

分类号 \_\_\_\_\_ 密级 \_\_\_\_\_

UDC <sup>注1</sup> \_\_\_\_\_

# 学 位 论 文

## 基于深度学习的人脸认证方法研究

(题名和副题名)

王 峰

(作者姓名)

指导教师

程 建

教 授

电子科技大学

成 都

(姓名、职称、单位名称)

申请学位级别 **博士**

学科专业 **电路与系统**

提交论文日期 **2018.10.07** 论文答辩日期 **2018.12.04**

学位授予单位和日期 **电子科技大学 2018 年 12 月**

答辩委员会主席 \_\_\_\_\_

评阅人 \_\_\_\_\_

注1：注明《国际十进分类法 UDC》的类号。

# **Research on Deep Learning Based Face Verification**

**A Doctoral Dissertation Submitted to  
University of Electronic Science and Technology of China**

**Discipline:** Circuits and Systems

---

**Author:** Feng Wang

**Supervisor:** Prof. Jian Cheng

**School:** Information and Communication Engineering

## 摘 要

人脸认证是当今模式识别与计算机视觉学术界和工业界的热门研究主题之一，在安防、金融、军事、交通、商务等领域均有广泛的应用前景。随着深度学习的飞速发展和数据量的日益提升，计算机的人脸识别能力在一定程度上已经超过了人类水平，目前正在向百万分之一甚至是亿分之一的误识率发展。在人脸认证模型的性能如此高的现在，如何能百尺竿头更进一步，更好地利用深度学习技术来进一步提升人脸认证模型的性能是如今的一大难题。损失函数是控制整个深度神经网络训练的中枢，本学位论文将深入探究深度学习中损失函数的机理，并把之前广泛用于通用图像识别的损失函数和多种用于度量学习的损失函数改造得更加适合人脸认证模型的训练。本学位论文的研究内容主要分为以下几点：

### 1. 在神经网络中进行 $L_2$ 超球面嵌入。

在通用图像识别中最常用的 Softmax 交叉熵损失函数所优化的是内积相似度，而在人脸认证模型的测试过程中，使用的却是余弦相似度。为了使训练与测试过程保持一致，需要在训练过程中引入向量归一化的步骤。本文通过分析 Softmax 损失函数的性质，发现了 Softmax 对特征幅度的过度优化现象，从而论证了使用余弦相似性度量的必要性。

然而直接将内积相似度简单地替换为余弦相似度会带来模型无法收敛至较小损失的问题，本文深入地分析了这一问题，提出了 Softmax 损失在归一化后的下界理论，从理论上解释了无法有效收敛问题的机理。该理论表明由于将特征嵌入到单位球时特征的模长固定为 1，导致损失函数的下界过高而无法收敛。

根据本文提出的理论，可以得到一个非常简单的解决方案，只需在计算完余弦相似度后乘以一个尺度系数  $s$  即可使损失函数能够优化至零损失。本文还分析了有关于  $s$  的一系列性质，这些分析将有助于对  $s$  进行设置。实验证明使用该解决方案对已有的人脸认证模型进行几轮微调即可得到可观的性能提升。

### 2. 提出对度量学习损失函数进行分类化改造的思路。

在传统的度量学习方法引入深度学习时，因为数据量和计算量非常巨大，会存在采样困难的问题。在向量归一化的基础上，本文研究表明 Softmax 交叉熵损失函数中的权重矩阵实际上起到的是“类代理”的作用。本文通过将“类代理”的概念引入度量学习方法中，将样本与样本的比对替换为样本与向量的比对，即可规避采样带来的问题，而同时又保持了度量学习能够同时减小类内距离和增大类间距离的优势。通过分析深度神经网络中归一化操作的性质，本文还分析了特征

向量归一化具备的难例挖掘功能，以及“类代理”归一化具备的缓解类不均衡问题的作用。

使用改造后的度量学习损失函数进行人脸认证模型的训练，在多个数据集上都取得了比原损失函数更加优秀的性能。

### 3. 引入加性间隔以增大类间距离。

在带有  $L_2$  超球面嵌入的 Softmax 交叉熵损失函数的基础上，本文进一步分析了分类损失函数与度量学习损失函数之间的异同，并将两者的优势互补，将间隔的概念引入了 Softmax 交叉熵损失函数。本文通过实验发现在多种间隔的形式中，加性的间隔具备更好的性能，实现方式也较为简单，非常容易复现。为了充分认识加性间隔的特性，本文对加性间隔的性质进行了一系列的数学分析与可视化展示。

对于人脸认证模型的性能评价，本文中提出了一种名为隐式间隔的评价指标，它能够在训练过程中就对模型的判别能力进行评价，从而指导训练过程中对模型参数的调整。

在  $L_2$  超平面嵌入的基础上引入加性间隔后，人脸认证模型的性能在多个大型数据集上取得了高达 5% 的性能提升，是目前最好的人脸认证损失函数之一。

**关键词：**深度学习，人脸认证，度量学习，损失函数，特征嵌入，加性间隔

**ABSTRACT**

Face verification is one of the most popular topics in computer vision and pattern recognition community. It is widely used for identity authentication in enormous areas such as finance, military, public security and so on. With the rapid development of deep learning and the growing data volume, the face verification ability of computers has already surpassed the human beings on several benchmarks. As the performance of face verification models is almost saturated, how to further improve the performance becomes a very challenging problem. Since the loss function plays a key role in guiding the training of deep neural networks, this dissertation delves into the loss functions, modifying the most widely used classification and metric learning loss functions to make them more suitable for face verification task. The main contribution of this dissertation includes the following aspects.

1. Proposing to apply  $L_2$  hypersphere embedding in neural networks.

The similarity used by the softmax cross-entropy loss is inner-product similarity. But during the deployment, the most common similarity measurement is cosine similarity. The difference between inner-product and cosine similarity is the vector  $L_2$  normalization. To make the training procedure consisting with the testing procedure, the inner-product layer should be replaced by the cosine layer. However, such a trivial modification would make the model unable to converge. This dissertation provides a theory to explain this problem. The theory describes the lower bound of softmax loss with regard to the norm of features. Since the norms of the features are normalized to 1, the lower bound is too high to make the model converge.

According to this theory, the solution can be inferred. By appending a scale factor after the cosine layer, the model is able to converge. This dissertation also analyzes several properties of the scale factor, which would help understanding it. By finetuning the face verification for a few epochs using the modified loss function, the performance improves significantly on several benchmarks.

2. Reformulating metric learning loss functions.

When the traditional metric learning methods are introduced into deep learning, sampling is a crucial step to find hard pairs or triplets of samples to feed into the neural network. However, the sampling algorithms are usually very tricky and difficult to imple-

ment. Based on the vector normalization, the weight matrix used by softmax cross-entropy loss is actually playing a role of “class agent”. By applying the class agent strategy on metric learning loss functions, the sampling problem can be avoided, while the advancements of metric learning loss functions are still retained.

By analyzing the normalization operation in neural networks, it can be discovered that the feature normalization has an effect of hard sample mining and the weight normalization has an effect of alleviating the class imbalance problem. All the loss functions proposed in this dissertation have feature and weight normalization, so they all have such properties. Experiments show that the reformulated loss functions have superior performance than the traditional metric learning loss functions.

### 3. Introducing the additive margin scheme into softmax cross-entropy loss.

Based on the softmax cross-entropy loss with  $L_2$  hypersphere embedding, this dissertation analyzes the similarities and differences between classification loss functions and metric learning loss functions. By introducing the margin strategy into softmax cross-entropy loss, the advancements of both classification and metric learning loss functions are merged into the proposed loss function. Among several margin schemes, additive margin achieves the best performance.

This dissertation also proposes an evaluation metric named “latent margin”. With the latent margin, researchers will get a direct feeling about the discriminative power of the training model. The relationship between the latent margin and our proposed additive margin is also analyzed to guide the parameter tuning during the model training.

By importing the additive margin, the performance of face verification model improves significantly on several benchmarks. It is the state-of-the-art face verification loss function by the time this dissertation is written.

**Keywords:** deep learning, face verification, metric learning, loss function, feature embedding, additive margin

## 目 录

第一章 绪 论 .....	1
1.1 研究意义 .....	1
1.2 问题描述 .....	1
1.3 主要困难 .....	2
1.4 研究现状 .....	4
1.4.1 深度学习的发展现状 .....	4
1.4.2 人脸认证的发展现状 .....	8
1.4.2.1 基于几何特征的人脸认证 .....	8
1.4.2.2 基于整体特征的人脸认证 .....	9
1.4.2.3 基于局部特征的人脸认证 .....	10
1.4.2.4 基于深度学习的人脸认证 .....	11
1.5 主要创新点 .....	12
1.6 章节安排 .....	13
第二章 基础理论与相关工作 .....	14
2.1 卷积神经网络 .....	14
2.1.1 模型结构 .....	14
2.1.1.1 卷积层 .....	14
2.1.1.2 内积层 .....	16
2.1.1.3 激活函数层 .....	16
2.1.1.4 归一化层 .....	18
2.1.1.5 残差网络 .....	20
2.1.2 损失函数 .....	21
2.1.3 优化 .....	22
2.2 人脸认证 .....	25
2.2.1 相关概念 .....	25
2.2.2 人脸数据集 .....	27
2.2.3 评价标准 .....	28
2.2.4 人脸检测与对齐 .....	30
2.2.5 特征脸 (EigenFace) 与判别脸 (FisherFace) .....	32
2.2.6 深度度量学习 .....	34

2.2.7 中心损失.....	35
2.2.8 乘性角度间隔.....	35
2.3 本章小结.....	38
<b>第三章 基于 <math>L_2</math> 超球面嵌入的人脸认证损失函数</b> .....	<b>39</b>
3.1 $L_2$ 超球面嵌入的必要性.....	39
3.1.1 训练与测试的不匹配问题.....	39
3.1.2 Softmax 交叉熵损失下的特征分布.....	40
3.1.3 Softmax 交叉熵损失函数的饱和问题.....	43
3.2 使用 Softmax 交叉熵损失函数优化余弦相似度.....	44
3.2.1 在神经网络中使用余弦相似度.....	45
3.2.2 直接应用余弦相似度遇到的困难与解决方案.....	47
3.2.3 分析与讨论.....	50
3.2.3.1 参数 $s$ 的升降条件.....	50
3.2.3.2 参数 $s$ 控制下的分类器权重分配.....	51
3.2.3.3 参数 $s$ 的几何意义.....	53
3.3 实验结果及分析.....	55
3.3.1 实验设置.....	55
3.3.1.1 基线模型.....	55
3.3.1.2 模型训练.....	55
3.3.1.3 模型测试.....	56
3.3.2 LFW 数据集.....	56
3.3.3 YTF 数据集.....	58
3.4 本章小结.....	59
<b>第四章 度量学习损失函数的分类化改造</b> .....	<b>61</b>
4.1 使用分类损失函数进行度量学习.....	61
4.2 误差分析.....	63
4.3 归一化的作用.....	66
4.3.1 归一化在难例挖掘方面的作用.....	66
4.3.2 权重归一化在类不均衡问题上的作用.....	67
4.4 实验结果及分析.....	68
4.4.1 实验设置.....	68
4.4.2 测试结果.....	69
4.5 本章小结.....	70

第五章 引入加性间隔的人脸认证损失函数 .....	72
5.1 引入间隔的必要性 .....	72
5.2 加性间隔 .....	73
5.2.1 定义 .....	73
5.2.2 角度间隔与余弦间隔 .....	75
5.2.3 几何意义 .....	75
5.2.4 特征分布可视化 .....	78
5.2.5 类空间分割可视化 .....	78
5.3 理论分析与讨论 .....	80
5.3.1 与三元组损失函数的联系 .....	80
5.3.2 从最优化的角度理解 Softmax 交叉熵损失函数 .....	83
5.3.3 模型判别能力指标 .....	86
5.3.4 模型统计量小结 .....	87
5.4 实验结果及分析 .....	90
5.4.1 数据集去重 .....	90
5.4.2 实验细节 .....	91
5.4.3 参数 $m$ 的作用 .....	92
5.4.4 模型隐式间隔 .....	94
5.5 本章小结 .....	95
第六章 全文总结与展望 .....	96
6.1 全文总结 .....	96
6.2 不足以及后续工作展望 .....	97
致 谢 .....	99
参考文献 .....	100
攻读博士学位期间取得的成果 .....	109

## 主要符号表

additive margin	加性间隔
AM-Softmax	带有加性间隔的 Softmax 交叉熵损失函数
bag-of-words	词袋模型
bias term	偏置项
center loss	中心损失
class agent	类代理
class imbalance problem	类不均衡问题
CMC	累积匹配特征, Cumulative Match Characteristic
DIR	检测与辨识率, Detection and Identification Rate
distractor set	干扰集
FAR	错误接受率, False Accept Rate
few-shot learning	少量样本学习
FMR	错误匹配率, False Match Rate
FNMR	错误不匹配率, False Non-Match Rate
FRR	错误拒绝率, False Reject Rate
FTA	获取失败率, Failure-to-Acquire Rate
gallery set	注册集
genuine match	真匹配
gradient saturation	梯度饱和 (也称梯度弥散)
hard sample mining	难例挖掘
HIK	直方图交叉核, histogram intersection kernel
hinge loss	合页损失
hyper-parameter	超参数
hypersphere embedding	超球面嵌入

ICA	独立成分分析, Independent Component Analysis
imposter match	假匹配
LBP	局部二值模式, Local Binary Pattern
LDA	线性判别分析, Linear Discriminate Analysis
LSE	指数和的对数, Log-Sum-Exp
LSTM	长短期记忆, Long Short-Term Memory
mean shift	均值漂移
momentum	动量项
moving average	滑动平均
non-target logit	非目标分数
PCA	主成分分析, Principle Component Analysis
probe set	查询集
Rank 1	首选正确率
ReLU	线性整流函数, Rectified Linear Unit
ROC	受试者工作特征, Receiver Operating Characteristic
SGD	随机梯度下降, Stochastic Gradient Descend
shattered gradient problem	破碎梯度问题
SVM	支持向量机, Support Vector Machine
target logit	目标分数
TPR	真正率, True Positive Rate
vanishing gradient problem	梯度消失问题
warm start	热启动
zero-shot learning	零样本学习

## 第一章 绪论

### 1.1 研究意义

人脸认证是一种让计算机能自动判断照片中人脸身份的技术，被广泛应用于安防、考勤、自助服务、信息安全等领域，例如我国的天网监控系统利用人脸认证技术对逃犯进行搜捕；在企业、住宅、商铺中安装人脸认证系统可以完成员工考勤、防盗和熟客提醒等功能；银行的 ATM 机上安装人脸认证系统可以防止他人盗刷银行卡；在手机和电脑的应用程序中使用人脸认证技术可以在密码的基础上提供进一步的安全保障。人脸认证过程通常只需要一个摄像头即可完成，与指纹、虹膜、静脉等其他生物特征相比，人脸认证的成本更加低廉且不需要人的接触与配合，这些因素使得人脸认证的应用更加广泛。

近年来，随着深度学习的快速发展，计算机对图像的理解能力有了质的飞跃。使用传统方法的人脸认证系统往往还需要人的配合才能达到实际可用的准确率；在使用深度学习技术后，人脸认证有了在非配合情况下应用的可能，这大大扩展了人脸认证技术的可用范围。现如今人脸认证已经可以取代人眼进行身份辨别，在一个具有百万人脸的数据集 MegaFace 上的评测结果显示，在千分之二误识率条件下，人眼只有 41% 的正确率，而最先进的人脸认证系统在更难的百万分之一误识率条件下，仍然可以达到 99% 以上的正确率。这样的性能不仅使人脸认证系统能够更广泛地在日常生活中使用，更在支付、保密单位身份认证等安全要求极高的任务上有了用武之地。

从研究的角度来看，人脸认证技术是计算机视觉与模式识别领域内的一个研究热点。大多数人脸认证方案采取的是度量学习框架，即学习出来一个距离度量模型。在度量学习框架内，人脸图像因其数据量大、模式较为稳定，长期以来人脸认证被当作度量学习算法的标杆性任务。一个优秀的度量学习算法必须经过人脸验证的考验才会被广泛认可，对人脸识别的研究也必须建立在对度量学习这一模式识别问题的深刻认识上。因此研究人脸认证的同时也是在对度量学习进行研究，研究者需要同时具备良好的数学能力与娴熟的工程能力，非常具有挑战性。

### 1.2 问题描述

一个完整的人脸识别系统包含图像采集、图像预处理、人脸检测、人脸对齐、特征提取、相似度计算等步骤，利用最后得到的相似度，根据任务的不同又分为人脸身份确认、人脸认证、人脸检索、人证合一等应用。本学位论文专注于人脸

认证这个任务的核心部分：特征提取，对于其他步骤和模块本文均使用目前较为先进的开源算法来实现，本文中不做深入的研究。

人脸认证模块要求输入两张人脸图像  $I_1, I_2$ ，并给出这两张人脸图像的相似度  $S(I_1, I_2)$ ，进而根据它们之间的相似度来判断这两张人脸图像是否来自同一个人。为了得到相似度，最常见的做法是从两张人脸图像上分别提取特征  $F(I_1)$  和  $F(I_2)$ ，然后使用这两个特征向量夹角的余弦值

$$S(I_1, I_2) = \frac{\langle F(I_1), F(I_2) \rangle}{\|F(I_1)\| \|F(I_2)\|} \quad (1-1)$$

作为两张人脸图像的相似度，其中  $\langle \cdot, \cdot \rangle$  为内积操作、 $\|\cdot\|$  为向量的  $L_2$  范数。

本论文中使用的特征提取  $F(I)$  步骤分为两步，第一步是图像预处理，首先将图像中的人脸与面部关键点检测出来并根据面部关键点将人脸对齐到竖直状态；第二步为卷积神经网络提取特征，卷积神经网络通过一系列的卷积、池化、激活函数层将对齐好的人脸图像逐步地抽象出更加贴近目标函数要求的特征。

卷积神经网络的优化需要由损失函数来引导，损失函数的输入为图像  $I$  通过卷积神经网络输出的特征  $F(I)$  以及图像  $I$  的身份标签  $L(I)$ ，损失函数会计算得到对特征的梯度，通过反向传播算法，这些对特征的梯度将逐层反传回前边的卷积、池化、激活函数层中，得到对整个卷积神经网络参数的梯度，并由随机梯度下降法进行迭代更新，直至收敛得到一个最终的人脸认证模型。

由此可见，损失函数是指导整个神经网络训练的核心所在。如今的各个视觉任务如识别、检测、分割等使用的主体网络结构大多相同，人脸认证模型使用的主体网络往往也是在通用图像识别领域验证过的网络结构，各人脸认证模型的主要区别主要体现在对损失函数的设计上。人脸认证模型的损失函数要保证同一个身份的样本之间的相似度尽可能高，而不同身份样本之间的相似度尽可能低，如何通过数学表达式将这两个要求建模成损失函数的形式，使其能够高速、有效地指导神经网络完成训练成为了人脸认证研究的核心问题之一。

### 1.3 主要困难

本论文所做工作集中于损失函数的设计上，人脸数据集有数据量大、身份多、各类别样本数量严重不均衡等特点，在设计损失函数时需要综合考虑这几个特性。因为数据量巨大，所以基于样本对的度量学习损失函数会遇到采样困难的问题；因为身份数量多，所以基于分类的损失函数会遇到分类器矩阵过大、类内信号不充足的问题；因为各类别样本数量严重不均衡，所以会出现模型更加倾向于样本数量多的类别，而一些样本数量少的类别所蕴含的模式甚至会被忽视。这些问题

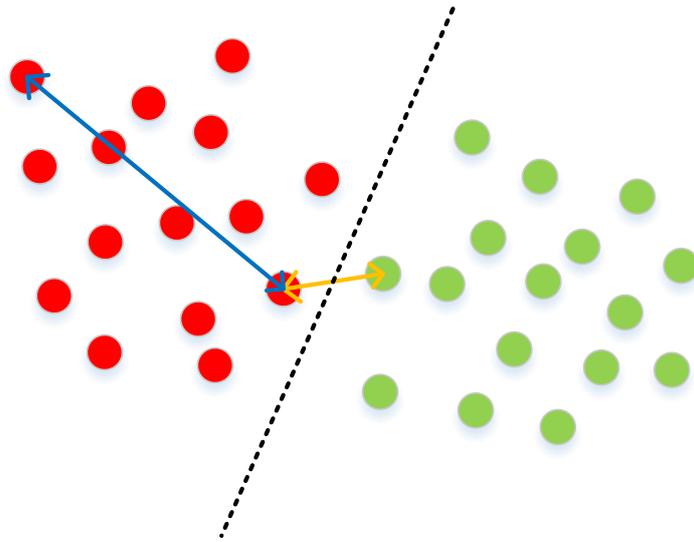


图 1-1 分类问题与度量学习问题的区别：分类模型只需要找到分界面将两类分开即可（虚线），而度量学习需要使类内相似度大于类间相似度（箭头）。

都是设计损失函数时避不开的，而针对这些问题来改进损失函数往往也能带来可观的性能提升。

人脸认证与通用的图像识别的不同之处还在于大部分人脸认证是一个开集任务，即训练样本的类别与测试样本的类别完全不重合。这类似于图像识别中的零样本（zero-shot）识别，然而在通用图像识别中，少量样本（few-shot）识别都是尚未解决的任务，零样本识别的识别率更是大多只有 30% ~ 40% 左右的水平。幸而人脸图像的模式（例如大眼、厚唇、宽鼻等）比较稳定且泛化性好，而且在进行了人脸对齐之后，任务的难度大大降低，这些因素使得零样本识别也成为了可能。

人脸认证本质上是一个度量学习的任务，度量学习与分类的区别在于度量学习进行的是特征的比对而分类是寻找分界面。如图1-1所示绘制了度量学习与分类问题的区别，从图上可以看出两类样本可以被一个分界面清晰地分开，但这个分界面的两侧的样本之间的距离（黄色箭头）可能会非常小，有可能会远小于类内样本之间的距离（蓝色箭头）。由此可见，度量学习要求更高的类间间隔。在分类问题中，往往也会要求训练类别之间拉开一定的间隔，然而这个间隔是为了提升模型的泛化能力、减小结构风险而引入的。在度量学习中，即使不考虑泛化性能只在训练集上进行度量也仍旧需要非常大的间隔，而分类任务如果不考虑泛化误差，其间隔甚至可以为 0，这也就意味着度量学习是比分类更加困难的任务。

## 1.4 研究现状

本节将对深度学习和人脸认证这两个研究方向的历史与发展现状进行简单的介绍，深度学习毫无疑问现在已经成为绝大多数人脸认证解决方案的核心技术，而人脸认证作为计算机视觉中重要的任务之一，在深度学习的发展过程中也起到了推波助澜的作用。

### 1.4.1 深度学习的发展现状

深度学习是在人工神经网络的基础上发展起来的，是人工神经网络的深层版本，而人工神经网络起源于联结主义<sup>①</sup>的哲学思想，早在公元前 300 年，受到柏拉图的启发，亚里士多德就曾经针对大脑的记忆与联想提出了联结主义的四大定律<sup>[1]</sup>：

- (1) 临近性：时空中相接近的物体与事件在大脑中倾向于连接在一起；
- (2) 频率性：两个事件共同发生的次数与它们之间的连接强度成正比；
- (3) 相似性：对于一个事件的思考将会激发与之相似的事件的思考；
- (4) 对比性：对于一个事件的思考也会激发与之相反的事件的思考。

在 2000 年后的今天，这四条定律仍然被当作机器学习方法的基本假设，如果将定律中的事件改写为特征，那么这四条定律所描绘的正是一个度量学习的模型：对于语义相似的特征要缩短它们之间的距离，同时也要拉长语义不同的特征之间的距离，出现频率高的样本会受到更多的训练，最终相似的特征会聚在一起，而不类似的特征会远离。

联结主义是神经网络的思想基础，而神经网络的数学模型的起源来自于神经医学家 Warren McCulloch 和数学家 Walter Pitts 所构建的基于逻辑电路的神经网络模型<sup>[2]</sup>：MCP 神经元模型。MCP 神经元模型是受神经细胞的响应机制的启发，其基本思想是一个神经元可以接受多个输入，这些输入的数值在乘以对应的权重后被加在一起，然后通过一个阈值来判断这个神经元是否受到激发。MCP 神经元模型与现代的神经网络的单层结构已经非常类似了，均包含线性计算与激活函数两个组成部分，它的提出为后世的感知机与神经网络的出现有很大的启发作用。

第一个提出权重可以从样本中学习得到的方法是 Hebbian 学习准则，其数学表达式非常简单：

$$\Delta\omega_i = \eta x_i y \quad (1-2)$$

<sup>①</sup> 联结主义 (Associationism) 与连接主义 (Connectionism) 是不同的概念，联结主义更多的是关于高层语义之间的关系，连接主义脱胎于联结主义，描述的是从底层表达到高层语义的全部连接关系。现代神经网络遵从的是连接主义思想。

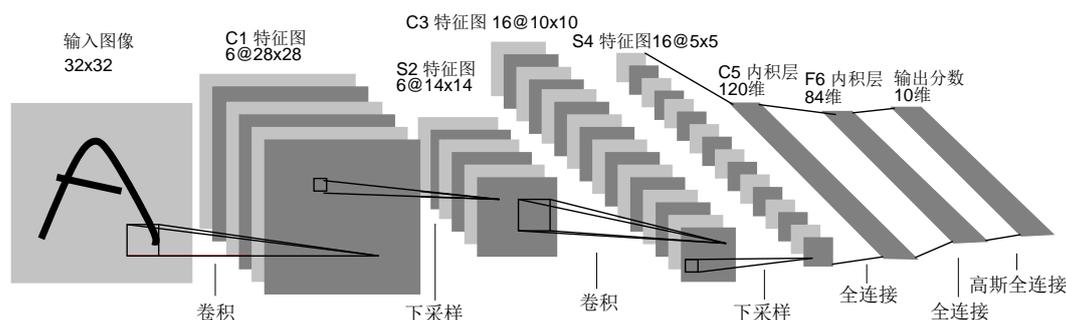


图 1-2 Lecun 教授在 1998 年提出的 LeNet-5<sup>[9]</sup> 的网络结构。

其中  $\Delta\omega_i$  是第  $i$  个神经元的突触权重  $\omega_i$  的变化量、 $x_i$  为输入信号、 $y$  为目标输出响应、 $\eta$  为学习率。用文字表述则为 Hebbian 学习准则服从联结主义的频率性原则，即两个单元之间的连接应随这两个连接之间共同发生次数的增加而增强。Hebbian 学习准则是联结主义模型通过学习的方法来得到权重的初步尝试，尽管这个学习方法后来被证明并不稳定<sup>[3]</sup>，但作为第一个学习模型的发明者，Hebb 仍然被认为是神经网络之父。

现代神经网络模型的基础模型为感知器模型<sup>[4]</sup>，直到现在也有人将深层神经网络模型称作多层感知器模型。感知器模型是由 Frank Rosenblatt 发明的单层二元分类器，感知机的权重是从样本中学习得到的，其学习方法被称为感知器学习算法。感知器学习方法被证明在线性可分的情况下，它一定能找到两个类别的分界面。

然而事情的发展不总是一帆风顺的，神经网络的发展也曾经经历了多次兴起和衰落。在 1969 年 Minski 在他著写的关于感知器的书籍中<sup>[5]</sup>，提出了感知器可以完成与、或、非这类逻辑的分界面的求解，但对于异或和同或这种逻辑的分界面就无能为力了。单层的感知器作为一个线性模型，确实无法解决异或这种非线性问题。虽然多层的感知器可以解决异或问题，但无法用单层感知器的学习方法来学习。这是一个数学论断，指出的问题也是真实存在的，但这个论断却导致了神经网络的研究在 1970 年代的沉寂。

转机出现自 1986 年 Rumelhart 等人在 Nature 上发表了一篇使用反向传播算法学习神经网络权重的文章<sup>[6]</sup><sup>①</sup>，该方法使得训练多层神经网络成为了可能，神经网络研究掀起了第二次浪潮。其中比较标志性的网络有 Lecun 教授提出的 LeNet 网络<sup>[9]</sup>（如图 1-2 所示），这已经与目前使用的卷积神经网络结构非常接近了，该

① 这并不是反向传播算法的第一次提出，实际上早在 1963 年和 1969 年出版的多部著作<sup>[7,8]</sup>中反向传播算法就已经被提出了，但由于当时信息传播较慢和研究者较少的缘故，该方法并未受到重视。

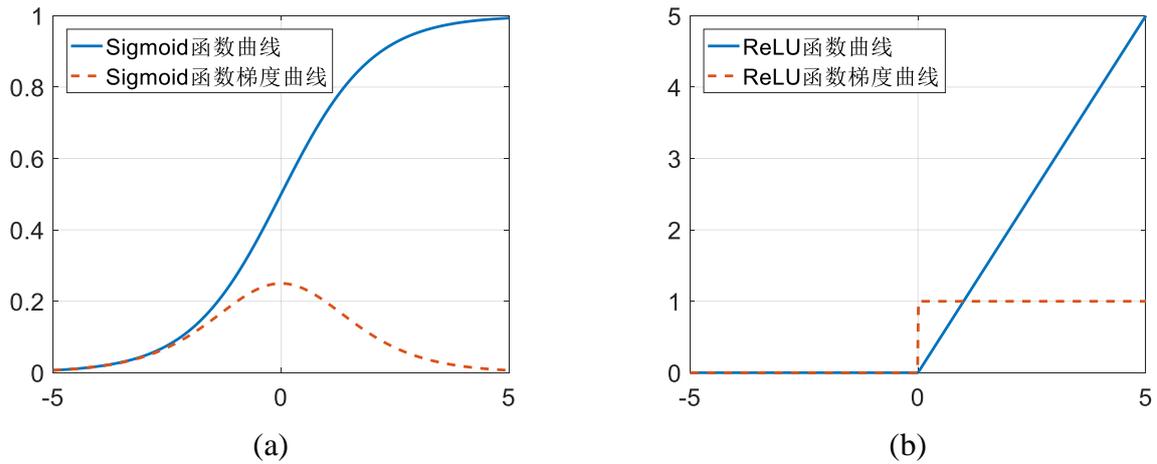


图 1-3 (a)Sigmoid 激活函数的函数曲线与梯度曲线；(b) ReLU 激活函数的函数曲线与梯度曲线。

方法被成功用在了银行、邮局的表单自动识别程序中。这期间 Schmidhuber 教授提出的 LSTM 网络<sup>[10]</sup>在时序信号如音频、自然语言的处理中也取得了大量成果。

可惜的是，因为当时使用的激活函数 Sigmoid 存在梯度弥散的问题，导致超过三层的神经网络就无法训练了，因此在 2000 年后神经网络的研究再一次陷入沉寂。当时人们把目光更多地投向了支持向量机这种凸模型中，而诟病神经网络这种非凸模型没有理论保证其能够收敛。直到 2006 年，Hinton 教授在 Science 上发表文章<sup>[11]</sup>，提出使用限制玻尔兹曼机来逐层无监督式地训练神经网络作为初始化，而后使用反向传播算法进行微调的策略，成功地训练出了多层神经网络，从此神经网络进入了深度学习时代。

2006 年 Hinton 教授提出的方法虽然能训练超过 3 层的神经网络，但使用的逐层初始化的方式仍旧比较繁琐，没有从根本上解决梯度弥散现象。梯度弥散现象本质上是由于 Sigmoid 函数的梯度不稳定导致的，Sigmoid 函数在接近 0 处的导数较大，而两端的导数极小接近于 0，如图 1-3(a) 所示。这导致在训练初期整个网络在快速变化时一旦输入的幅度过大，则反向的梯度就会极小，此时再用梯度对权值进行更新带来的改变就会微乎其微，导致网络再也无法恢复。Sigmoid 函数是由大脑神经元的激活模型启发得到的，而人们至今对大脑神经元的工作原理也不甚清楚，所以如何修改人工神经网络的优化算法也无从谈起，人工神经网络受生物神经网络的启发而诞生，却也因为生物神经网络研究的停滞而受到限制。

幸而这一次神经网络的研究并未陷入沉寂就迎来了转机，在 2009 年 Lecun 教授的文章中提出了使用 ReLU 函数  $\max(x, 0)$  作为神经网络的激活函数<sup>[12]</sup><sup>①</sup>，ReLU

① 在文章中作者使用的是绝对值函数  $\text{abs}(x)$ ，但也提到了使用 ReLU 函数的性能与绝对值函数类似。

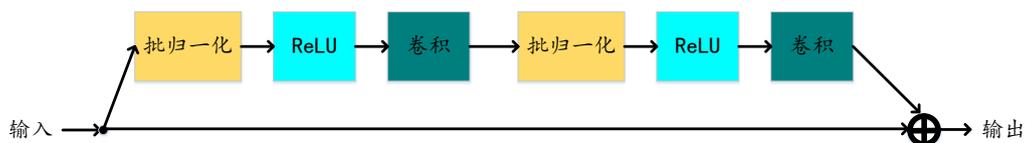


图 1-4 残差卷积神经网络中的残差模块。

函数的梯度并不随着输入幅度的改变而改变，如图1-3(b)所示。因此 ReLU 激活函数能够在训练的全程保持梯度幅度的稳定，这一小小的改动就解决了神经网络的梯度弥散问题，使得第三次神经网络浪潮得以延续下去。

然而在这段时间对于深度神经网络的研究仍然局限于几个研究组的小圈子中，大量学者并没有接受神经网络。直到 2012 年的 ImageNet 图像分类竞赛上，由 Hinton 教授指导的 Alex Krizhevsky 博士使用一个深层卷积神经网络 AlexNet<sup>[13]</sup> 赢得了冠军，而且领先第二名超过 10 个百分点（15.3% 与 26.2%），这一巨大的提升轰动了学术界与工业界。在无可辩驳的优势面前，人们对于神经网络的研究进入了爆发期。以 AlexNet 为界限，深度学习被分成了两个时代，第一个时代是以限制玻尔兹曼机和自动编码器为代表的逐层预训练方法；第二个时代因为 ReLU 激活函数的出现，逐层预训练的方式被抛弃，改为整体直接训练来得到模型。所以又有人将后 AlexNet 时代称之为“第二波深度学习浪潮”。

ReLU 函数虽然解决了梯度弥散问题，但它仍存在神经元“凋亡”的问题，即如果输入全部为负数，则 ReLU 激活函数的输出全部为 0，此时反向的梯度也会全部为 0，从而使网络无法更新，这种情况比较罕见但仍旧存在。为了解决这一问题何恺明博士提出了参数化的 ReLU 函数<sup>[14]</sup>，在输入小于 0 时激活函数输出  $ax$  而不是 0，这样即使输入全部为负数，神经网络仍然能获得梯度进行优化。

另一个针对梯度弥散现象的改进是批归一化<sup>[15]</sup>，它在每一层神经网络中都加入了归一化操作，使得输入激活函数的值均具备 0 均值和 1 标准差。这样不论训练多少层的神经网络，其梯度都如同训练单层神经网络一样不会产生明显的梯度弥散现象。使用该方法后，甚至连 Sigmoid 函数也能够在深层神经网络中继续使用。可惜 Sigmoid 函数的性能仍旧不如 ReLU 函数，所以现在使用最广的激活函数仍旧是 ReLU 函数。

何恺明在他的文章<sup>[14]</sup>中还发现了一个现象：深层神经网络在大概 20 层时就会达到性能的极限，这与“越深层的网络表达能力越强”的常识不符。何恺明等人针对这一问题提出了残差神经网络<sup>[16]</sup>，残差神经网络基于一个基本的方法论：如果第  $N+1$  层的神经网络是一个等值函数，那其性能应该至少不低于前  $N$  层构成的神经网络。残差神经网络的每一个模块都由一个等值函数与一个普通的卷积、批

处理、激活函数构成的非线性表达函数相加得到（如图1-4所示）。这样由等值函数这一路可以保证深层的神经网络的性能至少不会退化，而非线性表达函数这一路则可以提升性能。在实验中，作者们甚至构建了一个 1000 层的神经网络，取得了比 100 多层的神经网络更好的性能<sup>[17]</sup>。残差神经网络在 ImageNet2015 竞赛上取得了冠军，并获得了 CVPR 2016 的最佳论文奖。

在这一波深度学习浪潮中，计算机视觉的各个任务上均得到了极大的提升，图像识别<sup>[13-15,18,19]</sup>、图像目标检测<sup>[20-23]</sup>、图像语义分割<sup>[24,25]</sup>、图像生成<sup>[26-29]</sup>均达到了商业化应用的程度，也因此催生了一系列的创业公司，带来了大量的工作岗位，学术界也因此蓬勃发展，CVPR、NIPS 等会议的投稿数量与参加人数都呈指数式增长。

回顾神经网络几十年的发展，其中有多次热潮：感知器、反向传播、深度学习，也有两次沉寂：非线性问题、梯度弥散问题。起起落落之间可以看到一些人的坚持：正是 Hinton 教授几十年来对神经网络的坚守，在低谷时不放弃、在高潮时坚持基础研究，才迎来了反向传播算法和深度学习这两波热潮。这对新一代的研究者有着很好的教育意义。目前深度学习虽然在一些任务中取得了超越人类的效果，但其仍旧没有脱离拟合函数的机器学习范式，在推理、主动学习、无监督学习等方面还看不到有效的解决方案，人工智能这一终极目标在深度学习体系下目前仍看不到希望，仍需研究者们寻找新的研究思路。

## 1.4.2 人脸认证的发展现状

人脸认证，或者更广义的人脸识别是计算机视觉与模式识别领域的经典研究课题，因为人脸识别的应用范围很广，所以有大量的研究者在从事这方面的研究。根据信号源的不同，人脸识别算法又可以分为基于可见光图像的人脸识别、基于（主动/被动）红外图像的人脸识别与基于 3D 成像的人脸识别等，从机器学习角度来看这几个类别所使用的方法是互通的，但在特征提取方面不同的信号源有着不同的处理方式。其中基于可见光图像的人脸识别因为数据最多，研究得也最透彻，本文将着重介绍基于可见光的人脸识别技术。

基于可见光的人脸识别技术从特征提取的角度来看又分为几何特征、局部特征、整体特征三个大类，这三个大类并不是严格分离的，各种人脸识别算法也经常会从其他类别的算法中取长补短，因此这三大类的界线并不明显。

### 1.4.2.1 基于几何特征的人脸认证

面部的特征主要有面部器官的外观、面部器官之间的间距以及皮肤头发的纹理等。早期计算机的计算能力并不强大而且人们对“外观”、“纹理”这些概念的数

学表达尚不明确，因此早期的人脸识别主要是针对人脸的几何特征进行建模。早期的方法通过面部灰度的直方图投影等方式<sup>[30-32]</sup>或是通过模板匹配的方式<sup>[33]</sup>来辅助确定面部器官的位置，进而提取出面部器官相关的一系列几何特征，如各器官之间的间距、三个器官之间的夹角、某几个器官关键点所包围的面积等。

另外还有一个比较有意思的思路是从侧面人脸的剪影曲线中提取结构特征<sup>[34,35]</sup>，侧面人脸曲线非常容易提取，而且不容易受到光照、表情等因素的影响，可以算是一个比较有效的特征提取方式。

可以看到，早期的人脸识别算法是在计算能力非常有限、对视觉模式的理解较为浅薄情况下的初步探索。这些算法往往只能利用到很少的一部分人脸的特性，而对面部器官的外观、细节纹理等需要较高层语义的信息完全没有建模能力，因此这些算法的性能非常有限。面部器官的位置的提取在这些早期方法中仍需要人的配合，因此并不能算是自动化的算法，即使是在几十年后的今天，使用基于深度神经网络的算法在侧脸的面部器官定位方面仍有缺陷，因此基于几何特征的人脸识别算法研究在近几年慢慢地销声匿迹。从直觉上来看，人类在识别面部的时候也不会去精确地测量面部器官之间的距离、角度、覆盖面积这些物理量，而更多的是从外观上来识别人脸。

然而这类技术并不是没有用处，基于几何特征的人脸识别算法直接催生了面部器官定位这一任务。在现在的人脸认证框架中，仍需要对面部器官进行定位，然后根据面部器官的位置将人脸旋转缩放到统一的角度和尺度上来，从而减少后续特征提取的难度。

### 1.4.2.2 基于整体特征的人脸认证

真正的自动化人脸识别方法起源于 1991 年麻省理工学院的 Turk 与 Pentland 提出的特征脸算法（EigenFace）<sup>[36]</sup>，这一算法面世标志着人脸识别进入了自动处理时代。特征脸算法以整张人脸图像作为输入，通过计算一个降维矩阵来提取人脸图像的特征，由于降维是对整张图进行的，因此降维矩阵可以视为一系列的图像模板。麻省理工学院的 Brunelli 和 Poggio 于 1992 年做了一个对比实验，他们比较了基于模板的人脸识别与基于几何特征的人脸识别的性能，得到了一个比较确定的结论：基于模板匹配的方法在识别率和速度上都优于基于几何特征的人脸识别。在这个结论与特征脸算法共同作用下，基于结构特征的人脸识别方法的研究基本终止，基于整体特征（即模板匹配）的人脸识别算法慢慢成为主流。

特征脸算法是一个无监督算法，它仅仅只是将人脸图像拉成的向量在空间中拉得更开，其中并没有判别性的因素。基于线性判别分析的判别脸（FisherFace）<sup>[37]</sup>尝试着将监督信号引入降维矩阵的学习中，线性判别分析中引入了类内散度矩

阵和类间散度矩阵两个概念，并设计了一个优化目标使得类内散度尽量较小而类间散度尽量较大，从而能够得到更加具有判别性的降维矩阵。时至今日，最先进的人脸识别算法仍然延续了这个思路来使得类内距离变小而类间距离变大，区别只是在于特征提取变为非线性特征，散度矩阵变成了更加直接的距离优化。

独立成分分析 ICA<sup>[38]</sup> 将 PCA 中的相关性分析替换为了独立性分析，从其可视化图像上来看可以清晰地看到 ICA 训练得到的各个模板中包含有不同的人脸器官，而这些器官是相互独立的特征表征，因此 ICA 方法更加适合进行人脸识别任务。

随着机器学习的发展，支持向量机 SVM<sup>[39,40]</sup>、Boosting<sup>[41,42]</sup>、流形学习<sup>[43]</sup> 等方法都被尝试着用在人脸识别上，人脸识别的性能随着时间在稳步提升。在 2008 年，John Wright 博士与马毅教授提出了基于稀疏表示的人脸识别算法<sup>[44]</sup>，稀疏表达有着优美的理论，而且对于带有遮挡的人脸图像具备特别优秀的识别能力。

#### 1.4.2.3 基于局部特征的人脸认证

20 世纪末与 21 世纪初，图像的局部描述子得到了极大的发展，Gabor 特征<sup>[45]</sup>、SIFT 特征<sup>[46]</sup>、LBP 特征<sup>[47]</sup>、HoG 特征<sup>[48]</sup> 相继被提出。人脸识别的发展与计算机视觉和模式识别的发展是分不开的，因此这些方法也被应用到了人脸识别任务上。人脸上的各个器官的模式实际上差别非常巨大，因此几个整体特征提取的方法往往会在各个位置上产生相互的干扰，从人脸的各个位置分别提取局部特征然后再进行特征融合的办法可以解决这个问题。

实际操作中往往需要先使用人脸关键点检测算法得到多个人脸上的关键部位的位置，或者直接将人脸分成若干个矩形小块，然后在这些位置附近取各种特征，最后将这些特征直接简单地串联、或是使用词袋算法<sup>[49]</sup>、Fisher 向量编码<sup>[50]</sup> 算法等算法将特征融合作为人脸特征。

这类方法的优势在于利用了人脸各位置模式不同的特性，而且这类算法往往充分考虑了光照、旋转、尺度变化下的不变性，对于这些变化较为鲁棒。其缺点在于特征均为手工特征，其特征判别性是没有保证的，而且多个位置的局部特征进行串联或者编码都会导致特征的维度巨大，给特征比对带来不便。后来又有一系列需要学习的浅层特征如 LE<sup>[51]</sup>、PCANet<sup>[52]</sup> 等，但这类非监督的特征与手工特征相比并没有本质提升，其判别能力仍旧无法保证。

局部特征的巅峰期在 2013 年到 2014 年，微软亚研院的研究人员使用超高维度（10 万维）的 LBP 特征<sup>[53]</sup> 在 LFW 数据集上达到了 95.17% 的识别率，而后香港中文大学的陆超超使用贝叶斯学习的方法对 LBP 特征进行学习<sup>[54]</sup>，达到了 98.52% 的识别率。之后由于深度学习的出现，对于局部特征的研究逐渐陷入停滞。

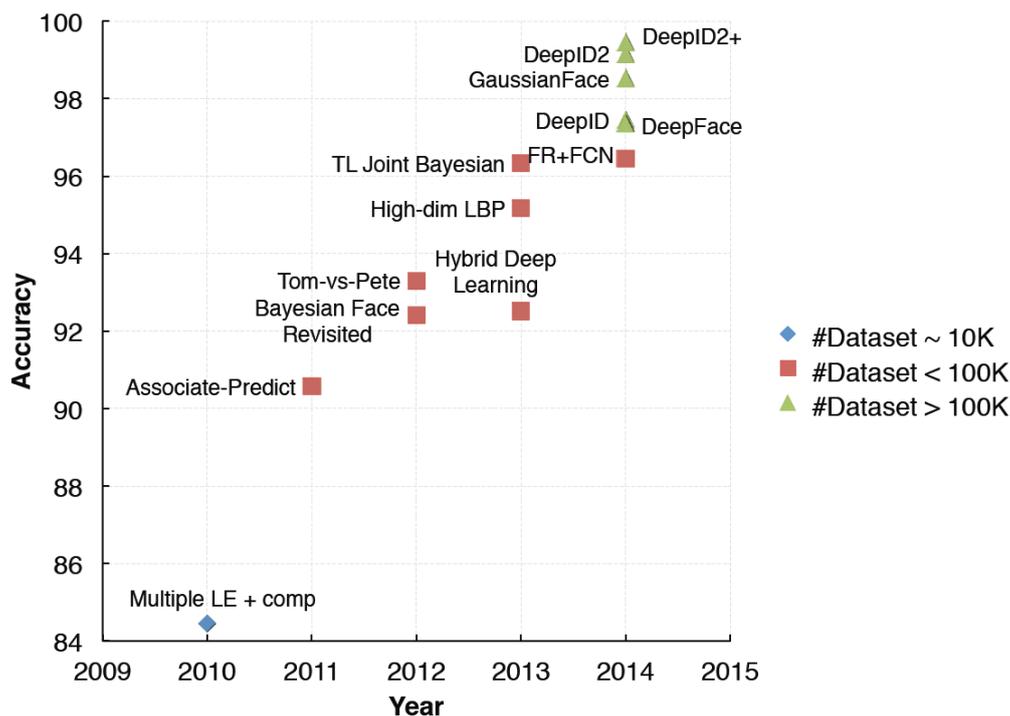


图 1-5 LFW 数据集<sup>[55]</sup>上的各代算法的识别率<sup>[56]</sup>。

#### 1.4.2.4 基于深度学习的人脸认证

随着 2012 年深度学习在 ImageNet 竞赛上的突破性表现<sup>[13]</sup>，深度学习进入飞速发展阶段，人脸识别也因此而受益。Facebook 公司在 2014 年提出了 DeepFace 模型<sup>[57]</sup>，在 LFW<sup>[55]</sup>上达到了 97.35% 的识别率，这与人眼的识别率 97.5% 已经非常接近了，之后香港中文大学的孙伟博士和汤晓鸥教授提出了 DeepID 系列文章<sup>[58,59]</sup>，将分类损失函数与度量学习损失函数结合在一起，在 LFW 上达到了 99.15% 的识别率，大大超过了人眼的能力。在 2015 年，Google 公司<sup>[60]</sup>、牛津大学的 VGG 组<sup>[61]</sup>、百度公司<sup>[62]</sup>相继提出使用三元组度量学习的方法来训练深度神经网络，在 LFW 上的性能此时已经达到 99.6% 到 99.8%，LFW 上的性能逐渐达到饱和，慢慢地退出了历史舞台（如图 1-5 所示）。

由于度量学习的方法存在着难以采样的缺点，学术界逐渐将目光转回基于分类的损失函数。2016 年温研东博士提出在 Softmax 损失函数的基础上添加中心损失作为约束来优化类内距离<sup>[63]</sup>，2017 年刘威杨博士提出了 SphereFace<sup>[64]</sup>，将间隔的概念引入了分类模型。分类的损失函数与度量学习损失函数相比能够学习到类内分布，因此在数据有部分缺失的情况下也能得到很好的性能，而且避免了度量学习算法中的采样步骤，直接将样本顺序输入网络即可，大大降低了实现难度。分类的损失函数的缺点在于需要维护各类的权重向量，在训练大型数据集时对显

存的消耗较大。

基于深度学习的人脸识别所使用的网络结构大多为通用人脸识别任务上经过广泛验证的网络结构,如 AlexNet<sup>[13]</sup>、VGGNet<sup>[18]</sup>、ResNet<sup>[14]</sup>等。因为人脸识别在工业界应用广泛,为了提升其在手机端和嵌入式芯片中的速度,轻量化模型<sup>[65,66]</sup>也被广泛应用在人脸识别领域中。此外也有一些专门针对人脸设计的网络结构,例如 DeepFace<sup>[57]</sup>和 DeepID2<sup>[59]</sup>中使用了一种局部连接的卷积层来适应人脸不同位置具备不同模式的特性;条件卷积神经网络<sup>[67]</sup>针对人脸图像可能来自不同模态的信号源的特性,设计了一种动态选取卷积核的方法来自动提取不同模态的特征。

基于深度学习的算法重新回到了整体特征提取的方式,这是因为深度学习的特征提取是逐层由局部到整体逐渐过渡的,它既有局部特征只关心局部模式的优势,又将整体特征整合在内,所以其表达能力相比于局部特征又有了很大提升。

目前基于深度学习的人脸认证方法虽然识别率已经达到商用水平,但仍然存在一些问题值得研究:

(1) 目前在损失函数层面存在度量学习和分类两个流派,它们各自具备一些优缺点,如何避免他们的缺点,并将两者的优点相结合是目前以及未来一段时间的研究重点。

(2) 人脸认证使用的深度学习模型仍然是通用人脸识别任务上经过广泛验证的网络结构,针对人脸专门设计网络结构的研究非常稀少,针对人脸这种特殊结构而设计网络结构也是一项值得研究的课题。

(3) 目前人脸检测、人脸对齐、人脸特征认证这三个步骤大多还是独立进行的,关于如何将这几个步骤整合到一起,形成一个端到端的模型仍需要进一步的探索。

(4) 基于视频或者人脸集合的人脸认证在实际应用中更为常见,如何利用好多张图像进行特征提取是非常值得研究的内容。

## 1.5 主要创新点

本论文深入分析了第1.3节所提到的多种困难,并针对它们提出了一系列的解决手段:

(1) 发现了使用基于 Softmax 交叉熵损失函数作为人脸认证模型的损失函数时,训练与测试阶段使用的相似性度量不匹配的问题,在尝试解决这一问题时又发现了直接套用余弦相似度造成的模型不收敛问题。本论文的小节3.1和小节3.2深入分析了这两个问题,提出了两个数学理论来解释它们,并给出了解决方案。

(2) 针对基于样本对的度量学习损失函数遇到的采样困难的问题,通过借鉴分类的损失函数的一些思想,小节第四章对度量学习的损失函数进行了改造,使其

既能保持原有的对相似性度量的优化与增大类间间隔的特性，又具备了分类的损失函数无需采样的优点。

(3) 针对图1-1所反映出的问题，在第一个创新点的基础上本文引入了类间间隔参数 (小节5.2)，进一步加大了类间的距离，使其更加适合人脸认证任务。该算法是目前学术界领先的 (state-of-the-art) 人脸认证损失函数。

(4) 本文对传统的 Softmax 交叉熵损失函数以及新提出的一系列改进进行了大量的理论分析，有别于传统理论更多的偏向概率解释，本文中提出的理论分析大多是基于特征、权重之间的几何关系与统计量进行的，由于特征与权重相比于 Softmax 概率更加底层，因此能够得到更多的数学结论。

## 1.6 章节安排

本文专注于针对人脸认证任务的损失函数设计，本文的章节结构安排如下：

第二章介绍了深度学习与人脸认证的基础理论和基本概念，并对与本文工作高度相关的几个算法进行简单的介绍，这一章的目的是使读者对整个研究领域有一个初步的认识，以便于理解本文所提出的方法。

第三章介绍了在使用深度学习进行人脸认证时将特征映射到  $L_2$  超球面上的必要性、映射方法以及优化方式，并通过一系列的数学推导描述了新提出的损失函数的一系列性质。

第四章介绍了如何规避传统的度量学习方法在大数据上应用时产生的难以采样的问题，本文对两种传统的度量学习方法进行了改造，解决了其难以采样的问题，也取得了更好的性能。

第五章在  $L_2$  超球面嵌入算法的基础上，引入了类间间隔参数，进一步加大了类间的距离。在这一章还介绍了类间间隔参数的几何意义，为了能够更加直观地展示本学位论文中各个损失函数的作用，在本章还绘制了一系列针对损失函数的可视化图像。

第六章回顾本文内容，阐述本文算法存在的不足之处，并为将来的工作提供一些可能的方向。

## 第二章 基础理论与相关工作

本章将会对深度学习以及人脸认证的基础理论进行一些简单的描述，让初学者对这两个研究领域建立起基本的概念，以方便阅读后续章节的内容。同时本章还会对后续章节使用到的一些基础方法与对比的算法进行简单的介绍，对于要进行基于深度学习的人脸认证方法研究的读者，请仔细阅读本章并检索本章中提到的相关文献进行全文阅读。

### 2.1 卷积神经网络

在这一波深度学习浪潮中，基于卷积神经网络的图像处理得到了长足的进步，卷积操作的局部连接、全局共享的特性非常适合进行图像的分析，而级联的卷积操作则可以完成局部细节到整体语义信息的转换。本节将分三个部分来介绍卷积神经网络的各个模块以及其优化方式，由于篇幅限制，每个小节仅介绍目前最流行的几个模块，同时也会给出一些参考文献供读者进行扩展阅读。

#### 2.1.1 模型结构

卷积神经网络从传统的模式识别架构角度来看，可以分为特征提取和损失函数两部分，本小节将介绍特征提取部分的一些结构，由于篇幅有限，本小节仅介绍后文中使用到的一些层和它们的一些扩展。

##### 2.1.1.1 卷积层

卷积神经网络的特征提取的最重要的部件为卷积层，它与传统的信号处理中的卷积的定义非常类似，即使用一个较小的卷积核在较大的二维图像上滑动，将卷积核覆盖部分的图像像素与卷积核对应位置相乘后加在一起作为输出。如图2-1所

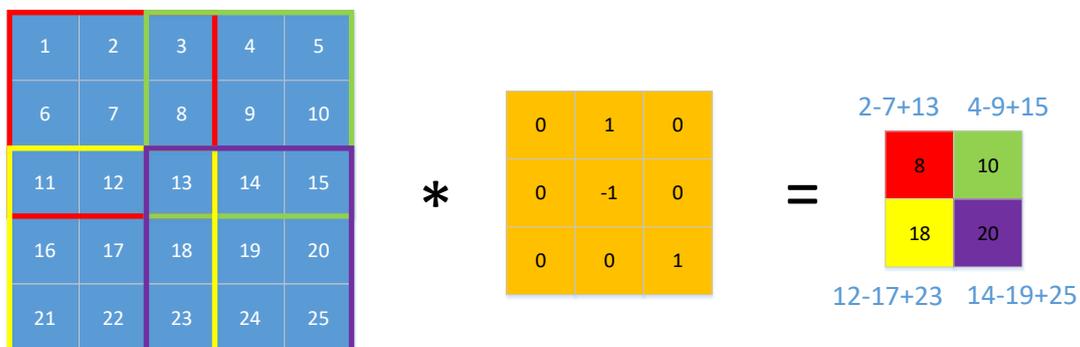


图 2-1 一个输入为  $5 \times 5$ ，卷积核大小为  $3 \times 3$ ，步幅为 2 的卷积操作示意图。

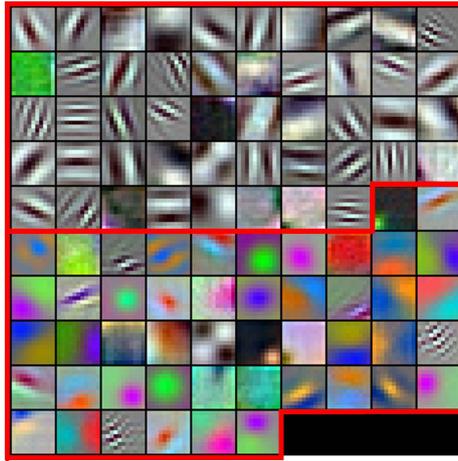


图 2-2 AlexNet<sup>[13]</sup> 的第一层卷积核的可视化图像。

示绘制了一个简单的卷积示意图，卷积核滑动的步长通常被设置为某个整数数值，图中即为步长为 2 的情况。图中的输出为一个  $2 \times 2$  的矩阵，若步长为 1 则输出就是  $4 \times 4$  的矩阵。此外，如果在步长为 1 时还需要输入输出的长宽保持一致，还可以在输入特征图周围补一圈 0，这样输入的大小被扩充到了  $7 \times 7$ ，而输出大小就可以保持在  $5 \times 5$  大小。

注意图像实际上是一个三维信号，除了横向和纵向之外，还有一个维度叫通道，例如最常见的彩色图像格式即为 RGB 图像，它包含红绿蓝三个通道。经过卷积层之后，可以任意设置输出的通道数，假设卷积核大小为  $k_w \times k_h$ 、输入通道数为  $c_1$ 、输出通道数为  $c_2$ ，那么这个卷积层的卷积核参数就有  $c_1 \times k_w \times k_h \times c_2$  个参数。

通过简单的计算即可知道，当输入输出的通道数较多时，卷积核的参数数量也会变得比较大。因此一些学者提出在存储空间受限情况下，可以将卷积核分组，假设将  $c$  个通道分为  $n$  组（前提是  $c_1$  和  $c_2$  可以被  $n$  整除），那么该层卷积核的个数就变为  $n \times (c_1/n) \times k_w \times k_h \times (c_2/n)$  个参数，参数总量降了  $n$  倍。这种卷积方式被称为分组卷积<sup>[13]</sup> 或深度可分离卷积<sup>[65,68]</sup>。

分组卷积会使原本致密的通道连接之间解耦，从而导致不同组的特征表达相互解耦。如图 2-2 所示，AlexNet<sup>[13]</sup> 的第二个卷积层被分为了两组，因此在第一层的卷积核上我们可以发现第一组的卷积核几乎都为类似于 Gabor 滤波器的边缘滤波器，而第二组卷积核则充满了颜色信息。两组卷积核的分工差别非常显著，这样的解耦现象如果一直持续下去，会导致网络只能提取一些互斥的、简单的特征，因此在分组卷积之后通常需要接一个  $1 \times 1$  的卷积层来使得不同通道的特征之间能够进行相互交流。

一个卷积操作实际上可以分解为采样（取出特征图中的像素点）、交互（图像

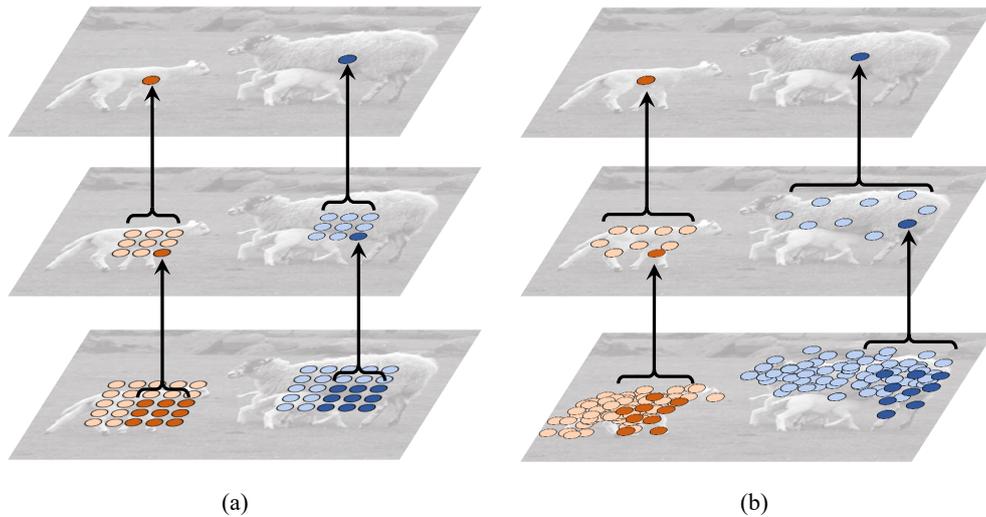


图 2-3 (a) 原始的卷积操作；(b) 可变形卷积。图片来自文献 [69]。

像素值与卷积核中的参数值之间进行计算)、聚合(将结果重新排列成图像)三个步骤。在原始的 Caffe 框架<sup>[70]</sup>中,卷积就是通过类似的操作(im2col-矩阵乘法-col2im)转换为矩阵相乘来进行并行计算加速。而在这三个步骤上都可以尝试进行改进,例如可变形卷积<sup>[69]</sup>就是将采样这一步的矩阵式采样修改为数据驱动的采样方式,使得卷积核不再为固定的  $n \times n$  矩阵,而是会随着图中物体的改变而改变,如图2-3(b)所示。类似的操作还可以施加在后面两步中,这是一个非常有趣的探索方向。

### 2.1.1.2 内积层

内积层,又称为全连接层,是神经网络的重要组成部分。若上一层的输出为特征向量  $x$ , 则其输出为:

$$y = W^T x + b, \quad (2-1)$$

其中  $W$  和  $b$  为内积层的参数。内积层的作用主要是为特征变换提供可学习参数,通过线性变换将输入特征映射到一个新的空间中,再通过激活函数的扭曲作用,多层之后即可完成非常复杂的特征变换。

### 2.1.1.3 激活函数层

前边提到的卷积层和内积层均为线性变换层,如果只是卷积层与内积层级联的话,因为线性变换乘线性变换仍旧是线性变换,所以整个网络所提供的变换还是线性变换,线性变换的表达能力是极其有限的,因此人们通常会在线性变换层

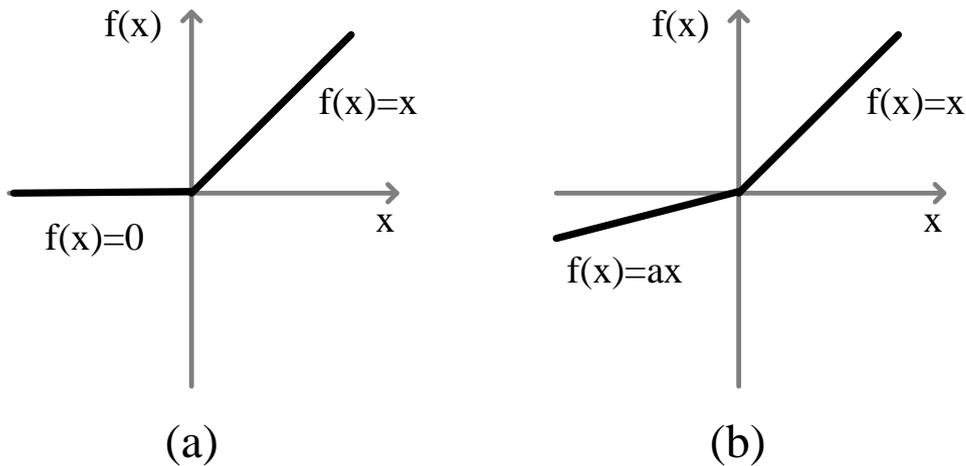


图 2-4 (a) ReLU 激活函数；(b) 参数化的 ReLU 激活函数。

中间穿插非线性的激活函数层。

对于激活函数的选择实际上非常自由，激活函数只需要满足两个条件：（1）激活函数必须为非线性函数；（2）激活函数必须可微以适用梯度传播。

在第二波深度学习浪潮之前，人们使用的激活函数大多为有界限的激活函数如 Sigmoid 激活函数  $\frac{1}{1+e^{-x}}$  和 Tanh 激活函数  $\tanh(x)$ ，这两个激活函数的特点是在  $x$  较大时都趋于平缓，且函数的上界下界都存在。这类激活函数是模仿生物神经元的兴奋模型而得来，神经元在未受到刺激时输出 0，在受到刺激时逐渐兴奋并向外输出信号，而因为生物体自身的耗能是有限的，所以这个信号也是有上限的。

根据仿生学得到的两种激活函数都存在梯度弥散现象，正如上一章的小节 1.4.1 和图 1-3 所解释的，Sigmoid 函数在接近 0 处的导数较大，而两端的导数极小接近于 0（如图 1-3 所示）。这个特点使得网络在训练初期快速变化时一旦输入的幅度过大，反向的梯度就会极小，此时再用梯度对权值进行更新带来的改变就会微乎其微，导致网络再也无法恢复。

而 ReLU 激活函数  $\max(x, 0)$  就不存在这个问题，ReLU 函数的梯度并不随着输入幅度的改变而改变（如图 1-3 所示），因此能够在训练的全程保持梯度幅度的稳定。ReLU 函数虽然解决了梯度弥散问题，但它仍存在神经元“凋亡”的问题，即如果输入全部为负数，则 ReLU 激活函数的输出全部为 0，此时反向的梯度也会全部为 0，从而使网络无法更新，这种情况比较罕见但仍旧存在。为了解决这一问题何恺明博士提出了参数化的 ReLU 函数<sup>[14]</sup>，在输入小于 0 时激活函数输出  $ax$  而不是 0，这样即使输入全部为负数，神经网络仍然能获得梯度进行优化（如图 2-4(b) 所示），其中  $a$  可由反向传播学习得到。

ReLU 激活函数及其变种的主要作用是“挤压”数据分布，如图 2-5 所示，原

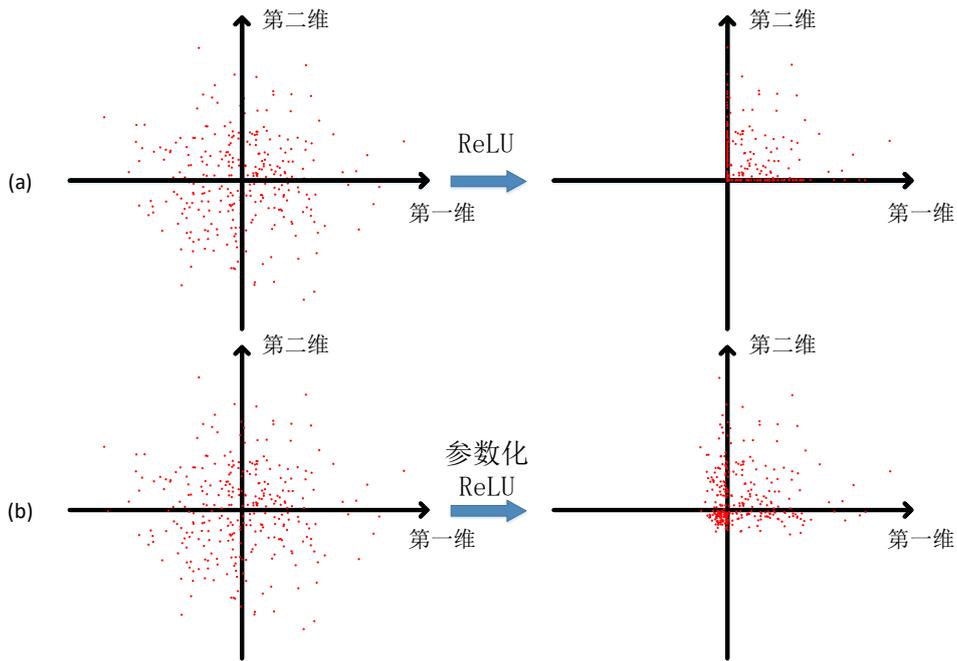


图 2-5 (a) ReLU 激活函数对特征的影响；(b) 参数化的 ReLU 激活函数对特征的影响。

始版 ReLU 激活函数会将所有小于零的数据都推向零点，导致除第一象限外的数据都被极大地收缩。而参数化的 ReLU 激活函数的挤压效果稍弱，仍能保持一些原始的局部分布。

此外，ReLU 激活函数的变种还包括将正数负数串联的 CReLU<sup>[71]</sup>，将负半轴改为指数函数的 ELU<sup>[72]</sup>，负数斜率随机变化的 RReLU<sup>[73]</sup> 等，感兴趣的读者可自行检索阅读。

#### 2.1.1.4 归一化层

在传统的机器学习或者浅层的神经网络中，对数据进行归一化往往能起到加速收敛的作用。这里主要的原因是幅度较高的数据维度与幅度较低的维度对应的权重应具备不同的学习率，而使用的梯度下降法进行优化时使用的是同一个学习率，这就导致整个网络都必须用较小的那个学习率来进行学习，进而导致网络收敛较慢。而且较小的学习率不容易跨过优化路途上的局部最优点，这往往会导致更差的泛化性能<sup>[74]</sup>。

而在神经网络中，由于层数非常深，学习率的不匹配不仅表现在同层的不同维度上，还表现在不同层的权重所适用的学习率不匹配。如果只对数据进行归一化，则对首层数据进行归一化的作用会随着层数的加深而越来越低，一个解决方案就是在网络的中间插入归一化层来归一化中间层特征。

这其中最著名的就是批归一化<sup>[15]</sup>，批归一化在每个卷积层后将一个批次内的

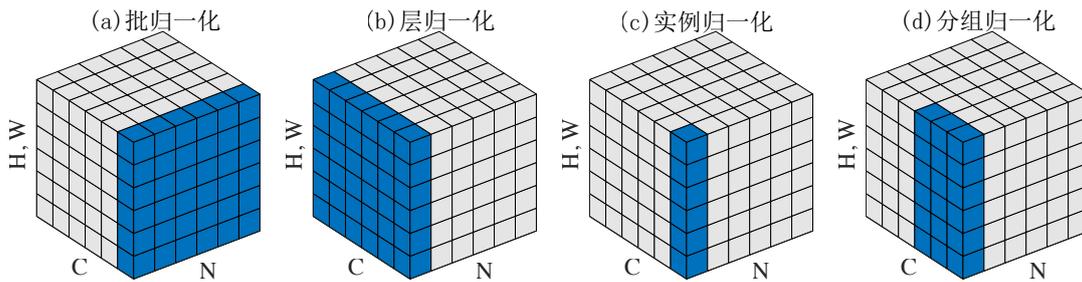


图 2-6 (a) 批归一化<sup>[15]</sup>、(b) 层归一化<sup>[75]</sup>、(c) 实例归一化<sup>[76]</sup>、(d) 分组归一化<sup>[77]</sup> 的归一化范围，其中 W、H 分别是特征图的高和宽，C 代表通道数，N 代表一个批次内的样本数量。图片来自文献 [77]。

同一个通道的数据归一化成 0 均值、1 标准差的分布。然而仅仅这样做会导致模型的退化，例如如果一个批次内的均值为 0，那前边卷积层的偏置项就没有作用了，而如果一个批次内的标准差都是 1，那得到的分布永远都是一个正超球形的分布，是无法表达非常复杂的特征的。因此在归一化之后还要针对每个通道学习一个尺度系数  $\gamma$  和一个偏置系数  $\beta$ ，最终的表达式为：

$$\hat{x} = \gamma \frac{x - \mu_B}{\sigma_B} + \beta. \quad (2-2)$$

在测试时，因为数据都是单个输入进网络的，所以不能再使用一个批次内的统计量，此时要将均值和标准差替换为整个训练集上的统计量：

$$\begin{aligned} E[x] &= E_B[\mu_B] \\ \text{Var}[x] &= \frac{m}{m-1} E_B[\sigma_B^2], \end{aligned} \quad (2-3)$$

需要注意的是其中对于方差的估计是其无偏估计。

归一化的最佳方案是对整个训练数据集上的所有数据做归一化。然而由于显存限制，只能一个批次一个批次地将数据输入网络，所以归一化操作也只能在一个批次内进行，这就导致了如果一个批次内的样本过少时会存在均值和方差估计不准的情况。而在检测和分割算法中，每个批次中通常只有 1 或者 2 个样本，此时批归一化将不再适用。

为了克服这个缺点，层归一化<sup>[75]</sup>、实例归一化<sup>[76]</sup>、分组归一化<sup>[77]</sup> 陆续被提出。如图 2-6 所示，它们都脱离了批次内样本数量这一维度来进行归一化，从而使得归一化摆脱了均值和方差估计不准的情况。然而，因为参与归一化的数值数量大大降低了，所以这些方法的效果都不如原始的批归一化。批归一化的作者在后继工作 [78] 中提出了一种渐进式的消除批次中的统计量与全局统计量不吻合现象

的方法，但这个方法只能将最低批数量从 32 降到 8，仍然无法满足检测和分割算法中使用的 1 或者 2 个样本的要求。如何能够在充分归一化的基础上真正地解决均值和方差估计不准的问题，仍待研究者的后续工作。

### 2.1.1.5 残差网络

在批归一化提出之后，深度神经网络的梯度弥散问题可以宣告解决了。但是在何恺明博士提出的参数化 ReLU<sup>[14]</sup>一文中，他和他的研究小组发现卷积神经网络的有效层数仅为 20 层左右，超过 20 层的卷积神经网络的性能将会逐渐下滑。这个现象是有悖于常识的，研究者们曾经认为神经网络越深，其表达能力越强，性能应该越好<sup>[79]</sup>。因为使用了批归一化，所以这个问题的原因不是梯度弥散；从实验中还发现训练和测试性能是同时下降的，因此也不是过深的网络对训练数据集的过拟合效应<sup>①</sup>，这其中真正的原因至今仍然没有定论。文献 [80] 中作者们提出了一个“破碎梯度”的假说来解释性能下滑的原因，但因为他们的实验过于粗糙，并不能说明其假说的正确性，但这篇文章仍然是对神经网络理论解释的一个很不错的尝试。

虽然这个问题还没有得到有说服力的解释，但也并不妨碍人们寻找解决方案。何恺明博士和他的团队在经过一系列的尝试后，提出了一种残差模块的神经网络结构。残差模块的设计基于一个基本的方法论：如果第  $N+1$  层的神经网络是一个等值函数，那其性能应该至少不低于前  $N$  层构成的神经网络。残差神经网络的每一个模块都由一个等值函数与一个普通的卷积、批处理、激活函数构成的非线性表达函数相加得到（如图 1-4 所示），这样由等值函数这一路可以保证深层的神经网络的性能至少不会退化，而非线性表达函数这一路则可以提升性能。在实验中，作者们甚至构建了一个 1000 层的神经网络，取得了比 100 多层的神经网络更好的性能<sup>[17]</sup>。

假设传统的神经网络卷积模块是一个输入为  $x$  输出为  $F(x)$  的非线性函数，那么残差模块的输出就是  $F(x) + x$ ，可以看到它仍然具备非线性变换的能力。而在反向传播时，其梯度为  $F'(x) + 1$ ，可以看到如果  $F'(x)$  损失了梯度的有效性，则 1 这一路还仍旧能将有效的梯度向前传播。

另一种对残差网络的解释来自 [81]，在这篇文章中作者们将残差网络解释为多个浅层网络的集成，最终网络的输出为所有的浅层网络输出之和，按照这个解释，似乎残差神经网络并没有真正解决梯度失效的问题。作者在文中也进行了实验，将较深层的梯度传播砍掉后，网络性能还有了稍许的提升，所以该理论很可能是正确的。根据这个理论，残差神经网络并没有真正地解决梯度失效问题，只

<sup>①</sup> 通常过拟合的表现表现为训练性能极好但测试性能较差。

是利用了跳层连接来训练大量的浅层网络再集成在一起，其深层分支的贡献仍然是比较差甚至是有害的。梯度失效问题仍待研究者的进一步研究。

### 2.1.2 损失函数

损失函数是指导整个神经网络训练的一个部件，如果说各模型结构是神经网络的躯干，那损失函数就是神经网络的大脑。一个完整的神经网络会输出一系列的数值，将这些数值与理想数值进行比对后向这些数值输出梯度，之后通过反向传播算法将梯度传播到所有层进行更新。

一般来说，在设计损失函数时要与测试指标进行匹配，即测试时使用什么样的指标就用什么样的损失函数。例如我们在文章 [82] 中就发现，在评价指标为均方误差时，使用均方误差作为损失函数效果最好；在评价指标为平均绝对误差时，使用绝对值误差效果更好；在指标为各类别同等权重时，对训练样本进行均匀采样可以大幅提升效果。

本论文所做工作均为针对损失函数的修改，其中大部分是基于 Softmax 交叉熵损失函数做的改进，Softmax 交叉熵损失函数是目前最常用的分类损失函数。若要将样本分为  $C$  个类别，在使用 Softmax 交叉熵损失时，需要将神经网络的最后一层输出设置为  $C$ ，得到  $C$  个分数  $z_i, i \in [1, C]$  后输入 Softmax 交叉熵损失函数：

$$\mathcal{L}_S = -\log \frac{e^{z_y}}{\sum_{i=1}^C e^{z_i}}, \quad (2-4)$$

其中  $y$  为当前样本所对应的标签。

Softmax 交叉熵损失函数实际上分为两步：求 Softmax 和求交叉熵损失，其中第一步操作可以得到当前样本属于某类别的概率  $P_i$ ，然后将这些概率与理想值 One-hot 向量求交叉熵，因为理想值是仅在第  $y$  个位置为 1，其他部分为 0，所以最终只保留了第  $y$  个位置的交叉熵： $-\log P_y$ 。

值得一提的是，在使用反向传播算法时，应当将 Softmax 交叉熵损失函数作为一个整体进行求导，否则在求交叉熵导数时有可能会遇到求  $\log(0)$  的问题，导致梯度幅度为无穷大。对于第  $i$  个样本 Softmax 交叉熵损失的梯度为：

$$\frac{\partial \mathcal{L}_S}{\partial z_i} = \begin{cases} P_i - 1, & i = y \\ P_i, & i \neq y \end{cases}. \quad (2-5)$$

这个即为 Softmax 交叉熵损失函数的概率解释，此外正文的第 5.3.2 节还给出了一个对于 Softmax 交叉熵损失函数的柔化目标函数的解释，读者可以结合这两种解释来理解该损失函数。

针对 Softmax 交叉熵损失函数的概率解释，有一些工作做了一些改进，例如标签柔化 (Label Smoothing) [83] 将理想分布的 One-hot 向量柔化为

$$\mathbf{q} = \left[ \frac{1}{C-1}\varepsilon, \dots, 1 - \varepsilon, \dots, \frac{1}{C-1}\varepsilon \right], \quad (2-6)$$

的形式，这样可以减轻 Softmax 概率无限尝试逼近 1 所带来的过拟合效应。此外，模型蒸馏法 [84] 将一个大模型输出的概率值进行柔化来作为理想分布优化小模型，这样可以使得小模型能够得到更多的信息来进行训练，而不只是简单地去学习 0 和 1。

当理想分布不再是 One-hot 向量而是一组其他概率值  $q = [q_1, q_2, \dots, q_C]$  时，交叉熵损失函数为：

$$\mathcal{L}_{CE} = - \sum_{i=1}^C q_i \log p_i, \quad (2-7)$$

梯度变为：

$$\frac{\partial \mathcal{L}_{CE}}{\partial z_i} = p_i - q_i. \quad (2-8)$$

可以看到梯度形式非常简单，实际上公式2-5也可以化为这种形式，这样的形式更容易理解与实现。

### 2.1.3 优化

神经网络大多使用反向传播算法 [6] 来逐层计算梯度，前一层的梯度为后一层梯度乘以该层的梯度，这种实现方式将层与层之间解耦，每次迭代只需逐层前向传播一次再逐层反向传播一次即可，比较容易进行模块化的实现。

得到梯度后即可进行梯度更新，最常用的梯度更新策略为随机梯度下降，设第  $t$  次迭代的网络权重参数为  $w_t$ ，梯度为  $\nabla w_t$ ，则下一时刻的网络权重为：

$$w_{t+1} = w_t - \eta \nabla_w L(w_t), \quad (2-9)$$

其中  $\eta$  为学习率，一般是一个比较小的数字， $\nabla_w L(w_t)$  为损失函数  $L$  对参数  $w$  的梯度。对于学习率的设定，一个常见的做法是先设置为一个能使网络收敛的最大的数，然后观察最终的损失值，当损失值长时间都不再下降时，将学习率除以 10 继续训练，重复这一过程 2-3 次即可结束训练。这种方式比较粗放，在大部分情况下能保证模型较好地收敛。还有一些其他的学习率下降方式例如指数下降、余弦式下降 [85] 等。有时网络初始化时权重初始化会不理想，无法接受较大的学习率，这时一般要使用较小的学习率先训练几百次迭代，然后再换用更大的学习率，这

种方式叫热启动<sup>[16]</sup>。还有一种在学习率较小后重新使用大的学习率来重新启动训练的方法<sup>[85]</sup>，这种学习率策略可以在网络陷入局部最优时通过加大步长来跳出局部最优。

由于在神经网络训练中一次只输入一个批次的样本，因此每次得到的梯度实际上只是一小部分样本上计算得到的，而之前样本的梯度方向实际上仍然是有指示作用的，一般在使用随机梯度下降法时需要增加一个动量项  $\gamma$  来对梯度进行滑动平均：

$$\begin{aligned} \mathbf{g}_t &= \gamma \mathbf{g}_{t-1} + \eta \nabla_w L(\mathbf{w}_t), \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \mathbf{g}_t, \end{aligned} \quad (2-10)$$

这里动量系数  $\gamma$  一般是一个比较接近 1 的数字，如 0.9。注意到如果按这个公式优化，在调整学习率  $\eta$  的时候，动量项仍然会保持原来的梯度幅度，不会及时地更新。在 PyTorch 框架的实现中，作者们将公式修改为：

$$\begin{aligned} \mathbf{g}_t &= \gamma \mathbf{g}_{t-1} + \nabla_w L(\mathbf{w}_t), \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \eta \mathbf{g}_t, \end{aligned} \quad (2-11)$$

这样在修改学习率  $\eta$  的时候，因为动量项也乘了  $\eta$ ，所以能够及时地进行更新，如果使用的是本小节之前提到的分阶段式下降法，那 PyTorch 的这种实现方式是更加恰当的。

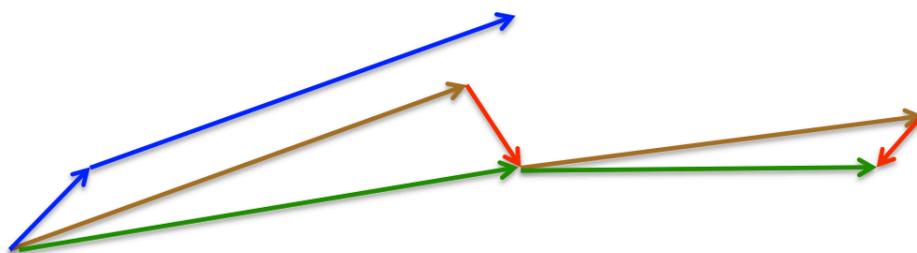
带有动量项的随机梯度下降算法是对当前位置的梯度进行修改，这会导致本次更新所计算的梯度方向不准确。所以 Nesterov 提出要将权重先行更新到历史梯度所指示的位置，然后进行一步步长较小的随机梯度下降，该方法被称为 Nesterov 加速梯度下降法<sup>[86]</sup>，如图2-7所示，左下角为起始点，蓝色箭头为使用动量的随机梯度下降法的更新方向，Nesterov 法则先将当前位置移动到棕色箭头指向的位置，然后进行一步不带动量的梯度下降（红色箭头）。

Nesterov 梯度下降法的迭代公式为：

$$\begin{aligned} \mathbf{g}_t &= \gamma \mathbf{g}_{t-1} + \eta \nabla_w L(\mathbf{w}_t - \gamma \mathbf{g}_{t-1}), \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \mathbf{g}_t. \end{aligned} \quad (2-12)$$

在实现时，在  $\mathbf{w}_t$  位置是无法得到  $\nabla_w L(\mathbf{w}_t - \gamma \mathbf{g}_{t-1})$  的，因此人们往往会直接到  $\mathbf{w}_t - \gamma \mathbf{g}_{t-1}$  位置来执行  $\eta \cdot \nabla_w L(\mathbf{w}_t - \gamma \mathbf{g}_{t-1})$ ，再更新历史梯度  $\gamma \cdot \mathbf{g}_{t-1}$ ，如图2-7所示先执行红色箭头再执行棕色箭头，这样就避开了无法预知更新了历史梯度后的损失函数梯度的问题。

不论动量法还是 Nesterov 法，都需要手动调节学习率的下降曲线，如果希望


 图 2-7 Nesterov 加速梯度下降法的梯度下降方向示意图<sup>①</sup>。

精细调参的话，这通常是一个非常繁琐的工作，因而有一些研究者在研究如何自动设置学习率。其核心思路是利用梯度的二阶动量来归一化梯度值，第一个如此尝试的是 AdaGrad 算法<sup>[87]</sup>，AdaGrad 算法会在训练过程中统计历史梯度的二阶动量，并将当前梯度除以该二阶动量：

$$V_t = \sum_{\tau=1}^t \nabla_w^2 L(w_\tau),$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{V_t} + \varepsilon} \nabla_w L(w_t). \quad (2-13)$$

AdaGrad 法的问题在于累积的二阶梯度会越来越大，使得学习率越来越低。但这个学习率降低的曲线并不一定是人们想要的曲线，它经常会过早地将学习率降低到一个很小的值，导致网络得不到充分的收敛。为了解决这一问题，RMSprop<sup>①</sup>在二阶动量上使用滑动均值法来取代累加以防止学习率过早地降低到很小的值：

$$V_t = \gamma V_{t-1} + (1 - \gamma) \nabla_w^2 L(w_t),$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{V_t} + \varepsilon} \nabla_w L(w_t). \quad (2-14)$$

其中  $\gamma$  的建议值为 0.9、 $\eta$  的建议值为 0.001。而后，类似于带动量的随机梯度下降法，Adam 算法<sup>[88]</sup>又在 RMSprop 的基础上添加了一阶动量：

$$g_t = \beta_1 g_{t-1} + (1 - \beta_1) \nabla_w L(w_t),$$

$$V_t = \beta_2 V_{t-1} + (1 - \beta_2) \nabla_w^2 L(w_t),$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{V_t} + \varepsilon} g_t. \quad (2-15)$$

然而，使用滑动均值的二阶动量虽然解决了学习率下降过快的问题，但又将问题带向了另一个极端：当优化进行到震荡阶段时，使用二阶动量的滑动均值会趋近于不变，使得学习率不能继续降低来进一步收敛。通常在手动调参时碰到损失不降时会手动降低学习率，但使用滑动均值的方法无法自动做到这一点。为了

<sup>①</sup> 未发表，来自 Hinton 教授的讲义 [http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_lec6.pdf](http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf)。

缓解这一问题，AMSgrad<sup>[89]</sup> 在二阶动量项上增加了一步约束： $\hat{V}_t = \max(\hat{V}_{t-1}, V_t)$ ，只有在当更新后的二阶动量大于先前的值时才进行更新，但这一方案仍然只能缓解，并不能彻底解决这类自适应梯度方法后期学习率收缩困难的问题。

使用二阶动量的方法的另一个缺陷是丢失掉了梯度之间的相关性<sup>[90]</sup>，如果一个维度的梯度始终较大，另一个维度的梯度始终较小，则其二阶动量也会相应的较大或较小。经过除法之后这些维度上的幅度差异会被抵消，导致更新的梯度的幅度相差不大。这个缺陷和上文提到的学习率下降过快或过慢的问题导致了自适应梯度法得到的模型性能往往不如随机梯度下降法，如何对自适应梯度法进行改进使得其至少能达到随机梯度下降法的性能仍需研究者的后续研究。

## 2.2 人脸认证

本节将介绍有关人脸认证的相关概念、评价标准、进行人脸认证所必须的前置步骤和一些经典的人脸认证算法，除此之外还会介绍一些最新的算法来与后章节提出算法进行比较。

### 2.2.1 相关概念

人脸识别是一个比较大类的问题，其中包括很多子问题如人脸认证、人脸辨识、人脸检索等。根据训练数据与测试数据之间有无身份重叠又分为两类：闭集人脸识别和开集人脸识别。

如图2-8所示，闭集人脸识别要求测试数据集中人脸对应的身份已经全部包含在训练数据集中，这样在进行人脸辨识、人脸认证时，只需输出人脸照片对应的身份即可进行相应的任务。而如果测试数据集中人脸的身份中含有训练集里没有的身份时，就必须使用开集人脸识别的方法，即通过提取特征、与库中人脸比对相似度的方式。毫无疑问在真正实用的场景中，开集人脸识别的应用范围更广，因此本文主要讨论的就是开集人脸识别问题。此外，在学术研究中，为了保证算法比较的公平性，通常会要求训练集与测试集的人脸身份完全不重合，这样的测试标准被称为严格的开集测试。

闭集人脸识别与开集人脸识别在方法上的主要区别在于闭集人脸识别通常使用分类问题的处理方式即可完成，而开集人脸识别需要使用度量学习的方法来解决。对于一个分类问题，只需要找到一系列的分界面将各个类别样本分开即可，而度量学习要求类间间距要大于类内间距，也就是说不仅要能分开，而且要分得足够远。

开集人脸识别包含以下任务：人脸认证（Face Verification）指的是给定两张人

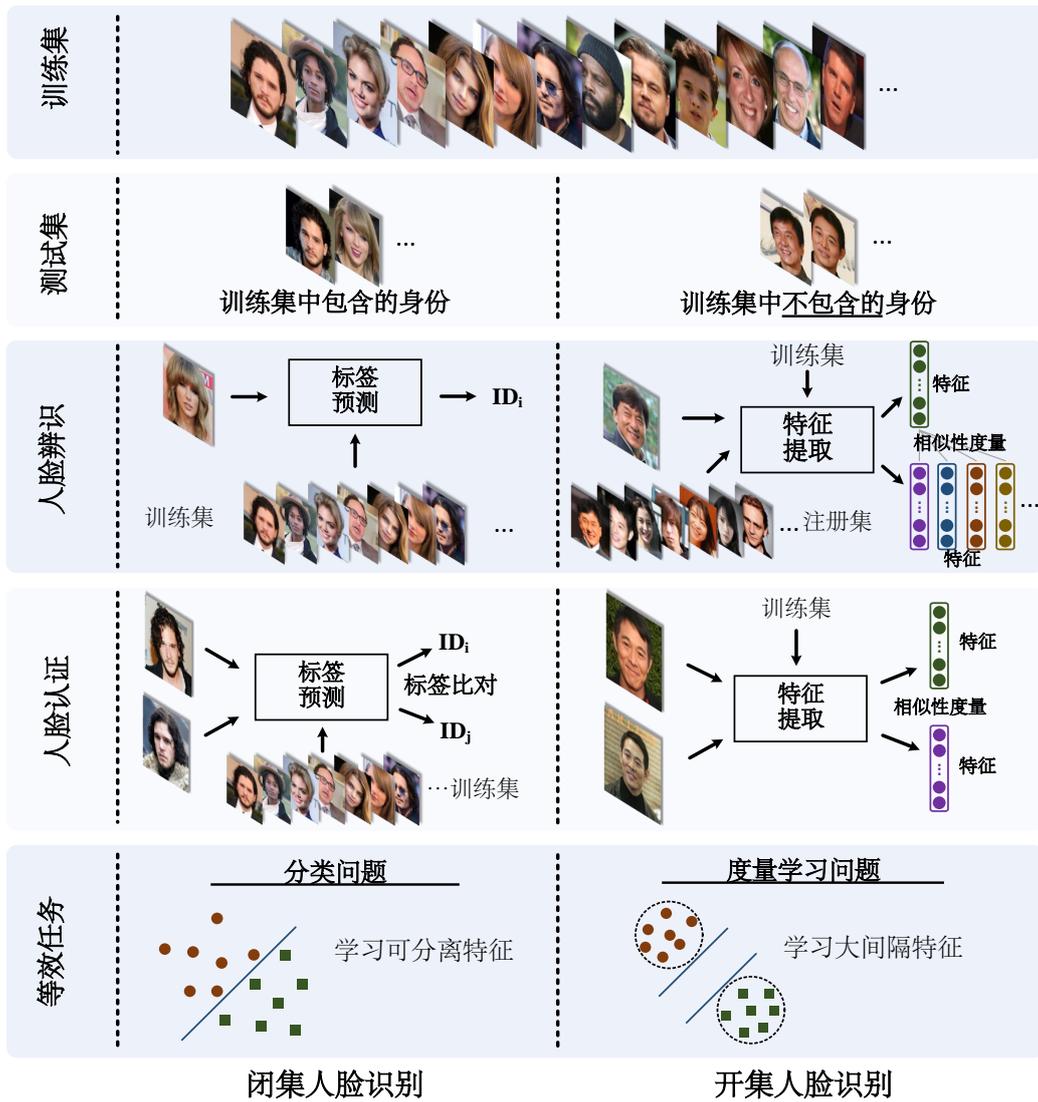


图 2-8 闭集、开集条件下的各任务流程的区别<sup>[64]</sup>。

脸图像，由计算机自动判定这两张人脸图像是否属于同一个人；人脸辨识（Face Identification）指的是给定一张图像，将它与数据库中的人脸进行比对，若数据库中该身份已经被注册过则返回这个身份，若没有注册过则返回“不在库中”；人脸检索（Face Retrieval）指的是给定一张人脸图像，在与数据库中的人脸图像进行比对后，返回相似度较高的若干张图像，并且将这些图像按相似度从大到小进行排列。人脸辨识与人脸检索的区别在于人脸检索并不需要判断出输入人脸对应的身份，而只需要列出相似的几个即可，人脸辨识则需要给出身份，相当于同时进行了人脸检索和人脸认证两个任务。

这些任务的共同之处在于它们都需要先进行特征提取，而后计算特征之间的相似度作为两张人脸图像的相似度。人脸认证任务只需要比对两张人脸图像，通

常是一个人拍摄的照片和其身份证件上的照片进行比对，这种比对方式又称为 1:1。而人脸辨识任务是将一个人拍摄的照片提取到的特征与数据库中所有的照片对应的特征进行比对，所以又称为 1: N。这些任务在模型层面上是一样的，都需要学习一个相似性度量的方式，区别仅在于如何利用相似度来完成各自的任务，所使用同一个模型来进行多种任务也是可行的。

### 2.2.2 人脸数据集

人脸认证数据集主要分为两类：训练数据集和测试数据集，其中训练数据集要求每个人都有较多的图片，而测试数据集则没有硬性要求。人脸认证这一领域比较热门，因此在学术界有很多公开数据集可以使用，表2-1中列出了常见的训练和测试数据集及其规模。这些数据集均为非配合条件下采集的人脸照片，具备多种不同的姿态、表情、年龄、人种、光照等条件下的人脸照片，其中 YTF<sup>[91]</sup>、IJB 系列<sup>[92-94]</sup>数据集包含有大量人脸视频，更加适合验证视频下的人脸识别性能；CFP-FP<sup>[95]</sup>数据集包含了很多较大旋转角度的照片，更加适合验证算法的姿态稳定性；AgeDB<sup>[96]</sup>数据集包含了非常大的年龄跨度下的人脸照片，更加适合验证算法的年龄稳定性。

表 2-1 学术界人脸认证数据集总览

数据集	类型	身份数	图像数
CASIA-WebFace <sup>[97]</sup>	训练集	10,575	494,414 张图像
VGGFace <sup>[61]</sup>	训练集	2,622	约 260 万张图像
MS-Celeb-1M <sup>[98]</sup>	训练集	100,000	约 1000 万张图像
UMDFaces <sup>[99]</sup>	训练集	8,501	367,920 张图像
VGGFace2 <sup>[100]</sup>	训练集 & 测试集	9,131	约 330 万张图像
Asian-Celeb <sup>[101]</sup>	训练集	93,979	约 283 万张图像
LFW <sup>[55]</sup>	测试集	5,749	13,233 张图像
YTF <sup>[91]</sup>	测试集	1,595	3,425 个视频
IJB-A <sup>[92]</sup>	测试集	500	5,712 张图像, 2,085 个视频
IJB-B <sup>[93]</sup>	测试集	1,845	11,754 张图像, 7,011 个视频
IJB-C <sup>[94]</sup>	测试集	3,531	31,334 张图像, 11,779 个视频
MegaFace <sup>[102]</sup>	测试集	690,572	约 100 万张图像
CFP-FP <sup>[95]</sup>	测试集	500	7,000 张图像
AgeDB <sup>[96]</sup>	测试集	568	16,488 张图像

由于本学位论文的重点在于验证算法的有效性，因此本学位论文选择 CASIA-

Webface 作为训练数据集，这个数据集的规模足够证明算法的有效性而需要的训练时间又不是很久，比较适合进行算法的比对。测试集方面本文选择了 LFW<sup>[55]</sup> 和 YTF<sup>[91]</sup> 这两个较小的数据集来快速测量各种参数下的模型性能，最后在 MegaFace<sup>[102]</sup> 这个较大的数据集上测试了本文提出的算法与其他算法的性能差异。

### 2.2.3 评价标准

为了测试人脸识别模型的性能，学术界提出了一系列的指标。其中对于人脸认证的主要指标为准确率（Accuracy, 也称识别率），还有一个是更加严格的在一定的错误接受率 FAR（False Accept Rate）<sup>①</sup>下的真正率 TPR（True Positive Rate）或错误拒绝率 FRR（False Reject Rate），其中 TPR 与 FRR 之间的关系为： $TPR = 1 - FRR$ 。

在人脸认证中因为会存在人脸照片质量过低无法进行匹配验证的情况，所以还有一个指标叫获取失败率 FTA（Failure-to-Acquire Rate），在获取有效信息失败的情况下，这些照片将会被直接拒绝，而剩下的样本才会进行匹配验证。图像质量评价的模块是一个完整的人脸认证系统应当包含的，对于质量较低的图直接拒绝进行匹配，这样造成的错误拒绝一般并不会被认为是真正的错误，而是会直接要求相关人员重新录入照片或者直接忽略这部分样本。一些专业进行人脸认证测试的机构<sup>②</sup>倾向于使用一定的错误匹配率 FMR（False Match Rate）下的错误不匹配率 FNMR（False Non-Match Rate）进行评价，这两个指标即为忽略掉错误样本后的错误接受率和错误拒绝率，它们之间有如下关系：

$$\begin{aligned} FAR &= FMR * (1 - FTA), \\ FRR &= FTA + FNMR * (1 - FTA). \end{aligned} \tag{2-16}$$

对于人脸辨识任务，评价指标通常为首选正确率（Rank 1）或者检测与辨识率 DIR（Detection and Identification Rate），其中首选正确率为人脸图像与其在库中检索到的最相似的人脸是同一身份的个数与全部待测图像总数之比。对于检测与辨识率，其检测部分意为给定一张图像，判断其身份是否在数据库中，而辨识部分则要给出这张人脸图像所对应的身份，所以其本质上是综合了人脸认证与人脸辨识任务。对检测与辨识率的计算首先要按照人脸认证的指标划定一个较低的错误接受率 FAR（通常为 1%），在这个划定的比率下计算出是/否是人脸的相似度阈值，然后在库中进行人脸检索，最后仍然以首选正确率作为最终的分数。检测与辨识率与首选正确率的区别就在于检测与辨识率多了一步计算是/否是相同人脸的阈值

① 相同意义的英文还有虚警率 False Alarm Rate 和假正率 False Positive Rate，也有的中文翻译为误识率。

② 如美国国家标准委员会举办的人脸识别供应商测试 FRVT。

的步骤，在实际应用中更加实用。

上述指标的结果均为单个的数字，有时需要对算法在多个限定的条件下的性能，此时就需要以曲线的形式来表示。人脸认证任务中的 TPR 与 FAR 之间可以绘制一条接收者操作特征曲线 ROC (Receiver Operating Characteristic)，在实际应用中，虚警（即将不是同一个人判定为同一个人）是非常严重的错误，意味着系统被攻破，因此需要将虚警率 FAR 控制在一个比较低的值上，而此时的召回率 TPR 则关系到用户体验的好坏，较低的 TPR 会导致用户需要多次尝试才能正确认证其身份。在实际应用中需要绘制出 ROC 曲线，根据实际需要在 ROC 曲线上选择一个比较理想的点，例如在金融领域通常需要将 FAR 设置得极低（通常为百万分之一到亿分之一），而在手机解锁应用上可以选择较高的 FAR 值来提升用户体验。

人脸辨识任务的 k 选正确率与 k 之间的关系也可以表现在一条 CMC (Cumulative Match Characteristic) 曲线上，这条曲线经常用在检索任务中，其意义为在检索出 k 个数据库中的样本时，里面至少有一个与待检索样本同类的比率。在人脸辨识任务中使用的通常都是首选正确率，即待检索人脸与最相似的一个人脸样本来自于同一个人的比率。首选正确率的应用最为广泛，但在一些比较大规模的应用中为了提高召回率，也可以使用较大的 k 值，例如在追踪逃犯时为了避免漏掉疑犯，需要多列出一些可能的人脸以供后续筛查。

在数据集 Labeled Face in the Wild (LFW) 数据集<sup>[103]</sup>上，比较常见的做法是使用其不限制额外数据协议（又称为 6,000 对协议）中的准确率这一指标来进行评测。LFW 官方提供了 6,000 个样本对，其中 3,000 个来自于同一个身份，3,000 个来自不同身份，这些样本被均分为 10 组，使用 10 折评测法进行学习及评价，通常会在其中的 9 折上（即 5,400 对样本）上进行特征领域适应来将训练的特征匹配到 LFW 数据集的样本特性上，然后计算得到 5,400 个相似度，取出最适合这 5,400 对样本的判别阈值，之后在剩下的 600 对样本上使用相同的领域适应方式与判别阈值计算准确率。这个方式仍然在 LFW 数据集上进行了领域适应以及判别阈值的计算，因此并不能算是完全的开集测试，真正的开集测试应当不包括领域适应，且阈值计算应在其他不同身份的数据上计算得到，然而为了比对的公平性，大多学者仍然使用该评测标准来进行评测。

在 LFW 数据集的 BLUFR 协议<sup>[104]</sup>中，进行了人脸认证和人脸辨识两项任务，其中人脸认证部分与 LFW 6,000 对协议类似，但是 BLUFR 协议使用了几乎全部的样本进行相似度计算，而且评价标准也变成了较低 FAR 下的 TPR 进行评价。同 LFW 6,000 对协议一样，BLUFR 的人脸认证任务也采用了 10 折测试法，通常也要进行领域适应，因此并不算严格的开集测试。在人脸辨识任务中，使用的是检测与

辨识度来评价，作者将 LFW 数据集分为注册集 (Gallery)  $G$  和两个查询集 (Probe Set)  $P_G$  和  $P_N$ ，其中  $P_G$  集为与  $G$  集相同的身份却是不同的照片组成， $P_N$  集中的照片与  $G$  集中无相同身份。BLUFR 协议使用  $P_N$  集进行领域适配以及判别阈值的计算，而使用  $P_G$  集与  $G$  集进行人脸检索，由此而保证了该测试是一个开集测试。

MegaFace 数据集<sup>[102]</sup> 也分为人脸辨识与人脸认证两项任务，MegaFace 中包含两个数据集，其中一个为查询集 (Probe Set)，另一个为干扰集 (Distractor Set)，查询集中样本数量较少而干扰集中包含上百万张人脸图像，两个数据集之间没有身份重叠<sup>①</sup>。对于人脸辨识任务 MegaFace 使用首选正确率作为评价指标，计算方式为对于查询集中的每个身份，将其中的一张图像加入干扰集中，之后使用所有的其他图像在干扰集中进行检索，若检索到的最相似的图像恰好为之前放入干扰集中的图像算作正确检索，最终的到整个查询集上的首选正确率。MegaFace 作者们还设计了一个修改后的类 CMC 曲线来辅助评价，该曲线并不是以排序值  $k$ ，而是以干扰集中的样本数量作为自变量，这样得到的曲线即为模型在不同规模的注册库下的性能。对于人脸认证任务 MegaFace 使用百万分之一的 FAR 下的 TPR 作为指标，其中真匹配 (Genuine Match) 在查询集中计算得到，而假匹配 (Imposter Match) 在查询集与干扰集中的所有样本上计算得到，共有大概 40 亿个负匹配。MegaFace 官方还使用 ROC 曲线作为人脸认证的辅助指标来观察在其他 FAR 值下人脸识别模型的性能。MegaFace 的查询集还可以替换为 FG-Net，该数据集中包含了 82 个人的多个年龄的图像，因此更加适合对跨年龄的人脸识别进行测试。

#### 2.2.4 人脸检测与对齐

一个完整的人脸认证系统包含图像采集、图像预处理、人脸检测、人脸对齐、特征提取、相似度计算等步骤，利用最后得到的相似度来处理不同的任务。其中与算法相关的部分有人脸检测、人脸对齐、特征提取与相似度计算等几步。本学位论文主要关注特征提取与相似度计算这两步，而人脸检测与人脸对齐则使用目前性能较好的多任务卷积神经网络<sup>[106]</sup> 来进行，为了本学位论文的完整性，这一小节将对该算法进行简要的介绍。

图像检测任务从本质上来说是对图像上所有可能的区域，包含所有位置、大小有时还有旋转的矩形框 (或任意形状的框) 进行  $C+1$  类的分类，其中  $C$  为类别数，1 为背景类，在人脸检测任务中  $C = 1$ ，即为二分类问题。因为包含了各种尺度和位置的目标框，即使是一张图像也能生成海量的分类样本，所以图像检测的核心问题即为如何进行如此大量样本之间的分类，如何减少计算量和参数量、如

<sup>①</sup> 实际上仍然有身份重叠，详情参见 [105]，干扰集中存在的相同身份照片会显著地降低测试结果，尤其是对于人脸辨识任务使用的首选识别率。

何处理背景类占比过高的类不均衡问题是图像检测中最为重要的两大问题。

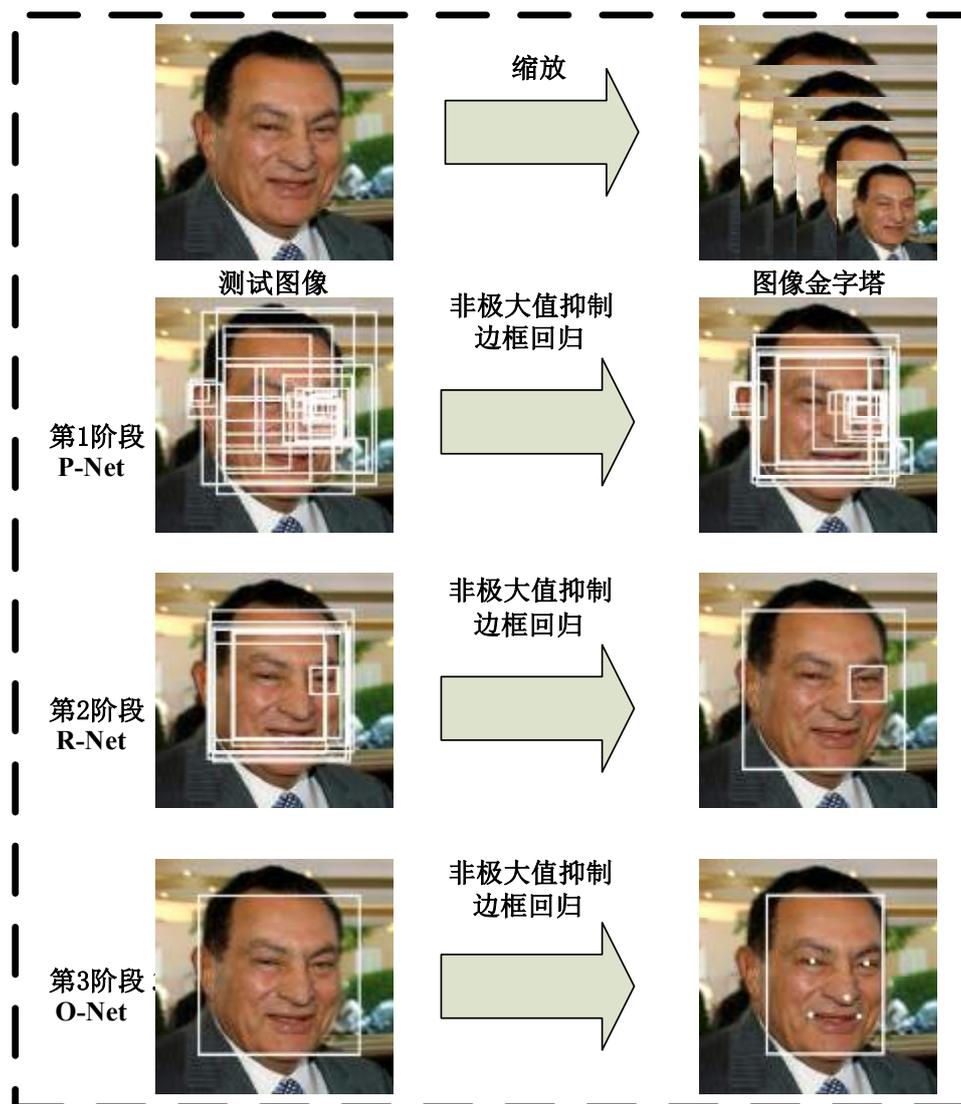


图 2-9 MTCNN 的整体流程<sup>[106]</sup>。

多任务卷积神经网络 MTCNN<sup>[106]</sup> 主要利用了图像在缩放时语义不变的特性，即一张人脸图像，不管是缩小到  $10 \times 10$  像素还是放大到  $1000 \times 1000$  像素，人眼都能立刻认出这是一张人脸。所以所有尺度的照片就都可以用一个网络进行识别，从而减少了网络的参数量，而且因为样本数量增多，也可以提高模型的泛化能力。

MTCNN 分为三个阶段：在第一阶段首先会建立一个图像的空间金字塔，然后在金字塔的所有层级上使用  $12 \times 12$  的滑动窗卷积神经网络来提取候选区域。对于每个候选区域回归 4 个候选区域到目标区域的框参数，使用回归得到的四个参数对候选区域进行修正后即可得到更加准确的目标框，对所有修正后的目标框按照提取候选区域时得到的分数进行非极大值抑制来剪除掉大量的重复目标框。

在得到第一步的所有目标框后，第二阶段即为使用这些目标框进行第二次筛选，由于人脸目标框大多为竖直的长方形，所以作者将目标框的两边扩充为正方形，然后用这些正方形去原图上提取图像块，之后将这些图像块缩放至  $24 \times 24$ ，输入进第二个卷积神经网络中得到新的人脸/非人脸的概率、目标框修正系数和 5 个人脸关键点位置等信息。与第一阶段一样，使用目标框修正系数对目标框进行修正，对修正后的目标框使用非极大值抑制来剪除掉重复的目标框。

第三阶段与第二阶段大体类似，只是分辨率变成了  $48 \times 48$ ，其余步骤均与第二阶段相同，用来得到最终的筛选结果。MTCNN 的整体流程图如图 2-9 所示。

自此人脸检测阶段就已经结束了，作者为了提高关键点检测的精度，在 v2 版本中加入了第四阶段，第四阶段只进行关键点回归操作，具体方法为取出第三阶段输出的各关键点附近的一个邻域的图像输入网络中，输出更加精细的关键点位置。这一步不再是由整张人脸图像来得到各关键点位置，而是由局部的图像块来得到，因此可以避免全局特征带来的干扰。

多任务卷积神经网络 MTCNN 的优势在于利用了图像尺度不变性来减少了参数量，大图与小图都使用较小的网络，而不像 RCNN 系列<sup>[20-22]</sup> 大图小图都使用较大网络、SSD 系列<sup>[23]</sup> 大图用较大网络小图用较小网络，因此也节省了计算量。不过 MTCNN 仍然有提升的空间，其第一步的空间金字塔环节计算量巨大，可以通过一个较小的网络先进行尺度筛选，只留下最有可能存在人脸的几个尺度<sup>[107]</sup>，或者只留下可能存在人脸的一些空间位置<sup>[108]</sup>。MTCNN 的另一个缺点在于它的计算时间和显存占用是随着图中人脸数量的变化而变化的，在工程部署时会对系统资源管理带来困难，而 RCNN 和 SSD 系列的算法则不存在这个问题。

在得到了 5 个关键点位置后，用它们与平均脸上的 5 个关键点计算出一个相似变换矩阵。相似变换矩阵只存在面内旋转、整体缩放和横纵坐标平移共四个参数，与仿射变换相比它不会造成面部照片畸变。然后使用相似变换矩阵对原图进行变换，通过双线性插值法即可得到对齐后的人脸图像。

### 2.2.5 特征脸 (EigenFace) 与判别脸 (FisherFace)

特征脸 (EigenFace)<sup>[36]</sup> 与判别脸 (FisherFace)<sup>[37]</sup> 都是上个世纪末期提出的人脸识别算法，它们都是使用图像的原始像素值作为输入，而后经过线性的降维来得到人脸特征进行后续的比对操作。这两个算法中用到的方法到现在仍然被大量用于数据分析当中，本小节将对它们进行一些简单的介绍。

特征脸算法使用了主成分分析法 (Principle Component Analysis, PCA) 来计算降维矩阵，特征脸的计算方法如下：

**算法 2-1** 特征脸算法

**输入:** 人脸像素值拉成的列向量  $X_i$ , 低维空间维度  $d$ 。

- 1: 将所有人脸样本拉成的列向量组成一个矩阵  $X = [X_1, X_2, \dots, X_n]$ ;
- 2: 将每个列向量都减去整体的均值来进行中心化:  $X_i \leftarrow X_i - \frac{1}{n} \sum_{i=1}^n X_i$ ;
- 3: 计算协方差矩阵  $XX^T$ ;
- 4: 对协方差矩阵  $XX^T$  进行特征值分解;
- 5: 取最大的  $d$  个特征值对应的特征向量  $W_1, W_2, \dots, W_d$ 。

**输出:** 投影矩阵  $W = [W_1, W_2, \dots, W_d]$

这里的低维空间维度通常是由用户指定的, 例如原始论文 [36] 中仅使用了 7 个维度。PCA 方法中得到的特征值即为其对应特征向量那个方向上样本的标准差, 所以另一个选择参数  $d$  的方式是由重构误差来进行选择, 例如选择保留  $d$  个特征值之和与全部特征值之和之比大于 99% 的最小的  $d$ 。

特征脸的主要问题在于它是一个无监督的算法, 它只能寻找到使样本尽可能分散开的几个维度, 而分散开也不代表一定能将人脸区分开来, 特征脸无法找到能够使样本之间的间距拉大的那些维度, 判别脸<sup>[37]</sup> 由此而生。

判别脸主要用到的技术为线性判别分析 (Linear Discriminant Analysis, LDA), 它的主要思想也是降维, 但降维的方向被设定为使得同类样本的投影点尽量相近、异类样本的投影点尽量远离的方向。针对  $C$  类的线性判别分析的算法流程如下:

**算法 2-2** 判别脸算法

**输入:** 人脸像素值拉成的列向量  $X_i$ , 每个样本对应的标签  $y(X_i)$ , 低维空间维度  $d$ 。

- 1: 计算每个类别的样本均值  $\mu_i$  和整体均值  $\mu$ ;
- 2: 计算类内散度矩阵  $S_w = \sum_{i=1}^C \sum_{y(X_j)=i} (X_j - \mu_i)(X_j - \mu_i)^T$ ;
- 3: 计算整体散度矩阵  $S_t = \sum_{j=1}^n (X_j - \mu)(X_j - \mu)^T$ ;
- 4: 计算类间散度矩阵  $S_b = S_t - S_w$ ;
- 5: 对矩阵  $S_w^{-1}S_b$  进行特征值分解;
- 6: 取最大的  $d$  个特征值对应的特征向量  $W_1, W_2, \dots, W_d$ 。

**输出:** 投影矩阵  $W = [W_1, W_2, \dots, W_d]$

可以看到, 主成分分析与线性判别分析的主要区别在于散度矩阵的定义, 主成分分析使用整体散度矩阵  $S_t$  进行优化, 而线性判别分析使用类间散度矩阵除以类内散度矩阵  $S_w^{-1}S_b$  进行优化。主成份分析使得全部样本尽可能分散, 而线性判别分析使得类间更加分散、类内更加聚拢, 因此线性判别分析更加适合进行人脸识别任务。

此外, 主成分分析与线性判别分析还有基于概率的解释, 主成分分析是在拟合一个整体的高斯分布, 而线性判别分析则是找到多个均值不同、方差相同的高斯分布之间的分界面, 详情请读者自行上网搜索, 在此不再赘述。

## 2.2.6 深度度量学习

度量学习或者相似度学习<sup>①</sup>是一个比较传统的领域<sup>[109,110]</sup>，在前深度学习时代，通常是需要学习一个 Mahalanobis 距离，Mahalanobis 距离的定义如下：

$$D_W(x_1, x_2) = (x_1 - x_2)^T W (x_1 - x_2), \quad (2-17)$$

当且仅当  $W$  是对称半正定矩阵时， $D_W$  是一个距离度量。而对称半正定矩阵又可以分解为  $D_W = L^T L$ ，因此上式可以化作：

$$D_W(x_1, x_2) = \|Lx_1 - Lx_2\|^2, \quad (2-18)$$

所以传统的度量学习大多数是在学习一个线性变换下的欧氏距离。

而深度度量学习<sup>[111]</sup>将线性变换扩展为了可学习的非线性变换，使得变换能力大大地增强了。目前比较流行的深度度量学习方法有 Contrastive 损失<sup>[59,112]</sup>：

$$\mathcal{L}_C = \begin{cases} \|f_i - f_j\|_2^2, & c_i = c_j \\ \max(0, m - \|f_i - f_j\|_2^2), & c_i \neq c_j \end{cases}, \quad (2-19)$$

和 Triplet 损失<sup>[60,61]</sup>：

$$\mathcal{L}_T = \max(0, m + \|f_i - f_j\|_2^2 - \|f_i - f_k\|_2^2), \quad c_i = c_j, c_i \neq c_k, \quad (2-20)$$

其中这两个  $m$  都表示类间间隔。这两个算法的主要区别在于 Triplet 损失在满足  $m + \|f_i - f_j\|_2^2 - \|f_i - f_k\|_2^2 < 0$  的条件时就不再进行类内距离  $\|f_i - f_j\|_2^2$  的优化，而 Contrastive 损失在什么条件下都要进行类内距离的优化，也就是说 Contrastive 损失能更多地收缩类内距离。需要注意的是，因为优化方程最后都是要达到类内与类间距离的动态平衡状态，所以过分优化类内距离并不一定是好事，一般来说 Triplet 损失能够取得更优秀的性能。

深度度量学习与传统度量学习的区别仅在于特征变换，而其损失函数则可以与传统度量学习完全一致，例如 Triplet 损失实际上也是来自于传统度量学习<sup>[113]</sup>，不过因为在传统度量学习中，直接对特征进行归一化是非常方便的一项操作，而在深度学习中，归一化操作是需要进行反向传播的，其复杂程度会大幅提高。但如果不进行归一化，那么类间间隔参数  $m$  的设置就会非常繁琐，例如在 [59] 中，需要在每训练若干轮之后就停止训练，然后在训练集上重新学习一个参数  $m$  再继

<sup>①</sup> 它们的区别在于度量学习需要满足距离的三定义：正定性、对称性和三角不等式，而相似度学习则比较自由。在实践中这三个条件不一定是完全被遵守的，例如 KL 变换作为距离度量时就不满足对称性，本篇文章中所用的余弦相似度不满足三角不等式，但因为度量学习已经发展成为了一个领域，所以这类方法通常仍然被称为度量学习。

续训练，所以目前的深度度量学习方法大多都需要进行特征的归一化操作。

### 2.2.7 中心损失

深度度量学习的方法比对的是特征与特征之间的距离，因此如何对数据集进行采样来得到成对或者成三元组的样本就成为一大难题。采样方法的好坏会极大影响最终的性能，因此有些研究者将目光转向了在分类的损失函数上增加约束来减小类内距离的方案，由于分类的损失函数不需要采样，因此可以规避采样困难的问题。

注意到在使用 Contrastive 损失的一篇文献 [59] 中，作者发现在于分类的损失函数配合时，仅优化类内距离的性能与同时优化类内距离和类间距离的性能相差不大，因此就诞生了仅仅优化类内距离的想法，中心损失<sup>[63]</sup>就是仅优化类内距离的一个损失函数。它的定义如下：

$$\mathcal{L}_c = \frac{1}{2} \sum_{i=1}^m \|f_i - c_{y_i}\|_2^2, \quad (2-21)$$

其中  $c_{y_i}$  为第  $i$  个样本所对应的类中心。在训练时，需要不断地维护各个类别的类中心向量，在训练一个样本时，就使用梯度下降法根据上式同时对  $f_i$  与  $c_{y_i}$  求梯度来进行更新<sup>①</sup>。

中心损失也需要配合 Softmax 交叉熵损失函数共同训练，由 Softmax 交叉熵损失函数提供分类的监督信号，由中心损失来收缩类内距离。最终的损失函数为  $\mathcal{L}_S + \lambda \mathcal{L}_c$ ，其中  $\mathcal{L}_S$  为 Softmax 交叉熵损失函数， $\lambda$  是平衡两个损失的一个超参数。中心损失的效果如图2-10所示，它能将各个类别的样本收缩为球形，从而将各个类别样本点之间的距离增大。

### 2.2.8 乘性角度间隔

中心损失的一个缺点在于其需要维护所有类别的类中心向量。注意到在 Softmax 交叉熵损失函数中，对于每一类也有一个类似于类中心向量的向量  $w_{y_i}$ ，而中心损失函数因为必须配合 Softmax 交叉熵损失一同使用，这也就意味着参数的冗余，因为人脸的数据集中类别数量通常特别巨大，所以多一倍的参数量往往也是不可接受的。

乘性角度间隔<sup>[64]</sup>则是在 Softmax 交叉熵损失函数的基础上设计的，只使用一个损失函数就能起到增大类间距离和缩小类内距离的作用。乘性角度间隔的出发点是将 Softmax 交叉熵损失函数中的内积操作  $w^T f$  分解为  $\|w\| \|f\| \cos(\theta_{w,f})$ ，这样就

① 在原始论文中，作者为  $c_{y_i}$  使用了一个单独的学习率，但后来在其公开的代码中发现直接使用全局的学习率也是可以的。

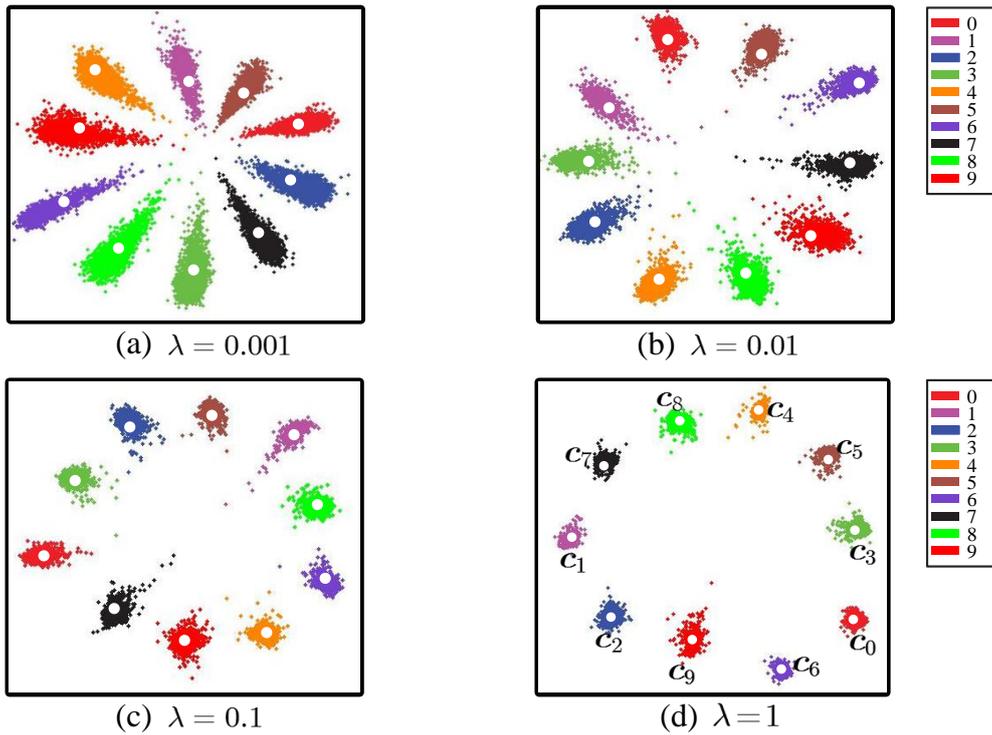


图 2-10 中心损失函数在取不同  $\lambda$  时的效果 [63]。

可以看出 Softmax 交叉熵损失函数主要是在优化特征与类中心向量之间的角度<sup>①</sup>，而如果将所有的  $\|w\|$  强制归一化为 1，那么针对一个给定的  $f$ ，类别之间的判定条件就只剩下特征  $f$  到各个类别的  $w$  之间的角度，因此直接在角度上进行修改是最贴近 Softmax 交叉熵损失函数的工作方式的。进行了权重归一化操作的 Softmax 交叉熵损失函数为：

$$\mathcal{L}_{WS} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|x_i\| \cos(\theta_{y_i,i})}}{\sum_j e^{\|x_i\| \cos(\theta_{j,i})}} \right). \quad (2-22)$$

首先考虑一个二分类问题，特征为  $f$ ，假设这个样本属于第 1 类，两个类别的权重向量分别为  $W_1$  和  $W_2$ ，那么分界面所在的位置即为  $\cos(\theta_{W_1,f}) = \cos(\theta_{W_2,f})$  的位置（图2-11中的  $P_0$  处），而如果要进行度量学习任务，需要满足最小的类间距离要大于最大的类内距离的条件，也就是  $P_1$  与  $P_2$  之间的夹角要大于  $P_1$  与  $P_3$  之间的夹角，即  $\angle_{P_1,P_2} \geq \angle_{P_1,P_3}$ ，此时分界面被移动到了  $P_1$  处。如果假设  $W_1$  恰好就是类中心，也就是说  $\angle_{W_1,P_1} = \angle_{W_1,P_3}$ ，那么就有  $\angle_{W_1,P_1} \leq 3\angle_{W_1,P_2}$ ，如果继续假设各类别分布相同，那么在新的分界面处就有  $\angle_{W_1,P_1} \leq 3\angle_{W_2,P_1}$ 。

于是乘性角度间隔方法增加了一个系数  $m$  乘在了特征与目标类别权重的夹角

① 还有一部分信号在优化与角度正交的幅度。

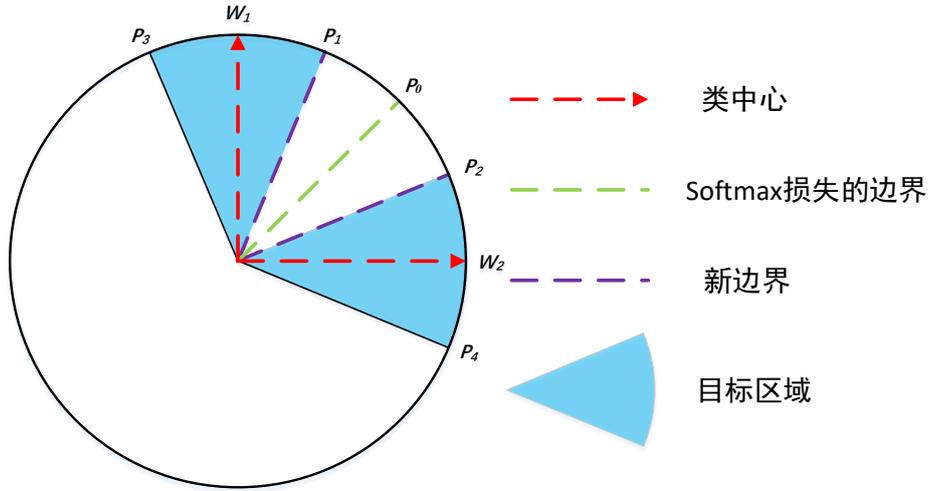


图 2-11 Softmax 交叉熵损失函数的分界面与度量学习要求下的分界面。

$\theta_{y_i,i}$  上, 这样损失函数就变为:

$$\mathcal{L}_{MAS} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|x_i\| \cos(m\theta_{y_i,i})}}{e^{\|x_i\| \cos(m\theta_{y_i,i})} + \sum_{j \neq y_i} e^{\|x_i\| \cos(\theta_{j,i})}} \right). \quad (2-23)$$

根据前文的分析, 这里的  $m$  需要满足条件  $m \geq 3$ , 而作者选取了  $m = 4$ 。但是如果  $m$  取到了 4 时, 函数  $\cos(m\theta)$  在  $\theta \in [0, \pi]$  上会有多次上升和下降, 而  $\cos(\theta_{y_i,i})$  却是单调下降的, 其上升部分会使梯度反向, 从而将特征推离类别权重, 这样有悖于减小类内距离的设计思路。所以作者在每次  $\cos(m\theta)$  上升时都对其进行翻转:

$$\begin{aligned} \psi(\theta_{y_i,i}) &= (-1)^k \cos(m\theta_{y_i,i}) - 2k, \\ \theta_{y_i,i} &\in \left[ \frac{k\pi}{m}, \frac{(k+1)\pi}{m} \right], k \in [0, m-1]. \end{aligned} \quad (2-24)$$

在实际操作时作者还发现由于网络的拟合能力限制, 往往无法达到  $m = 4$  的条件, 因此作者又引入了一个  $\lambda$  参数来中和  $\cos(m\theta_{y_i,i})$  与  $\cos(\theta_{y_i,i})$ , 因此最终的损失函数为:

$$\mathcal{L}_{AS} = -\frac{1}{n} \sum_{i=1}^n \log \frac{e^{\|f_i\| \psi(\theta_{y_i,i})}}{e^{\|f_i\| \psi(\theta_{y_i,i})} + \sum_{j=1, j \neq y_i}^c e^{\|f_i\| \cos(\theta_{j,i})}}, \quad (2-25)$$

其中,

$$\begin{aligned} \psi(\theta) &= \frac{(-1)^k \cos(m\theta) - 2k + \lambda \cos(\theta)}{1 + \lambda}, \\ \theta &\in \left[ \frac{k\pi}{m}, \frac{(k+1)\pi}{m} \right], \frac{(k+1)\pi}{m}, k \in [0, m-1]. \end{aligned} \quad (2-26)$$

乘性角度间隔带来的效果如图2-12所示, 它能同时缩小类内距离并扩大类间

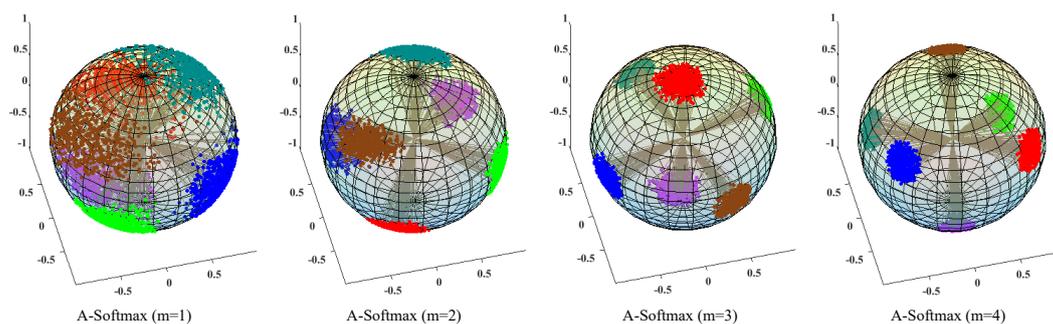


图 2-12 乘性角度间隔中设置不同的  $m$  带来的效果示意图 [64]。

距离，而且是在一个损失函数中完成的这个操作。文中提出的 Softmax 交叉熵损失函数对角度的优化的思想非常新颖，开创了对于 Softmax 改造来适应度量学习任务的新思路，本学位论文的第五章节即为对该方法的一种改进。

## 2.3 本章小结

本章对本学位论文中所要用到的深度学习与人脸识别的基础理论、相关概念进行了简单的介绍，以方便读者阅读后续章节。其中小节 2.2.4 介绍了本学位论文中使用的人脸图像预处理方法，主要是人脸检测与人脸对齐所用到的方法。小节 2.2.7 和小节 2.2.8 还选择了两个与本学位论文处于同一时代的算法进行了较为详细的介绍，这些方法在后续章节中将作为对比算法与本学位论文中所提出的方法进行比较。

本章不仅对各个基础算法进行了介绍，还列出了一些它们目前存在的问题、最新的改进与可能的研究方向，有志于学术的读者可以根据本章所引用的文献来进行扩展阅读，并对其中尚未解决的问题开展研究。

## 第三章 基于 $L_2$ 超球面嵌入的人脸认证损失函数

如前两章所介绍，神经网络在多项计算机视觉任务，尤其是图像分类上取得了令人瞩目的成绩。对于人脸认证任务，基于深度学习的方法也已经在多个数据集上超过了人类，然而之前的大多数方法仍然存在着一些缺陷，例如使用分类的损失函数在训练与测试过程中的不匹配问题，使用度量学习的损失函数又存在着难以找到合适的采样方法的问题，这些问题在以往的工作中往往被一笔带过，没有做过详细的分析，不过这也给后面的研究者留下了很多研究的空间。

本章将深入地分析这些问题，并尝试去解决它们。本章首先探讨了分类的损失函数在训练与测试中使用的相似度量不同的问题，而后将余弦相似度引入训练过程中，发现简单地使用余弦相似度会导致网络模型不收敛的问题。之后本章对该问题进行了理论分析，提出了一个数学理论来解释不收敛的原因，根据这个理论可以推导出一个解决方案：在余弦相似度后引入一个尺度系数，通过这个解决方案本文成功地解决了模型不收敛的问题。

### 3.1 $L_2$ 超球面嵌入的必要性

如第二章介绍，当今的人脸认证损失函数分为两大类：基于分类的损失函数和基于度量学习的损失函数，基于度量学习的损失函数的训练与测试过程是匹配的，而传统的基于分类的方法所采用的 Softmax 交叉熵损失函数则存在着训练与测试不匹配的问题。本节将分析并尝试解决这一问题。

#### 3.1.1 训练与测试的不匹配问题

一个典型的基于分类的方法在训练和测试中的流程图如图3-1所示，这一类的方法通常在训练时使用内积相似度，而在测试时使用余弦相似度来比对两张人脸图像的特征，这就产生了训练与测试之间不匹配的问题。

从数学表达式上来看，余弦相似度的定义如下：

$$S(I_1, I_2) = \frac{\langle F(I_1), F(I_2) \rangle}{\|F(I_1)\| \|F(I_2)\|}. \quad (3-1)$$

可以看到，内积相似度与余弦相似度之间的区别在于余弦相似度多了一步特征归一化的操作，不加这个归一化操作，模型的识别率会有显著的下降。这里本文做了一个小实验来测试加不加特征归一化对最终识别率的影响，本文使用了一个较为优秀的人脸认证模型，它是由 Center Loss<sup>[63]</sup> 在 CASIA-Webface 数据集<sup>[97]</sup> 上训

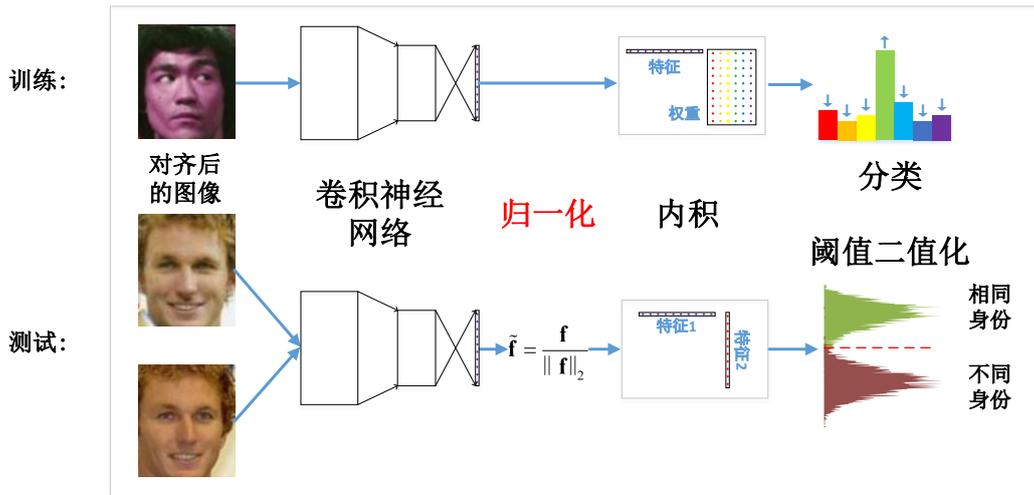


图 3-1 一个典型的基于分类的人脸认证模型在训练和测试时的流程图。

练得到的<sup>①</sup>，用它在不限定使用额外数据的 LFW 协议上进行测试，得到了以下结果：

表 3-1 特征归一化的影响

相似度度量	归一化前	归一化后
内积相似度	98.27%	98.98%
欧氏距离	98.35%	98.95%

从表中可以看出，特征归一化操作能够带来 0.6% ~ 0.7% 的提升，这对一个已经达到 98% 识别率以上的模型来说已经是非常显著的数值了。既然在测试时加和不加特征归一化的区别这么明显，那如果在训练的时候也加入特征归一化操作，使得训练与测试时使用的相似度相吻合，相信也能使模型的性能得到提升。

这种使训练与测试匹配的策略在深度学习领域叫作端到端学习，让神经网络直接对测试时的指标进行优化，往往能得到更好的结果。这种策略的优势尚未得到证明，现在只是作为一种思想观念在深度学习社区中流行，但直觉上来说，在训练时使用与测试时同样的评价标准，应当会得到更好的结果。

### 3.1.2 Softmax 交叉熵损失下的特征分布

上一小节提到了在测试时将特征归一化会使识别率得到显著的提升，这个现象在之前的很多人脸认证文章中都有所提及<sup>[57,63]</sup>，但在这些文章中，余弦相似度通常被作为一个小技巧来提升测试时的性能，而没有对其数学原理进行分析，本

① 模型下载地址：<https://github.com/ydwen/caffe-face>

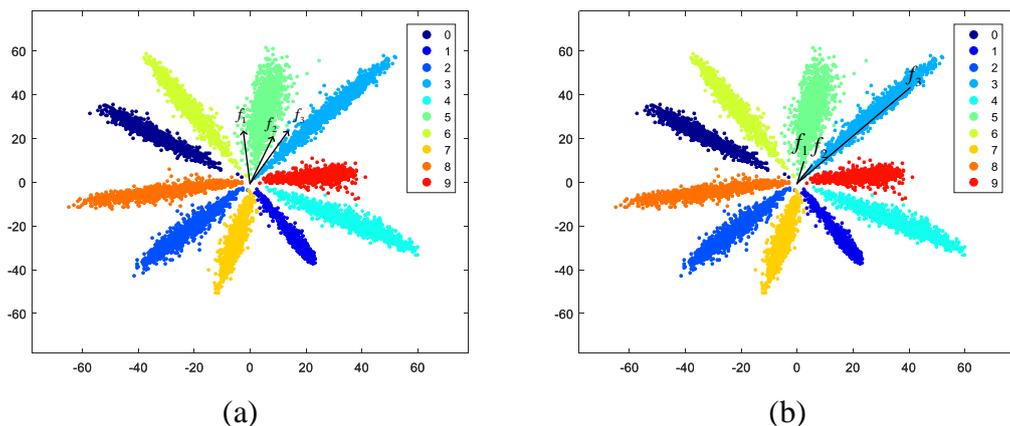


图 3-2 在 MNIST 数据集上，使用 Softmax 交叉熵损失函数训练得到的二维特征分布。

(a) 欧氏距离下该模型判别失败的例子；(b) 内积相似度下该模型判别失败的例子。

节将深入地探讨这一问题的数学本质。

为了能对 Softmax 交叉熵损失函数训练出的特征有一个直观的感受，本文做了一个小实验，这个实验是在 MNIST 数据集<sup>[14]</sup>上进行的，MNIST 数据集是一个包含了 6 万张手写数字 0-9 的数据集，经常被用作测试模型正确与否的标杆。本文构建了一个含有 6 个卷积层、3 个池化层和 2 个特征层的卷积神经网络，并将最后一个特征层的输出维度设置为 2，这样每一张手写数字图像进入网络后都可以输出 2 维的特征。因为特征只有 2 维，所以可以很方便地把它画在一个二维的平面上。如图 3-2 所示，在 (a) 图中特征  $f_1$  和  $f_2$  之间的内积相似度小于  $f_2$  与  $f_3$  之间的内积相似度，但是  $f_1$  和  $f_2$  才是来自同一个类别的特征；在 (b) 图中，特征  $f_1$  和  $f_2$  之间的欧氏距离远小于  $f_2$  与  $f_3$  之间的欧氏距离，但是  $f_2$  和  $f_3$  才是来自同一个类别的特征。

在图 3-2 上还可以观察到，Softmax 交叉熵损失函数训练出来的特征呈辐射状分布，在这种分布下，特征的长度改变将不影响该特征的类别判定，而不同类之间的区别主要由角度来决定，因此内积、欧氏距离这些与幅度相关的相似性度量的效果就不如余弦这种与幅度无关的相似性度量。

至于 Softmax 交叉熵损失函数训练出的特征分布为什么呈辐射型，是因为如果将特征幅度放大或缩小，那这个特征与各类别权重之间的相似度也会同时放大或缩小，这时最大的相似度仍旧还是最大的，不会对该特征被归到的类别产生影响。用正式的数学表达式来表述如下：首先列出 Softmax 交叉熵损失函数的定义

式:

$$\mathcal{L}_S = -\frac{1}{m} \sum_{i=1}^m \log \frac{e^{W_{y_i}^T f_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T f_i + b_j}}, \quad (3-2)$$

其中  $m$  是训练样本数、 $n$  是类别总数、 $f_i$  是第  $i$  个样本的特征、 $y_i \in [1, n]$  是第  $i$  个样本对应的标签、 $W$  和  $b$  是 Softmax 前面的内积层的权重矩阵和偏置向量、 $W_j$  是  $W$  的第  $j$  列对应第  $j$  个类别。Softmax 交叉熵损失函数是 Softmax 操作与交叉熵损失函数的组合，其中 Softmax 操作可以得到特征  $f$  在第  $i$  个类别下的概率:

$$P_i(f) = \frac{e^{W_i^T f + b_i}}{\sum_{j=1}^n e^{W_j^T f + b_j}}. \quad (3-3)$$

在测试过程中，使用如下公式来判断一个样本的类别:

$$\text{Class}(f) = i = \arg \max_i (W_i^T f + b_i). \quad (3-4)$$

在这种条件下，可以推导出:

**引理 3.1** 若特征  $f$  被判定为第  $i$  类，则有  $(W_i f + b_i) - (W_j f + b_j) \geq 0, \forall j \in [1, n]$ .

**证明:** 根据  $\arg \max$  的定义可知。 ■

利用该不等式，可以得出如下命题:

**命题 3.1** 对于**无偏置项**的 Softmax 操作，对任意的  $s > 1$ ，若  $i = \arg \max_j (W_j^T f)$ ，那么  $P_i(sf) \geq P_i(f)$  总是成立。

**证明:** 令  $t = s - 1$ ，在经过尺度放大后，**无偏置项**的 Softmax 概率等于:

$$\begin{aligned} P_i(sf) &= \frac{e^{W_i^T [(1+t)f]}}{\sum_{j=1}^n e^{W_j^T [(1+t)f]}} \\ &= \frac{e^{W_i^T f}}{\sum_{j=1}^n e^{W_j^T f + t(W_j^T f - W_i^T f)}}. \end{aligned} \quad (3-5)$$

根据引理3.1可以推导出  $t(W_j^T f - W_i^T f) \leq 0$  总是成立，代入上式可得:

$$\begin{aligned} P_i(sf) &\geq \frac{e^{W_i^T f}}{\sum_{j=1}^n e^{W_j^T f}} \\ &= P_i(f). \end{aligned} \quad (3-6)$$

这里的等号在  $W^T f = 0$  或者  $W_i = W_j, \forall i, j \in [1, n]$  时成立，这两种情况在实际模型中都几乎不可能发生。证明完毕。 ■

由于 Softmax 交叉熵损失函数总会使目标类别对应的概率值变得更大，因此

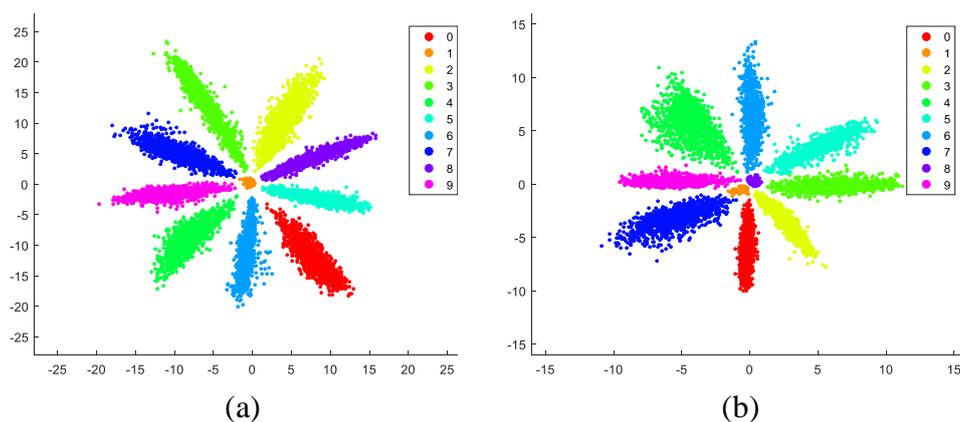


图 3-3 带有偏置项的内积操作下训练得到的散点图。(a) 中心有一个类别的点簇；(b) 中心有两个类别的点簇。

这个定理反映出 **Softmax** 交叉熵损失函数总是会鼓励已经被分类正确的特征得到更大的幅度。在迭代过程中，特征的幅度会被越拉越大，这就是为什么 **Softmax** 交叉熵损失函数训练出的特征总是呈现出辐射状分布的原因。由于前文的分析，幅度的拉长对类别的判定并没有影响，而在比较相似度的过程中，如果乘以特征的长度，反而会干扰到相似度的比对（如图3-2所示），因此在测试过程中需要对特征进行归一化处理来消除特征长度对相似度的影响。

同样的，在训练过程中，**Softmax** 交叉熵损失函数同时优化了特征幅度和特征与权重之间的角度，由于优化特征幅度并不能影响类别的判定，因此要采取措施避免 **Softmax** 交叉熵损失函数对特征幅度进行优化。经过归一化之后，**Softmax** 交叉熵损失函数将不再优化特征幅度而只优化特征与权重之间的角度，从而使 **Softmax** 交叉熵损失函数的优化更有效率。

注意到命题3.1强调了该定理只适用于无偏置项的 **Softmax** 操作，对于带有偏置项的 **Softmax** 交叉熵损失函数来说，如果两个类别的权重完全一致，模型仍然能够通过偏置项上的差别来分类。在实验过程中也发现了这样的例子，如图3-3所示，某些类别被收缩在原点附近，如果对这些原点附近的特征进行归一化操作，那它们将会与其他类别混淆在一起，无法进行有效的区分，此时特征归一化操作反而会降低模型的性能。为了避免这样的情况出现，尽管偏置项在图像分类领域是一个常用的操作，本篇文章中也不会最后在最后一个内积层上加入偏置项。

### 3.1.3 Softmax 交叉熵损失函数的饱和问题

如图3-4所示画出了数字 0 所在类别的概率值，从图上可以看到在边界上 **Softmax** 概率  $P_0$  迅速从 0 升到 1，边界区域所占面积较小，有大面积的区域概率值

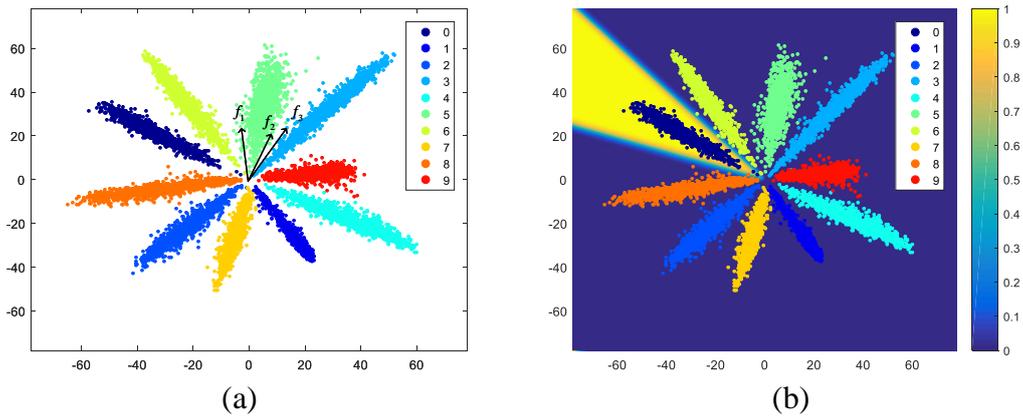


图 3-4 (a) 使用 Softmax 交叉熵损失函数训练得到的二维特征分布；(b) 第 0 类的概率值。

都接近于 1，由于 Softmax 对当前优化的类别的梯度为  $P_0 - 1$ ，所以除了边界之外，大部分区域的梯度都非常小。总之，Softmax 交叉熵损失函数仅仅只负责将各类别分开即可，而不是进一步优化类内方差，这种现象本文称其为 Softmax 交叉熵损失函数的饱和问题。

作为一个分类的损失函数，Softmax 交叉熵损失函数的饱和问题并不影响其分类的功能，若不考虑泛化误差，一个分类器能找到一系列的分界面将各类分开即可。绘制图3-4所使用的模型分类精度已经大于 99%，是一个优秀的分类器模型。但是从图上看，虽然类别之间有明显的分界面能把各类很好地分开，但是对于特征比对问题（即度量学习问题）来说，该模型并不理想，例如特征  $f_1$  和  $f_2$  之间的内积相似度小于  $f_2$  与  $f_3$  之间的内积相似度，而  $f_1$  和  $f_2$  才是来自同一个类别的特征。如果想让类间距离永远大于类内距离，就需要进一步地对类内距离进行优化，例如文献 [63] 中就显式地加入了一个类内距离约束项来进一步收缩类内距离。

那么如何约束 Softmax 交叉熵损失函数来进一步收缩类内距离呢？一个比较直观的方法是扩大边界的范围，使得概率在边界处缓慢地增加，这样梯度就不会迅速地降为接近于 0 的数，从而不断地向类中心收缩。小节3.2.3将会详细讨论如何实现这一方法。

### 3.2 使用 Softmax 交叉熵损失函数优化余弦相似度

上一节讨论了传统的 Softmax 交叉熵损失函数用在度量学习上的一些特性以及产生的一系列问题，可以看到，传统的 Softmax 交叉熵损失函数并不是很适合人脸认证这一度量学习任务。本节将会讨论如何通过改造传统的 Softmax 交叉熵损

失函数来解决这些问题，使其更加适合进行人脸认证任务。

### 3.2.1 在神经网络中使用余弦相似度

上一节反复讨论了在训练中将内积相似度修改为余弦相似度的必要性，如前文所述，余弦相似度与内积相似度的区别在于余弦相似度要对向量进行  $L_2$  范数归一化，在这里本文修改了  $L_2$  范数的定义：

$$\|\mathbf{x}\|_2 = \sqrt{\sum_i \mathbf{x}_i^2 + \varepsilon}, \quad (3-7)$$

其中  $\varepsilon$  是一个比较小的数字来防止分母除零的错误，在实现时可以使用  $10^{-8}$  或更小的数值。有了新的  $L_2$  范数的定义，就可以接着定义  $L_2$  归一化，对于一个输入向量  $\mathbf{x} \in \mathcal{R}^n$ ， $L_2$  归一化层会输出归一化后的向量：

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2} = \frac{\mathbf{x}}{\sqrt{\sum_i \mathbf{x}_i^2 + \varepsilon}}. \quad (3-8)$$

这里的  $\mathbf{x}$  可以是特征向量  $\mathbf{f}$ ，也可以是权重矩阵的一列  $\mathbf{W}_i$ 。在反向传播过程中， $L_2$  归一化层的梯度可以通过链式法则来得到：

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}_i} = \frac{\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}_i} - \tilde{\mathbf{x}}_i \sum_j \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}_j} \tilde{\mathbf{x}}_j}{\|\mathbf{x}\|_2}. \quad (3-9)$$

**证明：**对于这个公式的推导需要将公式3-8中的分子分母看作独立的两项，分别求导后使用链式法则将两项加起来：

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{x}_i} &= \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}_i} \frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{x}_i} + \sum_j \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}_j} \frac{\partial \tilde{\mathbf{x}}_j}{\partial \|\mathbf{x}\|_2} \frac{\partial \|\mathbf{x}\|_2}{\partial \mathbf{x}_i} \\ &= \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}} \frac{1}{\|\mathbf{x}\|_2} + \sum_j \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}_j} \frac{-\mathbf{x}_j}{\|\mathbf{x}\|_2^2} \cdot \frac{1}{2} \frac{1}{\|\mathbf{x}\|_2} \cdot 2\mathbf{x}_i \\ &= \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}} \frac{1}{\|\mathbf{x}\|_2} - \frac{\mathbf{x}_i}{\|\mathbf{x}\|_2} \sum_j \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}_j} \frac{\mathbf{x}_j}{\|\mathbf{x}\|_2^2} \\ &= \frac{\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}_i} - \tilde{\mathbf{x}}_i \sum_j \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}_j} \tilde{\mathbf{x}}_j}{\|\mathbf{x}\|_2}. \end{aligned} \quad (3-10)$$

值得一提的是， $L_2$  归一化有一个比较有趣的性质：

**性质 3.1** 输入向量  $\mathbf{x}$  与网络对它的梯度  $\frac{\partial \mathcal{L}}{\partial \mathbf{x}}$  相互垂直。

**证明：**将公式3-9写成向量形式：

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}} - \tilde{\mathbf{x}} \langle \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}}, \tilde{\mathbf{x}} \rangle}{\|\mathbf{x}\|_2}. \quad (3-11)$$

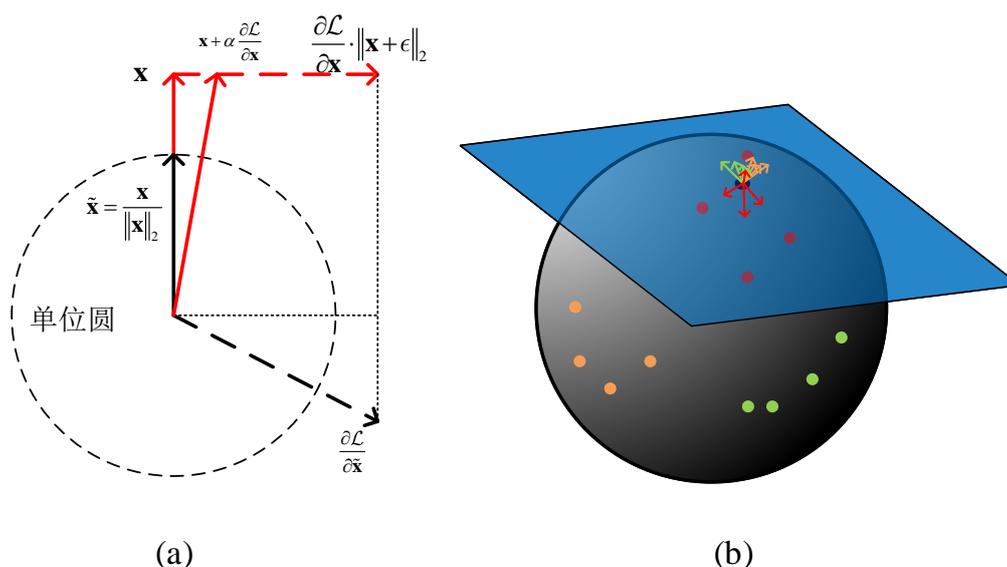


图 3-5 (a) 二维空间下  $L_2$  归一化操作和它的梯度；(b) 三维球面上一个分类器得到的梯度示意图。

那么，

$$\begin{aligned}
 \left\langle \mathbf{x}, \frac{\partial \mathcal{L}}{\partial \mathbf{x}} \right\rangle &= \frac{\langle \mathbf{x}, \frac{\partial \mathcal{L}}{\partial \mathbf{x}} \rangle - \langle \mathbf{x}, \tilde{\mathbf{x}} \rangle \langle \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}}, \tilde{\mathbf{x}} \rangle}{\|\mathbf{x}\|_2} \\
 &= \frac{\langle \mathbf{x}, \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}} \rangle - \frac{\langle \mathbf{x}, \mathbf{x} \rangle \langle \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}}, \mathbf{x} \rangle}{\|\mathbf{x}\|_2^2}}{\|\mathbf{x}\|_2} \\
 &= \frac{\langle \mathbf{x}, \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}} \rangle - \langle \tilde{\mathbf{x}}, \tilde{\mathbf{x}} \rangle \langle \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}}, \mathbf{x} \rangle}{\|\mathbf{x}\|_2} \\
 &= \frac{\langle \mathbf{x}, \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}} \rangle - \langle \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}}, \mathbf{x} \rangle}{\|\mathbf{x}\|_2} \\
 &= 0.
 \end{aligned} \tag{3-12}$$

由内积的性质可知，若两向量的内积为 0，则这两个向量相互垂直，证毕。 ■

从几何的角度来看，对向量  $\mathbf{x}$  的梯度  $\frac{\partial \mathcal{L}}{\partial \mathbf{x}}$  是网络传递过来的梯度  $\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}}$  在法向量  $\tilde{\mathbf{x}}$  对应的单位圆的切空间上的投影 (如图3-5所示)。从图3-5(a) 可以看出，根据勾股定理， $\|\mathbf{x} + \alpha \frac{\partial \mathcal{L}}{\partial \mathbf{x}}\|_2$  总是比  $\|\mathbf{x}\|_2$  要大，因此在迭代更新过程中  $\|\mathbf{x}\|_2$  会不断地增大。为了防止  $\|\mathbf{x}\|_2$  无限制地扩张，需要对  $\|\mathbf{x}\|_2$  进行  $L_2$  范数的权重衰减来抑制它的增大。

如图3-5(b) 所示，蓝色平面表示蓝色点处单位球的切空间，所有对蓝色点的梯度都在这个切空间内。红色、黄色、绿色的点分别是 3 个不同类别的归一化后的特征向量，蓝色点是红色点所对应类别中的一个。这里假设一个分类器会使得类

内距离拉近、类间距离拉远，那么他们的梯度就会如图中箭头所示，最后所有的这些梯度都会加在一起形成对蓝色点的梯度，所有的这些梯度也都在蓝色点处单位球的切空间上。

有了  $L_2$  归一化层，就可以在内积层的基础上构建余弦相似度层了，余弦相似度层以特征和各类的权重向量为输入，输出特征与各类特征权重之间的余弦相似度：

$$d(\mathbf{f}, \mathbf{W}_i) = \frac{\langle \mathbf{f}, \mathbf{W}_i \rangle}{\|\mathbf{f}\|_2 \|\mathbf{W}_i\|_2}. \quad (3-13)$$

使用这个层替换掉原网络中最后一个内积层，那 Softmax 就从内积相似度转而优化余弦相似度了。

### 3.2.2 直接应用余弦相似度遇到的困难与解决方案

然而通过实验发现直接进行一个简单的替换并不能训练得到一个好的模型，在实际的训练中，网络的损失下降了一点之后就收敛到一个很大的值上，之后不论是修改学习率还是降低权重衰减系数都不能使网络进一步地收敛。

造成这一现象的主要原因是做了归一化以后， $d(\mathbf{f}, \mathbf{W}_i)$  的取值范围只有  $[-1, 1]$ ，而一个正常的不做归一化的相似度绝对值大概在 20 到 80 之间。如此低的值域会造成即使在样本分得很好时，目标概率  $P_y$  仍然会非常小，举一个极端的例子：当样本与目标类别的相似度为最大值 1、与  $n-1$  个非目标类别的相似度全部为 -1 时， $P_y = \frac{e^1}{e^1 + (n-1)e^{-1}}$  仍旧非常的小，当  $n = 1,000$  时甚至只有 0.007。由于 Softmax 的梯度为  $P_y - 1$ ，在这种情况下即使一个样本已经分得非常好，它仍旧会得到一个非常大的梯度，相对而言，比较困难的样本就无法得到充足的梯度来训练。

为了能够更清晰地理解这一问题，本文对 Softmax 交叉熵损失函数在不同的值域范围下能得到的损失下限进行了分析。这里下限的意思就是在最理想的情况下，损失能降到的最低限度。为了得到这个下限，首先要定义什么是最理想的情况，显然当所有样本都与各自样本所对应的类别权重重合、且权重之间的夹角达到最大时，损失能达到最小值。

**定理 3.1** (Softmax 交叉熵损失在归一化后的下界) 假设每个类别都有同样多的样本数且所有的样本都已经被完全分好时，如果将特征和权重的每一列的  $L_2$  范数都归一化到  $\ell$ ，那 Softmax 交叉熵损失函数在最理想的情况下能达到的最低损失为  $\log\left(1 + (n-1)e^{-\frac{n}{n-1}\ell^2}\right)$ ，其中  $n$  是类别个数。

**证明：**由于已经假设了所有样本都已经完全分好，所以本文直接使用  $\mathbf{W}_i$  来代替第  $i$  个类别的样本特征。

Softmax 交叉熵损失函数的定义如下：

$$\mathcal{L}_S = -\frac{1}{n} \sum_{i=1}^n \log \frac{e^{\mathbf{W}_i^T \mathbf{W}_i}}{\sum_{j=1}^n e^{\mathbf{W}_i^T \mathbf{W}_j}}. \quad (3-14)$$

这个方程与公式3-2稍有不同，这是因为本证明假设了每个类别的样本数都相同，所以一个类别中的样本数量可以省略掉。在右边的分子分母上同时除以  $e^{\mathbf{W}_i^T \mathbf{W}_i} = e^{\ell^2}$  可得，

$$\begin{aligned} \mathcal{L}_S &= -\frac{1}{n} \sum_{i=1}^n \log \frac{1}{1 + \sum_{j=1, j \neq i}^n e^{\mathbf{W}_i^T \mathbf{W}_j - \ell^2}} \\ &= \frac{1}{n} \sum_{i=1}^n \log \left( 1 + \sum_{j=1, j \neq i}^n e^{\mathbf{W}_i^T \mathbf{W}_j - \ell^2} \right). \end{aligned} \quad (3-15)$$

由于  $f(x) = e^x$  是一个凸函数，根据 Jensen 不等式有  $\frac{1}{n} \sum_{i=1}^n e^{x_i} \geq e^{\frac{1}{n} \sum_{i=1}^n x_i}$ ，因此，

$$\mathcal{L}_S \geq \frac{1}{n} \sum_{i=1}^n \log \left( 1 + (n-1) e^{\frac{1}{n-1} \sum_{j=1, j \neq i}^n (\mathbf{W}_i^T \mathbf{W}_j - \ell^2)} \right). \quad (3-16)$$

这个等号当且仅当所有的  $\mathbf{W}_i^T \mathbf{W}_j, 1 \leq i < j \leq n$  都相等时取到，此时各类别的权重向量之间的夹角相等。然而，在  $d$  维空间中只能保证有  $d+1$  个不同的权重向量之间的夹角相等，此时这  $d+1$  个权重向量构成一个正  $d$  单纯形<sup>[115]</sup>，比如说正 2 单纯形是等边三角形而正 3 单纯形是正四面体。由于在人脸认证任务中类别数总是大于维度，所以这个等号在实际中往往并不能取到，由于本证明忽略了这一点，所以一个可能的改进是在放缩时将维度也考虑进去，这样可能可以得到一个更紧的下界。

类似于之前的  $f(x) = e^x$ ，Softplus 函数  $s(x) = \log(1 + Ce^x)$  在  $C > 0$  的条件下也是凸函数，所以  $\frac{1}{n} \sum_{i=1}^n \log(1 + Ce^{x_i}) \geq \log(1 + Ce^{\frac{1}{n} \sum_{i=1}^n x_i})$ ，于是可以继续推导，

$$\begin{aligned} \mathcal{L}_S &\geq \log \left( 1 + (n-1) e^{\frac{1}{n(n-1)} \sum_{i=1}^n \sum_{j=1, j \neq i}^n (\mathbf{W}_i^T \mathbf{W}_j - \ell^2)} \right) \\ &= \log \left( 1 + (n-1) e^{\left( \frac{1}{n(n-1)} \sum_{i=1}^n \sum_{j=1, j \neq i}^n \mathbf{W}_i^T \mathbf{W}_j \right) - \ell^2} \right). \end{aligned} \quad (3-17)$$

这个等号当且仅当  $\forall \mathbf{W}_i, \sum_{j=1, j \neq i}^n \mathbf{W}_i^T \mathbf{W}_j$  全部相等时取到。

注意到将  $\|\sum_{i=1}^n \mathbf{W}_i\|_2^2$  展开可得，

$$\left\| \sum_{i=1}^n \mathbf{W}_i \right\|_2^2 = n\ell^2 + \sum_{i=1}^n \sum_{j=1, j \neq i}^n \mathbf{W}_i^T \mathbf{W}_j, \quad (3-18)$$

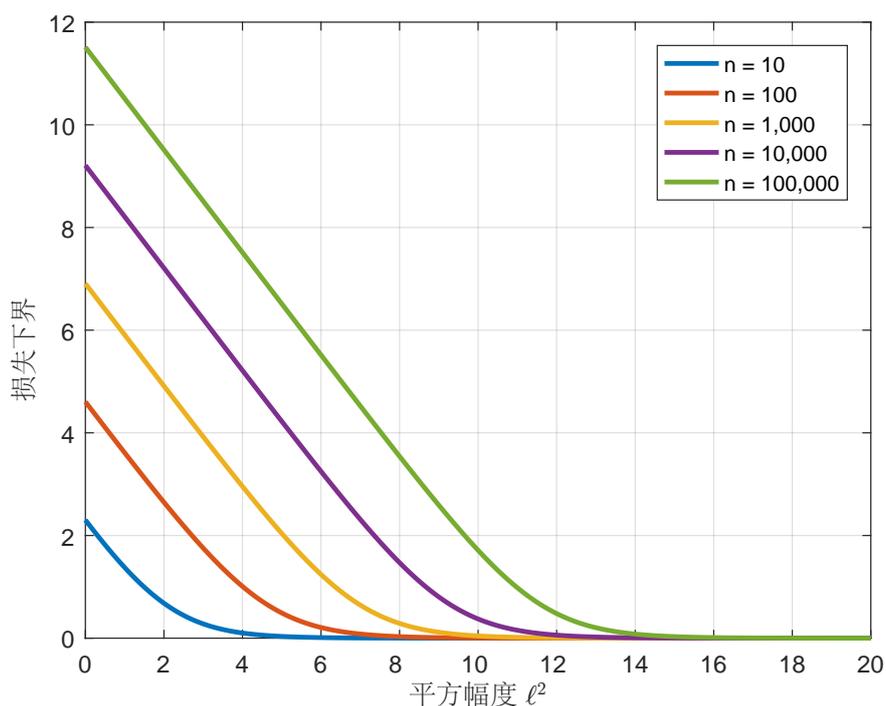


图 3-6 Softmax 交叉熵损失函数的下界与特征和权重的幅度之间的关系。

所以,

$$\sum_{i=1}^n \sum_{j=1, j \neq i}^n \mathbf{w}_i^T \mathbf{w}_j \geq -n\ell^2. \quad (3-19)$$

这个不等式里的等号当且仅当  $\sum_{i=1}^n \mathbf{W}_i = 0$  时取到, 因此,

$$\begin{aligned} \mathcal{L}_S &\geq \log \left( 1 + (n-1) e^{-\frac{n\ell^2}{n(n-1)} - \ell^2} \right) \\ &= \log \left( 1 + (n-1) e^{-\frac{n}{n-1} \ell^2} \right). \end{aligned} \quad (3-20)$$

证明完毕。 ■

尽管这个证明比较长, 但是因为在这个证明里有一些关于超球面流形的性质, 可以加深对超球面嵌入的理解, 所以推荐读者耐心阅读。

这个下界从数学上解释了为什么将特征和权重归一化到 1 时, Softmax 交叉熵损失会停留在一个非常大的数值而无法收敛的原因。举一个实际的例子, 在 CASIA-Webface 上训练时 ( $n = 10575$ ), 损失大概会从 9.27 下降到 8.5 左右, 而在这种情况下推导出的下界为 8.27, 非常接近于实际数值, 考虑到实际训练中还要加入权重衰减约束项, 这里推导得到的下界可以算是一个非常紧的数值。为了使

读者能够对这个下界有一个直观的感受，如图3-6所示绘制了在不同的  $n$  的取值下，Softmax 交叉熵损失函数的下界与特征和权重的幅度之间的关系，注意到由于本文在实现的时候是直接在余弦值上乘以一个系数，而余弦值是由两个向量计算得到的，所以图3-6的  $x$  轴实际上是幅度的平方  $\ell^2$ 。

在得到了这个下界之后，不收敛问题的解决方案也非常清晰了：将特征和权重矩阵的每一列都归一化到一个较大的数而不是 1，则损失就能够继续下降了。在实现时，可以在余弦相似度层后追加一个尺度变换层，该层只有一个可学习的参数  $s = \ell^2$ ，也可以参考图3-6来固定住一个  $s$ ，比如 20 或者 30 都比较适合 1 万类的人脸认证模型。但是为了避免引入新的超参数，这里采用的是通过反向传播来自动学习参数  $s$  的方案。最终，余弦 Softmax 损失函数定义为：

$$\mathcal{L}_{CS} = -\frac{1}{m} \sum_{i=1}^m \log \frac{e^{s\tilde{\mathbf{W}}_i^T \tilde{\mathbf{f}}_i}}{\sum_{j=1}^n e^{s\tilde{\mathbf{W}}_j^T \tilde{\mathbf{f}}_i}}. \quad (3-21)$$

### 3.2.3 分析与讨论

#### 3.2.3.1 参数 $s$ 的升降条件

在命题3.1中，只讨论了已经分类正确情况下的 Softmax 操作的特性，为了得到一个更加通用的结论，需要对神经网络的梯度进行分析。这里本文称  $z_i = s\tilde{\mathbf{W}}_i^T \tilde{\mathbf{f}} = s \cdot \cos(\theta_i)$  为第  $i$  类的分数<sup>①</sup>，而样本标签所对应的分数叫目标分数，不是样本标签所对应的分数叫非目标分数，那么单个样本的余弦 Softmax 交叉熵损失函数可化简为：

$$\mathcal{L}_{CS} = -\log \frac{e^{z_y}}{\sum_{j=1}^n e^{z_j}}, \quad (3-22)$$

$\mathcal{L}_{CS}$  对  $z_i$  的梯度为：

$$\frac{\partial \mathcal{L}_{CS}}{\partial z_i} = \begin{cases} P_i(\mathbf{f}) - 1, & i = y \\ P_i(\mathbf{f}), & i \neq y \end{cases}, \quad (3-23)$$

其中  $y$  是当前样本对应的标签。而  $z_i$  对特征幅度  $s$  的梯度为：

$$\frac{\partial z_i}{\partial s} = \cos(\theta_i) \quad (3-24)$$

<sup>①</sup> 在一些文献<sup>[116]</sup>中这个分数被称为对数比率 (logit)，但实际上此处既没有对数、也没有比率，所以本文使用分数一词代替，使其意义更加明确。

结合公式3-23与公式3-24可得：

$$\frac{\partial \mathcal{L}_{CS}}{\partial s} = \sum_{i=1}^C P_i(\mathbf{f}) \cos(\theta_i) - \cos(\theta_y), \quad (3-25)$$

即余弦 Softmax 损失函数对参数  $s$  的导数为 Softmax 操作加权平均后的分数减去目标分数，在分类正确的情况下，目标分数大于所有非目标分数，加权平均后的分数也小于目标分数，因此  $\frac{\partial \mathcal{L}_{CS}}{\partial s} < 0$ ，经过梯度下降算法， $s$  会不断上升。

令  $\frac{\partial \mathcal{L}_{CS}}{\partial s} = 0$ ，可得：

$$\cos(\theta_y) = \frac{\sum_{i=1, i \neq y}^C P_i(\mathbf{f}) \cos(\theta_i)}{1 - P_y(\mathbf{f})}. \quad (3-26)$$

由于  $\sum_{i=1}^C P_i(\mathbf{f}) = 1$ ，定义非目标 Softmax 操作为：

$$P_i^-(\mathbf{f}) = \frac{e^{\mathbf{w}_i^T \mathbf{f}}}{\sum_{j=1, j \neq y}^n e^{\mathbf{w}_j^T \mathbf{f}}}, \quad i \neq y. \quad (3-27)$$

结合公式3-25和公式3-27，可以得到在  $\frac{\partial \mathcal{L}_{CS}}{\partial s} = 0$  的条件下，有：

$$\cos(\theta_y) = \sum_{i=1, i \neq y}^C P_i^-(\mathbf{f}) \cos(\theta_i). \quad (3-28)$$

这个公式的含义是：

**定理 3.2** 当目标分数大于非目标 Softmax 加权的非目标分数时， $s$  会增大，反之亦然。

**证明：** 见前文推导过程。 ■

在训练过程中，目标分数起初会小于非目标 Softmax 加权的非目标分数，此时  $s$  会逐渐减小。随着训练的进行，当目标分数大于非目标 Softmax 加权的非目标分数时， $s$  会增大。在训练末期，目标分数通常会比所有的非目标分数大很多，但根据公式3-23，训练末期整体的梯度都会较低，因此在训练末期虽然  $s$  会持续增大，但增幅较为缓慢。如果再给  $s$  添加一个  $L_2$  范数的衰减约束来限制其大小，那么  $s$  最终会收敛到一个较大的值。

由于非目标 Softmax 加权的非目标分数总是小于最大的非目标分数，因此  $s$  并不是在分类正确之后才开始增大，实际上  $s$  只在训练初始的一小段迭代中是在减小的，后面都在不断增大，所以该定理是一个比命题3.1更精确的描述。

### 3.2.3.2 参数 $s$ 控制下的分类器权重分配

由公式3-23可以看出，Softmax 交叉熵损失函数对各个分数的导数只与 Softmax 概率  $P_i$  相关，从这个公式中可以推导出 Softmax 交叉熵损失的三个性质：

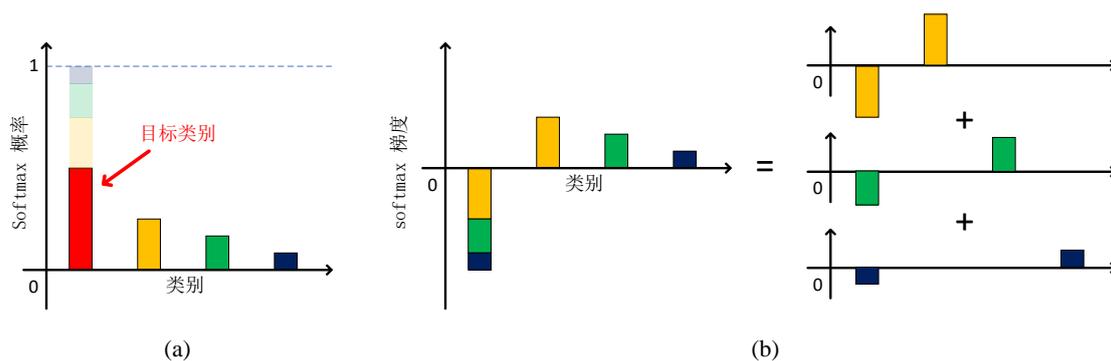


图 3-7 (a) 4 类情况下的 Softmax 概率示意图; (b) Softmax 交叉熵损失函数对各分数的梯度示意图。

- (1) Softmax 交叉熵损失对分数的梯度之和为 0，即  $\sum_{i=1}^n \frac{\partial \mathcal{L}_S}{\partial z_i} = 0$ ;
- (2) Softmax 交叉熵损失对非目标分数的梯度之和等于对目标分数梯度的绝对值，即  $\sum_{i=1, i \neq y}^n \frac{\partial \mathcal{L}_S}{\partial z_i} = \left| \frac{\partial \mathcal{L}_S}{\partial z_y} \right|$ ;
- (3) Softmax 交叉熵损失对分数的梯度绝对值之和为 2 倍的对目标分数之和，即  $\sum_{i=1}^n \left| \frac{\partial \mathcal{L}_S}{\partial z_i} \right| = 2 \left| \frac{\partial \mathcal{L}_S}{\partial z_y} \right|$ 。

根据这三条性质，如图3-7(b)所示本文画出了 Softmax 交叉熵损失函数对各个分数的导数，从图上可以看出，一个用于多分类的 Softmax 交叉熵分类器实际上等价于多个二分类器，这些二分类器之间的权重由公式3-27中的  $P_i^-$  来决定，这些二分类器的总体权重则由  $P_y$  来决定。

在这些  $P_y$  和  $P_i^-$  中，参数  $s$  起到了非常重要的作用。 $s$  值越小，Softmax 操作得到的数值就越平缓， $s$  值越大，Softmax 操作的结果就衰减得越快。其中两个极端是当  $s = 0$  时，Softmax 的输出全部为  $\frac{1}{n}$ ；当  $s \rightarrow +\infty$  时，Softmax 等效于 one-hot 模式下的 max 操作，即在最大值处取 1，其他位置全部为 0。

如图3-8所示，本文绘制了 Softmax 操作前后的非目标分数的变化，从图3-8(b)上可以看到  $P_i^-$  呈指数衰减，且当  $s = 30$  时衰减速度极快，在这种情况下起作用的大部分都是较大的非目标分数所对应的二分类器。

随着训练过程的进行， $s$  会不断增大，这也就意味着能起到作用的二分类器越来越少，Softmax 交叉熵损失慢慢地由全局损失变成只关心局部邻域的损失函数。全局损失能带来更多的监督信号，有利于快速收敛，而局部损失能更精确地将目标类别与相邻的类别区分开，有利于模型的精度，也就是说  $s$  的增大过程也就是一个从快速收敛到精确微调的过程。

与 Contrastive Loss 和 Triplet Loss 这类自始至终都只关心局部的损失函数相比，基于 Softmax 交叉熵损失函数的模型因为在一开始全局损失的特性，因此其收

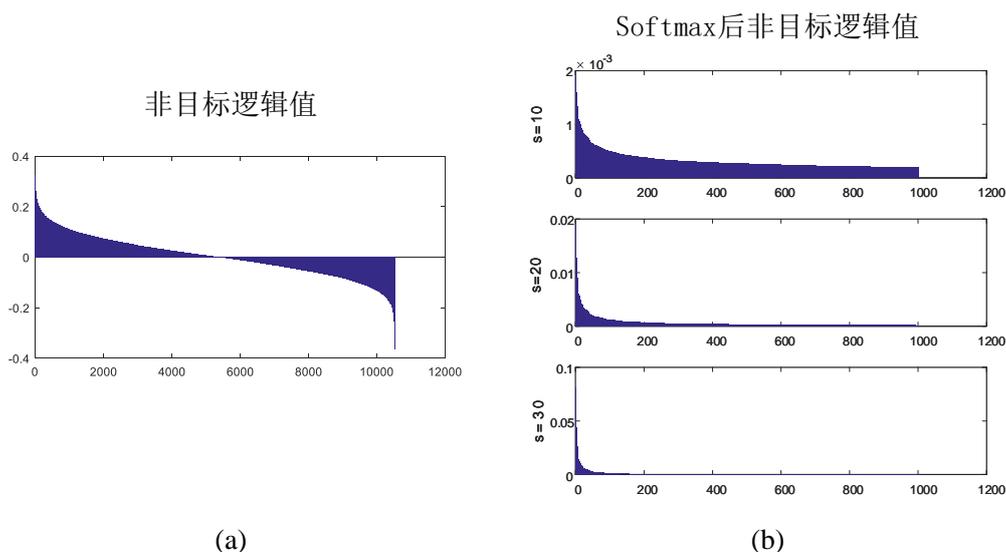


图 3-8 (a) 原始非目标逻辑值由高到低的分布; (b) 不同参数  $s$  下非目标分数在 Softmax 操作后的衰减速度。

收敛速度比 Contrastive Loss 和 Triplet Loss 快很多。而由于在训练末期 Softmax 交叉熵损失也会演变成局部损失，与 Contrastive Loss 和 Triplet Loss 这些专门做验证任务的损失函数类似，所以其验证精度也是有保障的。

### 3.2.3.3 参数 $s$ 的几何意义

第3.1.3节中提到了如果要让 Softmax 交叉熵损失函数适用于人脸认证任务，就需要增加类间的间隔，从而减小其饱和区域的大小，其中一个思路就是让类间间隔处的概率值缓慢增长，这样就不会出现梯度过快饱和的现象。类间间隔处的概率值增长速度可以由公式3-21中的参数  $s$  来控制，如图3-9所示， $s$  越小，则概率（即梯度）增长越缓慢，饱和区域就越小。但是过小的  $s$  会造成没有饱和区域的现象，因此  $s$  也不可以设置得过大，需要加以限制。由于 Softmax 交叉熵损失函数的形式较为复杂，而且饱和区域目前只是一个概念，并无数学定义，因此  $s$  的理论值也难以求得，该项工作将留到下一步工作中继续研究。

为了更加直观地展现不同  $s$  带来的影响，本文在 Fashion-MNIST 数据集<sup>[117]</sup>上做了一个小实验，Fashion-MNIST 数据集与 MNIST 数据集的数据结构一致，但是内容由数字变成了衣服、鞋子、包等图像，原始的 MNIST 在深度卷积神经网络下太容易识别，一般都能达到 99.5% 以上，以至于无法评测损失函数的优劣。而 Fashion-MNIST 在不改变数据规模的前提下，各个模型识别率相比于原始 MNIST 有了很大的降低，从而让我们有足够的空间来进行算法的比对。虽然 Fashion-MNIST 与人脸在数据集层面上具有很大差异，但由于本章所讨论的是损

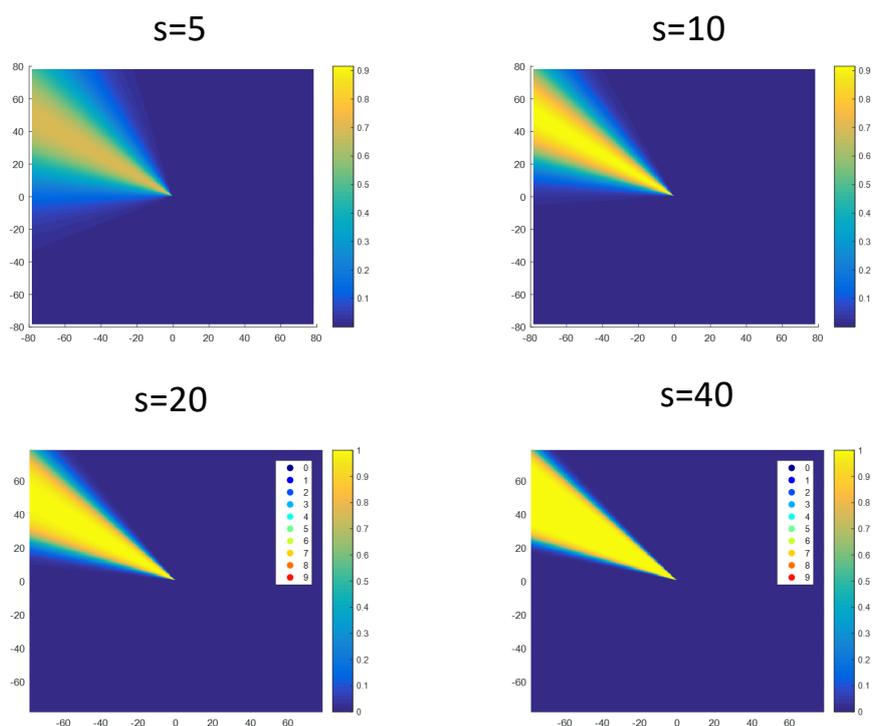
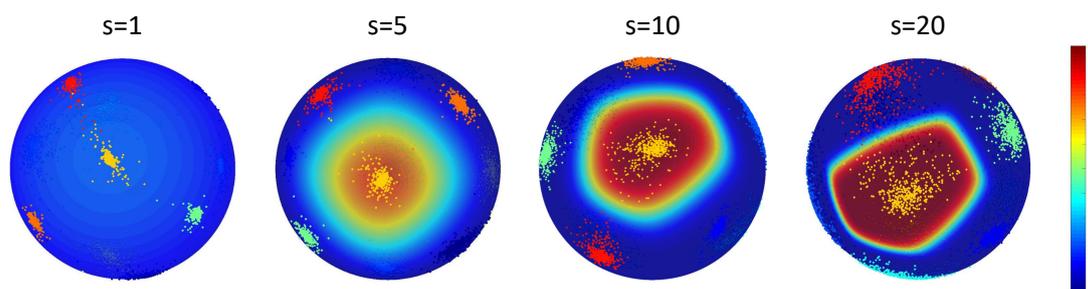


图 3-9 不同参数  $s$  下概率  $P_0$  的增长速度。

失函数，是在已经经过多层卷积神经网络抽取过的特征的基础上进行的操作，与数据源的关系不大，因此使用 Fashion-MNIST 也可以看出特征分布的差别。

由于进行了特征归一化，再使用二维特征会导致特征只有一维的自由度进行活动，这大大提升了模型收敛难度，因此本文将特征维度改为三维，这样在归一化以后特征就有了两个自由度进行运动，从而能更快地进行训练而无需细致的调参。如图3-10所示，本文绘制出了在不同  $s$  下余弦 Softmax 损失得到的特征分布以及第一类的概率  $P_0$ ，从图上可以看到，在  $s = 1$  时，整个球上的概率值都非常低，即使特征已经收敛到类中心，仍然会得到很大的梯度，因此特征就会非常聚拢，随着  $s$  的增大，概率  $P_0$  中逐渐有了饱和区域（红色部分），特征进入饱和区域后就几乎停止了收缩，因此特征分布也逐渐发散。

另一个值得注意的地方是，随着  $s$  的增大，概率  $P_0$  的等高线会渐渐地由圆形变成多边形，这里的原因与小节3.2.3.2里的解释类似， $s$  越大则参与分界面投票的非目标类别越少，大部分分界面上的点近似于仅由目标类别和一个非目标类别决定，因此等高线会表现为折线的形状。

图 3-10 不同参数  $s$  下训练得到的特征分布以及概率  $P_0$ 。

### 3.3 实验结果及分析

本节将对本章提出的一系列损失函数的有效性进行验证。首先小节3.3.1中将会介绍实验设置，之后将在第3.3.2和第3.3.3节分别在两个数据集上验证本文提出的损失函数的性能。本节所用到的代码和模型都已开放下载<sup>①</sup>。

#### 3.3.1 实验设置

##### 3.3.1.1 基线模型

为了验证本文提出的损失函数的通用性，本文选择了两个模型作为基线进行对比。其中一个是一个 10 层的带有 Maxout 激活函数<sup>[118]</sup> 的卷积神经网络 Light CNN<sup>[119]</sup>，另一个是一个 28 层的深度残差网络<sup>[16]</sup>ResNet-28<sup>[63]</sup>。其中 Light CNN 只使用了 Softmax 交叉熵损失函数进行训练，而 ResNet-28 使用了 Softmax 交叉熵损失函数与中心损失函数联合训练，两个模型所用的损失函数都没有进行特征归一化或权重归一化。为了比较的公平性，本文严格地使用了与这两个模型一样的实验设置，包括数据集、图像分辨率、图像对齐方式和评价标准。

##### 3.3.1.2 模型训练

本章所提出的损失函数都是直接追加在特征层之后，也就是倒数第二个内积层之后，特征和权重矩阵的每一行都进行了  $L_2$  归一化使其幅度均为 1，然后通过一个内积层来计算相似度，最后输入损失函数。整个网络是端到端来训练的，为了加速训练过程，本文直接从基线模型上进行了微调，微调时使用的学习率比较小，在 Light CNN 模型上使用了  $1e-4$  的学习率，而在 ResNet-28 模型上使用了  $1e-3$  的学习率，用这些学习率迭代 5,000 次后，再将学习率除以 10 再次迭代 5,000 次，模型的训练使用的是带动量的随机梯度下降法，其中动量值设置为 0.9。

<sup>①</sup> <https://github.com/happynear/NormFace>

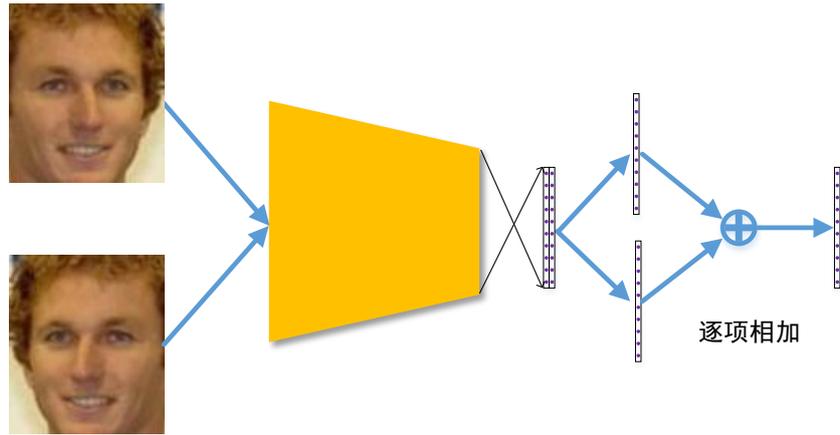


图 3-11 镜像脸示意图。

### 3.3.1.3 模型测试

本章使用了两个数据集来测试本章提出的损失函数，一个是 Labeled Face in the Wild (LFW)<sup>[55]</sup> 另一个是 Youtube Face (YTF)<sup>[91]</sup>，在测试这两个数据集时本文都采用了 10 折测试法。在训练过程的第二个 5,000 次迭代中，每隔 1,000 次迭代就会保存下来一个模型快照，最终的结果是这 5 个模型快照所得结果的平均。

对于每张测试图像，本文提出将对齐好的测试图像和它的镜像同时输入网络，并将得到的两个特征相加作为这张测试图像的最终特征（如图3-11所示）。在 10 折测试法的每 9 个训练集上，本文使用主成分分析法（PCA）来进行学习并对特征进行降维。最后使用余弦相似度来计算两张测试图像之间的相似度，并按照各个数据集的测试标准来计算分数。

### 3.3.2 LFW 数据集

Labeled Face in the Wild (LFW) 数据集<sup>[55]</sup> 包含了从 5,749 个人身上采集到的 13,233 张人脸图像，数据集中的人脸图像涵盖了多个姿态、表情和光照情况，所有的照片均收集自网络。在这个数据集上本文使用了两套不同的协议来测试，第一套协议为标准的无限制额外数据协议<sup>[103]</sup>（又称为 6,000 对协议），该协议将在 6,000 个人脸图像对上进行测试，其中有 3,000 个相同身份样本对和 3,000 个不同身份样本对，这些样本对被均分为 10 组，每次测试选择其中一组作为测试集，其余 9 组作为训练集，反复测试 10 次之后将 10 次的识别率平均即为最终识别率。

第二套协议为 BLUFR 协议<sup>[104]</sup>，这套协议使用了全部的 13,233 张图片，这套协议同时测试了人脸认证和人脸检索两种任务下的性能，对于人脸认证，该协议使用 ROC 曲线下固定 FAR 点时的 TPR 性能作为指标，对于人脸检索，该协议使用 CMC 曲线下固定名次时的召回率作为指标，相关指标的详细描述请参见小

## 节2.2.3。

在第一套不限制额外数据协议<sup>[103]</sup>下本文详细地比对了本章提出的所有的损失函数，结果展示在表格3-2中，从表格上可以看出，本章节中提出的损失函数与基线模型 **Softmax + Center** 损失函数有较为明显的提升，而且本章提出的损失函数并不需要加入中心损失的约束也能取得较好的性能。

表 3-2 LFW6,000 对协议下使用 ResNet-28 网络<sup>[16]</sup> 的准确率

损失函数	是否有归一化	准确率
Softmax	否	98.28%
Softmax + dropout	否	98.35%
Softmax + Center <sup>[63]</sup>	否	99.03%
Softmax	仅特征归一化	98.72%
Softmax	仅权重归一化	98.95%
Softmax	是	99.16% ± 0.025%
Softmax + Center	是	99.17% ± 0.017%

除了本章提出的损失函数，本小节还做了两个消融实验来确定究竟是特征归一化还是权重归一化，在实验过程中可以发现当只做了特征归一化时需要加上参数  $s$ ，而只做权重归一化则不需要，这里的原因暂时不详，有待后续研究。这两个消融实验的结果列在在表格3-2中，从表中可以看到仅仅只是归一化特征时会造成性能下降，而只是归一化权重则对性能没有影响，但也没有提升。这表明特征归一化和权重归一化需要同时使用才能提高识别率，只用一个则没有影响或者会造成识别率下降。

表 3-3 LFW6,000 对协议下使用 Light CNN 网络<sup>[119]</sup> 的准确率

损失函数	是否有归一化	准确率
Softmax	否	98.13%
Softmax + 镜像	否	98.41%
Softmax	是	98.75% ± 0.008%

为了使本章的实验更加有说服力，本文还在 Light CNN 网络<sup>[119]</sup>上进行了多个损失函数的对比实验，结果列在表格3-3中。注意到在文献 [119] 中，作者并没有使用本文提出的镜像脸技巧，所以本文将其使用了镜像脸技巧后的识别率也列在了表格中。从表上可以看出，在使用了归一化技术之后，Light CNN 的性能也有

了显著的提升。

表 3-4 本章算法在 LFW BLUFR 协议<sup>[104]</sup>下的性能

模型	损失函数	是否有归一化	TPR@FAR=0.1%	DIR@FAR=1%
ResNet-28	Softmax + Center <sup>[63]</sup>	否	93.35%	67.86%
ResNet-28	Softmax	是	95.77%	73.92%
Light CNN	Softmax <sup>[119]</sup>	否	89.12%	61.79%
Light CNN	Softmax	是	90.64%	65.22%

在 BLUFR 协议下<sup>[104]</sup>,  $L_2$  超球面嵌入技术的优势更加明显, 这里只比较了一部分损失函数在 ResNet-28 网络<sup>[16]</sup> 和 Light CNN 网络<sup>[119]</sup> 下的性能, 从表格3-4中可以看到, 使用了归一化技术后, 在 0.1% 的误识率 (FAR) 下的召回率 (TPR=Recall) 提升了 1 ~ 2%, 而在 1% 误识率 (FAR) 下的一选正确率 (DIR=Rank 1) 提升了 4 ~ 9%。

### 3.3.3 YTF 数据集

Youtube Face (YTF) 数据集包含了 1,595 个不同身份下的 3,425 个视频序列, 平均每个人有 2.15 个视频, 使用不限制额外数据协议来测试, 该协议使用了 5,000 对视频, 其中 2,500 对视频来自同一个人, 2,500 对视频来自不同人, 每对视频都可以计算得到一个相似度, 最终的识别率就由这 5,000 个相似度计算得来。

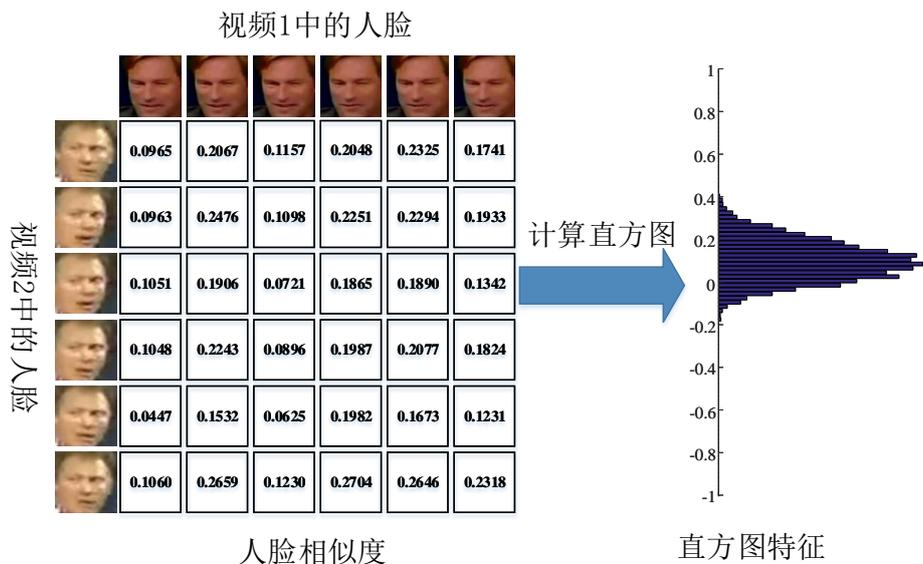


图 3-12 从两组视频中得到相似度直方图特征的示意图。

因为 YTF 数据集均为视频, 所以其计算相似度的方式也与基于单帧图像的人

脸认证不同，具体而言，两个视频的所有帧之间都可以计算得到人脸相似度，这些相似度能够构成一个相似度矩阵  $C$ ，其中的单个元素  $C_{ij}$  表示第一个视频的第  $i$  帧图像与第二个视频的第  $j$  帧图像中人脸的余弦相似度。最终的相似度则从相似度矩阵  $C$  中计算得到。一个较为简单的相似度聚合方式为平均相似度，即直接将  $C$  中的所有图像相似度平均作为两个视频之间的相似度，之后用 5,000 个视频相似度训练一个一维的分类器（例如 SVM）来得到相同/不同人脸的判别阈值。

本文提出使用相似度矩阵  $C$  统计得到的相似度直方图作为特征来训练 SVM 分类器。具体而言，本文将  $C$  中的元素统计成为一个 100 个桶的直方图(如图3-12所示)，之后将 5,000 个直方图输入到使用直方图交叉核的支持向量机（HIK-SVM）中来做二分类。这个方法相比于只有一维的平均相似度而言编码了更多的信息，能稍许提高一些分类的精度。

表 3-5 YTF5,000 协议下使用 ResNet-28 网络<sup>[16]</sup> 的识别率

损失函数	是否有归一化	识别率
Softmax + Center <sup>[63]</sup>	否	93.74%
Softmax	是	94.24%
Softmax + HIK-SVM	是	94.56%

最终的结果列在表格3-5中，从表格中可以看到，在 LFW 上表现较好的损失函数在 YTF 上表现也比较好，而本小节提出的直方图特征技术（表格中标记为 HIK-SVM）也能够较为明显地提升最终的分类性能。

### 3.4 本章小结

本章提出了使用余弦相似度来取代传统模型中的内积相似度的方法，本章使用了一个数学理论解释了为何要用余弦相似度并解决了直接替换为余弦相似度时遇到的问题。本章在损失函数中引入了一个新的超参数：尺度系数  $s$ ，本章解释了为何要加入这个超参数并分析了这个超参数的一系列性质。

本章提出的损失函数在两个数据集、两个基线模型、多个参数指标上都表现出了超越传统方法的性能，而且本章的方法仅仅只需要在原来的模型上进行微调即可显著提升性能，成本非常低廉。

本章还提出了两个非常简单的小技巧：镜像脸和相似度直方图来提升静态和动态人脸认证的识别率，它们实现起来都非常简单，也都能带来比较明显的性能提升。

本章方法的不足在于在基于 **Softmax** 的损失函数的改进上仍然是基于分类的, 没有考虑类内和类间距离之间的关系, 而在基于度量学习的损失函数的改进上没有考虑非目标分数的权重分配, 相信将这两者结合起来将会对识别率有进一步的提升。对于本章中引入的超参数本章中虽然给出了一些分析, 但目前仍旧没有方法能够自动找到最适合的超参数, 这也是一个比较值得研究的方向。

## 第四章 度量学习损失函数的分类化改造

度量学习，或者本文所研究的深度度量学习，通常需要输入一对或者一组样本，在经过一系列的特征变换后输出这些样本之间的距离，使得同样类别的样本之间的距离较小而不同类样本之间的距离较大。相比于基于分类的损失函数，度量学习的损失函数看起来更加适合人脸认证任务，因为人脸认证在测试时也是输入两张图像输出其相似度，这与度量学习的目标函数一致。然而在实际的训练过程中，本文发现度量学习的损失函数普遍存在难调参、难收敛的情况，这其中一部分原因是度量学习损失只关心局部，而丢弃掉了全局监督信号；另一部分原因是训练度量学习损失模型通常需要采样样本对或样本组来训练，而目前人脸识别的数据规模较大，通常达到百万到上亿量级。度量学习损失往往无法采样完全，比如说如果使用 Contrastive 损失<sup>[59,97]</sup>则需要采样  $\mathcal{O}(N^2)$  个样本组合，使用 Triplet 损失<sup>[60,61]</sup>则需要采样  $\mathcal{O}(N^3)$  个样本组合，这在目前的大型数据库下几乎是不可能实现的任务，这样庞大的样本组合内其实也有大量的样本组是已经满足了度量学习要求的，不需要再次进行训练。所以研究者们提出要使用难例挖掘的办法来进行采样，而这一过程往往需要引入一些超参数也需要很多技巧。而基于分类的损失函数只需要不断地将样本输入进网络即可，其样本复杂度只有  $\mathcal{O}(N)$ ，而且一般也不需要难例挖掘，因此本章提出将度量学习的损失函数改造成为分类损失函数，以降低其对样本采样的要求。

本章的主要创新点在于提出了“类代理”的概念，通过给每个类别分配一个类代理，就可以将多样本之间距离度量的学习转换为一个样本对多个类代理之间距离度量的学习，从而避免了样本采样带来的问题。

### 4.1 使用分类损失函数进行度量学习

最常用的深度度量学习方法有 Contrastive 损失<sup>[59,97]</sup>：

$$\mathcal{L}_C = \begin{cases} \|\tilde{\mathbf{f}}_i - \tilde{\mathbf{f}}_j\|_2^2, & c_i = c_j \\ \max(0, m - \|\tilde{\mathbf{f}}_i - \tilde{\mathbf{f}}_j\|_2^2), & c_i \neq c_j \end{cases}, \quad (4-1)$$

和 Triplet 损失<sup>[60,61]</sup>：

$$\mathcal{L}_T = \max(0, m + \|\tilde{\mathbf{f}}_i - \tilde{\mathbf{f}}_j\|_2^2 - \|\tilde{\mathbf{f}}_i - \tilde{\mathbf{f}}_k\|_2^2), \quad c_i = c_j, c_i \neq c_k, \quad (4-2)$$

其中这两个  $m$  都表示类间间隔。这两个方法都是在优化归一化后的两个特征向量的欧氏距离，注意到归一化后的欧氏距离与余弦距离之间有如下关系：

$$\|\tilde{x} - \tilde{y}\|_2^2 = 2 - 2\tilde{x}^T\tilde{y}. \quad (4-3)$$

利用这个公式，在第三章提出的余弦 Softmax 损失函数也可以改写为优化归一化后的欧氏距离的形式：

$$\begin{aligned} \mathcal{L}_{CS} &= -\frac{1}{m} \sum_{i=1}^m \log \frac{e^{s\tilde{W}_i^T\tilde{f}_i}}{\sum_{j=1}^n e^{s\tilde{W}_j^T\tilde{f}_i}} \\ &= -\frac{1}{m} \sum_{i=1}^m \log \frac{e^{-\frac{s}{2}\|\tilde{f}_i - \tilde{W}_i\|_2^2}}{\sum_{j=1}^n e^{-\frac{s}{2}\|\tilde{f}_i - \tilde{W}_j\|_2^2}}, \end{aligned} \quad (4-4)$$

受这个公式启发，我们将公式4-1和公式4-2中欧氏距离的其中一项修改为权重矩阵  $W$  的一列  $W_i$ 。由于权重的意义太过广泛，本文将  $W_i$  称作第  $i$  类的“代理”，意为通过优化一类的代理来代替优化这一类的单个样本，从而避免对这些单个的样本进行采样。与余弦 Softmax 损失类似，这里的  $W$  一样是通过反向传播求出梯度进行优化。在使用了类代理来替代各类样本之后，就可以得到分类版本的 Contrastive 损失：

$$\mathcal{L}_{CC} = \begin{cases} \|\tilde{f}_i - \tilde{W}_j\|_2^2, & c_i = j \\ \max(0, m - \|\tilde{f}_i - \tilde{W}_j\|_2^2), & c_i \neq j \end{cases}, \quad (4-5)$$

和 Triplet 损失：

$$\mathcal{L}_{CT} = \max(0, m + \|\tilde{f}_i - \tilde{W}_j\|_2^2 - \|\tilde{f}_i - \tilde{W}_k\|_2^2), \quad c_i = j, c_i \neq k. \quad (4-6)$$

为了将它们与各自的度量学习版本区分开，我们给这两个损失函数起名叫 C-Contrastive 损失和 C-Triplet 损失，其中字母 C 代表了它们是为了分类而设计的。

实际上，间隔的概念在支持向量机 (SVM) 时代就已经被广泛地分析讨论了，除了传统的线性 SVM 或核函数 SVM，SVM 也有在深度学习上的应用<sup>[120]</sup>，其使用的是合页损失函数 (Hinge Loss) 的多类版：

$$\mathcal{L}_H = \sum_{i \neq y} \max(0, 1 + W_j^T f - W_{y_i}^T f) + \lambda \sum_k \sum_n W_{k,n}^2, \quad (4-7)$$

其中后面一项来控制间隔的大小，这样的处理方式巧妙地将间隔固定为 1，而通过改变  $\|W_i\|^2$  来变相地调整间隔，对于较大的  $\|W_i\|^2$ ，1 的间隔相对就比较小，反之

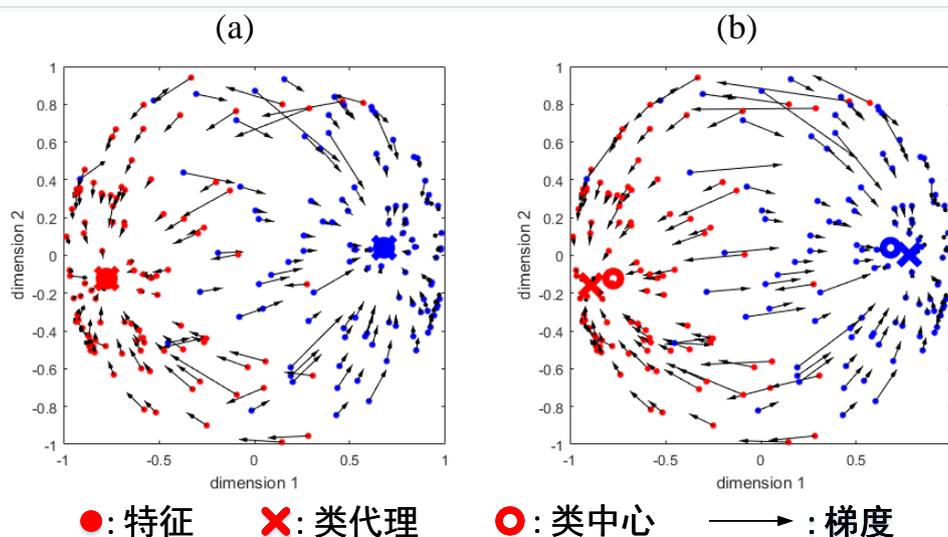


图 4-1 C-Contrastive 损失函数在两类、三维球面情况下，类代理与类中心之间关系的示意图。(a)  $m = 0$  时的情况；(b)  $m = 1$  时的情况。

亦然。然而由于本文提出的算法对特征和权重都进行了归一化，因此这个技巧也不再适用，所以本文将间隔 1 修改为可变的间隔  $m$ ：

$$\mathcal{L}_{Hm} = \sum_{j \neq i} \max(0, m + \mathbf{W}_j^T \mathbf{f} - \mathbf{W}_{y_i}^T \mathbf{f}), \quad (4-8)$$

可以看到这个公式与公式4-6是等价的，所以殊途同归，从 SVM 的角度也可以推导出 C-Triplet 的公式。

需要注意的是，虽然本文改造的这两个损失函数均变为了分类损失函数的形式，但不代表它们是用于分类的。其中最主要的区别在于分类中的间隔是为了模型的泛化能力考虑的（如 SVM），而本文提出的损失函数里的间隔是为增大类间间距而考虑的，与为了模型泛化能力而设置的间隔相比，为增大类间间距而设置的间隔要更大一些。

## 4.2 误差分析

上一节介绍了如何使用类代理来取代度量学习公式里的一部分样本特征，这一节将要对类代理进行一些数学分析。

类代理的作用比较像中心损失中的类中心<sup>[63]</sup>，之所以用“代理”一词而不用类中心是因为当类间的监督信号叠加在  $\mathbf{W}$  上时，各个代理将会偏离类中心，并带领各自类别的样本远离其他类别。如图4-1所示，图上的特征全部为随机生成，而代理和梯度则由 C-Contrastive 损失函数计算得到。其中，图4-1(a)为  $m = 0$  时的特

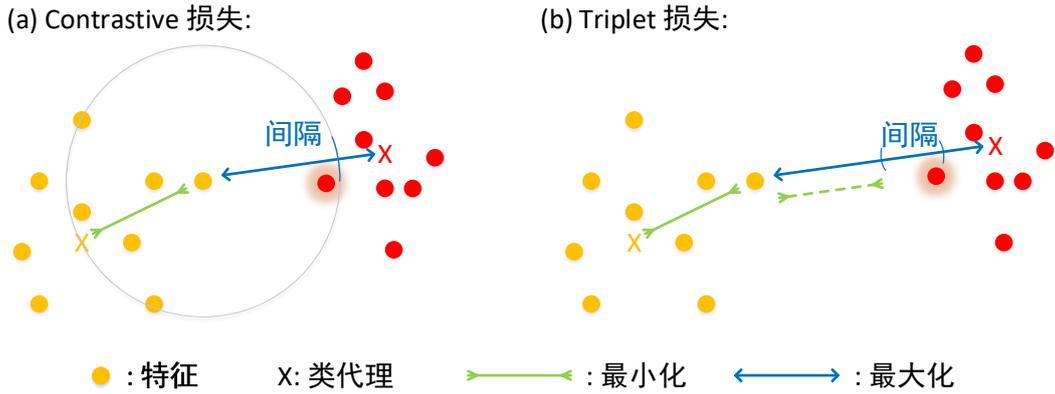


图 4-2 (a) 分类版本的 Contrastive 损失；(b) 分类版本的 Triplet 损失。

例，在这种情况下，代理只接受类内监督信号的作用，最终代理将收敛到其对应类别的类中心，与文献 [63] 类似。而图4-1(b)为正常的情况下的类代理与类中心的关系，在这种情况下，代理同时受到类内和类间监督信号的影响，此时代理将偏离类中心，而样本也将随着代理而偏离另外一类。从图4-1上可以看到，若没有类间信号，则代理的作用于类中心一致，当类间信号加入之后，代理会偏离类中心。

使用代理来替代样本特征的策略会引入一定的误差，如果还用与之前一样的间隔参数  $m$ ，处在边界上的一些点将不会被优化到，如图4-2所示，带有阴影的点即为在代理策略影响下被忽略掉的点，在原版的损失函数中，这些带有阴影的点仍然会被优化到。所以在引入替代策略后，需要加大  $m$  来优化更多的代理。然而从图4-2上可以看到，误差的来源是一个类里的某几个样本没有被优化，因此引入更多的代理并不能从根本上消除误差。

为了能更好地分析误差，本小节对误差进行了定量的分析：

**命题 4.1** 使用类代理来取代样本特征会带来  $\frac{1}{n_{C_i}} \sum_{j \in C_i} (d(\mathbf{f}_0, \mathbf{f}_j) - d(\mathbf{f}_0, \mathbf{W}_i))^2$  的误差，这个误差的上界为  $\frac{1}{n_{C_i}} \sum_{j \in C_i} d(\mathbf{f}_j, \mathbf{W}_i)^2$ 。

**证明：**由于  $d(x, y)$  是一个距离，由距离得三角不等式性质可得，

$$d(\mathbf{f}_0, \mathbf{W}_i) - d(\mathbf{f}_j, \mathbf{W}_i) \leq d(\mathbf{f}_0, \mathbf{f}_j) \leq d(\mathbf{f}_0, \mathbf{W}_i) + d(\mathbf{f}_j, \mathbf{W}_i), \quad (4-9)$$

所以，

$$-d(\mathbf{f}_j, \mathbf{W}_i) \leq d(\mathbf{f}_0, \mathbf{f}_j) - d(\mathbf{f}_0, \mathbf{W}_i) \leq d(\mathbf{f}_j, \mathbf{W}_i), \quad (4-10)$$

$$(d(\mathbf{f}_0, \mathbf{f}_j) - d(\mathbf{f}_0, \mathbf{W}_i))^2 \leq d(\mathbf{f}_j, \mathbf{W}_i)^2. \quad (4-11)$$

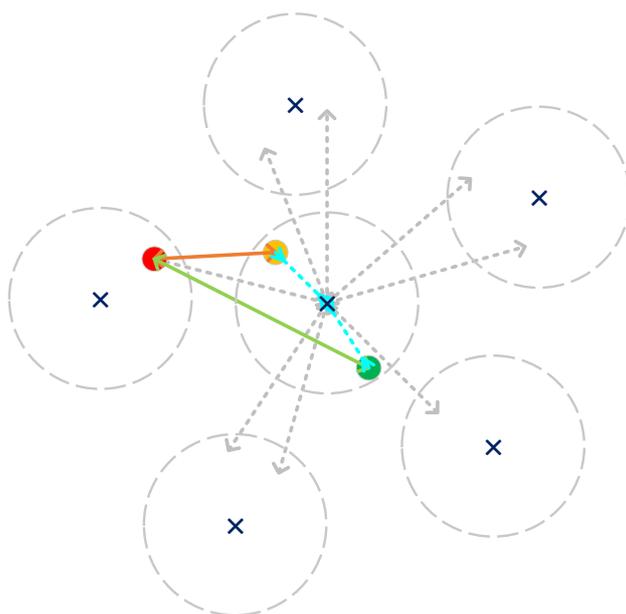


图 4-3 使用类代理 (x) 来取代样本 (圆点) 的示意图。

最终可得，

$$\frac{1}{n_{c_i}} \sum_{j \in C_i} (d(f_0, f_j) - d(f_0, W_i))^2 \leq \frac{1}{n_{c_i}} \sum_{j \in C_i} d(f_j, W_i)^2. \quad (4-12)$$

证明完毕。 ■

这个上界给了我们设定间隔  $m$  的理论依据，在训练的时候可以随时监视这个误差。根据实际经验，这个误差的上界  $\frac{1}{n_{c_i}} \sum_{j \in C_i} d(f_j, W_i)^2$  的大概取值范围为  $0.5 \sim 0.6$ ，在度量学习所使用的间隔参数上需要加上这一误差上界，因此我们推荐对 C-Contrastive 损失和 C-Triplet 损失分别设置 1.0 和 0.8 的间隔。

注意到间隔  $m$  的使用曾经是一个非常复杂的工作<sup>[97]</sup>，根据他们的步骤，需要在训练网络时经常中止训练，并使用块搜索算法来寻找最佳的  $m$ 。在使用了特征归一化技术之后，就不再需要如此繁琐的训练过程了，因为特征的幅度已经定下来，所以间隔  $m$  也可以自始至终使用同一个值。

虽然本节证明了本章提出的使用类代理的方式相比传统度量学习方法来说会带来一定的误差，但这并不意味着其性能会变差。实际上，传统的比对样本对的度量学习方式并不稳定，如图4-3所示，在优化红色点的时候，负样本可能会取到黄色点或绿色点，假设我们设置的间隔是取黄色点时进行类间优化而绿色点处不进行类间优化（Triplet 损失的情况则为绿色点完全不优化），那么如果训练集中没有黄色这个点，则红色点对于黄绿点所在类别的类间距离就不会得到任何优化。但使用改造后的基于分类的损失函数，可以学习到每一类的代理，这些类代理的位

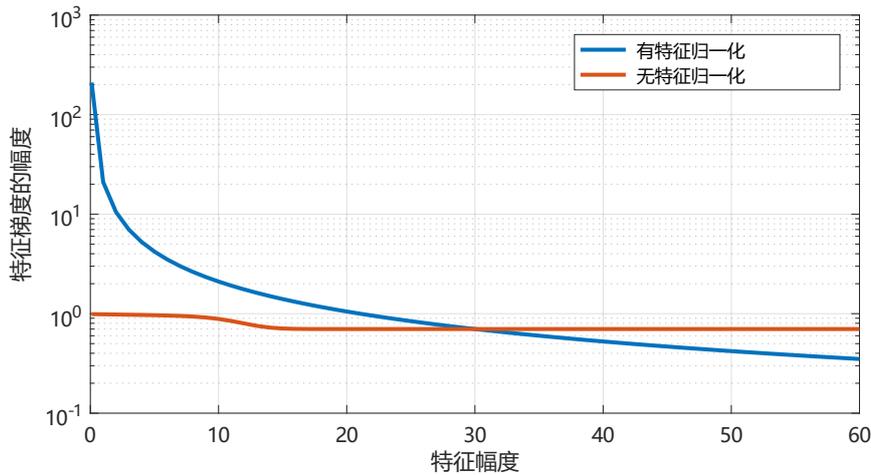


图 4-4 特征梯度的幅度关于特征幅度的函数。

置不仅由类内样本决定，还会在其他类别的样本的排斥作用（灰色虚线箭头）下与类内样本（青色虚线反箭头）共同优化决定，所以由样本到类代理的类间距离可以比样本到样本的类间距离更加稳定，对于样本缺失的情况也具备很好的鲁棒性。

简单来说，传统的度量学习方法学习到的是点到点之间的关系，而改造成的分类损失函数学习到的是点到分布之间的关系，因为这些分布是由大量类间类内样本所共同决定的，所以在样本有缺失的情况下分类损失函数具有更好的鲁棒性。

### 4.3 归一化的作用

在上一章和本章的损失函数中，均对所有的特征和权重进行了归一化操作，但这两章讨论得更多的是归一化的性质。本节将对归一化在人脸认证中所起到的作用进行分析。首先针对特征归一化，因为神经网络反向传播的特性，使得特征归一化操作在针对难例的梯度幅值更高，因此也能起到难例挖掘的作用；其次对于权重的归一化，因为权重的幅度关系到了特征空间的分配，所以将权重进行归一化可以使得空间分配更加均匀，起到了缓解类不均衡问题的作用。

#### 4.3.1 归一化在难例挖掘方面的作用

神经网络的反向传播有一个特殊的性质：

$$y = \frac{x}{\alpha} \Rightarrow \frac{dy}{dx} = \frac{1}{\alpha}. \quad (4-13)$$

也就是说前向传播时除以了某个数字，那反向传播时梯度也会除以这个数字。同样的，对于归一化层来说，一个具有较短幅度的特征在经过归一化层后会除以一

个较小的数字，而在反向传播的时候也会除以这个较小的数字，因此较短幅度的特征反而会得到相对较大的梯度（如图4-4所示）。而根据一些研究者的发现，较短幅度的特征往往代表着比较难以区分的类别，因此归一化层会倾向于优化比较难的样本，它的作用比较类似于图像检测、度量学习里面的难例挖掘。总之，带有归一化层的网络更加适合较低质量人脸图像的认证。

如图4-4所示，可以看到梯度幅度在特征幅度较小的时候会变得非常大，这潜在地增加了梯度爆炸的隐患，尽管在实际应用中我们并不经常会碰到这样的样本，但并不能排除一些噪声的存在。两条曲线的折中可能是一个比较理想的方案，这将会是一个比较值得研究的进一步工作。

### 4.3.2 权重归一化在类不均衡问题上的作用

本文的算法均对权重进行了归一化，除了建立起对余弦相似度的优化之外，权重归一化在均衡各类特征空间上也有很重要的作用。

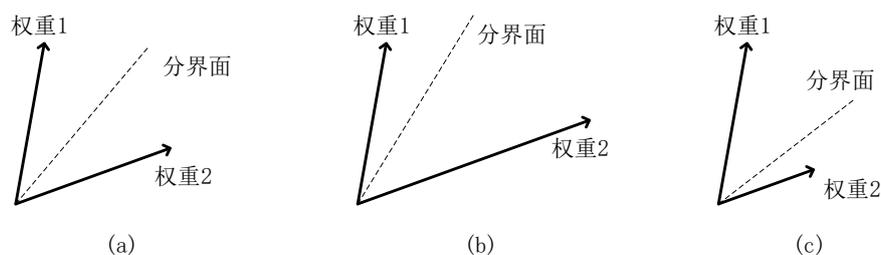


图 4-5 三种权重幅度下的类分界面。(a) 权重 1 等于权重 2；(b) 权重 1 小于权重 2；(c) 权重 1 大于权重 2。

如图4-5所示，本文绘制了三种情况下类分界面的划分情况，在分界面上的特征  $f$  满足  $W_1^T f = W_2^T f$  条件，可以看到，当  $\|W_1\| = \|W_2\|$  时，类分界面恰好是  $W_1$  与  $W_2$  的角平分线，而当  $\|W_1\| > \|W_2\|$  时，类分界面恰好会偏向  $W_2$  的一侧，导致  $W_1$  所在类别占用的空间大于  $W_2$  所占空间。

在分类问题中，不等长的权重幅度可以视作类别先验，比如说在一些类不均衡问题中，样本较多的类别占用的空间较多问题也不大，因为只要训练集与测试集的样本分布一致，且评价指标没有针对数量较少的类别有偏向性，那权重幅度就可以作为一个类别数量先验，样本多的类别占用更多的空间反而能提升测试集上的性能。

但是人脸认证任务是一个开集任务，测试集与训练集甚至没有相同的身份，类别数量先验也就无从谈起。在人脸认证任务中，需要秉持“人人平等”的原则，一个类别的样本多也不可以占用更多的空间。如果不做权重的归一化，类似于特征幅度或是参数  $s$  的优化(小节3.2.3.1)，在训练中后期如果没有权重衰减项的存在，

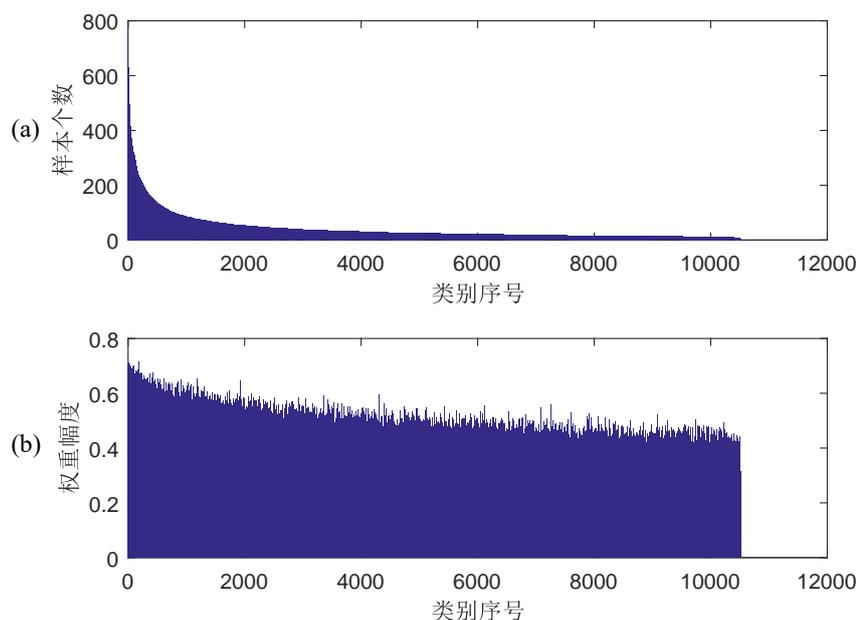


图 4-6 (a) 样本个数的分布；(b) 权重幅度的分布。

权重幅度是只升不降的，这时样本较多的类别对应的权重就会得到更多的训练机会，从而获得更大的权重幅度、占据更多的特征空间。本文使用传统的 Softmax 交叉熵损失函数做了一个实验，发现权重幅度与类别中样本的数量呈一定的正相关关系，如图4-6所示，样本数量较多的类别往往具备更高的权重幅度，这对人脸认证任务来说是有损害的，从这个角度来看也应当在人脸认证任务中对权重进行归一化操作。

## 4.4 实验结果及分析

本节将对本章改造的两个损失函数的有效性进行验证。本节沿用了大部分上一章所使用的实验设置，并在 LFW<sup>[55]</sup>、YTF<sup>[91]</sup>、MegaFace<sup>[102]</sup> 三个数据集上测试本章修改的损失函数的性能。

### 4.4.1 实验设置

在本章使用的基线模型为一个 28 层的深度残差网络<sup>[16]</sup>ResNet-28<sup>[63]</sup><sup>①</sup>，ResNet-28 使用了 Softmax 交叉熵损失函数与中心损失函数联合训练。对于传统的 Contrastive 损失和 Triplet 损失，本文将直接引用 [64] 中的结果，文献 [64] 中使用的模型为 64 层的 ResNet，其模型的表达能力比本章使用的 28 层更强，但在结果

<sup>①</sup> <https://github.com/ydwen/caffe-face>

中我们将会展示出本章提出的损失函数在 28 层的 ResNet 上的性能仍然要高于 64 层的传统度量学习损失函数。

为了加速训练过程，本文将在 Softmax 交叉熵损失函数与中心损失函数联合训练的模型上使用本章设计的损失函数进行微调，并在最后的 5,000 次迭代中每 1,000 次就取一个模型出来计算最终结果的均值作为最终结果。

同上一章一样，本章将使用 MTCNN<sup>[106]</sup> 对人脸照片进行检测和对齐，在测试时同时提取正脸和镜像脸的特征，相加之后作为该张人脸的特征。最后计算特征之间的余弦相似度，并按照各个测试集的评价协议进行测试。

#### 4.4.2 测试结果

在 LFW 不限制额外数据协议<sup>[103]</sup> 和 YTF 数据集<sup>[91]</sup> 上本节详细地比对了本章提出的损失函数，结果展示在表格4-1中。其中 C-Triplet + Center 损失函数是在 C-Triplet 损失函数的基础上，不论  $m + \|x_i - W_j\|_2^2 - \|x_i - W_k\|_2^2$  是不是小于 0，都优化  $\|x_i - W_j\|_2^2$  的一种损失函数。

表 4-1 本章提出的损失函数与相关工作的对比

损失函数	LFW <sup>[103]</sup>	YTF <sup>[91]</sup>
Softmax	98.28%	91.25%
Softmax + dropout	98.35%	91.8%
Softmax + Contrastive <sup>[59]</sup>	98.78%	93.5%
Triplet <sup>[60]</sup>	98.70%	93.4%
Softmax + Center <sup>[63]</sup>	99.03%	93.74%
C-Tontrasitve	99.15%	94.48%
C-Triplet	99.11%	94.64%
C-Triplet + Center	99.13%	94.58%
Softmax + C-Contrastive	99.19%	94.72%

从表格上可以看出，本章节中提出的损失函数与几个基线模型相比在两个数据集上均有较大的提升，最高的识别率 99.2167% 来自于 Softmax + 0.01 \* C-Contrastive 损失函数，其中 0.01 是枚举尝试出来的参数。这种使用两个损失函数组合的缺点就在于需要引入额外的参数来平衡两个不同的损失函数。

如图4-7所示，本文绘制了在 Softmax 与 C-Contrastive 联合训练时损失权重对最终性能的影响，从图中可以看出，C-Contrastive 损失对于损失权重更加鲁棒。这主要是因为 C-Contrastive 损失同时集合了类内监督信号和类间监督信号，即使只用 C-Contrastive 损失也能够将训练进行下去。而中心损失只优化类内方差，必须

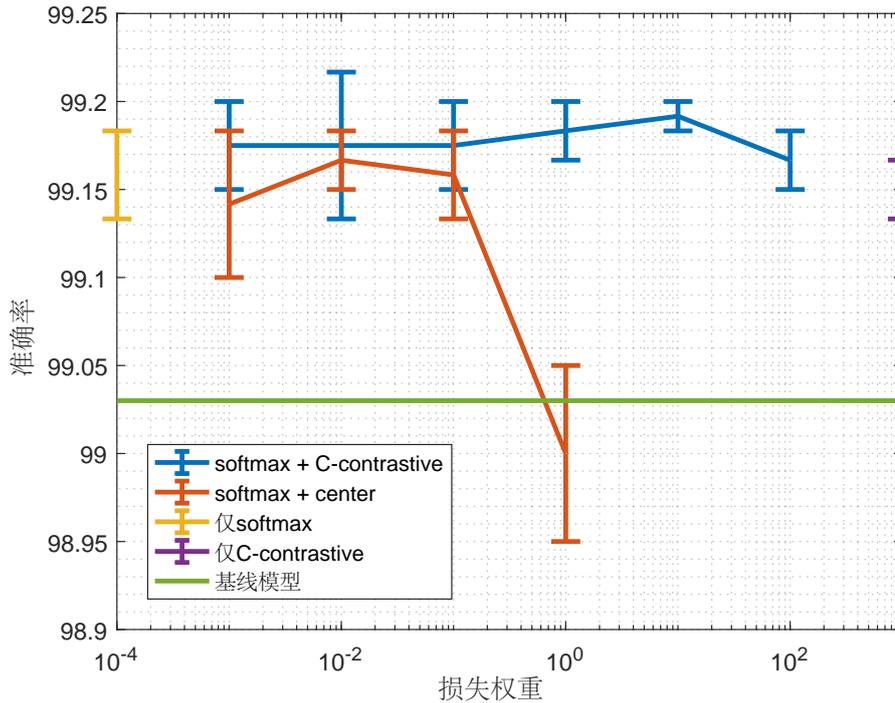


图 4-7 LFW 数据集的识别率与 C-Contrastive 或者中心损失的权重之间的关系。除了基线模型（Baseline）之外的所有方法均使用了归一化技术。

配合其他损失来使用，所以当中心损失权重较高时，网络得到的类间优化信号较少就无法训练出有效的模型了。

在百万级别的测试数据集 MegaFace<sup>[102]</sup> 上，由于其测试需要的计算量比较大，测试时间较长，因此本文仅选择了在 LFW 和 YTF 上效果最好的模型与之前的方法进行比较。测试结果如表4-2所示，本章改造后的损失函数较传统的度量学习损失函数有了明显的性能提升。

## 4.5 本章小结

本章针对传统度量学习难以优化的缺点，提出了“类代理”的概念，使用类代理来替换掉原始样本的特征，成功解决了传统度量学习难采样、难优化的问题。本章还分析了“类代理”的性质以及引入“类代理”时带来的误差，这些分析为设置损失函数中的间隔参数有重要的指导作用。

本章解释了使用类代理的本质是学习到了类别的类内分布，这个分布由类内样本和大量其他类的样本共同决定，因此在缺乏类内样本的情况下仍然能够得到一个比较好的估计。

本章所提出的方法的缺陷在于本章提出的每个类别的类内分布仅仅由类代理

表 4-2 多个损失函数在 MegaFace<sup>[102]</sup> 上的性能

损失函数	Rank1@1e6	VR@FAR=1e-6
Softmax	54.855%	65.925%
Softmax + Contrastive <sup>[59]</sup>	65.219%	78.865%
Triplet <sup>[60]</sup>	64.797%	78.322%
Softmax + Center <sup>[63]</sup>	65.234%	76.516%
Softmax + C-Contrastive	67.172%	79.436%

一个向量作为参数，其表达能力有限，类比于高斯分布的话相当于其协方差矩阵恒为单位矩阵。但是由于本章的损失函数中进行了向量归一化，样本均分布在单位球上，所以这些类内分布实际上是 von Mises-Fisher 分布<sup>[122]</sup>，关于 von Mises-Fisher 分布如何计算其类似高斯分布的协方差矩阵这样的二阶参数的数学方法在学术界的研究并不充分，所以对本章算法的进一步优化还需要建立在补充足够的数学工具的基础上。

本章所提出的使用类代理替代样本还有一个缺点在于类代理的数量与类别数量相同，在超大规模数据库上，类别数量可能会达到百万甚至上亿级别，在这种情况下使用类代理策略不论对显存还是计算量的需求都会显得过于巨大。如何采样部分的类代理来进行优化是一个不错的解决思路，但这又需要引入采样策略，只不过相比于对样本的采样，对类别进行采样的工作量会大大降低。

本章中提出的方法和相关分析并不仅限于人脸认证任务，在其他的度量学习任务上应该也可以有性能提升，比如说行人再识别、图像检索等领域都是非常适合进行尝试的，我们会在未来将本章提出的方法应用在这些任务上来看它们是否会有更好的表现。

## 第五章 引入加性间隔的人脸认证损失函数

第三章讨论了如何让 Softmax 交叉熵损失函数优化余弦相似度而不是内积相似度，其中的理论分析部分和第三章结尾处都提到了这样的修改只是修改了分界面处的斜率，而并没有对分界面本身做出修改。本章将进一步对人脸认证问题的本质进行分析，并提出加入间隔的思路来优化分界面的位置。

首先本章会讨论如何在一个分类模型中加入间隔参数，然后讨论所添加的间隔参数的一些性质以及它与其他算法中的间隔参数之间的异同。本章还将以可视化图形的形式展现一系列算法的目标概率曲面来帮助读者理解这些算法。最后本章将会在多个数据集上对本文提出的加性间隔的效果进行验证。

### 5.1 引入间隔的必要性

如第三章的小节3.1.3所述，Softmax 交叉熵损失函数存在着饱和现象，如图5-1所示，其概率值在边界处会快速上升，很快达到饱和，一旦 Softmax 概率达到饱和后，特征就几乎不会再向类中心移动了，这对于分类来说问题不大，因为分类仅仅是要求寻找到一组分界面将几类分开，但对于特征比对来说，类间距需要进一步拉开才能够有效地区分同类和异类样本。如图5-1(a)所示，虽然  $f_1$  和  $f_2$  之间的内积相似度小于  $f_2$  与  $f_3$  之间的内积相似度，但是  $f_1$  和  $f_2$  才是来自同一个类别的特征，从图上虽然能清晰地看出绿色和天蓝色样本之间有着清晰的分界线，但分界线两侧的特征之间的距离仍旧比较靠近，会存在比类内距离还要小的情况。

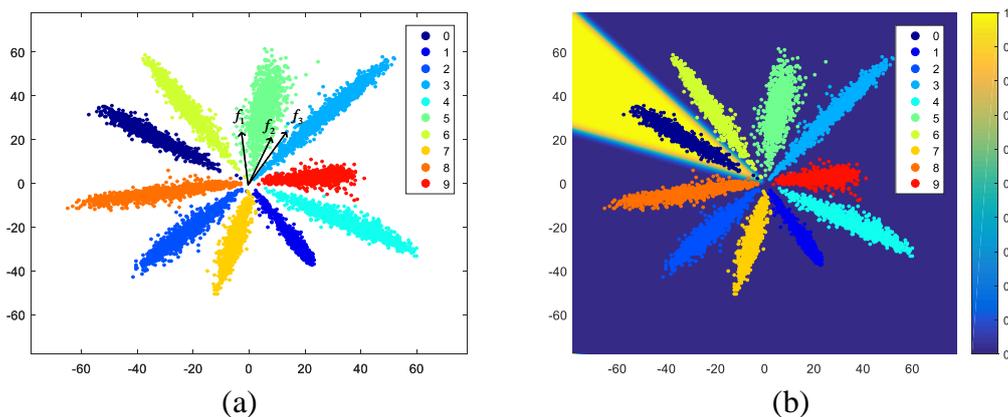


图 5-1 (a) 使用 Softmax 交叉熵损失函数训练得到的二维特征分布；(b) 第 0 类的概率值。

在第三章提出的方案实际上是减缓了分界面处概率上升的速度，通过这个方

式来减小饱和区域，然而注意到多类的 Softmax 交叉熵分类器实质上是多个二分类器的线性组合（参见小节3.2.3.2），第三章提出的余弦 Softmax 损失函数减缓的不仅会减缓当前类别和最近类别之间的概率上升速度，对于远离当前类别的其他类别也有相似的作用，这些类别已经离当前类别和当前特征比较远了，再赋予较缓慢的概率增长速度（本质上是给予了较大梯度）的意义不大，多给出的这些梯度反而会干扰余弦 Softmax 损失函数对当前类别和最近类别之间的分类，这也是参数  $s$  非常小的时候网络性能不佳的根本原因。

本章将会换一种增大间隔的方式，由减缓分界面处概率的增长速度改为直接挪动分界面的位置，让分界面直接划在更加靠近当前类别的位置，本章仍然保留了特征和权重的归一化，这样特征的嵌入空间就被限制在了单位球面上，在限制了空间的情况下，挪动分界面的位置既能减小类内距离，又能增大类间的间隔。这个方案与分界面不动仅修改分界面处概率上升速度相比更加直观，同时也不存在概率上升速度较慢时，较远的类别被赋予过高梯度的问题。

## 5.2 加性间隔

### 5.2.1 定义

类似于上一章提到的 Triplet 损失函数<sup>[60]</sup>，本章所定义的间隔也是施加在余弦相似度上的，对于目标分数  $z_y = \cos\theta_y = \mathbf{W}_y^T \mathbf{f}$ ，直接在上边减去一个间隔  $m$ ，得到：

$$\psi(\theta_y) = \cos\theta_y - m. \quad (5-1)$$

这样一来，本来  $z_y$  只需要稍高于  $\max\{z_j, j \neq y\}$ ，则样本就会被判定为分类正确，现在则需要其大于  $\max\{z_j, j \neq y\} + m$  才被认为分类正确。这相当于分界面向  $\mathbf{W}_y$  的方向推进了  $m$ ，由于分界面是双向的，也可以认为分界面由原来的一个超平面扩展为了一片间隔区域（如图5-2所示）。

在具体的实现中，本文直接将余弦相似度层得到的一系列余弦相似度中的目标分数减掉  $m$ ：

$$\Psi(z_y) = z_y - m. \quad (5-2)$$

在反向传播中，因为  $\Psi'(z_y) = 1$ ，所以无需计算该层的梯度，直接将上层的梯度传递给余弦相似度层即可。

接下来的操作与第三章的公式3-21一样，需要在余弦相似度上乘以一个尺度系数  $s$ ，然后输入进正常的 Softmax 交叉熵损失函数中，最终，损失函数的数学表

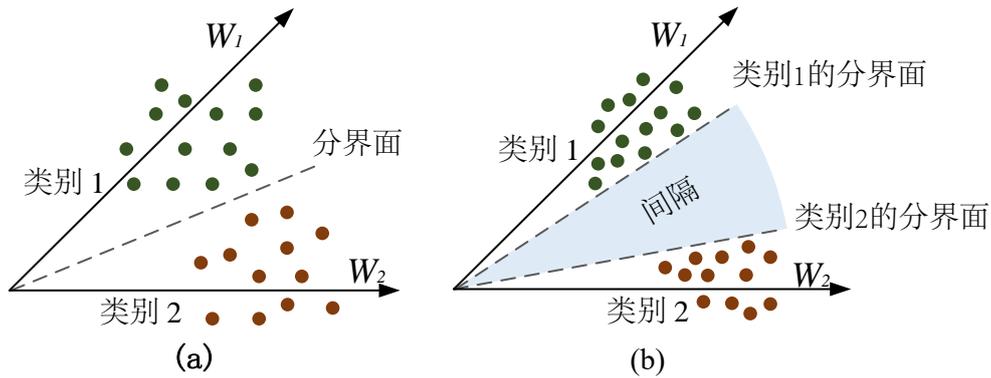


图 5-2 (a) 传统 Softmax 交叉熵损失函数的分界面；(b) 带有加性间隔的 Softmax 交叉熵损失函数的分界面。

达式如下：

$$\begin{aligned}
 \mathcal{L}_{AMS} &= -\frac{1}{n} \sum_{i=1}^n \log \frac{e^{s \cdot (\cos \theta_{y_i} - m)}}{e^{s \cdot (\cos \theta_{y_i} - m)} + \sum_{j=1, j \neq y_i}^c e^{s \cdot \cos \theta_j}} \\
 &= -\frac{1}{n} \sum_{i=1}^n \log \frac{e^{s \cdot (W_{y_i}^T f_i - m)}}{e^{s \cdot (W_{y_i}^T f_i - m)} + \sum_{j=1, j \neq y_i}^c e^{s \cdot W_j^T f_i}}.
 \end{aligned} \tag{5-3}$$

在本章中，除非有特别说明，否则都假定  $W_i$  与  $f$  已经被归一化到 1。第三章的算法设置让  $s$  自动学习，自动学习下的  $s$  会在初期有小幅度的下降后就开始慢慢上升。然而我们发现在引入了新参数  $m$  之后，由推论 3.2 可以得到，如果继续让  $s$  自动学习， $s$  会有更长的一段下降期，我们在实际实验中发现这段下降期经常会持续几千次迭代，在这段迭代过程中  $s$  会达到一个非常小的数值。根据定理 3.1，非常小的  $s$  会导致网络无法收敛，因此不可以继续让  $s$  自动学习。本章直接将  $s$  固定为一个比较大的值，例如对于 10,000 类可以使用 30、对于 100,000 类可以使用 40 作为  $s$  的取值。

在文献 [64, 123] 中，作者们提出使用一种退火的策略来设置超参数  $\lambda$  来避免训练初期网络崩溃，这样的退火策略往往包含了很多参数来定义退火的曲线，如何设置这样的曲线需要大量的实践来体会而没有一个固定的策略，这对新手来说非常不友好。在使用了本章提出的间隔定义之后，就不需要这样的退火策略了， $m$  在训练的过程中始终保持同一个数字，从而减轻了调参的工作量。

如图 5-3 所示，本文绘制了传统 Softmax 交叉熵损失函数，乘性角度间隔<sup>[64]</sup> 和本章的加性余弦间隔在使用最佳参数时所对应的  $\psi(\theta)$  值。图上同时还画出了一个典型的非目标  $\cos(\theta)$  值，非目标  $\cos(\theta)$  与  $\psi(\theta)$  的交点即为分界面所在位置，可以看到，在都取到最佳参数的情况下，使用加性间隔 Softmax 的交点要更小一些，也

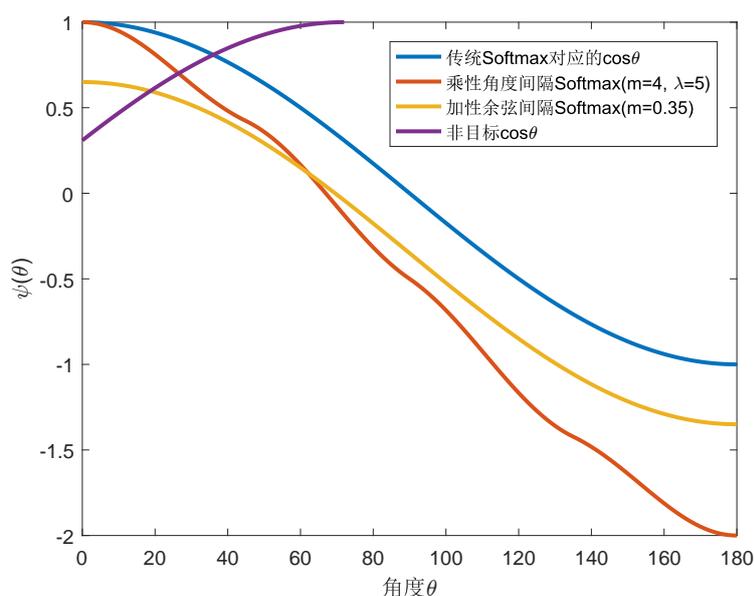


图 5-3 传统 Softmax 交叉熵损失函数，乘性角度间隔<sup>[64]</sup>和本章的加性余弦间隔在使用最佳参数时所对应的  $\psi(\theta)$  值。

就意味着加性间隔能够获得更小的类内方差。

### 5.2.2 角度间隔与余弦间隔

在文献 [64, 105] 中，作者们使用的是角度间隔，分别为乘性角度间隔和加性角度间隔，而本章的间隔是加性余弦间隔。尽管角度和余弦之间是一一对应的，然而在优化的时候两者还是有一些不同之处，余弦值与角度值的密度在小角度与大角度时非常不一样，余弦值在小角度时较为密集，而在大角度时比较疏松，这对于传统的 Softmax 来说并不能影响其分界面的取值，但当分界面向目标代理推动时，推动角度和推动余弦产生的效应会因为它们密度的不同而产生差异。

究竟应该取角度间隔还是余弦间隔应该取决于最终的优化方程，从本章的优化方程公式5-3来看，这个损失函数优化的实际上是余弦值，因此应当使用余弦间隔，如果要使用角度间隔，则需要在余弦上进行一个反余弦操作，而这样的操作不仅增加了计算量，而且因为反余弦操作在角度接近 0 时的梯度趋近于无穷大，会带来潜在的梯度爆炸的风险，因此本文坚持使用余弦间隔作为本章提出的损失函数的最终形式。

### 5.2.3 几何意义

加性间隔有着明确的几何意义，如图5-4所示，本文绘制了传统 Softmax 交叉熵损失函数和带有加性间隔的 Softmax 交叉熵损失函数的分界面，在这张图上，特

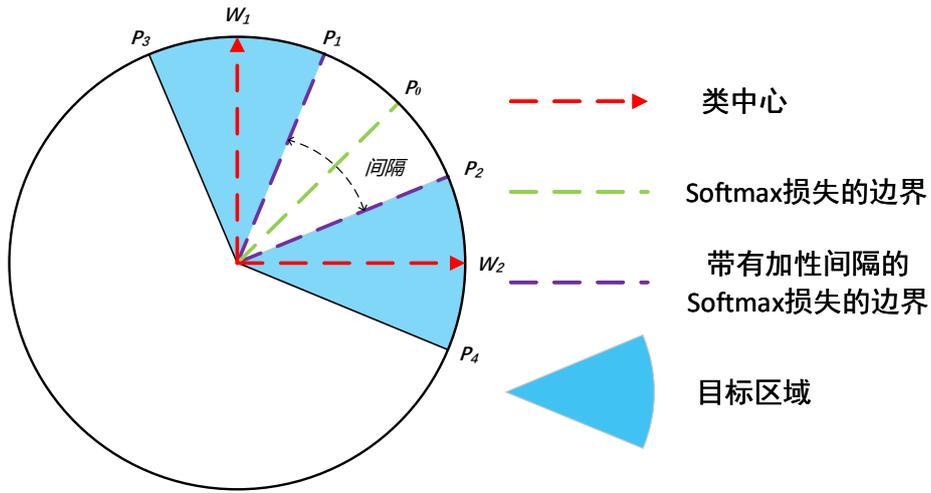


图 5-4 传统 Softmax 交叉熵损失函数的分界面与带有加性间隔的 Softmax 交叉熵损失函数的分界面。

征的维度为 2，在经过归一化后，特征分布在 1 维的圆圈上，传统 Softmax 的分界面即为 0 维的一个点  $P_0$ ，在这种情况下，有  $W_1^T P_0 = W_2^T P_0$ 。

对于本章新提出的带有加性间隔的 Softmax 交叉熵损失函数，类与类的分界不再是超平面而是一个间隔区域，在类别 1 的新分点  $P_1$  处有

$$W_1^T P_1 - m = W_2^T P_1 \Leftrightarrow \cos(\angle_{W_1, P_1}) - m = \cos(\angle_{W_2, P_1}), \quad (5-4)$$

如果进一步假设所有的类别都有相同的类内方差，即可得到

$$\cos(\angle_{W_2, P_1}) = \cos(\angle_{W_1, P_2}), \quad (5-5)$$

联合以上两式可得

$$m = \cos(\angle_{W_1, P_1}) - \cos(\angle_{W_1, P_2}), \quad (5-6)$$

这个公式意味着  $m$  值的含义为分界间隔两侧的余弦值的差。

由该公式可以得到一个关于  $m$  取值的粗略估计。在估计  $m$  之前首先要明确本章的目标，即让最大的类内距离小于等于最小的类间距离，至少在目标函数层面需要满足这个条件。在图 5-4 中，仍然假设各类别都具备同样的类内分布，即两个蓝色的目标区域的角度范围相等，此时需满足：

$$\angle_{P_1, P_2} \geq \angle_{P_1, P_3} = \angle_{P_2, P_4}. \quad (5-7)$$

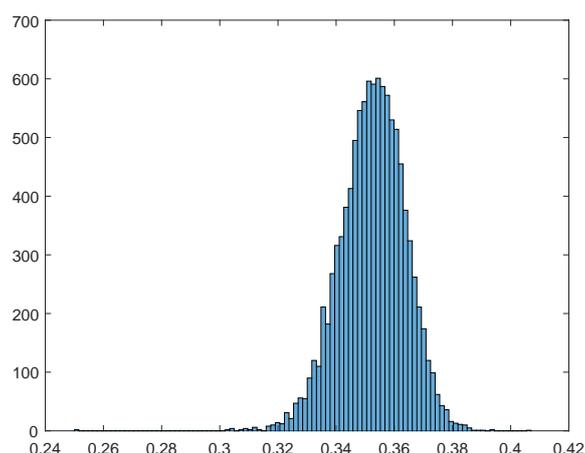


图 5-5 一个典型的人脸认证模型中所有类别计算得到的间隔的下界的分布。

由于  $W_1$ 、 $W_2$  分别是两类的类中心，因此有  $\angle_{P_1, P_3} = 2\angle_{W_1, P_1}$ ，所以：

$$\begin{aligned}\angle_{W_1, P_1} &\leq \frac{1}{4}\angle_{W_1, W_2}, \\ \angle_{W_1, P_2} &\geq \frac{3}{4}\angle_{W_1, W_2}.\end{aligned}\tag{5-8}$$

将上边两个公式代入公式5-6可得：

$$m \geq \cos\left(\frac{1}{4}\angle_{W_1, W_2}\right) - \cos\left(\frac{3}{4}\angle_{W_1, W_2}\right),\tag{5-9}$$

对于多分类问题（假设为  $C$  类）中的每一类，可以得到  $C - 1$  个不同的  $m$  值，因为需要找到一个  $m$  来满足所有的非目标类别，所以我们取这  $C - 1$  个  $m$  值中最小的那一个，对于第  $i$  类有：

$$m_i \geq \cos\left(\frac{1}{4}\min_{j \neq i}\{\angle_{W_i, W_j}\}\right) - \cos\left(\frac{3}{4}\min_{j \neq i}\{\angle_{W_i, W_j}\}\right),\tag{5-10}$$

由这个公式可以得到全部  $C$  个类别对应的间隔的下界。如图5-5所示，本文给出了一个典型的人脸认证模型中所有类别计算得到的间隔的下界的分布， $W$  是从一个训练好的模型中提取出来的，而后由这个矩阵  $W$  算出所有的下界值。从图中可以看到，这些下界的均值约为 0.35，这恰好与本章通过实验得到的最佳  $m$  值比较接近（见第5.4节）。

然而这个估计仍旧是一个比较粗糙的估计，在这个下界的推导过程中除了类内分布相同这个假设之外，本节还引入了两个假设：第一个是使用了最后一个全连接层的权重来表示类中心，但由上章的图4-1所示，这个权重并不是类中心，而是起到了一个“代理”的作用，相临近的两个代理会互相远离对方，因此相临近

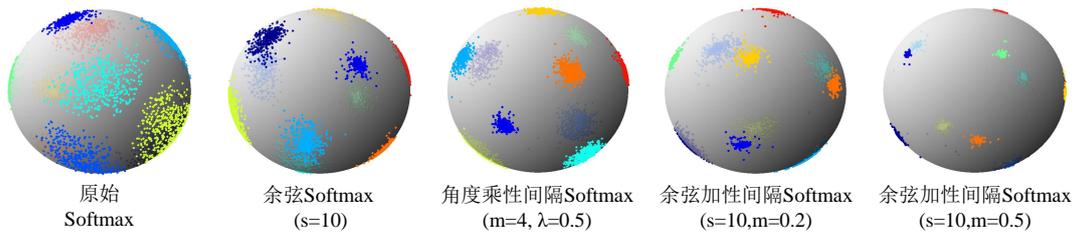


图 5-6 多个损失函数训练出的特征分布图。

的代理之间的夹角要比各自类中心的夹角要大，所以估计出的  $m$  值较理想值偏大；第二个假设是特征与两个类代理在同一个二维平面上，但实际上这三个向量会呈三角关系，在这个假设下估计出的  $m$  值又较理想值偏小。这两个假设一个会导致  $m$  变大一个会导致  $m$  变小，可以起到一个中和的作用，但无论如何，这里所得到的  $m$  确实是较为粗糙的，其更精确的数值仍需后续的研究来推导得到。

### 5.2.4 特征分布可视化

与第三章的小节3.2.3.3类似，本小节也绘制了一系列的特征可视化图像来展示不同损失函数对类内距离的收缩能力，与第三章一样，本小节在 Fashion-MNIST 数据集<sup>[17]</sup>上，使用一个7层的卷积神经网络来学习3维的特征，并将特征归一化后绘制在单位球上。

如图5-6所示，单位球上的每个点都代表了一个归一化后的特征，不同的颜色代表不同的类别。当设置为  $s = 10, m = 0.2$  时，本章提出的损失函数就已经与 A-Softmax<sup>[64]</sup> 的类内方差比较接近了，注意到这时 A-Softmax 已经取到了该数据集上的极限  $m = 4, \lambda = 0.5$ ，更低的  $\lambda$  会导致网络无法收敛，而本章所提出的损失函数的  $m$  还可以继续增大。图5-6上也画出了当  $m$  取值为 0.5 时的特征分布，可以看到在  $m = 0.5$  时特征更加聚拢，这也就意味着本章提出的损失函数对类内方差的收缩能力更强。

### 5.2.5 类空间分割可视化

由于 Softmax 交叉熵损失函数对目标分数的梯度值为  $P_y - 1$ ，而对所有非目标分数的梯度和为  $1 - P_y$ ，则梯度的绝对值之和为： $2|1 - P_y|$ （详情见第三章的小节3.2.3.2），因此只需要观察  $P_y$  的值即可知道该样本能够得到多少梯度。如图5-7所示，本文在3维球面上随机生成了10个点作为10个类代理，并计算球上各个点在多个损失函数下各类概率的最大值，这样就可以观察到各个类别所占有的饱和区域的形状和大小。图中余弦加性间隔 Softmax 的  $m$  值是由小节5.2.3中提出的算法计算得到的。在第三行上本文选择了两组角度乘性间隔 Softmax 的参数，第一

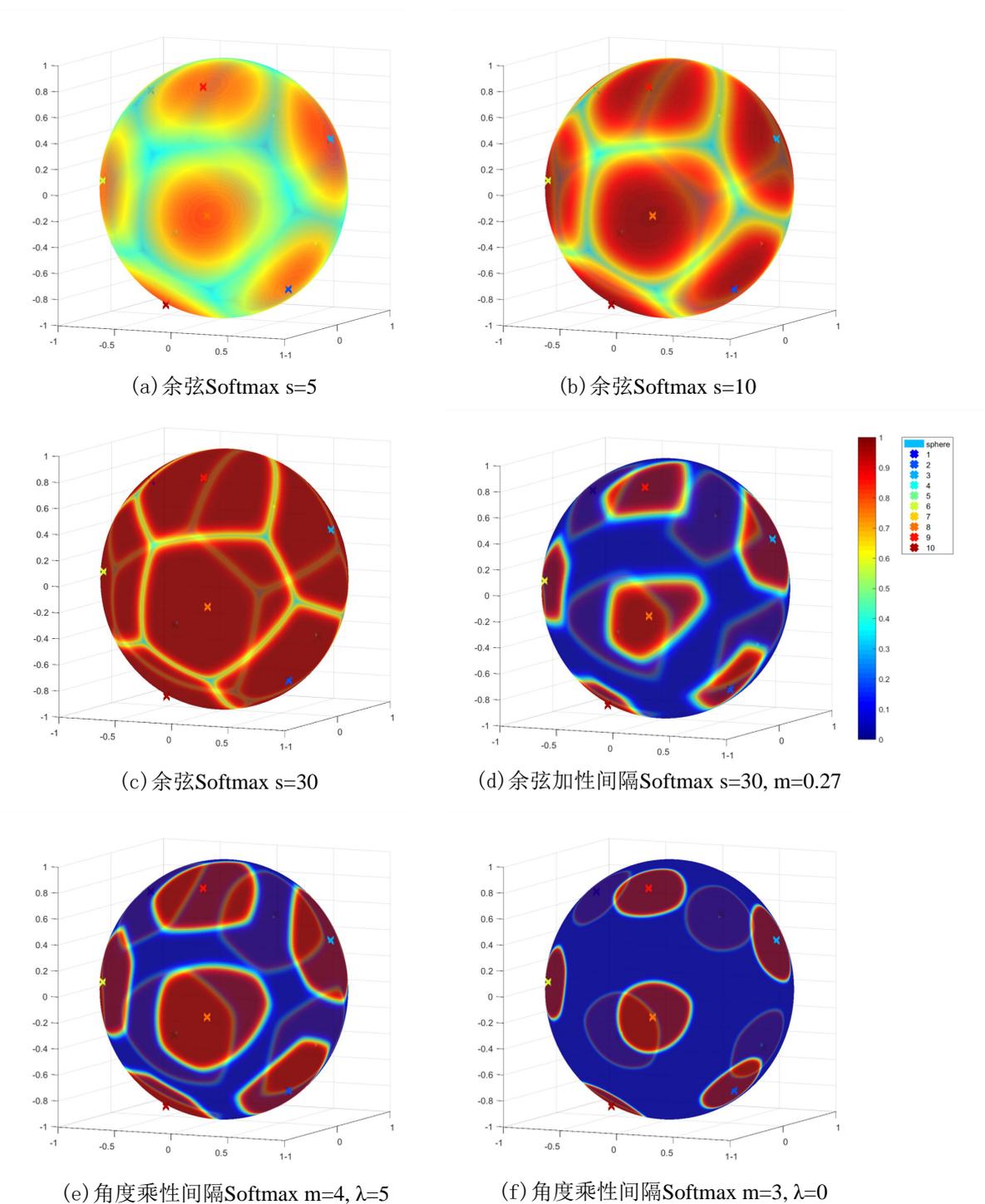


图 5-7 各损失函数在 3 维球面上各点的最大概率值示意图。

组  $m = 4, \lambda = 5$  为文献 [64] 中实际使用的参数，第二组  $m = 3, \lambda = 0$  为文献 [64] 中提出的理论参数。

从图中可以总结出如下特性：

(1) 余弦 Softmax 损失在  $s$  非常小的时候也具备较小的饱和区域，但其饱和区域与间隔区域的概率值差距不大，不具有明显的界限。

(2) 余弦 Softmax 损失在  $s$  较大时几乎不具备间隔，不能起到收缩类内距离的作用。

(3) 本章提出的余弦加性间隔 Softmax 损失在使用小节 5.2.3 中提出的算法计算得到的  $m$  时，有着充足的间隔区域，且饱和区域与间隔区域的概率值差别明显。

(4) 角度乘性间隔 Softmax 损失<sup>[64]</sup> 在其理论最佳参数  $m = 3, \lambda = 0$  下的间隔区域非常大，留给各类的饱和区域较小；而实际使用的参数  $m = 4, \lambda = 5$  下的饱和区域又过大，类与类之间没有充足的间隔。这个特性仅仅是由实验观测得来，究竟为什么角度乘性间隔 Softmax 损失<sup>[64]</sup> 在间隔较大时会更早地出现模型退化效应还没有理论上的解释，这项工作将留待后续研究来解决。

### 5.3 理论分析与讨论

本节将对本章提出的损失函数进行一系列的理论分析。首先本节分析了加性间隔与三元组损失函数和支持向量机中间隔参数的异同；之后本节进一步分析了 Softmax 交叉熵损失函数在神经网络中起到的作用；最后本节提出了“隐式间隔”的概念，隐式间隔辅助研究者在训练过程中对网络的判别能力进行观察，以方便及时地进行参数调整。

#### 5.3.1 与三元组损失函数的联系

本章提出的损失函数 (仅第  $i$  个样本)：

$$\mathcal{L}_{AMS} = -\log \frac{e^{s \cdot (\mathbf{W}_{y_i}^T \mathbf{f}_i - m)}}{e^{s \cdot (\mathbf{W}_{y_i}^T \mathbf{f}_i - m)} + \sum_{j=1, j \neq y_i}^c e^{s \mathbf{W}_j^T \mathbf{f}_i}}. \quad (5-11)$$

与三元组 (Triplet) 损失函数：

$$\mathcal{L}_{\mathcal{T}} = \max(0, m + \|\tilde{\mathbf{f}}_i - \tilde{\mathbf{f}}_j\|_2^2 - \|\tilde{\mathbf{f}}_i - \tilde{\mathbf{f}}_k\|_2^2), \quad c_i = c_j, c_i \neq c_k, \quad (5-12)$$

以及上一章修改过的分类版三元组损失函数：

$$\mathcal{L}_{C\mathcal{T}} = \max(0, m + \|\tilde{\mathbf{f}}_i - \tilde{\mathbf{W}}_j\|_2^2 - \|\tilde{\mathbf{f}}_i - \tilde{\mathbf{W}}_k\|_2^2), \quad c_i = j, c_i \neq k. \quad (5-13)$$

中都包含了间隔项  $m$ ，由于

$$\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|_2^2 = 2 - 2\tilde{\mathbf{x}}^T\tilde{\mathbf{y}}. \quad (5-14)$$

所以

$$\mathcal{L}_{CT} = 2 \max(0, m + \tilde{\mathbf{W}}_k^T \tilde{\mathbf{f}} - \tilde{\mathbf{W}}_j^T \tilde{\mathbf{f}}), \quad c_i = j, c_i \neq k. \quad (5-15)$$

在损失函数前乘以一个常数对损失函数的优化目标没有影响，由此可见，这几个损失函数中的间隔都是加性的余弦间隔，它们在本质上是相通的，本小节将会分析这些间隔形式与本章提出的加性间隔之间的关系。

为了要比较这两类加性间隔，首先要将其化成类似的形式，这里对本章提出的加性间隔 Softmax 损失进行一些变形操作（为了简略起见，默认所有向量都进行了归一化）：

$$\begin{aligned} \mathcal{L}_{AMS} &= -\log \frac{e^{s \cdot (\mathbf{W}_{y_i}^T \mathbf{f}_i - m)}}{e^{s \cdot (\mathbf{W}_{y_i}^T \mathbf{f}_i - m)} + \sum_{j=1, j \neq y_i}^c e^{s \mathbf{W}_j^T \mathbf{f}_i}} \\ &= \log \left( 1 + \frac{\sum_{j=1, j \neq y_i}^c e^{s \mathbf{W}_j^T \mathbf{f}_i}}{e^{s \cdot (\mathbf{W}_{y_i}^T \mathbf{f}_i - m)}} \right) \\ &= \log \left( 1 + \frac{e^{\log \sum_{j=1, j \neq y_i}^c e^{s \mathbf{W}_j^T \mathbf{f}_i}}}{e^{s \cdot (\mathbf{W}_{y_i}^T \mathbf{f}_i - m)}} \right) \\ &= \log \left( 1 + e^{s(LSE(\mathbf{f}_i; s) - \mathbf{W}_{y_i}^T \mathbf{f}_i + m)} \right) \\ &= s * \text{SoftPlus} \left( LSE(\mathbf{f}_i; s) - \mathbf{W}_{y_i}^T \mathbf{f}_i + m; s \right), \end{aligned} \quad (5-16)$$

其中，

$$\begin{aligned} LSE(\mathbf{f}; s) &= \frac{1}{s} \log \left( \sum_{j=1, j \neq y}^c e^{s \mathbf{W}_j^T \mathbf{f}} \right), \\ \text{Softplus}(x; s) &= \frac{1}{s} \log(1 + e^{sx}), \end{aligned} \quad (5-17)$$

第一个函数叫做 LSE 函数<sup>①</sup>，它被用来近似最大的非目标分数。它具有如下性质：

**性质 5.1** LSE 函数满足： $LSE(W, \mathbf{f}; s) > \max\{\mathbf{W}_j^T \mathbf{f}, j \neq y\}$ ，且当最大的非目标分数不等于第二大的非目标分数时有： $\lim_{s \rightarrow +\infty} LSE(W, \mathbf{f}; s) = \max\{\mathbf{W}_j^T \mathbf{f}, j \neq y\}$ 。

<sup>①</sup> 全称 LogSumExp 函数，原始版定义：<https://en.wikipedia.org/wiki/LogSumExp>，LSE 函数经常被用来作为最大值的近似，本文中引入了  $s$  参数来调节近似的幅度。

证明: 根据 LSE 函数的定义有:

$$\begin{aligned}
 LSE(f; s) &= \frac{1}{s} \log\left(\sum_{j=1, j \neq y}^c e^{sW_j^T f}\right), \\
 &> \frac{1}{s} \log\left(\max\{e^{sW_j^T f}, j \neq y\}\right) \\
 &= \max\left\{\frac{1}{s} \log\left(e^{sW_j^T f}\right), j \neq y\right\} \\
 &= \max\{W_j^T f, j \neq y\}.
 \end{aligned} \tag{5-18}$$

为方便书写, 设  $Q = \max\{W_j^T f, j \neq y\}$ ,  $q = \arg \max_j\{W_j^T f, j \neq y\}$ , 则当最大的非目标分数不等于第二大的非目标分数时有:  $\forall j \neq y, q, W_j^T f - Q < 0$ 。

根据此式继续推导:

$$\begin{aligned}
 \lim_{s \rightarrow +\infty} LSE(f; s) &= \lim_{s \rightarrow +\infty} \frac{1}{s} \log\left(\sum_{j=1, j \neq y}^c e^{sW_j^T f}\right), \\
 &= \lim_{s \rightarrow +\infty} \frac{1}{s} \log\left(\sum_{j=1, j \neq y}^c e^{s(W_j^T f - Q)} * e^{sQ}\right) \\
 &= \lim_{s \rightarrow +\infty} \frac{1}{s} \log\left(\sum_{j=1, j \neq y}^c e^{s(W_j^T f - Q)}\right) + Q \\
 &= \lim_{s \rightarrow +\infty} \frac{1}{s} \log(e^0) + Q \\
 &= Q.
 \end{aligned} \tag{5-19}$$

证明完毕。 ■

有了 LSE 函数的帮助, 可以将多分类的 Softmax 交叉熵损失函数看作是一个“二分类”的损失函数, 只不过原本的  $C - 1$  个非目标分数被统一到一个函数中了。通过这样的变换可以更好地分析 Softmax 交叉熵损失函数的性质。

第二个函数是 Softplus 函数, 第三章的定理3.1的证明中已经使用过它的一个变种, 这里本文引入尺度系数得到一个新的变种  $\log(1 + e^{sx})$ , Softplus 函数经常被用来当作 ReLU 函数:

$$ReLU(x) = \max(x, 0), \tag{5-20}$$

的近似。它们之间有如下关系:

性质 5.2 带有尺度系数的 Softplus 函数  $\frac{1}{s} \log(1 + e^{sx})$  总是大于 ReLU 函数  $\max(x, 0)$ , 且  $\lim_{s \rightarrow +\infty} \frac{1}{s} \log(1 + e^{sx}) = \max(x, 0)$ 。

证明: 由于 ReLU 函数  $\max(x, 0)$  是分段函数, 因此本证明也分段进行讨论:

对于  $x < 0$ :  $\frac{1}{s} \log(1 + e^{sx}) > \frac{1}{s} \log(1) = 0$ ;

对于  $x \geq 0$ :  $\frac{1}{s} \log(1 + e^{sx}) > \frac{1}{s} \log(e^{sx}) = x$ 。

对于  $x < 0$ :  $\lim_{s \rightarrow +\infty} \frac{1}{s} \log(1 + e^{sx}) = \lim_{s \rightarrow +\infty} \frac{1}{s} \log(1) = 0$ ;

对于  $x \geq 0$  有:

$$\begin{aligned} \lim_{s \rightarrow +\infty} \frac{1}{s} \log(1 + e^{sx}) - x &= \lim_{s \rightarrow +\infty} \frac{1}{s} \log\left(\frac{1 + e^{sx}}{e^{sx}}\right) \\ &= \lim_{s \rightarrow +\infty} \frac{1}{s} \log(e^{-sx} + 1) \\ &= \lim_{s \rightarrow +\infty} \frac{1}{s} \log(1) \\ &= 0. \end{aligned} \quad (5-21)$$

因此在  $x \geq 0$  时有:  $\lim_{s \rightarrow +\infty} \frac{1}{s} \log(1 + e^{sx}) = x$ 。

证明完毕。 ■

将  $\frac{1}{s} \log(1 + e^{sx}) > \max(x, 0)$  代入公式5-16中可得:

$$\mathcal{L}_{AMS} > s * \max(LSE(\mathbf{f}_i; s) - \mathbf{W}_{y_i}^T \mathbf{f}_i + m, 0). \quad (5-22)$$

对比公式5-22与公式5-13可以发现除了尺度因子  $s$ ，它们的形式非常类似，而尺度因子在求最小值时对最优化的参数是没有影响的，可以直接抹去，因此这个公式表明了加性间隔 Softmax 损失函数实际上是在优化分类三元组损失函数的一个上界。这两个损失函数的区别在于在加性间隔 Softmax 损失中会使用 LSE 函数来代替  $C - 1$  个非目标分数，而用于分类的三元组损失函数通过间隔  $m$  筛选了一部分非目标分数来进行优化，注意到 LSE 函数对非目标分数的梯度为:

$$\frac{\partial LSE(W, \mathbf{f}; s)}{\partial \mathbf{W}_j^T \mathbf{f}} = \frac{\mathbf{W}_j^T \mathbf{f}}{\sum_{j=1, j \neq y}^c e^{s \mathbf{W}_j^T \mathbf{f}}}, \quad (5-23)$$

即其梯度是由 Softmax 函数来进行分配的，而上一章提出的用于分类的三元组损失函数是使用了一个阶跃函数来分配梯度：当  $m + \|\tilde{\mathbf{f}}_i - \tilde{\mathbf{W}}_j\|_2^2 - \|\tilde{\mathbf{f}}_i - \tilde{\mathbf{W}}_k\|_2^2 > 0$  时给予 1 的梯度，否则就给 0 的梯度。使用 Softmax 函数进行梯度分配更加平缓容易优化，而且由于 Softmax 函数分配的梯度总和为 1，也不会存在非目标分数获得过多梯度的问题，在训练前期可以避免梯度爆炸。

### 5.3.2 从最优化的角度理解 Softmax 交叉熵损失函数

在大部分文献中，Softmax 交叉熵损失函数都是从概率角度来解释的。本小节将从另一个角度来推导出 Softmax 交叉熵损失函数，推导过程中利用到了多个上一小节所提到的函数，将这两个小节结合来看将会加深读者对一系列损失函数的

理解。

对于一个基于神经网络的多分类问题（假设为  $C$  类），普遍的做法是让原始信号或者经过手工加工的特征输入进神经网络中，由神经网络输出  $C$  个分数  $z_i, i \in [1, C]$ 。在测试时，人们通常取最大分数所对应的标签作为当前样本的类别，因此目标函数应当满足目标分数  $z_y$  大于所有非目标分数：

$$\begin{aligned} \min \mathcal{L}_{target} &= \min \sum_{i=1, i \neq y}^C \max(z_i - z_y, 0) \\ &= \min \sum_{i=1, i \neq y}^C \text{ReLU}(z_i - z_y), \end{aligned} \quad (5-24)$$

后续表述中将省略  $\min$  符号，所有目标函数与损失函数均默认要被求最小值。如果直接使用目标函数来作为损失函数，则目标分数刚刚大于最大的非目标分数即停止优化，这样得到的模型的泛化能力较差，因此在支持向量机和上一章提出的基于分类的 Triplet 损失函数中都引入了间隔项：

$$\mathcal{L}_{hinge} = \sum_{i=1, i \neq y}^C \text{ReLU}(z_i - z_y + m), \quad (5-25)$$

引入间隔项后，我们对目标分数有了更高的要求，即要大于非目标分数加上间隔值才可以，这样即使测试样本在目标分数上有一定扰动，仍然能够满足目标分数最大的要求。但是这个方法有两个缺点：首先是当类别数  $C$  特别大时，会有大量的非目标分数得到优化，这样每次优化时的梯度幅度不等且非常巨大，极易梯度爆炸；其次在训练末期的梯度又容易取到 0，此时网络会由于规则项（通常是权重衰减）的存在而逐渐退化，因此 SVM 中使用的折页（Hinge）损失函数在神经网络时代并没有得到大规模的使用。

而 Softmax 交叉熵损失函数用的是另外一个思路来近似上面的优化目标函数，注意到“ $z_y$  大于所有非目标分数”等价于“ $z_y$  大于最大的非目标分数”：

$$\mathcal{L}_{target} = \text{ReLU}(\max_{i \neq y} \{z_i\} - z_y), \quad (5-26)$$

这样每次最多就只有一个目标分数和一个非目标分数得到优化，梯度的幅度得到了限制，但这种解决方案因为最多只能优化一个非目标分数，所以收敛速度也会变得很慢。一个解决方案是将中间的最大值函数  $\max_{i \neq y} \{z_i\}$  替换为柔化的最大值函数替换为柔化的最大值函数 LogSumExp：

$$\mathcal{L}_{lse} = \text{ReLU} \left( \log \left( \sum_{i=1, i \neq y}^C e^{z_i} \right) - z_y \right), \quad (5-27)$$

这个函数在小节5.3.1中提到过，它经常被用来近似最大值函数<sup>①</sup>，替换之后，由于：

$$\frac{\partial \log \left( \sum_{i=1, i \neq y}^C e^{z_i} \right)}{\partial z_j} = \frac{z_j}{\sum_{i=1, i \neq y}^C e^{z_i}}, \quad (5-28)$$

所以给予非目标分数的 1 的梯度将会由 Softmax 函数分配到各个非目标分数上去，且非目标分数得到的梯度之和为 1，此时目标分数也将得到-1 的梯度，目标分数与非目标分数得到梯度的总和为 0，绝对值之和为 2，梯度的幅度因此而有了限制。

类似于性质5.1，LogSumExp 函数满足以下性质：

$$\log \left( \sum_{i=1, i \neq y}^C e^{z_i} \right) \geq \max_{i \neq y} \{z_i\}, \quad (5-29)$$

因此 LogSumExp 函数实际上是在  $\max_{i \neq y} \{z_i\}$  的基础上引入了一定的间隔。根据大间隔理论<sup>[124]</sup>，间隔的引入可以提升模型的泛化性能，因此 Softmax 交叉熵损失函数在使用 LogSumExp 函数柔化的同时还引入了一定的间隔来保证模型的泛化能力。

在使用 LogSumExp 函数替换 max 函数之后，训练初期的梯度爆炸情况得到了极大的缓解，但是之前提到的第二个缺点，即训练末期损失函数的梯度为 0，网络优化完全取决于规则项的问题仍然存在，所以还需要进行进一步的改造。这里使用 Softplus 函数  $\log(1 + e^x)$  来近似 ReLU 函数  $\max(x, 0)$ ，得到：

$$\begin{aligned} \mathcal{L}_{softmax} &= \log \left( 1 + e^{\log \left( \sum_{i=1, i \neq y}^C e^{z_i} \right) - z_y} \right) \\ &= \log \left( 1 + \frac{\sum_{i=1, i \neq y}^C e^{z_i}}{e^{z_y}} \right) \\ &= \log \frac{\sum_{i=1}^C e^{z_i}}{e^{z_y}} \\ &= -\log \frac{e^{z_y}}{\sum_{i=1}^C e^{z_i}}. \end{aligned} \quad (5-30)$$

这即为标准的 Softmax 交叉熵损失函数的定义，因为 Softplus 函数在 x 较小时仍然有值，因此不会产生梯度消失的现象。

至此本文用两个近似函数从代数角度而不是概率角度推导出了 Softmax 交叉熵损失函数，相信利用这个全新的角度研究者们将得到更多的新方法和新思路。

<sup>①</sup> 这里有一个经典的歧义，柔性的最大值函数实际上应该 LogSumExp 函数。而其直译的 Softmax 函数得到的并不是柔化的最大值，而是 One-hot 向量（最大值为 1，其他位置为 0 的向量）的柔化。

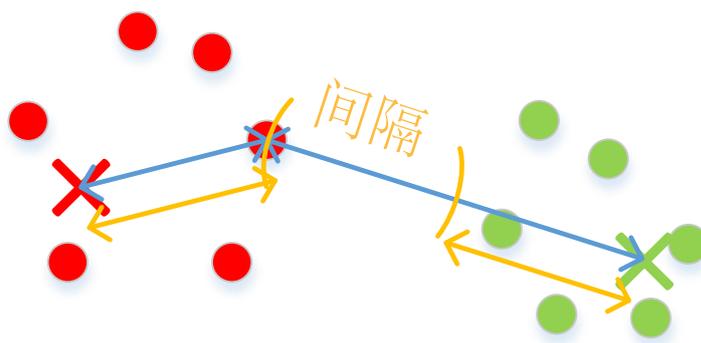


图 5-8 二类问题中单个样本的隐式间隔的示意图。

### 5.3.3 模型判别能力指标

前文讨论了引入间隔的意义，本小节将分析一个训练好的模型中究竟包含了多大的间隔。

首先考虑一个二类问题，在二类问题中单个样本的余弦 Softmax 损失函数的定义为：

$$\mathcal{L}_{CS2} = -\log \frac{e^{s \cdot W_1^T f}}{e^{s \cdot W_1^T f} + e^{s \cdot W_2^T f}} \quad (5-31)$$

这里假设这个样本的标签为 1，经过损失函数的优化， $W_1^T f$  会被不断推高而  $W_2^T f$  会被不断拉低，由此在  $W_1^T f$  与  $W_2^T f$  之间建立了一个“隐式间隔”（如图 5-8 所示）：

$$m_{latent} = W_1^T f - W_2^T f. \quad (5-32)$$

在训练时不断地采集这些间隔值，然后计算出它们所构建出的分布，这个分布即代表了模型的判别能力。

对于多类问题，假设类别数为  $C$ ，那就有了  $C-1$  个二分类器（参见小节 3.2.3.2），也就有了  $C-1$  个非目标分数，由此可以算出  $C-1$  个间隔值。因为需要找一个间隔值来满足所有的二分类器，所以本文取其中最小的一个二分类器对应的间隔值，该间隔值即为目标分数与最大的非目标分数之间的差：

$$m_{latent} = W_y^T f - \max_{j \neq y} \{W_j^T f\}. \quad (5-33)$$

与二类问题类似，在训练集上统计这些隐式间隔的值，就可以得到一个由所有隐式间隔所构成的分布，一个典型的分布如图 5-9(b) 所示，由于这个分布并不服从高斯分布，如果使用均值作为这个分布的统计量会产生一定的偏差，因此本文改用众数作为该分布的表征统计量，由图所示该分布是一个单峰分布，因此使用

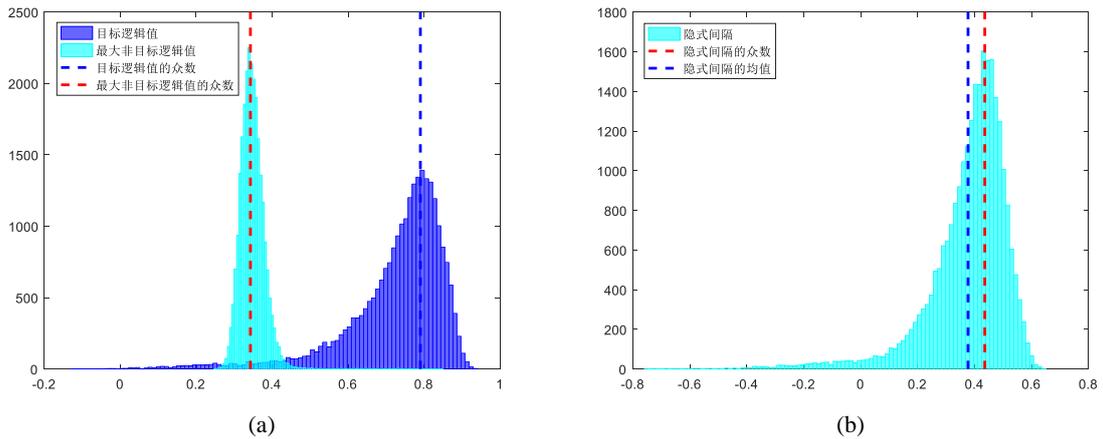


图 5-9 (a) 目标分数与最大非目标分数的分布；(b) 隐式间隔的分布。

均值漂移 (Mean Shift) 方法即可得到其众数。

在训练过程中，因为使用的是批量梯度下降法，所以如果要观察该统计量就需要在每个批次（一般为 256 个样本）上使用均值漂移进行一次迭代，并将得到的结果使用移动均值法 (Moving Average) 进行更新。本文中使用的均值漂移的初值为该批次样本的均值、窗口大小为该批次样本的标准差，移动均值法中使用的动量为 0.9。

使用该方法得到的隐式间隔的众数可以很好地反映出模型在训练集上建立起的间隔。注意到如果不考虑模型的泛化性能，对于分类问题来说间隔实际上是没有用处的，分类问题只需要找到分界面将样本分开即可，无需建立间隔，因此本节提出的指标并不能否定 Softmax 交叉熵损失函数在分类问题上的作用。

### 5.3.4 模型统计量小结

综合本章和第三章的理论分析部分，用于分析模型的统计量共有 4 个：

(1) 目标分数  $W_y^T \mathbf{f}$ 。

(2) LSE 函数  $LSE(\mathbf{f}; s) = \frac{1}{s} \log(\sum_{j=1, j \neq y}^c e^{s W_j^T \mathbf{f}})$ 。

(3) 最大非目标分数  $\max_{j \neq y} \{W_j^T \mathbf{f}\}$ 。

(4) 非目标 Softmax 加权的非目标分数  $\sum_{i=1, i \neq y}^C P_i^-(\mathbf{f}) W_i^T \mathbf{f}$ ，其中， $P_i^-(\mathbf{f}) = \frac{e^{W_i^T \mathbf{f}}}{\sum_{j=1, j \neq y}^n e^{W_j^T \mathbf{f}}}$ ， $i \neq y$

如图5-10所示，本文绘制了一个典型的人脸认证模型中，目标分数、LSE 函数值、最大非目标分数、Softmax 加权平均的非目标分数的分布直方图。从图上可以看到它们之间有着明显的大小关系，实际上后面三个统计量是存在大小关系的，即 LSE 函数值  $>$  最大非目标分数  $\geq$  Softmax 加权平均的非目标分数，其中第一个

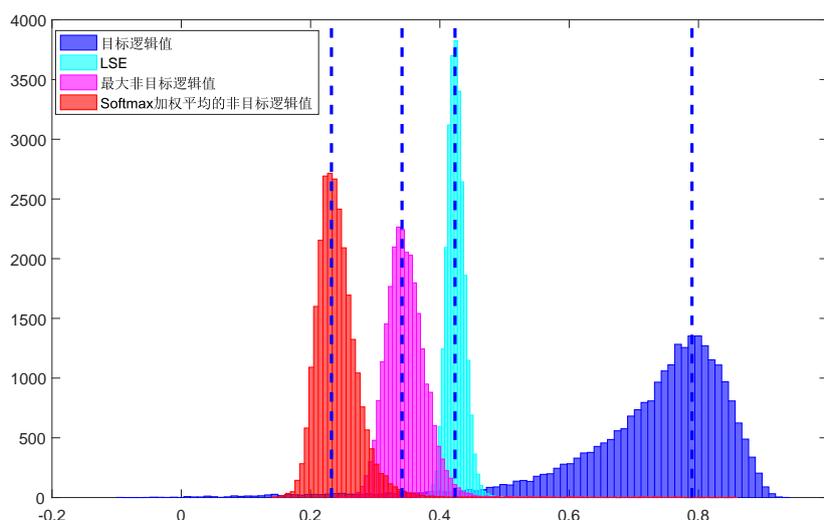


图 5-10 一个典型的人脸认证模型中，目标分数、LSE 函数值、最大非目标分数、Softmax 加权平均的非目标分数的分布直方图。

大于号的证明参见性质5.1，第二个大于等于号的证明如下：

证明：

$$\begin{aligned}
 \sum_{i=1, i \neq y}^C P_i^-(\mathbf{f}) \mathbf{W}_i^T \mathbf{f} &\leq \sum_{i=1, i \neq y}^C P_i^-(\mathbf{f}) \max_{j \neq y} \{\mathbf{W}_j^T \mathbf{f}\} \\
 &= \max_{j \neq y} \{\mathbf{W}_j^T \mathbf{f}\} \sum_{i=1, i \neq y}^C P_i^-(\mathbf{f}) \\
 &= \max_{j \neq y} \{\mathbf{W}_j^T \mathbf{f}\}.
 \end{aligned} \tag{5-34}$$

这里的等号当且仅当所有的  $\mathbf{W}_j^T \mathbf{f}, j \neq y$  均相等时成立，这种情况在实际应用中几乎不可能发生。证明完毕。 ■

这四个统计量之间的差值描述了多个模型的性质，例如：

(1) 目标分数与最大非目标分数之差称为隐式间隔，它用来评价模型的判别能力，详情参见小节5.3.3。

(2) 目标分数与非目标 Softmax 加权的非目标分数之差被用来分析参数  $s$  在优化过程中的升降问题，参见第三章的推论3.2。

(3) LSE 函数与最大非目标分数之差为 Softmax 交叉熵损失函数用于分类时，与目标函数之间的误差。

参见小节5.3.2，在推导 Softmax 交叉熵损失函数时，本文使用了 LogSumExp 函数来替换 Max 函数，由引理5.1可知，LSE 函数是大于 Max 函数的，用 LSE 函数

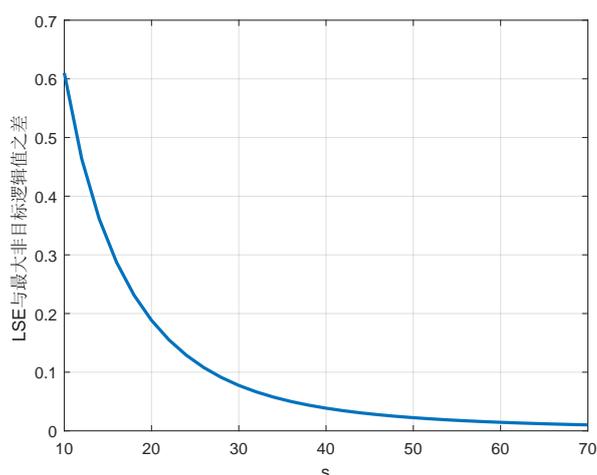


图 5-11 一个典型的人脸认证模型中，LSE 与最大非目标分数之差关于  $s$  的函数。

取代 Max 函数起到的作用与直接加一个间隔有异曲同工之效，另外，LogSumExp 函数引入的间隔并不是不变的，它会随着分数的尺度变化而变化，当  $s$  趋近于正无穷时 LSE 函数等于最大非目标分数，然而实际训练过程中不可能使用正无穷作为  $s$  的值，因此在训练时损失函数实际上是在加大目标分数与 LSE 函数之间的差，而不仅仅是最大的非目标分数。

本文从一个训练好的模型上随机取一个样本得到特征，并由该特征乘以不同的尺度系数来计算 LSE 函数，最后得到的 LSE 函数与尺度系数  $s$  之间的关系绘制在了图5-11中，从图中可以看到，在使用参数  $s = 30$  时，LSE 与最大非目标分数之差大概有  $0.8 \sim 0.9$  左右，这意味着通过实验得到的最优模型  $s = 30, m = 0.35$  的隐式间隔  $0.4406$  中，有大概  $0.8 \sim 0.9$  是由参数  $s = 30$  带来的。

在原始的 Softmax 交叉熵损失函数中，特征的幅度值会在初期有小幅度的下降，之后就会持续上升直到  $60 \sim 80$  左右（见第三章的小节3.2.3.1），也就是说 Softmax 交叉熵损失函数在训练初期会给予较大的间隔，而随着训练的进行渐渐地变成几乎仅优化最大非目标分数也就是前文提到的目标函数，这种逐步退火、逐渐接近理想情况的过程在训练过程中是随着特征幅度的增长而自动进行的，这也是 Softmax 交叉熵损失函数的优点之一。

(4) 目标分数与 LSE 函数之差是在消除掉  $s$  带来的影响下的隐式间隔。

由上一条的分析可知，隐式间隔中有少量的间隔是由参数  $s = 30$  带来的。此外，如果只做权重归一化，令特征长度自由优化，最终的特征幅度的平均值可以高达 70 左右，由图5-11可知，在尺度系数达到 70 时 LSE 与最大非目标分数之差几乎为 0，但带权重归一化的 Softmax 交叉熵损失函数仍然具备大概 0.13 左右的隐式间隔。这些隐式间隔一部分是由 Softplus 函数带来的，Softplus 函数会使得目

标分数超过 LSE 函数之后仍然具有较小的梯度，这样一些比较容易优化的样本对应的目标分数会得到进一步的提升，从而带来少量的隐式间隔。实际上即使仍然使用 ReLU 函数而不是 Softplus 函数，仍然有部分样本的目标分数会随着其他样本的优化而间接地得到提升，在这种情况下也会产生少量的隐式间隔。

由后续的实验（见小节5.4.4）可知，参数  $m$  与隐式间隔之间的关系近乎线性关系，隐式间隔与  $m$  回归出的直线的斜率大概是 0.62，也就是说 0.1 的  $m$  值仅能带来 0.062 的隐式间隔，最终使用的  $m = 0.35$  能够带来大概 0.22 的实际间隔，由此可见参数  $m$  带来的间隔仍然是隐式间隔最重要的组成部分。

由此可见隐式间隔实际上是由四部分组成：1) Softplus 函数带来的间隔；2) 其他样本间接优化提升的间隔；3) LSE 函数近似时带来的间隔；4) 参数  $m$  带来的间隔。以本章得到的最好的模型  $s = 30, m = 0.35$  为例，它的隐式间隔大概为 0.44，这其中前两项的贡献大概在 0.13 左右，LSE 函数的贡献大概在 0.08 左右， $m$  带来的间隔大概在 0.22 左右。

通过这些模型统计量可以建立起模型在训练数据集上的判别能力的直观感受，相信这些统计量会对将来对损失函数的进一步研究，尤其是对一些超参数的自动化设置上起到重要作用。

## 5.4 实验结果及分析

本小节将要对本章提出的损失函数的效果进行测试，为了公平地比较本章提出的方法与之前最好的几个方法，首先在小节5.4.1中对训练数据集与测试数据集进行了去重操作，以保证测试是开集测试；之后小节5.4.2介绍了实验使用的协议与细节；最后小节5.4.3将展示本章提出的算法与同时期的几个算法的性能对比。

### 5.4.1 数据集去重

如小节2.2.1所述，目前的人脸识别的性能测试主要都是在做开集测试，也就是说训练数据集和测试数据集中不应包含重复的身份，然而一直以来这条准则在学术界没有得到很好的遵守，比如说 CASIA-Webface 与 MegaFace set1 的比对集有 42 个身份重叠，然而 MegaFace set1 的查询集中总共只有 80 个身份，也就是说超过一半的身份都已经被训练过了，这样测试出来的结果就会受到很大的干扰，不能反映算法的真实性能。

鉴于目前大部分数据集都提供了图像身份对应的姓名列表，可以直接从姓名中检查出训练集和测试集里重合的身份，由于语言以及收集数据的方式的不同，还需要对姓名列表进行一些处理。有的欧洲国家的人的姓名里带有特殊的声调符

号，例如 *Šárka Vaňková* 就需要转换成为不带声调的 *Sarka Vankova*；有些国家的人名在一个数据集中是英文，在另一个数据集中却是自己国家的文字，这时需要统一将其翻译为英文来方便比对，例如神戶蘭子就需要翻译为 *Kobe Ranuko*；有的数据集中人的姓名用的是艺名，这样的名字需要替换成其本名，例如嘻哈音乐歌手 *T-Pain* 的本名实际上是 *Faheem Rashad Najm*。

在讲姓名列表中的语言统一后，就可以进行姓名比对了。姓名比对时要考虑到东亚地区的姓和名的前后顺序与西方是不同的，有的数据集中是以“姓名”的形式记录的，而有的数据集是按照西方的“名姓”的形式记录的，例如章子怡女士在 LFW 数据集中的记录是 *Zhang Ziyi* 而在 CASIA-Webface 数据集中的记录是 *Ziyi Zhang*。还有一个要考虑的问题是西方人有的时候会将名字进行缩写，只保留首字母，这样的情况极容易匹配错误，因此在匹配时同时也会去网络上检索出使用该缩写的最有名的人物来辅助判断缩写前后的两个名字是否是同一个人的。

最后，本文在 CASIA-Webface 与 LFW 之间检查出了 17 对重复身份，在 CASIA-Webface 与 MegaFace set1 之间检查出了 42 对重复身份，为了确保测试流程是开集测试，在后续的实验中会将这些身份从训练集中剔除出去再进行训练，这样做会使得一些过去论文中的算法性能不如其原始论文中的结果（表5-1），但为了学术严谨性，本文认为去重之后的结果才是真实的、可比较的。

表 5-1 去重前后的模型性能对比

损失函数	去重与否	MegaFace Rank 1	MegaFace VR
AM-Softmax	否	75.23%	87.06%
AM-Softmax	是	72.47%	84.44%

本小节中所使用的姓名处理方法均为计算机自动处理，代码已经公开<sup>①</sup>，以方便其他研究者们使用，同时也呼吁研究者在训练模型时使用去重后的数据集进行训练。

#### 5.4.2 实验细节

本章的实验设置与前两章大体相同，只在基础网络结构上做了一些调整，与 ResNet [16] 中的残差模块一致，本章在网络中抛弃了池化层，而改用了步进为 2 的卷积层来作为下采样的手段。为了能更快地验证算法的性能，本文使用了仅含有 20 个卷积层的残差卷积神经网络。

由于上一节所述的训练与测试集重合的问题，本节将不采用任何之前文章的

<sup>①</sup> <https://github.com/happynear/FaceDatasets>

表 5-2 多个损失函数在 LFW<sup>[55]</sup><sup>[104]</sup> 数据集上的性能

损失函数	$m$	LFW <sup>[55]</sup>	LFW BLUFR <sup>[104]</sup>	LFW BLUFR <sup>[104]</sup>
		6,000 对	VR@FAR=0.01%	DIR@FAR=1%
Softmax	-	97.08%	60.26%	50.85%
Softmax+75% dropout <sup>[125]</sup>	-	98.62%	77.64%	63.72%
Center Loss <sup>[63]</sup>	-	99.00%	83.30%	65.46%
NormFace <sup>[121]</sup>	-	98.98%	88.15%	75.22%
A-Softmax <sup>[64]</sup>	~1.5	99.08%	91.26%	81.93%
AM-Softmax	0.25	99.13%	91.97%	81.42%
AM-Softmax	0.3	99.08%	93.18%	84.02%
AM-Softmax	0.35	98.98%	93.51%	84.82%
AM-Softmax	0.4	99.17%	93.60%	84.51%
AM-Softmax	0.45	99.03%	93.44%	84.59%
AM-Softmax	0.5	99.10%	92.33%	83.38%
AM-Softmax 无特征归一化	0.35	99.08%	93.86%	<b>87.58%</b>
AM-Softmax 无特征归一化	0.4	99.12%	<b>94.48%</b>	87.31%

结果，而是挑选了其中一些方法，如纯 Softmax 交叉熵损失函数、带 Dropout<sup>[125]</sup> 的 Softmax 交叉熵损失函数、中心损失<sup>[63]</sup>、余弦 Softmax 交叉熵损失函数<sup>[121]</sup>、SphereFace<sup>[64]</sup> 等，本文使用其开源代码按照其原来的参数重新在去过重的数据集上重新训练并测试得到新的结果作为对比。因为在增加了间隔项后，梯度整体都有所提升，所以本文在文献 [64] 的 3 个学习率阶段（0.1、0.01、0.001）的基础上增加第 4 个阶段（0.0001），第四个阶段共持续 2,000 次迭代。

本章所提出的模型将在两个测试集 LFW<sup>[55]</sup> 和 MegaFace<sup>[102]</sup> 上进行测试，LFW 上使用 6,000 对协议与 BLUFR 协议<sup>[104]</sup> 进行测试，这两个协议在第三章中有过详细的描述，在此不再赘述。在 MegaFace<sup>[102]</sup> 上采用在 100 万张干扰图像下的首选正确率与百万分之一虚警率下的召回率两个指标进行定量评测，同时也会给出 CMC 曲线与 ROC 曲线来展示本章提出的算法在多选正确率与其他虚警率下的性能。

### 5.4.3 参数 $m$ 的作用

参数  $m$  是本章提出的加性间隔的核心参数，表格 5-2 与表格 5-3 分别展示了 LFW 和 MegaFace 上的性能，本文设置  $m$  从 0.25 到 0.5 之间变化，每 0.05 做一次

表 5-3 多个损失函数在 MegaFace<sup>[102]</sup> 上的性能

损失函数	$m$	隐式间隔	MegaFace <sup>[102]</sup>	MegaFace <sup>[102]</sup>
			Rank1@1e6	VR@FAR=1e-6
Softmax	-	0.0838	45.26%	50.12%
Softmax+75% dropout <sup>[125]</sup>	-	0.2117	57.32%	65.58%
Center Loss <sup>[63]</sup>	-	0.2492	63.38%	75.68%
NormFace <sup>[121]</sup>	-	0.2926	65.03%	75.88%
A-Softmax <sup>[64]</sup>	~1.5	0.3944	67.41%	78.19%
AM-Softmax	0.25	0.3430	70.81%	83.01%
AM-Softmax	0.3	0.4158	72.01%	83.29%
AM-Softmax	0.35	0.4406	<b>72.47%</b>	<b>84.44%</b>
AM-Softmax	0.4	0.4731	72.44%	83.50%
AM-Softmax	0.45	0.4957	72.22%	83.00%
AM-Softmax	0.5	0.5142	71.56%	82.49%
AM-Softmax 无特征归一化	0.35	0.419	70.71%	82.66%
AM-Softmax 无特征归一化	0.4	0.4563	70.96%	83.11%

实验。从两个表格中可以看到，当  $m$  取到 0.35 到 0.4 之间时，模型性能达到最佳。除了本章提出的方法之外，本章还进行了无特征归一化时的实验，实验结果表明在没有特征归一化时，LFW BLUFR 上的性能会得到显著提升，但 MegaFace 上的性能有明显的下降，根据第三章的 4.3.1 节的理论，特征归一化起到的是难例挖掘的作用，因此特征归一化在较难分别的样例上效果比较好，而 LFW 的样本质量大多较高，所以使用无特征归一化的方法效果较好。

从表格 5-2 中可以看到，LFW 的 6,000 对协议上的指标已经饱和，失去了评判模型质量好坏的能力，从 Center Loss<sup>[63]</sup> 开始，模型在 LFW 6,000 对协议上的性能都相差不大，均在误差范围以内，算法带来的提升比模型本身的误差还小。该指标在过去几年中为推动人脸认证领域的发展做出了卓越的贡献，现在已经完成了其历史使命，在今后的研究中我们将不会再使用该指标来评价模型的质量。

图 5-12 绘制了 MegaFace 上的 CMC 曲线和 ROC 曲线，从图上可以看到，本章提出的方法比之前的方法有了较大幅度的提升，尤其是在虚警率需要控制在比较低（例如百万分之一）的场景下的性能提升尤为显著。

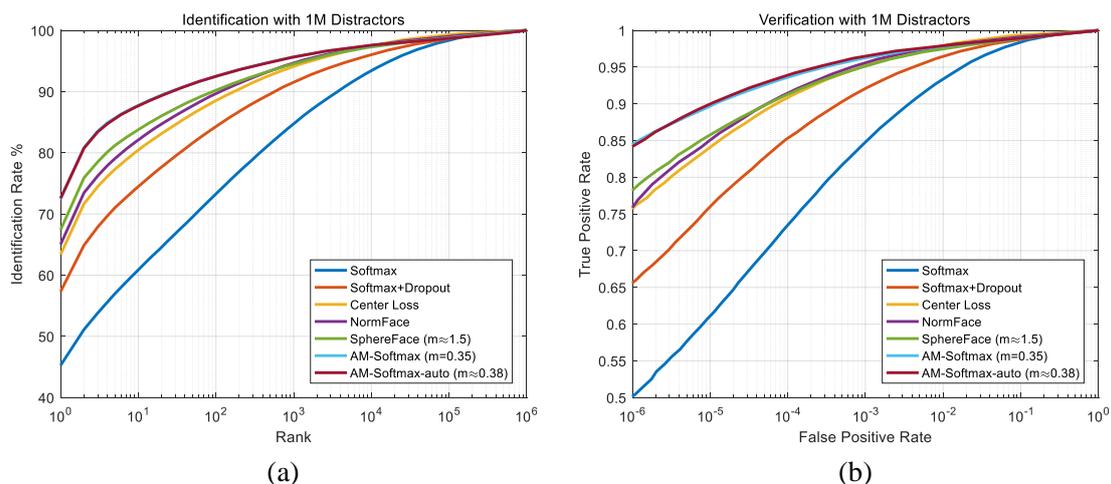


图 5-12 (a) MegaFace<sup>[102]</sup> Set 1 上各损失函数在 100 万个干扰图下的 CMC 曲线；(b) MegaFace<sup>[102]</sup> Set 1 上各损失函数的 ROC 曲线。

### 5.4.4 模型隐式间隔

表格5-3中列出了各个模型所建立的隐式间隔的众数，可以看到在隐式间隔达到 0.45 之前，模型的性能随着隐式间隔的提升而提升，而一旦超过 0.45 之后，模型的性能反而会随着隐式间隔的提升而下降，这意味着模型的隐式间隔并不是越高越好，而是有一个阈值的存在，为什么有这个阈值的存在、这个阈值代表了什么含义将会是下一步的重点研究方向。

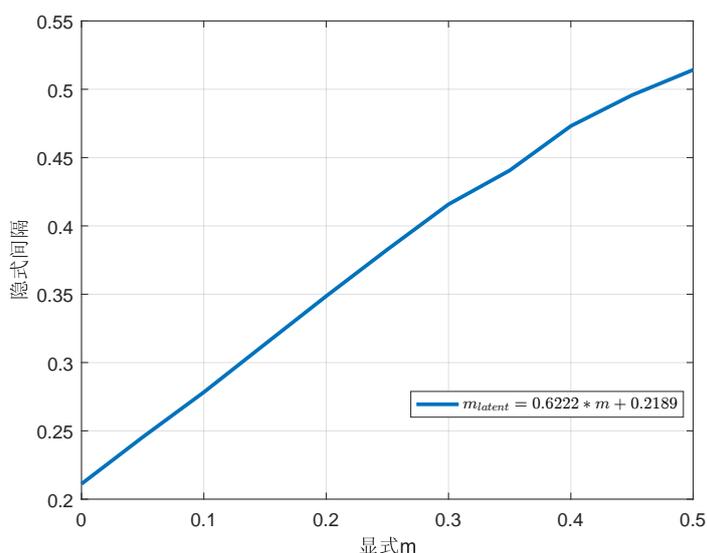


图 5-13 隐式间隔关于显示参数  $m$  的函数。

如图5-13所示，本文绘制了隐式间隔关于显示参数  $m$  的函数。从图上可以看

到，参数  $m$  带来的判别能力几乎是线性的，不过因为  $m$  有上限的存在，所以模型的判别能力也不能无限制提升。当  $m$  设置为 1 时，其隐式间隔将达到 0.5327，这比  $m = 0.5$  时所产生的 0.5142 的隐式间隔只多了一点点，但模型的性能会产生大幅度下降。其本质原因在于在这种情况下即使目标分数达到最大值 1，目标分数所接受到的梯度仍旧为 1，与完全没有分对的样本的梯度一样。困难样本与容易样本的权重完全一致会导致模型没有侧重性，大量的梯度信号都浪费在了已经分对了的样本上，导致困难样本相对而言得不到足够的梯度来向类代理靠拢。

## 5.5 本章小结

本章提出了在类间引入间隔的思想，在第三章的余弦 Softmax 交叉熵损失函数的基础上增加了一个间隔项来进一步提升类间间距，从而带来了更好的判别能力。本章提出的损失函数在两个测试集 LFW<sup>[55]</sup> 和 MegaFace<sup>[102]</sup> 上均比第三章的损失函数有了巨大的提升，也超过了其他研究者所提出的损失函数的性能。

本章对间隔项进行了理论分析，阐释了间隔项的几何意义，提出了模型的隐式间隔的概念，隐式间隔能很好地表达模型的判别能力。本章还对各损失函数进行了多项可视化展示，以方便读者能更加直观地理解各个损失函数起到的作用。

本章算法的不足之处在于引入了两个新的超参数。这两个超参数都需要手动调整来得到最佳的性能，考虑到现在的训练数据集都非常庞大，动辄需要几天的时间才能训练一次，两个超参数的调参工作是非常繁重的，因此超参数的自动调节是一项非常重要的研究课题。

本章中提出了一系列的分析模型判别能力的统计量，在未来研究超参数的自动调节时，相信这些统计量能够帮助研究者们理解模型并推导出一些模型的性质来辅助超参数的设置。

本章所提出的损失函数的间隔项是一个全局共享的变量，对于所有类别、所有样本使用的都是同一个间隔参数，然而类别之间的距离有远有近，样本之间也有相似与不相似之分，全部使用同一个变量可能并不是一个很好的方案，在未来的工作中可以尝试针对类别、甚至针对单个样本分别设置不同的间隔，有可能会提升模型的性能。

## 第六章 全文总结与展望

### 6.1 全文总结

本学位论文是作者在博士期间对人脸认证算法研究的总结与讨论。本文的主要工作在于设计适用于人脸认证系统的深度学习损失函数，目的是设计出一套学术上创新、工业上实用的人脸认证损失函数。从这个目的出发，本文在传统的 Softmax 交叉熵损失函数的基础上进行了两项改进，这两项改进相辅相成，均为目前最先进的人脸认证损失函数的组成部分。本文还创新性地提出了将度量学习的方法改造为分类损失函数的策略，使其既具有度量学习方法类间距离较大的优势，又避免了度量学习方法需要进行样本采样的缺点。本学位论文所有代码全部开源，为人脸识别领域贡献了一份力量。本文的主要贡献总结如下：

#### 1. $L_2$ 超球面嵌入

本文通过数学理论阐释了在人脸认证损失函数中进行  $L_2$  超球面嵌入的原因和意义，为  $L_2$  归一化操作提供了理论支持。本文通过实验发现当在神经网络中直接进行  $L_2$  超球面嵌入时，会产生网络不收敛的问题。之后本文又通过一个数学定理解释了网络不收敛的原因，并且提出需要在  $L_2$  超球面嵌入后引入尺度因子的办法来解决这一问题。本文对这个尺度系数的取值以及性质进行了一系列的分析，并且还解释了  $L_2$  超球面嵌入在难例挖掘和处理类不均衡问题上的作用。实验证明仅需使用该损失函数对训练好的模型微调若干轮即可使模型性能得到可观的提升。 $L_2$  超球面嵌入目前已经成为业内的标准做法。

#### 2. 度量学习损失函数的分类化改造

在进行了  $L_2$  归一化之后，本文指出 Softmax 交叉熵损失函数前的权重矩阵的本质作用是为每个类别分配一个“类代理”向量。通过将类代理的概念引入到度量学习的一系列损失函数中，本文将两种度量学习损失函数改造成了分类的损失函数，成功地解决了度量学习算法难以对样本进行采样的缺点。本文还通过一个定理给出了这样替换带来的误差以及这个替换对于两个损失函数中超参数  $m$  的影响。

#### 3. 加性间隔

本文在  $L_2$  超球面嵌入的基础上提出了在类别之间插入间隔的算法，通过一系列对间隔形式的尝试，最终发现加性的间隔的效果最好，而且形式较为简单，非常容易复现。本文还提出了一种名为“隐式间隔”的评价指标，它能够在训练过程中就对模型的判别能力进行评价。本文还对隐式间隔与本文提出的加性间隔之

间的关系进行了一系列的分析，希望能对自动设置间隔带来一些启发。

本文还从最优化的角度对传统的 Softmax 交叉熵损失函数进行了分析，有别于传统的概率角度的解释，基于最优化的解释是在神经网络输出分数的基础上进行的，比概率更提前一层。从最优化的角度很自然就可以看出引入间隔项的意义，而从概率角度则很难对间隔项进行理论分析。

相比于传统的 Softmax 交叉熵损失函数，结合了  $L_2$  超球面嵌入与加性间隔的 Softmax 交叉熵损失函数在大型测试数据集 MegaFace<sup>[102]</sup> 上提升了超过 27 个百分点，是目前学术界领先的人脸认证损失函数。

#### 4. 数据集去重

本文对一些训练数据集与测试数据集之间存在的身份重复问题进行了研究，通过一系列的姓名匹配策略找到了这些数据集之间的重复身份，并通过实验说明了重复身份对最终性能的影响。

#### 5. 开源代码

本学位论文所用到的代码和模型均在<https://github.com/happynear>上进行了开源，为人脸识别社区的发展贡献了一份力量，这些代码已经被用在多项研究和工程项目当中。

## 6.2 不足以及后续工作展望

人脸认证虽然在各个数据集上均取得了很好的成绩，但对于人脸认证以及深度学习模型的认识目前仍有不足。对于分类损失函数和度量学习损失函数之间的异同目前只是有了一些初步的理解，对于其背后的理论我们仍然几乎一无所知。我们目前的研究模式更多的是进行一系列的尝试之后，发现效果较好的一部分算法再分析其原因，而不是从问题的本质出发来设计算法。具体而言，本文所存在的不足有以下几个方面：

#### 1. 超参数的自动设置

在本文提出的基于  $L_2$  超球面嵌入和加性间隔的 Softmax 交叉熵损失函数中，引入了两个超参数  $s$  和  $m$ ，虽然本文对这两个超参数起到的作用进行了一系列的分析，但对于这两个超参数何时达到最优仍然没有做出有说服力的解释，对于这两个超参数目前还只能通过暴力搜索来进行参数调整。

#### 2. 其他的间隔形式

本学位论文只介绍了乘性角度间隔和加性余弦间隔两种间隔形式，实际上我们也对一些其他的间隔形式进行了尝试，但目前对于何种形式的间隔是最优的没有任何解释，其他的更具备想象力的间隔形式还有待挖掘。

### 3. 类不均衡问题

人脸数据集的各个类别中的样本数量往往差别很大，使用本文的算法会产生对数量较少的类别训练不足的情况，虽然归一化操作能够对类不均衡问题进行一定的缓解，但并未从根本上设计算法来解决这一问题。

### 4. 数据集噪声问题

学术界的人脸数据集往往采集自网络，其中存在大量的错误标签，本文所提出的算法对这些错误的样本仍然会给以非常大的梯度，这些较大的梯度会使得神经网络的训练遭到很大的干扰。

为了解决这些问题，我们计划从以下几个方面来开展研究：

1. 研究两个超参数  $s$  和  $m$  的本质，尝试找到其最优解，或者如何在网络的训练过程中自动地对参数进行调整，而无需人工干预。

2. 对间隔的作用进行进一步的分析，找到更好的间隔形式或者解释为什么加性间隔的效果优于其他间隔形式。

3. 尝试在损失函数的设计时就考虑类不均衡问题和数据集噪声问题，在建模层次上解决这两大问题。

4. 在工业界的超大规模数据集上，例如如果有几百万上千万个身份时，使用基于分类的损失函数其最后的权重矩阵会非常巨大，这通常是不可接受的，因此研究如何在类别特别巨大的情况下使用本文所提出的方法仍有待研究。

5. 本文中提出的方法和相关分析并不仅限于人脸认证任务，在其他的度量学习任务上应该也可以有性能提升，比如说行人再识别、图像检索等领域都是非常适合进行尝试的，我们会在未来将本文提出的方法应用在这些任务上来看它们是否有更好的表现。

6. 为人脸识别设计专用的网络结构。目前人脸识别使用的网络结构均为通用图像识别上所研究出来的网络，而没有针对人脸的特性专门设计网络结构。人脸在对齐之后各个区域的模式特征具备明显的区别，而现有网络并没有利用到这一点。

## 致 谢

本科、硕士、博士，转眼之间我在电子科技大学已经度过了 10 个年头，在这临近尾声之际，我想对帮助过我支持过我的老师、家人和朋友表达我的感激之情。

首先要感谢我硕士和博士期间的恩师程建教授，还记得当初我在考研选择方向的时候，想起了在程建老师的模式识别课程上风趣幽默的讲授，使我对模式识别以及人工智能领域产生了极大的兴趣，也促使我选择了程建老师作为我的本科毕设以及研究生导师。在这 6 年多的时间里，程建老师给予了我很多生活、学业、工程和研究方面的支持，带领我从一个对学术研究一无所知、甚至连阅读英文论文都吃力的本科生成长为一个合格的博士生。感谢程老师一路的帮助和鼓励，我将一生铭记。

感谢我在美国做访问学生时的导师 Alan Yuille 教授，Yuille 教授以 60 多岁的高龄，仍然在坚持进行科学研究，经常能在实验室里看到 Yuille 教授亲自推公式、与学生争辩技术细节的场景。还记得他拿到我的论文的第二天就带着标记得密密麻麻的几页文章来找我，并且一句一句地教我如何修改论文。Yuille 教授对科研的认真执着将一直激励我在未来的研究之路上前行。

还要感谢李鸿升老师，李鸿升老师在我研一的时候向我介绍了深度学习，正是对深度学习产生的兴趣使我决定继续深造，感谢李鸿升老师将我带进深度学习这个学术领域。

感谢曾经一起奋斗的同学们，这几年与刘海军、刘畅、项翔、罗晨旭、王惠宇等博士的相互讨论中，我萌生了一系列的想法最终形成了我的几篇论文，也祝愿你们能顺利博士毕业，有一个远大的前程。感谢在实验室一起工作、一起玩耍的小伙伴们，我会永远记得这几年度过的欢乐时光。也要谢谢无数在 QQ 群、知乎、GitHub 上与我讨论技术的朋友们，愿你们学业和工作顺利。

最后要感谢我的父母和我的妻子姜妍女士，谢谢你们在生活上给予我的帮助和关怀，在我低谷的时候给我鼓励，在我有所成就的时候与我一同分享快乐，你们是我追求学术理想时的坚强后盾。

## 参考文献

- [1] W. H. Burnham. Memory, historically and experimentally considered. i. an historical sketch of the older conceptions of memory[J]. *The American Journal of Psychology*, 1888, 2(1): 39-90
- [2] W. S. McCulloch, W. Pitts. A logical calculus of the ideas immanent in nervous activity[J]. *The bulletin of mathematical biophysics*, 1943, 5(4): 115-133
- [3] J. C. Principe, N. R. Euliano, W. C. Lefebvre. *Neural and adaptive systems: fundamentals through simulations*[M]. Wiley New York, 2000
- [4] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain.[J]. *Psychological Review*, 1958, 65(6): 386
- [5] M. Minsky, S. A. Papert. *Perceptrons: an introduction to computational geometry*[M]. MIT Press, 1969
- [6] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning representations by back-propagating errors[J]. *Nature*, 1986, 323(6088): 533
- [7] A. E. Bryson, W. F. Denham, S. E. Dreyfus. Optimal programming problems with inequality constraints[J]. *AIAA Journal*, 1963, 1(11): 2544-2550
- [8] A. E. Bryson. *Applied optimal control: optimization, estimation and control*[M]. Routledge, 1969
- [9] Y. LeCun, L. Bottou, Y. Bengio, et al. Gradient-based learning applied to document recognition[J]. *Proceedings of the IEEE*, 1998, 86(11): 2278-2324
- [10] S. Hochreiter, J. Schmidhuber. Long short-term memory[J]. *Neural Computation*, 1997, 9(8): 1735-1780
- [11] G. E. Hinton, R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks[J]. *Science*, 2006, 313(5786): 504-507
- [12] K. Jarrett, K. Kavukcuoglu, Y. LeCun, et al. What is the best multi-stage architecture for object recognition[C]. *IEEE International Conference on Computer Vision*, 2009, 2146-2153
- [13] A. Krizhevsky, I. Sutskever, G. E. Hinton. Imagenet classification with deep convolutional neural networks[C]. *Advances in Neural Information Processing Systems*, 2012, 1097-1105
- [14] K. He, X. Zhang, S. Ren, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification[C]. *IEEE International Conference on Computer Vision*, 2015, 1026-1034

- 
- [15] S. Ioffe, C. Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift[C]. International Conference on Machine Learning, 2015, 448-456
- [16] K. He, X. Zhang, S. Ren, et al. Deep residual learning for image recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2016, 770-778
- [17] K. He, X. Zhang, S. Ren, et al. Identity mappings in deep residual networks[C]. European Conference on Computer Vision, 2016, 630-645
- [18] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition[C]. International Conference on Learning Representations, 2015
- [19] J. Hu, L. Shen, G. Sun. Squeeze-and-excitation networks[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2018, 7132-7141
- [20] R. Girshick, J. Donahue, T. Darrell, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2014, 580-587
- [21] R. Girshick. Fast r-cnn[C]. IEEE International Conference on Computer Vision, 2015, 1440-1448
- [22] S. Ren, K. He, R. Girshick, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]. Advances in Neural Information Processing Systems, 2015, 91-99
- [23] W. Liu, D. Anguelov, D. Erhan, et al. Ssd: Single shot multibox detector[C]. European Conference on Computer Vision, 2016, 21-37
- [24] J. Long, E. Shelhamer, T. Darrell. Fully convolutional networks for semantic segmentation[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2015, 3431-3440
- [25] L.-C. Chen, G. Papandreou, I. Kokkinos, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(4): 834-848
- [26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al. Generative adversarial nets[C]. Advances in Neural Information Processing Systems, 2014, 2672-2680
- [27] M. Arjovsky, S. Chintala, L. Bottou. Wasserstein generative adversarial networks[C]. International Conference on Machine Learning, 2017, 214-223
- [28] J.-Y. Zhu, T. Park, P. Isola, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]. IEEE International Conference on Computer Vision, 2017, 2242-2251

- [29] L. A. Gatys, A. S. Ecker, M. Bethge. Image style transfer using convolutional neural networks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, 2414-2423
- [30] W. W. Bledsoe. The model method in facial recognition[J]. Panoramic Research Inc., Palo Alto, CA, Rep. PR1, 1966, 15(47): 2
- [31] A. J. Goldstein, L. D. Harmon, A. B. Lesk. Identification of human faces[J]. Proceedings of the IEEE, 1971, 59(5): 748-760
- [32] T. Kanade. Picture processing system by computer complex and recognition of human faces[M]. Kyoto University, 1974
- [33] I. J. Cox, J. Ghosn, P. N. Yianilos. Feature-based face recognition using mixture-distance[C]. IEEE Conference on Computer Vision and Pattern Recognition, 1996, 209-216
- [34] G. J. Kaufman, K. J. Breeding. The automatic recognition of human faces from profile silhouettes[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1976, 113-121
- [35] L. D. Harmon, W. F. Hunt. Automatic recognition of human face profiles[J]. Computer Graphics and Image Processing, 1977, 6(2): 135-156
- [36] M. A. Turk, A. P. Pentland. Face recognition using eigenfaces[C]. IEEE Conference on Computer Vision and Pattern Recognition, 1991, 586-591
- [37] P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection[C]. European Conference on Computer Vision, 1996, 43-58
- [38] M. S. Bartlett, J. R. Movellan, T. J. Sejnowski. Face recognition by independent component analysis[J]. IEEE Transactions on Neural Networks, 2002, 13(6): 1450
- [39] P. J. Phillips. Support vector machines applied to face recognition[C]. Advances in Neural Information Processing Systems, 1999, 803-809
- [40] G. Guo, S. Z. Li, K. Chan. Face recognition by support vector machines[C]. IEEE International Conference on Automatic Face and Gesture Recognition, 2000, 196-201
- [41] G.-D. Guo, H.-J. Zhang. Boosting for fast face recognition[C]. IEEE International Conference on Computer Vision Workshop, 2001, 96-100
- [42] G. Zhang, X. Huang, S. Z. Li, et al. Boosting local binary pattern (lbp)-based face recognition[C]. Advances in Biometric Person Authentication, 2004, 179-186
- [43] X. He, S. Yan, Y. Hu, et al. Face recognition using laplacianfaces[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(3): 328-340

- 
- [44] J. Wright, A. Y. Yang, A. Ganesh, et al. Robust face recognition via sparse representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(2): 210-227
- [45] C. Liu, H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition[J]. IEEE Transactions on Image processing, 2002, 11(4): 467-476
- [46] D. G. Lowe. Object recognition from local scale-invariant features[C]. IEEE International Conference on Computer Vision, 1999, 1150-1157
- [47] T. Ahonen, A. Hadid, M. Pietikainen. Face description with local binary patterns: application to face recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 2037-2041
- [48] N. Dalal, B. Triggs. Histograms of oriented gradients for human detection[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2005, 886-893
- [49] Z. Li, J.-i. Imai, M. Kaneko. Robust face recognition using block-based bag of words[C]. International Conference on Pattern Recognition, 2010, 1285-1288
- [50] K. Simonyan, O. M. Parkhi, A. Vedaldi, et al. Fisher vector faces in the wild.[C]. British Machine Vision Conference, 2013, 4
- [51] Z. Cao, Q. Yin, X. Tang, et al. Face recognition with learning-based descriptor[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2010, 2707-2714
- [52] T.-H. Chan, K. Jia, S. Gao, et al. Pcanet: a simple deep learning baseline for image classification[J]. IEEE Transactions on Image Processing, 2015, 24(12): 5017-5032
- [53] D. Chen, X. Cao, F. Wen, et al. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2013, 3025-3032
- [54] C. Lu, X. Tang. Surpassing human-level face verification performance on LFW with gaussian-face[C]. U.S.A. National Conference on Artificial Intelligence, 2015, 3811-3819
- [55] G. B. Huang, M. Ramesh, T. Berg, et al. Labeled faces in the wild: a database for studying face recognition in unconstrained environments[R]. Technical Report 07-49, University of Massachusetts, Amherst, 2007
- [56] M. Wang, W. Deng. Deep face recognition: a survey[J]. arXiv preprint arXiv:1804.06655, 2018
- [57] Y. Taigman, M. Yang, M. Ranzato, et al. Deepface: Closing the gap to human-level performance in face verification[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2014, 1701-1708

- [58] Y. Sun, X. Wang, X. Tang. Deep learning face representation from predicting 10,000 classes[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2014, 1891-1898
- [59] Y. Sun, Y. Chen, X. Wang, et al. Deep learning face representation by joint identification-verification[C]. Advances in Neural Information Processing Systems, 2014, 1988-1996
- [60] F. Schroff, D. Kalenichenko, J. Philbin. Facenet: a unified embedding for face recognition and clustering[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2015, 815-823
- [61] O. M. Parkhi, A. Vedaldi, A. Zisserman. Deep face recognition.[C]. British Machine Vision Conference, 2015, 6
- [62] J. Liu, Y. Deng, T. Bai, et al. Targeting ultimate accuracy: Face recognition via deep embedding[J]. arXiv preprint arXiv:1506.07310, 2015
- [63] Y. Wen, K. Zhang, Z. Li, et al. A discriminative feature learning approach for deep face recognition[C]. European Conference on Computer Vision, 2016, 499-515
- [64] W. Liu, Y. Wen, Z. Yu, et al. Sphereface: Deep hypersphere embedding for face recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2017, 6738-6746
- [65] A. G. Howard, M. Zhu, B. Chen, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017
- [66] S. Chen, Y. Liu, X. Gao, et al. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices[J]. arXiv preprint arXiv:1804.07573, 2018
- [67] C. Xiong, X. Zhao, D. Tang, et al. Conditional convolutional neural network for modality-aware face recognition[C]. IEEE International Conference on Computer Vision, 2015, 3667-3675
- [68] F. Chollet. Xception: Deep learning with depthwise separable convolutions[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2017, 1800-1807
- [69] J. Dai, H. Qi, Y. Xiong, et al. Deformable convolutional networks[J]. CoRR, abs/1703.06211, 2017, 1(2): 3
- [70] Y. Jia, E. Shelhamer, J. Donahue, et al. Caffe: Convolutional architecture for fast feature embedding[C]. ACM International Conference on Multimedia, 2014, 675-678
- [71] W. Shang, K. Sohn, D. Almeida, et al. Understanding and improving convolutional neural networks via concatenated rectified linear units[C]. International Conference on Machine Learning, 2016, 2217-2225
- [72] D.-A. Clevert, T. Unterthiner, S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus)[J]. arXiv preprint arXiv:1511.07289, 2015

- [73] B. Xu, N. Wang, T. Chen, et al. Empirical evaluation of rectified activations in convolutional network[J]. arXiv preprint arXiv:1505.00853, 2015
- [74] J. Bjorck, C. Gomes, B. Selman. Understanding batch normalization[J]. arXiv preprint arXiv:1806.02375, 2018
- [75] J. L. Ba, J. R. Kiros, G. E. Hinton. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016
- [76] X. Huang, S. J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization.[C]. IEEE International Conference on Computer Vision, 2017, 1510-1519
- [77] Y. Wu, K. He. Group normalization[C]. European Conference on Computer Vision, 2018, 3-19
- [78] S. Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models[C]. Advances in Neural Information Processing Systems, 2017, 1945-1953
- [79] B. C. Csáji. Approximation with artificial neural networks[J]. Faculty of Sciences, Eötvös Loránd University, Hungary, 2001, 24: 48
- [80] D. Balduzzi, M. Frean, L. Leary, et al. The shattered gradients problem: If resnets are the answer, then what is the question[J]. arXiv preprint arXiv:1702.08591, 2017
- [81] A. Veit, M. J. Wilber, S. Belongie. Residual networks behave like ensembles of relatively shallow networks[C]. Advances in Neural Information Processing Systems, 2016, 550-558
- [82] F. Wang, X. Xiang, C. Liu, et al. Regularizing face verification nets for pain intensity regression[C]. IEEE International Conference on Image Processing, 2017, 1087-1091
- [83] C. Szegedy, V. Vanhoucke, S. Ioffe, et al. Rethinking the inception architecture for computer vision[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2016, 2818-2826
- [84] G. Hinton, O. Vinyals, J. Dean. Distilling the knowledge in a neural network[C]. NIPS Deep Learning and Representation Learning Workshop, 2015
- [85] I. Loshchilov, F. Hutter. Sgdr: Stochastic gradient descent with warm restarts[C]. International Conference on Learning Representations, 2017
- [86] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence  $O(1/k^2)$ [C]. Doklady AN USSR, 1983, 543-547
- [87] J. Duchi, E. Hazan, Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization[J]. Journal of Machine Learning Research, 2011, 12(Jul): 2121-2159
- [88] D. P. Kingma, J. Ba. ADAM: a method for stochastic optimization[C]. International Conference on Learning Representations, 2015

- [89] S. J. Reddi, S. Kale, S. Kumar. On the convergence of ADAM and beyond[C]. International Conference on Learning Representations, 2018
- [90] Z. Zhang, L. Ma, Z. Li, et al. Normalized direction-preserving ADAM[J]. arXiv preprint arXiv:1709.04546, 2017
- [91] L. Wolf, T. Hassner, I. Maoz. Face recognition in unconstrained videos with matched background similarity[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2011, 529-534
- [92] B. F. Klare, B. Klein, E. Taborsky, et al. Pushing the frontiers of unconstrained face detection and recognition: IARPA janus benchmark a[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2015, 1931-1939
- [93] C. Whitelam, E. Taborsky, A. Blanton, et al. IARPA janus benchmark-b face dataset[C]. CVPR Workshop on Biometrics, 2017
- [94] B. Maze, J. Adams, J. A. Duncan, et al. IARPA janus benchmark-c: Face dataset and protocol[C]. 11th IAPR International Conference on Biometrics, 2018
- [95] S. Sengupta, J.-C. Chen, C. Castillo, et al. Frontal to profile face verification in the wild[C]. IEEE Winter Conference on Applications of Computer Vision, 2016, 1-9
- [96] S. Moschoglou, A. Papaioannou, C. Sagonas, et al. Agedb: the first manually collected, in-the-wild age database[C]. IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2017, 5
- [97] D. Yi, Z. Lei, S. Liao, et al. Learning face representation from scratch[J]. arXiv preprint arXiv:1411.7923, 2014
- [98] Y. Guo, L. Zhang, Y. Hu, et al. Ms-celeb-1m: a dataset and benchmark for large-scale face recognition[C]. European Conference on Computer Vision, 2016, 87-102
- [99] A. Bansal, A. Nanduri, C. D. Castillo, et al. Umdfaces: an annotated face dataset for training deep networks[C]. IEEE International Joint Conference on Biometrics, 2017, 464-473
- [100] Q. Cao, L. Shen, W. Xie, et al. Vggface2: a dataset for recognising faces across pose and age[C]. IEEE International Conference on Automatic Face & Gesture Recognition, 2018, 67-74
- [101] DeepGlint. Asian celebrity faces[OL]. <http://trillionpairs.deepglint.com/overview/>
- [102] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, et al. The megaface benchmark: 1 million faces for recognition at scale[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2016, 4873-4882

- [103] G. B. Huang, E. Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures[J]. Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep, 2014, 14-003
- [104] S. Liao, Z. Lei, D. Yi, et al. A benchmark study of large-scale unconstrained face recognition[C]. IEEE International Joint Conference on Biometrics, 2014, 1-8
- [105] J. Deng, J. Guo, S. Zafeiriou. Arcface: additive angular margin loss for deep face recognition[J]. arXiv preprint arXiv:1801.07698, 2018
- [106] K. Zhang, Z. Zhang, Z. Li, et al. Joint face detection and alignment using multitask cascaded convolutional networks[J]. IEEE Signal Processing Letters, 2016, 23(10): 1499-1503
- [107] Y. Liu, H. Li, J. Yan, et al. Recurrent scale approximation for object detection in cnn[C]. IEEE International Conference on Computer Vision, 2017
- [108] K. Zhang, Z. Zhang, H. Wang, et al. Detecting faces using inside cascaded contextual cnn[C]. IEEE International Conference on Computer Vision, 2017, 3171-3179
- [109] K. Q. Weinberger, J. Blitzer, L. K. Saul. Distance metric learning for large margin nearest neighbor classification[C]. Advances in Neural Information Processing Systems, 2006, 1473-1480
- [110] J. V. Davis, B. Kulis, P. Jain, et al. Information-theoretic metric learning[C]. International Conference on Machine learning, 2007, 209-216
- [111] J. Hu, J. Lu, Y.-P. Tan. Discriminative deep metric learning for face verification in the wild[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2014, 1875-1882
- [112] R. Hadsell, S. Chopra, Y. LeCun. Dimensionality reduction by learning an invariant mapping[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2006, 1735-1742
- [113] K. Q. Weinberger, L. K. Saul. Distance metric learning for large margin nearest neighbor classification[J]. Journal of Machine Learning Research, 2009, 10(Feb): 207-244
- [114] Y. LeCun, C. Cortes, C. Burges. The mnist database of handwritten digits[OL]. <http://yann.lecun.com/exdb/mnist/>
- [115] W. Rudin, et al. Principles of mathematical analysis, chapter 10[M]. McGraw-Hill New York, 1964
- [116] G. Pereyra, G. Tucker, J. Chorowski, et al. Regularizing neural networks by penalizing confident output distributions[C]. International Conference on Learning Representations Workshop, 2017
- [117] H. Xiao, K. Rasul, R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms[J]. arXiv preprint arXiv:1708.07747, 2017

- [118] I. J. Goodfellow, D. Warde-Farley, M. Mirza, et al. Maxout networks.[C]. International Conference on Machine Learning, 2013, 1319-1327
- [119] X. Wu, R. He, Z. Sun. A lightened cnn for deep face representation[J]. arXiv preprint arXiv:1511.02683, 2015
- [120] Y. Tang. Deep learning using linear support vector machines[J]. arXiv preprint arXiv:1306.0239, 2013
- [121] F. Wang, X. Xiang, J. Cheng, et al. Normface: L2 hypersphere embedding for face verification[C]. ACM international Conference on Multimedia, 2017
- [122] A. Banerjee, I. S. Dhillon, J. Ghosh, et al. Clustering on the unit hypersphere using von mises-fisher distributions[J]. Journal of Machine Learning Research, 2005, 6(Sep): 1345-1382
- [123] W. Liu, Y. Wen, Z. Yu, et al. Large-margin softmax loss for convolutional neural networks[C]. International Conference on Machine Learning, 2016, 507-516
- [124] C. Cortes, V. Vapnik. Support vector networks[J]. Machine learning, 1995, 20(3): 273-297
- [125] N. Srivastava, G. E. Hinton, A. Krizhevsky, et al. Dropout: a simple way to prevent neural networks from overfitting.[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958

## 攻读博士学位期间取得的成果

- [1] **F. Wang**, X. Xiang, C. Liu, et al. Regularizing face verification nets for pain intensity regression[C]. 2017 IEEE International Conference on Image Processing (ICIP), 2017, 1087-1091 (CCF C 类会议)
- [2] **F. Wang**, X. Xiang, J. Cheng, et al. Normface: L2 hypersphere embedding for face verification[C]. Proceedings of the 25th ACM International Conference on Multimedia, 2017 (CCF A 类会议)
- [3] **F. Wang**, H. Liu, J. Cheng. Visualizing deep neural network by alternately image blurring and deblurring[J]. Neural Networks, 2018, 97: 162-172 (一区 SCI, IF 7.197)
- [4] **F. Wang**, J. Cheng, W. Liu, et al. Additive margin softmax for face verification[J]. IEEE Signal Processing Letters, 2018, 25(7): 926-930 (三区 SCI, IF 2.813)
- [5] **F. Wang**, J. Cheng, Y. Jiang. Ridge-slope-valley feature for fingerprint liveness detection[C]. The Proceedings of the Third International Conference on Communications, Signal Processing, and Systems, 2015, 857-865 (EI 会议)
- [6] J. Cheng, H. Liu, **F. Wang**, et al. Silhouette analysis for human action recognition based on supervised temporal t-sne and incremental learning[J]. IEEE Transactions on Image Processing, 2015, 24(10): 3203-3217 (二区 SCI, IF 5.071)
- [7] H. Liu, J. Cheng, **F. Wang**. Sequential subspace clustering via temporal smoothness for sequential data segmentation[J]. IEEE Transactions on Image Processing, 2018, 27(2): 866-878 (二区 SCI, IF 5.071)
- [8] 程建, 王峰, 黎兰, 等等. 一种高分辨率遥感图像分割方法 [P]. 中国专利, 201310347917.4, 2016-05-04
- [9] 程建, 周圣云, 王峰, 等等. 一种基于 SVM 和稀疏表示的假指纹检测方法 [P]. 中国专利, 201310259382.5, 2016-11-16
- [10] 程建, 梁昊, 王峰, 等等. 一种基于低秩矩阵表示的目标跟踪方法 [P]. 中国专利, 201510916027.X, 2018-02-16
- [11] 程建, 邹瑞雪, 王峰, 等等. 基于 RGBD 描述符的室内场景语义分割方法 [P]. 中国专利, 201610023292.X, 2018-04-17
- [12] 参与项目: 国家自然科学基金面上项目, “基于深度时空层级模型的人体活动识别方法研究”, 编号: 61671125, 2017.01-2020.12

- [13] 参与项目：中国工程物理研究院横向科技合作项目，“远距离人脸识别系统研制”，2013.07-2015.12.
- [14] 参与项目：深圳中集天达空港设备有限公司横向科技合作项目，“基于激光扫描和机器视觉的可视化智能飞机泊位引导系统”，2014.01-2016.12