湖北中医药大学

Hubei University of Chinese Medicine

毕业论文（设计）

题　　目：基于 LSTM 人体活动识别研究

姓　　名：程亮

指导老师：肖瑞

学　　号：20160702036

专　　业：医学信息工程

年　　级：2016级

实习单位：

完成日期：2020年　05月　日

摘　要

本文对人体活动识别进行了研究，首先研究了传感器采集过程，然后介绍了机器学习需要的背景知识，然后

关键词:活动识别、机器学习、长短期记忆网络

目 录

第 1 章　引言

1.1 研究背景与研究意义

1996 年 Weiser, Mark 在一次演讲 1 中第一次提出了普适计算(Ubiquitous computing)的概念，普适计算是让身边的任何设备在任何时间和任何地点能够人类提供计算能力，而如何让在具体环境中的计算机准确快速的理解人类行为和动作来实现计算机与人类无缝衔接的互动成为其中的关键。人体活动识别是一项与计算机视觉和人机交互相关的研究领域 2,3，4 在传统的研究中，人体活动识别研究对象主要分为图像视觉，深度传感器，可穿戴设备 5。最近出现了基于无线网络的人体活动识别方面的研究。人体活动识别在日常生活中主要用与监控，卫生保健，人机交互。在监控技术中，使用人体活动识别技术辅助监控系统可以部署在银行，机场等公共场所 6,7,8。Ryoo 等人提出了一种可以在公共场所提前预防犯罪行为和危险活动的新方法 9。通过调查结果，他们证实了使用此方法能够在活动早期预测人群的活动。在卫生保健领域，人体活动识别被应用于医疗系统和康复中心。例如在慢性疾病管理通过监控老年人活动情况做到疾病预防 10。在家中使用基于人体活动识别的监测系统估计患者的能源消耗，提醒患者面临肥胖的风险 11。在人机交互方面，人体活动识别被应用在体感游戏操控中，例如微软 Kinect 控制器，任天堂 Switch 体感功能。

1.2 国内外研究现状

1.2.1 人体活动识别研究现状

有关与人体活动识别的研究最早可以上个世纪 80 年代，按照研究对象的不同，人体活动识别可以分为单用户活动识别，多用户活动识别和群体活动识别。在 1991 年，ROY WANT 等人提出一种徽章定位系统 12。通过接收端不断接收到员工佩戴的徽章中的传感器发送的信息实现员工的定位，实现了多用户活动识别。2005 年 Nishkam Ravi 等人使用多种分类器对用户活动产生的加速度数据进行了分析 13。2013 年，Dawud Gordon 等人在他们的文章中提出了一种

使用移动设备传感器对一个协作小组的活动进行分布式识别的方法。按照人体活动识别使用的不同方法来分，可分为基于逻辑推理，概率推理，数据挖掘三种基本方法。基于逻辑的方法会连续观察所有动作逻辑上的一致性，1987 年 Kautz 在他的博士论文中提出一种以事件逻辑化开始 14。概率论和统计学习模型是另外一种应用于人体活动识别的方法，适应于对不确定性动作的预测问题 15。时间概率模型在人体活动识别中有广泛的应用，例如隐马尔可夫模型（HMM），动态贝叶斯，条件随机场（CRF）它们都是常见的选择。常规的时间概率模型，一般是直接对活动与产生传感器数据之间的相关性进行建模。近年来，人们开始尝试使用层次模型，这种模型考虑了人体活动数据中存在的丰富层次结构。与直接对活动与数据之间的关系不同的是，先将活动分为更小的子活动，然后对一些基础的活动与数据之间进行建模。例如炒菜的活动可以分解为切菜，炒菜，盛菜装盘等子活动。这个示例构建的是分层隐马尔可夫模型，相比传统非分层模型，这些模型具有相对的优势。现在，基于数据挖掘方法的人体活动识别开始流行，相比以前的方法，数据挖掘更注重发现数据之间深层次之间的关系。2018 年王晋东等人在他们的文章中讨论了基于深度学习的人体活动识别为何能改进性能等问题 16。按照传感器数据来源不同，主要有基于图像传感器，运动传感器数据等，最近出现了一些基于 Wifi 室内空间活动识别的新方法。2017 年，Shugang Zhang 等人对基于视觉图像的人体活动识别技术进行了总结 17。2019 年 Li He Ju 等人提出了 Wi-Motion，这是一种基于 WiFi 的人类活动识别系统，与现有系统不同的是，这种系统联合利用了信道状态信息（CSI）中提取的幅度和相位信息 18。

1.2.2 深度学习研究现状

深度学习的历史最早可以追溯到 20 时间 40 年代。在学术界一般认为深度学习经历了三次发展浪潮：20 世纪 40 年代至 60 年代，深度学习的基本概念出现在控制论中，Rosenblatt 在 1958 年在单个神经元上进行训练。1986 年 Rumelhart 通过反向传播来训练具有较少隐藏层的的神经网络；到 2006 年以后，深度学习又开始复兴。图 1 展示了深度学习发展的曲线图。在机器学习领域，用来处理序列数据的一类神经网络模型通常被称为循环神经网络。与一般

的神经网络模型相比，循环网络在它的模型各个部分之间共享参数，这使得模型能够处理输入数据之间存在位置信息。

1.3 机器学习安全与隐私问题

人工智能在早期的发展中主要偏向于研究具体实现过程，而对于安全性方面的研究相对较少。而近年来产生于实验室的人工智能开始应用于工业，商业等等实际领域，对于人工智能应用过程中产生的隐私问题逐渐浮现。现在，不少应用于特定领域的机器学习算法与模型在处理自然发生的输入时，它的表现效果超过了人类。但是，早期在设计机器学习系统时的学者，为保证设计系统达到是安全和可以依赖的预期效果，通常会提出企图攻击机器学习系统出现的攻击人的攻击强度和攻击对象的假设，即所谓的威胁模型。当前，大多数的针对的威胁模型设计实现的机器学习模型处理威胁的能力较弱，没有过多的考虑攻击者。尽管在面对自然预期的输入时，这些模型被设计有非常完美的表现，但在现实环境中，这些机器学习模型可能会遇到大量的恶意用户甚至是攻击者。针对机器学习的模型攻击可能会影响机器学习模型的机密性，完整性和可用性。

保密性攻击：保密性又被称为隐私性，就是要求禁止模型泄露敏感数据。在文献[2]中的研究指出，有时机器学习平台是安全可靠的，但机器学习模型提供者可能提供错误的算法。当作非专业人员或者忽视风险数据持有者使用机器学习即服务这种模式训练自己的预测模型时，可能会选用到由攻击者精心构造的恶意模型，在这种模型中，攻击方有机会将数据持有者的个人信息嵌入到模型参数中，最后可能通过分析模型参数信息来窃取用户的隐私信息。

可用性攻击：机器学习模型的正常运行也可能受到攻击。例如，前不久，一名来自德国的艺术家在街上用手拖车拉着 99 部手机 走到谷歌柏林办公室的时候，被谷歌地图错误的判断在公司门口出现堵车。

完整性攻击：机器学习模型有可能受到攻击者的篡改，通常攻击会发生在模型的学习阶段和模型的推理预测阶段。一旦模型受到攻击者的篡改，模型的预测结果有极大概率偏离预期。在模型的学习阶段对机器学习模型的训练过程进行干扰，一般的攻击策略是当用户于生产时让机器学习模型出现更多的

错误，在该阶段，最常见的攻击就是数据污染攻击[4]，攻击方可以通过修改现有的训练集或者增加额外的恶意数据，影响模型的训练过程，破坏模型的完整性从而达到降低模型在预测推理阶段准确性的目的。

1.4 本文的研究内容

1.5 论文组织结构

第 2 章　人体活动识别技术概述

2.1 人体活动识别基基础

在人体活动识别研究领域，目前存在三种不同的研究方向。其中利用运动传感器的人体活动识别具体来说，是使用可穿戴设备或者智能手机传感器数据对用户活动进行预测。通常具有部署灵活，数据采集方便，采集成本低等优势。在实际使用中需要检测对象亲自佩戴，推广难度大，不适合对群体对象的检测。生活中通常应用在个人健康管理，运动记录等方面。基于图像视觉传感器的人体活动识别则是研究基于图像传感器数据对检测对象的行为识别预测等方法的技术。通常基于图像视觉的人体活动识别具有非侵入的优点，识别对象通常不易察觉，对识别对象的影响最小。但同时具有隐私保护难度大，数据量大等缺点。通常适合对固定地点人群的监控，生活中常见于部署在商场，机场等人流量较大的地区。最近几年，不断出现与无线网络有关的活动识别的研究，利用无线网络的人体活动识别技术同时具有非侵入式和隐私保护以及数据量小等优点，但识别精度较低，无法进行细粒度的人体活动识别。因此，这种技术通常难以单独使用，而更多用于人体活动识别的融合识别，辅助识别等方面。

特点	基于图像视觉	基于运动传感器	基于无线网络
部署难度	大	小	小
检测范围	小	大	小
环境干扰	影响大	影响小	影响较小
监测方式	非侵入	侵入	非侵入
数据量	大	小	小

检测对象数	多个	单个	多个
识别效果	好	好	差

通常人体活动识别研究的过程如图，主要的过程包括数据采集，数据预处理，特征选择，构建模型，训练模型，模型评估等。通常人体活动识别的数据有图像数据，传感器运动数据，无线网络的 RSSI，CSI 信息数据等。数据在采集过程中可能由于各种原因产生了噪声，根据数据类型的不同，我们需要采用不同的滤波器进行数据降噪，并且对于均有时序序列的信息，我们需要以滑动窗口的方式对数据进行取样。在对数据预处理之后，我们需要根据将采用的模型进行特征处理，对于基于神经网络以外的模型，我们通常需要进行数据降维，主成分分析等特征处理。因为神经网络的层次结构的特征，特征提取的过程被集成与模型训练中。对于通过数据训练出来的模型，通常需要对模型进行评估，具体方法有留出法，交叉验证，k 折验证法等。通常数据集按照一定比例被分为训练集和验证集，但是对于使用神经网络的模型，我们通过调节神经网络的参数降低模型在训练集上的泛化误差，但是泛化误差低不代表泛化能力好，我们需要在模型训练之前提前分出一部分作为验证集，使用验证集评价神经网络模型泛化能力更科学。

2.2 带时序序列的运动传感器数据的人体活动识别

机器学习算法与模型的研究工作需要在具体的数据的基础上展开，因此我们需要一个有记录的数据收集过程。本节将介绍数据采集的具体流程，利用智能手机设备中携带的三轴加速度计和三轴陀螺仪传感器在有记录活动标签的情况下收集被实验对象的活动产生的数据。对于一次实验对象产生的数据，根据传感器采样频率，将开始采集数据的时候作为起点，为数据加入时间维度。通常这样的收集的数据还需要校准数据漂移的情况，而且对原始数据中存在的噪声，我们需要采取合理的算法对数据进行降噪处理。最后处理后的数据在时间维度上依然是按照原始数据中特定采样频率的单次采样为时间节点，无法直接应用于循环神经网络的模型训练。我们需要将一段以一定长度的采样时间点下的数据集作为循环神经网络中用于训练的单个样本。通常，我们使用滑动窗口的方法将数据处理为用于训练模型的数据。

2.2.1 数据采集

对于用来研究模型的数据集，我们需要简化数据采集是的具体活动场景，在数据采集实验中尽量进行一般具有代表性的活动。为了尽量模拟真实活动情况中产生的数据，在该实验中将按照如图所示的三个相互垂直的面向外面的线性加速度方向为正和三个相互垂直并且按照右手大拇指旁边的四指方向为正进行数据收集。

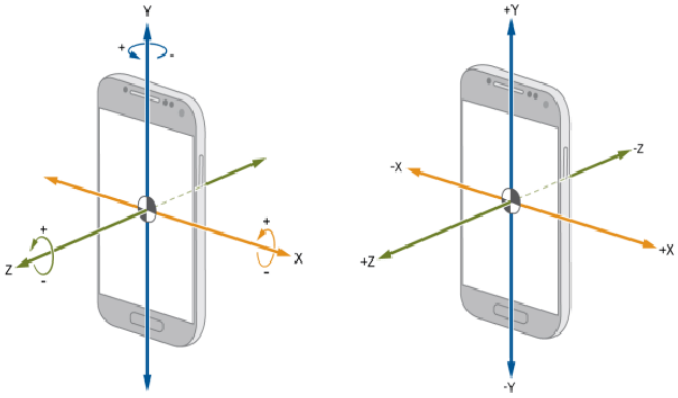

图 2-1 智能手机加速度和角速度方向

本实验中的数据是由一组 30 位 19 到 48 岁年龄段的志愿者产生的。如图为某一志愿者正在进行静座的活动，在该志愿者的腰部绑定有 Samsung Galaxy S II 用来采集数据。

图 2-2 正在采集志愿者的数据

在本次实验中，所有志愿者在腰部佩戴了智能手机，如图 1 每一位志愿者分别进行了六项基本的活动：

图 2-3 数据采集中进行的六种活动

在实验中我们记录了对应于上述活动中对应的 6 中活动标签，如图：

活动标签	活动名称
WALKING	步行
WALKING_UPSTAIRS	上楼梯
WALKING_DOWNSTAIRS	下楼梯
SITTING	静坐
STANDING	站着
LAYING	平躺

在实验中我们 50Hz 的频率对传感器进行数据采样。

2.2.2 校准数据

在数据采集的实际过程当中，智能手机传感器产生的数据可能存在着数据移位的问题，通常在静置状态下，数据中的时间序列存在有轻微的摆动，数据偏移现象可能造成传感器所采集的时间序列值不是真实的数值，一般我们可以数据校准来处理偏移量问题。有时候，传感器在测量的过程中还夹杂着热噪声等干扰实际数据信号的噪声信号，为此，我们需要采用合适的滤波器除噪。

以下两个方面可能是导致传感器数据漂移的出现的原因：一方面，受到传感器设备制造工业水平的限制，传感器导出的数据在被实验的志愿者身体的轻微晃动后容易出现数据漂移；另一方面，在把智能手机安装在每一位志愿者身上是的方面可能不完全相同，安装的方向也容易引起加速度和角速度的数据漂移。

2.2.3 数据除噪

在处理完传感器采集的时序序列数据时发生的数据移位问题后，噪声依然存在，需要更进一步采用滤波去除噪声的方式进行处理。如图，选取在滤波除噪之前的一段 x 轴方向的随着时间变化的加速度的折线图。仔细观察加速时间序列的变化曲线，可以发现在曲线变化的峰谷中仍然存在毛刺噪声。

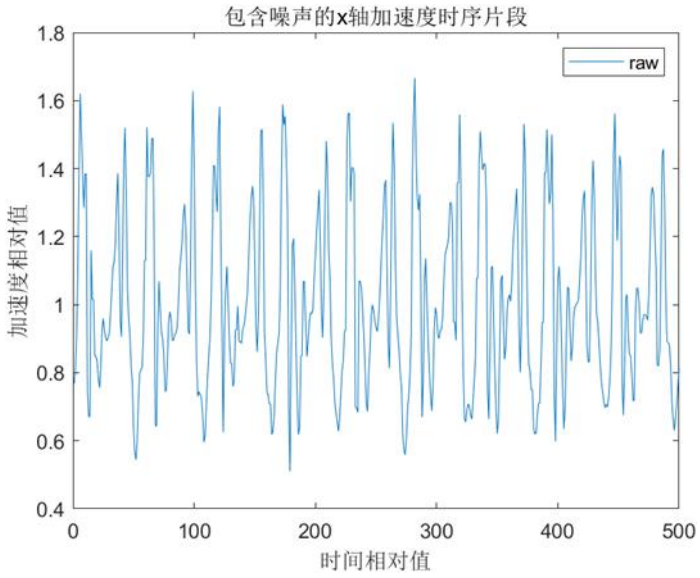

图 2-4 包含噪声的 x 轴加速度时序片段

对于传感器在实际环境中收集数据时受到环境噪声的影响而产生的噪声，可以使用低通滤波，均值滤波，高斯滤波等方法进行处理。本次实验中产生的时间序列数据中主要包含了高频噪声，因此需要采用低通滤波器来过滤掉时序序列中的高频噪声信息。我们使用 matlab 设置一个截止频率为 3Hz 巴特沃斯低通滤波器。滤波器的相位和幅度随标准化频率的图像如下。

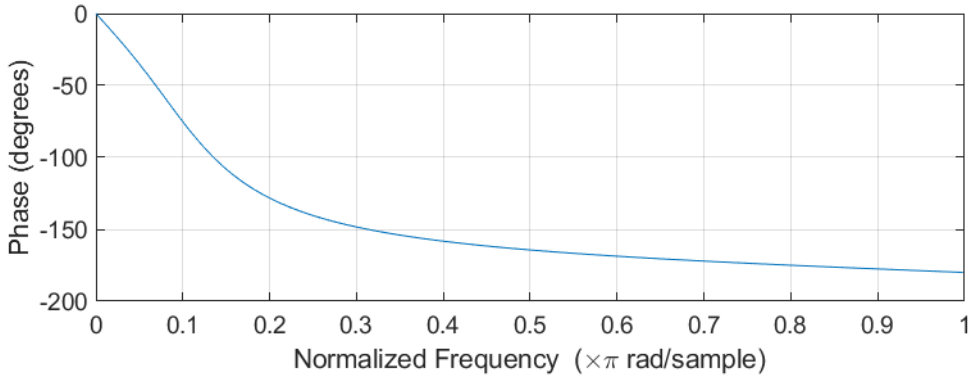

图 2-5 相位随频率变化图

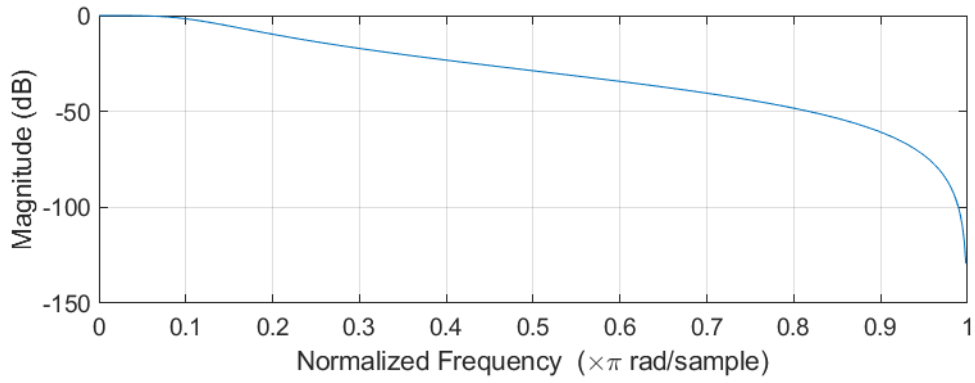

图 2-6 幅度随着频率变化图

从图 的 raw 和 lowpass filter 曲线可以看出，经过二阶巴特沃斯低通滤波器后的数据，相比未被处理之前，时序片段的数据波形更加平滑。在二阶巴特沃斯低通滤波器处理后，时序片段中的毛刺噪声得到了有效处理。

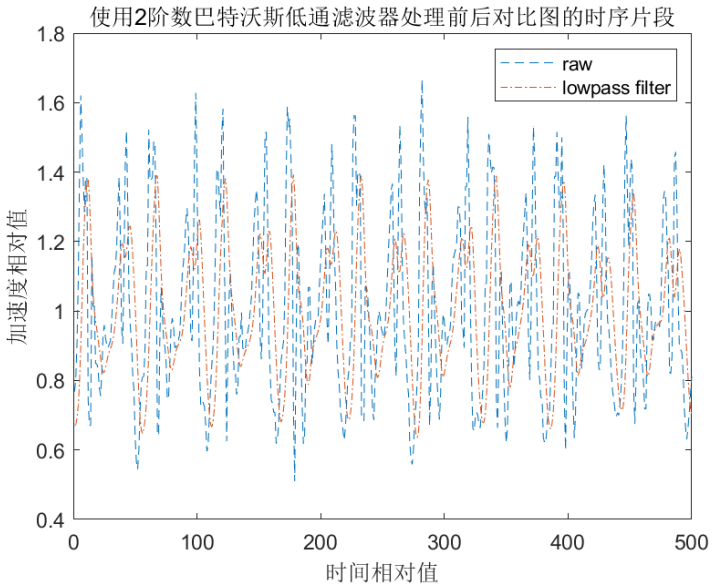

图 2-7 除噪前后对比图

2.2.4 基于固定宽度的滑动窗口采样

经过滤波器除噪的时序数据为方便神经网络模型进行训练，我们需要将时序数据按照一定大小的片段进行采样，同时，为了保证时序数据中的连续序列的信息不被丢失，在采样的时候保持一定比率的重复采样。如图，我们以 128 个采样样本作为采样窗口的固定大小，在下一次采样过程中，把上一次 50%的采样样本作为本次采样的一部分。对时序数据采样后，我们最后得到了 10299 个样本。

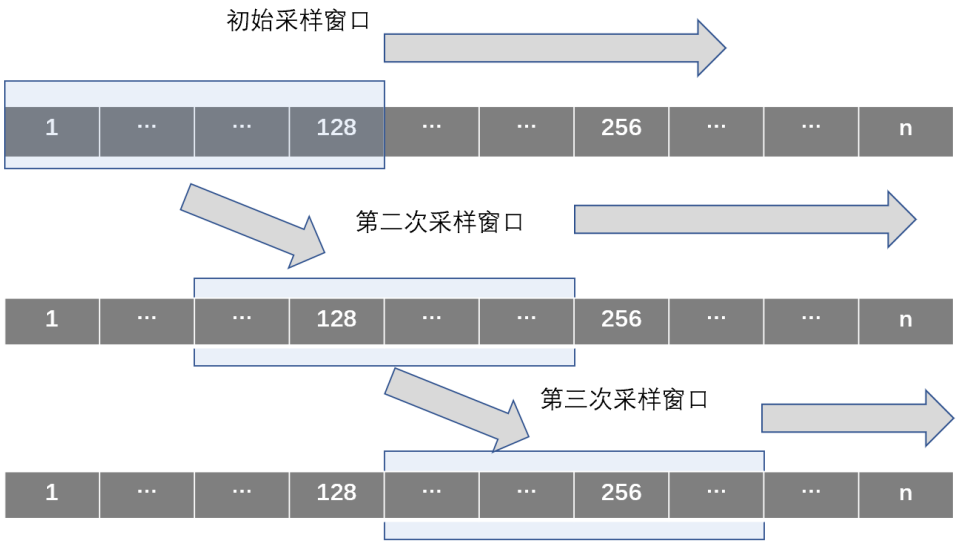

图 2-8 滑动窗口采样演示

第 3 章　本文所使用的机器学习背景知识

3.1 深度学习基础

3.1.1 分类问题

分类问题是最先出现在统计学中，后来又扩展到机器学习领域里面。机器学习中的分类问题，是基于包含已知成员类别的观察值（或实例）的训练数据集来确定新观察值属于一组类别（子种群）中的哪个的问题。常见的例子有，电子邮件系统将电子邮件分成给“正常邮件”和“垃圾邮件”两个类别；根据收集到的患者特征信息（例如性别，血压，是否患有疾病等）为特定的患者进行诊断。在机器学习的术语中，分类属于是监督学习范围的问题，即可以在识别预先给定目标结果的前提下的训练集合的情况下进行的学习。一般而言，将每一个观察结果看作为一组可量化的属性，这些属性被称为特征，对于这些属性，我们可以进行分类任务。通常，一种可以实现分类的算法，在一个具体的应用中，称为一个分类器。有时候“分类器”也指由分类算法实现的数学函数，这种数学函数可以将输入数据映射到不同类别之中。

3.1.2 逻辑回归与损失函数

在统计学中，逻辑回归用于对特定类型或事件的概率进行建模。这可以扩展为对几类事件建模，例如在图像判断是否包含猫，狗，狮子等动物。在图像中被检测到的每个对象都将被分配一个介于 0 和 1 之间的概率，并且其总和为 1。

具体的例子，在一个识别图片中是否包含猫的任务中，规定输入结果只能包含“是”与“不是”两种结果。为完成该分类任务，我们将预期输出结果的标签用 y 代表，y 在本问题中只能为 0 或者 1，即该任务为一个二分类任务。接着将一幅图片中的信息作为特征向量 x，本任务的目标是识别这是否是一张猫的图片。因此我们需要某一种算法能够输出一个预测值，这里称输出值为 yhat,这代表了对预期结果标签 y 的一种估计。具体的来说，yhat 是当给定输入特征向量 x 时，预测标签 y 为 1 的概率。这里 x 应该是一个 n_x 维的向量，约定逻辑回归的参数是 w，w 也是一个 n_x 维的向量，另外，参数 b 是一个实数。在使用激活函数之前，如果让$y^{hat} = w^T + b$直接作为输出结果，因为要求y^{hat}的值在 0 到 1 之间，而按照这种方法得到的y^{hat}值可能为大于 1 或者小于零的值，所以很难达到要求。在使用激活函数$sigmoid()$之后，根据$sigmoid()$函数的图像，当$w^T + b$值在实数范围变化时，y^{hat}的值在 0 到 1 范围之间，符合分类任务中y^{hat}的要求。在实现逻辑回归任务时，我们需要不断学习优化参数 w 和参数 b，使得y^{hat}的值不断接近预期值 y。

$$orgin:\quad x \in \mathbb{R}^{nx}; y \in \mathbb{R}; \sigma = sigmoid()$$

$$parameters:\quad w \in \mathbb{R}^{nx}; b \in \mathbb{R}$$

$$output:\quad y^{hat} = \sigma(w^T + b)$$

在本分类任务中，为达到不断学习优化参数 w 和 b 的目的，我们需要预先定义一种量化y^{hat}与 y 接近程度的函数，这里称之为损失函数。常用的损失函数有如下：

$$loss\ function1:\quad \varphi(y^{hat}, y) = \frac{1}{2}(y^{hat} - y)^2$$

$$loss\ function2:\quad \varphi(y^{hat}, y) = -(ylogy^{hat} + (1-y)\log(1-y^{hat}))$$

损失函数的对象是单个样本，在实际量化y^{hat}与 y 接近程度时，我们需要基于样本集的对象来进行量化，这里将它称为交叉函数。

$$cost\ fuction:\ \ J(w,b)=\frac{1}{m}\sum_{i=1}^{m}\varphi\left(y^{hat(i)},y^{(i)}\right)$$

3.1.3 梯度下降

根据上文，我们知道使用损失函数和代价函数来量化模型对单个样本和样本集的拟合效果。现在需要定义一种算法是参数 w 和参数 b 向着模型更接近预期目标接近。根据上文，我们有代价函数$J(w,b)$可以量化模型的拟合效果，公式如下：

$$y^{hat}=\sigma(w^T+b)$$

$$\varphi(y^{hat},y)=-(ylogy^{hat}+(1-y)\log(1-y^{hat}))$$

$$J(w,b)=\frac{1}{m}\sum_{i=1}^{m}\varphi(y^{hat(i)},y^{i})$$

函数$J(w,b)$计算的是样本集中第一个样本到第 m 个样本损失函数之和，第 i 个样本的损失函数由第 i 个样本预期值y^i和模型预测值$y^{hat(i)}$决定，而$y^{hat(i)}$的值由参数 w 和参数 b 决定，因此我们可以认为代价函数仅与参数 w，b 相关。为找出模型接近最优的结果，需要向找到函数$J(w,b)$的最小值，这里将函数$J(w,b)$的求解过程简化在一个三维空间内，如下图，我们从图中空间曲面中某一点开始，通过不断调节参数 w 和 b 到达空间曲面的最低点。

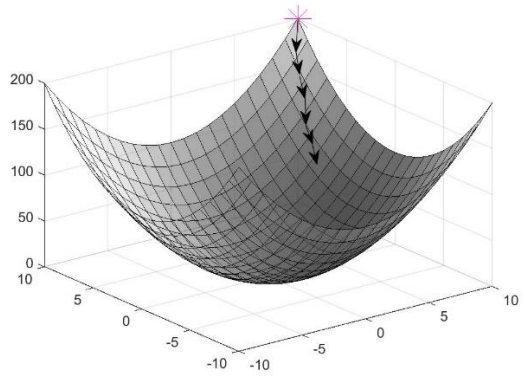

图 3-1 梯度下降图示

在求解代价函数最小值之前，首先会用一些初始值来初始化 w 和 b，初始点可能在图中的任意一点。梯度下降算法从初始点开始，向梯度最最小的地方

前进，即向最陡的下坡方向行进。为具体说明梯度下降的过程，这里将函数 J 简化成只与参数 w 相关的函数$J(w)$,它的图像如下：

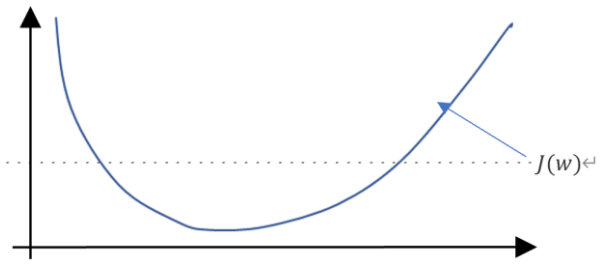

图 3-2 平面梯度下降

$$w:=w-a(\frac{dJ(w)}{dw})$$

在这里使用“:=”符号表示 w 进行迭代过程，a 为模型学习过程中的学习速率，使用 a 的大小可以控制梯度下降的步长大小，$\frac{dJ(w)}{dw}$即为函数 J 对变量 W 求导。如果初始点在图中所示的位置，参数 w 在迭代的过程中应该减小，无论初始位置在何处，w 应该总是向全局最小的方向移动。

3.1.4 激活函数

当我们需要建立神经网络的时候,通常需要考虑输出结果的范围，为达到指定输出范围的目的，我们需要使用激活函数。 一种常见的激活函数 sigmoid 的公式如下：

$$f(x)=\frac{1}{1+e^{-x}}$$

如图为 sigmoid 激活函数的图像，在自变量 x 为 0 的时候该激活函数为 0.5，而当自变量无限大或者无限小的时候，该激活函数无限接近 1 和 0。由该激活函数的属性，我们可以来二分类问题上使用它。

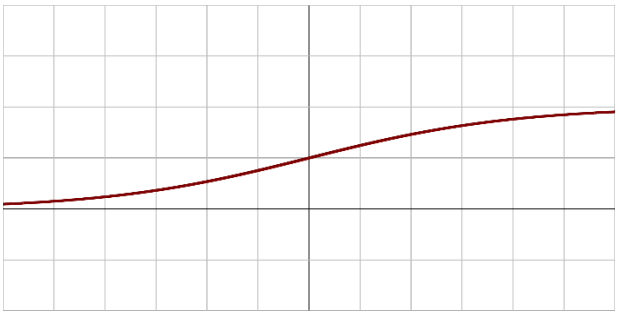

图 3-3 sigmoid 激活函数

另外与 sigmoid 类似当更通用的一种激活函数叫 tanh,tanh 的函数公式如下：

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

tanh 的图像如下:

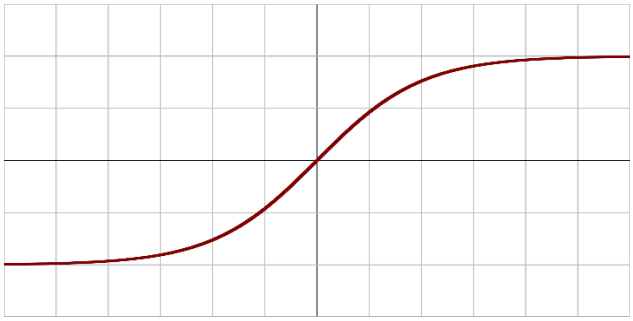

图 3-4 tanh 激活函数

与 sigmoid 相比，tanh 看起来是将 sigmoid 经过向下平移得得到的。这种激活函数的自变量为零时激活函数值为零，tanh 激活函数的平均值为零，在实际训练一个学习算法的时候，我们需要将数据的平均值接近零，所以使用 tanh 在大多数时候都好于 sigmoid。除非是一个二分类任务，因为对于二分类任务，输出结果不外乎为 0 或者 1，它正好使得y^{hat}成为一个有意义的数字，否则我们应该选择使用 tanh 而非 sigmoid。sogmoid 函数和 tanh 函数具有一个共同缺点是，如果自变量 x 很大或者很小的时候，那么这个函数的梯度（或者导数）将会很小，故如果 x 非常大或者非常小,那么该函数的斜率将最终接近 0, 这样会减慢梯度下降的速度。在某些问题中，为避免出现上述现象，通常通常会选择一个称为整流线性单元函数（又称 relu）,它的公式非常简单，如下：

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Relu 的图像如下：

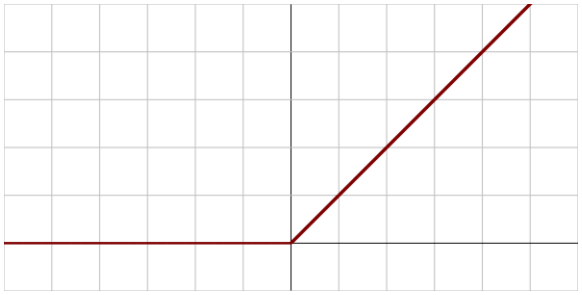

图 3-5 Relu 激活函数

当自变量 x 大于零时，该函数的梯度为 1，而当自变量小于零时，梯度为零。
当 a 为负值时, 梯度为 0 。为解决 relu 梯度有时候为的问题，可以选择 relu 的改进版 leaky relu。公式如下：

$$f(x) = \begin{cases} ax, & x < 0, a < 1 \\ x, & x \geq 0 \end{cases}$$

Leaky relu 将 relu 中原本梯度为零的地方设置为一个梯度小于 1 的值。在实际的机器学习模型中，激活函数的选择没有特别好的指导方法，对于特定问题，通常需要应用不同的激活函数进行比较实际效果来选择。

3.1.5 计算图，前线传播与反向传播

对与上文逻辑回归和梯度下降的介绍说明中已经介绍了神经网络模型中前向传播计算神经网络的损失函数和反向传播过程中的求梯度或者微分的过程，这里，为更清楚说明前向传播和反向传播的具体细节而引入计算图的概念，计算图解释了为什么以这种方式来组织　。

在这里，为了方便的演示计算图，仅仅使用一个单层神经网络　。如果我们想计算函数$J(a,b,c)$,它的函数具体如下：

$$J(a,b,c)=5(a+bc)$$

当需要计算函数 J 时可以分为三个步骤：

首先设：

$$bc=z$$

$$a+z=u$$

则函数 J 为：

$$J(a,b,c)=J(u)=5u$$

为了计算的函数 J，我们可以使用如下的计算图进行计算

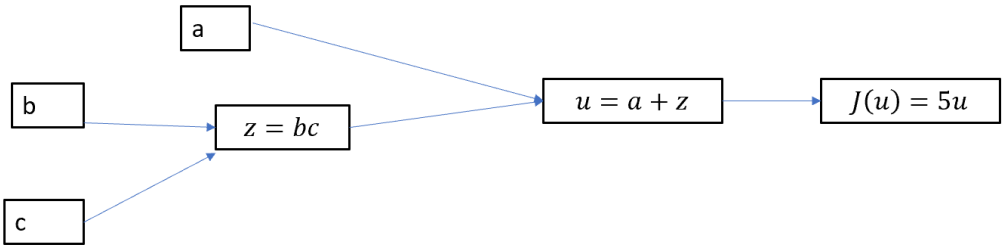

图 3-6 前向传播计算过程

函数 J 即是我们需要求解的代价函数，上图的计算过程即神经网络的前向传播。从上图的右边向左边分别计算

$$\frac{dJ(u)}{du}=5,\frac{dJ(a)}{da}=5,\frac{dJ(z)}{dz}=5$$

$$\frac{dJ(b)}{db}=\frac{dJ(z)}{z}\cdot\frac{dz}{db}$$

$$\frac{dJ(a)}{da}=\frac{dj(z)}{dz}\cdot\frac{dz}{da}$$

这一过程即神经网络的反向传播。

3.2 超参数调节，正则化与优化

3.2.1 泛化能力与容量

在机器学习过程中，我们需要的不只是在训练集上得出较好的表现，更重要的是如何让算法必须能够在先前未观测到的新输入上具有表现良好。泛化能力是指在先前未被涉及到的数据上具有良好的表现。一般的情况下，训练机器学习模型时，我们可以使用某个训练集，在训练集上计算一些被称为训练误差的度量误差，目标是降低训练误差。在这里，我们在讨论的是一个的优化问题，我们希望泛化误差能够更低，泛化误差被定义为新输入的误差期望。这里，期望的计算基于不同的可能输入，这些输入采自系统在现实中遇到的分布。通常，我们度量模型在训练集中分出来的测试集样本上的性能，来评估机器学习模型的泛化误差。在上文示例中，通过最小化训练误差来训练模型:

$$J(w,b)_{train}$$

但是我们真正关注的是测试误差:

$$J(w,b)_{test}$$

如果我们只能观测到训练集，如何能够对测试集的性能产生影响呢？在统计学习理论中，如果可以对训练集和测试集数据的收集方式提供一些假设而不是以随机的方式收集训练集和测试集，那么我们可以对算法进行改进。我们能观察到训练误差和测试误差之间的直接联系是，随机模型训练误差的期望和该模型测试误差的期望是一样的。假设我们有概率分布$p(x,y)$，从中重复采样生成训练集和测试集。对于某个固定的 w，训练集误差的期望恰好和测试集误差的期望一样，这是因为这两个期望的计算都使用了相同的数据集生成过程。这两种情况的唯一区别是数据集的名字不同。在使用机器学习算法时，我们不会提前固定参数，然后采样得到两个数据集。我们采样得到训练集，然后挑选参数去降低训练集误差，然后采样得到测试集。在这个过程中，通常测试误差期望会大于或等于训练误差期望。决定机器学习算法效果可以参考下列因素：

(1)降低训练误差。

(2)缩小训练误差和测试误差的差距。

这两个因素对应在机器学习中的两个问题：欠拟合和过拟合。欠拟合是指模型不能在训练集上获得足够低的误差，而过拟合是指训练误差和测试误差之间的差距太大。

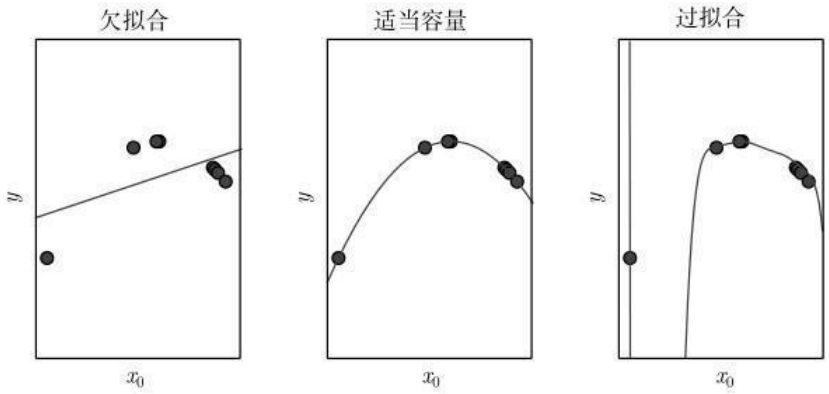

图 3-7 模型的拟合程度

我们将模型拟合各种函数的能力称为模型的容量。调整模型的容量，我们可以改变模型是否偏向于过拟合或者欠拟合。因此，我们可以知道，容量低的模型可能很难拟合训练集。容量高的模型可能会过拟合，因为记住了不适用于测试集的训练集性质。

3.2.2 正则化

通过上文我们知道，机器学习中的一个核心问题是设计不仅在训练数据上表现好，而且能在新输入上泛化好的算法。在机器学习中，许多策略被显式地设计来减少测试误差(可能会以增大训练误差为代价)。这些策略被统称为正则化。

在机器学习中我们可以使用正则化来防止过拟合，并以此来降低网络的泛化误差。这里以上文中的逻辑回归为例,我们的目标是求得最小的代价函数，该代价函数定义为每一个训练样本的预测的损失之和,其中 w 和 b 是逻辑回归的参数,w 是一个 x 维的参数向量 b 是一个实数 。

$$w \in \mathbb{R}^{nx}; b \in \mathbb{R}$$

$$\min_{w,b} J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \varphi(y^{hat(i)}, y^{i})$$

要为逻辑回归正则化，我们在函数J加上一个正则化参数 $\lambda/2m$与 w 范数的平方的乘积。

$$\min_{w,b} J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \varphi(y^{hat(i)}, y^{i}) + \frac{\lambda}{2m}\|w\|_2{}^2$$

$$\|w\|_2^2 = \sum_{j=1}^{nx} w_j^2 = w^T w$$

我们称上述公式为L_2正则化，L_2正则化在机器学习是一种最常见的正则化的方式，因为在实际的神经网络中，参数 w 的个数远大与 b,因此在使用L_2正则化是一般省略参数 b 的正则化。另外一种在机器学习不常用正则化方式为L_1正则化。

$$\min_{w,b} J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \varphi(y^{hat(i)}, y^i) + \frac{\lambda}{2m}\|w\|_1$$

除了L_2正则化，通常在机器学习我们会使用一种另外一种有效的正则化技术，一般称它为随机丢弃正则化（dropout）。在一个神经网络模型中使用 dropout 后，将在该层神经网络节点中随机按照设置的概率丢弃该节点的信息，

3.2.3 多分类与 one-hot 编码

在机器学习中，通常我们有多个分类类别，在上文中我们使用的数据就有六种分类，我们称之为这个是一个多分类问题。如果对于多个分类类别之间相互独立没有逻辑关系，我们一般使用 one-hot 编码，对于上文的六种类别，可以使用下面矩阵表示，矩阵中每一个向量表示一种分类。

$$\begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{matrix}$$

3.3 序列模型

3.3.1 循环神经网络

在机器学习中，一般的神经网络模型如下图，样本由特征$x^{<1>}, x^{<2>} \ldots x^{<T_x>}$所组成，如果输出按照 one-hot 编码，那么输出$y^{<1>}, y^{<2>} \ldots y^{<T_y>}$中有$T_y$个 0 或者 1 的数字表示输出$T_y$个维度。

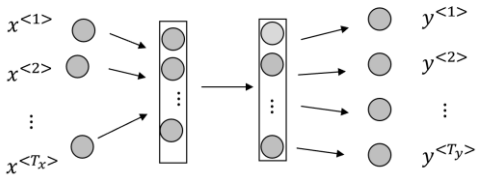

图 3-8 基本神经网络模型结构

这种一般的神经网络模型在处理一些任务时，可能面临两个难以解决的问题。在机器学习任务中，有时候模型输入每个样本的数据项大小不一致，一般使用填充法对数据缺失项进行填充，这样并不是一个很好的办法。另外一个更严重的问题的是，使用这种模型进行机器学习，模型无法共享样本中不同位置的信息，这样模型在面临一些例如机器翻译，语音识别，语义分析等输入数据存在关系的任务时，显得无能为力。为了解决上述存在的问题，在机器学习中一般使用循环神经网络（RNN）等一些的模型处理上述任务。通常来说，循环神经网络是一类可以用于处理序列数据的神经网络。我们可以认为循环神经网络

是一个由外界因素 x （t） 驱动的动态系统

$$s^{(t)} = f(s^{(t-1)}, x^{(t)}, \theta)$$

由上述公式我们知道，循环神经网络的状态$s^{(t)}$与上一个单元的状态$s^{(t-1)}$和本单元的样本输入和参数 θ 有关，可以看到，模型的当前状态中包含了过去序列的部分信息。如图，最一般的循环神经网络模型结构如下

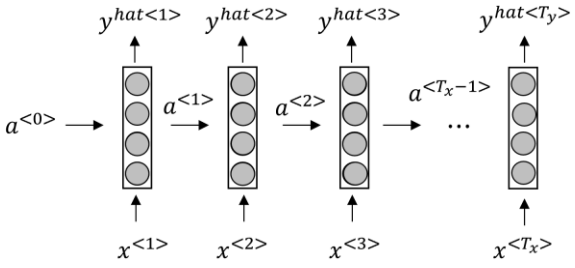

图 3-9 基本 RNN 结构

一般，对于第一个单元的输入$a^{<0>}$，我们可以进行置为零向量

$$a^{<0>} = \vec{0}$$

这里我们定义两个激活函数g_1，g_2，一般我们使用$tanh$作为计算$a^{<t>}$的激活函数，在二分类问题中，使用$sigmoid$计算$y^{hat<t>}$的值。

$$g_1 = tanh$$

$$g_2 = sigmoid$$

由图中的结构我们可以$a^{<t>}$和$y^{hat<t>}$在各个单元的值。

$$a^{<1>} = g_1(W_{aa}a^{<0>} + W_{ax}x^{<1>} + b_a$$

$$y^{hat<1>} = g_2(w_{ya}a^{<1>} + b_y)$$

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a$$

$$y^{hat<t>} = g_2(w_{ya}a^{<t>} + b_y)$$

3.3.2 长短期记忆网络

由上文可知，一般的循环神经网络的结构如下图，每一个时间步包括当前的样本输入和上一个时间步的输出作为输入，$a^{<t-1>}$中存在着之前模型中存在的信息，通过将$a^{<t-1>}$作为输入的一部分，使当前模型包含了过去的信息。在一些领域，这种循环神经网络已经开始被使用。

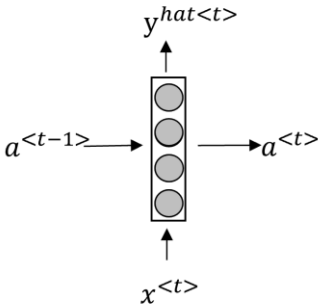

图 3-10 RNN 单元

但是，随着研究的深入，当把循环神经网络使用在需要记住样本中较远的地方的信息时，使用这种结构的模型很难达到要求。更一般的说法是，循环神经网络只能记住较短时间内的信息，当要处理存在长期依赖信息的任务时，不能完成要求。为了解决这样的问题，1997 年提出了长短期记忆网络(LSTM),这种模型就像它的名字一样，LSTM 确实能够长时间记住样本的信息。如图，LSTM 的内部结构如下。⊛和⊕分别表示点积和相加操作。在 LSTM 的内部存在两个输入$c^{<t-1>}$,$a^{<t-1>}$和两个输出$c^{<t>}$，$a^{<t>}$。

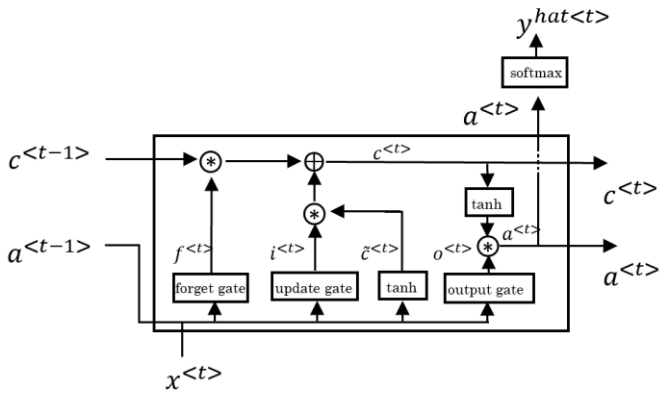

图 3-11 LSTM 内部结构

在 LSTM 内部存在 6 种基本的运算，其中我们把$f^{<t>}$, $i^{<t>}$, $o^{<t>}$分别叫遗忘门，更新门和输出门。σ函数通常是 sigmoid 函数，我们知道 sigmoid 会输出在 0 至 1 范围内的数字，使用 sigmoid 可以起到闸门的作用，它可以限制信息通过的大小。

LSTM 的运算过程可以分为三步，第一步，LSTM 选择从$a^{<t-1>}$中选择性的丢失部分信息，将代表包含记忆信息的$a^{<t-1>}$和当前的输入$x^{<t>}$作为$f^{<t>}$函数输入,然后经过 sigmoid 函数后输出一个 0 到 1 之间的数然后与$c^{<t-1>}$相乘。如果输入接近 1 表示几乎不遗忘的输出，接近 0 的时候表示几乎完全遗忘掉信息。

$$f^{<t>} = \sigma(W_f[\,a^{<t-1>}, x^{<t>}] + b_f)$$

第二步是确定在 LSTM 中要存储哪些新信息，这里会使用到$i^{<t>}$和$\tilde{c}^{<t>}$的计算过程。同样，$i^{<t>}$使用 sigmoid 后与$\tilde{c}^{<t>}$进行点积运算，起到了决定选择保留部分记忆信息的作用。

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$$

$$i^{<t>} = \sigma(W_u[\,a^{<t-1>}, x^{<t>}] + b_u)$$

最后一步决定是我们最后输出$y^{hat<t>}$和$a^{<t>}$，它由经过 tanh 的$c^{<t>}$与$o^{<t>}$的点积运算组成。

$$o^{<t>} = \sigma(W_o[\,a^{<t-1>}, x^{<t>}] + b_o)$$

通过对 LSTM 内部基本结构的介绍，我们可以知道 LSTM 外部的输入公式：

$$c^{<t>} = \; y^{<t>} * \tilde{c}^{<t>} + f^{<t>} * c^{<t-1>}$$

$$a^{<t>} = o^{<t>} * \tanh\,(c^{<t>})$$

$$y^{hat<t>} = softmax(a^{<t>})$$

现在如果我们隐藏 LSTM 的内部细节，将多个计算步连接在一起，我们方向输入仅有$x^{<t>}$,输出为$y^{hat<t>}$。从外部来看，LSTM 的输入和输出与 RNN 相同。

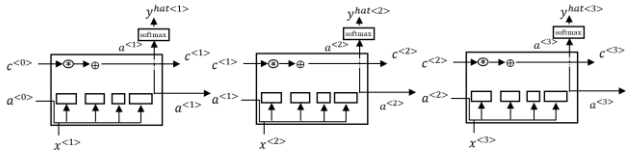

图 3-12 LSTM 外部结构

3.3.3 双向循环神经网络

从上文我们可知，无论是 RNN 还是 LSTM,如果隐藏内部细节，它们都可以用下面图表示：

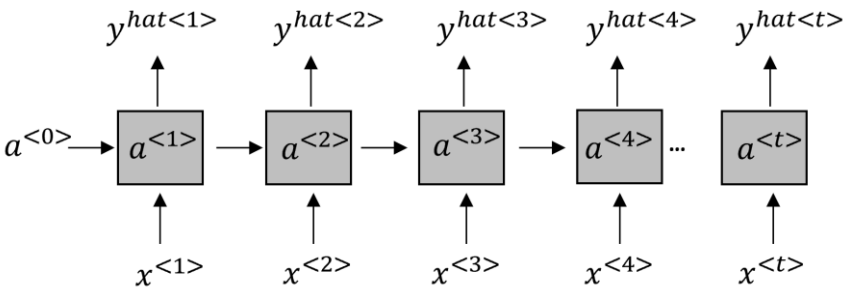

图 3-12 循环神经网络一般结构

从上图可以直观的知道第 t 步的模型包含了 t 步之前的信息，但是在第一步中，模型不存在模型后面学习步骤的信息。有时候我们需要从未来的时间步骤中学习表示，用来处理需要依赖上下步骤信息的任务。如下图，我们可以使用双向的网络结构来解决这一问题。

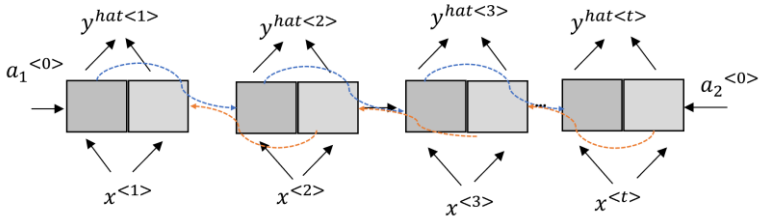

图 3-13 双向循环神经网络结构

在上图所示结构中，在第 t 个时间步，输入$x^{<t>}$由该时间步的两个基本单元共享，而$y^{hat<t>}$由该时间步的两个单元的输出共同决定。在模型的内部，包含记忆信息的$a_1{}^{<t>}$和$a_2{}^{<0>}$从相反的方向传递。

第 4 章　基于 LSTM 的人体活动识别

4.1 实验度量

4.1.1 混淆矩阵

在机器学习中，为了全面衡量神经网络模型的拟合效果，通常需要多方面来评价模型的效果，因此我们引入混淆矩阵，如图

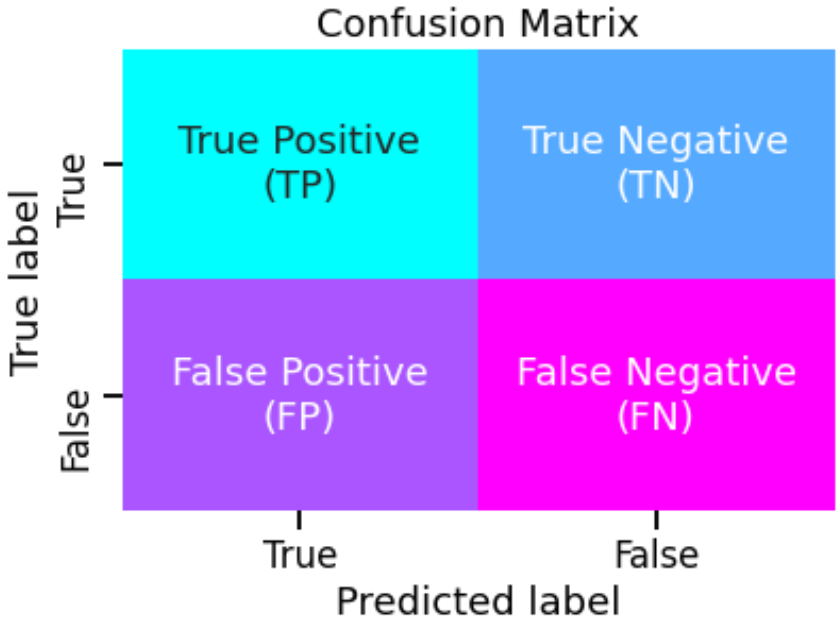

图 4-1 二分类混淆矩阵

在一个二分类问题中，混淆矩阵的 TP 表示在真实的类别为 True，模型预测也为 True 的数量，FP 表示在真实的类别为 True，模型错误的预测成了 False，TN 表示在真实的类别为 Fasle，模型错误预测成了 True，FN 表示在真实的类别为 False，模型预测也为 Fasle 的数量。

当在一个多分类中，混淆矩阵同样适用。在一个需要通过模型区分 t 类对面的多分类任务中，混淆矩阵应该为：

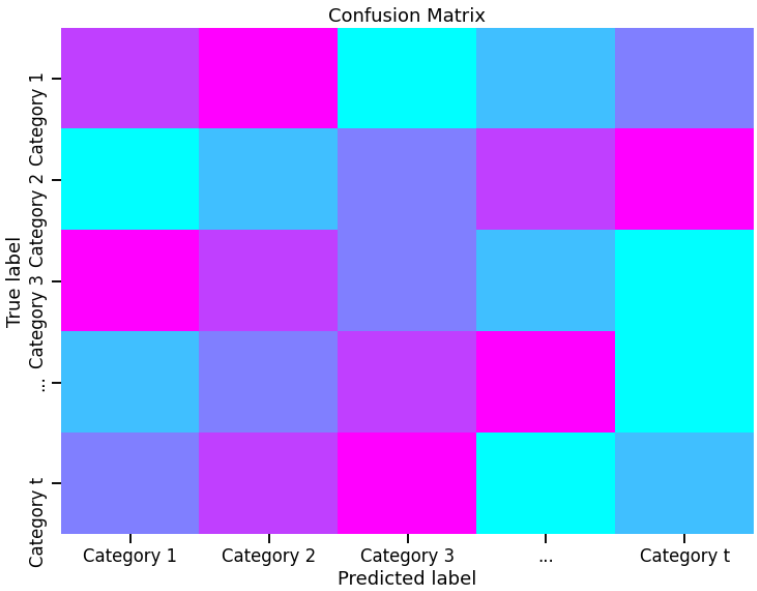

图 4-2 多分类混淆矩阵

4.1.2 度量指标

在机器学习中，常用 4 种指标度量分类器模型的效果，这四种指标分别是准确率（accuracy），精确率（precision），灵敏性（sensitivity），特异性（specificity），在一个二分类问题它们均由上文混淆矩阵 4 类情况的值组合而来。

准确率表示分类器对一个给定的数据集，正确预测的样本数占总样本数之比。精确率表示表示分类器预测结果与真实结果均为 True 的样本与分类器预测为 True 的样本的比。灵敏性表示

在本文中，我们使用的度量指标 accuracy 的值即为在多分类问题中模型预测与预设的类别相同的值。在上图中，accuracy 的值应该为：

$$accurcy = \frac{A1 + A5 + A9}{\sum_{i=1}^{9} Ai}$$

为了更全面评价模型的性能，我们还使用 precision 作为度量指标，precision 显示的是预设的类别与模型预测为该类别的比的值。在上图中，precision 应该为:

$$precision = \frac{1}{3}\sum_{i=1}^{3} \frac{A(4i-3)}{A(3i-2) + A(3i-1) + A(3i)}$$

最后，为评价分类器效果，这里引入 ROC 曲线。，为方面理解 ROC 曲线，本文仅使用二分类举例。我们引入 TPRate 和 FPRate 两个变量，由图 中的二分类混淆矩阵，我们令：

$$TPRate = \frac{TP}{FN + TP}$$

$$FPRate = \frac{FP}{TN + FP}$$

由公式可知 TPRate 的意义是所有预设类别为 Positive 的样本中，预测类别为 Positive 的比例。FPRate 的意义是所有预设类别为 Negative 的样本中，预测类别为 Positive 的比例。以 TPRate 和 FPRate 分别为纵坐标和横坐标建立的图像称为 ROC 曲线。

如图

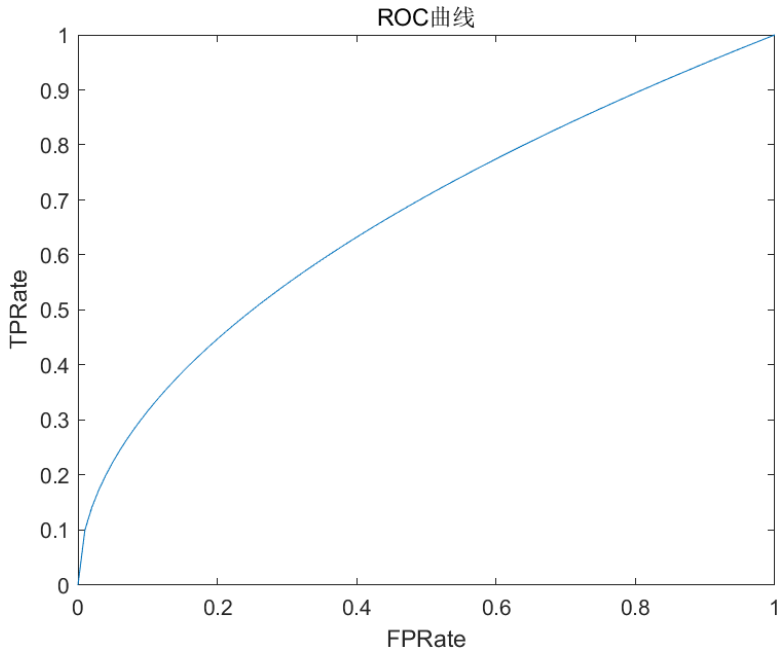

图 4-3 ROC 曲线

我们取 ROC 曲线与横坐标相交而形成的面积，称它为 auc。由 auc 的计算方法我们可知，auc 值同时考虑了分类器对于多种类别的分类能力，在样本不平衡的情况下，依然能够对分类器作出合理的评价

4.2 实验.

4.2.1 实验环境

本实验环境为谷歌 colab 提供的免费 tensorflow 环境，为了方便的进行实验，本文只使用 tensorflow 中的 keras api 构建模型，并且本文中的实验使用神经网络模型使用常见的层次模型搭建。

4.2.2 实验配置

本实验使用的数据集为公开数据集，数据集按照 7：3 分为训练集和测试集，其中训练集有 7532 个样本，测试集有 2947 个样本。每一个样本中包含按照时间顺序进行 128 采样的数据，每一次采样的数据包含 9 个维度的数据。每一个样本的类型被标注，标注的分类共有 6 种。

实验之前，预先搭建了一个包含三层双向 LSTM 结构的模型，并且使用 RMSprop 优化器，优化器的学习率设置为 0.0025，使用 CategoricalCrossentropy 作为损失函数。为提高模型的泛化能力,在 LSTM 结构的后面我们使用了 0.3 的 dropout 值。因为本文的模型使用了 one-hot 编码，在最后为了控制数据的输出格式，在最后使用了一层 Dense,将数据输出格式改为 6。

4.2.3 实验结果

本实验对模型进行了 128 次迭代，分别观察每一迭代过程后度量指标的变化。通过实验，我们发现数据集在第 20 次迭代之后，度量指标基本处于稳定。

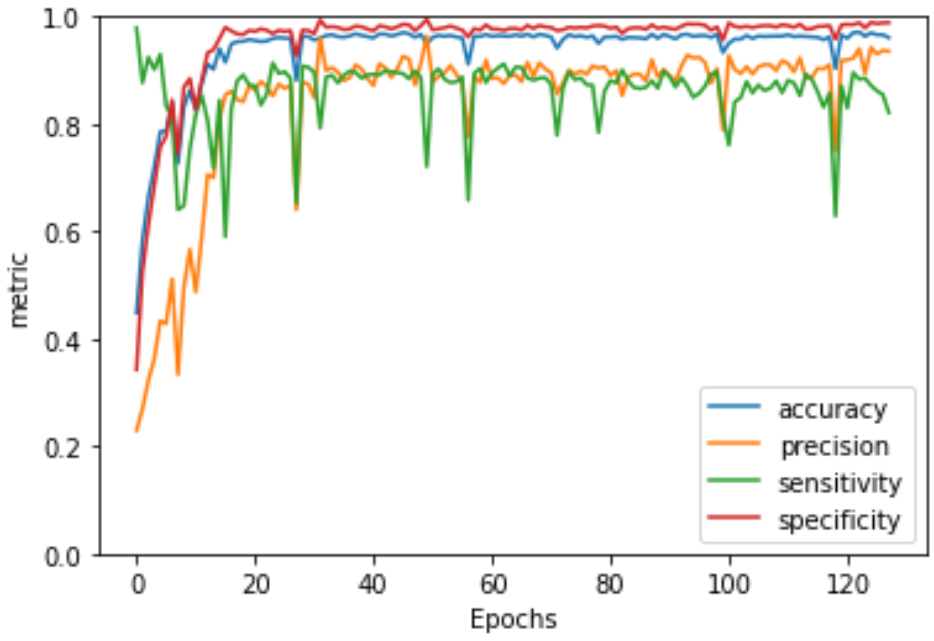

图 4-4 模型训练效果

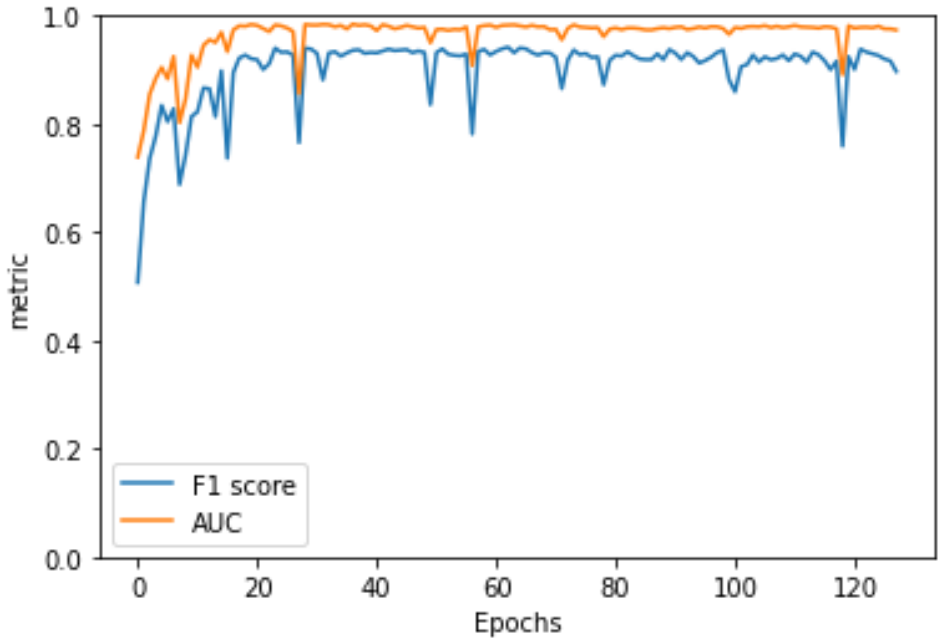

图 4-5 模型训练效果

其中 accuracy 的值趋于 0.95，precision 趋 0.87，sensitivity 为 0.86，specificity 为 0.97。为了更好的进行评价，我们使用综合性更高的度量指标，这里 F1 度量为 0.92， auc 趋于 0.97。从模型的训练结果来看，使用该模型的分类效果较好，但是由上文我们知道，模型训练效果不代表模型的最后泛化能力，我们需要将分类器在测试数据集上进行测试。

因此，这里将训练好的模型对测试数据集进行预测，为更加直观的看出分类器的效果，这里将预测结果使用混淆矩阵表示，如图。

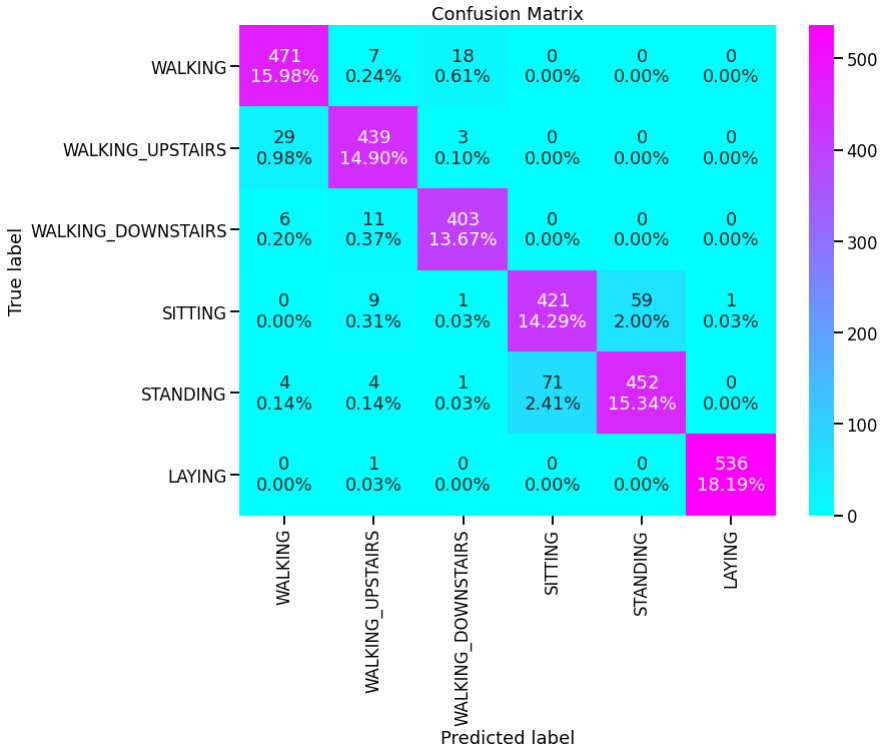

图 4-5 模型预测测试集混淆矩阵

由混淆矩阵我们可以计算出，accuracy 的值为 0.94，precision 为 0.87，F1 度量为 0.91。与模型训练时的效果接近，故可以看出此模型的分类效果较好。

致　谢

参考文献

附录

[1] Dr Mark Weiser. 1996 Computer Science Challenges for the Next 10 Years [EB/OL]. https://www.youtube.com/watch?v=7jwLWosmmjE

[2] Iosifidis A, Tefas A, Pitas I. Multi-view action recognition based on action volumes, fuzzy

distances and cluster discriminant analysis[J]. Signal Processing, 2013, 93(6): 1445-1457.

[3] Weinland D, Ronfard R, Boyer E. A survey of vision-based methods for action representation, segmentation and recognition[J]. Computer vision and image understanding, 2011, 115(2): 224-241.

[4] Ali S, Shah M. Human action recognition in videos using kinematic features and multiple instance learning[J]. IEEE transactions on pattern analysis and machine intelligence, 2008, 32(2): 288-303

[5] Ann O C, Theng L B. Human activity recognition: a review[C]//2014 IEEE international conference on control system, computing and engineering (ICCSCE 2014). IEEE, 2014: 389-393.

[6] L. Chen, H. Wei, and J. Ferryman, "A survey of human motion analysis using depth imagery," Pattern Recognit. Lett., vol. 34, no. 15, pp. 1995–2006, Nov. 2013

[7] Preis J, Kessel M, Werner M, et al. Gait recognition with kinect[C]//1st international workshop on kinect in pervasive computing. New Castle, UK, 2012: 1-4.

[8] Tamás V. Human Behavior Recognition In Video Sequences[J]. Technical University of Cluj-Napoca, 2013.

[9] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in 2011 International Conference on Computer Vision, 2011, no. Iccv, pp. 1036–1043.

[10] Banos O, Damas M, Pomares H, et al. Daily living activity recognition based on statistical

feature quality group selection[J]. Expert Systems with Applications, 2012, 39(9): 8013-8021.

[11] Sazonov E S, Fulk G, Hill J, et al. Monitoring of posture allocations and activities by a shoe-based wearable sensor[J]. IEEE Transactions on Biomedical Engineering, 2010, 58(4): 983-990.

[12] Ravi N, Dandekar N, Mysore P, et al. Activity recognition from accelerometer data[C]//Aaai. 2005, 5(2005): 1541-1546.

[13] Kautz H. A formal theory of plan recognition[J]. 1987.

[14] Charniak E, Goldman R P. A Bayesian model of plan recognition[J]. Artificial Intelligence, 1993, 64(1): 53-79.

[15] Wang J, Chen Y, Hao S, et al. Deep learning for sensor-based activity recognition: A survey[J]. Pattern Recognition Letters, 2019, 119: 3-11.

[16] Zhang S, Wei Z, Nie J, et al. A review on human activity recognition using vision-based method[J]. Journal of healthcare engineering, 2017, 2017.

[17] Li H, He X, Chen X, et al. Wi-motion: A robust human activity recognition using wifi signals[J]. IEEE Access, 2019, 7: 153287-153299.