间这些隐藏层的设计就是冗余的，因为无论多少层矩阵运算都可以简化成一层隐藏层，这样的话神经网络的提出就失去了意义。从图 2-3 可以看到，除了每一层之间的矩阵运算，每一层神经元还会经过激活函数运算再输出。激活函数的作用就是为每一层的线性运算添加非线性，通过这种方式神经网络可以模拟任何非线性函数^[30]。这样对于某些线性不可分的数据，神经网络也能正确将其分类。

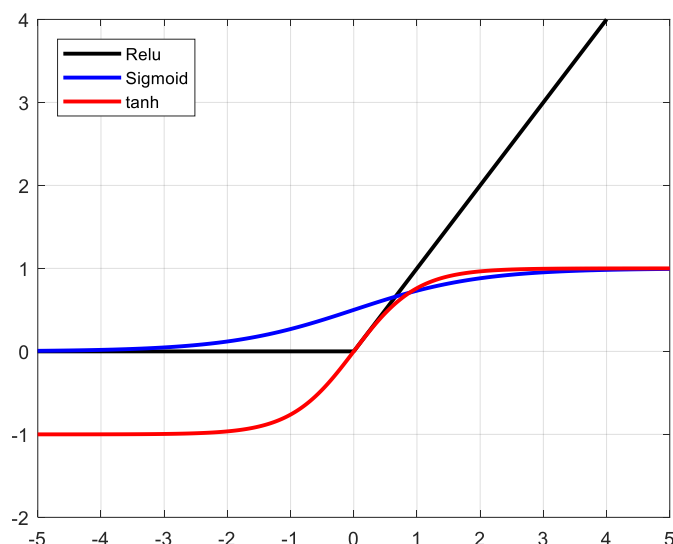


图 2-4 激活函数

常用的激活函数有三种，分别是 Sigmoid 函数、tanh 函数和 Relu 函数，它们的图像见图 2-4，公式见(2-3)到(2-5)。Sigmoid 函数可以压缩数据在 0-1 区间还便于求导，但它的输出不是零均值的，并且容易出现梯度消失现象。tanh 函数将数据压缩在-1~1 区间内，所以它解决了非零均值的问题，但同样存在梯度消失问题。Relu 函数是当前较为常用的激活函数，它计算简单，并且梯度不会饱和，解决了梯度消失问题。

$$Sigmoid: f(x) = \frac{1}{1+e^{-x}} \quad (2-3)$$

$$\tanh : f(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (2-4)$$

$$\text{Relu} : f(x) = \max(0, x) \quad (2-5)$$

当经过所有隐藏层时，得到的输出向量 Y 在一定程度上已经可以反映神经网络的预测概率。比如图 2-3 中输出 Y 可以是 $(3, 0.1)$ 这样的向量，明显可以判断分类结果为第一类，但希望的输出应当是所有情况和为 1 的概率向量。所以还需要经过 Softmax 函数的计算，公式见(2-6)，将输出结果正规化处理为 $(95\%, 5\%)$ 这样的概率向量 P 。

$$P_i = \frac{e^i}{\sum_j e^j} \quad (2-6)$$

其中， p_i 表示第 i 类结果的预测概率， e^i 表示对 Y_i 做指数运算。以上是一个深度神经网络的基本结构，基本上每一种神经网络都会使用到以上的计算单元，不同的则是神经元之间的连接方式，以及各种隐藏层的选择。目前在人体姿态检测领域用得比较多的神经网络模型主要有：卷积神经网络和循环神经网络以及它的变体长短时记忆网络^[31]。

2.3.1 卷积神经网络 (CNN)

卷积神经网络是一种专门为了处理图像数据而开发的神经网络，自从被提出后，它在图像处理，语音识别、自然语音处理等领域大放异彩。人体姿态检测作为一个时间序列分类任务，和语音识别任务有着许多共同之处，所以近年来有越来越多的研究者将卷积神经网络应用在人体姿态检测当中，提高了人体姿态检测的分类准确率。

前面提到，传统的人体姿态检测方法都应用特征工程的方法，需要研究者有足够的信号处理知识，能够对传感器数据人工提取特征。如果研究者对数据研究比较深入，对训练数据集的特征足够高了解，才能提取并选择到比较有用的特征。否则，用于训练的特征可能对结果没有影响或者存在冗余，从而影响训练速率以及识别精度。而卷积神经网络能够自动提取传感器数据的特征，并且性能优于使用特征工程训练的传统机器学习模型^[32]。

卷积神经网络主要包括卷积层、池化层、展平层、全连接层。卷积层是卷积神经网络的核心，其内部包含多个卷积核，它的主要作用就是提取数据的特征。对于时间序列数据来说，每个数据和它周围时刻的数据联系是比较紧密的，和较远时刻的数据关联则更小。针对这一特点，卷积层内的神经元被设计为只与上一层中位置

关系接近的神经元相连，并且一个卷积核共享同一个权值。经过这些改进，整个网络就减少了很多的参数。输入序列通过与卷积核的数学运算，可以提取出序列的某些指定特征（Feature Map），不同卷积核，提取的特征也是不一样的。池化层的作用是下采样，通过去掉 Feature Map 中不重要的样本，进一步减少参数。目前最常用的池化层函数是 MaxPooling 函数，它的作用就是取 n 个样本中的最大值作为样本值。展平层的作用则是将池化层输出的特征展平为一维的结构，以便和全连接层连接，通过全连接层进行预测分类。

与应用在图像处理领域的卷积神经网络不同，人体姿态检测的卷积层采用一维卷积提取特征。人体活动数据的组成格式为一个大小为 $M \times N$ 的矩阵，每次输入模型的数据帧数为窗口大小 N ，每一帧则包含维度为 M 的特征向量。一维卷积是卷积核只在数据帧方向从左往右移动，由于要把所有特征覆盖，所以卷积核的行数也必须是 M ，所以描述卷积核时只需要知道列数（kernel size）以及个数（filters）^[33]。以 kernel size 等于 3，输入序列时间长度为 5，特征维度为 3，激活函数为 Relu 为例，一维卷积过程以及最终结果如图 2-5 所示。

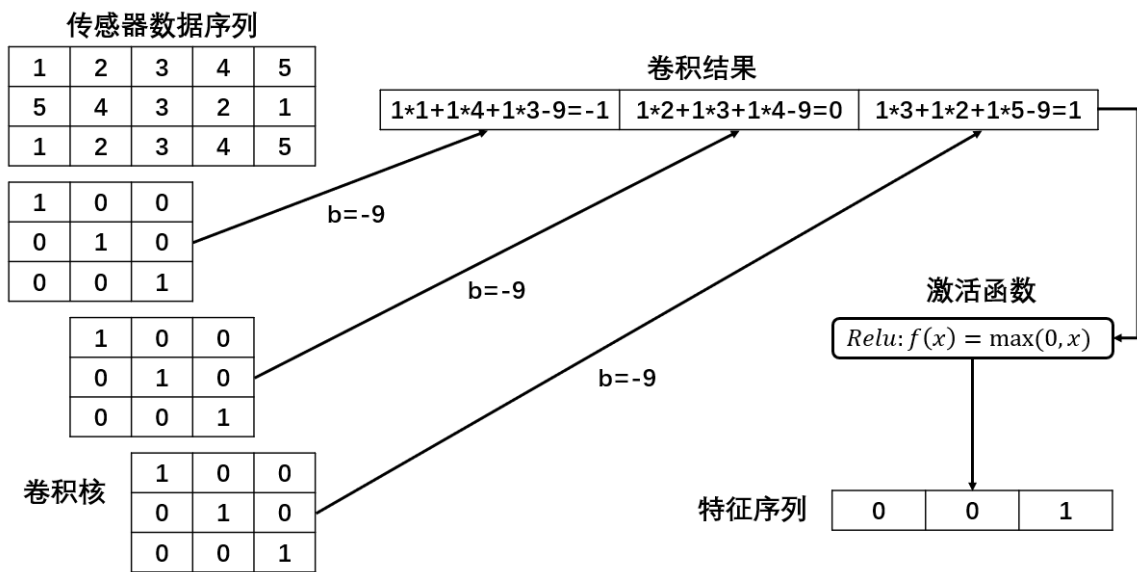


图 2-5 一维卷积过程

通过图 2-5 可以非常直观地感受卷积神经网络，在卷积层的整个计算过程。其中最复杂的部分是一维卷积，在没有边界扩展（no padding），移动步长（stride）为 1 时，一维卷积的计算公式如式(2-7)所示。

$$a_i = f\left(\sum_{m=0}^M \sum_{n=0}^K w_{m,n} \cdot x_{m,i+n} + b\right) \quad (2-7)$$

其中， a_i 为输出结果的第 i 列元素， $f()$ 表示激活函数， K 为卷积核的 kernel

size-1, $w_{m,n}$ 表示卷积核第 m 行第 n 列权重, $x_{m,i+n}$ 表示传感器数据序列第 $i+n$ 时刻的第 m 维特征, b 为偏置项。

2.3.2 循环神经网络 (RNN)

循环神经网络最早被提出是在自然语言处理领域, 某些任务需要更好的发现数据序列中前后输入之间的关系, 循环神经网络就能很好的解决这一问题。在人体姿态检测中, 一个人体动作的完成不是瞬时发生的, 而是一段连续的过程, 有些动作比如行走、站立更是持续整个时间窗口的持续活动。分析动作序列时, 不能只单独的去分析每一帧, 而应该分析整个序列。所以将循环神经网络应用在人体姿态检测领域, 具有非常大的研究价值^[34]。

循环神经网络的基本结构包括: 输入层、隐藏层、循环层和输出层。图 2-6 是一个简单的循环神经网络示意图, 右边是它按时间线展开后的数据处理流程。

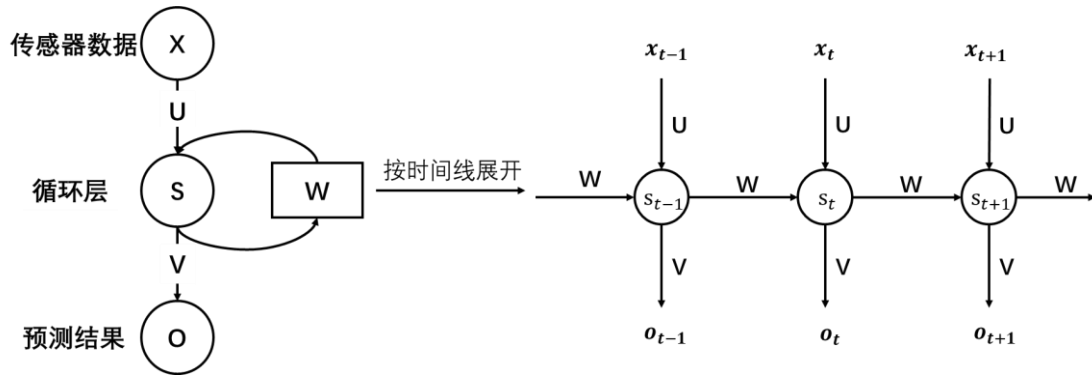


图 2-6 循环神经网络

可以看到, 循环神经网络的基本结构和一个普通的全连接网络相比, 在循环层多了一个权重矩阵 W , 将循环层展开看, 可以发现循环层当前时刻 t 的值 s_t 不仅受到当前时刻输入 x_t 的影响, 还受到上一时刻值 s_{t-1} 的影响, 上一时刻的 s_{t-1} 会跟权重矩阵 W 相乘再反馈到当前时刻。循环神经网络的数据计算流程, 可以用式(2-8)和式(2-9)来表示:

$$s_t = f(U \cdot x_t + W \cdot s_{t-1}) \quad (2-8)$$

$$\begin{aligned} o_t &= g(V \cdot s_t) \\ &= Vf(U \cdot x_t + W \cdot s_{t-1}) \\ &= Vf(U \cdot x_t + W \cdot f(U \cdot x_{t-1} + W \cdot s_{t-2})) \end{aligned} \quad (2-9)$$

其中, o_t 是输出层 t 时刻的输出, V 是上一层到输出层的权重矩阵, f 和 g 是激活函数, U 是输入层到下一层的权重矩阵。从式(2-9)可以看出, 循环神经网络 t 时刻的输出 o_t 是受前面每个时刻输入 x_t 、 x_{t-1} 、 x_{t-2} 、 \dots 影响的, 这就是循环神经

网络的核心思想。

但循环神经网络无法处理窗口较长的时间序列，主要是因为循环神经网络很容易发生梯度消失和梯度爆炸，这样导致梯度在较长序列中训练时无法一直传递下去，从而导致网络训练效果不好^[35]。梯度爆炸相对来说处理起来容易一些，可以通过设置梯度阈值来解决这一问题。当发生梯度爆炸时，直接将梯度设置为阈值继续向下传递。而梯度消失问题则更能解决，可以通过使用 Relu 激活函数或者合理初始化权重解决。当然也可以对网络结构进行改进，以避免梯度消失。

2.3.3 长短时记忆网络 (LSTM)

长短时记忆网络的提出，是为了解决循环神经网络存在的梯度问题。因为在长序列处理上有着比循环神经网络更优异的表现，长短时记忆网络在语言识别，自然语言处理以及人体姿态检测等领域中都被广泛应用。相对的，它的网络结构也更加复杂，需要花费一定的精力才能理解和使用。

通过分析循环神经网络无法产生长期依赖的原因，可以发现主要原因是循环神经网络只有一个对短期输入敏感的状态 h 。长短时记忆网络的解决方法就是增加一个保存长期依赖关系的单元状态 c ，并实现对它的控制。

从图中可以看出，实现长期记忆的思路就是使用三个门控单元。第一个门是遗忘门，负责控制继续保存单元状态 c ，决定当前时刻的单元状态 c_t 保留多少上一时刻的单元状态 c_{t-1} 。遗忘门的计算公式见式(2-10)， W_f 是遗忘门的权重矩阵， $[h_{t-1}, x_t]$ 是 h_{t-1} 和 x_t 拼接后的向量， b_f 是遗忘门偏置项， σ 是 Sigmoid 函数：

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2-10)$$

第二个门是输入门，负责控制把即时状态输入到单元状态 c ，决定保存多少当前时刻的输入 x_t 到单元状态 c_t ，输入门的计算公式见式(2-11)， W_i 是输入门的权重矩阵， b_i 是输入门偏置项：

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2-11)$$

第三个门是输出门，负责控制是否把单元状态 c 作为当前长短时记忆网络的输出，决定当前时刻单元状态 c_t 有多少输出到当前时刻输出 h_t 。输出门的计算公式见式(2-12)， W_o 是输出门的权重矩阵， b_o 是输出门偏置项：

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2-12)$$

通过这些门，可以控制单元状态 c 的变化。 c'_t 是用于描述当前输入的单元状态，它是根据上一次的输出和本次输入来计算的，计算公式见式(2-13)：

$$c'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2-13)$$

c_t 是当前时刻的单元状态，它是由网络关于当前的记忆 c'_t 和长期的记忆 c_{t-1} 加权相加得到，计算公式见式(2-14)， \circ 表示矩阵对应元素相乘，又称为 Hadamard 乘积，Hadamard 乘积是矩阵的一类运算，对形状相同的矩阵进行运算，并产生相同维度的第三个矩阵。遗忘门和输入门在这里就像两个阀门一样，控制了 c'_t 和 c_{t-1} 对当前状态的影响。

$$c_t = f_t \circ c_{t-1} + i_t \circ c'_t \quad (2-14)$$

长短时记忆网络当前时刻的最终输出 h_t 则是由输出门和单元状态共同决定的，计算公式见式(2-15)。

$$h_t = o_t \circ \tanh(c_t) \quad (2-15)$$

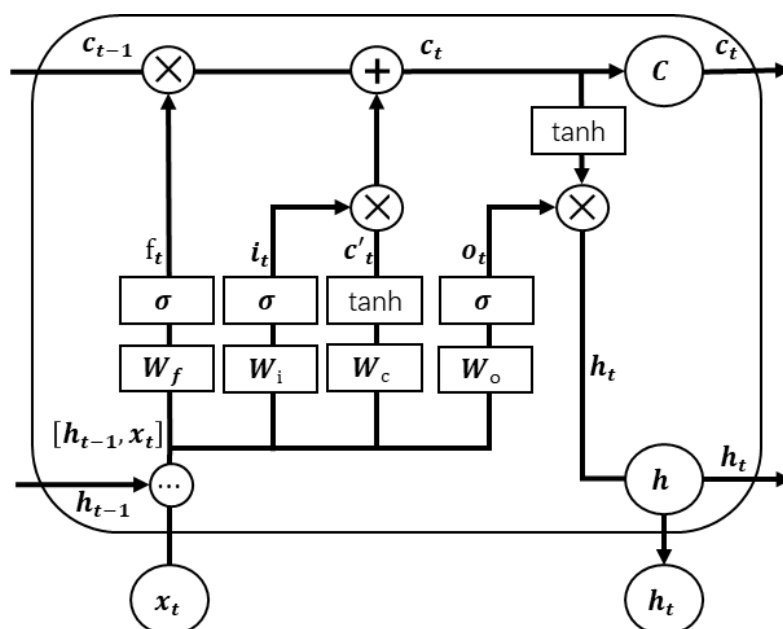


图 2-7 长短时记忆网络结构图

图 2-7 表示长短时记忆网络的整个计算过程。通过图 2-7 以及公式(2-10)到(2-15)，清晰地描述了长短时记忆网络的整个前向计算过程^[36]。因为 Sigmoid 函数的取值范围是 0-1，当门的值为 0 时，则被控制向量将无法通过；当门的值为 1 时，被控制向量将直接通过；当门的值为 0-1 之间，被控制向量将部分通过。这样，通过门的取值就能达到控制向量输出的目的。

一般的，一个网络的计算过程越复杂，它的训练难度也就越大。长短时记忆网络的计算过程相比于之前几个网络复杂得多，可想而知，它的训练时间也要大大增加。好在 GPU 的发明问世，为计算机提供了强大的算力支持，使得长短时记忆网

络的训练问题得以解决。在大部分情况下，因为深度神经网络可以自动提取数据特征并发现连续数据之间的关联性，它的表现会比传统的机器学习方法要好，所以本文决定采取深度学习方法来原因解决人体姿态检测问题。

2.4 本章小结

本章主要介绍了基于手机传感器的人体姿态检测主要流程以及相关理论，首先介绍了手机传感器在姿态检测中的应用，并对比可穿戴传感器说明了手机传感器在该领域的优势；其次介绍了在姿态检测中使用机器学习方法的主要步骤，并简单介绍了支持向量机、朴素贝叶斯、决策树等在姿态检测领域使用较多的传统机器学习方法；最后介绍了深度学习，通过全连接神经网络说明神经网络的基本结构，并详细介绍了在姿态检测中应用最多的几种神经网络：卷积神经网络、循环神经网络以及长短时记忆网络。

第三章 基于 Multi-channel CNN 的人体姿态检测

本章主要内容为数据集构建流程以及一维卷积神经网络和多通道卷积神经网络的设计与实现。首先介绍了经典数据集 UCI-HAR，通过对该数据集的分析，采集并制作了自己的数据集 MyData。然后搭建了用于人体姿态检测的一维卷积神经网络，并对该网络的超参数配置进行了一系列的实验优化。最后设计了一种多通道卷积神经网络，进一步提高了人体姿态检测的识别准确率。

3.1 数据集构建

上章介绍了机器学习使用预定义的数据集来训练系统，便于其在遇到新数据时能做出预测。计算机训练效果的好坏不只取决于所建立神经网络模型结构的好坏，很大程度上还受到训练所用数据集的影响。如果数据集中的数据本身就不可分，那么再好的模型也无法将它们进行分类。并且对于不同的任务目标，我们需要制作不同的数据集。所以，如何制作一个好的数据集是机器学习领域具有挑战性的部分。

3.1.1 经典数据集分析

每个研究领域都会有一些前面研究者公开的经典数据集，这些数据集可能不会完全符合你研究的任务要求。但也有非常大的研究价值，通过分析这些数据集，可以学习其它研究者对同类型数据的处理方法，有很多好的方法值得借鉴，这样可以帮助你制作一个完全符合你要求的优良数据集。同时，在公开数据集上验证你的算法性能也是非常有必要的，这样可以横向对比，以此评估你的算法优劣性，帮助你进行更深入的研究。

在人体姿态检测领域，许多研究者都公开了他们的数据集，其中有些数据集实验丰富，在一般的模型训练效果也很好，被业内公认为经典数据集，如 UCI-HAR、WISDM、MotionSense 等。本文主要对 UCI-HAR 数据集进行分析，并通过该数据集对比验证算法性能。UCI-HAR 是意大利热那亚大学提出的数据集，该数据集采集的是人体活动时的手机传感器数据^[14]。实验对象由 30 名年龄在 19-48 岁之间的志愿者组成。每人腰部佩戴智能手机（三星 Galaxy S II），进行六项活动（步行、上楼、下楼、坐、站、躺），表 3-1 为活动分布情况。利用手机内置的加速度计和陀螺仪，在 50Hz 的恒定速率下采集三轴总加速度（total acc）和三轴角速度（body gyro）。实验采用录像的方式对数据进行人工标注。获得原始数据后，实验人员对原始数据使用噪声滤波器过滤传感器噪声；然后使用巴特沃斯低通滤波器，从包含

重力加速度的总加速度数据中去除重力分量，从而获得人体加速度数据 (body acc)；最后使用滑动窗口将数据分割为 50%重叠的 128 个数据 (2.56 秒) 的窗口。所获得的数据集被随机分成两组，其中 70% 的实验对象被选择生成训练数据，30% 的实验对象被选择生成测试数据。

表 3-1 UCI-HAR 数据集活动类别分布表

数量 \ 百分比	步行	上楼	下楼	坐下	站立	躺下	总计
训练集	1226/16.7	1073/14.6	986/13.4	1286/17.5	1374/18.7	1407/19.1	7352/100
测试集	496/16.8	471/16.0	420/14.3	491/16.7	532/18.1	537/18.2	2947/100
总计	1722/16.7	1544/15.0	1406/13.7	1777/17.3	1906/18.5	1944/18.9	10299/100

可以看到每个类别的数量有波动，但分布还是相当均匀。假设参与者必须向上和向下走相同数量的楼梯，并且知道智能手机具有恒定的采样率，那么上楼和下楼时应该有相同数量的数据点。不考虑数据有缺陷的可能性，参与者向下走的速度快了大约 10%。

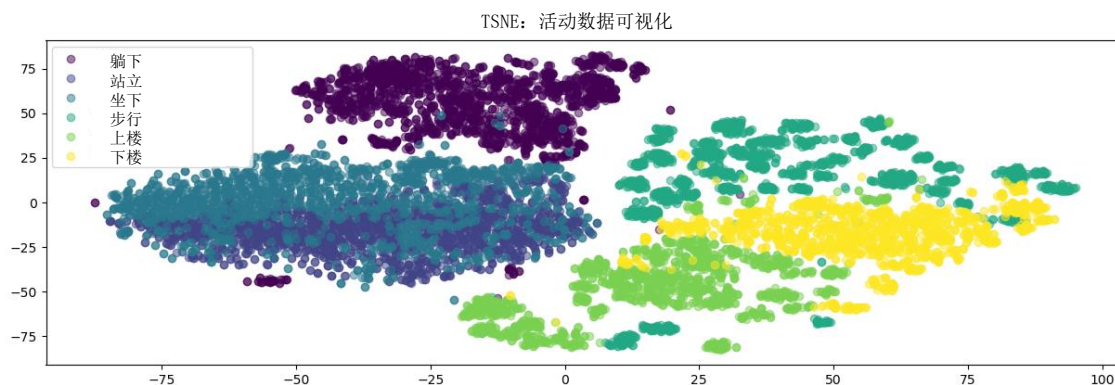


图 3-1 TSNE 活动数据可视化

图 3-1 是活动数据的 TSNE 数据降维可视化图。TSNE 是一种数据可视化的工具，能够将高维数据降到 2-3 维，然后画成图^[37]。TSNE 是目前来说效果最好的数据降维与可视化方法，当我们想要对高维数据进行分类，又不清楚这个数据集有没有很好的可分性，可以通过 TSNE 投影到 2 维或者 3 维的空间中观察一下，是否同类之间距离小，异类之间距离大。如果在低维空间中具有可分性，则数据是可分的；如果在低维空间中不具有可分性，可能是数据不可分，也可能仅仅是因为不能投影到低维空间。从图 3-1 可以看出，在二维空间中，UCI-HAR 数据集采集的六类活动只有站立和坐下两个活动的部分数据重合，其余活动数据分布基本分开，所

以该数据集具有很好的可分性。

现在选取 7 号志愿者的所有训练数据，来分析三类传感器数据变化是否受到人体活动的影响。由于数据集由 50%重叠的滑动窗口截取，所以不能之间将所有数据窗口拼接，需要先将重叠数据去除，切片截取每个窗口的后半部分数据，正好避开重叠部分（上一行的后半部分和下一行的前半部分是重叠的），再将所有数据拼接变成一个完整的活动时间序列。图 3-2 为 7 号志愿者的完整活动时间序列以及三类传感器 X 轴数据时间序列曲线图，其中 Class1-6 分别对应步行、上楼、下楼、坐下、站立、躺下。从图中可以看出，活动 1、2、3（动态活动）对应的传感器数据变化较大，同时三类活动的传感器数据波形、频率也有所不同；活动 4、5、6（静态活动）对应的传感器数据则较为平缓，基本没有变化，三个活动之间的区别主要是数据的取值区间不同。

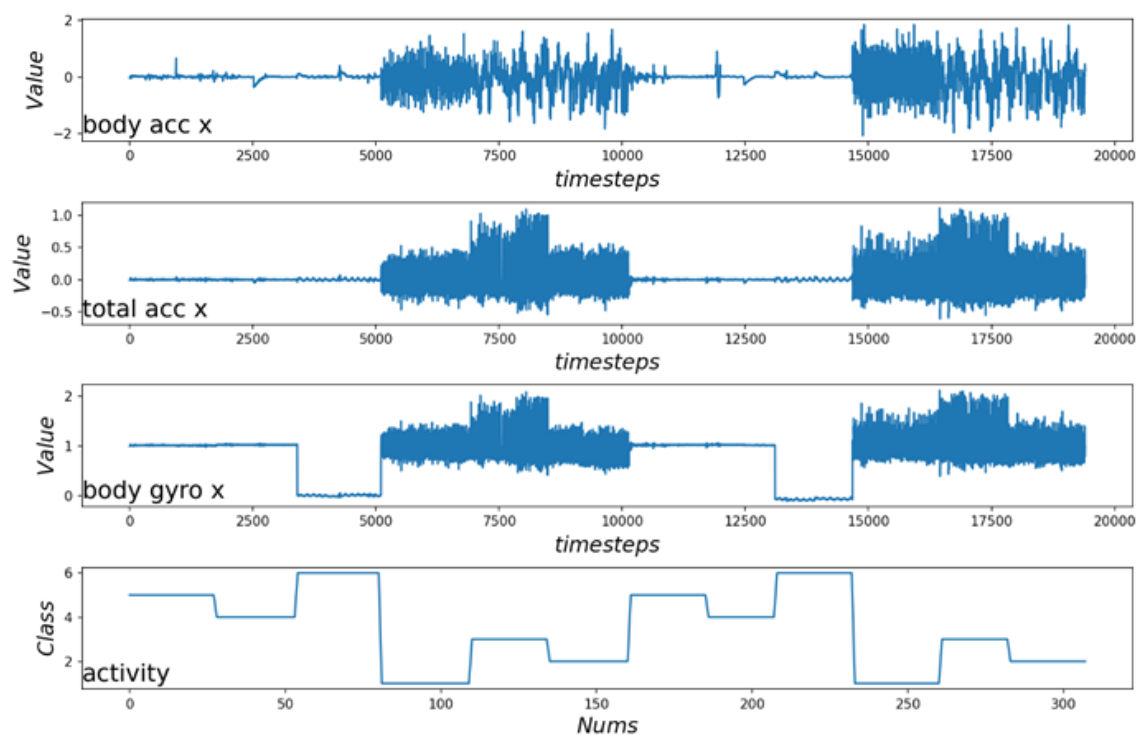


图 3-2 志愿者传感器数据与活动变化曲线

通过以上对 UCI-HAR 数据集的分析，可以发现总加速度数据、人体加速度数据以及角速度数据会随着人体活动类别变化，活动类别不同传感器数据的时频域特征也有明显不同，各类活动的传感器数据也具有很好的可分性。实验数据表明使用传感器数据能够完成人体姿态检测的任务。

3.1.2 数据采集

虽然 UCI-HAR 数据集可以提供非常完整的经过预处理的数据集，以便于帮助分析问题、数据建模，但该数据集也有一些缺点。实验过程中，所有实验者均将采集手机横放绑定在腰间，但有些不同类的人体姿态如坐下和站立，在腰部以上的身体动作是一致的，这样会导致部分采集的数据重合，影响后续分类。并且日常生活中大部分手机使用者很少会将手机放在腰部位置，通常都将手机竖直放置在裤子口袋中。同时由于该数据集提出时间较早，受到手机设备限制无法直接采集人体加速度数据，人体加速度数据需要采集到总加速度数据后，通过低通滤波器人工分离。而现在的手机都嵌入了线性加速度计，可以直接采集人体加速度数据，简化数据处理过程，缩短手机检测时间。所以 UCI-HAR 数据集只能用来评估模型性能，与其他研究者提出的算法进行横向对比，不利于直接将该数据集训练的模型在实际应用中使用。在本文中，我建立了一个数据集命名为 MyData，最后在移动端实际应用的模型为此数据集上训练的模型。

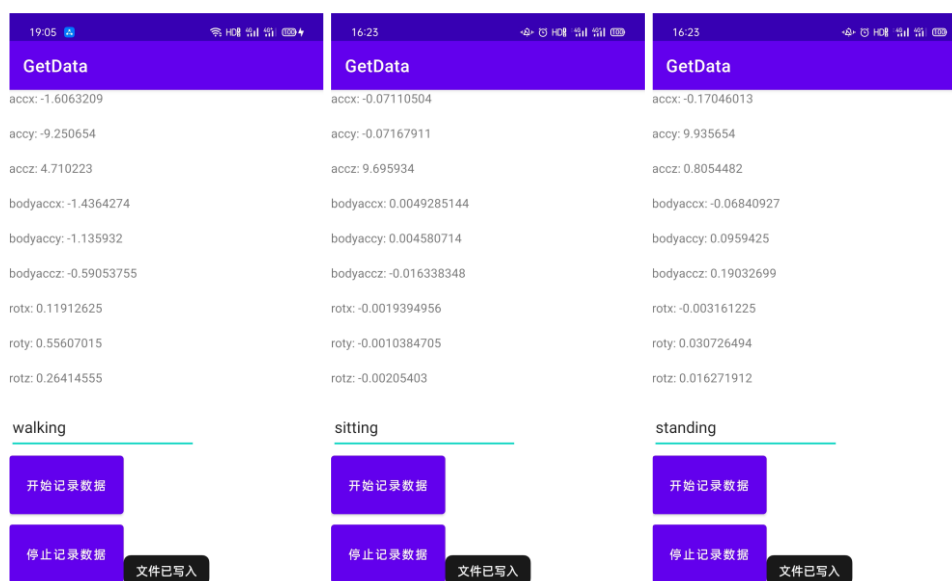


图 3-3 采样 APP

本文使用 Android Studio 开发工具开发采样 APP，开发版本为 Android11.0(R)，开发的采样 APP 命名为 GetData。通常人体活动所产生的信号有效频率成分出现在 25Hz 以下，大于 25Hz 的频率成分基本上为噪声数据，根据奈奎斯特采样定理，将采样频率设置为 50Hz。图 3-3 为该 APP 的 UI 界面，完整的采集流程为：输入保存文件的名名称（如 walking）；点击开始记录数据按钮，手机会开始显示实时三轴加速度计、三轴线性加速度传感器以及三轴陀螺仪数据，并将数据记录在列表中；然后实验者将手机放入裤子左右口袋中，持续进行同一种日常活动如步行；最后点

击停止记录数据按钮，会将列表中的所有数据写入 txt 文件并将该文件保存到手机内存。本数据集邀请教研室的 10 名同学作为志愿者，年龄分布为 23-25 岁，实验时使用两部手机分别放在裤子左右口袋中进行预设活动。由于日常活动中，躺下姿态通常是睡觉时的姿态，此时手机一般与用户分离，所以 MyData 数据集不采集躺下姿态，而是多采集一种运动姿态。最终采集的姿态有六类：步行、站立、跑步、坐下、上楼和下楼。一次采集过程获得的传感器数据如图 3-4 所示，每一行为一个时刻采集的数据，总共有 10 列元素，前六列分别为三轴合加速度和三轴人体加速度数据，单位为 g（重力加速度 $9.8m/s^2$ ），第七到九列为三轴角速度数据，单位为 rad/s ，第 10 列为对应活动标签。

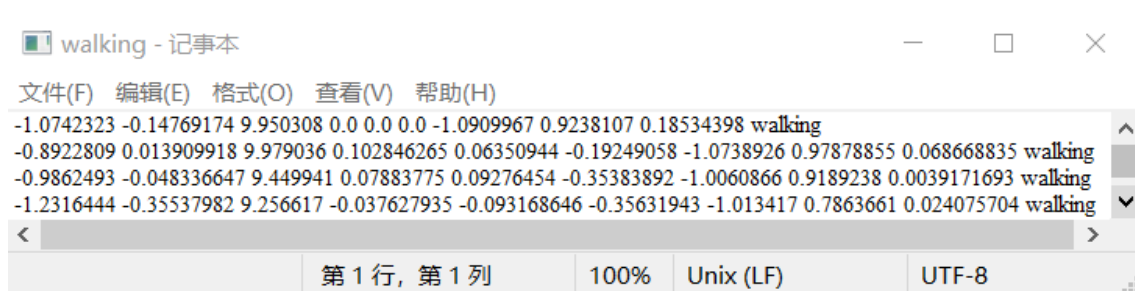


图 3-4 传感器数据展示

3.1.3 数据预处理

通常情况下，采集到的原始数据会受到传感器误差以及随机噪声的影响，并且采集到的数据为一段长时间的数据，不能直接用于神经网络训练模型的输入，需要经过滤波器过滤、数据分割等预处理操作才能获得用于训练模型的数据集。

对采集到的原始数据进行滤波处理，能起到去除传感器噪声、消除异常值等作用。常用的滤波器有均值滤波器、中值滤波器、低通滤波器以及卡尔曼滤波器等，主要分为时域滤波和频域滤波两类，使用者需要根据自己的任务目标选择合适的滤波器，这里选择的是中值滤波器。中值滤波器是一种平滑滤波，可以消除孤立噪声，使数据曲线更加平滑，同时还保持数据原有的波形^[38]。中值滤波的基本实现原理是通过一个固定长度的滑动窗口划过整个数据序列，将窗口中的数据进行排序找出中值，再将窗口正中间的样本替换成中值。中值滤波的窗口大小 n 选择尤为重要，当 n 过小时，会导致滤波效果不明显，不能很好的滤除异常值；当 n 过大时，会导致滤波后的波形失真，不能反映原波形的时频域特征，并且当 n 越大，滤波后数据的时间延迟越大。图 3-5 为中值滤波器窗口大小 $n=3,5,7$ 时的滤波效果对比，通过对比发现，当 $n=3$ 时，能在较好的滤除异常采样值的同时保持数据原有的波形特征，故本文选用窗口大小为 3 的中值滤波器对原始数据进行滤波。

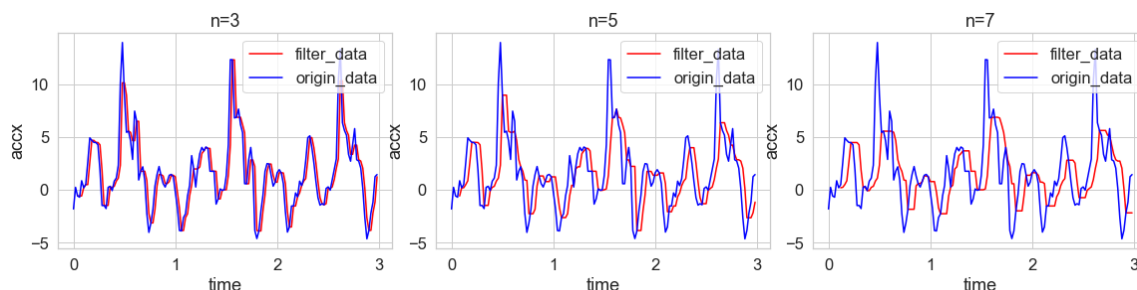


图 3-5 中值滤波效果对比图

经过中值滤波处理之后，数据仍是很多段长短不一的时间序列。人体姿态检测问题本质上是时间序列分类问题，这类问题中为神经网络训练准备输入数据时，需要保证数据的维度尺度统一，所以还需要对数据进行分割。由于实验中志愿者将手机设置好开始检测后，需要完成将手机放入裤子口袋中的动作，并且我们需要的是志愿者正在进行某一重复性活动的数据。考虑到志愿者正式进入状态的时间，我将所有传感器数据序列开始的 5 秒以及结束前的 5 秒视为异常数据，将这些数据切割后丢弃，继续处理剩下的数据。

接下来需要将剩下的数据分段，采用 50%重叠滑动窗口的方法将数据分为一系列长度一致的数据段。与前面中值滤波使用的滑动窗口类似，如果窗口过小就无法包含足够的人体活动信息来给计算机训练识别；如果窗口过大，可能会在一个窗口内包含多种活动的信息，会出现某些活动检测不到或者窗口内主要活动判别错误的情况。普通人走路的步频在 $[90, 130]$ 步/分钟以内，即至少 1.5 步/秒。考虑到节奏较慢的人如老年人和残疾人，以下限的 50%为最低速度，即 0.75 步/秒。每个窗口样品上至少有一个完整的步行周期（两步），所以将窗口大小设置为 2.56 秒比较合适。采样 APP 设置的采样频率为 50Hz，即一个窗口内包含 128 个样本点。完成一次切割后，窗口会向后滑动。50%重叠滑动窗口指上一个窗口的后半部分与下一个窗口的前半部分重合，即滑动步长为 64 个样本点。相比于无重叠窗口，这样的好处在于，不会出现窗口和窗口之间将一个完整活动切成两半无法检测的情况，因为每个活动的数据一定会在某个窗口保持完整。

最后将每类活动的数据随机划分，其中 80%的数据定为训练集，用于训练模型输入；另外 20%的数据定为测试集，用于检测训练完的模型的性能。以上为 MyData 数据集完整的采集以及预处理流程。

3.2 一维卷积神经网络（1D CNN）

随着深度学习研究的发展，应用级的神经网络模型搭建训练都可以借助开源

的应用程序编程接口（Application Programming Interface, API）来实现，许多公司都提出了简单易上手的深度学习框架，如 Google 的 TensorFlow、Keras，Facebook 的 Pytorch 等。其中 TensorFlow 提出较早，使用人数较多，拥有庞大的开源文档社区供开发者讨论交流。Google 公司还基于 TensorFlow 提出了 TensorFlow Lite 框架，可帮助开发者在移动设备、嵌入式设备和 IoT 设备上运行模型，以便实现设备端机器学习。Keras 是由 Python 编写并仅支持 Python 开发的高层神经网络 API，并被添加到 TensorFlow 中，成为了 TensorFlow 官方的高级 API。Keras 的封装性非常好，使用时不需要过多关注底层实现，便于使用。因此，本文的所有实验均基于 TensorFlow 平台实现，使用 Keras 库训练神经网络模型，并使用 Python 包含的科学计算库如 pandas、numpy 等等完成数据分析。

3.2.1 1D CNN 模型搭建

卷积神经网络最初是针对图像处理任务提出的，模型的输入是二维矩阵的数据形式，采用的卷积方法是二维卷积，并不完全适合构建时间序列分类任务的模型。对于多传感器数据序列，采用一维卷积神经网络（1D CNN）能更好的提取特征，与经过特征工程处理的传统机器学习方法相比，能达到相似甚至更好的性能。使用深度学习库 Keras 搭建 1D CNN 模型，训练模型时要求的输入格式为[样本数，时间步长，特征数]，与前面的数据处理步骤一致，样本数等于时间窗口数，时间步长为 128，特征数为 9（每个时刻的九轴传感器数据）。模型的输出则是一个一维向量，每个元素表示每一类活动的预测概率。

图 3-6 为 1D CNN 网络结构示意图，模型采用 Keras 经典的顺序化结构（Sequential Model）。首先是两个一维卷积（Conv1D）层，主要作用是使模型学习输入数据中的特征，单层卷积可能会使学习效果不好，所以一般采用两层卷积层。卷积层的学习速度非常快，容易产生过拟合。过拟合原因包括特征维度过多，参数过多，训练集样本较少，噪声过多等等，导致训练的模型对训练集预测结果好，但对测试集中的数据预测结果差，这样训练出来的模型泛化能力较差。所以，卷积层后面需要连接一层遗忘（Dropout）层，遗忘层可以降低过拟合风险，提高模型的泛化能力。遗忘层会在训练时以一定的概率 p （通常是 50%）关闭遗忘层神经元的输出，也就是输出为 0，这样可以使神经元之间减小依赖性，因为随机的关闭可以使得神经元之间不会总是同时作用。然后是一维池化（MaxPooling1D）层，池化层的主要作用是降采样，这里采用的是最大池化即选取区域内的最大值的神经元作为该区域的概括。通过降采样可以压缩特征去除冗余信息，从而大大减少神经元的数量，简化神经网络的规模。池化层之后连接一个展平（Flatten）层，顾名思

义，展平层会将学习到的多维特征展平为一个一维长向量，常用在从卷积层到全连接层的过渡。之后是全连接层，在学习的特征和输出之间提供缓冲区，由一个特征空间线性变换到另一个特征空间。因此，全连接层的目的是将前面提取的特征，在全连接层经过非线性变换，提取这些特征之间的关联，最后映射到输出空间上。全连接层在 Keras 中用 Dense 层表示，Dense 层通过参数设置可以作为输入层、中间层和输出层使用。所以输出层也用 Dense 层表示，用于输出预测结果。

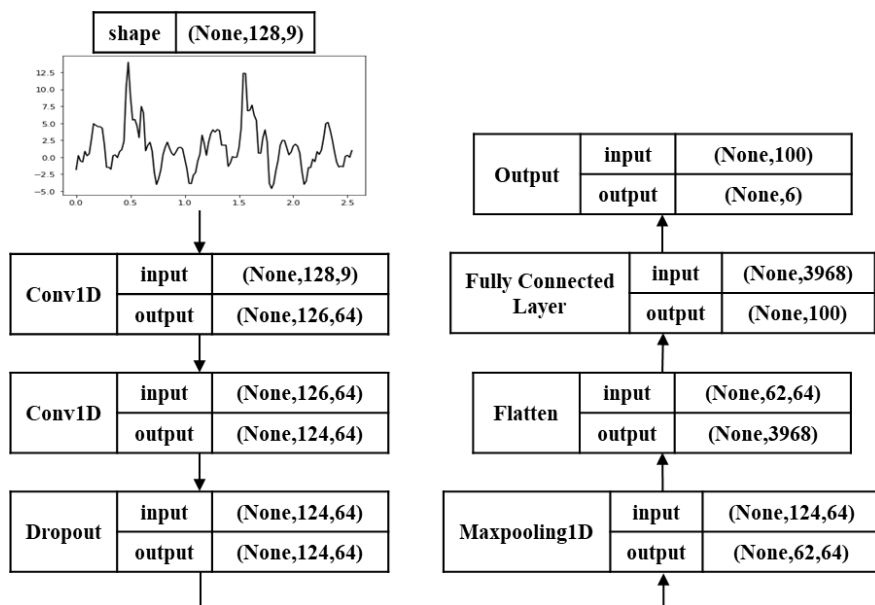


图 3-6 一维卷积神经网络结构

输入输出数据尺度根据 UCI-HAR 数据集设置，并显示了各层的输入输出维度。其中主要参数设置如下：两个一维卷积层的参数设置相同，每个卷积层中，卷积核的数量都为 64，卷积核的大小为 3，激活函数为 Relu；遗忘层的遗忘率为 0.5；一维池化层采用 Maxpooling 方法，池大小为 2；全连接层的激活函数也为 Relu；输出层的激活函数为 Softmax。由以上参数设置，也可推出如图 3-6 所示的神经网络各层的输入输出维度。

在上述参数配置的情况下，分别使用 UCI-HAR 和 MyData 数据集来训练 1D CNN 模型。由于设计的网络中含有遗忘层，不同批次的实验中遗忘层会以 0.5 的概率随机关闭神经元，具有一定的随机性，所以网络的准确率具有一定的浮动，每次实验的结果不会完全一致。实验过程中，需要在每个数据集上重复进行多次训练，并统计准确率的平均值以及标准差，来更加准确的评估网络性能。该网络在 UCI-HAR 数据集上进行 10 次重复实验后，在测试集上取得的平均准确率为 90.696%，标准差为 0.378%；在 MyData 数据集上进行 10 次重复实验后，在测试集上取得的平均准确率为 96.768%，标准差为 0.294%。虽然以上初步构建模型均在两个数据

集上取得良好识别效果,但网络的超参数配置均根据经验设置,可以通过一系列实验设置更好的超参数,来进一步提高网络性能。

3.2.2 超参数调整

在神经网络训练的过程中,网络模型的识别性能很大程度上受到一系列超参数的影响,如网络中神经元的数量、遗忘层的遗忘概率、卷积层的数量、卷积核的数量以及卷积核的大小等。这些超参数的配置根据经验设置并不完全准确,需要经过大量实验才能找到最适合本识别任务的配置,使模型识别准确率达到最佳。对于 1D CNN 网络来说,最重要的是卷积层提取数据特征的效果,本文对卷积层的相关超参数进行实验,进一步提升模型的识别性能。

首先需要确定训练时的 epoch 大小。当一个完整的数据集在神经网络中进行了一次正向传播和一次反向传播之后将所有训练集样本都训练了一边的过程称为“一代训练”(epoch)。在这样的一代训练中,神经网络里所有的权重矩阵的值会通过梯度下降的迭代过程得到更新,使网络的预测误差更小,准确率更高。一次这样的更新过程是远远不够的,需要经过多次这样的过程使网络达到最优。当 epoch 增加,网络一直在重复训练更新权重,性能也会从欠拟合变成过拟合。epoch 的设置没有一个标准,对于不同的数据集和不同的神经网络,它的影响都是不一样的。所以需要找到一个合适的 epoch 大小,使网络既不欠拟合也不过拟合。

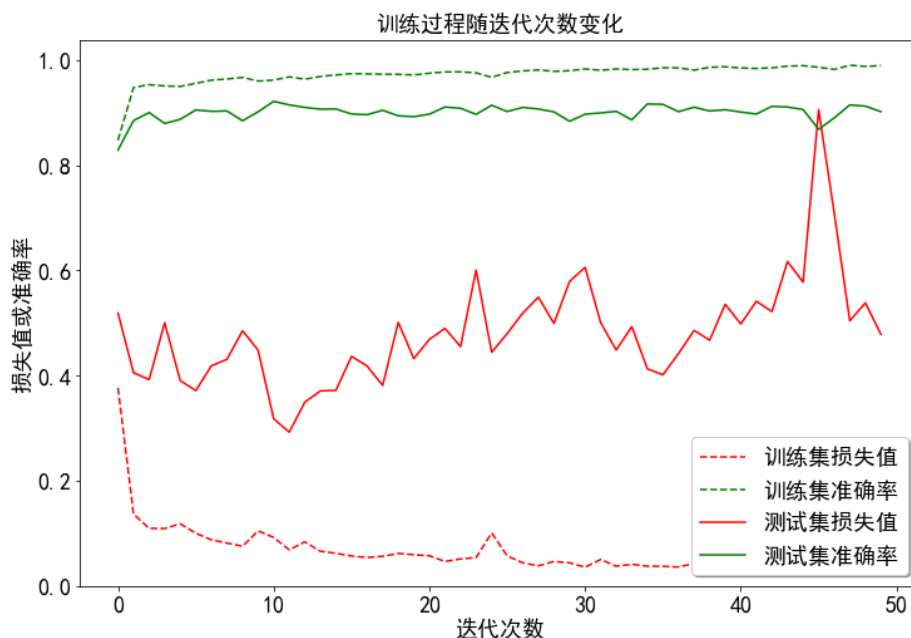


图 3-7 使用 UCI-HAR 数据集 1D CNN 的 loss 和 accuracy 随 epoch 变化曲线

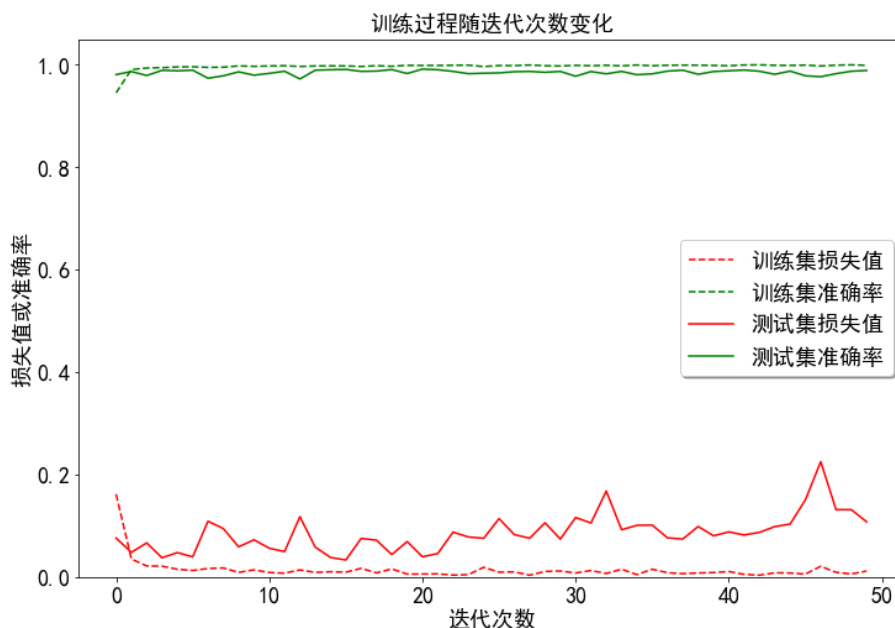


图 3-8 使用 MyData 数据集 1D CNN 的 loss 和 accuracy 随 epoch 变化曲线

图 3-7 和图 3-8 为在两个数据集上 1D CNN 模型训练准确率 (accuracy) 和损失函数值 (loss) 随 epoch 变化的曲线。损失函数是神经网络每次迭代更新参数的依据, 每次前向计算过程得到输出的预测值后, 用损失函数计算误差项 (预测值和真实值之间的差值), 在反向传播的过程中, 根据损失函数的导数, 使用梯度下降法回传误差, 更新之前得到的所有权重矩阵的值, 直到损失函数越来越小。从图 3-7 和图 3-8 看到, 两个数据集随着 epoch 的增加, train loss 一直下降, train accuracy 一直上升, 说明随着迭代次数增加, 网络对训练集样本的预测越来越准确。而 test loss 先是一直下降, 当 epoch 增加到 10 左右开始上升, 说明网络性能产生过拟合现象, 即在训练集上表现优异, 在测试集上则表现一般。所以, 1D CNN 在两个数据集上的 epoch 大小都设置为 10。

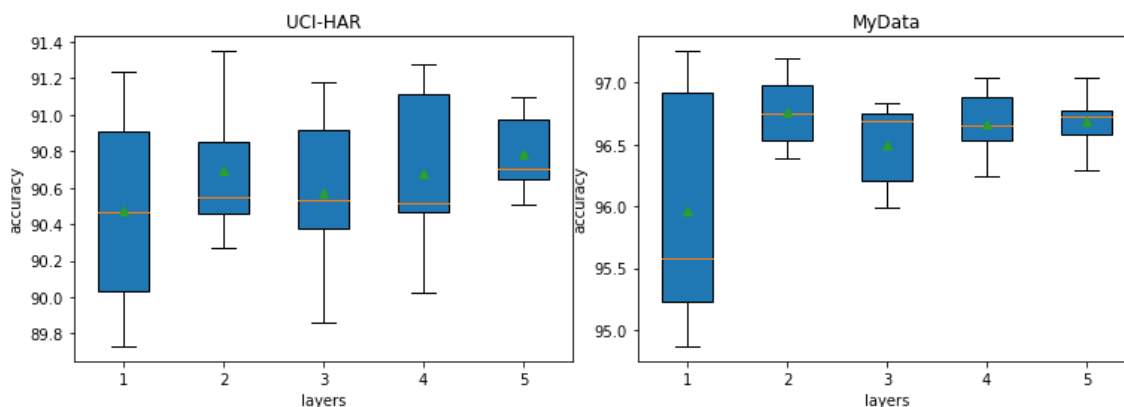


图 3-9 卷积层数对识别率影响箱型图

然后对 1D CNN 网络的卷积层数量进行配置, 已找到最适合的实验结果。一般来说, 卷积层层数越多, 整个网络的误差也就越小, 但是会使整个网络复杂化, 增加网络的训练时间, 也会导致出现过拟合的情况。图 3-9 为 1D CNN 网络卷积层数变化与网络识别准确率变化箱型图, 重复进行 10 次实验, 绿色三角形为网络准确率的平均值, 上下边缘分别表示最大值和最小值, 蓝色矩形上下边缘分布表示上下四分位数, 橙色横向表示中位数, 通过箱型图可以非常直观地看出数据的分布情况。可以看到当只有一层卷积层时, 不仅平均准确率较低, 同时数据波动也较大, 随机性比较大。当卷积层数增加时, 网络识别准确率在增加, 但当层数超过两次层之后, 识别准确率会开始上下小范围波动。当层数到达 5 层时, 性能回升并且数据波动范围减小, 但计算量过于庞大。综合比较来看, $n=2$ 的网络在准确率、稳定性以及复杂度等方面都有较好的表现, 所以定卷积层数为 2。

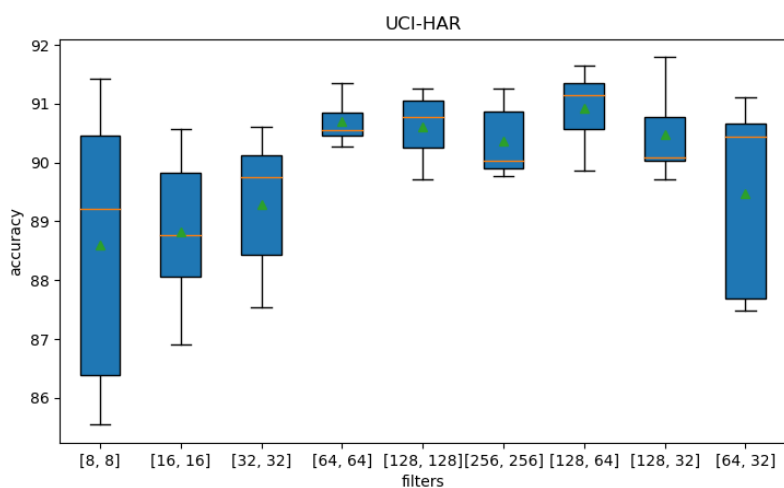


图 3-10 UCI-HAR 数据集卷积核数量影响箱型图

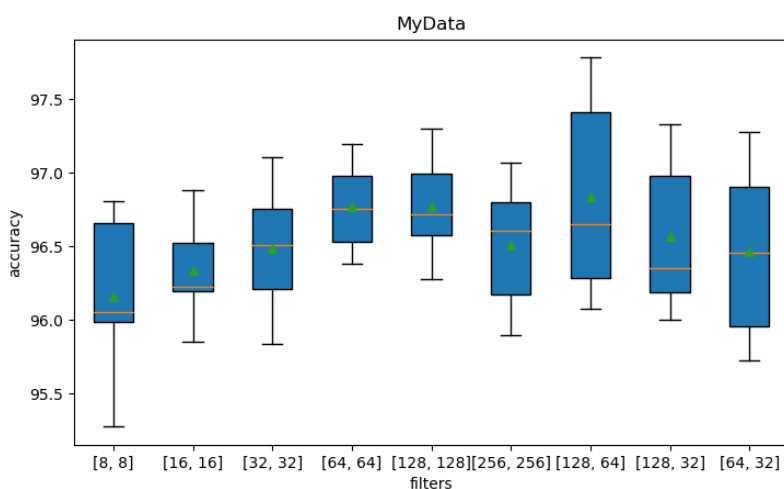


图 3-11 MyData 数据集卷积核数量影响箱型图

接下来分析卷积层卷积核数量（filters）对网络性能的影响，一般的卷积核越多，得到的特征图就越多，网络的计算能力就越强。如果卷积核过少，网络提取的特征不足，会导致网络的拟合能力较差造成欠拟合；如果卷积核过多，会导致参数量大大增加，对于小数据量的数据集容易造成过拟合，降低网络的泛化能力。图 3-10 和图 3-11 为在两个数据集上，1D CNN 网络卷积核数量变化与网络识别准确率变化箱型图，每组实验都重复进行 10 次。首先对两层卷积层都使用相同数量的卷积核，将卷积核数量设置为 8, 16, 32, 64, 128, 256。可以看到，在 UCI-HAR 数据集上随着卷积核数量的提升，网络的识别准确率也明显得到提升，当数量为 64 和 128 时都取得较好表现，当数量为 256 时，准确率开始下降，这是因为出现了过拟合现象；在 MyData 数据集上，随着卷积核数量增加，也有相同的变化趋势，但准确率波动范围不明显。再在第一组实验的基础上，对两个卷积层添加不同数量的卷积核，增加[128, 64]、[128, 32]、[64, 32]三组对照实验。发现在第一层使用 128 个卷积核，在第二层使用 64 个卷积核的网络表现，比两层都使用 128 或 64 个相同数量卷积核的网络更好。分析原因，主要是因为第一层提取特征的作用最大，使用更多数量的卷积核能保证充分提取特征，而第二层则不需要那么多数量的卷积核对特征图再次提取特征。所以，卷积核数量选用[128, 64]的配置。

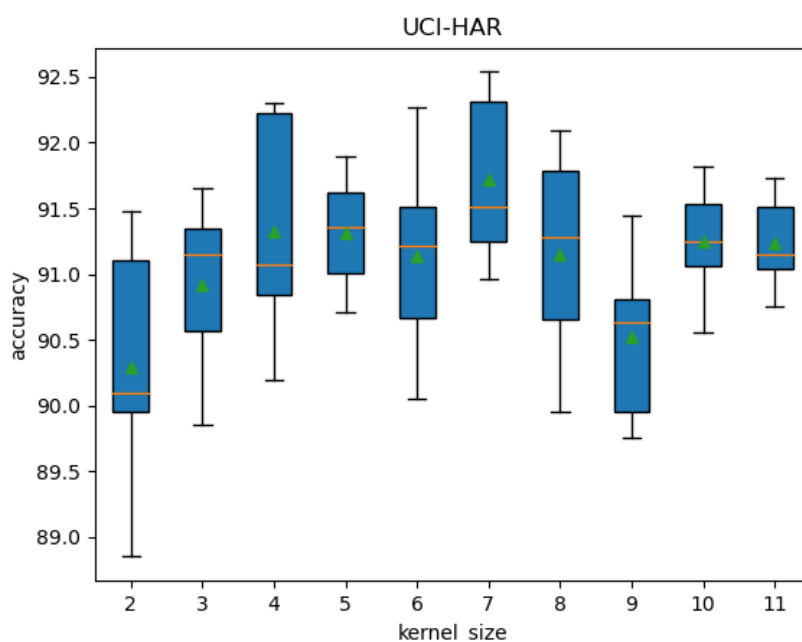


图 3-12 UCI-HAR 数据集 kernel_size 影响箱型图

最后考虑卷积核的大小（kernel size）改变对模型的影响，卷积核大小的作用与滑动窗口类似，控制每次读取输入数据进行卷积操作时的时间步长。根据数据的表示内容不同，它们的时间相关性也不同，不同的数据对于卷积步长的敏感度也不同，

kernel size 越大表明对数据的时间特征要求越宽松。将两个数据集上每层的 kernel size 依次设置为 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 每组重复进行 10 次实验, 得到如图 3-12 和图 3-13 所示的网络识别准确率箱型图。发现随着 kernel size 增加, 网络识别性能也在提升, 当增加到 7 时, 平均准确率达到最大, 之后开始起伏波动。所以将 kernel size 大小设置为 7。

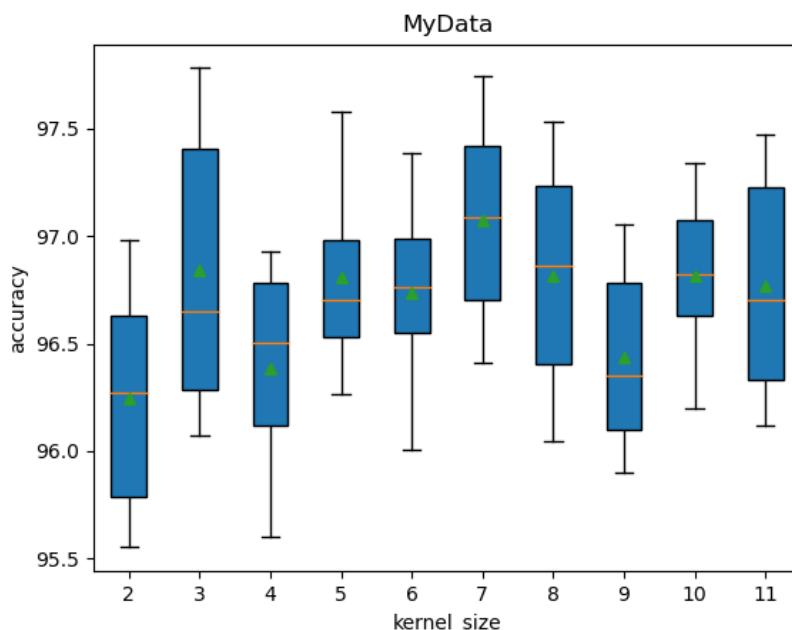


图 3-13 MyData 数据集 kernel_size 影响箱型图

经过上述一系列实验后, 发现网络的超参数配置对网络识别准确率的影响很大。对于一个新构建的神经网络, 需要经过不断寻找才能获得最适合它的超参数配置。本文最终获得的 1D CNN, 共包含两层卷积层, 第一层卷积核的数量为 128, 第二层卷积核的数量为 64, 每层卷积核 kernel size 都为 7。在两个数据集上最终得到的识别准确率分别为 91.716% (+/-0.610) 和 97.074% (+/-0.480), 比最初的 90.696% (+/-0.378) 和 96.768% (+/-0.294) 更高。

3.2.3 评价指标

通过以上实验, 修改参数配置后获得了一个性能更加优秀的卷积神经网络, 之前只是从平均准确率以及标准差等方面来评价网络的性能, 还需要用更多数学工具直观的了解它的性能好坏。在神经网络学习过程中, 需要对测试集上的分类结果进行精度评估, 用到一个重要概念——混淆矩阵。混淆矩阵是用来验证模型预测性能的矩阵。对于 k 元分类, 其实它就是一个 $k \times k$ 的表格, 用来记录分类器的预测结果。最简单的二分类混淆矩阵如表 3-2 所示。

表 3-2 二分类混淆矩阵

	预测值=1	预测值=0
真实值=1	TP	FN
真实值=0	FP	TN

模型的分类结果用 F (False)、T (True)、P (Positive)、N (Negative) 表示结果是否正确，模型的预测结果有以下四种：

- (1) TP (True Positive), 真阳性, 真实值为正例预测值为正例的正确样本数。
- (2) TN (True Negative), 真阴性, 真实值为反例预测值为反例的正确样本数。
- (3) FP (False Positive), 假阳性, 真实值为反例预测值为正例的错误样本数。
- (4) FN (False Negative), 假阴性, 真实值为正例预测值为反例的错误样本数。

上一节使用的准确率 (Accuracy) 是衡量神经网络性能最重要的指标, 它就是根据混淆矩阵的相关指标得到的, 它表示所有分类样本中被正确分类所占的比率, 具体的计算公式见(3-1)。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3-1)$$

对于多分类任务, 假阳性为真实值是其它类预测值为本类的错误样本数, 假阴性为真实值是本类预测值是其它类的错误样本数。虽然准确率能评估模型性能对数据总体的表现好坏, 但是准确率是一项总体性的指标, 对于各类样本没有偏向, 对于各类样本分布不均匀的数据集或者在某一类样本预测结果偏差较大的数据集, 无法发现并进行评估, 需要混淆矩阵这样体现每一类数据预测性能的指标来进行评价。

图 3-14 所示为用 UCI-HAR 数据集训练得到的 1D CNN 网络的混淆矩阵, 可以更加清楚地看出网络对于每一类活动的识别效果。网络对平躺姿态的检测率为 100%, 步行、上下楼这三类姿态检测准确率较高, 但对于站立和坐下两类姿态检测准确率较低, 网络经常会将站立误判成坐下、坐下误判成站立, 数据总体的检测准确率就主要受到这个问题的影响, 如果解决这个问题, 可以使模型性能大大提升。但这个问题是数据集本身的问题, 之前在图 3-1 中对传感器数据降维可视化分析时有提到过, 因为采集方式导致站立和坐下的传感器数据有大量重合。主要原因在于人在站立和坐下时, 腰部以上部分的身体姿态变化不大, 而 UCI-HAR 数据集采集过程中志愿者将手机绑在腰间, 这样的采集方式会导致采集到站立和坐下的数据

部分重合。这个问题无法通过改进网络结构来解决，只能通过改变采集数据的方式或者更新数据集来解决。

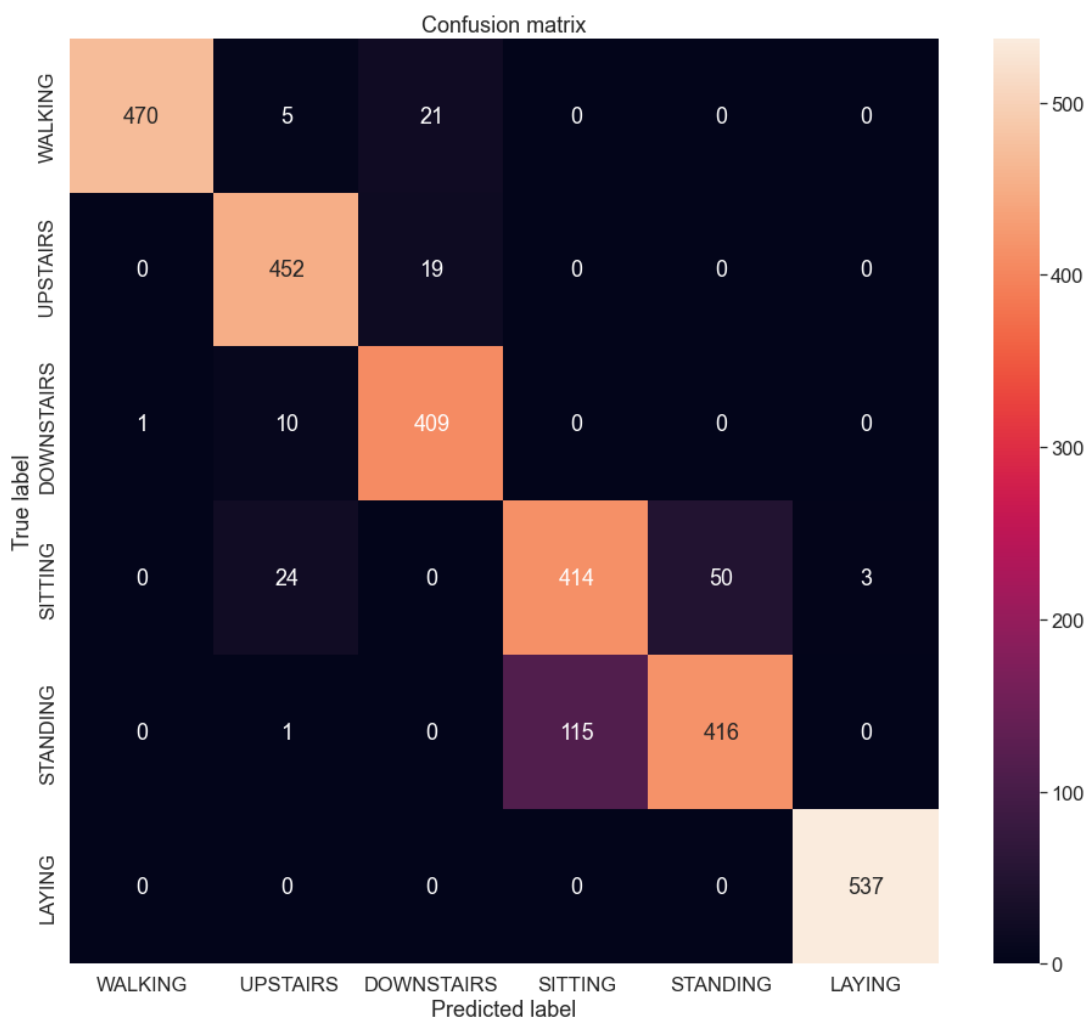


图 3-14 1D CNN 在 UCI-HAR 上的混淆矩阵

图 3-15 所示为用 MyData 数据集训练得到的 1D CNN 网络的混淆矩阵。可以看到步行、站立、跑步以及上下楼五种姿态的检测效果较好，没有出现像 UCI-HAR 数据集一样坐下和站立大量混淆的情况，但坐下姿态则有一大部分被检测成步行，可能是因为志愿者身高和椅子的匹配程度不同，所以坐下时大腿的角度也不同，导致将坐下误判为步行。同样的模型在两个数据集上表现不同，是因为 MyData 数据集采集设备的位置在裤子左右口袋处，让大腿代表整个身体的姿态，而不是腰部。这样的好处在于，可以完全分开坐下和站立的数据，使模型对这两类姿态的检测准确率大大提高，这也是同一个模型在 MyData 数据集上表现更好的原因。

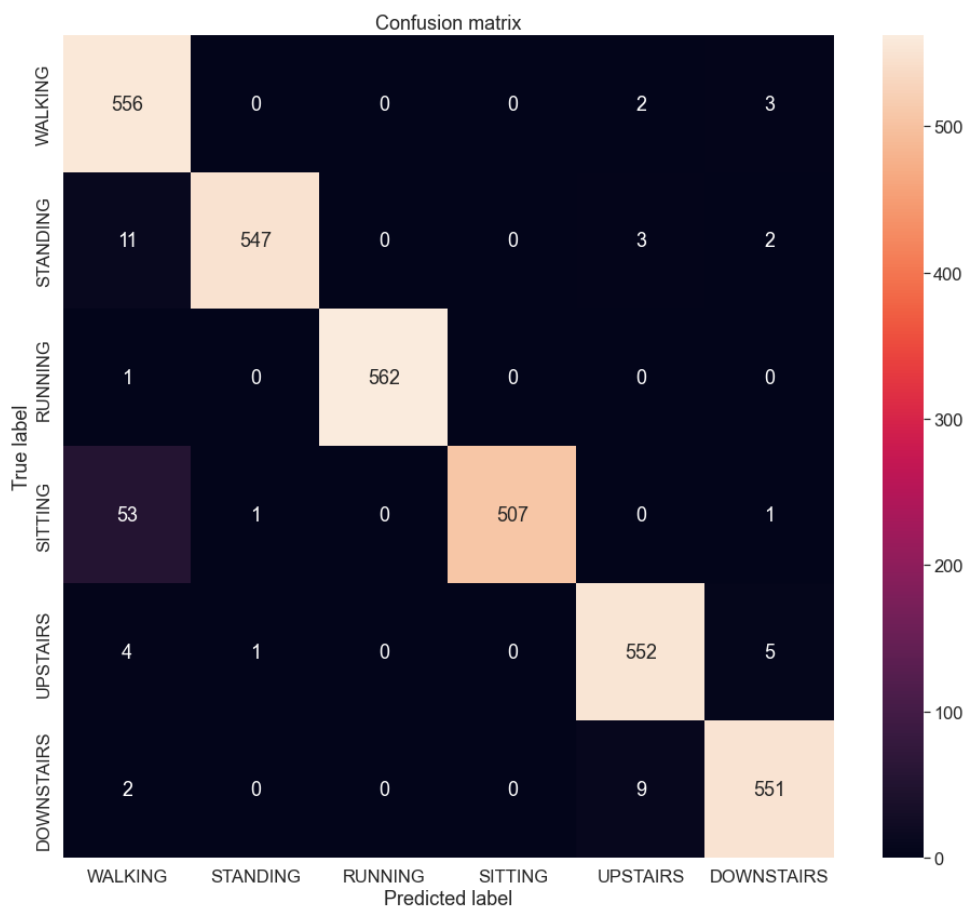


图 3-15 1D CNN 在 MyData 上的混淆矩阵

对于深度学习模型的研究来说，更看重模型设计思路以及网络架构的优劣，这些都体现在模型的性能表现上，最直观的体现就是准确率。而对于模型在移动端部署来说，保证算法性能的同时，还需要注意设备内存占用以及计算能力的限制。衡量这两者的指标是网络的参数量（params）和计算量（FLOPs），参数量代表模型消耗的计算机内存，体现了网络的空间复杂度；计算量代表模型计算需要的时间，体现了网络的时间复杂度。

（一）参数量

不同的网络层参数的计算方法不同，1D CNN 网络中只有 1D CNN 层和全连接层占用内存，1D CNN 层参数的计算公式如式(3-2)所示：

$$params = (M \times K + 1) \times filters \quad (3-2)$$

其中， M 为卷积核高度， K 为卷积核宽度， $filters$ 为卷积核数量。本节优化超参数后的 1D CNN 模型中，Conv1D_1 的参数量为 $(9 \times 7 + 1) \times 128 = 8192$ ，Conv1D_2 的参数量为 $(128 \times 7 + 1) \times 64 = 57408$ 。

全连接层参数的计算公式如式(3-3)所示：

$$params = (N_{in} + 1) \times N_{out} \quad (3-3)$$

其中, N_{in} 为输入特征向量的维度, N_{out} 为输出特征向量的维度。本节优化超参数后的 1D CNN 模型中, 全连接层的参数量为 $(3712+1) \times 100 = 371300$, 输出层的参数量为 $(100+1) \times 6$ 。所以整个 1D CNN 网络的参数量为 437506, 在 Keras 库中可以直接调用 `model.summary` 函数获得参数量。

(二) 计算量

神经网络中用浮点运算次数 (Floating Point Operations, FLOPs) 代表计算量的大小。在一个模型进行预测计算时, 所有的操作如卷积、池化、激活函数都会产生相应的计算量^[39]。其中卷积操作产生的计算量最大, 对于一个 $k \times k$ 大小的卷积核, 一次卷积操作就会产生 $k \times k \times n$ 次乘法, 以及 $k \times k \times n - 1$ 次加法, n 表示输入特征维度。计算各层 FLOPs 的方法比较复杂, 在此不详细展开。在 keras 框架中, 可以通过导入 `keras_flops` 包调用它的 `get_flops` 函数直接获得整个网络的 FLOPs。整个 1D CNN 网络的 FLOPs 为 39109500。

3.3 多通道卷积神经网络 (Multi-channel CNN)

上一节设计的一维卷积神经网络为单通道输入卷积神经网络, 这样的网络结构对数据所有的特征分量都使用相同大小的卷积核提取特征。这样的操作就默认所有的特征分量对卷积的反应是一致的, 但对于多传感器数据融合的人体姿态检测过程, 每个传感器数据对于卷积的敏感程度可能并不相同, 使用同样配置的卷积头相当于对所有类型的数据取平均效果, 并不是对所有类型的数据都达到最好的效果, 所以应该设计一个卷积神经网络对每一类的数据都达到最好的特征提取效果, 从而进一步提高网络识别性能。

为了达到这一目的, 对于多类型输入数据来说, 可以使用多通道卷积神经网络 (Multi-channel CNN)。Multi-channel CNN 网络的设计思想在于将每个传感器的三轴数据分开处理, 使用三个单独通道的卷积头对每个传感器的数据进行独立的特征提取^[40]。这样的好处在于, 可以根据不同类型传感器数据对于时间长度的敏感程度不同, 模型配置不同的超参数。例如, 三个通道设置不同大小的 `kernel size`, 允许模型以不同的分辨率提取特征、解释传感器数据。分开提取到特征图后, 对特征图降采样减少参数数量, 然后让三个通道的特征图经过展平层展平成一维向量。在全连接层做出预测之前, 先将三个一维向量拼接成一个一维长向量, 最后输出层输出预测结果。总结起来, 就是分开提取特征, 合并进行预测。

Multi-channel CNN 网络设计的主要目的是, 可以使用不同大小的卷积核对不同类型的传感器数据卷积提取特征, 其余超参数的影响与单通道卷积神经网络一

致。每个卷积头的超参数设置除了卷积核大小不同，其余与 1D CNN 网络寻找到的最佳配置一致，即卷积层数为 2，第一层有 128 个卷积核，第二层有 64 个卷积核。将三个卷积头的卷积核大小分别设置，在两个数据集上每组重复进行十次实验，统计网络识别准确率，得到图 3-16 和图 3-17 所示结果，可以发现每个卷积头都使用大小为 7 的卷积核提取特征效果最好，网络识别准确率分别达到 93.729% (+/-0.898) 和 97.792% (+/-0.342)，与之前使用相同超参数配置的单通道卷积神经网络相比提升很大。

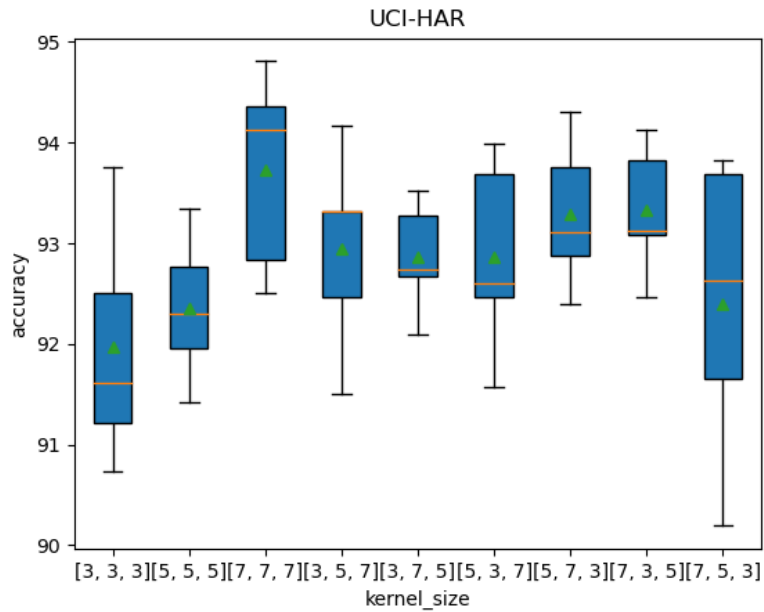


图 3-16 Multi-channel CNN 在 UCI-HAR 上的网络性能

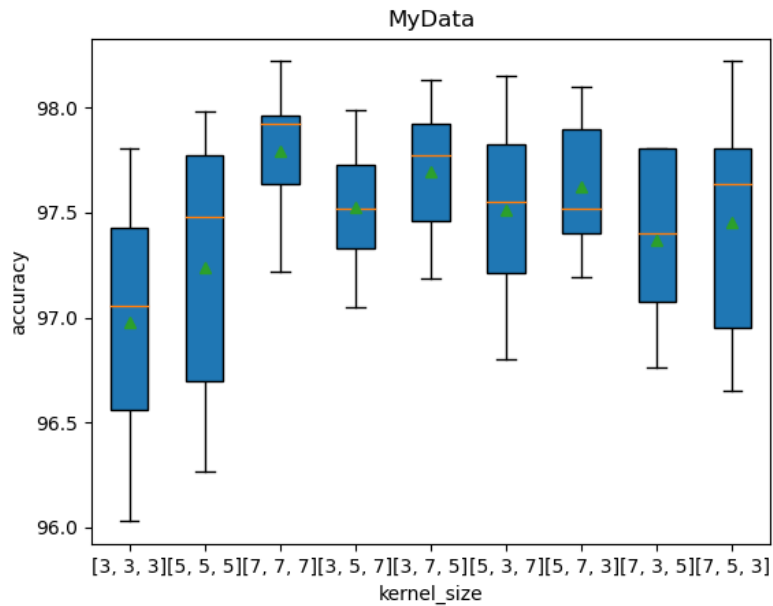


图 3-17 Multi-channel CNN 在 MyData 上的网络性能

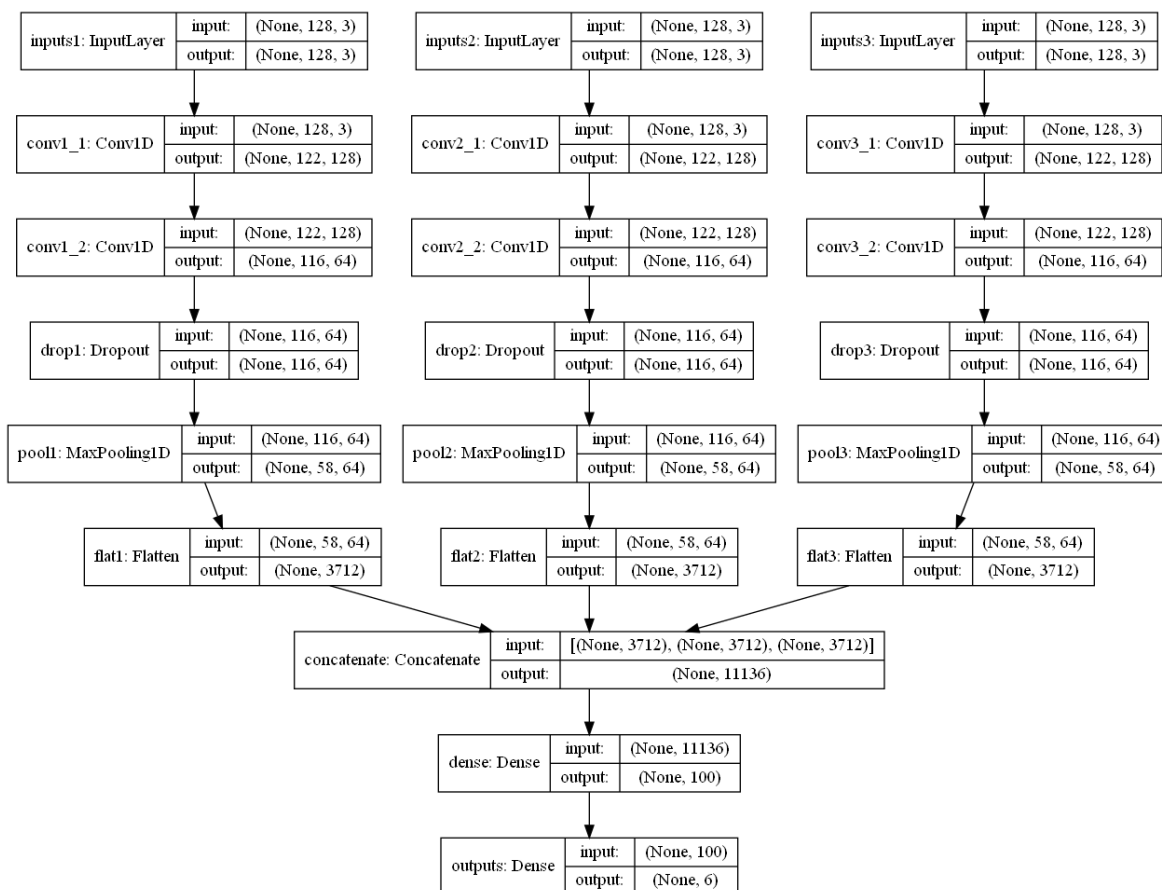


图 3-18 多通道卷积神经网络结构图

图 3-18 所示为 Multi-channel CNN 网络结构图，将三类传感器数据分开提取特征，加速度计数据从 inputs1 输入，线性加速度数据从 inputs2 输入，角速度数据从 inputs3 输入，并行处理之后合并解释特征，做出预测。Multi-channel CNN 网络的结构比更加复杂，参数量和计算量都有所增加，Multi-channel CNN 网络的参数量为 1294978，FLOPs 为 32547459。

3.4 实验结果对比

图 3-19 所示为用 UCI-HAR 数据集训练得到的 Multi-channel CNN 网络混淆矩阵。与图 3-14 对比可以发现，1D CNN 网络中对站立和坐下两个姿态的检测混淆的问题，在 Multi-channel CNN 模型中仍然存在，但情况得以改善，测试集中站立姿态被误检测成坐下姿态的样本数明显减少，这一部分样本是对 1D CNN 模型检测准确率影响最大的部分，所以 Multi-channel CNN 整体检测准确率要高得多。

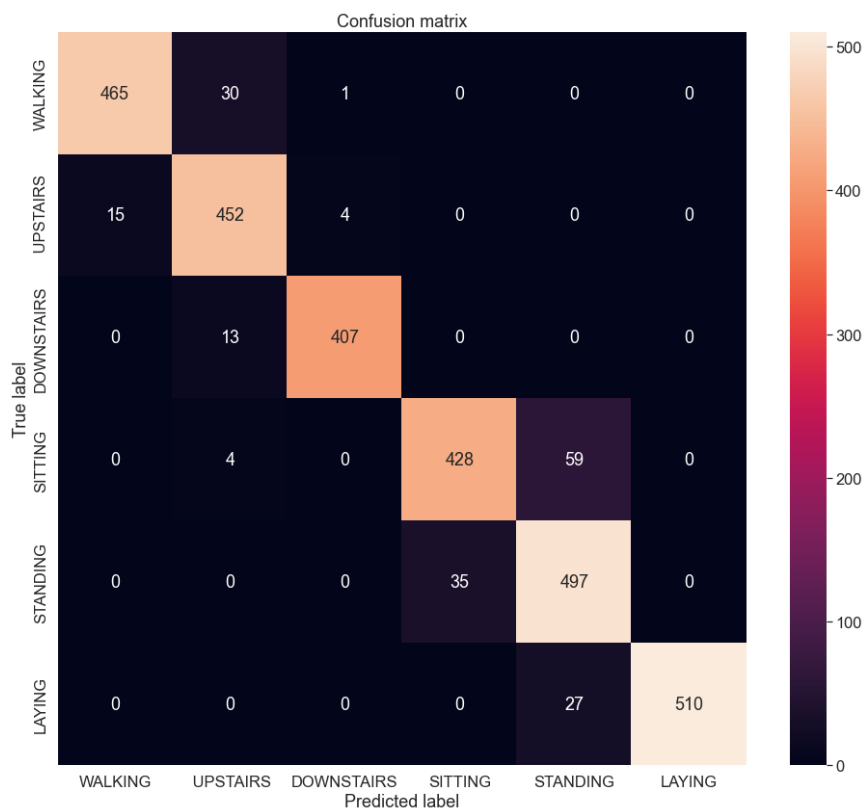


图 3-19 Multi-channel CNN 在 UCI-HAR 上的混淆矩阵

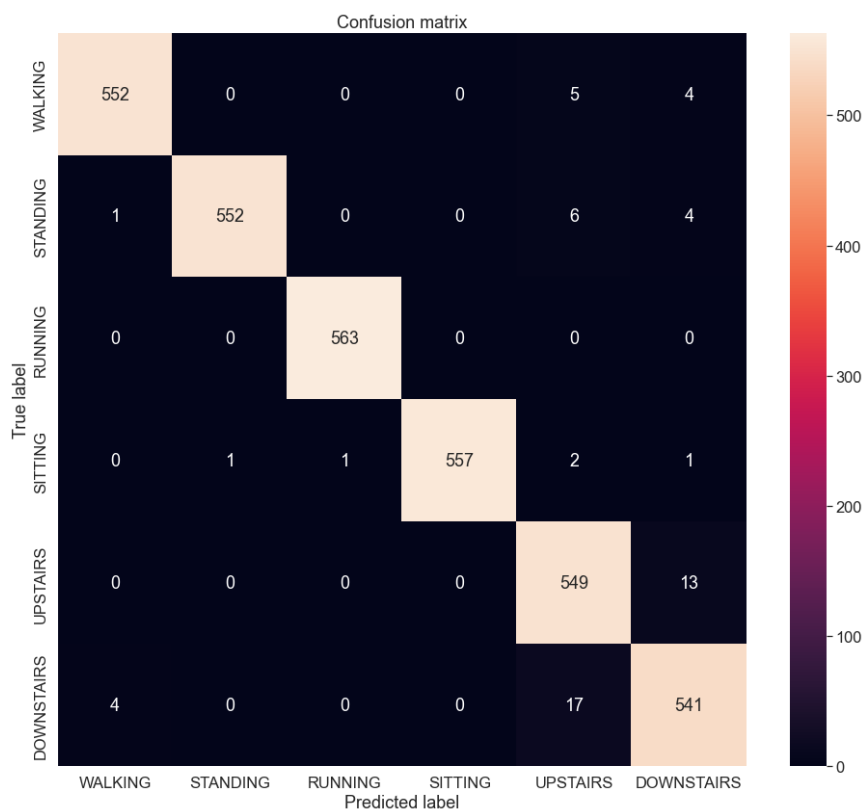


图 3-20 Multi-channel CNN 在 MyData 上的混淆矩阵

图 3-20 所示为用 MyData 数据集训练得到的 Multi-channel CNN 网络的混淆矩阵。与图 3-15 对比可以发现，1D CNN 模型中坐下姿态有一大部分被检测成步行的问题，在 Multi-channel CNN 模型中得以解决，只有一小部分的上下楼姿态检测相互混淆。通过以上对比实验可以发现，Multi-channel CNN 网络不仅总体检测准确率远高于 1D CNN 网络，还能解决后者存在的某类姿态大量误检测问题，是一个表现更加优秀的神经网络。

3.5 本章小结

本章主要介绍了数据集构建的过程以及 1D CNN 网络和 Multi-channel CNN 网络的设计与实现。首先对 UCI-HAR 数据集进行数据分析，证明了使用传感器数据进行人体姿态检测的可行性，并针对该数据集的缺陷改良数据采集方式，制作了 MyData 数据集。然后针对传感器数据的特点设计并实现了 1D CNN 网络，并通过一系列超参数配置实验，优化了网络的检测性能，提高检测准确率。最后针对 1D CNN 网络的缺陷设计并实现了 Multi-channel CNN 网络，进一步提高了网络的检测准确率。

第四章 基于 ConvLSTM 的人体姿态检测

本章主要内容为长短时记忆网络 (LSTM) 以及它的变体 CNN-LSTM 网络和 convLSTM 网络的设计与实现。首先分析了循环神经网络的缺陷, 然后介绍了 LSTM 网络的模型结构以及相关超参数实验。然后设计了 CNN-LSTM 网络, 使用 CNN 层对输入数据进行特征提取, 然后结合 LSTM 层来支持序列分类任务。然后设计了 convLSTM 网络, 将 LSTM 神经元中每个门的矩阵乘法换成了卷积运算, 通过卷积操作来捕获基础空间特征。最后对三四章提出的所有网络性能与其他研究者提出的分类算法进行了横向对比。

4.1 循环神经网络 (RNN) 及其缺陷

上一章介绍了卷积神经网络的特性, 并设计了两个用于提取单个时序信号特征的卷积神经网络, 但它们都只能单独处理每一段时序序列, 与前一段数据和后一段数据没有任何关联。人体姿态检测中, 每个滑动窗口都会包含几个完整活动, 也可能会出现一个完整活动被前后两个窗口切割的情况。所以人体姿态检测任务需要更好地处理连续性传感器数据, 传感器数据中前后的输入具有关联性, 我们分析传感器数据时不能只单独地去分析某一帧的数据, 而是要分析所有帧连接起来的整个序列。深度学习领域中, 研究者提出了一类非常重要的神经网络——循环神经网络 (RNN), 用来专门解决训练具有时间相关性数据的问题。

但是在实际应用中发现, RNN 只能很好解决具有短时间依赖的数据, 对于处理传感器数据这种较长时间序列的效果并不理想。本文实验中搭建了一个具有单层 RNN 的神经网络, 分别在两个数据集上重复训练 10 次, 在 UCI-HAR 数据集上得到的准确率仅为 67.550% (+/-8.514), 在 MyData 数据集上得到的准确率仅为 75.436% (+/-9.358), 不仅准确率低, 而且随机性非常高, 不是一个很好的网络。主要原因在于, RNN 存在梯度消失和梯度爆炸问题, 在训练较长序列的过程中无法将梯度一直向前后传递, 使得网络无法获得数据前后长时间的影响^[41]。

循环神经网络的训练算法叫 BPTT 算法, 主要包括以下四个步骤:

- 1、前向计算每个神经元的输出值。
- 2、反向计算每个神经元的误差项 δ , 由误差函数 E 对每个神经元 j 的加权输入 net_j 求偏导得出。
- 3、计算每个权重的梯度。
- 4、通过随机梯度下降算法更新权重。

BPTT 算法将 δ_k^T 定义为第 T 层 k 时刻的误差项, 在反向传播的过程中, 不仅要向上一层传播误差项, 得到 δ_k^{T-1} , 这部分误差只和权重矩阵 U 相关; 还要从当前时刻开始, 向之前所有时刻传播误差项直到初始时刻 t_1 , 得到 δ_t^T , 这部分只和权重矩阵 W 相关。 δ_k^T 沿时间反向传播的算法如式(4-1)所示, 其中 $diag$ 为对角矩阵。

$$\begin{aligned}\delta_k^T &= \frac{\partial E}{\partial net_k} = \frac{\partial E}{\partial net_t} \frac{\partial net_t}{\partial net_k} = \frac{\partial E}{\partial net_t} \frac{\partial net_t}{\partial net_{t-1}} \frac{\partial net_{t-1}}{\partial net_{t-2}} \dots \frac{\partial net_{k+1}}{\partial net_k} \\ &= Wdiag[f'(net_{t-1})]Wdiag[f'(net_{t-2})] \dots Wdiag[f'(net_k)]\delta_t^T \\ &= \delta_t^T \prod_{i=k}^{t-1} Wdiag[f'(net_i)]\end{aligned}\quad (4-1)$$

由式(4-1)可式(4-2)得:

$$\begin{aligned}\|\delta_k^T\| &\leq \|\delta_t^T\| \prod_{i=k}^{t-1} \|W\| \|diag[f'(net_i)]\| \\ &\leq \|\delta_t^T\| (\beta_w \beta_f)^{t-k}\end{aligned}\quad (4-2)$$

其中 β_w 和 β_f 均为矩阵模的上界, 从当前时刻向之前时刻传递误差项 δ_k^T 的过程中, $t-k$ 会逐渐减小, 由于指数爆炸的性质, 会导致误差项的值迅速增大一个很大的值或迅速缩小到 0 (取决于 $\beta_w \beta_f$ 是大于 1 还是小于 1), 导致误差项无法传递很远, 这就是梯度爆炸和梯度消失的原因。为了解决这个问题, 研究者提出了一种改进的循环神经网络: 长短时记忆网络。

4.2 长短时记忆网络 (LSTM)

长短时记忆网络由 Hochreiter 和 Schmidhuber 两位科学家提出^[42], 它解决了原始循环神经网络存在的梯度传播问题, 成为了目前最流行的一种循环神经网络。

4.2.1 LSTM 模型搭建

第二章提到过, 原始 RNN 的隐藏层只有一个短期状态, 它只对于短期的输入敏感, LSTM 网络则多添加了一个长期状态, 并通过三个门控制。由于每一个神经元都包含这些门控单元以及一些中间状态, 参数数量也会成倍增加, 参数的数量与输入数据的特征维度以及 LSTM 层的神经元数量有关, 具体计算公式见式(4-3)。

$$params = (m * n + n^2 + n) * 4 \quad (4-3)$$

其中, $params$ 为 LSTM 层参数数量, m 为输入数据的特征维数, n 为 LSTM 层神经元的数量。假设输入数据为 9 维, 一个有 128 个神经元的单层 LSTM 层网

络，单是 LSTM 层参数数量就达到 70656 个，所以网络结构比起卷积神经网络来说要复杂得多，网络的训练过程也更加麻烦。

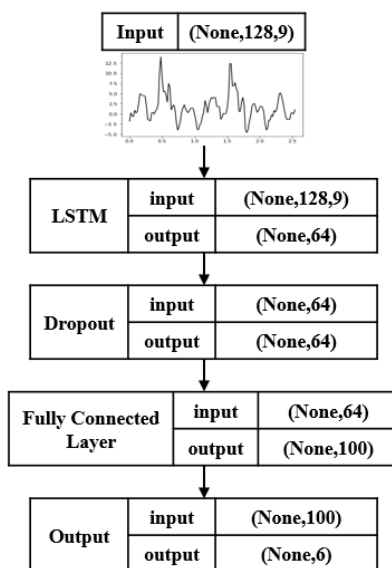


图 4-1 LSTM 网络结构

LSTM 网络的网络层数宏观上没有卷积神经网络层数那么复杂，不需要连接池化层以及展平层，整个网络结构只包括 n 层 LSTM 层、一层 Dropout 层、一层全连接层以及一层输出层^[43]，图 4-1 为只包含一层 LSTM 层的 LSTM 网络结构示意图，其中 LSTM 层神经元数量设置为 64，Dropout 层遗忘概率设置为 0.5，全连接层激活函数为 Relu。LSTM 层具有时间循环结构，对于前后时刻输入之间有长时间依赖关系的序列数据，可以很好地提取其时间特征；Dropout 层的作用是减少模型对训练数据的过拟合，提高模型的泛化能力；全连接层的目的是将前面提取的时间特征，在全连接层经过非线性变换，提取这些特征之间的关联，最后映射到输出空间上；最后通过输出层做出预测。分别使用 UCI-HAR 和 MyData 数据集来对其进行训练，每组实验重复 10 次。训练结束后在测试集上进行评估，在 UCI-HAR 数据集上得到的识别准确率为 89.671% (+/-1.049)，在 MyData 数据集上得到的准确率为 92.100% (+/-4.115)。通过经验配置超参数的 LSTM 网络，在两个数据集上的检测准确率相比于卷积神经网络都偏低，需要通过实验获得更高准确率的超参数配置。

4.2.2 超参数调整

对于一个 LSTM 网络的训练过程，首先也应该寻找一个合适的训练 epoch 大小，使得网络既不欠拟合也不过拟合。图 4-2 和图 4-3 为在两个数据集上 LSTM 模

型训练准确率 (accuracy) 和损失函数值 (loss) 随 epoch 变化的曲线。与图 3-7 和图 3-8 对比发现, LSTM 网络的训练难度确实比卷积神经网络更高, 随着 epoch 逐渐增加, train loss 的缓慢下降, 直到 epoch 达到 20 时 (在卷积神经网络中为 10), 才达到一个比较平稳的水平。观察 test loss 可以发现, test loss 随着 epoch 增加逐渐减小, 直到 epoch 达到 20 时开始回升。所以, 训练 LSTM 网络参数迭代过程 epoch 大小取 20 为最佳。

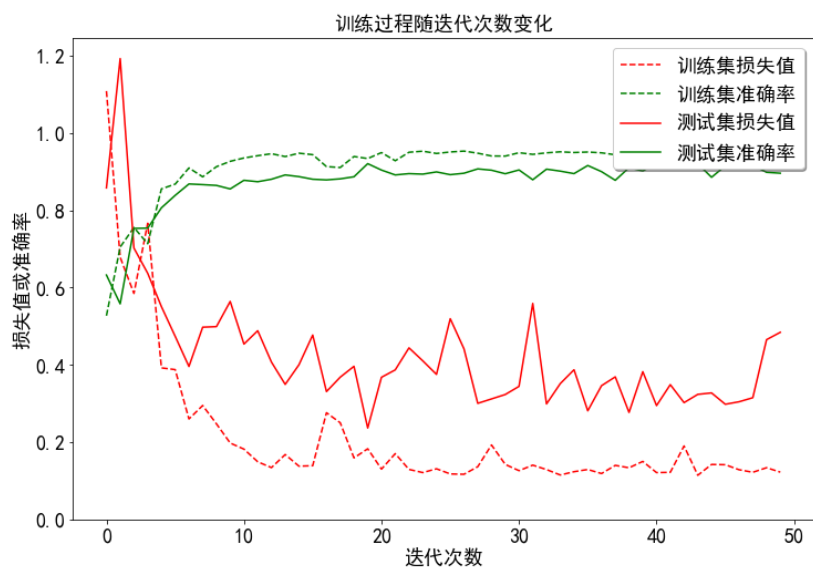


图 4-2 使用 UCI-HAR 数据集 LSTM 的 loss 和 accuracy 随 epoch 变化曲线

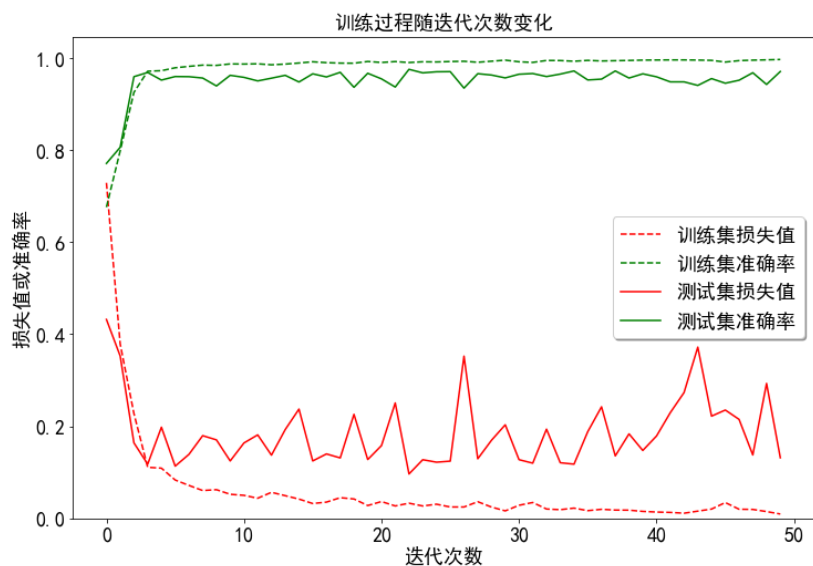


图 4-3 使用 MyData 数据集 LSTM 的 loss 和 accuracy 随 epoch 变化曲线

然后对 LSTM 层的数量进行实验, 已找到最适合的层数。理论上 LSTM 网络可以将第一个时刻的状态信息一直保留到最后一个时刻, 能处理无限时间跨度的

信息。LSTM 层的层数越多，模型能记住信息就越多，能处理时间跨度更长的时间序列。并且能增大参数搜索的空间，可以进一步提升学习信息的效率。但也不是一味地增加 LSTM 层的数量就能提高模型检测的准确率，因为模型能够记住的信息是有限的，一些距离当前时刻没有那么久远的信息就显得没那么重要，需要适当的取舍。针对不同的训练任务和数据输入，网络需要提取时间信息的跨度也不同，有些时间跨度较短的序列可能只需要一层 LSTM 层就能很好处理，层数增多反而会产生过拟合现象，降低网络检测性能。同时层数增加还会增加参数的数量，增加训练难度以及模型计算的时间。网络检测准确率相差不大的情况下，网络的层数越小越好。

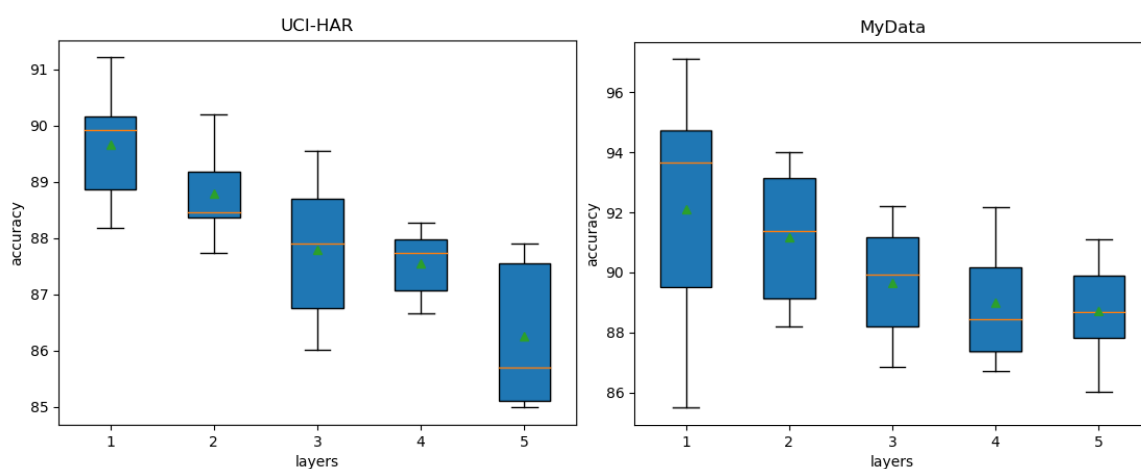


图 4-4 LSTM 网络中 LSTM 层数对准确率的影响

本文将 LSTM 层的层数设置为 1,2,3,4,5，在两个数据集上分别进行 10 次重复实验，统计结果并绘图。图 4-4 为 LSTM 网络中 LSTM 层数变化与网络识别准确率变化的箱型图。可以发现当层数为 1 层时网络在两个数据集上的平均准确率都最高，随着 LSTM 层数增加，模型在训练集上的平均识别准确率一直在下降。在稳定性方面，网络在 UCI-HAR 数据集上的检测准确率波动更小，而 MyData 数据集在层数为一层时，准确率波动很大，随着层数增多，波动范围开始减小。虽然在 MyData 数据集上设置单层 LSTM 准确率波动较大，但可以通过调节其它网络参数使模型更加稳定。所以 LSTM 层数设置为单层为最佳配置。

LSTM 层的另一个重要超参数是神经元数量 (units)。LSTM 层神经元的数量越多，模型能记住信息就越多，能处理时间跨度更长的时间序列。但和层数的影响一样，对于不同的训练任务，神经元的数量并不是越多越好。神经元的数量需要根据提取输入数据时间的时间跨度来设置，使模型刚好提取与当前时刻有关联的时间跨度内的信息为最佳，这样网络既不会欠拟合也不会过拟合。另外，随着 LSTM

层神经元数量增加, LSTM 网络还会显露一个非常明显的缺陷, 那就是参数的数量会随着神经元数量增加成平方级增加, 会使模型的训练难度和计算时间大大增加。本文将 LSTM 层的神经元数量分别设置为 16,32,64,128,256,512, 输入数据维度为 9, 根据(4-3)计算得到参数数量分别为 1664, 5376, 18944, 70656, 272384, 1069056。可以看到, 随着 LSTM 层神经元数量的增加, 单是一层 LSTM 层的参数数量增速都十分迅速。当神经元数量增加到 512 时, 网络规模对于移动端平台的计算能力来说, 已经算比较庞大了, 会影响模型在移动端运行的速度, 提高检测延迟。所以在工程应用中, 有些情况宁愿牺牲一部分模型检测的准确率, 也希望把 LSTM 层的神经元数量尽可能减少。

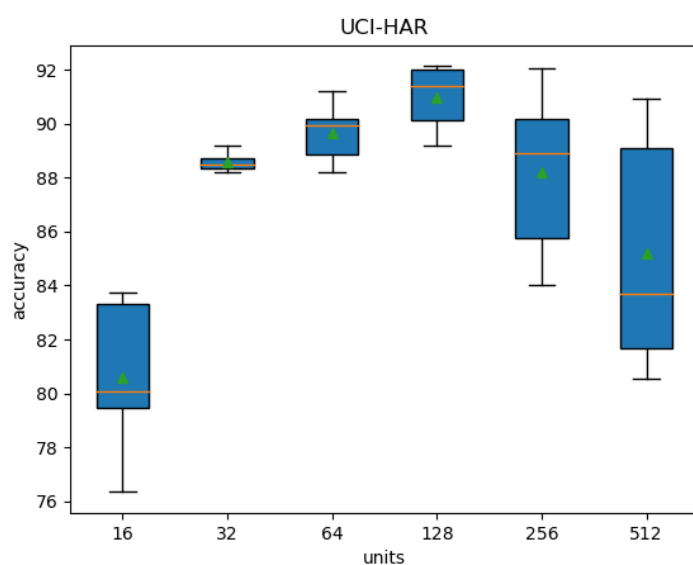


图 4-5 UCI-HAR 数据集 LSTM 层神经元数量对准确率的影响

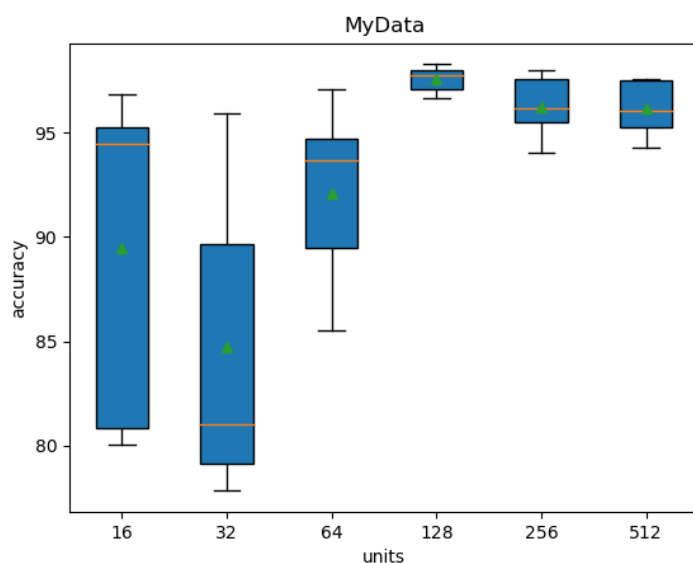


图 4-6 MyData 数据集 LSTM 层神经元数量对准确率的影响

将以上神经元数量配置，在两个数据集上分别进行 10 次重复实验，统计结果并绘图。图 4-5 和图 4-6 为 LSTM 网络中 LSTM 层神经元数量变化与网络识别准确率变化的箱型图。可以看到神经元数量较少时，网络性能极差，随着神经元数量增加网络平均检测准确率迅速增加，当神经元数量增加到 128 时表现最好，所以将 LSTM 层的神经元数量设置为 128。

经过上述一系列实验后，最终获得的 LSTM 网络，包含一层 LSTM 层，隐藏层的神经元数量为 128。整个网络结构所包含的参数量为 84162，FLOPs 为 3349890。最终在 UCI-HAR 数据集上得到的识别准确率为 90.974% (+/-1.129)，在 MyData 数据集上得到的识别准确率 97.545% (+/-0.597)，比最初的 89.671% (+/-1.049) 和 92.100% (+/-4.115) 更高。

4.3 1D CNN-LSTM 网络设计

通过在 LSTM 网络里面添加门控单元，使梯度问题得到了一定程度的解决，但还是不够。它可以处理一定长时间依赖关系的序列，但总有它的极限，当超过记忆状态的极限时，模型的预测效果也会下降，并且对于多维输入中所具有的空间关联特征不能很好的提取。只对于在输入中存在时间关联的学习任务比较适用，如视频中前后图像帧之间的关联以及传感器数据前后输入的关联等等。而卷积神经网络则能有效的发现输入数据的空间特征，对于在输入中具有空间关联的学习任务，如发现二维图像像素之间的关联以及姿态检测中多传感器数据之间的关联，这类问题比较适用。

对于一些输入中既存在时间关联又存在空间关联的学习任务，如视觉时间序列预测问题和生成图像序列（如视频）的文本描述问题。有科学家提出一种 CNN-LSTM 网络，将卷积层作为网络的前端来提取输入数据的特征，并紧接着 LSTM 层来进行序列预测或者分类，这种网络最早被称为长期递归卷积网络或 LRCN 模型^[44]。CNN-LSTM 是一类在时间上和空间上都很深的模型，在生成图像的文本描述、语音识别以及自然语言处理等问题上都有良好表现。基于多传感器数据的人体姿态识别任务，作为同样在空间和时间上都有很大关联性的一类问题，也能尝试将 CNN 网络和 LSTM 网络结合来解决。对此，本文设计了一种针对传感器数据输入特点的 1D CNN-LSTM 网络模型。

首先应该解决的就是 CNN 层和 LSTM 层之间的连接问题，要使两个层的输入输出维度相匹配。先来回忆一下两个层的输入输出维度。卷积层要求的输入格式为[样本数 (samples)，时间步长 (timesteps)，特征数 (features)]，卷积层对第二维和第三维数据进行一维卷积，计算得到的输出格式为[样本数，特征图长度

(`feature_size`), 特征图数量 (`maps`)], 经过展平层后会将其第二三维展平成一个长向量, 维度变为[样本数, 特征图长度*特征图数量]。而 LSTM 层要求的输入格式也是[样本数, 时间步长, 特征数]这样的三维数据, 这样的话, 卷积层部分的输出维度和 LSTM 的输入维度是不匹配的。

第一种方法是在展平层之后再连接一个 Reshape 层。Reshape 层的作用是修改除第一维之外的其它数据的维度, 不管是改变之前的还是改变之后的 `shape`, 数据第一维的 `None` 都是批样本数 (`batch_size`)。将展平之后的长向量输入 Reshape 层之后可以改变其尺寸, 特征图的维度由[样本数, 特征图长度*特征图数量]变为[样本数, 特征图长度*特征图数量, 1]。这样, 时间步长对应展平向量的长度, 特征数为 1 的单变量数据作为 LSTM 层的输入。

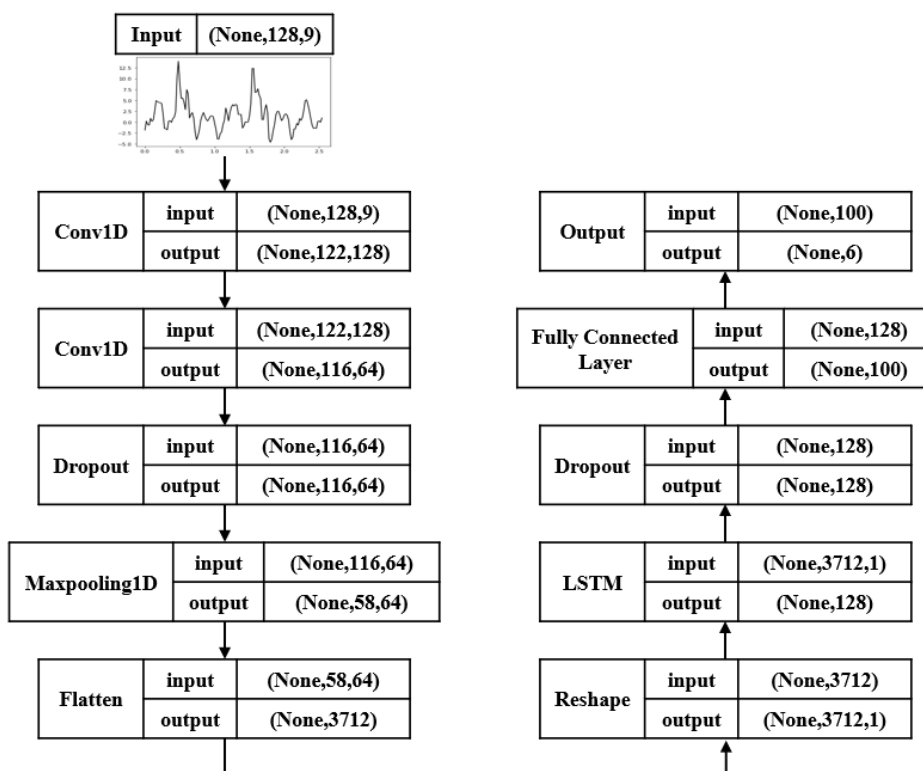


图 4-7 使用方法一维度匹配的 CNN-LSTM 网络

图 4-7 为使用上述方法进行维度匹配的 CNN-LSTM 网络, 三种方法中 CNN 层以及 LSTM 层的超参数配置一致, 皆沿用第三章和第四章实验得到的最佳设置。第一层卷积层 `filters` 设置为 128, `kernel_size` 设置为 7, 第二层卷积层 `filters` 设置为 64, `kernel_size` 设置为 7, LSTM 层神经元数量为 128。从网络中数据维度的变化可以发现, 这种方法有个非常大的缺陷, 就是 LSTM 层输入数据的时间步长过长, 达到了 3712。LSTM 层虽然能解决长期依赖的问题, 但隐藏层能记忆的信息

还是有上限的，训练的序列越长，发生梯度问题的风险也就越高，并且还会导致训练的时间大大增加。在一些大型的 LSTM 模型中，时间步长都会限制在 250-500 之间，而现在都到了 3712，实验过程中训练时间长到在服务器上训练都无法结束，可以预知最终的训练结果也不会很好。当然，也有减小 LSTM 层时间步长的方法，可以减少卷积层卷积核数量或者增加 kernel_size，这样做的目的是减小卷积层输出的特征图长度以及数量，就能减小展平后向量的长度，从而减小 LSTM 层输入的时间步长。但要使时间步长能减小到 500 以内的话，就需要大大减少卷积核的数量并增加 kernel_size 大小，例如 filters 设置为 16、kernel_size 设置为 35，得到的展平向量长度为 480，或者 filters 设置为 8、kernel_size 设置为 7，得到的展平向量长度为 464。由第三章相关实验可知，上述卷积层超参数配置对传感器数据进行特征提取效果很差，同样会导致网络最终的训练结果很差。这样的维度匹配方法，会导致 CNN 层和 LSTM 层相互矛盾，提升了一个的性能，就会导致另一个的性能下降。经过以上实验和分析发现，方法一虽然能实现 CNN-LSTM 网络的维度匹配，但会导致网络的性能非常差，所以排除方法一。

第二种方法是第一种方法的改进，在数据还没有输入到网络进行训练之前，先将输入数据用 reshape 函数改变维度，使卷积层的输入数据升维，每次读取输入数据前先将整个序列划分为若干等长的子序列，在以块的形式读取子序列，将卷积层输入数据的维度变成[样本数(samples), 块数(blocks), 子序列时间步长(length), 特征数(features)]这样的四维数据，维度转换关系见式(4-4)：

$$timesteps = blocks * length \quad (4-4)$$

输入卷积层后会对每个块进行卷积，对第三维和第四维数据进行一维卷积，计算得到的输出格式为[样本数, 块数, 子序列特征图长度, 子序列特征图数量]，经过展平层展平后变为三维数据[样本数, 块数, 子序列特征图长度*子序列特征图数量]，块数对应时间步长，子序列特征图长度*子序列特征图数量对应特征数，这样可以直接作为 LSTM 层的输入。

如图 4-8 所示，在传感器数据输入网络训练之前，先将每个输入样本的 128 个时间步长划分成 4 个子序列，每个子序列的时间步长为 32，供卷积层处理。将 CNN 模型应用于每个输入子序列，并将每个输入子序列的输出作为单个时间步长，所有的子序列输出组成一个完整的时间步长传递给 LSTM 模型处理，在 Keras 框架中通过 TimeDistributed 层来实现。使用 TimeDistributed 层可以将整个 1D CNN 模型包装起来，以允许相同的 1D CNN 模型读取窗口中四个子序列中的每一个子序列。然后将提取的特征展平并提供给 LSTM 模型进行读取，在最终映射到活动之前提取其自身的特征。在两个数据集上分别进行 10 次重复实验，在 UCI-HAR 数据集

上得到的识别准确率为 91.605% (+/-0.629)，在 MyData 数据集上得到的准确率为 97.736% (+/-0.577)。

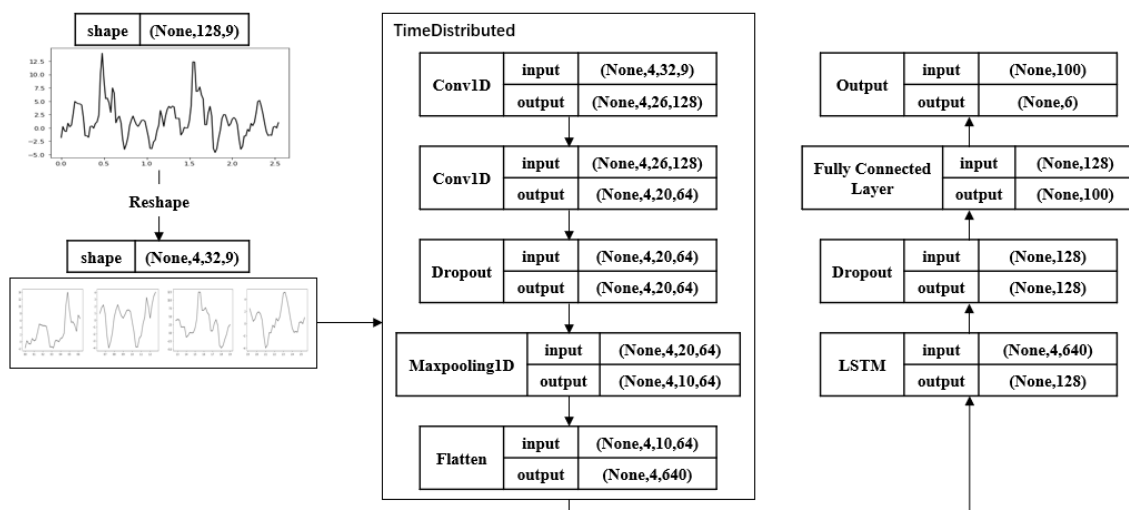


图 4-8 使用方法二维度匹配的 CNN-LSTM 网络

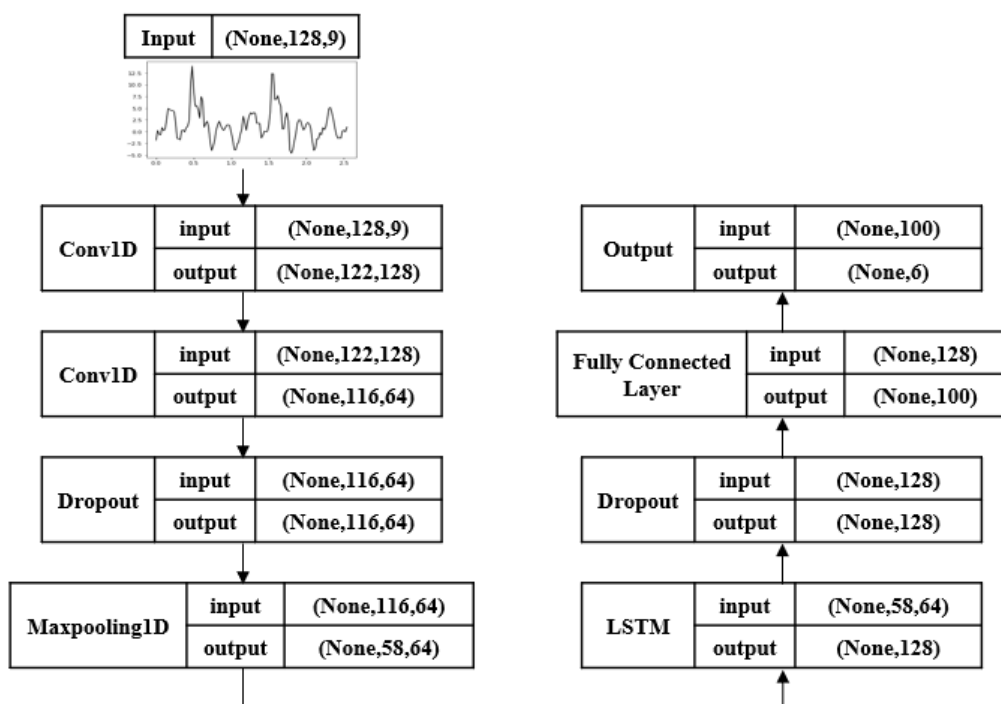


图 4-9 使用方法三维度匹配的 CNN-LSTM 网络

第三种方法是不在网络中添加展平层，不对卷积层输出的特征图展平，这样不会将卷积层的输出维度改变。可以直接将卷积层的输出维度和 LSTM 层的输入维度进行匹配，即把 LSTM 层输入的第二维（时间步长）对应卷积层输出数据第二维特征图长度，把 LSTM 层输入的第三维（特征数）对应卷积层输出数据第三维

特征图的数量。图 4-9 所示为使用方法三对卷积层输出和 LSTM 层输入进行维度匹配的 CNN-LSTM 网络。在两个数据集上分别进行 10 次重复实验，在 UCI-HAR 数据集上得到的识别准确率为 89.634% (+/-1.148)，在 MyData 数据集上得到的准确率为 96.196% (+/-1.561)。采用方法三实现的 CNN-LSTM 网络，不仅平均准确率上比较低，而且标准差也很大，网络性能波动较大。横向对比之前的网络，性能不但比方法二实现的 CNN-LSTM 网络差，还比第一节配置的 LSTM 网络差。主要原因在于不将特征图展平，会影响网络后续对特征图的解释。

综合以上实验和分析发现，在相同超参数配置下，采用方法二实现维度匹配的 CNN-LSTM 网络的平均准确率最高，分别达到了 91.605% (+/-0.629) 和 97.736% (+/-0.577)。整个网络的参数量为 472834，FLOPs 为 17593880。现在使用方法二并改变输入数据的子序列分割方式为 [8, 16] 和 [16, 8]，重复训练 10 次，以确定最佳的子序列分割方式，如图 4-10 所示。可以看出 [4, 32] 的子序列分割方式为最佳。

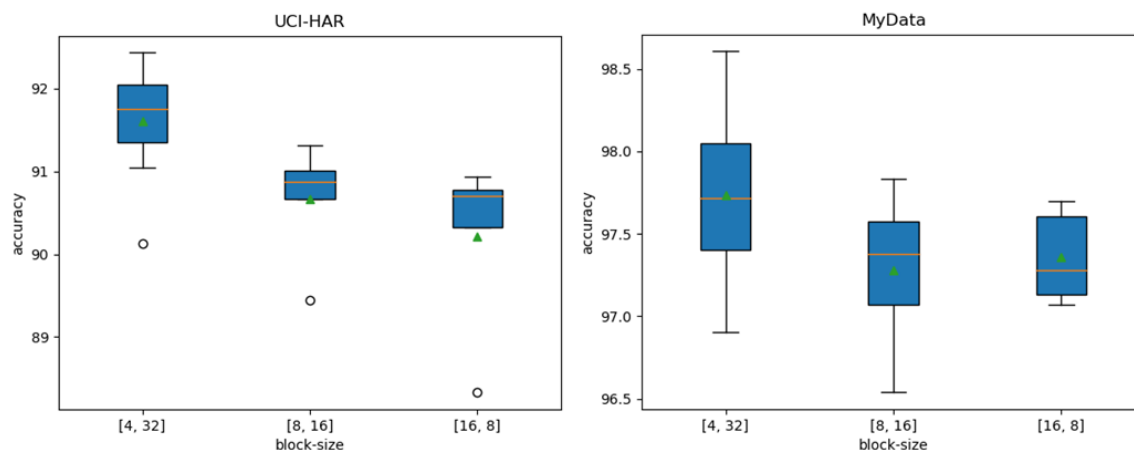


图 4-10 CNN-LSTM 子序列分割方式影响

4.4 卷积循环神经网络 (ConvLSTM)

之前提到 LSTM 网络包含大量空间的冗余数据，无法提取传感器数据的空间相关性。针对上述问题，Xingjian Shi 等人提出一种卷积循环神经网络 (Convolutional Long Short Term Memory Network, ConvLSTM) 结构，不仅可以向 LSTM 一样建立时序关系，而且可以像 CNN 一样刻画局部空间特征。该模型在论文中被首次提出是为了解决降水临近预报，后来逐渐被其他研究者应用在解决其他时空序列的预测问题^[45]。在本文中，设计了一个 ConvLSTM 网络用来解决基于传感器数据的人体姿态检测问题。

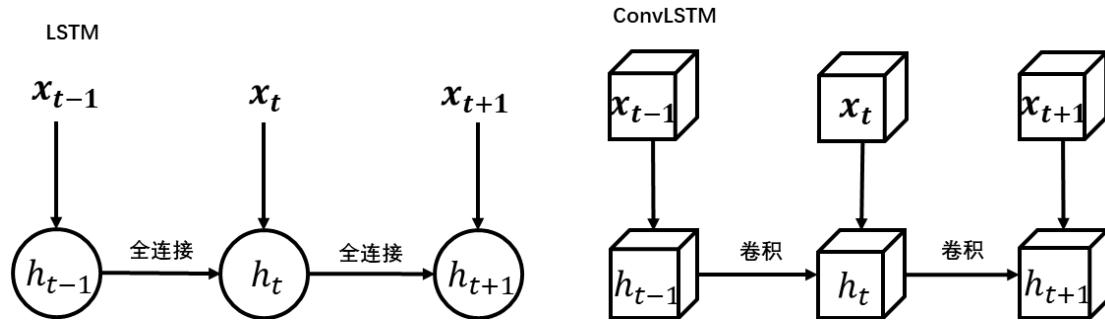


图 4-11 LSTM 和 ConvLSTM 的区别

如图 4-11 所示, ConvLSTM 网络是基于 LSTM 网络进行改进的一种网络。两者的输入维度不同, LSTM 每一个时间步输入是一个向量, 对于 ConvLSTM 来说, 输入是三维形式的, 即每一个时间步都是一个三维输入。另外, ConvLSTM 将 LSTM 每个门控单元中的矩阵乘法换成了卷积运算。矩阵乘法的作用将数据的整体信息直接全部相乘得到一个值, 这样无法有效的提取数据的时空特征。而改成卷积运算, 则可以很好的提取特征, 如果卷积核大, 捕捉到的动作更快, 如果卷积核小, 捕捉到的动作更慢。

LSTM 中门控单元、长期状态以及短期状态的计算公式, 在第二章中式(2-10)到式(2-15)就已经给出, 其中权重矩阵 W 都是由两个矩阵拼接而成的, 一个是 W_h , 大小为 $d_c \times d_h$, 对应输入 h_{t-1} ; 一个是 W_x , 大小为 $d_c \times d_x$, 对应输入 x_t 。所以矩阵乘法可以改写为式(4-5)的形式

$$W \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} = \begin{bmatrix} W_h & W_x \end{bmatrix} \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} = W_h h_{t-1} + W_x x_t \quad (4-5)$$

将式(4-5)代入式(2-10)到式(2-15), 可以推出 LSTM 网络中所有状态的更新公式, 如式(4-6)到(4-10)所示。这里面的关于权重 W 的操作, 跟全连接类似, 因此这种 LSTM 又可以叫 FC-LSTM, 也是上一节所使用的 LSTM 网络结构。

$$i_t = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \quad (4-6)$$

$$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \quad (4-7)$$

$$o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + W_{co} \circ c_t + b_o) \quad (4-8)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c) \quad (4-9)$$

$$h_t = o_t \circ \tanh(c_t) \quad (4-10)$$

其中, x 、 h 、 c 、 i 、 o 、 f 都是一维特征向量, \circ 表示矩阵对应元素相乘。ConvLSTM 网络的核心思想和 LSTM 网络还是一致的, 将之前的输出记忆在短期

状态 h 和长期状态 c 中，作为当前层的输入。不同之处在于将状态和状态之间的运算由矩阵乘法换成了卷积运算，如式(4-11)到式(4-15)所示。

$$I_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \quad (4-11)$$

$$F_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \quad (4-12)$$

$$O_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \quad (4-13)$$

$$C_t = F_t \circ C_{t-1} + I_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \quad (4-14)$$

$$H_t = O_t \circ \tanh(C_t) \quad (4-15)$$

其中， X 、 H 、 C 、 I 、 O 、 F 都是三维张量，后两个维度表示行和列的空间信息， $*$ 表示卷积运算。

Keras 框架向开发者提供了 ConvLSTM2D 封装层，通过该类搭建 ConvLSTM 模型支持处理二维输入数据，对于一维多变量的时间序列分类任务，通过适配输入数据的维度也可以进行处理。根据 Keras 官方的 API 文档说明，ConvLSTM2D 层的输入数据要求的维度(shape)为[样本(samples), 时间步长(timesteps), 行(rows), 列(columns), 通道(channels)]，其中每个时间步的输入数据格式都定义为(行 \times 列)的图像数据形式。要处理多源传感器数据序列，首先应该通过 reshape 函数将数据维度进行匹配。可以采用与 CNN-LSTM 网络相同的子序列方法来划分 ConvLSTM2D 的输入。将每个输入样本的 128 个时间步长划分成 4 个子序列，每个子序列的时间步长为 32，划分的子序列数对应时间步。对于一维序列，可以将行或者列中的一维设置为 1，另一维设置为每个子序列中的时间步长。经过维度匹配后，输入数据维度由[n, 128, 9]变成了[n, 4, 1, 32, 9]，具体的参数说明如下：

- 样本 (samples): n, 表示训练集样本总数。
- 时间步长 (timesteps): 4, 将一个有 128 个时间步长的输入数据窗口划分为 4 个子序列窗口。
- 行 (rows): 1, 表示一维子序列的形状。
- 列 (columns): 32, 表示每个子序列中的时间步长。
- 通道 (channels): 9, 本来是图像处理任务中的概念，在多维传感器数据分类任务中对应的是特征数 (features)，对应每个时刻的九轴传感器数据。

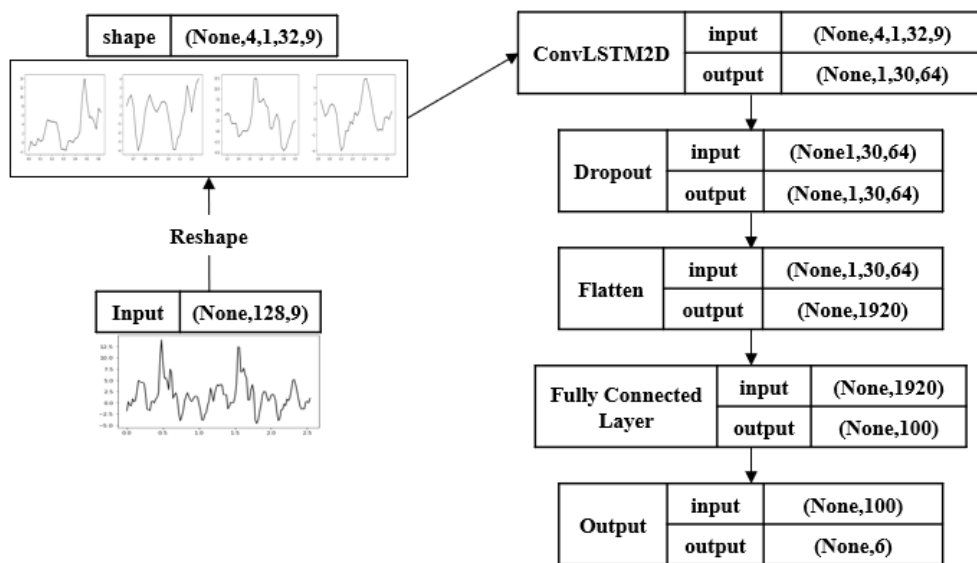


图 4-12 ConvLSTM 网络结构

修改好输入数据的维度后，开始搭建 ConvLSTM 网络，ConvLSTM 网络结构如图 4-12 所示。输入数据首先输入到单层 ConvLSTM2D 层中，超参数配置包括 CNN 和 LSTM 的相关超参数，如神经元数量，二维卷积核大小以及激活函数。本文将神经元数量设置为 128，二维卷积核的大小设置为 (1,7)，激活函数使用 Relu 函数。其中卷积核的行数必须设置为 1，因为子序列行数为 1。二维卷积的计算公式如式(4-16)所示：

$$a_{j,i} = f\left(\sum_{m=0}^M \sum_{n=0}^N w_{m,n} \cdot x_{j+m,i+n} + b\right) \quad (4-16)$$

其中， $a_{j,i}$ 为特征图的第 j 行，第 i 列元素， $f()$ 表示激活函数， M 为卷积核的行数减一， N 为卷积核的列数减一， $w_{m,n}$ 表示卷积核第 m 行第 n 列的权重， $x_{j+m,i+n}$ 表示数据第 $j+m$ 行第 $i+n$ 列元素， b 为偏置项。当卷积核的行数设置为 1 时 $M=0$ ，式(4-16)就简化为式(2-7)所示的一维卷积公式，一维子序列也同样可以用二维卷积核来提取特征。与 CNN 模型和 LSTM 模型一样，数据经过 ConvLSTM2D 层处理之后还需要经过 Dropout 层降低过拟合风险，遗忘概率设置为 0.5。之后连接 Flatten 层将提取到的特征图展平，才能输入到全连接层中解释特征，最后输入到输出层输出检测结果。

搭建的 ConvLSTM 网络在两个数据集上分别进行 10 次重复实验，在 UCI-HAR 数据集上得到的识别准确率为 94.117% (+/-0.731)，在 MyData 数据集上得到的准确率为 98.591% (+/-0.513)。网络识别性能获得大大提升，整个网络的参数量为 825026，FLOPs 为 25428346。

4.5 实验结果对比

为了完成基于多传感器数据的人体姿态检测任务，本文在第三章设计了一维卷积神经网络以及它的变体多通道卷积神经网络，在本章首先设计了 LSTM 网络，然后结合 CNN 网络以及 LSTM 网络的特点设计了 1D CNN-LSTM 网络和 ConvLSTM 网络。每个网络都分别在公开数据集 UCI-HAR 以及自建数据集 MyData 上进行训练，并在测试集上评估性能，统计了网络的平均准确率、标准差、参数量以及计算量等指标。为了验证五个网络的优劣性，现在对比每个网络在两个数据集上的各项指标，如表 4-1 和表 4-2 所示。

表 4-1 UCI-HAR 数据集在各网络性能

网络名称	平均准确率	标准差	参数量	FLOPs
1D CNN	91.716%	+/-0.610	437,506	39,109,500
Multi-channel CNN	93.729%	+/-0.898	1,294,978	32,547,459
LSTM	90.974%	+/-1.129	84,162	3,349,890
1D CNN-LSTM	91.605%	+/-0.629	472,834	17,593,880
ConvLSTM	94.117%	+/-0.731	825,026	25,428,346

表 4-2 MyData 数据集在各网络性能

网络名称	平均准确率	标准差	参数数量	FLOPs
1D CNN	96.768%	+/-0.294	437,506	39,109,500
Multi-channel CNN	97.792%	+/-0.342	1,294,978	32,547,459
LSTM	97.545%	+/-0.597	84,162	3,349,890
1D CNN-LSTM	97.736%	+/-0.577	472,834	17,593,880
ConvLSTM	98.591%	+/-0.513	825,026	25,428,346

将以上五个模型部署在 OPPO R17 手机上运行，搭载处理器为高通 SDM670 八核，发现都能满足实时检测要求，因此优先选取分类准确率更高的模型。发现对基于多传感器数据的人体姿态检测任务来说，传统的 CNN 模型和 LSTM 模型都只能够在时间或空间一个维度上提取数据的特征，所以识别准确率较低。Multi-channel CNN 模型通过设置多个卷积头的方法，能有效提高模型的预测准确率。而 ConvLSTM 模型通过将门控单元的矩阵乘法换成卷积核的方法，不仅进一步提高模型的预测准确率，还在内存占用和计算时间上优化，比 Multi-channel CNN 模型拥有更少的参数量和计算量，更适合在移动端部署。

为了说明本文提出的 Multi-channel CNN 模型和 ConvLSTM 模型在人体姿态检测上的效果，将两个模型在公开数据集 UCI-HAR 上重复训练 10 次，取其中的最佳结果作为最终的检测准确率，与其他研究者在该数据集上提出的分类方法进行横向对比。

表 4-3 本文与传统方法比较

研究者	分类方法	准确率
Davide Anguita 等 ^[15]	支持向量机	89%
Charissa Ann Ronao 等 ^[46]	隐马尔科夫模型	91.85%
Shilong Yu 等 ^[47]	BiLSTM 模型	93.8%
本文	Multi-channel CNN 模型	94.231%
本文	ConvLSTM 模型	94.639%

由上表可知，深度学习方法不仅比传统机器学习方法的分类效果更好，还不需要对传感器数据进行特征工程。经过对比，本文提出的 Multi-channel CNN 模型和 ConvLSTM 模型都比传统方法的性能优秀。最终选取 ConvLSTM 模型作为在移动端部署的模型，其在 UCI-HAR 数据集上的最佳准确率为 94.639%，在 MyData 数据集上的最佳准确率为 99.140%。

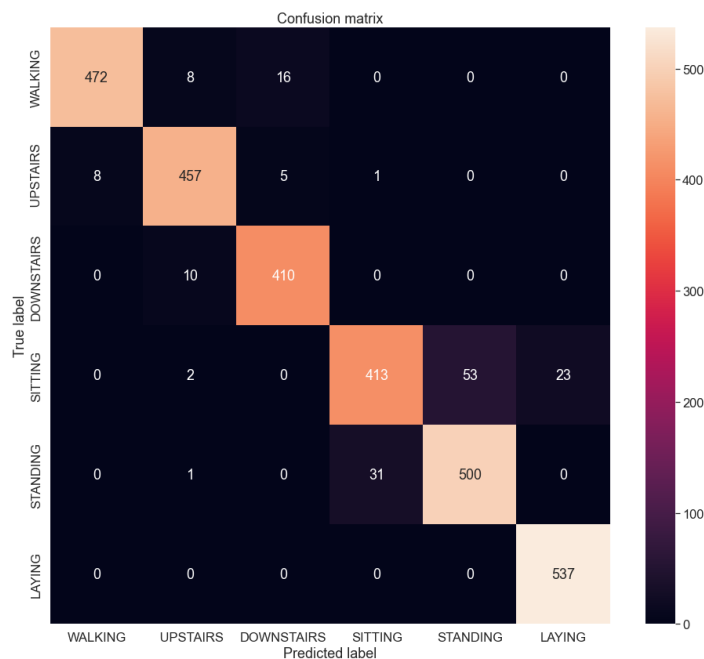


图 4-13 ConvLSTM 在 UCI-HAR 上的混淆矩阵

图 4-13 为 ConvLSTM 模型在 UCI-HAR 数据集上获得最佳准确率时的混淆矩阵。与第三章 Multi-channel CNN 模型相比,除了坐下姿态识别正确样本略有下降,其余姿态识别正确样本均有提升。

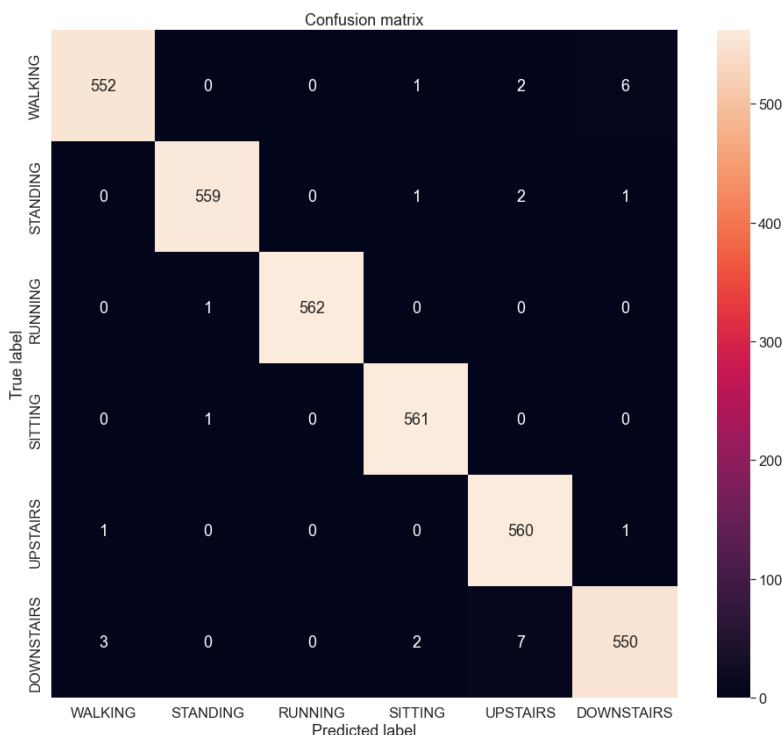


图 4-14 ConvLSTM 在 MyData 上的混淆矩阵

图 4-14 为 ConvLSTM 模型在 MyData 数据集上获得最佳准确率时的混淆矩阵。与 Multi-channel CNN 模型相比,上楼和下楼姿态的识别准确率明显提升。将本次在 MyData 数据集上训练得到的 ConvLSTM 模型保存,作为最终在移动端进行部署的模型。

4.6 本章小结

本章主要介绍长短时记忆网络 (LSTM) 以及它的变体: CNN-LSTM 网络和 ConvLSTM 网络。本章首先介绍了 RNN 网络,并分析了其在处理传感器数据上表现不佳的原因;随后设计了针对 RNN 网络缺陷改进的 LSTM 网络,并通过一系列超参数配置实验获得了有关 LSTM 层的最优配置;然后结合 CNN 网络和 LSTM 网络的特点设计了 CNN-LSTM 网络,并解决了 CNN 层和 LSTM 层维度匹配的问题;然后设计了 ConvLSTM 网络,它将 LSTM 层门控单元中的矩阵乘法换成了卷积运算,网络识别性能大大提升;最后对比了第三章和第四章设计的所有网络的性能,选择了 ConvLSTM 模型作为在移动端部署的模型。

第五章 人体姿态检测系统设计与实现

本章主要介绍了第四章得到的 ConvLSTM 模型在移动端部署以及人体姿态检测系统。首先介绍了人体姿态检测系统实现的主要功能。然后介绍了通过 TensorFlow 训练好的 ConvLSTM 模型转化为可移植模型的方法，并详细介绍了移动端应用和后台管理系统实现的过程。最后通过实验验证了在手机 APP 中部署的移动端神经网络模型的准确率。

5.1 人体姿态检测系统概述

本文实现的人体姿态检测系统的主要功能包括：姿态检测、活动记录以及健康提醒。具体功能如下：利用手机多个传感器数据进行人体姿态检测，然后将用户的检测结果、检测时间以及个人信息等数据存储在本地和服务器。通过这种方式，对用户的日常活动进行监控和统计，并将统计结果反馈给用户，以使用户知道自己的每日活动情况。通过用户活动记录可以进行健康监控，监控用户一些不健康的生活习惯，比如久坐、运动过量等等，并予以相关的健康提示，例如提醒用户久坐后需要起来活动一下。

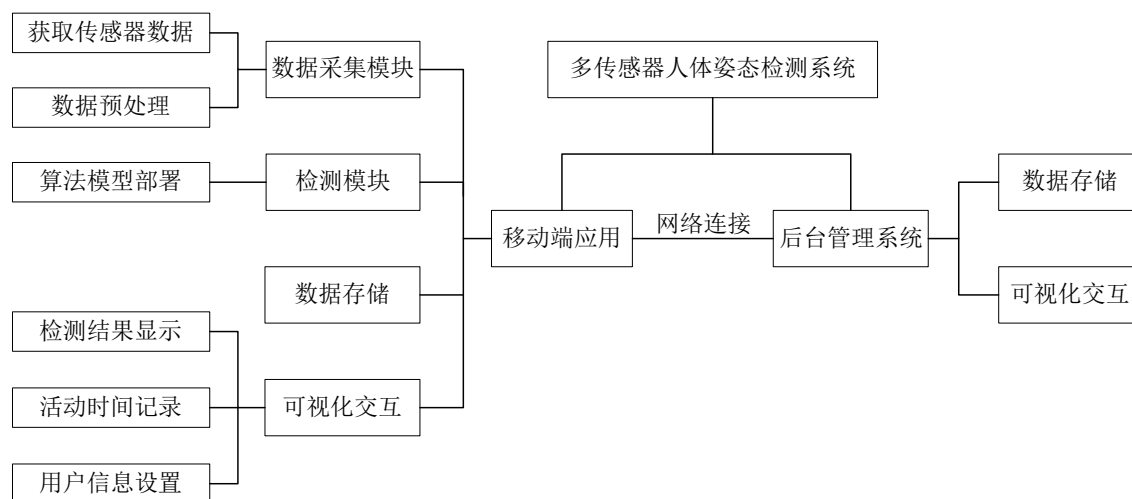


图 5-1 人体姿态检测系统架构

根据上述功能要求，实现的人体姿态检测系统的架构如图 5-1 所示。该系统总共分为两部分，分别是移动端应用和后台管理系统。移动端应用需要实现的模块包括：数据采集模块、检测模块、数据存储、可视化交互以及网络连接模块。在手机 APP 上，通过这些模块实现通过传感器数据检测人体姿态并记录活动数据，通过

网络连接将数据同步到后台管理系统。后台管理系统则需要实现数据存储、可视化交互以及网络连接。后台管理系统通过网络连接获取所有用户的活动数据，保存在数据库中，管理员通过 Web 前端管理用户数据。

5.2 移动端应用实现

5.2.1 神经网络模型的移动端转化与部署

随着近几年人工智能和智能手机的迅速发展，如何在智能手机上快速运行预训练好的深度学习模型成为了新的研究热点。目前市场上现有使用深度学习的应用主要采取如图 5-2 所示的两种方式：

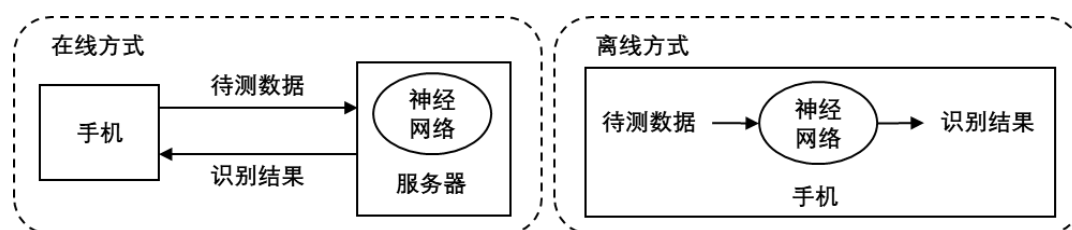


图 5-2 移动端深度学习模式

一种是在线方式，在服务器上部署深度学习模型，当移动端获得数据后只进行预处理，然后把处理好的数据传到服务器上，输入深度学习模型执行预测，最后将预测结果返回给移动端。这种方式的优点是模型在服务器上部署相对简单，主流的框架中都有对应的接口。并且服务器的计算能力远大于移动端，能够处理比较复杂的模型。但缺点就是必须保证网络连接，当移动端无法与服务器连接或者数据传输缓慢时，会无法进行识别或者结果返回延迟很大。

另一种是离线方式，在移动端直接部署深度学习模型，获得数据后直接在本地模型中计算并返回结果。这种方式的优点是不需要网络连接，这样能降低识别延迟。并且所有的数据都保存在本地，保证了用户数据的安全性。但缺点就是会受到移动端计算能力的限制，无法及时运算较大的模型。本文中，基于多传感器数据的深度学习模型的网络规模不大，通过将前两章训练好的模型冻结转化，可以实现在移动端部署计算。

由于离线方式不需要网络连接的巨大优势，国内外主流的机器学习平台都相继推出了移动端深度学习框架。目前主流的移动端深度学习框架包括 Facebook 提出的 Caffe2、苹果提出的 Core ML 和谷歌提出的 TensorFlow Lite。

TensorFlow Lite 是谷歌公司基于 TensorFlow 框架提出的轻量级深度学习框架，旨在移动端用更少的内存更快的运行深度学习模型。主要特性有：轻量级：缩减了

模型和二进制文件的大小；跨平台：支持多种平台包括 Android、IOS 和嵌入式设备，支持多种语言包括 Java、Swift、C++和 Python；快速：减少了模型加载时间并且支持硬件加速。

本文使用 Python 语言开发深度学习模型，需要将训练好的模型移植到 Android 平台，在 Android 平台上使用 Java 语言对模型进行调用。对于这种跨平台跨语言的模型转化需求，TensorFlow Lite 能够轻松实现。并且本文训练模型使用的是 Keras 框架，Keras 框架默认的后端为 TensorFlow，可以将训练好的模型保存为 TensorFlow 格式的模型。TensorFlow Lite 提供的 TensorFlow Lite Converter 可以将 TensorFlow 模型转换为 TensorFlow Lite 模型，使用非常方便。

在模型训练好之后，通过 Saver.save 函数将模型保存到后缀名为 ckpt 的文件中。TensorFlow 会将网络图的结构和参数值分开保存，在 save 这个文件夹中会出现 3 个不同后缀名的文件，分别保存了当前图的结构、参数名和参数值等信息。在进行移植时，这种网络结构和权重数据分开保存的模型不是很方便，需要将这些结合成一个文件，生成 ConvLSTM.pb 文件，这个过程叫冻结模型（freeze_graph）。在冻结模型之前，最好给需要在移动端用到的参数比如输入输出先命名，否则在 Android Studio 中使用时很难获取到这些参数，因为 TensorFlowLite 中的 fetch 函数是根据 pb 模型中保存的参数名来获取参数的。可以使用 tensorboard 可视化工具对 ConvLSTM.pb 模型进行可视化分析，通过 tensorboard 可以看到模型的网络结构，双击各层还可以展开观察各层内部的详细结构。如果之前没有设置好输入输出节点的名字，也可以通过观察模块的输入输出信息获取。

最后通过 TFLiteConverter.from_frozen_graph 工具将 ConvLSTM.pb 模型转化为能在移动设备上使用的 ConvLSTM.tflite 模型。通过 TensorFlow Lite 对模型压缩后，内存占用从原来的 9.3MB 减少到了 3.3MB，同时模型的推理在较小二进制数下进行，大大减少了模型加载和计算时间。至此，基于多传感器数据的 ConvLSTM 模型已经完全准备就绪，可以在移动端部署了。

5.2.2 数据采集模块和检测模块

接下来详细介绍移动端应用的实现过程。移动端应用的基础功能是人体姿态检测，其余功能都是基于姿态检测功能实现的。移动端应用中最核心的模块是数据采集模块和检测模块，它们具体的算法流程如图 5-3 所示。

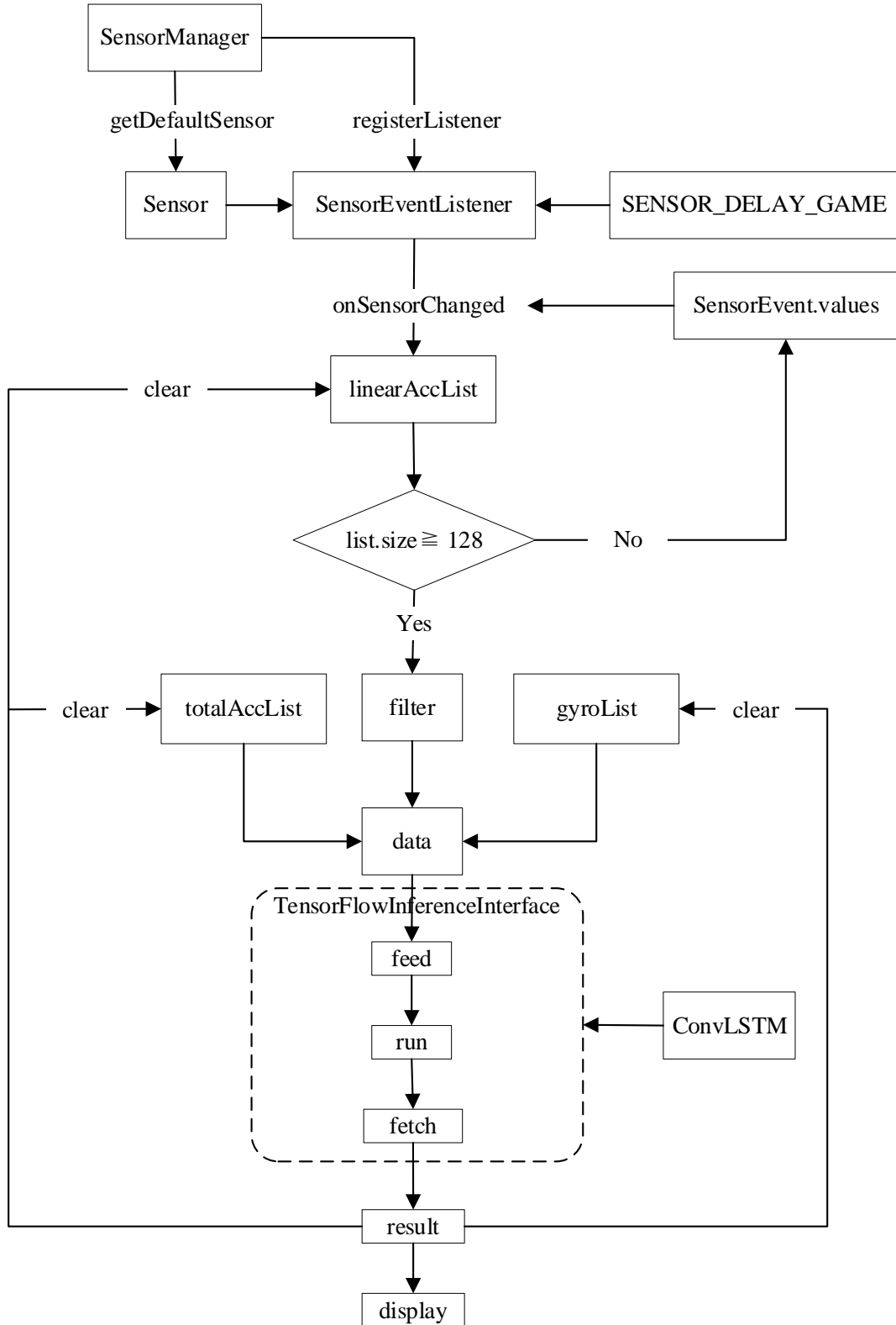


图 5-3 数据采集模块和检测模块流程图

数据采集模块主要功能为采集手机传感器数据并预处理。Android 系统中所有传感器数据的获取方法都类似。传感器数据在 Android 系统中是一项系统服务，首

先要使用 `getSystemService` 方法获得对应的系统服务权限，从而得到传感器管理器（`SensorManager`）对象，`SensorManager` 负责管理 Android 系统的所有传感器。然后调用 `SensorManager` 的 `getDefaultSensor` 方法获取任意类型的传感器（`Sensor`）对象，通过在方法中传入表 2-1 所示的常量来指定传感器类型。然后需要实现传感器事件监听器（`SensorEventListener`）接口来监听传感器信号，每当传感器数据发生变化时就会调用接口中的 `onSensorChanged`（`SensorEvent`）方法，传感器输出的信息都存放在 `SensorEvent` 的 `values` 数组中。最后调用 `SensorManager` 的 `registerListener` 方法来注册 `SensorEventListener` 使其生效，`registerListener` 需要三个参数，第一个是 `SensorEventListener` 对象，第二个是 `Sensor` 对象，在之前步骤已经获得，第三个是传感器的采样频率，设置为 `SENSOR_DELAY_GAME`（50Hz）。

通过上述一系列对象和方法调用就能开始采集手机传感器数据了。当传感器数据发生改变，会自动调用 `onSensorChanged`（`SensorEvent`）方法，`onSensorChanged` 方法将 `SensorEvent.values` 数组中的传感器数据分别缓存到传感器各轴数据的列表（`list`）中，其中 `values[0]`、`values[1]`、`values[2]` 分别对应传感器的 XYZ 轴数据。图中以线性加速度传感器为例，将数据存入 `linearAccList`，合加速度数据与角速度数据也都是一样的获取流程，分别存入 `totalAccList` 和 `gyroList` 中。

同时，在每次调用 `onSensorChanged` 方法时，都判断存储传感器数据的 `list` 的大小（`size`）是否到达 128，如果没有到达 128，则继续将新来的数据存储在 `list` 中；如果到达 128，则对 `list` 中的数据进行滤波。这里的滤波操作与第三章对数据集的滤波操作一致，选用窗口大小为 3 的移动中值滤波器对原始数据进行滤波。然后将所有滤波后的数据按顺序存在 `data` 列表中。

检测模块实现在 `HAR Classifier` 类中，主要用到上一节冻结的移动端神经网络模型 `ConvLSTM.Tflite`。在编写检测模块代码之前，需要先将 `ConvLSTM.Tflite` 模型添加到 Android Studio 的资源文件夹（`Assets Folder`）中，然后在 `gradle` 文件中添加 `TensorFlow Lite` 的依赖，将这些配置都完成之后才能正常调用模型。

在 `HAR Classifier` 类中，首先需要加载 `libtensorflow_inference.so` 库文件，然后初始化 `TensorFlowInferenceInterface` 对象并将模型在 `Assets` 中的存储路径作为参数传入，之后便可以通过调用 `TensorFlowInferenceInterface` 对象的一系列方法来使用模型。首先调用 `feed` 方法将 `data` 列表输入，并将 `data` 数据格式调整为模型要求的输入格式，在这里调整为 `[1, 4, 1, 32, 9]`，除了第一维表示单个样本以外，其余维度的意义均与第四章训练 `ConvLSTM` 模型时一致。然后调用 `run` 方法运行 `ConvLSTM` 模型，最后通过 `fetch` 方法得到预测结果（`result`）。

这样就完成了一次人体姿态检测过程，并将预测结果在用户界面上显示

(display) 出来。然后将所有传感器数据缓存列表清空 (clear), 开始进行下一次检测过程, 以达到持续检测人体姿态的目的。

5.2.3 APP 功能设计



图 5-4 APP 界面。(a)菜单；(b)用户信息；(c)检测结果；(d)活动记录

图 5-4 所示为手机 APP 实现的所有功能以及交互界面, 完成的功能为实时检测人体姿态, 将这些信息记录并显示出来。该应用实现的功能包括: 用户信息注册、人体姿态检测、活动信息记录以及用户数据的本地存储和服务器上传。如图 5-4.(b) 所示, 用户通过手机号码注册账户并填写用户名称、性别、年龄等信息之后, 会在本地 SQLite 数据库中保存这些信息, 同时还会将信息上传至后台服务器存储。如图 5-4.(c)所示, 用户可以通过手机检测人体姿态, 通过检测模块获得当前所有姿态的预测概率, 并将其中概率最高的活动预测为当前姿态标记并显示出来, 并将预测结果保存在本地数据库中。如图 5-4.(d)所示, 还会更新用户在一天的检测活动的时间记录, 并存储在本地数据库中。将用户所有活动的持续时间进行排序, 并从大到小显示, 统计用户静态活动总时长和动态活动总时长, 根据用户的活动时间对用户健康提出建议。

5.3 后台管理平台实现

后台管理平台主要分为网络连接模块、数据存储模块和可视化交互界面三部分。网络连接模块通过 HTTP 协议实现手机 APP 端和服务器数据库的通信。HTTP

协议主要内容为实现客户端(手机 APP)和服务器的请求和应答的标准,通过 HTTP 协议可以实现用户向服务器上传或下载活动数据。

数据存储模块使用关系型数据库 MySQL 对用户数据进行管理。其中存储表项主要包括:用户表和数据表。用户表存储所有用户的个人信息,包括用户的编号、姓名、手机号码、注册时间和登录密码。数据表存储所有用户的活动数据,包括:活动的类别、起止时间戳和持续时间。其中每个用户关联三张数据表,分别为日活动数据表、周活动数据表和月活动数据表。

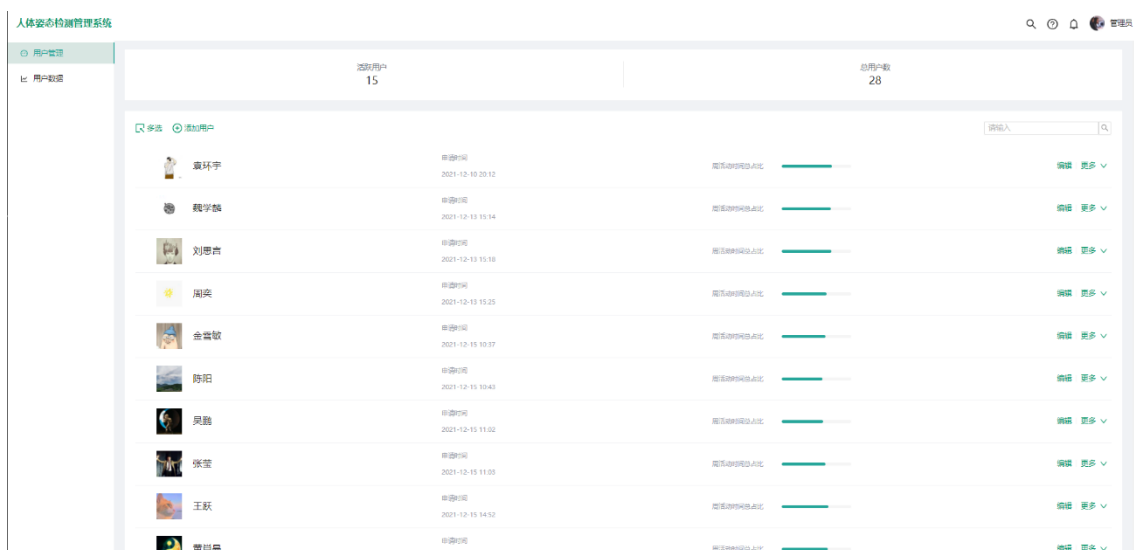


图 5-5 用户管理界面



图 5-6 用户活动数据统计界面

可视化交互界面为 Web 管理界面,后台管理系统的 Web 前端通过 React 框架搭建。React 框架不仅代码逻辑简单,而且性能也很好,是目前最热门的前端框架

之一。Web 前端界面实现的主要目的为方便管理员管理用户，监控用户活动数据。实现的功能包括：用户管理界面和用户活动数据统计界面。用户管理界面如图 5-5 所示，管理员可以在后台进行批量删除、导出、创建用户等操作，通过点击每个用户项可以跳转到该用户的个人活动数据统计界面。用户活动数据统计界面如图 5-6 所示，通过点击日周月按钮切换该用户的对应活动数据，并使用多种统计数据方法直观显示，包括各时段活动分类柱状图和活动占比饼状图。

5.4 实验及结果分析

为了测试本文实现的基于多传感器数据的移动端人体姿态检测模型的表现，邀请了 10 名同学作为志愿者，在他们的手机上安装了人体姿态检测 APP 进行实验，实验分为两组。

实验一：10 名志愿者打开 APP 放入裤子口袋中，开始检测人体姿态，依次对 APP 中检测的六种人体姿态进行实验，每次实验过程要求志愿者保持同一种姿态，持续 150 秒，窗口大小为 128，理论上应该有 58 次预测结果。然后取出志愿者手机中的数据库文件，获得所有的预测结果，得到的结果数在 56-58 之间，考虑到人工误差和传感器数据缺失，模型的运算满足实时性检测的要求。处理每次实验结果时，为了排除志愿者手机放入取出口袋的影响，去除开始后和结束前一段时间的预测结果，只保留中间 128 秒的预测结果，统计所有检测结果得到如表 5-1 所示混淆矩阵。

表 5-1 实验一检测结果混淆矩阵

真实值 \ 预测值	站立	坐下	跑步	行走	上楼	下楼
站立	500	0	0	0	0	0
坐下	50	450	0	0	0	0
跑步	0	0	465	22	8	5
行走	3	0	7	447	19	24
上楼	1	10	5	11	448	25
下楼	4	0	12	13	35	436

其中，坐下姿态预测成站立姿态的样本为同一志愿者的实验结果，对该名志愿者进行重复实验皆获得同样结果，可以排除实验过程因素。分析产生此结果的原因，可能是志愿者的身高与座椅高度不匹配，导致大腿倾角过大，以及手机与大腿摆放的相对位置，使模型将结果预测为站立。

从实验一的检测结果混淆矩阵可以得出，移动端部署的 ConvLSTM 模型在检测持续性活动时，预测准确率为 91.533%。并且该模型能够很好地区分静态活动（站立、坐下）和动态活动（跑步、行走、上楼、下楼），二分类准确率达到 99.4%。对于静态活动的预测非常准确，但对于动态活动的预测准确率并没有表现得那么好，存在跑步姿态预测为行走姿态，行走姿态预测为上下楼，以及上下楼互相预测等主要错误预测情况。

实验二：10 名志愿者打开 APP 放入裤子口袋中，开始检测人体姿态后，可以在六种姿态中任意切换自由活动，但尽量均匀覆盖每一项活动，活动时间 150 秒。实验过程中，采用录像的方式记录活动真实值，保留中间 128 秒的实验结果，与数据库中的检测结果比较，得到如表 5-2 所示混淆矩阵。

表 5-2 实验二检测结果混淆矩阵

真实值 \ 预测值	站立	坐下	跑步	行走	上楼	下楼
站立	417	0	0	16	3	1
坐下	66	553	0	3	1	0
跑步	0	0	287	45	17	8
行走	19	0	67	616	33	21
上楼	3	15	2	29	389	31
下楼	9	0	5	21	29	294

从实验二的检测结果混淆矩阵可以得出，在加入日常活动的随机性之后，模型检测的准确率会下降，检测准确率为 85.2%。对于静态活动和动态活动的区分效果依然很优秀，二分类准确率为 97.667%。

模型在移动端部署后，准确率下降的原因在于：人体姿态检测以窗口的形式进行，每 2.56 秒检测一次，日常活动切换的过程中，会发生一个完整活动被相邻两个检测窗口分开，或者一个检测窗口内包含两种活动的情况；数据集制作的环境固定，而日常应用中的环境多变；日常应用中用户使用手机型号多样，与采集数据的手机型号不同，传感器的精度也会有区别。在实际应用中，以上这些因素都会导致模型的准确率下降。

经实验证明，本文提出的基于多传感器数据的 ConvLSTM 人体姿态检测模型，部署在移动端之后，对于持续性活动的检测准确率为 91.533%，对于随机性活动的检测准确率为 85.2%。考虑到移动端设备的影响以及日常应用中的一些随机性，该模型的表现非常优秀。

5.5 本章小结

本章首先介绍了基于多传感器数据的人体姿态检测系统的主要功能。然后介绍了其具体实现过程，包括如何将训练好的 ConvLSTM 模型转化为可以在移动端部署的文件，人体姿态检测手机 APP 和后台管理系统。最后通过实验证明，虽然模型部署在移动端后，准确率会有所下降，但其在日常活动中的检测效果尤其是二分类问题上，表现非常优秀。

第六章 总结与展望

6.1 总结

本文研究了基于移动终端多传感器数据的人体姿态检测技术。本文研读整理了基于传感器数据的人体姿态检测相关文献发现，传统方法多用人工提取数据的特征向量，将特征向量输入到机器学习模型训练的方法来进行分类。随着深度学习的发展，研究者提出了很多自动提取特征的深度学习模型，本文基于经典的神经网络算法，提出了两种处理多传感器数据的神经网络模型，均比传统方法的分类准确率更高。此外，还将实现的神经网络模型部署在移动端，以此为基础搭建了一套人体姿态检测系统。本文的主要工作如下：

(1) 本文提出了一种基于多通道卷积神经网络 (Multi-channel CNN) 的人体姿态检测模型。首先对使用一维卷积神经网络提取传感器数据特征，并对人体姿态分类进行深入研究。对使用一维卷积提取时间序列特征的方法，卷积神经网络的网络结构、损失函数以及超参数的选择等对人体姿态检测准确率的影响分别进行了研究，说明了卷积神经网络在处理传感器数据上的可行性。然后基于独立提取不同传感器数据特征的想法，提出了一种多通道卷积神经网络。该网络可以使用多个参数设置不同的卷积头并行提取特征，然后将特征融合解释并做出预测。该网络性能表现优秀，在同一个数据集上检测准确率高于传统机器学习方法。

(2) 本文提出了一种基于卷积长短时记忆网络 (ConvLSTM) 的人体姿态检测模型。首先对擅长处理具有时间依赖性数据的长短时记忆网络进行深入研究，说明了长短时记忆网络能处理长时间依赖数据的原因，并对长短时记忆网络的网络结构、损失函数以及超参数的选择分别进行了研究。想将卷积神经网络和长短时记忆网络在提取特征上的优势结合起来，同时提取传感器数据的时空特征。提出了 ConvLSTM 网络，将 LSTM 网络门控单元的矩阵乘法换成卷积运算，取得优秀的预测效果，检测准确率高于 Multi-channel CNN 网络。

(3) 本文设计并实现了基于手机传感器数据的人体姿态检测系统。将训练好的 ConvLSTM 模型冻结为可移植文件，部署到移动端应用中，实现人体姿态检测功能。基于这一基础功能构建了人体姿态检测系统，本系统包括移动端 APP 和后台管理系统。在移动端 APP 主要设计实现了数据采集模块、检测模块、数据存储模块以及可视化交互界面，后台管理系统主要设计实现了数据存储模块和 Web 前端界面。本系统可以有效的检测人体姿态并生成用户每日活动记录，可以帮助用户进行健康监控和锻炼提醒。

6.2 展望

使用手机传感器数据进行人体姿态检测受到越来越多人的关注，有着诸多应用场景以及巨大的商业价值。尽管本文提出的人体姿态检测系统有较好的分类性能，但由于本人能力和时间的限制，研究中仍存在许多不足，主要包括以下几个方面：

(1) 本文制作数据集的过程中，由于人力和时间限制，采集数据的志愿者人数较少，不能完全排除个体差异对模型性能的影响；采集活动的种类也比较基础，只包含一些日常活动；采集的数据量也比较小，每名志愿者的采集时间在一个小时左右。下一阶段的研究工作中，可以改善的数据集制作过程，增加实验人数，增加采集时间，使数据集更加丰富和完善。同时采集一些其他类别的活动，如其他的日常活动如骑车，以及一些异常活动如摔倒，丰富模型的分类性能。

(2) 本文获得的人体姿态检测模型能够很好的区分静态活动和动态活动，有优秀的二分类性能。但是对静态活动和动态活动各类之间的分类准确率需要进一步提升，尤其是在日常活动活动频繁切换，模型检测准确率还会略微下降。下一阶段的研究工作中，可以尝试更复杂的神经网络模型，或增加一些活动之间的过渡分类，以提升模型在日常应用中的检测准确率。

(3) 本文设计实现的人体姿态检测系统目前功能比较单一，仅包含姿态检测、活动记录、运动提醒以及用户管理等功能，起到健康监控的作用。后续可以继续开发，丰富系统功能，加入摔倒检测、摔倒报警以及实时定位等功能，来保障伤者及时获得救治。

致 谢

行文至此，百感交集。成电七年，转眼一瞬，2015年初入电子科技大学犹如昨日之事。在这七年的时光里，我从一个少不更事的高中生成长为了一个能够独立思考、自主学习的硕士研究生。在此，我由衷地感谢成长路上所有对我提供过帮助的老师、同学和朋友。

首先我要感谢我的导师武畅副教授。在武老师的悉心教导下，我从一个科研小白逐渐找到了自己的科研方向，使我在科研路上少走了很多弯路。武老师治学严谨、学贯中西、风趣幽默，和我们学生相处融洽、亦师亦友，无论是在学习还是生活中遇到挫折，武老师总会给出中肯的建议。武老师常说：我们要成为自己想成为的人，帮助我树立独立自主、处变不惊、辩证理性的处世观，让我受益良多。

感谢教研室的游远、吴鹏、陈阳和金雪敏，在三年同窗的时间里我们互相鞭策，共同进步。感谢课题组的俞浩然、魏学麟、刘思言等同学，在我毕业论文工作中提供了帮助。感谢张康、杨帆、李杰、阿天仁几位师兄，在我择业就业过程中提供了宝贵的经验和建议。感谢教研室的所有同学，和你们的相处让我的研究生生活更加多姿多彩，祝大家都能顺利毕业。

感谢我的父母，自从上高中以来，我们聚少离多，可是你们对我的关心却有增无减。无论是在精神上还是物质上，每当我需要帮助时，你们总是倾囊相授，不计回报，你们永远是最坚实的后盾。感谢你们二十多年来对我的养育栽培。

最后我要感谢各位评审老师和答辩老师，谢谢你们百忙之中抽空评阅我的论文，提出宝贵的修改意见。

参考文献

- [1] 徐越.人体活动识别研究现状综述[J].电脑知识与技术,2020,16(14):221-222.
- [2] 张林丛,张策,敖然.可穿戴无线传感器网络对医疗健康的影响[J].电子技术与软件工程,2019(09):25-26.
- [3] 苏炳添,李健良,徐慧华,等.科学训练辅助:柔性可穿戴传感器运动监测应用[J].中国科学:信息科学,2022,52(01):54-74.
- [4] 童玉芬.中国人口的最新动态与趋势——结合第七次全国人口普查数据的分析[J].中国劳动关系学院学报,2021,35(04):15-25.
- [5] 侯逸仙.老年人摔倒检测与预警系统的设计与实现[D].哈尔滨工业大学,2017.
- [6] Ann O C, Theng L B. Human activity recognition: a review[C]. 2014 IEEE international conference on control system, computing and engineering, Penang, Malaysia, 2014: 389-393.
- [7] Wang Y, Cang S, Yu H. A survey on wearable sensor modality centred human activity recognition in health care[J]. Expert Systems with Applications, 2019, 137: 167-190.
- [8] Demrozi F, Pravadelli G, Bihorac A, et al. Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey[J]. IEEE Access, 2020, 8: 210816-210836.
- [9] 何理.基于加速度传感器的人体摔倒检测系统设计[D].重庆大学,2016.
- [10] 罗薇.基于加速度传感器的人体动作识别方法研究[D].天津大学,2017.
- [11] 冯增涛.基于集成学习的人体日常行为活动识别系统研究[D].东南大学,2016.
- [12] 熊富海.基于卷积神经网络和循环神经网络的活动识别研究[D].厦门大学,2018.
- [13] Kwapisz J R, Weiss G M, Moore S A. Activity recognition using cell phone accelerometers[J]. ACM SigKDD Explorations Newsletter, 2011, 12(2): 74-82.
- [14] Anguita D, Ghio A, Oneto L, et al. A public domain dataset for human activity recognition using smartphones[C]. Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning, Bruges, Belgium, 2013: 437-442.
- [15] Anguita D, Ghio A, Oneto L, et al. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine[C]. InInternational workshop on ambient assisted living, Berlin, Heidelberg, 2012: 216-223.
- [16] Ahmed M, Mehmood N, Nadeem A, et al. Fall detection system for the elderly based on the classification of shimmer sensor prototype data[J]. Healthcare informatics research, 2017, 23(3): 147-158.

- [17] Kim T, Vecchietti L F, Choi K, et al. Machine learning for advanced wireless sensor networks: a review[J]. *IEEE Sensors Journal*, 2020, 21(11):12379-12397.
- [18] 杨佳现. 基于智能手机多传感器融合技术的人体活动识别研究[D].北京工业大学,2020.
- [19] You Y, Wu C. Hybrid indoor positioning system for pedestrians with swinging arms based on smartphone IMU and RSSI of BLE[J]. *IEEE Transactions on Instrumentation and Measurement*, 2021, 70:1-15.
- [20] Samuel A L. Some studies in machine learning using the game of checkers[J]. *IBM Journal of research and development*, 2000, 44(1. 2): 206-226.
- [21] Sathya R, Abraham A. Comparison of supervised and unsupervised learning algorithms for pattern classification[J]. *International Journal of Advanced Research in Artificial Intelligence*, 2013, 2(2): 34-38.
- [22] Bulling A, Blanke U, Schiele B. A tutorial on human activity recognition using body-worn inertial sensors[J]. *ACM Computing Surveys (CSUR)*, 2014, 46(3):1-33.
- [23] Cortes C, Vapnik V. Support-vector networks[J]. *Machine learning*, 1995, 20(3): 273-297.
- [24] Hsu C W, Lin C J. A comparison of methods for multiclass support vector machines[J]. *IEEE transactions on Neural Networks*, 2002, 13(2): 415-425.
- [25] Rish I. An empirical study of the naive Bayes classifier[J]. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 2001, 3(22): 41-46.
- [26] Safavian S R, Landgrebe D. A survey of decision tree classifier methodology[J]. *IEEE transactions on systems, man, and cybernetics*, 1991, 21(3): 660-674.
- [27] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016, 73-83.
- [28] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. *nature*, 2015, 521(7553):436-444.
- [29] Sainath T N, Vinyals O, Senior A, et al. Convolutional, long short-term memory, fully connected deep neural networks[C]. *2015 IEEE international conference on acoustics, speech and signal processing, Brisbane, Australia, 2015: 4580-4584.*
- [30] Sharma S, Sharma S, Athaiya A. Activation functions in neural networks[J]. *towards data science*, 2017, 6(12): 310-316.
- [31] Ronao C A, Cho S B. Human activity recognition with smartphone sensors using deep learning neural networks[J]. *Expert systems with applications*, 2016, 59: 235-244.
- [32] Albawi S, Mohammed T A, Al-Zawi S. Understanding of a convolutional neural network[C]. *2017 international conference on engineering and technology, Antalya, Turkey, 2017: 1-6.*
- [33] Cho H, Sang Y. Divide and Conquer-Based 1D CNN Human Activity Recognition Using Test Data Sharpening[J]. *Sensors*, 2018, 18(4):1055.

- [34] Murad A, Pyun J Y. Deep recurrent neural networks for human activity recognition[J]. *Sensors*, 2017, 17(11): 2556.
- [35] Bengio, Y, Simard, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. *Neural Networks IEEE Transactions on neural networks*, 1994, 5(2):157-166.
- [36] Olah C. Understanding lstm networks[OL]. 2015. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [37] Van der Maaten L, Hinton G. Visualizing data using t-SNE[J]. *Journal of machine learning research*, 2008, 9(11).
- [38] 张斐,刘志杰.加速度传感器信号数据处理中滤波算法的应用[J].*电脑与信息技术*,2018,26(03):1-4.
- [39] Hsia S C , Wang S H , Chang C Y . Convolution neural network with low operation FLOPS and high accuracy for image recognition[J]. *Journal of Real-Time Image Processing*, 2021, 18(4):1309-1319.
- [40] Feng Y , Cheng Y . Short Text Sentiment Analysis Based on Multi-Channel CNN With Multi-head Attention Mechanism[J]. *IEEE Access*, 2021, 9: 19854-19863.
- [41] 罗威. 长时记忆循环神经网络算法研究[D].浙江大学,2020.
- [42] Hochreiter S, Schmidhuber J. Long short-term memory[J]. *Neural computation*, 1997, 9(8):1735-1780.
- [43] Chen Y, Zhong K, Zhang J, et al. LSTM networks for mobile human activity recognition[C]. *2016 International Conference on Artificial Intelligence: Technologies and Applications*, Bangkok, Thailand, 2016: 24-25.
- [44] Mutegeki R, Han D S. A CNN-LSTM approach to human activity recognition[C]. *2020 International Conference on Artificial Intelligence in Information and Communication*, Fukuoka, Japan, 2020: 362-366.
- [45] Shi X, Chen Z, Wang H, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting[J]. *Advances in neural information processing systems*, 2015, 28.
- [46] Ronao C A, Cho S B. Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models[C]. *2014 10th international conference on natural computation*, 2014: 681-686.
- [47] Yu S, Qin L. Human activity recognition with smartphone inertial sensors using bidir-lstm networks[C]. *2018 3rd international conference on mechanical, control and computer engineering*, 2018: 219-224.

攻读硕士学位期间取得的成果

1. 发表的论文和专利

- [1] 武畅, 袁环宇, 袁翼飞, 孔孝童, 陈阳, 杨森林. 一种人体姿态多通道卷积神经网络检测方法: 202210518089.5[P]. 2022年5月12日.
- [2] 武畅, 袁环宇, 游远, 魏学麟, 金雪敏, 陈阳. 一种针对摆臂行人的行人航位推算方法: 202110296433.6[P]. 2021年3月19日.
- [3] 武畅, 游远, 袁环宇, 金雪敏, 杨帆, 吴鹏. 一种基于蜂窝区域限制辅助定位的方法: 110572777B[P]. 2020年7月21日.
- [4] 武畅, 游远, 金雪敏, 袁环宇, 杨帆, 陈阳. 基于离群点剥离的中值卡尔曼滤波的定位方法: 202010193137.9[P]. 2020年3月18日.
- [5] 武畅, 游远, 杨帆, 金雪敏, 袁环宇, 陈阳. 一种基于泰勒展开的无线传播模型: 202010191954.0[P]. 2020年3月18日.
- [6] 武畅, 由俊威, 阿天仁, 高璇, 吴鹏, 袁环宇. 一种基于携能通信技术的环境感知系统: 202010266121.6[P]. 2020年4月7日.
- [7] 武畅, 杨帆, 张莹, 王跃, 游远, 吴鹏, 金雪敏, 袁环宇. 一种基于水晶型信标布局的改进位置指纹定位方法: 113242518B[P]. 2022年4月29日.

2. 其他获奖

电子科技大学 2019-2020 学年研究生学业一等奖学金

电子科技大学 2020-2021 学年研究生学业一等奖学金

电子科技大学 2020-2021 学年“优秀研究生干部”

电子科技大学信息与通信党委 2020-2021 年“优秀共产党员”