

实验 7-Face Recognition

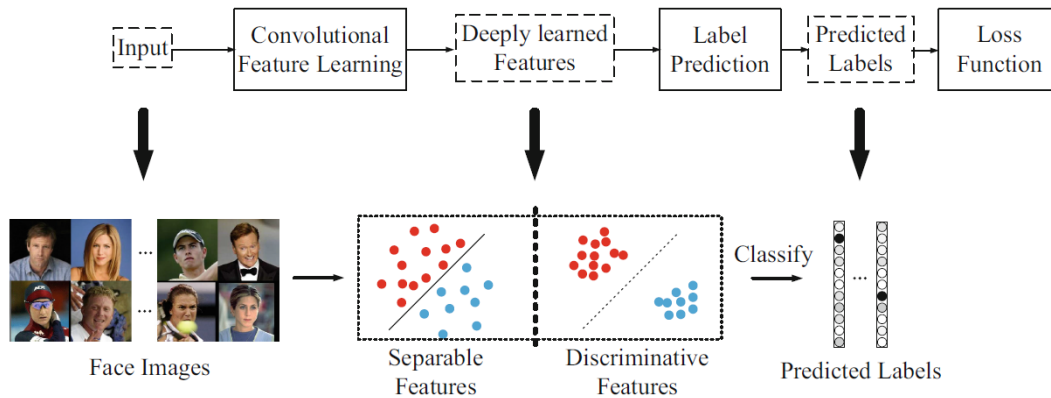
一、实验目的

- ✓ 学习深度学习环境的搭建
- ✓ 实现基于 center loss 的人脸识别算法

二、实验原理

2.1 人脸识别框架

基于深度学习的人脸识别算法流程如图所示：



其中特征提取网络可选择 ResNet18、ResNet50 等网络结构。

本次实验网络模型是 ResNet，如下表所示，共有 5 组卷积。第一组卷积的输入大小是 224x224，第五组卷积的输出大小是 7x7，缩小了 32 倍。每次缩小 2 倍，总共缩小 5 次。

| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|------------|-------------|---|---|---|--|--|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| conv2_x | 56×56 | 3×3 max pool, stride 2 | | | | |
| | | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ |
| conv3_x | 28×28 | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$ |
| conv4_x | 14×14 | $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$ |
| conv5_x | 7×7 | $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | 1.8×10^9 | 3.6×10^9 | 3.8×10^9 | 7.6×10^9 | 11.3×10^9 |

2.2 Center Loss 简介

center loss 能够直接对样本特征之间的距离进行约束。center loss 添加的约束是，特征与同类别的平均特征的距离要足够小，这要求同类特征要接近它们的中心点，公式如下表示：

$$\mathcal{L}_C = \frac{1}{2} \sum_{i=1}^m \| \mathbf{x}_i - \mathbf{c}_{y_i} \|_2^2$$

其中 \mathbf{x}_i 表示第 i 个样本的提取特征， \mathbf{c}_{y_i} 表示样本 i 所对应的类别的所有样本特征的平均特征，或者说同类别样本特征的中心点， m 表示样本个数。

center loss 有一个难点，如何计算 \mathbf{c}_{y_i} 。通过计算同一类别所有样本的特征，然后求平均值，这种方法是不切实际的，因为我们的训练样本非常庞大。作者另辟蹊径，使用 mini-batch 中的每个类别的平均特征近似不同类别所有样本的平均特征。在梯度下降的每一次迭代过程中， \mathbf{c}_j 的更新向量是：

$$\Delta \mathbf{c}_j = \frac{\sum_{i=1}^m \delta(y_i = j) \cdot (\mathbf{c}_j - \mathbf{x}_i)}{1 + \sum_{i=1}^m \delta(y_i = j)}$$

其中 $\delta(y_i = j)$ 是指示函数，当 j 是类别 y_i 时，函数返回 1，否则返回 0。分母的 1 是防止 mini-batch 中没有类别 j 的样本而导致分母为 0。论文中设置了一个 \mathbf{c}_j 的更新速率参数 α ，控制 \mathbf{c}_j 的更新速度。

训练的总损失函数是：

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_S + \lambda \mathcal{L}_C \\ &= - \sum_{i=1}^m \log \frac{e^{W_{y_i}^T \mathbf{x}_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T \mathbf{x}_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^m \| \mathbf{x}_i - \mathbf{c}_{y_i} \|_2^2 \end{aligned}$$

2.3 LFW 数据集简介

LFW (Labeled Faces in the Wild) 人脸数据库是由美国马萨诸塞州立大学阿默斯特分校计算机视觉实验室整理完成的数据库，主要用来研究非受限情况下的人脸识别问题。LFW 数据库主要是从互联网上搜集图像，一共含有 13000 多张人脸图像，每张图像都被标识出对应的人的名字。

这个集合被广泛应用于评价 face verification 算法的性能。



(a) Face images in LFW

三、实验环境

ubuntu 系统
python 3.6
pytorch 0.4
torchvision 0.2
matplotlib
opencv-python
requests
scikit-learn
tqdm

四、实验步骤

1、搭建环境

- 安装 Anaconda，新建虚拟环境
- 安装所需的第三方库
例：pip install torch==0.4.0 torchvision==0.2.0

2、补充 trainer.py 里 loss 计算的代码

- 调用 pytorch 函数，计算 softmax loss (cross-entropy loss)

```
torch.nn.functional.cross_entropy(input, target, weight=None, size_average=None, ignore_index=-100, reduce=None, reduction='mean')
```

此程序中与 input 对应的变量为 *logits*，与 target 对应的变量为 *targets*
- 计算 cross-entropy loss 与 center loss 加权求和后总的损失函数。此程序中与加权系数 λ 对应的变量为 *self.lamda*

3、训练模型

- 运行 main.py 来训练模型
python main.py --arch resnet18 --batch_size 64 --epochs 50
该命令会自动完成 LFW 数据集下载、模型训练、模型验证等操作。

4、人脸验证

- 用训练好的模型做人脸验证。对于 images 文件夹中给定的四张人脸图像，计算其两两之间的特征向量的距离。
python main.py --verify-model logs/models/epoch_50.pth.tar --verify-images images/Taylor1.jpg, images/Taylor2.jpg
运行该指令会输出两张图片特征之间的 distance。

5、调整参数

- 尝试改变 main.py 中的训练参数，如 batch size、epochs 等，观察模型性能的变化。

五、参考文献

Wen Y, Zhang K, Li Z, et al. A discriminative feature learning approach for deep face recognition[C]//European conference on computer vision. Springer, Cham, 2016: 499-515.