2.5 CNN网络实战技巧

2.5.1 迁移学习(Transfer Learning)

- **利用数据、任务或模型之间的相似性**， 都是分类问题
- **在旧的领域学习过或训练好的模型**
- **应用于新的领域进行训练**

2.5.1.2 微调

fine tuning,即微调

- 调整模型参数不需要过多调整
- 调整模型结构，微微调整
- Pre-trained：预训练模型
- fine tuning:微调之后的模型

2.5.1.3 过程

- 1、确定当前场景任务B，修改原始模型结构（修改后面全连接层）
- 2、确定B任务的数据大小
  - B任务数据量大，可以放开A模型的所有训练，A结构+修改的全连接层一起训练（A模型有已训练好的参数）
  - B任务数据量小，将A模型冻结掉不去训练，只训练全连接层

3.1 迁移学习案例

- 读取本地的图片数据以及类别
  - keras.preprocessing.image import ImageDataGenerator提供了读取转换功能
  - train_generator = ImageDataGenerator()
    - flow(x, y, batch_size):
    - flow_from_derectory(
      - directory=path,# 读取目录
        - 要求数据存储必须

```
data/
    train/
        dogs/
            dog001.jpg
            dog002.jpg
            ...
        cats/
            cat001.jpg
```

```
            cat002.jpg
            ...
    validation/
        dogs/
            dog001.jpg
            dog002.jpg
            ...
        cats/
            cat001.jpg
            cat002.jpg
            ...
```

        - target_size=(h,w),# 目标形状
        - batch_size=size,# 批数量大小
        - class_mode='binary', # 目标值格式，One of
        - "categorical", "binary", "sparse",
            - "categorical"： 2D one-hot encoded labels
            - "binary" will be 1D binary labels
    - ImageDataGenerator(rescale=1.0 / 255.0)
    - train_generator.flow_from_directory(self.train_dir, target_size=self.image_size, batch_size=self.batch_size, class_mode='binary', shuffle=True)
- 模型的结构修改（添加我们自定的分类层）
    - notop模型：
    - **是否包含最后的3个全连接层（whether to include the 3 fully-connected layers at the top of the network）。用来做fine-tuning专用，专门开源了这类模型。**
    - VGG16(weights='imagenet', include_top=False)
    - ## GlobalAveragePooling2D:全局池化
        - 使用前提：做迁移学习不需要大量的参数去微调
- freeze掉原始VGG模型
    - 让VGG结构当中的权重参数不参与训练,只训练我们添加的最后两层全连接网络的权重参数
- 编译以及训练和保存模型方式
    - compile
    - ModelCheckpoints
    - fit_generator():可以使用callbacks中的ModelCheckpoints
- 输入数据进行预测
    - model.predict

4.1 项目演示

4.1.2 项目结构

- 数据采集层：数据收集标注
- 深度模型层：YOLO，SSD模型，模型导出，Serving 部署
- 用户层：前端交互，（web后台）对接部署的模型

4.2 目标检测任务描述

4.2.1 目标检测算法分类

- 两步走的目标检测：
    - 1、先找出候选的一些区域
    - 2、对区域进行调整，分类
- 端到端的目标检测：
    - 采用一个网络一步到位
    - 输入图片，输出有哪些物体，物体在什么位置

4.2.2 目标检测的任务

- 输入：图片
- 输出：物体类别，物体的位置坐标
- xmin, ymin, xmax, ymax：物体位置的左上角、右下角坐标

4.2.3 目标定位的简单实现思路

- 增加一段全连接输出4个位置，做损失计算

4.2.4.1 两种Bounding box名称

- Ground-truth bounding box：GT 图片真实目标位置（标记结果）
- Predicted bounding box：预测的框位置
- 掌握分类
- 分类与定位：图片中只有一个物体需要检测
- 目标检测：图片由多个物体需要检测

4.3 R-CNN

- 对于多个目标的情况，就不能以固定个数输出物体的位置值

4.3.1 目标检测-Overfeat模型

4.3.1.1 滑动窗口

- 首先定义若干个大小窗口，K个
- K中每个窗口都要滑动图片，每个窗口滑动M词
- K x M

4.3.1.2 Overfeat模型总结

- 暴露破解方式

- 计算消耗太大

4.3.2 目标检测-R-CNN模型

- 在CVPR 2014
- 候选区域方法（region proposal method）：提供了额物体检测的一个重要思路
- RCNN步骤：
  - 1、对于一张图片，找出默认2000个候选区域
  - 2、2000个候选区域做大小变换，输入AlexNet当中，得到特征向量
    - [2000 , 4096]
  - 3、经过20个类别的SVM分类器，对于2000个候选区域做判断，得到[2000, 20]得分矩阵
  - 4、2000个候选区域做NMS，取出不好的，重叠度高的一些候选区域，得到剩下分数高，结果好的框
  - 5、修正候选框，bbox的回归微调

- **4.3.2.2 候选区域（Region of Interest（ROI））得出（了解）**
  - **SelectiveSearch**在一张图片上提取出来约2000个侯选区域
  - 由于长宽不定，不能直接输入alexNet
  - 2000个候选区域做大小变换
- **4.3.2.4 特征向量训练分类器SVM**
  - **R-CNN选用SVM进行二分类。假设检测20个类别，那么会提供20个不同类别的SVM分类器，每个分类器都会对2000个候选区域的特征向量分别判断一次，这样得出[2000, 20]的得分矩阵**
  - 猫分类对：2000个候选区域做判断，得到2000个属于猫的类别
  - 狗分类对：2000个候选区域做判断，得到2000个属于猫的类别
  - ....
  - .....
  - [2000, 20]
- **4.3.2.5 非最大抑制（NMS）**
  - 目的
    - **筛选候选区域，目标是一个物体只保留一个最优的框，来抑制那些冗余的候选框**
  - RCNN预测2000个候选框，得到3个（比如有3个ground truth）比较准确的候选框
  - 迭代过程
    - 1、对于所有的2000个候选区域得分进行概率筛选，0.5
      - 2000—》5个
    - 2、剩余的候选框
      - 对于每个候选框找到自己对应GT
    - 3、第一轮：对于右边车辆，假设B是得分最高的，与B的IoU＞0.5删除。现在与B计算IoU，DE结果＞0.5，剔除DE，B作为一个预测结果 第二轮：对于左边车辆，AC中，A的得分最高，与A计算IoU，C的结果＞0.5，剔除C，A作为一个结果
    - 最终结果：理想状态，每一个Ground truth都有一个候选框预测

- **4.3.2.6 修正候选区域**
    - 为了让候选框标注更准确率，去修正原来的位置
    - A 是候选框， G是目标GT框
    - 让A与G做回归训练，得到**四个参数**
- RCNN输出：一张图片预测一个X候选框，x x w = y_locate
    - y_locate:是真正算法输出的位置

4.3.3 检测的评价指标

- **4.3.3.1 IoU交并比**
    - IOU交并比：0 ~1 之间的值
    - 位置的考量
- **4.3.3.1 平均精确率（mean average precision）map**
    - 物体检测的：分类准确的考量
    - 定义：多个分类任务的AP的平均值
        - mAP = 所有类别的AP之和 / 类别的总个数
        - (AP1+AP2+。。。。+AP20)/20
        - 对于每个类别计算AP的值
            - 1、对于猫类别：候选框预测是猫类别的概率做一个排序，得到候选框排序列表（比如8个）
            - 2、**对于猫当中候选框排序列表（比如8个）进行计算AP**
            - 3、最终得到20个类别，20个AP相加
- 总结：**在VOC2007数据集上的平均精度map达到66%**
    - 训练时间长，模型多个模型
    - 处理速度慢
- # 4.3.8 改进-SPPNet
    - 1、 图片输入到网络先得到一个feature map
    - 2、原图中通过SS得到的**候选区域直接映射feature map中对应位置**
        - 左上角的点：
            - **x'=[x/S]+1**
        - 右下角的点：
            - **x'=[x/S]−1**
        - 论文当中S= 16=2 x 2 x 2 x 2
        - 原图：特征图中xmin‘,ymin’ = [xmin/16]+1,ymin/16+1
        - 特征图：xmax‘,ymax’ = [xmax/16]-1,ymax/16-1
    - 3、映射过来的（假如还是2000个）候选区域的特征，**经过SPP层(空间金字塔变换层)**，S输出固定大小的特征向量
    - **3.3.1.2 spatial pyramid pooling**

        - **候选区域的特征图转换成固定大小的特征向量**
        - spp layer会将每**一个候选区域分成1x1，2x2，4x4三张子图**
        - (16+4+1)x256=21x256=5376
        - Spatial bins（空间盒个数）：1+4+16=21
    - 缺点：
        - 分阶段训练网络：选取候选区域、训练CNN、训练SVM、训练bbox回归器, SPPNet反向传播效率低

4.4 Fast R-CNN

4.4.1 Fast R-CNN

改进的地方：

- 提出一个RoI pooling
- 分类是用softmax计算：K个类别加上”背景”类
- 与SPPNet一样的地方
    - 首先将整个图片输入到一个基础卷积网络，得到整张图的feature map
    - 将选择性搜索算法的结果region proposal（RoI）映射到feature map中
- **4.4.1.1 RoI pooling**
    - 为了减少计算时间并且得出固定长度的向量
    - 使用一种4 * 4= 16空间盒数
    - **因此Fast R-CNN要比SPPNet快很多也是因为这里的原因**
- 训练会比较统一：废弃了svm以及sppnet
    - roi pooling layer + softmax
    - **4.4.2 多任务损失-Multi-task loss**
    - **平均绝对误差（MAE）损失即L1损失**+交叉熵损失
- 缺点
    - 使用Selective Search提取Region Proposals，没有实现真正意义上的端对端，**操作也十分耗时**

4.5 Faster R-CNN

- 候选区域筛选融合到网络当中
    - 四个基本步骤（候选区域生成，特征提取，分类，位置精修）终于被统一到一个深度网络框架
- **区域生成网络（RPN）+Fast R-CNN**
- RPN替代了SS选择性搜索算法
    - RPN网络用于生成region proposals
    - **通过softmax判断anchors属于foreground或者background**

    - **bounding box regression修正anchors获得精确的proposals**
    - 得到默认300个候选区域给roi pooling继续后面fast rcnn的步骤

4.5.2 RPN原理

- **用n×n(默认3×3=9)的大小窗口去扫描特征图**得到K个候选窗口
- 每个特征图中像素对应的9个窗口大小?
- 三种尺度{ 128，256，512 }， 三种长宽比{1:1，1:2，2:1}
- 3 x 3=9不同大小的候选框
    - 窗口输出 [N , 256]——>分类：判断是否是背景
    - 回归位置：N个候选框与自己对应目标值GT做回归，修正位置
    - 得到更好的候选区域提供给ROI pooling使用

4.5.3 Faster RCNN训练

- RPN训练：
    - 分类：二分类，softmax，logisticregression
    - 候选框的调整：均方误差做修正
- Fast RCNN部分的训练：
    - 预测类别训练：softamx
    - 还有预测位置的训练:均方误差损失
- 样本准备：正负anchors样本比例：1:3
- 优点
    - 提出RPN网络
    - 端到端网络模型