第三次大作业实验报告

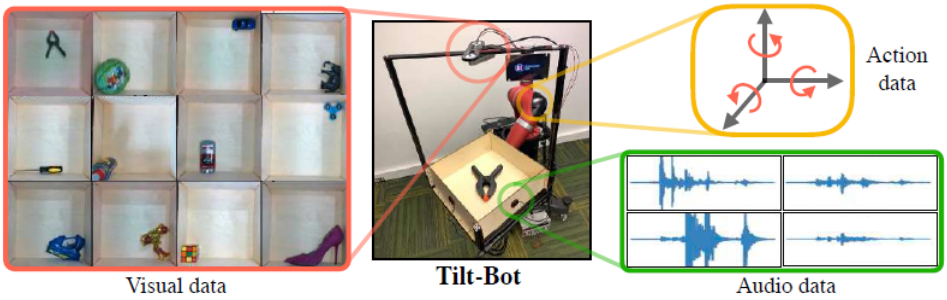
zxdclyz duskmoon314 BobAnkh

December 2020

1 原理介绍

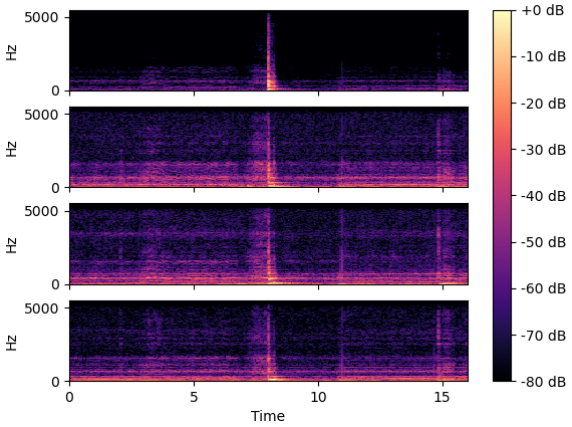
1.1 基本思路

本次实验要求我们根据给出的视频和音频的数据，完成音频分类和视频音频匹配的任务。音频数据为四个壁上的麦克风采集到的声音，我们首先对声音数据进行短时傅里叶变换，将得到的四个特征图作为四个 channel，直接使用 CNN 进行图像分类。而对于匹配问题，我们的想法是根据音频提取出运动相关的一些信息，再与从视频中提取出的相同信息做相似，根据相似度使用 KM 算法进行匹配，具体选取的特征信息为运动方向和碰撞位置。在本次实验中我们对于视频信息的处理和特征提取是基于传统方法的，我们还设计了基于深度学习的方法（但未将其进行实现），此部分内容详见问题与不足部分。



1.2 任务一

首先对音频数据进行预处理，先对音频数据进行降采样（44100Hz→11000Hz），然后对其进行 STFT 操作以得到其频谱特征，STFT 的具体参数参考了数据集来源的论文 [1]。然后对于得到的四个特征图，我们将其当作四个 channel 拼接为一个 tensor，作为训练用的数据。



特别的，并不是所有音频数据都是一样长的，绝大多数的音频长度都是 4s，对于长度不到 4s 的音频，我们直接在其末尾使用 0 进行延拓 (padding)。

由于我们得到的训练数据是一个 4 通道的频谱图，我们可以将其当作一个图像分类任务使用 CNN 来解决，本次实验中我们使用 resnet 来解决任务一。

1.3 任务二、三

任务二与任务三除了匹配部分没有本质的区别，我们采用的主要思路为先利用类别信息进行分组，然后再在组内进行匹配。对于匹配特征的提取，我们训练了两个网络来分别根据声音判断物体的运动方向和碰撞的位置。事实上，我们认为只根据声音数据，并不能准确恢复出物体的运动方向，

因为缺少其开始时的位置信息，因此我们打算使用碰撞位置来作为关键信息进行匹配。但是通过实验发现，许多匹配错误是最终碰撞的位置相同引起的，因此我们加入了并不完全准确的运动方向信息（平均误差在 15 度）来辅助位置信息进行匹配，从实验结果上看这样的特征组合对匹配成功率有着较为显著的提升。

匹配任务是一个非常典型的二部图最大权匹配，我们采用经典的 Kuhn Munkres 算法（下称 KM 算法）来得到匹配结果。考虑到 KM 算法的时间复杂度较高，先对音频和图像进行较高准确度的分类后，再对每一类内的所有音频和图像信息进行匹配可以有效减少复杂度。如前文所述，音频数据经网络处理后得到“标签-位置-方向”的数据，对图像使用 OpenCV 处理得到类似的数据。我们将位置和方向拼成一个三维向量 $[x, y, \theta]$，并对两份数据的笛卡尔积的每一项都计算两个向量差的二范数的负数作为“匹配度”，即有

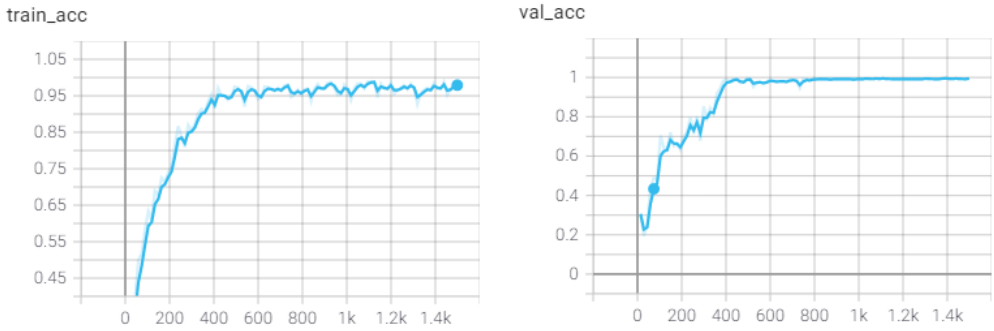
$$S = -\left[(x_1 - x_2)^2 + (y_1 - y_2)^2 + (\theta_1 - \theta_2)^2\right]$$

笛卡尔积使二部图中两部分完全连接，而二范数的负数是为了适应 KM 算法的最大权匹配的特点。从训练集的实验结果来看，这种方法的可以比较好的对两种数据进行匹配。

2 具体实现与结果分析

2.1 任务一

基本流程如前所述，先对音频进行了降采样（44100Hz→11000Hz），然后以参数 n_fft=510, hop_length=128，对音频进行了 stft，这里音频处理使用的库为 librosa，与作业二相同。之后我们使用了 Resnet-v2[2] 进行分类任务，通过多次尝试，我们选取了参数较少的 resnet20，使用 Cross Entropy loss，选用 Adma 作为优化器，设置初始学习率为 1e-3，并在 epoch=25, 50, 70 对学习率进行 0.1 的 decay。训练和验证的准确率绘制如下，训练了 100 个 epoch，最终训练集上准确率达到 98%，验证集上准确率达到 99%。

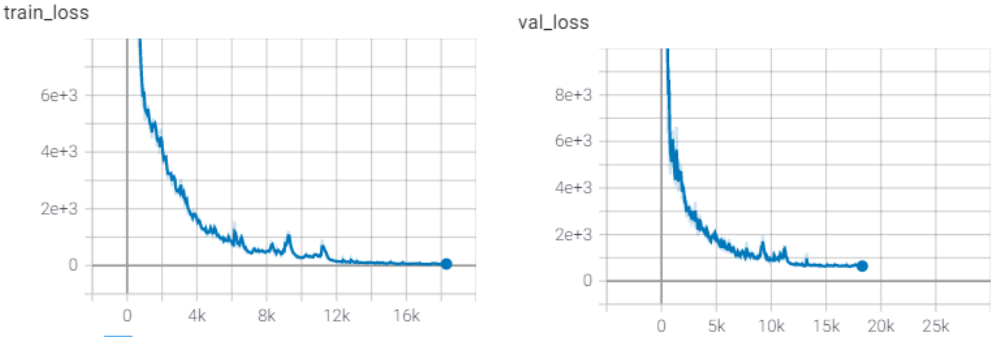


2.2 任务二、三

2.2.1 信息提取

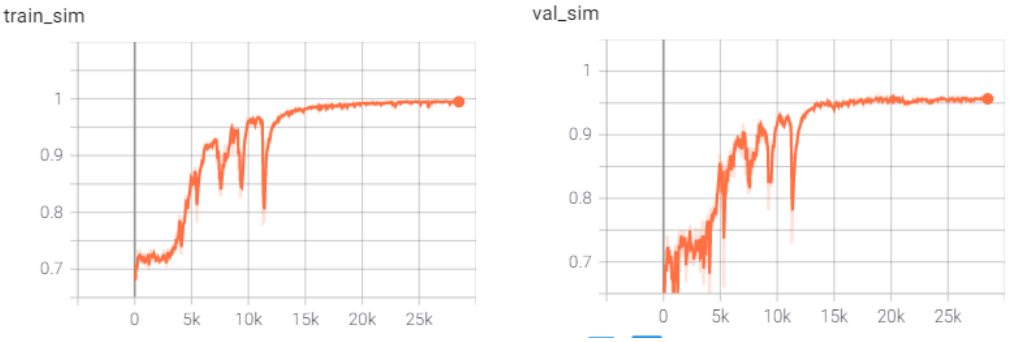
首先是对音频数据中信息的提取。我们的目标是获取最终的碰撞位置和运动角度的信息，处理的数据与任务一无异，因此我们同样使用 Resnet-v2 来完成。

对于位置判断，将 resnet 的输出设为 2 维，经测试我们选用了 resnet110，使用 MSE loss，选用 Adma 作为优化器，设置初始学习率为 1e-3，并在 epoch=200, 400 对学习率进行 0.1 的 decay。训练和验证的 loss 绘制如下，总共训练了 320 个 epoch，最终训练集上 loss 达到 46，验证集上 loss 达到 613，可见效果虽然不错但还是有一些偏差的。



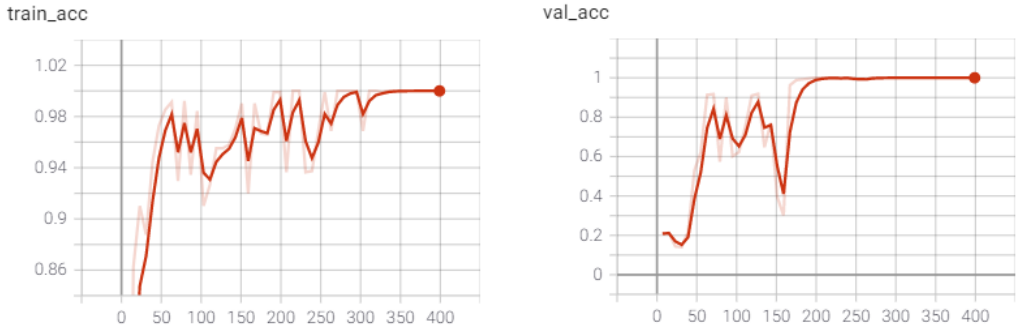
对于角度判断也相同，将 resnet 的输出设为 1 维即可，我们同样使用 resnet110，使用 MSE loss，选用 Adma 作为优化器，设置初始学习率为 1e-3，并在 epoch=200, 400 对学习率进行 0.1 的 decay。这里我们定义了两个角度差的 cos 值作为相似度来评判模型训练的好坏，经过 500epoch 的

训练，最终训练集上相似度达到 0.99，验证集上相似度达到 0.96，0.96 对应平均有 16 度的误差，如此大的误差已经会影响到我们进行匹配，因此我们没有选择角度作为唯一特征。



其次是对于图像的处理，为了与音频信息计算相似度，图像也同样需要碰撞位置和运动角度信息，同时还需要对图像进行分类来与音频先进行一个粗匹配。

对图像的分类十分简单，依然使用 Resnet-v2 完成。选取 resnet44，使用 Cross Entropy loss，选用 Adma 作为优化器，设置初始学习率为 1e-3，并在 epoch=20, 40 对学习率进行 0.1 的 decay。训练和验证的准确率绘制如下，仅训练了 50 个 epoch，训练集和验证集上准确率均达到 100%。



从图像中提取碰撞位置和运动角度信息的任务使用传统方法完成。使用掩膜图像计算了其各阶矩，图像中心位置可由一阶矩除以零阶矩得到，从而根据最后一张图像得到碰撞位置，由始末两张图像得到其运动角度。

2.2.2 匹配计算

我们以 Github 上开源的 KM 算法 [3] 为基础，并更改 KM 算法过程中权值的更新以适配负的权值，增加一个入口函数以适配网络给出的预测结果。由于 KM 算法本身的特性会尽可能的将二部图匹配起来，这便要求输入的音频数据和图像数据数量尽可能接近。同时为了避免无连接时的 0 导致的问题，也为了增大准确度，我们使用笛卡尔积将邻接矩阵完全填满。实验表明这种做法提升了匹配的结果。

在匹配权重的选择上，我们尝试了单独以位置或角度为权重，在训练集上的最好结果不超过 65%。而使用二者结合后的二范数显著提高了准确率。我们认为后一种方法可以拉开正确匹配与错误匹配间权重的差别，既提升了匹配的正确率，也减小了 KM 算法更新的次数。

由于数据集本身较小，在完全划分足够测试匹配准确度的验证集的情况下训练数据会减少很多，我们只能在整个数据集上直接划分出一小块来进行匹配的准确度测试，在划分出的 50 个样本中（每个类别各 5 个），我们达到了 48/50 的匹配准确度，并且我们发现匹配出错的原因主要在于音频分类不准确，所以我们有理由认为对于每个类别中较少的需要匹配的个体，我们设计的基于角度、位置信息的相似度计算算法可以较好地完成匹配任务。

3 问题与不足

3.1 目前的问题

1. 我们目前对于视频信息的利用还是比较少的，只用到了初始和最终的两张图来做简单的特征工程，对于我们这次简单的匹配任务还算应付地来，但显然对于其中信息仍可以做到更好地提取以实现更好的效果。
2. 没有充分检验整个算法的有效性。
3. KM 算法使用的相似度在部分数据间差别较小，还有进一步优化的空间。

3.2 优化的设计

- 对于视频信息的处理，我们最初的设计是使用 ConvLSTM 来进行 embedding 提取信息，对应的，对于音频的处理我们也打算使用 embedding 的方式来进行，但这样的方式比较有研究意义而对本次实验的帮助可能并不直接，并且调试起来比较耗时，于是我们没有采取这一方案，但我们认为这样的思路对于这一数据集的利用是有帮助的。

4 使用说明与文件清单

4.1 使用说明

1. 运行 `setup.py` 来安装需要的依赖包。
2. 确保 `test.py` 文件与 `src` 文件夹和 `weights` 文件夹处于同一级目录下。

   注：**测试需在 GPU 环境中运行**。另外由于不确定测试设备性能，所以各个 dataloader 都设置了较小的 batchsize，如有需要可自行调大该数值。

4.2 文件清单

文件名称	说明
test.py	测试所用主程序
setup.py	Python 3.7 下配置运行环境
requirements.txt	依赖库说明文件
train.py	训练所用主程序
src/audio_process.py	预处理 audio 数据
src/dataloader.py	定义 dataset
src/image_center.py	处理 video 数据
src/___init___.py	
src/KM.py	KM 匹配算法
src/model.py	定义各任务所使用的神经网络模型
weights	存放模型权重参数的文件夹

5 成员分工

模型的设计、报告的编写由三人共同合作完成，其余主要分工如下：

- ***：音频数据处理，模型的实现与训练
- ***：视频数据处理，模型的实现与训练
- ***：匹配算法的实现，dataset 和 dataloader 编写

参考文献

[1] Dhiraj Gandhi, Abhinav Gupta, and Lerrel Pinto. Swoosh! rattle! thump!–actions that sound. *arXiv preprint arXiv:2007.01851*, 2020.

[2] JJBOY. Cnn-repository-resnetv2. `https://github.com/JJBOY/CNN-repository/blob/master/model/ResNetv2.py`.

[3] xyxYang. Kuhn-munkres-algorithm. `https://github.com/xyxYang/Kuhn_Munkres_Algorithm`.