机器学习——任务二（样本不均衡问题）

班　　级：　硕 19069

组　　长：　赵嘉文　　ZF1906249

小组成员：　孙　伟　　ZF1906261

王　凯　　ZF1906246

魏家琦　　ZF1906270

杨文雪　　ZF1906248

一、问题描述

数据集中各个危险品类别的数据量是不均衡的，如何解决样本不均衡条件下模型训练的类别偏好问题是一个热点。训练集中的危险品包括带电芯充电宝和不带电芯充电宝两个类别。比例为1:10（带电芯充电宝：不带电芯充电宝，500：5000）。划分训练集训练模型，划分测试集计算测试集在模型上的mAP。

二、解决方案

1. 样本不均衡问题的解决方案

该问题中，带电芯充电宝和不带电芯充电宝的比例为1:10（500 : 5000），两类样本类别分布严重不均衡。若不对样本进行处理，样本少的一类由于样本特征少，使得不容易发现特征规律，很容易出现过拟合问题，导致模型的准确性会很差。

解决样本不均衡问题的方法有很多，如过抽样、欠抽样以及改变两类样本的权重等。在该问题中，我们选择了过抽样方法：增加分类中少数类样本的数量来实现样本均衡，即增加带电芯充电宝的样本数量。由于在实际情况中，两类样本比例达到1:3左右即可认为样本达到均衡，故在实验中，我们选择的方法是：对带电芯充电宝进行左右、上下翻转，将原来500个样本扩充到2000个。扩充后，带电芯充电宝和不带电芯充电宝的比例变为1:2.5， 大大减小了两者的比例，使得样本几近达到均衡。

2. 使用模型简介

用于目标检测的算法有很多，如：RCNN、YOLO、SSD等，通过对比，由于SSD模型运行速度较快，检测精度较高，最终我们选择了使用SSD模型。

2.1 SSD模型原理简介

SSD模型将图像切分成N个区域，对每个区域进行单目标检测，并汇总所有的单目标检测结果。SSD采用多尺度特征图用于检测，其中，小的特征图负责检测大目标，大的特征图用来检测小目标；除此之外，SSD在每个单元设置了不同尺度和长宽比的先验框，对于每个单元的每个先验框，都会输出一套独立的检测值，对应一个边界框。在预测过程中，置信度最高的那个类别就是边界框所属的类别。SSD采用VGG16作为基础模型，然后在VGG16的基础上新增了卷积层来获得更多的特征图以用于检测。

2.2 训练过程

（1） 先验框匹配

首先确定图片中的ground truth（真实目标）与哪个先验框匹配，与之匹配的先验框所对应的边界框将负责预测该真实目标。其中，匹配原则有两点：首先，对于图片中每个ground truth，找到与其IOU（IOU = 预测边框与真实边框交集/预测边框与真实边框并集）最大的先验框进行匹配；其次，对于剩余的未匹配的先验框，若与某个ground truth的IOU大于某个阈值（一般为0.5），那么该先验框也与这个ground truth进行匹配，若有多个ground truth与某个先验框IOU都大于阈值，则先验框只与最大的先验框进行匹配，若某个先验框没有与之匹配的ground truth，则该先验框与背景匹配。将与ground truth匹配的先验框称为正样本，与背景匹配的先验框称为负样本。

（2） 损失函数

SSD的损失函数定义为位置误差(locatization loss, loc) 与置信度误差(confidence loss, conf)的加权和：

$$L(x,c,l,g)=\frac{1}{N}(\mathrm{L_{conf}}(x,c)+\ \alpha\mathrm{L_{loc}}(x,l,g))$$

$$L_{loc}\ (x,l,g)=\sum_{i\in pos}^{N}\sum_{m\in\{cx,cy,w,h\}}x_{ij}^{k}\,smooth_{L1}(l_i^m-g_j^m)$$

$$L_{conf}(x,c)=-\sum_{i\in Pos}^{N}x_{ij}{}^{p}\log(C_i^p)\ -\sum_{i\in Neg}log(C_i{}^0)$$

$$C_i^p=\frac{exp(c_i^p)}{\sum_p exp(c_i^p)}\quad,\quad smooth_{L1}(x)=\begin{cases}0.5x^{\ 2}, & if\ \ |x|<1\\ |x|-0.5\ , & otherwise\end{cases}$$

其中，N 是先验框的数量，i 为先验框序号，j 为真实框序号，p 为类别序号；$x_{ij}{}^{p}\in\{1,\ 0\}$ 是一个指示参数，取 1 时表示第 i 个先验框与第 j 个 ground truth 匹配，且该 ground truth 的类别为 p；l 为先验框对应边界框的位置预测值，g 为 ground truth 的位置参数，其中，位置误差采用$smooth_{L1}\ loss$；$c_i{}^{p}$表示第 i 个先验框对应类别 p 的预测概率；权重系数 α 通过交叉验证设置为 1。

2.3 预测过程

对于每个预测框，首先根据类别置信度确定其类别与置信度值，并过滤掉属于背景的预测框。然后根据置信度阈值（如 0.5）过滤掉阈值较低的预测框。将留下的预测框进行解码，根据先验框得到其真实的位置参数，解码之后，根据置信度进行降序排列，仅保留 top-k 个预测框。之后进行 NMS 算法，过滤掉重叠度较大的预测框，最后剩余的预测框即为检测结果。

3. 样本的预处理

在所给样本中，类别不止“带电芯充电宝”和“不带电芯充电宝”两种，还有很多其他种类，所以我们需要对所给样本进行处理，即删除其他种类的标注信息。

除此之外，由于网络上关于 ssd 模型的训练代码几乎都是用的 VOC 数据集，所以我们需要将数据集的格式转换为 VOC 的标准格式，以便于模型的训练。VOC 数据集文件夹结构分为：Annotations、ImageSets、JPEGImages、SegmentationClass 和 SementationObject，其中，我们要用到的文件夹是 Annotations、ImageSets 和 JPEGImages。Annotations 文件夹存放的是 xml 文件，用以标注对应图片的基本信息；ImageSets 文件夹的 Main 目录下存放的是 4 个 txt 文件，分别说明了训练集的图片文件名、验证集的图片文件名、训练和验证的图片文件名以及测试集的图片文件名。

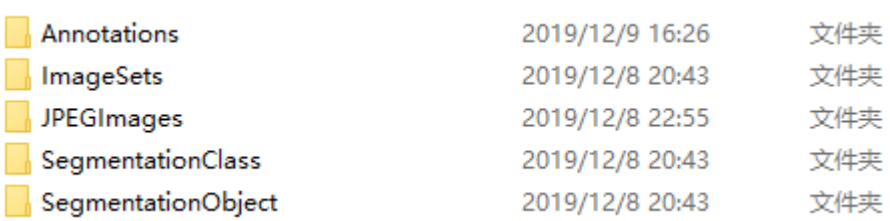

图 1 VOC 文件格式

样本的预处理过程如下：首先，对小比例样本进行扩充，即对不带电芯充电宝分

别进行左右翻转和上下翻转，由原来的 500 个样本扩充至 2000 个，翻转时，对说明文件中标注的坐标也进行相应的转换；其次，删除说明文件中其他类别的标注，只剩下“带电芯充电宝”和“不带电芯充电宝”两个类别；最后，将数据集转化为 VOC 格式的数据集，将训练集和测试集比例设置为 8：2，即训练集样本 5600 张，测试集 1400 张。至此，已完成样本的预处理过程。

三、实现结果及其分析

说明：实验代码在 https://github.com/amdegroot/ssd.pytorch 基础上进行修改。该实验基于 pytorch1.2.0，结合 cuda 进行训练，使用 visdom 可视化 loss 下降过程。

1. 数据训练过程中 loss 变化如下：

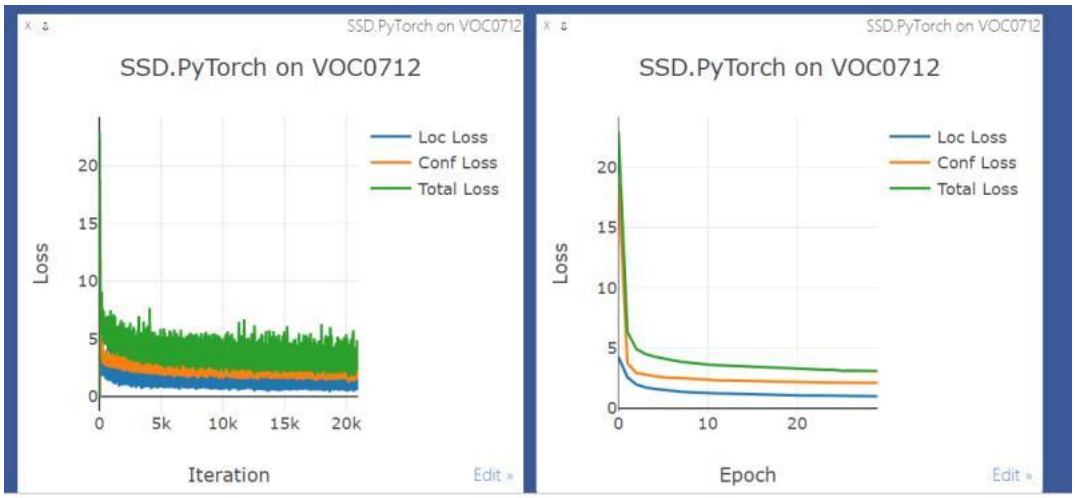

图 2 数据训练过程中 loss 的变化

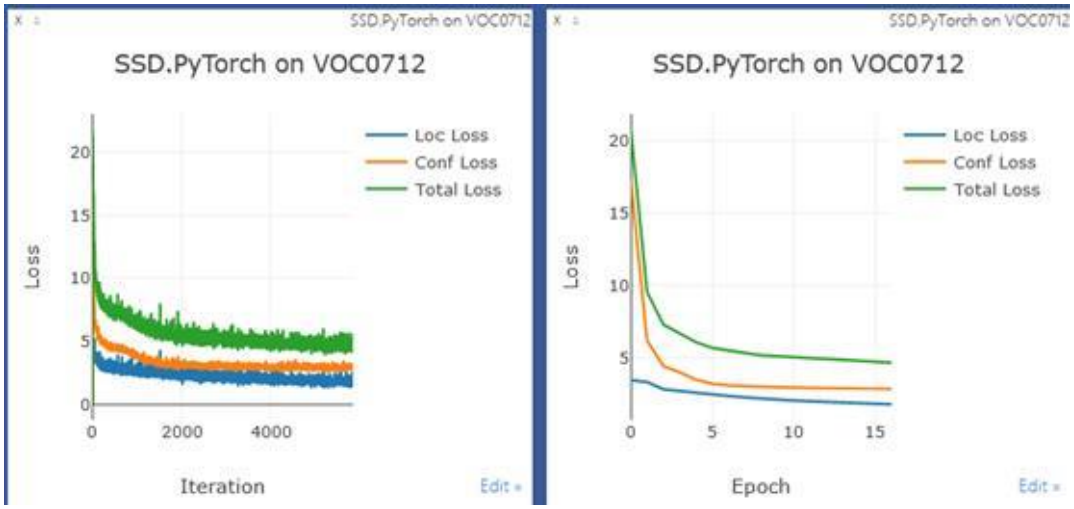

图 3 数据训练过程 loss 的变化

在图 2 的训练过程中，学习率的大小为$8 * e^{-5}$，迭代次数为 21000。通过图片可以看出，随着迭代次数增加，loss 值几乎达到收敛，值的大小稳定在 2-3 之间。在此组参数设置下，mAP 可以达到 0.7730。

在此之前，我们尝试过多种学习率和迭代次数的组合。学习率过小，下降的速度缓

慢；学习率过大，会出现震荡，学习率的选择需要我们不断地进行尝试，才能获得相对较好的效果。通过图 3 loss 下降曲线可以看出，在学习率一定的情况下，迭代 4000 次左右也趋于稳定，但是 loss 值却较大，在 6-7 之间，最终的测试结果也不太理想。这说明，虽然迭代次数到达一定数值时，loss 的变化很缓慢，但是，随着迭代次数的增加，loss 仍然往小的方向变化，迭代次数越多，最终测试效果就越好。由于电脑配置以及时间的关系，我们所尝试的迭代次数最多为图 2 中的 21000，其所得到的结果是目前为止最好的结果。

2. 测试结果（测试代码用的老师给的代码）

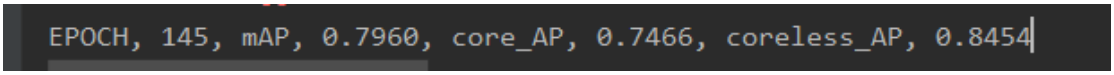

图 4 一个 EPOCH 测试结果 mAP

图 5.1 不带电芯充电宝类别对带电芯充电宝识别结果部分输出

图 5.2 不带电芯充电宝类别对不带电芯充电宝识别结果部分输出

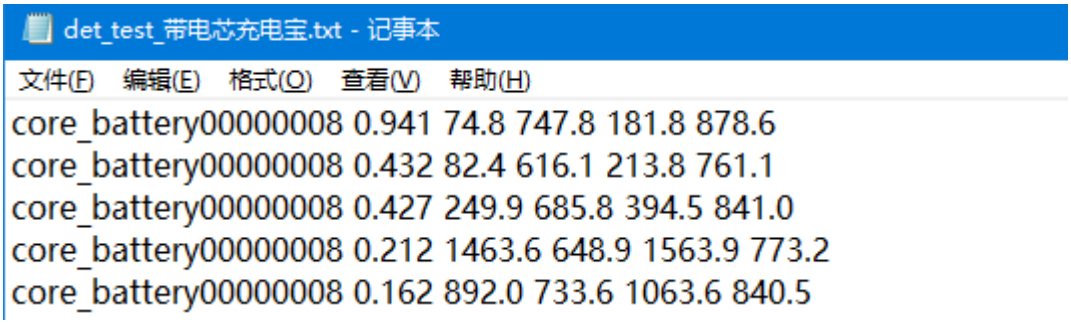

图 5.3 带电芯充电宝类别对带电芯充电宝识别结果部分输出

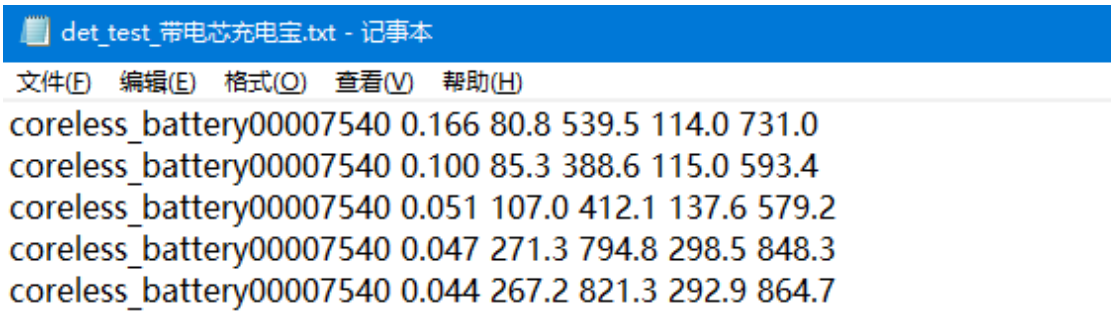

图 5.4 带电芯充电宝类别对不带电芯充电宝识别结果部分输出

如图 4，可以看到，在学习率为$8*e^{-5}$，迭代次数为 21000，loss 值在 2-3 之间时，mAP 为 0.7730，其中，带电芯充电宝（少数样本）的 AP 为 0.7313，不带电芯充电宝（多数样本）的 AP 为 0.8146。

很显然，不带电芯充电宝比带电芯充电宝的测试结果要好很多，说明样本个数越多效果越好，如果能够将样本比例扩充到 1:1，将会有更好的结果。但由于时间关系，我们没有继续对不带电芯充电宝的样本数量进行扩充。

四、遇到的问题及解决方案

1. 第一次训练的模型测试得到的 mAP 仅为 0.0165，最主要的原因是当时样本中已经将其他类别的标注删掉了，只剩下“带电芯充电宝”和“不带电芯充电宝”两类，但是 ssd.py 文件的代码中的类别数目没改，还是之前的 21，改掉之后，mAP 直接提高到 0.69。
2. 在搭配环境过程中也遇到了一些问题，根据显卡版本安装 pytorch 1.3.1 和 cuda10.1 后，虽然 torch.cuda.is_available()返回 True，但代码仍然由于 cuda 问题不能运行，重新安装了 pytorch1.2.0 后解决了此问题。
3. 代码运行过程中遇到的问题
   在改完基本的代码后，第一次运行时，遇到了很多问题，比如发现了所给样本中除了“带电芯充电宝”和“不带电芯充电宝”两类外，还有其他类别、如何使用 visdom 可视化 loss 下降过程以及其他各种大大小小的问题，我们通过查阅资料、讨论等方式，逐步解决这些问题，最终代码成功开始运行。

五、小组成员分工

首先，全体成员参与讨论样本不均衡问题的解决方案以及模型的选取。

王凯、杨文雪、赵嘉文负责代码的实现及模型的训练。其中，王凯负责小样本的扩充、数据集格式的转换以及文件路径的读取，杨文雪和赵嘉文负责参数的修改和模型的训练以及解决在运行过程中出现各种 bug。

魏家琦负责报告的撰写。

孙伟负责 ppt 的制作和答辩。