声音事件检测任务

1 提交文件简要说明

本次实验使用超算 hpc 进行，代码改动集中在 `model.py`，并未对原有其他代码框架进行大幅更改。数据预处理根据 `data` 目录下的 `prepare_data.sh` 文件进行，提取的数据保存在该目录下，并未从 `stu168` 拷贝原始数据集（可以软链接，但并不需要）。按照原有代码框架脚本，实验结果默认保存在 `experiments/Crnn` 目录下，超算运行日志保存在 `slurm_logs` 文件夹下。

2 任务理解

2.1 声音事件检测模型理解

2.2 弱监督情况下进行时间轴预测的难点

一个是弱标签数据，即只有声音事件的分类，没有时间信息在分类任务中，数据必须是带标注的，但对于声音事件检测任务而言，大多数情况下，我们能处理的数据都是不带标签的，或是标签不精确的，我们将这类数据称之为弱标注 (weakly label) 数据。

现行开源数据集中，强标注数据是非常少的，这使得我们要想训练超大规模的检测模型只能使用弱标注和无标注数据。弱标注下的声音事件检测任务可以抽象成多实例学习 (multiple instance learning) 任务。将含有少量标注的大段数据看作一个大包，只知道这个包中含有某类事件，但不知道这些类的事件在数据包中的具体位置。声音事件检测的任务就是检测出这些事件的类别和事件的发生位置。

目前声音事件检测存在许多挑战，例如：

1、音事件有非常不同的声学特征，有些声音很短，比如枪声，有些声音很长，比如说话声等等。

2、在声音事件检测的实际应用中，需要检测的声音距离麦克风很远，导致麦克风接收到的目标事件的声压级低于环境中发生的其他声音的声压级，增加了检测的难度。

3、生活中发生的声音事件通常是多音的，意味着多个声音事件会在同一时间发生，也增加了检测的难度。

4、音频数据量少，并且标注困难，耗时大。导致目前音频数据集无标签的数据多，有标签的数据很少。

2.3 Baseline 设计的原理

整个 CRNN 网络结构包含三部分，从下到上依次为：

CNN（卷积层），使用深度 CNN，对输入图像提取特征，得到特征图；RNN（循环层），使用双向 RNN（BLSTM）对特征序列进行预测，对序列中的每个特征向量进行学习，并输出预测标签（真实值）分布；CTC loss（转录层），使用 CTC 损失，把从循环层获取的一系列标签分布转换成最终的标签序列。CRNN 算法最大的贡献，是把 CNN 做图像特征工程的潜力与 LSTM 做序列化识别的潜力，进行结合。它既提取了鲁棒特征，又通过序列识别避免了传统算法中难度极高的单字符切分与单字符识别，同时序列化识别也嵌入时序依赖（隐含利用语料）。

2.4 Crnn 模型实现

用于声音事件检测的通用网络体系结构是卷积递归神经网络（CRNN），CNN 做为特征提取器，RNN 可以依据近乎无限长的上下文信息做出逐帧的决策。下图展示了一个由三个卷积块组成的 CRNN 体系结构，CNN

后面接两个递归层和两个全连接层。

3 实验

3.1 数据预处理

将脚本 cp 到个人 hpc 目录后运行 `data` 文件夹下的 `prepare_data.sh` 脚本，将根据 `stu168` 用户下的原始数据提取用于本次训练的后续相关数据。需要使用单线程数据处理脚本，并对 librosa 版本降级，从而成功提取数据。

3.2 实验结果和分析

`baseline.yaml` 中未提供大量的参数供调整，本任务的重点在于 Crnn 的模型搭建和结构调整，故本部分重点在于总结不同 Crnn 结构下的实验结果，并展示模型结构改进对评测指标的影响。本次任务最终提交的 `model.py` 即保留了最优情况下的模型参数配置，其脚本则为 `run.sh` 及相关的配置文件。

本次实验最开始我的直观想法是直接将 Crnn 所需要的神经网络模块，包括但不限于 fully connect, convolution, lstm, batchnorm 等融合在一起，不对参数进行细致的配置，直接使用 relu, sigmoid 等激活函数，在成功搭建后的初步实验结如下：

Metrics	f_measure	precision	recall
event_based	0.00743036	0.00666628	0.00990206
segment_based	0.196863	0.310598	0.161337
tagging_based	0.449451	0.568862	0.387041
mAP	0.5058306091886279		

表 1: 简单 Crnn 结构：双层卷积

初步实验成功之后，进行模型结构调整，增加卷积层和 BatchNorm 层的个数，使用两种不同的池化方式，结果如下：

Metrics	f_measure	precision	recall
event_based	0.00831543	0.0116714	0.00656456
segment_based	0.226929	0.42916	0.156146
tagging_based	0.553278	0.630679	0.511583
mAP	mAP: 0.5788640257440317		

表 2: 改进 Crnn 结构：三层卷积、batchnorm、两种 pooling

按照 slides 给出模型结构进行最终改进（此即提交文件中的模型结构），结果如下：

Metrics	f_measure	precision	recall
event_based	0.0108622	0.0139943	0.00960896
segment_based	0.231898	0.418706	0.163764
tagging_based	0.582452	0.628041	0.565621
mAP	mAP: 0.6098265732798768		

表 3: 最终 Crnn 结构：模型结构保留在 model.py 中

模型结构确定，继续进行参数调整，包括但不限于各个卷积层的 in_channels, out_channels, LSTM 网络的隐层维度、层数（即提交文件中的默认参数），主要是增大卷积层 channel 的个数和 LSTM 网络隐层向量的维度，结果如下：

Metrics	f_measure	precision	recall
event_based	0.00938473	0.0115726	0.00811011
segment_based	0.240401	0.423576	0.172586
tagging_based	0.633284	0.637923	0.634307
mAP	mAP: 0.6330678570776226		

表 4: 最终 Crnn 参数：模型参数是 model.py 中的默认参数