

密级： 保密期限：

# 北京邮电大学

## 硕士学位论文



题目： 基于对抗神经网络的人脸图片属性识别与生成

学 号： \_\_\_\_\_

姓 名： \_\_\_\_\_

专 业： 电子与通信工程

导 师： \_\_\_\_\_

学 院： 信息与通信工程学院

二〇一七年十一月三十日



# 基于对抗神经网络的人脸图片属性识别与生成

## 摘要

在模式识别与多媒体搜索领域，深度学习卷积神经网络是近年的新技术，凭借着简洁、有效、易训练等优势迅速在图像处理领域得到了广泛的应用。尤其是在人脸相关的领域，卷积神经网络的出现极大提升了人脸识别和人脸属性识别的准确率，俨然已经成为目前人脸领域的主流技术和最具前景的技术方向。作为神经网络另类演变，对抗生成网络初期是为了探究神经神经网络的内部构造原理。随着相关技术的不断进化，借助其可以生成逼真图像的特性，在图像重建领域也体现出了很强的实用价值。与此同时，随着监督式学习的性能瓶颈到来，迁移学习的理念作为非监督学习与监督学习之间的过渡，提出可以通过现有场景的数据和方法来探索未知场景下的识别任务，体现出具有较高的研究意义。

本文在工程实践和理论研究方面有所兼顾，首先介绍了卷积神经网络的基础结构，传统的理论训练和测试方法；随后介绍了工程上如何使用分布式多卡训练加快神经网络的训练；实际的生产中对于网络前馈所用到的优化技巧，诸如卷积优化，多个计算步骤合并等。最后借助于深度学习框架和指令集技术可以大幅提高训练速度以及 10 倍的前馈速度提升。

研究方面，本文主要研究人脸属性基础识别和人脸属性的迁移学习两个方面的问题。基础的人脸属性识别存在着对于多个数据集任务难以共同利用和网络输出可信程度的把控两个问题，通过调整网络的结构、改进图片预处理的方式、设计自评模块等方法对于相关问题进行了针对性地解决，也提升了人脸图片的属性识别准确性。在 morph 年龄数据集上绝对误差只有 3.5 岁，在 chalearn fotw 性别和微笑数据集准确率上也超过了 90%，而在实验室自己标注的的 5 类年龄数据集 top1 准确率达到了 93.6%。

另一方面结合对抗生成网络探究迁移学习在人脸属性上的应用，首先通过对于人脸图片进行生成，不断优化合成数据的真实度和广泛性，获得了可以生成真实人脸的神经网络。而为了让人脸属性模型应用于不同的使用场景，借助于人脸超分辨率的技术结合迁移学习的思想构建了可以对于 celeA 和 lfwA 两个数据集都具有良好表现的 40 类人脸属性模型训练方法。相比于原有模型提高了 10 个百分点。

**关键词：** 对抗生成网络 人脸属性 迁移学习 多机多卡 前馈优化

## GENERATIVE ADVERSARIAL NETWORK BASED FACE ATTRIBUTE RECOGNITION AND REGENERATION

### ABSTRACT

In the field of pattern recognition and multimedia search, deep learning convolutional neural network is a new technology in recent years. With its advantages of simplicity, efficiency and easy training, it has been widely used in image processing field. Especially in the face-related fields, the emergence of convolutional neural networks has greatly improved the accuracy of face recognition and face recognition, which has become the mainstream technology and the most promising technical direction in the current face area. As an alternative to the evolution of neural networks, the early days of confrontation generation networks were to explore the internal construction of neural networks. With the continuous evolution of related technologies, with its characteristics that can generate realistic images, it also shows strong practical value in the field of image reconstruction. At the same time, with the arrival of performance bottleneck in supervised learning, the concept of Migration Learning, as a transition between unsupervised learning and supervised learning, proposes that data and methods of existing scenes can be used to explore recognition tasks in unknown scenarios. Out of a higher research significance.

In this paper, both engineering practice and theoretical research take into account, first introduced the convolution neural network infrastructure, the traditional theory of training and testing methods; then introduced how to use the project to accelerate the training of distributed multi-card neural network training; actual Optimization techniques used in network feedforward, such as convolution optimization, merging multiple computation steps, and so on. Finally, with the help of deep learning framework and instruction set technology, training speed and 10 times feedforward speed increase can be greatly improved.

In the aspect of research, this paper mainly studies two aspects of the ba-

sic recognition of face attributes and the migration of face attributes. There are two basic problems in face recognition, such as the difficulty of multi-dataset tasks and the credibility of network output. By adjusting the network structure, improving the way of image pre-processing and designing self-assessment modules, Relevant issues have been targeted to solve, but also enhance the accuracy of face recognition of property images. The absolute error in the morph age dataset was only 3.5 years old, more than 90% in the chalearn fotw gender and smile dataset accuracy, and the top 5 accuracy of the 5 age-related datasets annotated by the laboratory reached 93.6% .

On the other hand, we combine the countermeasure generation network to explore the application of relocation learning to face attributes. Firstly, we generate the neural network that generates the real face by generating the face images and optimizing the authenticity and universality of the synthesized data. In order to apply the facial attribute model to different usage scenarios, the 40 types of face attributes that can perform well on both celeA and lfwA datasets are constructed with the help of the theory of face-super-resolution combined with migration learning Model training methods. Compared to the original model increased by 10 percentage points.

**KEY WORDS:** GAN face attribue transfer leearning Multi-machine multi-card Feedforward optimization

# 目 录

<b>第一章 绪论 .....</b>	1
1.1 课题研究的背景与意义 .....	1
1.1.1 研究背景 .....	1
1.1.2 研究意义 .....	1
1.2 国内外研究现状 .....	2
1.2.1 人脸属性的任务目标发展 .....	2
1.2.2 人脸属性识别方法变化 .....	2
1.3 本文的工作与贡献 .....	3
1.3.1 研究内容 .....	3
1.3.2 主要贡献 .....	3
1.3.3 论文的组织结构 .....	4
<b>第二章 卷积神经网络的相关技术介绍 .....</b>	5
2.1 卷积神经网络的基础操作和训练 .....	5
2.1.1 卷积神经网络结构的基本组成 .....	5
2.1.2 卷积神经网络常用激活函数 .....	12
2.1.3 卷积神经网络常用的参数初始化方法 .....	13
2.1.4 卷积神经网络的训练与优化 .....	14
2.2 神经网络训练速度的提升 .....	15
2.2.1 并行模式 .....	15
2.2.2 参数更新方式 .....	16
2.2.3 基于机器学习框架的多机多卡训练 .....	17
2.3 神经网络前馈速度优化 .....	18
2.3.1 卷积计算的优化方式 .....	19
2.3.2 不同网络层的合并 .....	23
2.3.3 本章小结 .....	24
<b>第三章 人脸多属性属性识别的架构 .....</b>	25
3.1 人脸属性性质分析 .....	25

3.1.1 人脸属性的类别 .....	25
3.1.2 多属性标签表示形式 .....	25
3.1.3 属性之间的相互联系 .....	26
3.2 人脸属性数据库简介 .....	27
3.3 基于传统特征的人脸属性识别 .....	30
3.4 基于共享神经网络特征和最大间隔分类器的人脸属性识别 .....	31
3.5 基于共享特征和子任务模块的端到端的人脸属性学习 .....	32
3.6 使用 SA-softmax 进行模型稳定化输出 .....	33
3.6.1 SA-softmax .....	33
3.6.2 动态调整标签的训练方法 .....	34
3.7 实验设置 .....	35
3.7.1 网络结构和训练环境 .....	35
3.7.2 数据集并行训练的方式改进问题一 .....	35
3.7.3 人脸矫正固定输入格式改进问题二 .....	37
3.7.4 网络自评估模块改进问题三 .....	38
3.8 实验结论与分析 .....	39
3.9 本章小结 .....	40
<b>第四章 对抗生成网络在人脸属性中的应用 .....</b>	<b>41</b>
4.1 对抗生成网络相关技术的介绍 .....	41
4.2 探究对抗神经网络的应用 .....	42
4.2.1 使用对抗生成网络生成真实图像 .....	42
4.2.2 使用对抗生成网络提高人脸分辨率 .....	46
4.3 结合对抗生成超像素实现迁移学习 .....	48
4.3.1 人脸属性的监督式学习困境 .....	48
4.3.2 人脸属性的迁移学习猜想 .....	49
4.3.3 基于人脸超分辨率的人脸属性迁移学习实验 .....	49
4.4 实验结果分析与结论 .....	49
4.5 本章小结 .....	51
<b>第五章 总结与展望 .....</b>	<b>53</b>
5.1 全文总结 .....	53
5.2 未来展望 .....	53

参考文献 .....	55
------------	----



## 表格索引

3-1	celeA 中的属性表 .....	27
3-2	SA-softmax 的置信度判断对照表 .....	33
3-3	在 Morph 和 LBS 上的准确率结果 .....	37
3-4	在 Chalearn 数据集上的性别和微笑准确率结果 .....	38
3-5	在 LBS 数据集上的性别准确率结果 .....	38
4-1	在 CELEA 数据集 .....	50



## 插图索引

2-1 卷积操作的示意图 .....	7
2-2 max-pooling 的操作示意图 .....	9
2-3 全连接层的操作示意图 .....	10
2-4 BN 层的操作说明 .....	11
2-5 激活函数的具体表达式以及出现时间 .....	12
2-6 sigmoid(a)、relu(b)、prelu(c) 函数的函数曲线示意图 .....	12
2-7 使用 gabor 固定初始化训练对于消耗能源的降低 .....	14
2-8 模型并行与数据并行示意图 .....	16
2-9 使用 7 层 FOR 循环实现卷积操作 .....	20
2-10 im2col 的操作示意图 .....	21
2-11 winograd 算法的算法过程 .....	21
2-12 MEC 算法的算法过程 .....	21
2-13 MKL/MKL-DNN 的简介 .....	22
2-14 nnpack .....	22
2-15 CUDNN .....	23
2-16 MKL、MKL2017、MKLDNN、openblas 加速方法的具体速度 .....	24
3-1 人脸属性之间的相互联系 .....	26
3-2 数据集图片示例 .....	29
3-3 基于 DIF 特征的人脸属性识别 .....	30
3-4 基于共享神经网络特征和 SVM 分类器的人脸属性识别 .....	31
3-5 基于端到的人脸属性学习 .....	32
3-6 数据集并行的网络结构 .....	36
3-7 数据集并行的网络结构 .....	36
3-8 人脸矫正效果演示 .....	38
4-1 DCGAN 的网络结构:(a)DCGAN 中的生成网络模型结构; (b)DCGAN 中的判决模型网络结构 .....	43
4-2 minist 图片生成的数字图片 .....	43
4-3 cifar10 图片生成的数字图片 .....	43

4-4	BGAN 的网络结构.....	44
4-5	人脸图片生成效果图 .....	44
4-6	对抗生成网络生成图片的局限性 .....	45
4-7	人脸超分辨率所使用的网络结构 .....	46
4-8	TRGAN 中基于 resnet 的三种网络结构改进.....	47
4-9	超分辨率使用 lfw 图片的效果示意图 .....	47

# 第一章 绪论

## 1.1 课题研究的背景与意义

### 1.1.1 研究背景

自从人类第一次张开眼睛观察世界开始，图像这一最早的原始信息传递媒介就开始以各种各样的形式在人类的信息传递过程中发挥着非同寻常的作用，正所谓“一图胜千言”，“耳听为虚，眼见为实”说明的正是这个道理。随着图像的表现形式不断的发展和图像数据的日益增加，如何提取图像中所包含的海量信息以及信息的分析使用成了现代多媒体，人工智能，自动化控制等多个领域都亟需解决的问题。近些年来图像领域人工智能迅猛发展，神经网络技术因为简单，高效，对于数据适应性强等特性，在各种图像识别的领域大规模训练使用，取得了非常好的效果。人脸领域受相关技术的影响，有了很大的进步，从慢慢接近人眼的人脸识别分辨效果，到不断超越，以至于后来的百万级人脸搜索 99% 的准确率，可以说正在慢慢朝着可以实际应用的方向发展。因此人脸领域也逐渐成为整个深度学习技术革命的排头兵。

### 1.1.2 研究意义

在人脸领域之中，包含人脸检测、人脸 landmark 点、人脸识别、人脸属性识别等多个分支。其中人脸识别作为图像信息中具有身份信息生物特征的一部分，可以广泛的用在安防，娱乐，多媒体等领域。而如果说人脸识别是识人，辨人。那么人脸属性识别就可以说是“相面算命”了，比如说人机交互中的表情识别互动，又比如说视频播放网站中限制级视频对于低龄观众限制，都是人工智能领域中不可缺少的功能。而在用户数据统计的过程中一张人脸图片，就可以识别出用户性别，年龄，是否戴眼镜，基本面部特征，发型状态等信息，这不仅让人听起来就很兴奋，而且可以在诸多面向用户的业务中实现个性化的分析与定制推荐。这些应用在逐渐强调个性化发展的社会中具有很高的市场。

但是人脸属性作为一项人脸中的重要研究领域，在深度学习中的技术进展却不如人脸识别领域一样快速，无论是准确率还是实际使用都有一定的发展空间。其中主要的问题在于网络结构上对于人脸属性多样性兼容问题，以及人脸属性任务对于人脸图片数据要求的复杂和严格性，人脸属性种类繁多，且人脸场景分布极为复杂，

标注工作难度较大，且歧义性较大。因此，本文旨在在深度学习对于图像识别任务有较大推动的今天，研究网络结构和数据分布对于属性识别的影响。从属性识别的网络结构探索和不同数据分布整合对于属性数据的提高效果。同时结合迁移学习的思想为提高不同环境下人脸属性的识别准确率。

## 1.2 国内外研究现状

### 1.2.1 人脸属性的任务目标发展

从时间角度来看，基于人脸图像的多种人脸属性预测估计在上世纪 90 年代就开始，1990 年，MIT 的 Cottrell 和 Metcalfe 把基于 AutoEncoder 的特征降维用于性别和表情识别<sup>[1]</sup>；1999 年，塞浦路斯学院的 Lanitis 构建了 FGNET 年龄估计数据库（共 82 人，1002 张图像），当时用 PCA 做特征提取<sup>[2]</sup>；2006 年，北卡的 Ricanek 和 Tesafaye 构建了首个大规模年龄、性别、种族数据库 MORPH(1.3 万人，5.5 万图像)<sup>[3]</sup>；2008 年，哥伦比亚大学的 Kumar 等人构建了包含 10 个属性（后在期刊文章里扩展到 60 多个）的大规模名人数据库 PubFig（共 200 人，6 万张图像）仅部分公开，提取了手工设计特征，之后对每个属性训练 SVM<sup>[4]</sup>；2010 年，MIT 的 Pho 等人首次研究了基于普通摄像头的非接触式心率估计，这是“由表及里”的一次突破；2015 年，中科院计算所 VIPL 研究组首次研究了人与机器在属性识别上的性能差异（可控），并发现机器在年龄、性别和种族的识别上已经可以超过人类<sup>[5]</sup>；NIST 组织了年龄和性别预测方面的评测竞赛，并且出了一个报告概括了领域相关工作<sup>[6]</sup>；此外，香港中文大学汤老师组构建了大规模互联网名人的 40 个属性数据集 celeA<sup>[7]</sup>。由此可见，研究工作的时间跨度并非很大，但是各方面工作的丰富性和多样性还是令人瞩目的。

### 1.2.2 人脸属性识别方法变化

从特征的表示方法来看，是一个从全局特征、细节特征到深度特征的过程，具体来讲：全局表观特征：包括 Intensity、PCA<sup>[8]</sup>、BIF 生物启发式特征<sup>[9]</sup>，局部二值模式 LBP (Local binary patterns)<sup>[10]</sup>，加窗傅立叶变换 (Gabor) 等。细节特征如：主动外观模型 AAM (Active Appearance Model)<sup>[11]</sup>，纹理，肤色，人脸形状，sift 特征等。深度学习特征<sup>[12][13]</sup>如 CNN,DNN 中网络的不同层卷积输出。其中是一个不断演变但是也时有结合的过程。从特征分类方法上来看：研究的任务形式也从单任务学习（常用方法：每个属性训练一个分类器）慢慢演变到多标签学习<sup>[7][5]</sup>（回归目标不仅是数，而是向量形式）而后根据不同的细粒度额精确化需求，发展出层级式的分类

器（由粗到细，特别适用于年龄分类，如先确定年龄范围，再进行具体年龄分类）和多任务学习<sup>[14]</sup>（多任务限制玻尔兹曼机，多任务 CNN 等等）。总结来看，是一个从手工设计特征到深度特征、从组合式的学习到端到端学习、从 STL 到 MTL（从单任务学习到多任务学习）的发展过程。

人脸视觉属性学习并不简单，特别是在非可控的真实场景下。影响因素有以下几个方面：传感环境（尤其在室外）的不可控性以及人物的不配合性，这会引起姿态、光照、遮挡等多种因素的影响；属性之间的相关性以及差异性；属性数量的增多引起内存消耗的增加，因此不仅需要高效的模型，而且需要对于不同场景能够行之有效的迁移学习方法。

## 1.3 本文的工作与贡献

### 1.3.1 研究内容

在本文的主要研究的内容有两个，第一项是结合人脸属性的性质探究在人脸属性在深度学习技术下的表现。其中包括人脸属性数据的总结与整理，规划人脸属性的标注类型，单任务模型下的人脸属性的表现，多任务模式下人脸属性的表现等，主要的衡量指标是在不同模型组合和模型策略的情况下人脸属性模型的准确率。另一方面，是针对于现实环境中图片采集的不可控制性，使用对抗生成网络来模拟不同场景的人脸数据，并且探究如何使用迁移学习的思想来对提高人脸属性对于不同场景泛化能力。在这一任务中，除了最终对于人脸属性的准确率提升之外，人脸图片的生成质量也是衡量得指标之一。

### 1.3.2 主要贡献

在这项研究工作之中主要贡献包括研究内容上的工作和一定的工程优化工作，具体如下：

研究上的工作：根据人脸属性任务的性质，基于 Alexnet<sup>[15]</sup> 设计人脸属性的单任务和多任务网络，保证具有一定的可复用性。设计具有网络输出置信评估的模块，增加网络对于自身的输出的感知能力，从而可以更加精确的把握模型的输出准确性。研究对抗生成网络的使用，使用对抗生成网络成功通过噪声模拟出数字，物体和人脸图片，且效果较为逼真。研究对抗生成网络在图像超分辨率领域上的应用，大幅度提升对抗生成网络的图像生成质量。并基于此项技术，在不直接使用 celeA 数据的

情况下，仅使用 lfwA 和对应的超分辨率图像进行训练，提高了模型在 celeA 数据集上的准确率。证明基于超分辨率率的迁移学习是可行的。

工程上的工作：研究如何使用多机多卡的训练，并基于机器学习框架的使用完成了对于人脸属性相关任务的训练，提升了训练和算法迭代的速度。在具体的网络前馈过程之中，使用多线程、指令集等优化方式，提升识别模型输出的速度，包括人脸属性的概率输出和超分辨率的图片生成的速度。

### 1.3.3 论文的组织结构

第二章：笔者主要介绍涉及人脸属性在深度学习技术种一些基本常识和常见的操作和笔者对相关瓶颈操作的一些优化。具体包括：在卷积神经网络的基础操作中介绍所谓卷积操作的多种实现和使用方式介绍，激活函数的具体使用，常见的网络参数初始化方法和网络训练相关细节。在多机多卡的部分介绍，在多卡训练中数据的同步和分发方式，模型参数的更新策略，多机训练中需要注意的一些关键选项配置，以及如何简单的通过机器学习框架完成多机多卡的训练。在网络前馈的优化部分会介绍一些实用性非常强的快速卷积算法，如 im2col+gemm,Winograd 等。对于网络中常见操作的如卷积、BatchNorm 层的合并等。

第三章笔者主要介绍针对于在人脸属性所进行的一些实验和创新的过程的一些相关工作，包括对于人脸属性数据的性质分析、人脸常见数据集的介绍、人脸属性识别中常见的识别方法等。在此基础上总结了三个人脸属性识别所面临的问题：充分利用标签不同的数据库，选择怎样的预处理方式才有助于人脸属性任务的学习，怎样更加精确的把控人脸属性的模型输出。并在提出问题的基础上进行了解答，通过数据集并行训练的方式改进问题一，使用人脸矫正固定输入格式改进问题二，加入网络自评估模块改进问题三。

第四章笔者会介绍如何使用对抗生成网络对于不同场景下的人脸进行学习并且根据噪声生成人脸图片。使用超像素的方式对于人脸图片进行一定程度上的效果增强和场景迁移。通过结合迁移之后的人脸图像进行学习可以方便的改进人脸属性中由于数据分布不同导致准确率下降情况

第五章笔者主要对实验过程做一个综合性的概述并且自我评价一下整个实验过程中出现的问题和解决问题的方法。回顾在解决问题种反应的一些现实层面的现象以及个人对这些现象出现的原因和结果的思考。当然也包含一点关于未来和未解决工作的思索。

## 第二章 卷积神经网络的相关技术介绍

在这一章中主要介绍涉及人脸属性的一些深度学习基本常识如卷积神经网络的基本操作、训练方法等；也会介绍具体的工程实现过程和相关的瓶颈优化如：多机多卡训练的同步训练方式和异步训练方式、网络前馈中对于卷积操作的优化算法，工程中使用向量处理器结合向量指令集完成对于卷积操作的加速等。这些工作都是我整个研究生生涯花费了大量的精力去理解并且思考的，在很多地方也总结了一些看法和规律，在我的研究生科研和工作过程中起到了非常重要的作用。

### 2.1 卷积神经网络的基础操作和训练

卷积神经网络一般是指是针对以共享式多通道卷积操作为代表的一连串数学操作计算组合的一个总称，因为卷积计算是其中的主要计算过程和核心特征提取方式，而计算的过程往往需要加入一步非线性的激活函数用以增加整个计算过程对于非线性过程的模拟，这个过程和人体的神经元结构非常相似，所以从计算科学的角度称为卷积神经网络。由于卷积神经网络具有参数多，训练数据广的特点，难以通过正常的线性代数和微积分求得最优解，一般会使用反向传播算法<sup>[16]</sup> 进行训练，也称梯度下降算法。

在图像识别领域中，卷积神经网络和普通的神经网络，玻尔兹曼感知机等很像：都是把输入数据最后转化成输出；都要使用输入并进行点积运算；都使用可以学习参数的神经元；神经元都含有非线性激活函数；在最后都加入分类的损失函数等等。

而卷积神经网络，借鉴于其常见的多维向量点乘的结构的设计，使其对于图像的 2d 结构具有良好的亲和性，利用这个特点，基于图像的卷积神经网络结构层出不穷，各自的识别效果也几乎是节节攀升。接下来，笔者将从基本网络结构的组成、常用非线性激活函数、常用初始化参数方法、卷积神经网络的训练与优化四个方面来介绍卷积神经网络的相关内容。

#### 2.1.1 卷积神经网络结构的基本组成

层是卷积神经网络结构的基本组成单位，不同的网络结构中都是使用类似或者基础的层来进行搭建完成，少则三、四层，多达成百上千，但无论是深度还是广度的

扩增，都是通过增加层的使用来完成。

### 2.1.1.1 卷积层

卷积层<sup>[15]</sup>是卷积神经网络最重要的层，因为卷积层承担着从图像到高层语义的转化任务，实际使用中也占据了整个网络计算量中的绝大部分。甚至毫不客气地讲，对于卷积层的实现和优化好坏，决定着一个深度学习框架的工作使命和存在意义。（最直观的例子就是最著名的卷积神经网络框架 caffe<sup>[17]</sup>就是因为对于卷积的实现做得好，训练和测试的速度较快，从而取代了 cuda convnet 和 CXXnet 成为了主流，也是目前影响最深的深度学习框架。）

具体来讲：每一个卷积层都会使用 N 个不同参数的卷积滤波器核，每一个卷积滤波器核对整个输入特征图进行类似于滑动窗的卷积操作，由于在这个卷积过程中只是用那一个核的参数，也将常规的卷积操作成为参数共享的卷积神经网络。参数层面：卷积滤波器核超参数包括：

1. N 卷积滤波器核数目
2.  $Kernel_h, Kernel_w$  卷积滤波器核高度和宽度
3.  $Stride_h, Stride_w$  卷积在高、宽维上的步长
4.  $PadH, PadW$  对于高、宽二维上对于 feature map 的空白补足

输入输出参数：

1.  $Coutput, Houtput, Woutput$  分别为输出通道数、输出高度、输出宽度
2.  $Cinput, Hinput, Winput$  分别为输入通道数、输入高度、输入宽度

其中输出参数由如下公式确定：

$$\begin{aligned} Woutput &= \frac{Winput - KernelW}{StrideW} + 1 \\ Houtput &= \frac{Hinput - KernelH}{StrideH} + 1 \\ Coutput &= N \end{aligned} \quad (2-1)$$

卷积层的内部参数包括权重  $W$  和偏置  $b$ ，

1.  $W$  是一个维数是  $Coutput * Cinput * KernelH * KernelW$  多维数组。

2.  $b$  是维数为  $C_{output} * 1$  的数组。

卷积的基本计算公式如下：

$$x_j = f\left(\sum_{i \in (convscale)} x_i * W_i + b_j\right) \quad (2-2)$$

$convscale$  是指对应一次卷积操作中对应的  $C_{output} * kernel_h * kernel_w$  范围内的输入图片像素值集合和对于第  $j$  层的  $W_j$  和  $b_j$ . 在计算机对于卷积的实现操作过程中因为参数和输入的特征都是按照数组的方式进行储存，所以可以简单的认为是对于一定长度的一维数组进行点乘操作，实际上为了减少内存的读取所带来的延时，很多快速算法都实用类似的想法。具体计算的示意图 2-1 如下：

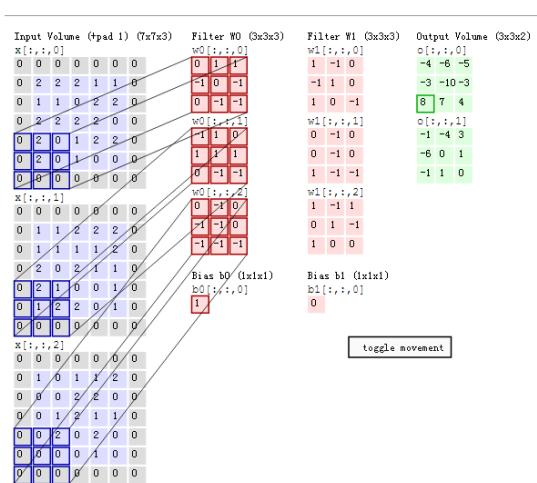


图 2-1 卷积操作的示意图

除此之外，很多不同形式的卷积更加注重对于图像信息提取的卷积形式也被提出，比如为了减少计算量而增加网络深度的  $1 \times 1$  卷积<sup>[18]</sup>，为了扩大感受野而获取更多图像信息的放射卷积<sup>[?]</sup>，为了能够对于图片中具体图像有细节感知的形变卷积<sup>[19]</sup>等，他们都不再局限于对于固定范围内特征值进行卷积，而是着重于在输入特征图上更多具有影响力的范围内进行输入位置的选取。

### 2.1.1.2 池化层

池化（Pooling）层可以相比于信号处理中的采样操作，一般对于输入的特征图片图进行降采样操作，从而起到去除噪声，增加网络的平移不变性和旋转不变性，提升整体的训练效果。事实上池化层有时也被用作上采样，也就是 uppooling，通过输

入的特征图进行等间隔复制的方式得到原尺度多倍的输出，也就是常见的放大操作。但是随着卷积神经网络的发展至今，人们开始主要使用反卷积的方式来实现上采样的操作，以至于 uppooling 的操作慢慢不被人所熟知。池化层的超参数包括：

1. KernelH,KernelW 高宽方向上的池划范围大小
2. StrideH,StrideW 高宽方向上的池化步长

输入输出参数包括：

1. Channel,Height,Width 分别为输入通道数、输入高度、输入宽度
2. Coutput,Houtput,Woutput 分别为输出通道数、输出高度、输出宽度

其中输出参数由如下公式确定：

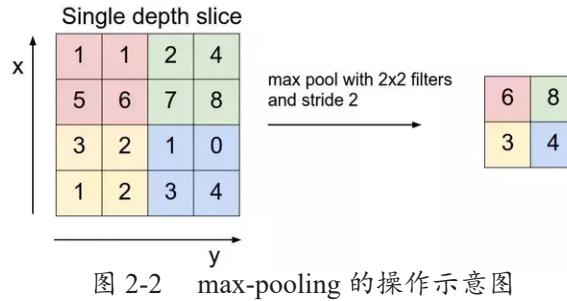
$$\begin{aligned} Woutput &= \frac{Winput - KernelW}{StrideW} + 1 \\ [h]Houtput &= \frac{Hinput - KernelH}{StrideH} + 1 \\ Coutput &= Cinput \end{aligned} \quad (2-3)$$

池化层的计算公式如下：

$$[h]x_j = poolmethod(x_i) \quad i \in (poolscale) \quad (2-4)$$

pool scale 和 conv scale 类似，是指一次池化操作中对应的  $kernel_h * kernel_w$  范围 poolmethod 代表的是具体的下采样函数。通常有三种类型，一种是最大型池化，在 poolscale 中选取值最大的数作为结果的输出；一种是均值型池化，将 poolscale 中的数值求和做平均输出，还有是随机型池化，也就是将 poolscale 中的数值随机选取进行输出。其中最大型池化使用最为广泛，因为其最能够体现池化层对于平移和旋转操作的鲁棒性，而均值池化层主要用在特征归一化操作和最后高位信息的整合上面，以最常见的 max-pooling 操作为例，具体的计算过程如图 2-2 如下：

和卷积层的发展类似，pooling 层也慢慢发展出很多不同的分支，除了前面提到的 up pooling 层之外，在物体检测中存在着使用非常广泛的 roi-pooling<sup>[20]</sup> 和空间金字塔 pooling（也就是 sppnet）<sup>[21]</sup>。



### 2.1.1.3 全连接层

全连接（Fully-connected, FC）层是普通机器学习方法中使用最多的层，包括SVM，随机森林，boosting 种处处可见与之呼应的操作，简单来讲输入的类似于一维向量通过一个矩阵做矩阵乘法运算得到另一个一维向量的过程。

全连接层使用过程中需要设定输出大小即可，其自身的参数是一个输入大小乘以输出大小的矩阵，如果采用带有偏置的算法则还需要一个和输出长度相同数目的偏置项，总结来讲全连接层的参数包括：

1.  $Fc_{i\text{nput}}$  输入大小
2.  $Fc_{o\text{utput}}$  输出大小
3.  $Fc_W$  模型参数  $W$
4.  $Fc_b$  模型参数偏置项

具体的计算过程可以通过公式进行表达：

$$Fc_j = X_i \cdot W_j^T + b_j \quad (2-5)$$

从实现的角度上看，可以比较明显的看出全连接层的整体实现可以通过矩阵乘法来完成，通过和卷积层类似的方法将所有的特征输入扩展成多维的等长向量也就是矩阵然后通过，矩阵乘法来快速的将全连接层的相关问题进行实现。如下图 2-3 是输入为 800，输出为 500 的全连接层示意图。

但是需要注意的是，在使用过程中，全连接层的的参数量是  $n^2$  也就是说全连接层很容易出现参数量过大的问题。比较著名的案例就是 VGGnet<sup>[22]</sup> 中最后的三个全连接层层，几乎占据了整个网络一半的参数量，实际中完全可以使用  $1\times 1$  卷积<sup>[18]</sup>，global pooling<sup>[18]</sup> 等类似的操作对其进行替代以减少参数量。但是在最新的研究中发

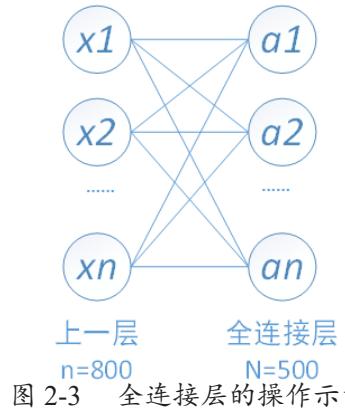


图 2-3 全连接层的操作示意图

现，全链接层可在模型表示能力迁移过程中充当“防火墙”的作用。具体来讲，假设在一个数据库中上预训练得到的模型为，则该数据库可视为源域（迁移学习中的 source domain）。微调（finetuning）是深度学习领域最常用的迁移学习技术。针对微调，若目标域（target domain）中的图像与源域中图像差异巨大（如相比 ImageNet，目标域图像不是物体为中心的图像，而是风景照），不含全连接层的网络微调后的结果要差于含全连接层的网络。因此全连接层可视作模型表示能力的“防火墙”，特别是在源域与目标域差异较大的情况下，全连接层可保持较大的模型 capacity 从而保证模型表示能力的迁移。也就是说，冗余的参数并不无是处<sup>[23]</sup>。

#### 2.1.1.4 归一化层

归一化层指的是对每一层的输出进行标准化操作，使得下一层的输入保持一个较为稳定的分布。常用的标准化层有局部区域归一化（Local Region Normalization, LRN）层，批量标准化（Batch Nomalization, BN）<sup>[24]</sup> 层等。这里着重介绍一下 BatchNormalization 层：

BatchNormalization 也就是批量规范化，在网络训练时，对每个 minibatch 的特征在卷积之后做一次规范化，使得其输出的特征符合均值为 0 和方差为 1 的概率分布，与此同时再在其后面加入一个偏移量和一个尺度信息，用于将标准化的分别重新映射到对应层学习到的尺度空间中。实际上，BN 层的引入本质上是为了使得每一个卷积层的输入数据的分布统一化，这样做有利于网络训练，另外一个主要的原因则是防止梯度消失和梯度爆炸，通过将输入归一化到固定的均值和方差，使得原本尺度特别大或者特别小的特征，在后馈计算时有效的均衡其梯度的大小，从而很好的防止梯度弥散和爆炸。

BN 层的参数包括 global status（使用网络参数中的均值方差还是根据输入数据

重新计算)、moving average fraction (每次计算的累加方式)、eps (为了防止方差为了 0 加入的偏置项)，具体 BN 层的操作过程可以参考图 2-4。

<b>Input:</b> Values of $x$ over a mini-batch: $\mathcal{B} = \{x_1 \dots m\}$ ;
Parameters to be learned: $\gamma, \beta$
<b>Output:</b> $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$
$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$ // mini-batch mean
$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2$ // mini-batch variance
$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$ // normalize
$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i)$ // scale and shift

图 2-4 BN 层的操作说明

BN 的提出在神经网络的发展过程中具有非常重要的意义，大大改善了神经网络的收敛问题，也可以说从某种程度上改进了神经网络的整体效果。

#### 2.1.1.5 损失函数 loss 层

损失函数是神经网络中用来衡量网络预测和真实值之间的误差情况，最常用的决策层损失函数是：Softmax 损失函数和欧几里得损失函数。

Softmax 损失函数主要用于多分类任务，其具体的损失函数表达为：

$$l(y, z) = -\log \left( \frac{e^z}{\sum_{j=1}^m e^z_j} \right) \quad (2-6)$$

其中  $m$  表示分类的类别总的数目， $y$  表示标签， $z$  表示网络预测的类别。也就是说将所有类别的预测值取他们的指数值求和，然后判断实际标签中样本在其中所占比重，并将其取  $\log$  作为 loss 函数和优化的值。

欧几里得损失函数主要用于回归任务，具体的回归损失函数如下：

$$l(y, z) = (z - y)^2 \quad (2-7)$$

其中  $y, z$  同 softmaxloss 的含义相同。

除了这两种常用的 loss 函数之外，还有比如交叉熵损失函数，smooth L1<sup>[25]</sup> 损失函数等，都是在分类和检测中经常使用的损失函数。

### 2.1.2 卷积神经网络常用激活函数

卷积输出之后通常会使用激活函数进行非线性激活，从而增强网络的模拟变换能力，不然只是线性变化的组合可以涵盖的空间非常有限。图 2-5 总结了神经网络中经常使用的激活函数：

Name	Formula	Time
<b>sigmoid</b>	$y = 1/(1 + e^{-x})$	1986
<b>tanh</b>	$y = (e^{2x} - 1)/(e^{2x} + 1)$	1986
<b>ReLU</b>	$y = \max(0, x)$	2010
<b>SoftPlus</b>	$y = \ln(e^x + 1) - \ln 2$	2011
<b>LReLU</b>	$y = \max(x, \alpha x), \alpha \approx 0.01$	2011
<b>maxout</b>	$y = \max(W_1x + b_1, W_2x + b_2)$	2013
<b>APL</b>	$y = \max(0, x) + \sum_{s=1}^S a_i^s \max(0, -x + b_i^s)$	2014
<b>VLReLU</b>	$y = \max(x, \alpha x), \alpha \in 0.1, 0.5$	2014
<b>RReLU</b>	$y = \max(x, \alpha x), \alpha = \text{random}(0.1, 0.5)$	2015
<b>PReLU</b>	$y = \max(x, \alpha x), \alpha \text{ is learnable}$	2015
<b>ELU</b>	$y = x, \text{if } x \geq 0, \text{else } \alpha(e^x - 1)$	2015

图 2-5 激活函数的具体表达式以及出现时间

其中比较重要的是 sigmoid 函数、relu 函数、以及 prelu 函数，下面是他们的函数曲线图：

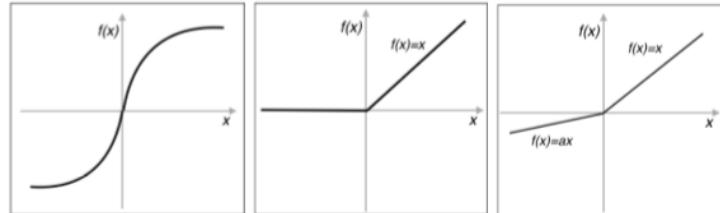


图 2-6 sigmoid(a)、relu(b)、prelu(c) 函数的函数曲线示意图

#### 2.1.2.1 sigmoid 函数

sigmoid 函数是神经网络最早使用的激活函数，从函数的特性上可以看到其能把输出映射到区间 (0,1)：若输入趋于负无穷，则趋近于 0；若输入趋于正无穷，则输出趋近于 1。近年来由于梯度下降法的使用增加，sigmoid 在数值较大的情况下，导数趋近于 0，这样导致的后果则是对应的梯度也会慢慢消失，导致训练的过程变得缓慢难以得到正常的收敛效果。

### 2.1.2.2 relu 函数

RELU (Rectified Linear Unit) 激活函数，中文也称修正式线性激活函数。由于自身的非饱和特性，修正式线性单元极大程度的加速了深度卷积网络的收敛。但是修正式线性单元也存在一个很大的问题。在训练的时候，修正式线性单元比较脆弱并且可能“死亡”。举个例子来说，当一个很大的梯度流过修正式线性单元的神经元的时候，可能会导致梯度更新到一种特别的状态，在这种状态下神经元将无法被其他任何数据点再次激活。如果这种情况发生，那么从此所以流过这个神经元的梯度将都变成 0。也就是说，这个单元在训练中将不可逆转的死亡，而这样会导致数据多样化的丢失。

### 2.1.2.3 prelu 函数

prelu 函数在负轴上加入了固定大小斜率的  $a$ , 从而确保梯度不会因为突然死亡的问题而导致网络崩溃。

除此之外，tanh 是作为 sigmoid 函数在 (-1, 1) 域上的扩展，ELU 函数是 RELU 和 prelu 函数在负值域上的变换斜率的优化方式。Maxout 是使用更加粗暴的阶段方式对于神经网络进行截取来获得非线性的激活函数。

## 2.1.3 卷积神经网络常用的参数初始化方法

神经网络求解的是局部最小值，一个好的参数初始化方法能使得卷积神经网络收敛且收敛的更快。常用的卷积神经网络初始化方法有如下几种。

1. 常数初始化: 使用固定的常数初始化每个参数，常用来初始化的常数一般比较小，通常为 0。常数初始化方法通常对偏置项所使用。
2. 均匀分布初始化: 假设参数服从在区间  $[l, h]$  上的均匀分布，进而为参数进行初始化。通常为权重参数使用。均匀初始化方法中比较常见的是在 2010 年提出的 Xavier<sup>[26]</sup> 方法，Xavier 初始化方法能够使得每一层的输出方差尽量相等，从而让网络中的信息更好的流动。具体表达式为：

$$W = U \left[ -\sqrt{\frac{6}{n_{input} + n_{output}}}, \sqrt{\frac{6}{n_{input} + n_{output}}} \right] \quad (2-8)$$

3. 高斯分布初始化：为 0 方差的高斯分布， $\sigma$  为参数进行初始化。通常为权重参数使用。 $\sigma$  可以是人为制定也可以是通过输入输出计算得到。

比较流行的高斯分布初始化的方法是 MSRA<sup>[27]</sup>, 由微软亚研院提出, 具体公式如下:

$$W \sim N \left[ 0, \sqrt{\frac{4}{n_{input} + n_{output}}} \right] \quad (2-9)$$

实验证明, 对于较深的卷积神经网络, MSRA 初始化方法比 Xavier 初始化方法更容易收敛。

4. gabor 初始化<sup>[28]</sup>:gabor 初始化的方法是根据 gabar filter 的参数直接作为神经网路的参数, 一般在网络的第一层进行使用。由于其具有天然的图像滤波特性, 所以很多时候可以固定参数。不对其就行学习, 从而达到减轻训练时间和负担的作用, 从下图中可以看到针对于不同的模式识别任务使用 gabor 滤波器都可以降低 31-35

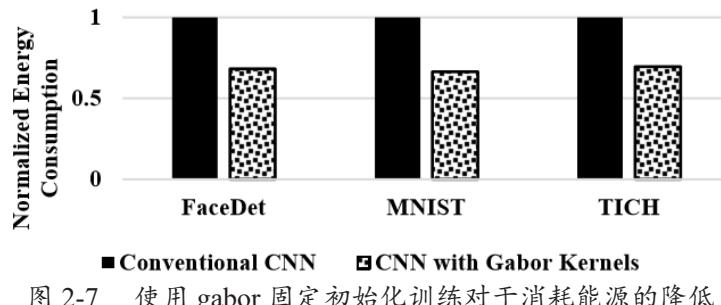


图 2-7 使用 gabor 固定初始化训练对于消耗能源的降低

#### 2.1.4 卷积神经网络的训练与优化

通常对于包含  $N$  个数据的数据集, 优化的损失函数可以写成:

$$J(W, b) = \frac{1}{N} \sum_{i=1}^N l(y_i, z) + \lambda \Phi(W) \quad (2-10)$$

以具有动量的 SGD 梯度下降方法为例:

$$\begin{aligned} V_{t+1} &= \mu V_t - \alpha \bigtriangledown l(W_t) \\ W_{t+1} &= W_t + V_{t+1} \end{aligned} \quad (2-11)$$

其中  $t+1$  是迭代的当前轮数,  $W$  是需要更新的参数,  $\alpha \bigtriangledown l(W_t)$  是目标损失函数对于  $W_t$  的偏导数,  $V_t$  是上一次参数的更新量,  $V_{t+1}$  是本次的参数更新量,  $\mu$  是动量值,  $\alpha$  是学习率。

卷积神经网络的训练主要使用基于梯度的反向传播（Backpropagation, BP）<sup>[16]</sup> 算法。假设卷积神经网络一共有  $N$  层，记作  $L_1 \dots L_n$ ， $y$  代表样本标签， $z$  代表每一层的输出， $a$  代表每一层输出的激活值， $\delta$  代表每一层传回的梯度值， $W$  为权重， $b$  为偏置项，则 BP 算法步骤如下：

1. 进行前馈运算，利用前向传导公式，得到  $L_1 \dots L_n$  的激活值
2. 对  $L_N$  层，计算损失函数对应的偏导值
3. 对于第  $i$  层  $L_i, i = N1, \dots, 2$ ，计算输出的梯度：
4. 依次计算每一层参数的梯度
5. 利用更新的公式更新参数值。

## 2.2 神经网络训练速度的提升

在大型数据集上进行训练的现代神经网络架构可以跨广泛的多种领域获取可观的结果，领域涵盖从语音和图像认知、自然语言处理、到业界关注的诸如欺诈检测和推荐系统这样的应用等各个方面。但是训练这些神经网络模型在计算上有严格要求。尽管近些年来 GPU<sup>[29]</sup> 硬件、网络架构和训练方法上均取得了重大的进步，但事实是在单一机器上，网络训练所需要的时间仍然长得不切实际。幸运的是，我们不仅限于单个机器：大量工作和研究已经使有效的神经网络分布式训练成为了可能。多机多卡训练也可以理解为分布式训练，但是由于传统的分布式训练主要基于 cpu 进行计算。而现代深度学习的分布式框架学习，gpu 的使用是其中不得不实现的一个部分，所以多机多卡的形容更加贴切。

关于多机多卡的训练策略和普通的单机训练相面临更多的挑战，作为研究生生涯中非常感兴趣的一部分，也是整个神经网络实验的基础研究部分，将我对于多机多卡的一些调研和感悟简单做介绍：

### 2.2.1 并行模式

多机多卡训练一般有两种模式：数据并行和模型并行。具体的示意如图 2-8 所示。

模型并行（model parallelism），在分布式系统中的不同机器分别负责在单个网络的不同部分计算——例如每层神经网络可能会被分配到不同的机器。

数据并行 (DataParallelism)，不同的机器有着整个模型的完全拷贝；每个机器只获得整个数据的不同部分。计算的结果通过某些方法结合起来。

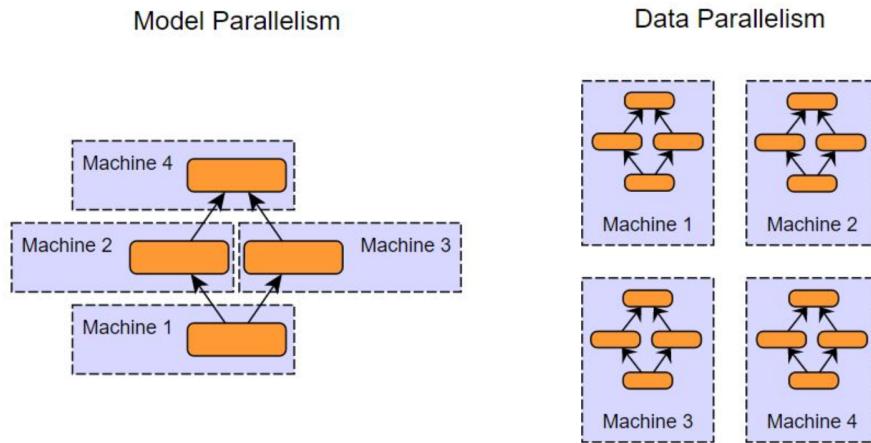


图 2-8 模型并行与数据并行示意图

当然，这些方法并不是互相排斥的。想象一个多 GPU 系统的集群，我们可以对每个机器使用模型并行（将模型分拆到各个 GPU 中），并在机器间进行数据并行。尽管在实践中模型并行可以取得良好的效果，但数据并行毫无争议是分布式系统中最适的方法，而且也一直是更多研究的焦点。实现性、容错性和好的集群利用率让数据并行比模型并行更加简单。分布式训练中的数据并行方法在每一个分布机器上都有一套完整的模型，但分别对训练数据集的不同子集进行处理。数据并行训练方法均需要一些整合结果和在各工作器（worker）间同步模型参数的方法。

## 2.2.2 参数更新方式

对应着数据并行训练方式，如何对于参数进行更新也就成了非常关键的问题，目前比较主流的参数更新方法有两种：参数平均化的更新方式，异步随机梯度下降的方式。

**参数平均化更新方式：**参数平均化是概念上最为简单的数据并行方法。使用参数平均时，训练按照如下方式执行：

1. 根据模型配置随机初始化网络参数
2. 将现有的参数的一个副本分配给每一个 worker machine
3. 在该数据的一个子集上对每一个 worker 进行训练

4. 从每一个 worker 的平均参数上设立一个全局参数
5. 当还需要处理更多数据时, 回到第 2 步

同参数平均化相似的方法是: 基于更新的数据并行化 (update based data parallelism)。两者的基本区别在于: 不会将参数从 worker 传递给参数服务器, 而是传递更新 (例如: 梯度柱型的学习率和动量 (gradients post learning rate and momentum))。当我们放松同步更新的要求时, 基于更新的数据并行变得越来越有趣 (毫无疑问它更有用)。即在更新参数的变化值被计算的时候就应用于参数向量 (而不是等待所有 worker 的  $N \geq 1$  次迭代)。这就催生了异步随机梯度下降算法

**异步随机梯度下降:** 异步的随机梯度下降故顾名思义, 根据一定的算法对于多个 machine 的更新的过程进行权衡处理, 但是保持训练机器的训练过程不中断, 可以预见异步随机梯度下降有两个主要优点: 首先, 潜在可能在整个分布式系统中获得更高的通量: worker 可以将更多时间花在执行有用的计算而不是等待参数平均化步骤完成。其次, worker 有可能可以集成来自其它 worker 的信息 (参数更新), 这比使用同步 (每  $N$  个步骤) 更新更快通过在参数向量中引入异步更新, 也引入了一个新问题, 也就是过期梯度问题 (stale gradient problem)。过期梯度问题的产生是因为梯度 (更新) 的计算需要时间, 在一个 worker 完成这些计算并将结果应用于全局参数向量前, 这些参数可能已经更新过许多次了, 所以如何设定这些更新的策略也是现在多机多卡的研究热点所在。

### 2.2.3 基于机器学习框架的多机多卡训练

#### 2.2.3.1 caffe 中的多机多卡

caffe<sup>[17]</sup> 是由贾扬清开发的轻量级 C++ 深度学习框架, 因为其出现时间早, 计算速度快等优势被 Alex 用于训练 imagenet 并且大获成功, 从而大获成功, 是影响比较大的学习框架之一。作为用户, 在 caffe 中实现多卡训练比较简单, 主要在命令行中设置参数 `-gpu gpu_ID` 就可以选择希望占用的 GPU。在综合了代码和测试效果来看, caffe 是采用了数据并行, 参数平均化更新的策略, 实际测试中也可以很明显的看到默认 0 卡的显存占用更大, 而且存在一定的间隔内, 显卡中的使用率是会有明显空缺的, 除去数据读取的原因之外, 参数同步也是其中重要的一部分原因。

在 nVidia 开源的 nVidia-caffe 中, 则大胆的采用了异步的 SGD 参数更新方式, 可以很明显看到不仅训练的速度快, 而且由于其训练和测试都被分摊到了单独的显卡,

而不是都在 0 卡上进行同步，导致其训练的所占用的显存要比普通的 `caffe` 版本更加小很多，但是所带来的问题也同样明显，在训练的过程中可以明显看出其收敛的稳定性不够出色，同样的训练参数和训练数据在普通的 `caffe` 中训练，可以收敛 loss 曲线也非常平稳，但是 `nVidia-caffe` 中却迟迟不能收敛，而且网络的训练也不够稳定，易出现崩坏的情况。

再来看基于分布式的多机多卡训练，在 `intel/caffe` 中提供了多机多卡的训练方式，使用的方式稍微复杂，需要安装 `ansible`，然后在用来训练的机器上配置好主机的 SSH 公钥验证，然后使用 `ansible` 统一安装 `caffe` 所需要的软件同时编译 `caffe`，设置好同步的文件夹，然后就可以选择需要的训练配置文件和数据就可以了。使用 `mpi` 命令配合执行 `caffe train` 命令就可以实现多机分布式训练网络。

综合来看，从工程上来讲，目前的分布式的多机训练其实是建立以往分布式训练的基础之上，有很强的 `spark` 和 `Hadoop` 烙印，对于多卡训练，大多数的实现都是基于 `nvidia-NCCL` 的多 GPU 通信库<sup>[30]</sup>。`NCCL` 是 Nvidia Collective multi-GPU Communication Library 的简称，它是一个实现多 GPU 的 collective communication 通信（`all-gather, reduce, broadcast`）库，Nvidia 做了很多优化，以在 `PCIe`、`Nvlink`、`InfiniBand` 上实现较高的通信速度。

而从算法上来看，无论是同步训练还是异步训练其实都有其各自的挑战，对于同步训练来讲，虽然一定程度上加速了网络的训练速度，但神经网络在 `batch` 数目较大情况下的优化其实不是一帆风顺，而且随着 `cluster` 的数目增加，同步时延会成为性能关键，树形和环形拓扑都会成为其下一步的改进方向。

实际上异步算法更加受到人们的期待，就是因为其一定程度上摆脱了同步时延限制，能够实现了硬件性能的线性加速。从其他机器学习的时间和发展过程来说，如果深度学习的使用得到广泛的应用，那么异步算法的优化就会是大势所趋，因为对于实际应用的成本压缩和人们的对于性能的追求决定了算法的方向。

### 2.3 神经网络前馈速度优化

网络前馈又被称为网络的 `inference`，一般是指经过一定的数据训练，对于输入数据和输出结果具有一定正向的科学计算过程，可能这样的说法不是很准确，因为很多时候为了获得更为理想的判断结果，往往会采取人工检查和机器过滤的两种方式进行结合，那么在我的工作之中主要是对于机器过滤中对于所需要使用的科学计算过程所进行的一些速度上的优化。具体算法的表述形式可以参照 2.1 节中的神经网

络的基础算法过程进行对比。

### 2.3.1 卷积计算的优化方式

对于卷积层的优化有三个方向，第一种是对于目前的算法下，通过编程技巧减少内存瓶颈和增加缓存命中率的方式直接对卷积的过程进行优化；第二种相对比较直接，是直接使用包装成熟的第三方的加速库如 blas, FFT 等对卷积计算进行加速，而实际上这些库使用的加速方法和第一步是类似的，但因为类似的加速库都会从汇编级指令进行不必要的操作的剪除所以效率上会有很大的改进；第三种是结合数据结构和矩阵理论的算法对于卷积的操作进行创新和改良。在这三个方向中，第一种适合大部分操作系统平台 Windows/Mac/Linux/Unix/Arch 等，和 CPU 计算平台，如 intel/AMD/ARM/PpwerPC/龙芯等，相对来说扩展性和移植性都非常的出色，适合具有一定开发能力的团队核心开发和使用。第二种方法主要会依赖于计算库的特性，具有一定的适用范围，包括操作系统和 CPU 环境等。第三种方法则一般局限于理论研究，通过以减少算法复杂度的方式来减少卷积的计算量，也是理想情况下加速卷积速度的最佳选择，但在实际计算机内存和缓存架构上并不一定可以取得良好的效果。

#### 2.3.1.1 通过编程优化卷积速度

**CPU 直接计算卷积：**正如之前的基础知识中介绍的，普通的卷积操作需要对于输入特征中的 N, C, H, W 四维数据中进行提取，然后和卷积参数中 Coutput,Cinput,kernelH,KernelW 各个维数中的参数进行分别点乘，通常的写法使用 for 循环进行完成的话，需要使用 7 个 for 循环来完成，具体操作过程如图 2-9：这种粗暴的实现方式会使用  $N*Cinpu*Coutpu*Woutpu*Houtpu*KernelW*kernelH$  次乘法和加法操作，并且读取相同次数的内存。所以有很多优化的可能性。

首先通过选择合适的特征图存取方式减少内存读取费用：普通的特征图计算，按照图片个数、通道数、特征图长、特征图宽的顺序存储；但是在实际读取过程中，对于通道数这一层是读取负载最重的，应该将不同通道之间的读取负载降低。于是内存存储上使用普通的特征图计算，按照图片个数、特征图长、特征图宽、通道数的顺序存储会更加合理。

其次通过使用 SIMD 指令集和并行向量处理机的提高计算速度：向量处理器，又称数组处理器，是一种实现了直接操作一维数组（向量）指令集的中央处理器（CPU）。并行向量处理机一个最大的优点是它能够允许软件传递大量的并行任务给

---

## Algorithm 1 Forward Propagation

---

```

1: for  $i_0 \in 0, \dots, minibatch$  do
2:   for  $i_1 \in 0, \dots, ifm$  do
3:     for  $i_2 \in 0, \dots, ofm$  do
4:       for  $i_3 \in 0, \dots, out_h$  do
5:         for  $i_4 \in 0, \dots, out_w$  do
6:           for  $i_5 \in 0, \dots, k_h$  do
7:             for  $i_6 \in 0, \dots, k_w$  do
8:                $output[i_0, i_1, i_3, i_4] +=$ 
9:                $input[i_0, i_1, i_3 * s + i_5 - 1, i_4 * s + i_6 -$ 
                 $1] * wts[i_1, i_2, i_5, i_6]$ 

```

---

图 2-9 使用 7 层 FOR 循环实现卷积操作

硬件。

向量处理机所配合使用的指令集主要是指 SIMD (Single Instruction Multiple Data) 顾名思义就是“一条指令处理多个数据（一般是以 2 为底的指数数量）”的并行处理技术，相比于单个指令处理单个数据的方式，运算速度将会大大提高。而只需要一条很短的指令即可。在 SSE4 指令集中可以一次进行 4 次乘法操作，而在 AVX512 指令集中，可以一次完成 16 个 float 的乘法。结合这一思想，可以有效加速对于卷积中的乘法操作。尤其在移动端的优化，

### 2.3.1.2 卷积的快速算法

im2col+GEMM<sup>[17]</sup>: 卷积算法优化中最常见的卷积快速算法是 imcol+gemm,(gemm 是矩阵乘法的简称) , 在卷积层的介绍中，可以看出笔者把卷积的每一个输出值都用多个输入数据和卷积层参数值相称的求和的方式表示，也就是向量积，一共需要做  $Coutput * Houtput * Woutput$  次这样的向量乘法，而且每次向量乘法的维数都一致，所以可以通过矩阵乘法来实现相关操作，具体的操作可以参见图 2-10。

Winograd 卷积算法<sup>[31]</sup>: 得益于 Shmuel Winograd<sup>[32]</sup> 之前对于计算机超算的工作，可以使用线性代数分解的方式将一些固定卷积核尺寸大小卷积操作如 2x2,3x3 等，分解成多个具有固定参数的小矩阵相乘的方式，从而大幅度提升了卷积的速度。具体操作过程可以参考图 2-11。

MEC<sup>[33]</sup>: im2col+gemm 的改进版，在减少内存的同时顺便可以提升一些速度。

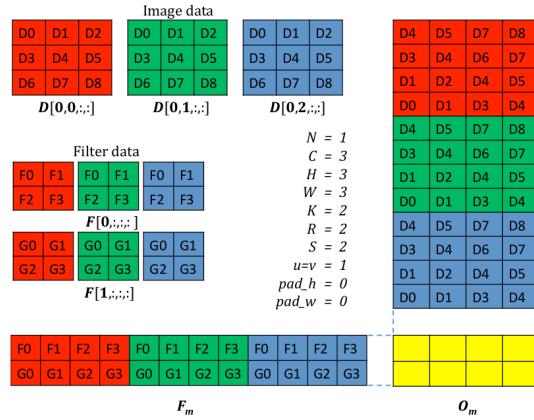


图 2-10 im2col 的操作示意图

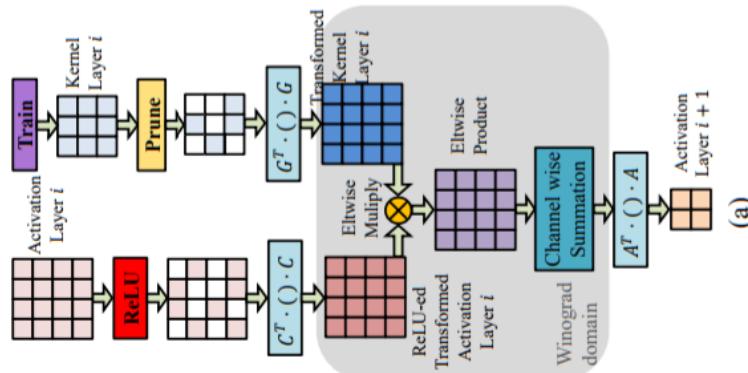


图 2-11 winograd 算法的算法过程

具体操作可以参考图 2-12。

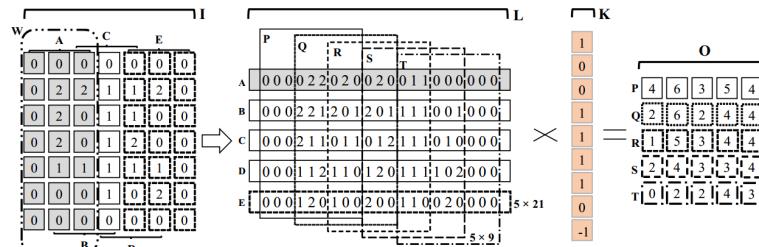


图 2-12 MEC 算法的算法过程

### 2.3.1.3 借助第三方的计算库对于卷积计算进行优化

得益于开源代码的普及，很多专注于计算机超算的公司和组织都为神经网络推出了相关的加速库，在这些库的使用过程中，可以明显感受到现代工业计算机产业化所带来的强大超算力量，很可能一个简单的链接改变，就可以带来几倍甚至十几倍

的速度改观。当然在使用第三方加速库的同时就面临着不同场景下的匹配问题，这也一定程度上考察了从业者的开发能力。

1. 在使用 im2col+GEMM 的过程中，矩阵乘法的计算是主要的计算过程，而在世纪的工程开发过程当中，矩阵乘法可以借助 openblas、MKL、altblas 等的线性代数优化库，可以有效地提升时间。
2. 使用 MKL 和 MKLDNN<sup>[34]</sup> 中的卷积层实现，用于深度神经网络的英特尔数学内核库（MKLDNN）是用于加速 intel 体系结构上的深度学习框架的深度学习应用程序的开源性能库。intel MKL-DNN 包括高度矢量化和线程化构建模块，用于实现具有 C 和 C++ 接口的卷积神经网络（CNN）。

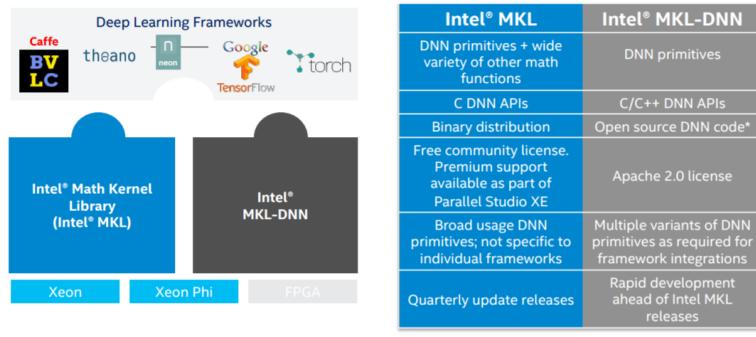


图 2-13 MKL/MKL-DNN 的简介

3. 使用 NNPACK<sup>[35]</sup> 中的 FFT 卷积操作，NNPACK 也是神经网络计算的加速包，基于傅立叶变换和 Winograd 变换的快速卷积算法，优势在于没有附加的依赖库，非常适合于移动端的开发。

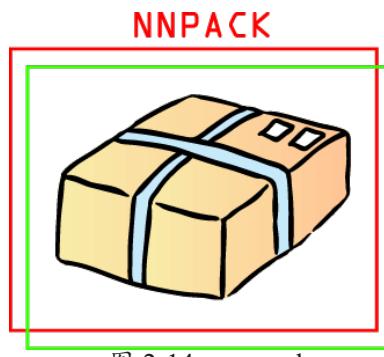


图 2-14 nnpack

4. 使用 CUDNN<sup>[36]</sup> 和 TensorRT 中的卷积实现，cudnn 和 tensorRT 是使用在 gpu 上的算法加速库，都是 nVidia 为了加速神经网络而开发的闭源库，但是可以通过下载现有的公开库进行使用，cudnn 和 tensorrt 是所有加速库中能够获得加速比最高选择，原因在于其对于 GPU 的出色使用。



图 2-15 CUDNN

### 2.3.2 不同网络层的合并

事实上，随着神经网络的不断发展，卷积神经网络的层的种类其实是不断扩充的，有很多在随后的岁月中被发明并且广泛的使用，如 BatchNorm 层、relu 层等等，这些网络层在图像识别的发展过程中都起到了非常关键的作用，也极大加速了网络训练的收敛速度和预测效果，但是在工程的生产实践过程之中，很多网络结构在 inference 的过程中显得非常冗余，也就是说他们可以和其他的操作进行合并。

1. scale 层和 Batch norm 层的合并，在 BN 层的介绍之中，在 BatchNorm 层将输入的特征分布转换均值为 0，方差为 1 的同分布之后，还需要连接一个 scale 层，让网络重新学习数据的分布，这在训练的过程中，确实确实很有效。但是在前馈的过程中，可以把 scale 层的参数和 batchNorm 中的除方差一步结合起来，从而减少计算量和多余的内存使用。
2. BN 层和卷积层的合并，接着上面的优化方向，既然 scale 层可以和 BatchNorm 层相互合并，那么同样的道理，我们将 batchnorm 层中储存的方差的值和卷积层中的网络参数值相互合并，将 batchNorm 中的 bias 除以方差后和卷积中的 bias 合并，就完全可以在卷积层一层的实现过程中完成所有的计算，而不用再次使用 batchnorm 层。
3. 卷积层和 relu 层的合并，relu 层实质上就是一个符号函数，在卷积完成之后，使用使用一个符号函数就可以避免为 relu 层重新新建层。

### 2.3.3 本章小结

本章中我们介绍了卷积识别网络中的基本结构和组成包括卷积层, pooling 层、全链接层、激活函数、初始化方式、训练方法等。。同时也介绍了神经网络的加速训练的方法, 即多机多卡的训练方式。为了能够加速神经网络的在现实环境中的使用, 同时介绍了神经网络种常见的加速方法。包括卷积层的快速算法和不同网络层的合并等。

总结来讲, 这一章的主要内容是作为卷积神经网络的基础。神经网络中训练和测试中的速度优化是具体识别系统在现实中的应用也是算法发展快速迭代的根本。

在接下了的章节中会有很多细节都使用了本章中的技术。可以从图 2-26 中简单了解一下具体不同化方式速度对比。

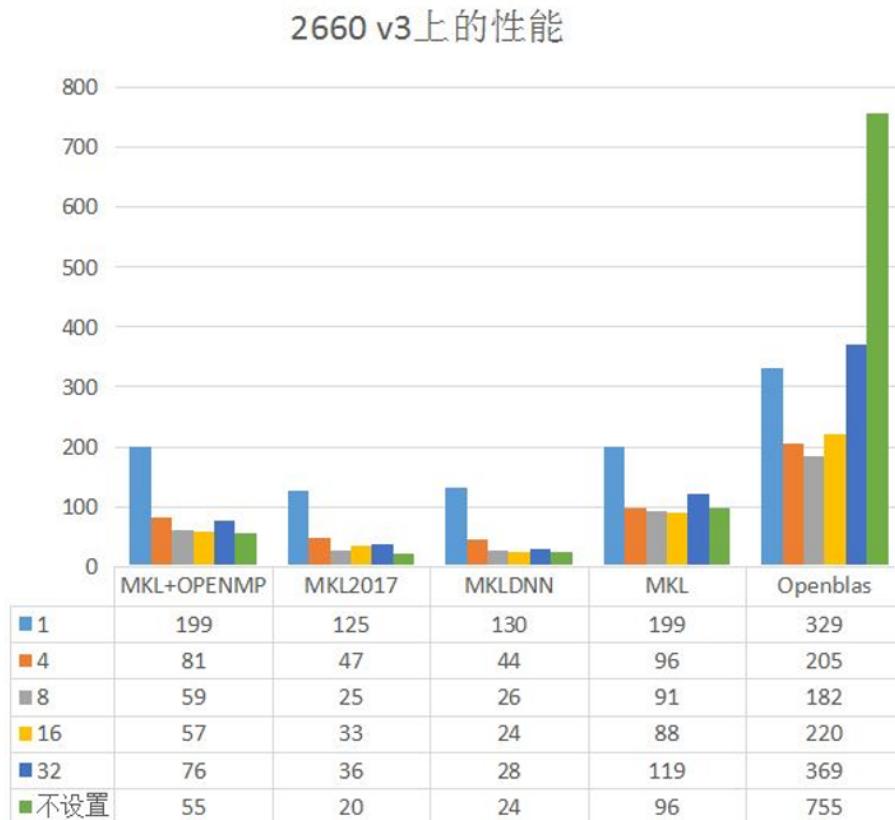


图 2-16 MKL、MKL2017、MKLDNN、openblas 加速方法的具体速度

## 第三章 人脸多属性属性识别的架构

本章主要介绍人脸属性识别任务数据库和一些人脸属性识别中常用的方法，并且根据这些方法的弱点和问题，提出改进方法改进并进行实验。包括人脸属性中输入图片对于识别效果的影响，改进网络结构对于属性性能的提升，设计面向多数据分布和多属性分类的神经网络框架、以及对于网络输出置信度模块的构建。

### 3.1 人脸属性性质分析

#### 3.1.1 人脸属性的类别

人脸属性的数据标注的环境各有不同限制性场景和非限制性场景（如固定摄像头拍摄和日常采集的场景），其中标签往往具有很多种表示和性质，比如相对性标注和绝对属性标注（如颜值数据标注之间只有相互的高低，但没有绝对的属性标签）但总体分为有序性与无序性，整体性与局部性等<sup>[37]</sup>，具体包括：

1. 无序性：无序性的属性有两个或两个以上的类别（值），但在类别之间没有内在的顺序。例如，种族是具有多个类别的名义属性，例如黑色，白色，亚洲等，并且这些值（类别）没有内在排序。；
2. 有序性：有序性的属性具有明确的变量排序。例如，一个人的年龄，通常从0到100，是不平均的。（实际上，年龄不仅是相互独立的存在，在不同的年龄标签中，具有一定钟形的分布）
3. 整体性：整体性标签描述了整个人脸的特征，诸如年龄，性别，种族等；
4. 局部性：和整体性标签相反，局部性描述了部分人脸的特征，例如：尖鼻子，大嘴唇等。

本文中也主要根据上面的人脸属性的性质来设计网络和分析问题。

#### 3.1.2 多属性标签表示形式

在训练的过程中，人脸属性通常以分类或者回归问题的形式出现，但是在多属性识别的任务中，通常使用标签编码或者多标签回归的方式。

方法一：标签编码：将多属性标签组合进行编码（比如，将一岁亚洲男性标记为 001，将一岁非洲男性标记为 002 等），将多属性问题转化为分类编码问题，也就是单一属性。

方法二：多标签回归通过回归的方法，使预测的特征向量与 Groud-truth 属性向量的损失越来越小，二者趋向接近，由此得到预测的特征向量。

### 3.1.3 属性之间的相互联系

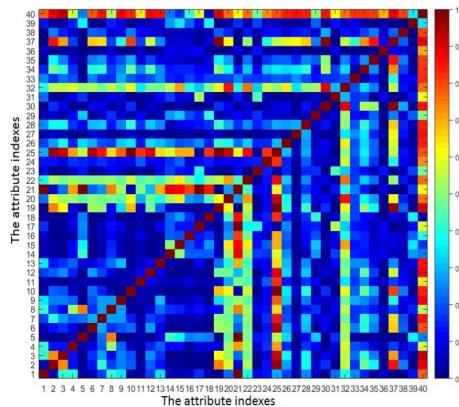


图 3-1 人脸属性之间的相互联系

正如上文提到的，很多人脸属性其实并没有关系，具有非常大的异构性，比如比如头发长短和是否微笑是没有必然关系的，而年龄是可量化的，而种族是类别化的，在表达方式上就是不一样的，这就需要不同的处理方式，也是为什么要把人脸属性作为一个 multi-task 的场景来处理的基本原因。

但是作为人脸特征，它们同时在很多表现过程之中，也有很多共同的地方。通过对 CelebA 数据集的 40 个属性做了成对的 Co-occurrence 计算，如图 3-1 所示，它揭示了属性的相关性是普遍存在的，比如性别男女和是否有胡子和头发长短是具有很强的相关性，我们认为它对属性学习有所帮助。因为属性之间具有相关性也就意味着所使用的表示特征也具有一定的相关性，而且很可能处在同一个线性空间中，具有很高的复用价值。所以完全有理由对于不同的任务之间使用共享特征作为识别的基础。为了利用属性之间的相关性，包括正相关和负相关等来进行互相补足；在设计的过程中我们更倾向于用的是单一主网络多任务自网络方式。

表 3-1 celeA 中的属性表

属性序号	属性	属性序号	属性
1	5OClockShadow	21	GrayHair
2	Male	22	Sideburns
3	ArchedEyebrows	23	BigLips
4	MouthSlightlyOpen	24	Smiling
5	BushyEyebrows	25	BigNose
6	Mustache	26	StraightHair
7	Attractive	27	Blurry
8	NarrowEyes	28	WavyHair
9	BagsUnderEyes	29	Chubby
10	NoBeard	30	WearEarrings
11	Bald	31	DoubleChin
12	OvalFace	32	WearHat
13	Bangs	33	Eyeglasses
14	PaleSkin	34	WearLipstick
15	BlackHair	35	Goatee
16	PointyNose	36	WearNecklace
17	BlondHair	37	HeavyMakeup
18	RecedingHairline	38	WearNecktie
19	BrownHair	39	HighCheekbones
20	RosyCheeks	40	Young

## 3.2 人脸属性数据库简介

这一章主要对于对于具体的数据库进行介绍：

**MOROH II:** MORPH 是一个大型的 mugshot 图像数据库，每个数据库都有相关的元数据，包含三个标注属性：年龄（有序），性别（无序）和种族（无序）。通过调查 MORPH AlbumII (MORPHII) 上的所有三个属性估计任务，其中包含大约 78K 的超过 20K 个主题的图像。

**CelebA:** CelebA 是一个大型的人脸属性数据库拥有超过 10 万个身份的 200K 个名人图像，每个人拥有 40 个属性注释。该数据集中的图像在姿态，表情，种族，背景等方面存在较大的变化，使得面部属性估计具有挑战性。此外，由于有 40 个属性标注，CelebA 数据库在特征学习效率方面对联合属性估计算法提出了挑战。

**LFWA:**<sup>[38]</sup> 是另一个无约束的人脸属性数据库，其中包含来自 LFW 数据库的脸部图像（5,749 个身份的 13,233 张图像），以及与 CelebA 数据库中相同的 40 个属性

注释。

**Adience:**<sup>[12]</sup> Adience 数据集来源为的 Flickr 相册，由用户使用 iPhone 或者其它智能手机设备拍摄，该数据集主要用于进行年龄和性别的未经过滤的面孔估计。同时，里面还进行了相应的具有里程碑意义的标注，其中包含 2284 个类别和 26580 张图片。

**Chalearn LAP and FotW:**<sup>[39]</sup> 挑战系列从 2011 年开始，在促进人们视觉或多模式分析方面取得了非常成功的成果。LAPAge2015 是一个无约束的脸部数据库，用于在 ICCV 2015 上发布的视在年龄估计。该数据库包含 4,699 张脸部图像，每个平均年龄至少由 10 个不同的用户估算。数据库被分割为 2,476 张图像进行训练，1,136 张图像进行验证，1,087 张图像进行测试。由于年龄信息的测试不可用，主要使用 validation 集进行测试。FotW 数据库是通过收集来自互联网的公开可用图像创建的，其中包含两个数据集，一个用于辅助分类，另一个用于性别和微笑分类。FotW 数据集分别包含 5,651,2,826 和 4,086 幅用于训练，验证和测试的面部图像；每个都用七个二进制附件属性注释（见表 5 (a)）。FotW 性别和笑容数据集分别由 6171 个，3086 个和 8505 个面部图像组成，用于训练，验证和测试；每个都注明三元性别（男性，女性，不确定）和二元微笑的属性。

**LBS:** LabeledBySelf，即实验室自行标注的属性数据集，包括性别，年龄，表情，发型，墨镜等 9 种属性标注，标签的标准有些特别，全都是无序性的分类标准，即便是生活中普遍认为的有序性标签年龄，也被按照分类标准分成了婴儿，儿童，青年，中年，老年五个类别。一共有 85416 张图，其中 65416 张作为训练集，20000 张作为测试集。

这些数据库可以根据所使用的注释方法分为三类：(i) 具有序性标签的数据库 (MORPHII 和 LFWA)，(ii) 具有二进制属性的数据库 (CelebA, LFWA) 和 (iii) 具有单个属性的数据库 LAPAge2015 和 LBS 数据库。我们可以看到，除了 MORPH II 数据库，其他数据库主要包含真实场景下的人脸图像。

可以看到人脸属性的数据集其实比较庞大，如果都能够充分利用，可以获得相比于单一数据集更加出色的识别效果。但是每个数据集之间的数据不同、数量不同、标注不同，实际使用中往往使用先训练一个再训练一个的流程，非常耗时而且不能够保证模型效果在原有数据集上保持良好的效果。实际使用来看，使用 celeA 训练的数据对于 lfwA 的数据效果并不好，而加入 lfwA 的数据训练就可以提升相关 lfwA 上的准确率。但需要注意的是 LFWA 的数据量远小于 celeA 的数据量，简单的把两个数据合起来训练，两个数据库之前的差异分布其实并不能得到特别好的弥补，训练



图 3-2 数据集图片示例

的准确率还是不能和单独使用 lfwA 相比。

类似的问题对于年龄这一属性更严峻一点，不同的数据库标注是不一样的，在 morph 中是连续的标签，但是在 Adience 数据集上，年龄的标注是 7 个单独的类别，如果强行进行 label 的转换就会存在很多不匹配的现象，无法对其进行测试，但根据 Adience 重新 finetune，那么就会存在类似的数据匹配和模型输出改变的问题。

### 3.3 基于传统特征的人脸属性识别

基于传统特征的人脸属性识别<sup>[5]</sup>往往采用特征提取和分类器结合的方式，其中较为经典的是基于 DIF 特征的人脸属性识别是经典的属性学习方法，在 morphII 上一度取得了非常优秀的实验结果，基本框架概述如下：

前端为特征提取阶段，旨在提取对属性有判别力的特征，而不是完全无监督的。后端连接一个层级式的分类器，用于属性学习。具体结构见下图：

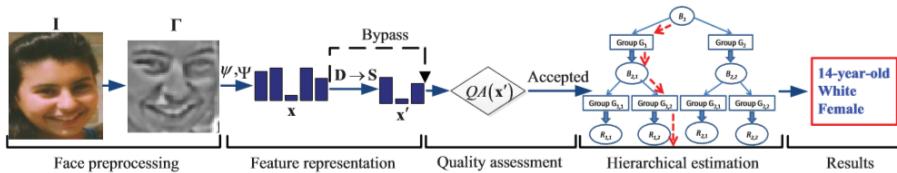


图 3-3 基于 DIF 特征的人脸属性识别

其中有两个主要部分：DIF（Demographic informative features）特征提取，层级式分类器。

#### 1) DIF 特征：

DIF (Demographic informative features) 是基于 BIF (生物启发式特征) 的。比如，输入一个人脸部件，先用 Gabor 滤波器提特征 (12 个尺度，8 个方向)，再做一些池化操作，以减小特征图的数目和维度 (6 个尺度，8 个方向)，将得到的特征串成一个 4280 维的长向量，用来做之后的分类等任务。总体上还是一个无监督的特征处理方法。所以之后，又对此工作做了改进，旨在不仅能够抓住图像细节，还能减小冗余性，提高特征与最终识别任务的相关性。这一部分主要引入一些特征学习工作，从之前的特征集中不断特征子集，挑选出最相关的特征，比如：学习一个新的特征子空间 (如 LDA)，基于 Boosting 的特征选择。

#### 2) 层级分类器的建设：

层级分类器主要针对年龄。比如，首先进行年龄组分类 (针对数据集设定阈值)，在此按是否超过 18 岁分为两类；低于 18 岁的一类再判断是否低于 7 岁，再分为两类，然后低于 7 岁再进行回归得到具体的年龄数值，以此类推，先一层一层地通过多个分类器树形展开得到具体的人脸年龄段，然后在具体的人连年龄分段中及进行回归。实验证明，这种层级式的分类方式要优于直接分类方法。

基于 DIF 特征的属性识别方法是经典的基于传统特征和分类器的属性识别方法，即使在现在，特征融合、层级分类器建设等操作依然具有一定的借鉴意义

问题与不足：尽管效果不错，但整个方法还是有很多的问题，例如，层级分类器确实能够提升分类的效果，但是复杂都明显过高，并不简洁，使用基于传统滤波器和表层信息的图像特征，需要大量的特征筛选和过滤工作。而且总结来讲是 DIF 系统还处在各个部分的分开设计，整个系统并不是处在一个整体性学习的状态。需要较多的人工干预和训练才能得到较好的效果。

### 3.4 基于共享神经网络特征和最大间隔分类器的人脸属性识别

随着深度学习方法的提出，深度学习的特征慢慢取代了传统的手工设计的特征，结合深度学习中经常使用的分类器，得到了更高的效果。具有代表性的是基于级联 CNN 网络和 SVM 分类器的识别方法[7][? ]，使用两个 CNN 框架 Lnet 和 Anet 进行级联学习，其中 Lnet 负责检测图片中的人脸，Anet 针对于 Lnet 中检测人脸使用交叉熵 loss 进行训练，为了提升识别的准确性使用 SVM 对于 ANet 中的特征进行训练。最后由 SVM 分类器输出具体的人脸属性预测值。其中不难发现器图片的标签就采用的是标签化编码的方式。

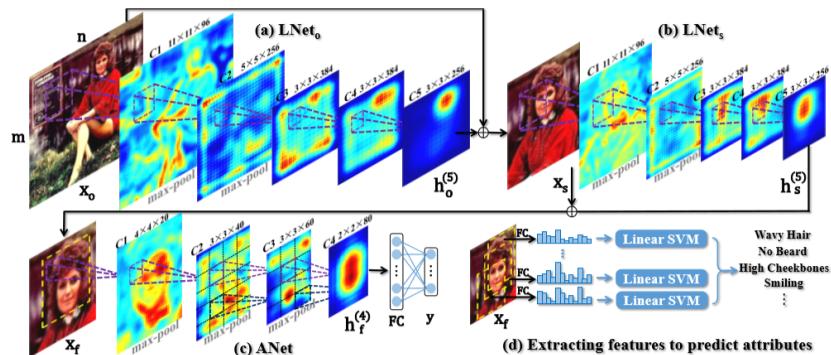


图 3-4 基于共享神经网络特征和 SVM 分类器的人脸属性识别

问题与不足：其实在训练的过程中，神经网络就已经可以对于属性进行预测，但是识别的效果却并不如 SVM 训练的结果，说明在这一框架中，神经网络对于不同属性的自网络决策层和 loss 函数设计的不够好。从图中可以看到，不同的人脸属性之间都是用的同样的 FC 层全连接而来的。不能够体现出属性整体和局部特性。

### 3.5 基于共享特征和子任务模块的端到端的人脸属性学习

机器学习神经网络中的端到端，一般是指输入原始数据，输出最后结果的过程。对于人脸属性的识别过程来讲：如何解决人脸属性多任务的输出是解决问题的关键，目前比较主流的方法是使用网络共享单元和网络子任务模块相互组合的方式<sup>[?]</sup>，具体来讲可以参考下图：对于不同的人脸识别任务，如果希望能够在同一个框架中通过

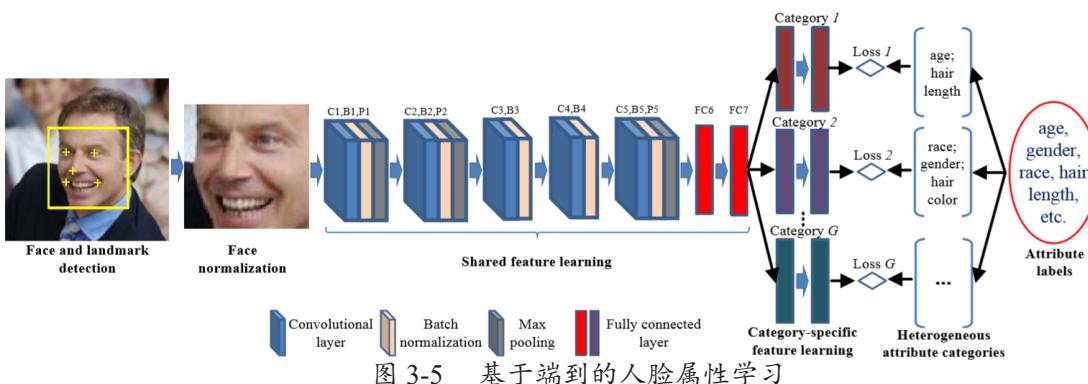


图 3-5 基于端到端的人脸属性学习

端到端的方式解决，而且具有一定可用性，就需要将不同属性识别任务中重复性学习的工作整合成共享模块，然后不同的人脸属性模块再基于共享模块单独对于自身的学习任务单独进行学习。端到端基于共享模块的学习方法可以很大程度上加快过程中的使用效率，同时简化了训练流程。而类别特定的子模块学习旨在对共享特征进行精细调整，以便对每个异构属性类别进行最优估计。由于有效的共享特征学习和类别特定的特征学习，基于共享单元和网络子任务模块的方式在保持低计算代价的同时，实现了更高精度的属性估计精度，使其在许多人脸识别应用中具有价值。

**问题与不足：**基于共享模块和子网模块的识别方式虽然看上去简化了识别的过程，但却是以减少了模型训练的先验规则为代价。也就是说把更多的学习过程交给了模型本身，那么模型的学习过程一方面取决于模型自身设计的学习能力和使用的训练方法，而更重要的是训练数据的选择和实用。

尤其对于端到端网络来讲，由于整体网络结构变得封闭和训练方式的固定化，训练数据的选择和训练过程中对于真实环境的模拟就变得尤为重要。但真实场景中的数据具有较大的变化，包括数据的场景来源不同，数据中样本的比例不均匀，数据自身的姿态和角度变化等。训练样本的选取和预处理过程成了和网络结构选择一样重要的问题。

## 3.6 使用 SA-softmax 进行模型稳定化输出

模型的稳定性输出体现在很多方面，如常见的连续变化的视频，不同设备收集的图像等。在这些场景中，网络可能偶尔会发生判断错误的情况，影响用户的体验。虽然很多时候可以采用平滑的策略进行弥补，但是网络本身的稳定性性能是整个识别系统的关键。为了提升属性识别模型的稳定化输出，我提出了基于带有自评功能的 softmax（Self-assessment softmax SA-softmax），以及相应的动态标签（Dynamic tag, Dt）训练方法来增强网络的稳定性。

### 3.6.1 SA-softmax

传统的 softmax with loss 面对 N 个输入类别，或有 N 元数字作为输入，同时输入的标签是从 0 N-1 的一个整形数字用以表示具体的 ground truth。通过 softmax 操作，计算出每个类别的相对概率值。将对应标签的概率输出取负对数作为损失函数和优化的对象。

SA-softmaxLoss 在分类结果上将网络分类概率输出由 N 个，改为 N+1 个，第 N+1 维值定义为无法识别类（unrecognizable class）。分类 label 由原来的一元数字标签变为一个四元的整形数组分别用于储存图片在过去训练过程中出现的次数、图像被判正确的次数、图像真实的标签，图像目前的标签。训练的过程中使用动态标签（Dynamic tag, Dt）训练方法，不断更新训练图片的类别。

测试的过程中，首先使用 softmax 对于所有 N+1 类求出无法识别类的概率输出，使用 1-无法识别类的概率作为网络的置信输出。根据置信度对照表 3-2 给出网络置信度评价，对其余的类别重新使用 softmax 进行概率获取。最后的输出形式是：输出类别概率 + 网络置信度概率。

表 3-2 SA-softmax 的置信度判断对照表

网络置信度	模型自评描述
S:80-100	判定模型输出非常自信
A:60-80	判定基本正确
B:40-60	判定可以猜测但不对结果负责
C:20-40	判定难以识别
D:0-20	判定完全无法识别

### 3.6.2 动态调整标签的训练方法

对于 softmax 简单的加入一类未知类，还远远不够，至少训练的样本都没有，这分支的梯度永远都是 0，不会对网络产生训练作用。但实际上训练开始的时候我们是无法知道哪些图片是网络不能够正确识别的，所以引入动态调整标签的训练方法：于是在现有的框架下，我们设计了如 Algorithm1 的训练方法：

---

#### **Algorithm 1** 动态标签调整的算法流程

---

```

1: 首先不加入 SA-softmax，将分类网络训练至收敛。
2: 将训练收敛的网络模型的 softmax 改成 SA-softmax。
3: while 模型收敛 or 达到训练次数 do
4:   for epoch  $i \in [0, 9]$  do
5:     更新训练图片的标签四元组中的出现次数和判断正确次数。
6:   end for
7:   if 图片现有标签和实际标签是否一致 then
8:     if 训练 10 次准确率高于 0.5 then
9:       不对标签进行修改，说明被模型对该图片训练良好。
10:    else {准确率低于 0.5}
11:      将图片标签标记成 N+1，说明网络无法将其良好训练。
12:    end if
13:   else
14:     if 准确率高于 0.5 then
15:       不对标签进行修改，这类图片被成功分到 N+1 类，模型自评成功。
16:     else {准确率低于 0.5}
17:       改回原来的标签，说明依然并不能将其分类，应当重新观察。
18:     end if
19:   end if
20:   清空出现次数和被判断正确的次数
21: end while

```

---

通过动态标签调整的算法，我们就可以动态的调整网络无法识别的样本标签，一方面防止标签的错误标识，另一方面也为自评网络模块提供了相应的正样本。这样经过一定轮数的调试就可以获得具有自评模块的神经网络预测网络。

## 3.7 实验设置

在上一节中，总结了在人脸属性任务的主流方法和发展过程，可以看出随着深度学习的发展和端到端学习在模式识别过程中的发展，很多问题都得到了改善，但依然存在着一些主流问题和场景困境一直存在，我仔细对于相关问题进行了思考总结。

1. 问题一：人脸数据库的标注各不相同，使用怎样的框架才能将不同的数据库都充分利用起来。
2. 问题二：使用怎样的数据输入和数据处理方式和训练方式，才能发挥深度学习的特性。
3. 问题三：提高模型输出的稳定性，减轻网络错误输出的偶然性和影响。

针对对于问题的不同种场景分别设计任务和实验进行验证和训练网络优化。具体介绍如下：

### 3.7.1 网络结构和训练环境

在本文中，所有的网络结构都以改进版的 Alexnet 为基础，进行训练。具体网络结构细节包括：Alexnet 的基础结构包括：5 层卷积，5 层 pooling，2 个全连接层；每个卷积层后面都加入 BatchNorm 层，所有的激活函数都是用 relu，将原来的卷积扩充为 ConvUnit；所有的 pooling 都采用 kernel 为  $3 \times 3$ ，步长为 2 的 Max pooling；除此之外在最后层的 pooling 输出，不同的任务会采取不同的 subnet 结构作为训练任务的分支，这在不同的实验中会一一说明。

输入人脸图片以人脸检测输出的正方形人脸框为基础，输入大小都为  $256 \times 256$ 。在训练属性之前，将网络在人脸识别的数据中进行了预训练。实验数据中使用了两台拥有 4 块 nvidia-GTX1080 显卡的机器，使用 caffe 和 mxnet<sup>[40]</sup> 进行了相关训练，包括在一些任务上的分布式的多机多卡训练。

下面给出关于多机多卡训练的网络结构示意图：关于超参数的选择学习率使用 multi-step 的方式进行训练，前 50000 轮学习率设置为 0.001，之后每 100000 轮降低 0.1，weightdecay 设置为 0.00005。momentum 设置为 0.9。

### 3.7.2 数据集并行训练的方式改进问题一

对于不同的数据集来讲，预处理的方式往往类似，也就是说输入图片虽然不同，但是输入的格式是一致的（事实上即使不同也没有什么问题，关键是图片数据不同），

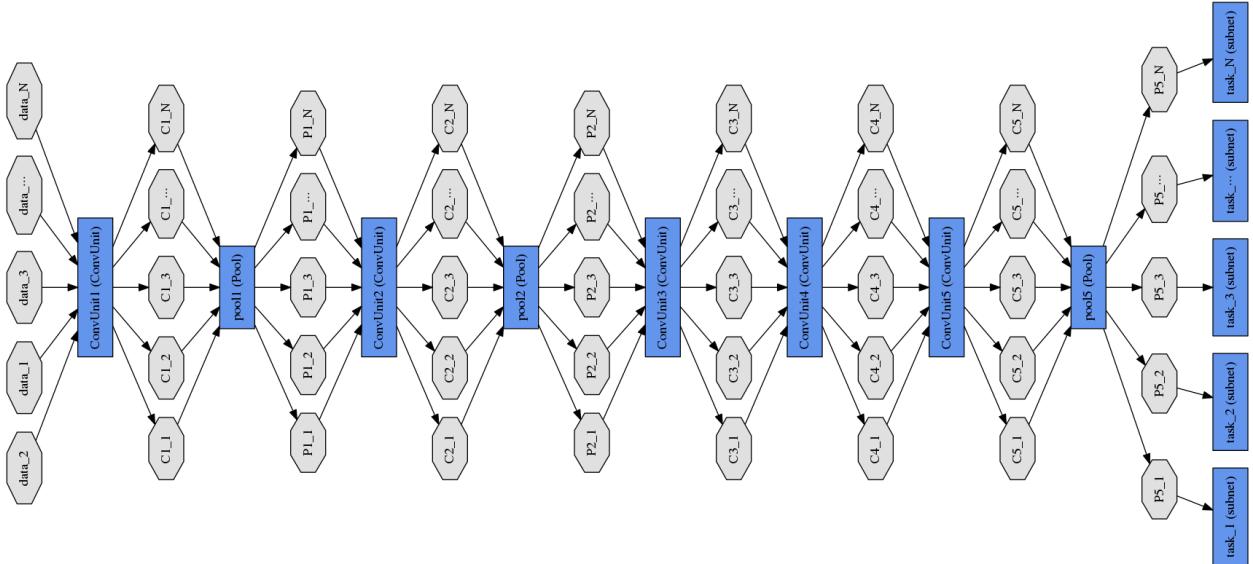
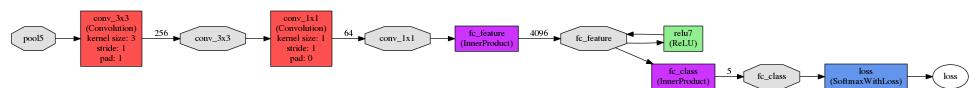


图 3-6 数据集并行的网络结构

但是因为标签不同，导致在一个网络结构中无法进行统一的训练。那么不妨就按照多个单独的网络对于图像进行训练，每个网络在特征提取阶段采用相同参数的全卷及网络结构进行特征提取，但每一层的特征图会单独进行存储，训练的时候每个数据集都按照自己的数据结构特性为了拟合 loss 层的设计，采用不同的子网络结构。在这一过程中需要使用上一节提到的多机多卡的技术，不同的单机多卡并不能满足实验对于计算资源的需求。而如果为了训练采用 batchsize 较小的策略，往往不能够取得良好的效果。

为了验证这方案的有效性，我们使用两个数据集 Morph 和 LBS 数据集作为联合训练输入，Morph 种年龄的标签按照 0~100 的有序性分布，而在 LBS 数据中，年龄是无序性的分成了 5 个类别，为此单独为此单独设计了相关的子网络模型，其子网络结构分别设计如下：



(a) 针对于 LBS 的无序性 5 类年龄标签子网络结构



(b) 针对于 Morph 的 0~100 的有序性标签子网络结构

图 3-7 数据集并行的网络结构

其中  $3 \times 3$  的子网络是为了学习共享特征图中的信息， $1 \times 1$  的卷积是为了减少计算量，`fc_feature` 为了提取相关任务特征，`fc_class/regression` 是对应结果的表示形式。

测试方法对于回归的年龄数值使用 MAE(mean absolute error)，对于分类的年龄使用 Precision 来衡量性能：

$$\begin{aligned} MAE &= \frac{\sum_{i=1}^n |y_i - x_i|}{n} \\ Presion &= \frac{N_{tp}}{N} \end{aligned} \quad (3-1)$$

经过训练可以得到具体的实验结果可参见下表：

表 3-3 在 Morph 和 LBS 上的准确率结果

训练方式	MorphII	LBS	Morph 和 LBS 并行训练
Morph II 上的 MAE (绝对误差)	4.2	NA	3.5
LBS 上的 Presion (百分比)	NA	82.9	85.2

### 3.7.3 人脸矫正固定输入格式改进问题二

在很多人脸任务中，由于人脸是一个基于模板的识别对象，所以对于固定空间分布有着一定的要求，直接检测的人脸对于人脸属性的任务很多时候效果不佳，这一问题在人脸识别的相关算法上已经有所体现，那么使用人脸矫正的相关与处理方式可以很好地解决人脸对于固定空间分布的要求，具体经常采用的方式是根据人脸 landmark 点的人脸 alignment 处理：

在得到人脸五个位置的关键点即左眼眼球中心、右眼眼球中心、鼻尖、最左角和右嘴角之后，通过计算五点与标准脸五点之间的仿射变换矩阵，之后将图片中所有点与仿射变换矩阵相乘，得到了经过放射变化的图像，通过简单的截取可以得到对应的矫正之后的图像。实验中标准脸大小为  $450 \times 450$ ，其中五个点的中心坐标分别为：(200.0,260.0)、(288.0,260.0)、(244.0,488.0)、(206.0,370.0)、(282.0,370.0)，矫正效果：

针对于人脸矫正对于训练的影响主要使用了 Chalearn 数据集中的性别和微笑属性作为任务目标，为了验证实验效果没有使用上一步中数据并行的策略。只是简单的式使用了基本的 alexnet 结构，输入的图片处理方式不同。单独训练两个网络，最后得到准确率结果如下：



(a) 人脸矫正之前的图像 (b) 人脸矫正之后的图像

图 3-8 人脸矫正效果演示

表 3-4 在 Chalearn 数据集上的性别和微笑准确率结果

测试集	性别准确率	微笑准确率
人脸框直接输入	84.5	81.4
人脸矫正后输入	89.9	84.3

### 3.7.4 网络自评估模块改进问题三

在这一模块中，依然选取了 LBS 数据集中的年龄 5 类作为训练和验证的测试集，依然基于基本的 Alex 网络结构，子网络模块会分别使用 softmax 和 SA-softmax。

主要比照过程，是使用普通 softmax 训练的得到的人脸年龄准确率，和加入 SA-Softmax 训练策略之后网络整体性能对于，以五类年龄分类的 top1 准确率作为评价指标。但值得注意的是，由于自评模块的输出和正常的输出不一致，所以使用简单的准确率进行评判其实并没有完全利用自评网络模块的结果输出，我们通过自评模块置信度对照表，对于不同区间的网络准确率进行计算，并且统计各个区间之间图片的图片出现次数。一一和普通网络的输出的准确率进行对照，得到结果如下：

表 3-5 在 LBS 数据集上的性别准确率结果

置信分段	年龄分类准确率	测试图片数目
正常训练	85.2	20000
自评训练	93.1	20000
S	95.9	15575
A	88.8	2830
B	80.1	1031
C	77.8	384
D	30.9	180

### 3.8 实验结论与分析

从实验的结果来看：

1. 数据集并行训练的方式进行训练的过程之中，经过并行训练的数据集得到的模型在 morph 上 MAE 误差减少了 0.7。也就是说减少 18% 的误差，而在 LBS 数据集的测试上准确率提升了 2.3%，减少了 13% 的误差。可以看到联合训练的方式能够起到增加训练数据的效果，在不同的数据集中都取得了很好的提升。而且得益于联合训练的结构特性，可以在一个模型中输出两种数据格式的预测值，也就是年龄的回归值和分类值，相比于原本每种数据格式都需要重新训练的方式更加简洁。
2. 对于问题二，使用人脸矫正的图片可以提高人脸性别和微笑的识别效果，对于性别任务来讲任务从准确率 84.5 提升到 89.9，错误率减少了 34.8%，微笑的准确率从 81.4 提升到 84.3，错误率减少了 15.5%。

说明在使用人脸矫正的方法情况下，可以提升对于微笑和性别的识别准确率。但是和原本的直接将人脸检测的图片框截取的图片作为输入的方式，人脸矫正的处理方式则稍显复杂，而且还一定程度上取决于人脸 landmark 的效果。而且仔细分析来看，对于微笑任务的识别效果提升并不如人脸性别的识别效果提升。说明人脸矫正的策略可能并不是所有的人脸属性任务都会具有相同的效果，比如人脸姿势和角度相关属性，通过矫正甚至可能改变人脸属性。但是人脸属性矫正还是作为当年最流行的人脸任务与处理方式，在大多数任务中被证明有效。

3. 对于问题三，自评网路对于人脸年龄的识别有比较明显的提升，对于使用普通 softmax 和 SA-softmax 的不同策略的模型，使用自评网络训练的方法将人脸年龄 5 分类的 top1 准确率从 85.2 提升到 93.1，效果显著。而具体来看，有 15575 张图片的置信度为 S 级，而且 S 级的人脸准确率达到 95.9%。而评级为 A 的图片数量为 2830，准确率为 88.8%。其余评级的准确率都比正常训练要低。也就是说有 92.0% 的图片可以通过自评模块获得比正常训练更高的准确率。这也正是自评网络的优势所在，能够提升系统的整体判断准确率。从而减少人工审核的工作量。

### 3.9 本章小结

在本章中，我们通过对于人脸属性和性质的分析，不同人脸属性的数据库简介。以及人脸属性中的常见做法的分析和介绍，逐步分析了人脸属性中面临的问题，并将其归纳为三个部分，并针对不同部分设计实验。

在设计人脸属性识别的方案过程中，增加 BN 和自网络模块，以神经网络为基础设计人脸属性的识别模型。对于具有不同标签的人脸属性的数据，共享基础结构，单独设计自网络模块的方式并行训练，并采用多机多卡的方式加速训练。而对于人脸属性中输入格式形式，使用经典的基于 landmark 的形式进行人脸矫正。最后为了稳定网络的输出，我们为网络加入具有自评估能力的 SA-softmax，不仅提高了人脸的属性识别的准确率，同时增加了模型对于自身能力的评估能力。

## 第四章 对抗生成网络在人脸属性中的应用

在上一章节中，主要使用基本的卷积神经网络对于人脸的属性任务进行识别，虽然取得了一定的效果，但是还是局限于传统监督式学习的方法，在固定的数据库上做分析，那么有没有什么办法能够让网络摆脱标注数据的限制，在未经过训练的数据中依然能够有良好的表现呢，结合对抗生成网络研究，我们对其进行了一定的探索。

### 4.1 对抗生成网络相关技术的介绍

早在 2014 年，人们对于神经网络技术的研究非常狂热的同时，也有一部分理智的科学家认为神经网络的输出判断具有非常高的风险，输出具有非常高的不稳定性，所谓数据集上的准确率超越人类不过是一场谎言，为了戳穿这一谎言，他们在神经网络判断正确的图片上简单加了一些噪声，对于人类来说根本没有察觉图像的变化，但是在神经网络却完全将其判断成另外一种物体。同时科学家们宣称这样极具欺骗性的图片并非偶然得到，而是可以量产的，比如通过对抗生成网络。

借助于博弈论中的零和博弈思想（在零和博弈中，游戏玩家之间的利益总和是固定的，即一方获得收益，另一方就要承担损失。）Goodfellow 极具想象力的提出了可以通过搭建两个对抗的网络，各自的目的就是降低对方的准确率，或者说提升对方 loss。通过这样非常具有竞争性的训练过程，最够提升两个网络的性能。具体来讲：在对抗生成网络中，玩家的角色会分别有生成模型 (generative model) 和判别式模型 (discriminative model) 充当。生成模型 G 捕捉样本数据的分布，判别模型 D 是一个二分类器，估计一个样本来自于训练数据（而非生成数据）的概率。G 和 D 可以是线性代数的算法操作组合，也可以是神经网络的网络模型，都可以理解成或者定义成非线性函数。通过不断调整 G 和 D，直到 D 不能把事件区分出来为止。在调整过程中，一方面需要优化 G，使得它尽可能的让 D 混淆；另一方面需要优化 D，使得它尽可能的能区分出假冒的东西；当 D 无法区分出事件的来源的时候，可以认为，G 和 M 是一样的。从而，就获得了能够以假乱真的数据。

但是对抗生成网络很明显并不能像正常的 CNN 网络一样对于具体的模式识别任务，但是作为探究 CNN 生成原理的一部分，对抗生成网络主要是希望能够了解 CNN 能够从图像中学习到什么样的信息，怎样学习的，并且能否以较为直观的形式也就

是生成图像来表示出来，（尽管学习到的东西很多时候并不能够以图像的形式进行展现）。

## 4.2 探究对抗神经网络的应用

在之前对于对抗生成网络的介绍中，可以发现对抗生成网络最初是来证明神经网络算法对于数据分布具有一定的局限性。而慢慢发展，人们并不在乎神经网络是否对于数据分布有一定的局限性，而狂热的希望能够通过对抗生成网络获得以假乱真的机器生成图片。似乎人们觉得如果机器能创造他，那机器肯定可以了解他，那么识别他也是轻而易举。于是乎这种炫酷，但是有一定投机取巧性质的思路不仅开始影响最初使用对抗生成网络探究神经网络有效性的本意，也影响着各种识别任务的传统数据 + 模型的预测方式。

在现有基于模型和分类算法的场景下，对抗神经网络还很难展现出对于属性任务有明显的提升，而对抗生成网络往往以生成各种图片来获得关注和人们的注意，而在业内确实有一些方法可以生成带有人脸属性的人脸图片。那么是否可以通过生成图片的方式为神经网络增加训练的数据，从而起到为属性识别增添数据的作用就成了对抗生成网络在人脸属性上的应用方向。首先可以从使用对抗生成网络直接生成训练数据的方向入手。

### 4.2.1 使用对抗生成网络生成真实图像

首先参考了 DCGAN<sup>[41]</sup> 的方法和思路，采用了如图 4-1 的网络结构作为生成图像的基本架构：

生成模型：输入 100 维的噪声到第一个全连接层，将其映射为 1024 维，然后再把 1024 的一维向量重塑成 1024 个通道的 4x4 的特征图。基本规律是生成网络的每一个下一层是反卷积层，通道数减半，图像尺寸加倍。

判别模型：就是一个没有池化层的全卷积网络，输入是生成模型输出的图像，输出一个标量，表示输入数据属于训练数据而非生成样本的概率。

#### 4.2.1.1 生成数字图像

首先使用 mnist 数据<sup>[42]</sup> 作为训练样本，实现了从 100 维的噪声生成数字图像，发现对抗生成网络确实可以生成较为逼真的数字图片，而且生成的图片并不局限于一种状态。

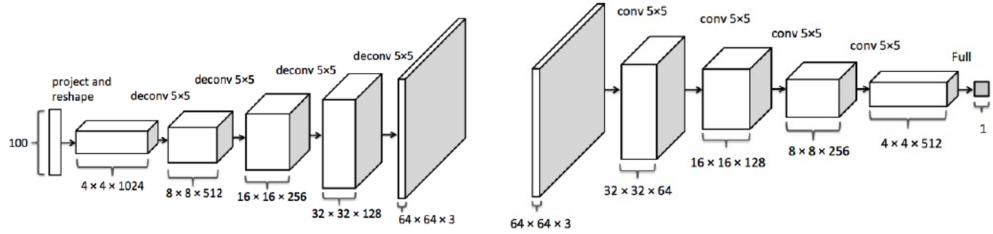


图 4-1 DCGAN 的网络结构:(a)DCGAN 中的生成网络模型结构; (b)DCGAN 中的判决模型网络结构

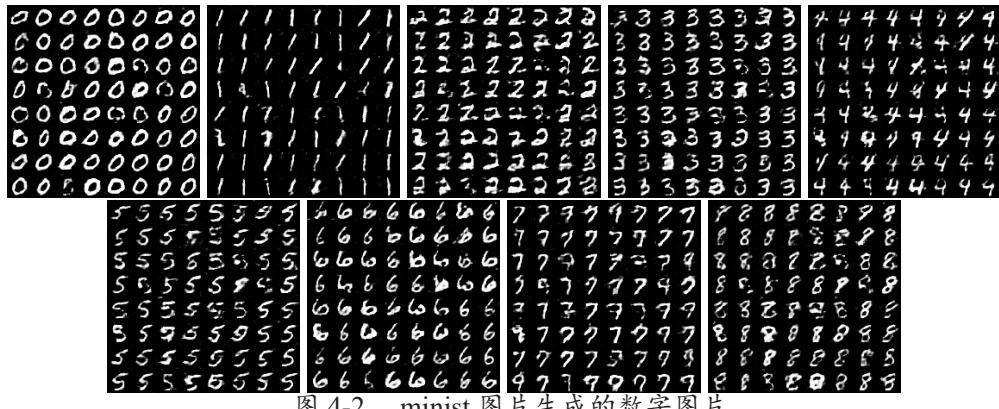


图 4-2 minist 图片生成的数字图片

#### 4.2.1.2 生成物体图像

于是，我们加大了训练的难度，更换 cifar10<sup>[43]</sup> 数据作为训练的样本，实现了 32\*32 的彩色物体图像生成，但是发现效果似乎并不够出色，在生成的过程中，虽然能够看物体的生成具有一定的形状信息和物体轮廓，但是整体的效果却不是很好，不能够生成出具体的能够和真是图像匹配的物体图片。



(a) 真实的 cifar10 图像数据 (b) 伪造的 cifar10 图像数据

图 4-3 cifar10 图片生成的数字图片

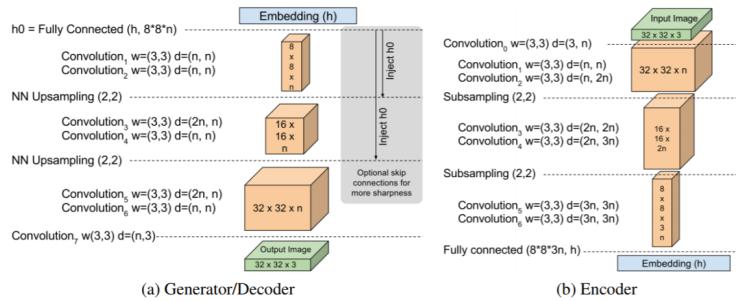


图 4-4 BGAN 的网络结构

### 4.2.1.3 生成人脸图像

那么任务换到在人脸图片的生成上，我们决定放弃 DCGAN 的结构，转用 BGAN<sup>[44]</sup> 和 WGAN 距离<sup>[45]</sup> 的方式进行人脸图像的生成，事实证明相比于 DCGAN，这种方式可以取得更加鲁棒和逼真的效果。

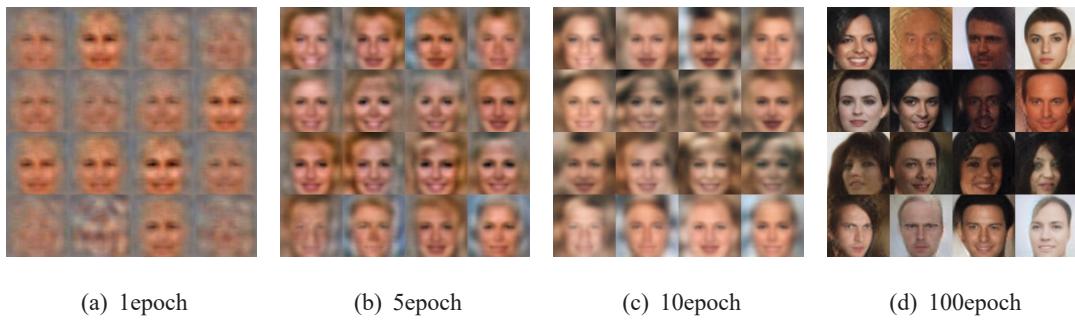


图 4-5 人脸图片生成效果图

结果显示在人脸生成的过程之中，可以产生非常具有迷惑性和真实性的人脸，是对抗生成网络作为人脸生成的基石之一。但是在监督学习的框架之下，使用对抗生成训练数据还有关键的一步，就是如何获取生成样本的标签。虽然在上文提到的 C 对抗生成和 Info 对抗生成都有对于生成带有标注的人脸的尝试，但是从具体的效果中发现其效果还具有一定的不自然性，对于这种情况我们分析之后觉得没有尝试的必要，

主要有以下几点原因：. 对抗生成网络的原理是拆解训练图片中的图像分布因子，将其储存在网络的参数之中，而随机噪声的输入，随着不断反卷积的操作，将输出的模式不断固化并且最终映射回原本的图像空间，其实是一个低维向高维投射的过程，但是因为投射的两维其实都是具有无限可能的，所以确实有完成这种映射的可能，更何况人脸认为现实中存在的脸本身的范围就要比图像空间所能表示的范围要小得多。但是基于目前 CNN 方法的对抗生成网络的原理是学习数据中的图像分布

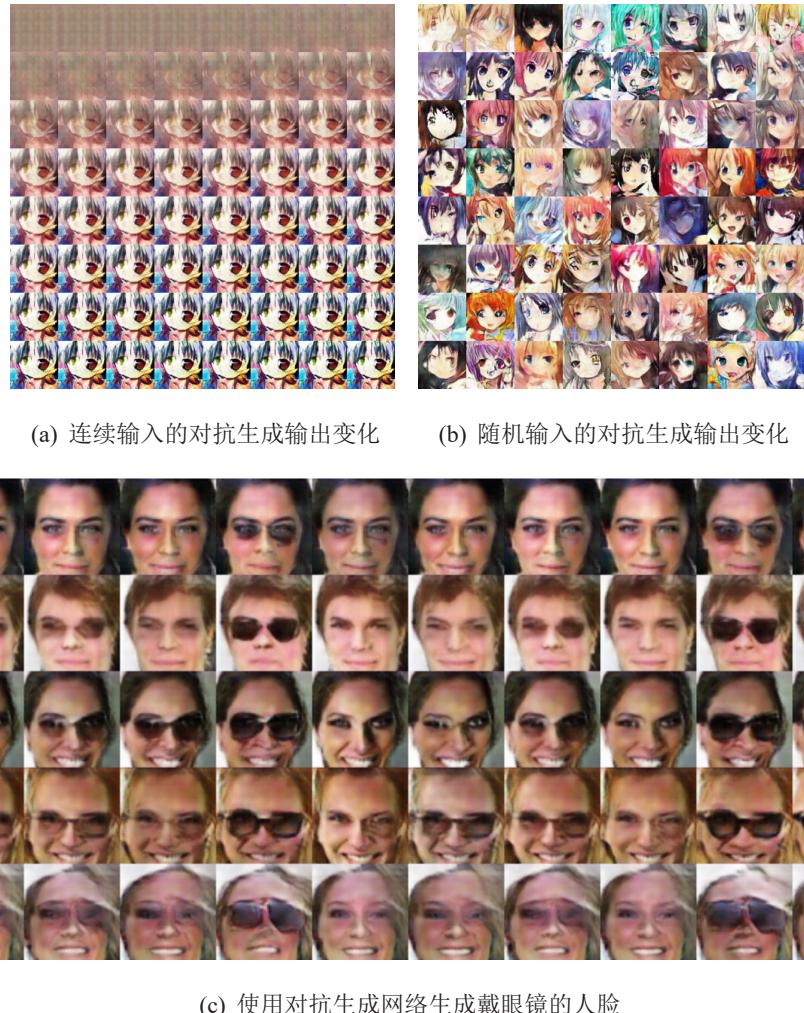


图 4-6 对抗生成网络生成图片的局限性

和梯度下降法的学习方法其实核心是针对于损失函数的优化，虽然在 WGAN 中对于对抗生成网络的损失函数又一次做出了优化，但依然不能够完美的和我们的任务相匹配，我们甚至不清楚自己想要的 `groundtruth` 是什么样的，所以损失函数的制定其实还不够出色，这也是为什么对抗生成网络的输出虽然显得很惊艳，但大多数时候还是会给人一种不真实感觉的原因。所以在没有合格的损失函数之前，是不能够使用现有的对抗生成网络来进行神经网络的监督学习训练的，因为对抗生成网络生成的图片并不能真实的代表真实图片的分布。

但是在本论文的实验后期，我们想到了使用对抗生成网络来尝试迁移学习上的能力，其中使用了对抗生成网络在超分辨率上的应用。

#### 4.2.2 使用对抗生成网络提高人脸分辨率

在发现对抗生成网络其实并不能直接从噪声生成具有一定训练意义的图片之后，我们并没有气馁。在参考了很多具有使用意义的对抗生成网络工作之后，决定从超像素的方向重新研究。超分辨率是通过硬件或软件的方法提高原有图像的分辨率的过程。在人脸超分辨率的之中，核心的思想就是在不改变人脸身份信息和属性信息的情况下，将人脸图像的更多细节还原出来。

##### 4.2.2.1 使用对抗生成网络生成高分辨率的图片

在借鉴了超分辨率的框架之后，设计了超分辨率神经网络框架是 TRGAN<sup>[46]</sup>，

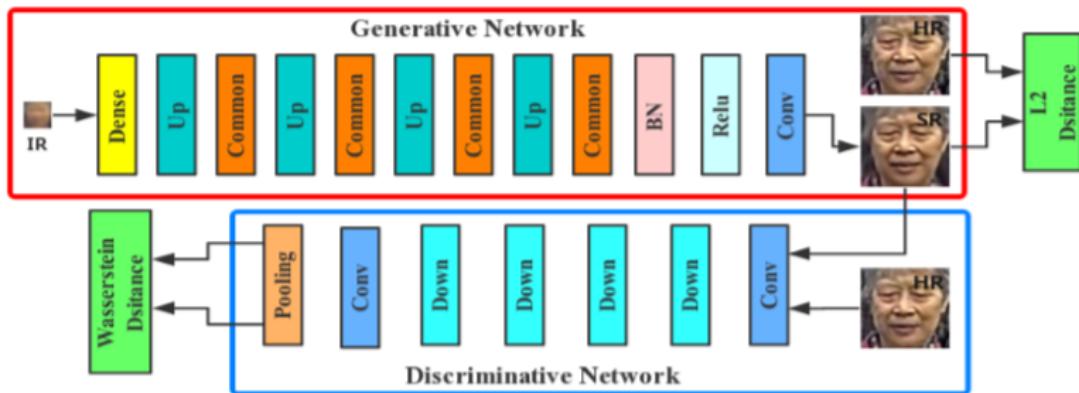


图 4-7 人脸超分辨率所使用的网络结构

具体来讲：common 模块：参考了 resnet<sup>[47]</sup> 的 block 模块，输入特征图分成两个分支，一个分支接 BatchNorm 之后接卷积层，再接一层 batchnorm 层和一层卷积层，最后该分支的 block 输出的特征图与另一个未经操作的分支使用元素对位相加 (elementWise)。其目的是为了增加网络的深度，同时又能够为多层特征的融合提供不同的通道。

upSample 模块：在 common 模块的基础上，输入层的第一个分支连接到第一层卷积之后使用 pixel shuffle<sup>[48]</sup> 的操作之后将图像尺寸变为原来的 2 倍。然后再接一层 BatchNorm 和卷积；与此同时，第二层分支使用常规的双线性插值的方法扩大到两倍大小，使用 1x1 卷积连接之后将两个分支 elementWise 相加。upsampling 的操作相比 common 层的操作，可以让特征图变为原来的两倍大小，又增加了一定的特征融合。

downsample 模块：downsample 的操作在 common 的基础上分别在两个分支的后面加入了 stride 为 2 的 pooling 层，可以保证输出特征图的大小为原来的一半。使

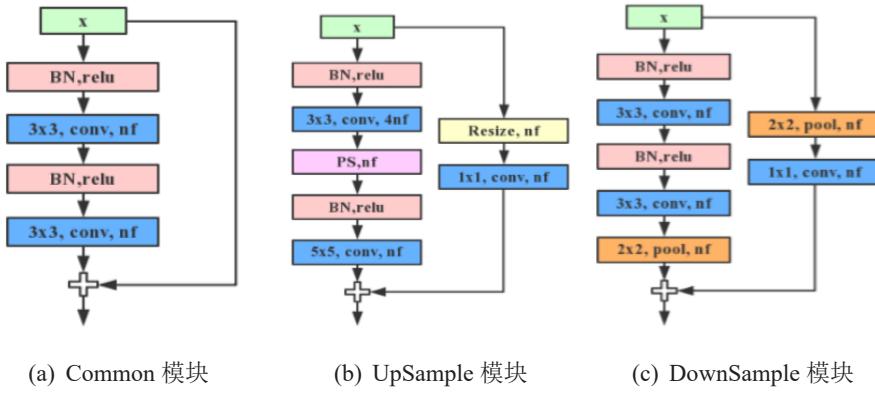


图 4-8 TRGAN 中基于 resnet 的三种网络结构改进

用该框架在 celeA 数据集上进行性训练，其中输入是 8\*8 的低分辨率人脸，输出是 64\*64 的高清人脸。优化的损失函数分别是输出的图片与原始高清图片的差值以及 WGAN 中的 Wassestein Distance.

$$W(P_r, P_g) = \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_g}[f(x)] \quad (4-1)$$

具体的超分辨率实验结果：



图 4-9 超分辨率使用 lfw 图片的效果示意图

可以看到对抗生成网络在超分辨率领域上取得了非常好的效果，可以在基本不改变人脸属性和身份信息的情况下，获取高清的图片。但是我们也发现一些有趣的

现象：生成的图片虽然基本的轮廓不变，但是整体都相比原来更加白皙明亮了一些，整体给人的感觉都更加好看了一些。这其实是因为训练数据的图片都是 celeA 中的明星，其图片质量都比较出色，画风也带有一定的电影元素，似乎是把 celeA 的风格带到了 lfw 中，于是我有了一个大胆的想法。

### 4.3 结合对抗生成超像素实现迁移学习

#### 4.3.1 人脸属性的监督式学习困境

首先引入一个经典的模式识别场景，泛化能力的问题：在之前的工作中，经常可以发现一个经常出现的问题，使用 MTL 的人脸属性框架进行人脸属性识别的过程中，具同样 40 个属性标签的两个数据集 lfwA 和 celeA，两个在各自数据集上训练之后的模型，在各自数据集上的准确率都很高，但是在对方的测试集效果都比较糟糕。

如何进行改善呢？我们针对于这种情况设计了这样的思路：问题引出：对于相同的网络模型，使用相同的训练方法，在不同数据集中的训练之后，对自身数据集的测试集准确率要远远高于其他数据集的测试集。问题分析：首先这不是一个过拟合问题，因为对于数据集中训练集和测试集的准确率较高，所以网络的训练没有问题。但是对于不同数据集的测试集准确率很低，所以推测问题的出现是因为数据的分布不同

尝试解决办法：首先我们先假定网络模型容量可以容纳两个数据的分布（数据的分布可能不满足线性加法，但是应该满足集合性合并不减的特性，所以假定两种数据的分布集合会比原来更大，所以对于网络容量的要求会更大），既然数据的分布不同，就应当减少数据分布对于模型训练带来的影响。

第一种方法就是将两个数据集合并训练，如果标签相同，那么可以简单的将两个数据集合并成一个数据集训练，也可以首先在一个数据集上份训练，再经过另一个数据集 finetune，又或者采用上一章所提到的主干网路参数共享，不同数据集分别使用一个网络支线进行训练。都可以直观地学习到两个数据集之间的数据分布。往往就可以取得较好的效果，有效的提高在不同数据集上准确率的表现。

缺点：最致命的缺点就在于不同数据集的准确率提高，但是难以保证在自身的数据集上数据的准确性。即使采用较小的学习率谨慎的进行 finetune，对于不同任务的训练过程也即将面临着大量的手动干预，还是处于一个监督学习的框架之中。

对此我们决定使用类似于迁移学习的方式来完成这个任务，并且结合对抗生成网络来完成我们的任务。

### 4.3.2 人脸属性的迁移学习猜想

迁移学习<sup>[?]</sup>是把一个领域（即源领域）的知识，迁移到另外一个领域（即目标领域），使得目标领域能够取得更好的学习效果。通常，源领域数据量充足，而目标领域数据量较小，迁移学习需要将在数据量充足的情况下学习到的知识，迁移到数据量小的新环境中。在图像领域，最主要完成迁移学习的方式就是将在源领域训练得到的模型作为特征提取器，然后在新的环境下使用诸如 SVM、boosting 等方式进行特征学习。或者固定模型的主要参数层，只重新训练后面针对于场景输出的特征提取类别。然后作为新环境的模型。

通常意义上的 finetune, metric Learning 也是迁移学习的分支。而实际上迁移学习并没有明确的范围，其主要的目的是如何利用好已经学习到的知识，并将其应用到实际的应用场景之中。但实际场景虽然没有标注的数据，也是可以通过其他手段获取的知识，也就是说实际场景其实本身也是一种已经存在的知识，我们使用一些技巧将其已有的这些提取出来，并融合现有知识，完全有可能获得更加好的目标域识别效果。

### 4.3.3 基于人脸超分辨率的人脸属性迁移学习实验

在发现可以通过对抗生成网络将低分辨率人脸分辨率提高之后，我们设计了实验来针对上述问题进行研究。决定在 lfwA 的训练过程中混入超分辨率生成的 LFWA 人脸图像，目的也很明显，就是希望这些超分辨率的图片能够把 celeA 中的图像分布带到 lfw 中，从而提升使用 lfwA 数据训练的模型在在 celeA 数据上的性能。

实验中使用的网络是改进的 alexnet 主网络加上 40 个二分类的子网络模块。卷积层中的 stride 全部为 1，保证网络特征图尺寸最后不会为消失。输入大小为 64\*64 的人脸，通过上一章中提到的人脸矫正作为预处理的方式。使用人脸识别的与训练模型作为预训练基础。数据混合策略是按照将超像素图片和 lfwA 的原始图片按照 1:1 的比例作为训练数据。

实验结果如下：

## 4.4 实验结果分析与结论

从训练的结果来看可以看出，使用 celeA 训练数据的模型，在 celeA 测试集中取得了 89.1% 的准确率，但是在 lfwA 上的准确率就降低了和蒂诺。只有 70.7%。类似

表 4-1 在 CELEA 数据集

数据集训练	数据集测试	40 属性平均准确率 (%)
LFWA	CeleA	66.2
CeleA	CeleA	89.1
LFW-hr	CeleA	76.2
LFWA	LFWA	83.4
CeleA	LFWA	70.7
LFW-hr	LFWA	83.3

的使用 lfwA 数据进行训练，也具有同样的现象。这符合人脸属性种监督学习的困境。再加入了人脸超分辨率像素的进行训练之后，情况有了改善，不仅使用超分辨率像素训练的模型在原本的 lfwA 数据集上具有良好的效果，只是下降了 0.1 个百分点，在 celeA 数据集上也提升 10 个百分点。这说明基于超分辨率的像素来完成人脸属性识别是可行的。通过改变训练数据的情况下，对数据的预处理过程添加了一定其他环境噪声分布，这和传统通过数据增强的方式完成的数据噪声加入是有很大不同的。原始的数据增强例如平移、旋转、色彩通道等变换等都还是对于现有数据分布从离散化的输入到连续化扩充，是可以充分的利用输入数据的分布资源。但超分辨的图像变换形式其实可以更深层次的改变图像的状态而最大限度保留图像的标注信息不发生改变。这样是最直接的增加数据分布的方式。尤其是在本实验中 lfwA 中的图片数量较少，数据的分布也更加偏僻且具有离散化，和 celeA 也有很大的差距，简单的数据增强其实并不能完全解决这个问题。所以使用超分辨率的方式进行扩充训练可以取得更好的效果。

但是不得不承认，虽然本文中没有常熟将 celeA 和 lfwA 数据进行混合训练的方式，但是无疑这样的方式其实最直接提升在不同数据场景下识别效果的方式，而且流程更加简单，使用训练数据并行的训练方式其实是可以取得优异的效果的。然而基于超分辨率的迁移学习的优势其实体现在对于未知数据场景的学习能力。在现有数据中进行模型的学习，然后针对于不同的数据场景，使用超分辨率的模型对于场景图片进行学习和复现，这样可以轻松从有标注的数据的低分辨率版本获得有标注数据在对应场景的移植高分辨率版。再将移植的高分辨率的图像通过正常训练的图片。

## 4.5 本章小结

在本章中首先对于对抗生成网络的相关技术做了介绍，然后研究了对抗生成网络现实中比较常见的应用包括使用对抗生成网络生成真实图像和使用对抗生成网络提高图片的分辨率。在探究和实验的过程之中，我们完成了相关的图片生成任务，g 对抗生成网络也体现出了非常令人眼前一亮的伪造图片能力。但针对于人脸属性任务对于图片的要求，生成图片依然很难满足相关的真实程度和标签要求。

但是使用超分辨率对于低分辨率人脸的修复和生成体现了良好的效果。于是结合迁移学习的方式，我们对于人脸属性识别在不同场景的识别效果做了探究性实验。总结来说，迁移学习是监督学习之后最具有研究方向的领域，我们对此进行探究，并发现在我们固定的方法和 celeA 与 lfwA 40 种属性分类的场景下，使用基于超分辨率的学习方法确实可以有一定的效果提升。



## 第五章 总结与展望

### 5.1 全文总结

通观全文，与其说实在研究如何提高人脸属性的识别准确率，倒不如说是在各种偏离基础算法使用场景的情况下，解决一个又一个出现的问题，包括为了能够提高训练速度，在训练中的不同框架，尝试探究多机多卡。为了在实际测试中，具有较高的反馈和实用价值，在基础的神经网络操作中，对于基本算法的加速和改进。为了适应针对数据集训练和评测的这种模式，设计针对于多种属性标签，多种数据集的网络架构。为了对于不同的场景数据分布存在偏差的问题，针对于背景的变化，使用对抗生成网络构建超分辨率系统对于人脸图片进行了人脸的跨域重构，从而提高目标域的识别效果。在这个过程中，对于图像中模式识别的基本算法有所掌握，同样也印证着发现问题，分析问题和解决问题的思路。

从整个毕业设计的效果来看，第三章节中属性识别部分，以探究问题为主，主要工作量体现在设计实验证明不同策略对于模型的提升作用，目标比较明确，其实是常规的科研工作。而在第四章，使用对抗生成网络探究超分辨率的过程，其实是对未知问题的探索，能够参考的信息其实不够多，整体的一步步实验进展，也是建立在对抗生成网络的发展之上，诸如 DCGAN、BGAN、WGAN 对抗生成网络的发展，对我的实验进展起到了关键的推动作用，而在一步步观察生成图片的效果过程之中，既是一个辛苦的重复性验证工作，也是一个见证模型输出逐步稳定变好的过程。

总结来讲，其实做实验的结果可能不会多整个科研学术上有足够的推动作用，但是整个实验的过程和解决实验中所用到的方法将作为我人生中的难忘经历和宝贵财富。

### 5.2 未来展望

本文所介绍的人脸属性识别属于图像识别种基于监督学习的分支，同时也是非常具有代表性的任务之一，类似的人物包括物体识别中的物体性质识别，如经典的鱼种类识别等。所以人脸属性识别的进展需要依托于整个图像识别的基础技术进展和图像数据库的建设。而图像识别领域基础技术的进展其实更加依托于更加基础计算机科学的进步与发展，细节小到晶体管的制造工艺，计算机内存和缓存的读取速

度，处理器的主频提升，布局大到整个体系结构的变革，冯诺依曼体系的变革，量子计算机的进化等，都会对于模式识别算法有着较为深远的影响。

除了对于底层科学的依赖，现实生活中的应用落地也同样具有重大意义，比如慢慢成熟的人脸识别，自动驾驶等新兴技术行业，无一不是有基础的模式识别技术发展而来，但是却无一不在现实的产业结构中引发巨大热潮，让实验室的算法走出实验室出现在人们的现实生活种，可以极大激励人们对于人工智能的探索的热情和改变人类生活状态的前行动力。并且在实际生活中慢慢探索图像识别的规律，加速人工智能领域的快速发展。

## 参考文献

- [1] Cottrell G W, Metcalfe J. EMPATH: Face, emotion, and gender recognition using holons[A]. // NIPS[C]. Proc, 1990: 564– 571.
- [2] Facial Image Processing and Analysis (FIPA) “FG-NET Aging Database,” [EB/OL]. <http://fipa.cs.kit.edu/433.php#Downloads>.
- [3] Ricanek K, Tesafaye T. MORPH: A Longitudinal Image Database of Normal Adult Age-progression[A]. // Proc. FGR[C]. IEEE, 2006: pp. 341–345.
- [4] N Kumar P N B, Nayar S. Facetracer: A search engine for large collections of images with faces[A]. // Proc. ECCV[C]. Springer.
- [5] H Han X L C Otto, Jain A K. Demographic estimation from face images: Human vs. machine performance[J]. Trans. Pattern Anal. Mach. Intell., June.2015: vol. 37, no. 6, pp. 1148–1161.
- [6] PJ Phillips S A R P R H Moon. The FERET Evaluation Methodology for Face-recognition Algorithms[J]. Trans. Pattern Anal. Mach. Intell, 2000: vol. 22, no. 10, pp. 10901104.
- [7] Z Liu X W P Luo, Tang X. Deep learning face attributes in the wild[A]. // ICCV[C]. IEEE, 2015: pp. 3730–3738.
- [8] M Ehrlich T A T J Shields, Amer M R. Facial attributes classification using multi-task representation learning[A]. // CVPR Workshops[C]. IEEE, 2016.
- [9] Guo G, Mu G. A framework for joint estimation of age, gender and ethnicity on a large database[J]. Image Vision Comput., Oct.2014: vol. 32, no. 10, pp. 761–770.
- [10] E Eidinger R E, Hassner T. Age and gender estimation of unfiltered faces[J]. Trans. Inf. Forensics Security, Dec. 2014: vol. 9, no. 12, pp. 2170–2179.
- [11] X Geng Z H Z, Smith-Miles K. Automatic Age Estimation Based on Facial Aging Patterns[J]. Trans.Pattern Anal. Mach. Intell, Dec. 2007: vol. 29, no. 12, pp. 2234–2240.
- [12] Levi G, Hassner T. Age and gender classification using convolutional neural networks[A]. // CVPR Workshops[C]. IEEE, 2015.
- [13] M Uricar R R J M R Timofte, Gool L V. Structured output SVM prediction of apparent age, gender and smile from deep features[A]. // CVPR Workshops[C]. IEEE, 2016.
- [14] Hand E M, Chellappa R. Attributes for improved attributes: A multi-task network for attribute classification[A]. ArXiv e-prints, Apr. 2016.
- [15] A Krizhevsky I S, Hinton G E. ImageNet classification with deep convolutional neural networks[A]. // Proc. NIPS[C]. MIT Press, 2012.
- [16] Rumelhart D E, Hinton G E, Williams R J. Cambridge, MA, USA: MIT Press, 1988: 696–699.

- [17] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional Architecture for Fast Feature Embedding[J]. 2014.
- [18] Lin C Q M, Yan S. Network in network[A]. // ICLR[C]. 2014.
- [19] Dai J, Qi H, Xiong Y, et al. Deformable Convolutional Networks[J]. 2017, abs/1703.06211.
- [20] Girshick R B. Fast R-CNN[J]. 2015, abs/1504.08083.
- [21] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. 2014, abs/1406.4729.
- [22] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. 2014, abs/1409.1556.
- [23] 魏秀参. 全连接层的作用是什么 [EB/OL]. zhihuhttps://www.zhihu.com/question/41037974/answer/150522307.
- [24] Ioffe S S C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[A]. // arXiv: 1502.03167[C]. arXiv preprint, 2015.
- [25] Ren S, He K, Girshick R B, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. 2015, abs/1506.01497.
- [26] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks[A]. // Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics[C]. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010: 249–256.
- [27] He K R S e a Zhang X. Delving deep into rectifiers: Surpassing human-level performance on im-agenet classification[A]. // International Conference on Computer Vision[C]. Proceedings of the IEEE, 2015.
- [28] Sarwar S S, Panda P, Roy K. Gabor Filter Assisted Energy Efficient Fast Learning Convolutional Neural Networks[J]. 2017, abs/1705.04748.
- [29] Corporation N. NVIDIA CUDA C Programming Guide[S]. 2010.
- [30] Awan A A, Hamidouche K, Venkatesh A, et al. Efficient Large Message Broadcast Using NCCL and CUDA-Aware MPI for Deep Learning[A]. // Proceedings of the 23rd European MPI Users' Group Meeting[C]. New York, NY, USA: ACM, 2016: 15–22.
- [31] Lavin A. Fast Algorithms for Convolutional Neural Networks[J]. 2015, abs/1509.09308.
- [32] Shmuel Winograd[EB/OL]. https://en.wikipedia.org/wiki/Shmuel\_Winograd.
- [33] Cho M, Brand D. MEC: Memory-efficient Convolution for Deep Neural Network[A]. // Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017[C]. 2017: 815–824.
- [34] Corporation I. Intel(R) Math Kernel Library for Deep Neural Networks (Intel(R) MKL-DNN)[EB/OL]. https://github.com/01org/mkl-dnn.
- [35] Dukhan M, Tulloch A. Acceleration package for neural networks on ulti-core CPUs[EB/OL]. https://github.com/Maratyszczka/NNPACK.

- 
- [36] Chetlur S, Woolley C, Vandermersch P, et al. cuDNN: Efficient Primitives for Deep Learning[J]. 2014, abs/1410.0759.
  - [37] Han H, Jain A K, Shan S, et al. Heterogeneous Face Attribute Estimation: A Deep Multi-Task Learning Approach[J]. 2017, abs/1706.00906.
  - [38] G B Huang T B M Ramesh, Learned-Miller E. Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments[R]. University of Massachusetts, 07-49, 2007.
  - [39] S Escalera H J E X Barro, Guyon I. Chalearn looking at people: A review of events and resources[A]. // CVPR Workshops[C]. IEEE, Jan. 2017.
  - [40] Chen T, Li M, Li Y, et al. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems[J]. 2015, abs/1512.01274.
  - [41] Radford A, Metz L, Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks[J]. 2015, abs/1511.06434.
  - [42] LeCun Y, Cortes C. MNIST handwritten digit database[J]. 2010.
  - [43] Krizhevsky A, Nair V, Hinton G. CIFAR-10 (Canadian Institute for Advanced Research)[J].
  - [44] Boundary-Seeking Generative Adversarial Networks[A]. eprint arXiv:1702.08431, 02.2017.
  - [45] Arjovsky M, Chintala S, Bottou L. Wasserstein Generative Adversarial Networks[A]. // Proceedings of the 34th International Conference on Machine Learning[C]. International Convention Centre, Sydney, Australia: PMLR, 2017: 214–223.
  - [46] Xu J. Face Hallucination with Tiny Images in Surveillance via Generative Adversarial Networks[A]. // CVPR submission.Paper ID 1832[C].
  - [47] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015, abs/1512.03385.
  - [48] Shi W, Caballero J, Huszár F, et al. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network[J]. 2016, abs/1609.05158.

