基于音频和图像序列的物体撞击匹配

马啸阳 (2018011054)，刘圣禹 (2017010302)，刘坤瓒 (2018011064)
清华大学电子工程系

小组分工—马啸阳：音视频特征提取、匹配算法框架；刘圣禹：模型训练框架、报告撰写；刘坤瓒：模型训练及参数优化、报告撰写。

I. 实验原理

A. 基于 *ResNet* 的分类

ResNet（Residual Network，深度残差网络）[1] 提出的主要动机是为了解决深度网络的退化问题。理论上，使用层数更多的神经网络可以使用比层数较少的网络获得更高的准确率，然而当网络层数增加到一定程度时，由于梯度消失等原因，继续增加网络层数反而会导致准确率的降低。事实上，深层网络可以将多余的层学习为恒等映射，出现这种问题说明用非线性层表示恒等映射是很困难的。

基于这种想法，记网络输入为 $\boldsymbol{x}$，需要学习的目标函数为 $\mathcal{H}(\boldsymbol{x})$，ResNet 网络将学习 $\mathcal{F}(\boldsymbol{x}) := \mathcal{H}(\boldsymbol{x}) - \boldsymbol{x}$。令 $\boldsymbol{g} = \mathcal{F}(\boldsymbol{x}, \boldsymbol{W}) + \boldsymbol{x}$ 表示网络输出，$\mathcal{F}(\boldsymbol{x}, \boldsymbol{W})$ 表示需要被学习的映射。此时网络只需将 $\mathcal{F}(\boldsymbol{x})$ 学习为 0 即可实现恒等映射，降低了学习的难度和资源消耗。由此分析得到 ResNet 的基础结构残差块，如图 1所示，其中 $\boldsymbol{x}$ 直连到输出。

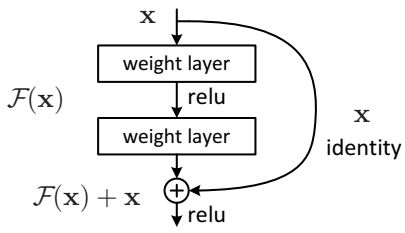


图 1. ResNet 的基础结构残差块

ResNet 与一般网络在 ImageNet[2] 上的对比结果如图 2所示。对照网络在网络层数为 34 层时的错误率要高于 18 层的网络，即说明网络随着层数增多可能取得更低的准确率；而 ResNet 在 34 层时的错误率要低于 18 层，说明 ResNet 可以解决这一退化现象，同时发现 ResNet 的收敛速度也要更快。总体而言，ResNet 通过引入残差块提高网络深度，获得更强的学习能力，近来有诸多网络变体，是实现分类网络的较好选择。实验中使用的 ResNet 网络结构如图 3所示。

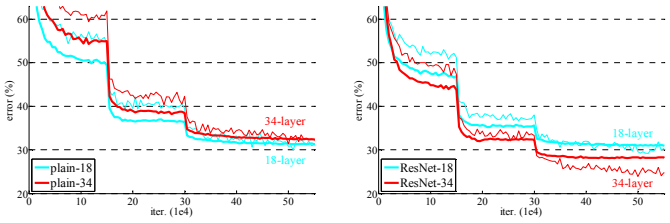


图 2. ImageNet 分类实验结果。左：一般网络；右：ResNet。

B. 对比损失函数

在音频和视频匹配问题中我们采用的损失函数是对比损失函数（Contrastive Loss）[3]，相比于以往的损失函数，对比损失函数学习成对样本的关系。

记 $\boldsymbol{W}$ 为网络参数，$\boldsymbol{x}_1, \boldsymbol{x}_2$ 为两个训练集中样本，对应的网络输出分别为 $\boldsymbol{g}_{\boldsymbol{W}}(\boldsymbol{x}_1), \boldsymbol{g}_{\boldsymbol{W}}(\boldsymbol{x}_2)$，$Y$ 表示一个标签，当 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 相似时记为 1，否则为 0。定义两个样本的距离为

$$D_{\boldsymbol{W}} = \left\| \boldsymbol{g}_{\boldsymbol{W}}(\boldsymbol{x}_1) - \boldsymbol{g}_{\boldsymbol{W}}(\boldsymbol{x}_2) \right\|_2,$$

则损失函数写为

$$\mathcal{L}(\boldsymbol{W}, Y, \boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{1}{2} Y D_{\boldsymbol{W}}^2 + \frac{1}{2}(1-Y)\max^2\{0, m - D_{\boldsymbol{W}}\}, \tag{1}$$

其中 m 表示一个阈值，仅当两个输出距离小于 m 时才会被计入损失函数，如图 4所示。

对比损失函数拉近了相似样本的距离，扩大了不相似样本的距离，同时引入阈值 m 减少了对距离较大的样本的计算量。

C. 其他使用的神经网络

1) 多层感知机: 多层感知机（MultiLayer Perceptron, MLP）是一种前馈神经网络，它由输入层、隐藏

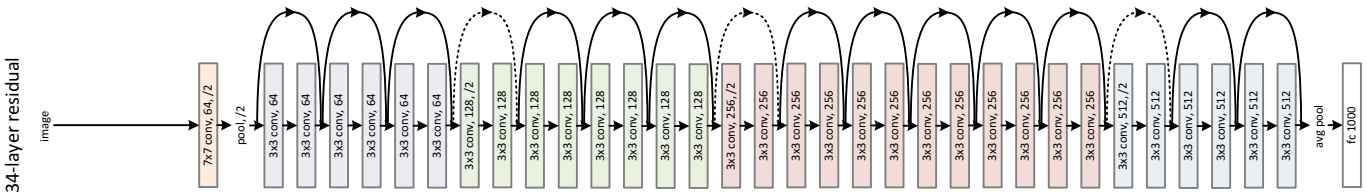


图 3. ResNet 网络结构

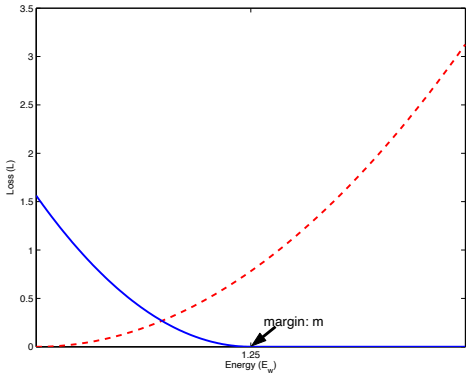


图 4. 对比损失函数与距离 D_W 的关系：红线表示对相似样本，蓝线表示对不相似样本，其中存在一个阈值 m。

层、输出层几部分组成，每个神经元中使用激活函数引入非线性，使用反向传播方法进行训练，可以用于解决线性不可分问题。

2) 3D 卷积神经网络： 3D 卷积神经网络（3-Dimensional Convolution Neural Network, 3DCNN）[4] 相比于传统卷积神经网络的特点在于使用三维卷积核进行卷积运算，如图 5所示。它的优点在于可以更好地提取帧间信息，所以会更好地学习多通道的音视频特征。

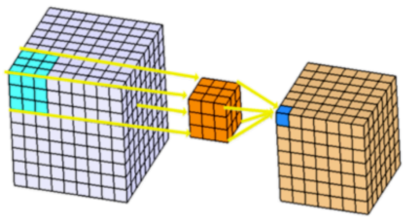


图 5. 3D 卷积神经网络使用三维卷积核进行卷积运算

II. 实现细节

A. 任务一：音频分类

1) 语谱图生成： 很多语音信号处理工作证实了用短时傅立叶变换（STFT）得到的语谱图表征音频是一项重要的预处理，尤其是对于使用神经网络处理的音频[5]。将本实验中得到的四通道音频转换为语谱图得到的结果如图 6所示，它的大小是 $513 \times 196 \times 4$，我们将这个张量作为 ResNet 的输入。

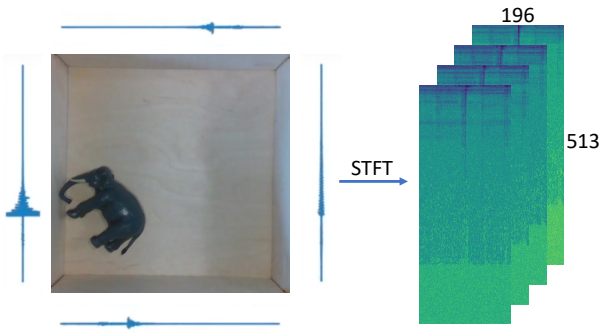


图 6. 将 STFT 变换得到的语谱图作为 ResNet 输入，图中表示以 toy_elephant 为例，将 4 个通道音频做 STFT 得到的 $513 \times 196 \times 4$ 张量。

2) 随机梯度下降优化器： 任务一优化器我们使用了随机梯度下降（Stochastic Gradient Descent, SGD）[6]。传统的梯度下降在更新每一参数时都使用所有的样本来进行更新，导致训练过程效率在样本较多的情况下很低。随机梯度下降为了解决这个问题在每次参数更新时只选用部分样本，实现过程中我们调用了 `torch.optim.SGD`，并考虑了权重衰减（Weight Decay）来缓解过拟合问题。

B. 任务二/三：完全/不完全匹配

1) 系统结构： 匹配问题的解决我们考虑了度量学习 [7] 的办法，系统结构如图 7所示。

训练集构建。 首先我们要构建出成对的音频和视频的训练集，图 7每中用蓝色框表示，每个数据是一个三元组 $(\boldsymbol{x}_1, \boldsymbol{x}_2, Y)$，其中 $\boldsymbol{x}_1$ 表示音频，$\boldsymbol{x}_2$ 表示视频，Y 表示标签，且音频 $\boldsymbol{x}_1$ 和视频 $\boldsymbol{x}_2$ 对应物体的类别相同，当二者属于同一次碰撞时记 Y 为 1。假设同一个类训练集中各有 γ 个音频和视频，则可构建出上述三元组 γ^2 个，其中正样本 γ 个，负样本 $\gamma^2 - \gamma$ 个。注意到负样本个数远多于正样本个数，存在一种不平衡性 [8]，

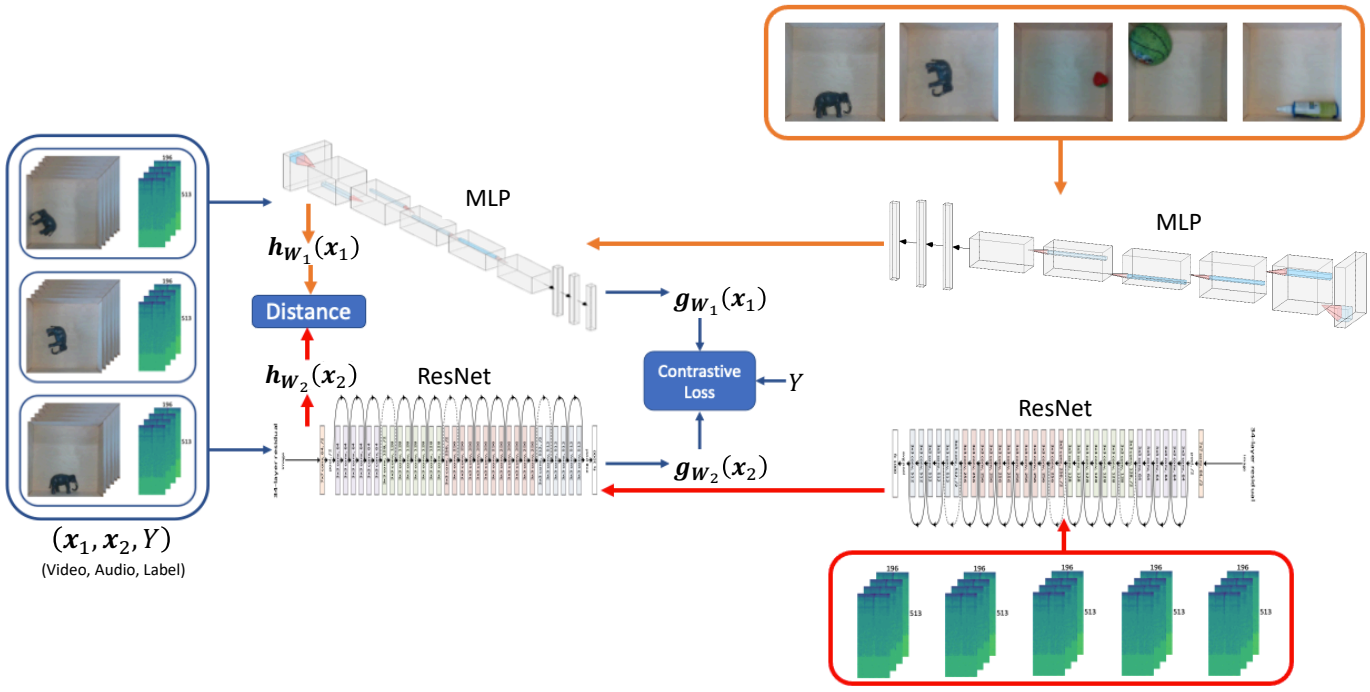


图 7. 完全/不完全匹配任务系统结构。蓝色框：音视频联合数据集；橙色框：视频测试集；红色框：音频测试集。蓝色箭头表示训练过程，橙色和红色箭头表示匹配的预测过程。

实际实现中我们构建了大小可调的数据集使得正负样本数量接近。同时又修改了损失函数(1)为

$$\begin{aligned}&\tilde{\mathcal{L}}(\boldsymbol{W}, Y, \boldsymbol{x}_1, \boldsymbol{x}_2)\\&=\lambda Y D_{\boldsymbol{W}}^2 + (1-\lambda)(1-Y)\max^2\{0, m - D_{\boldsymbol{W}}\},\end{aligned}$$

其中引入 $\lambda \in (0,1)$，用于解决正负样本的不平衡性。

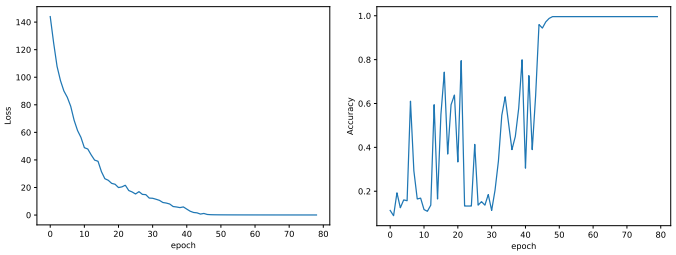
训练过程。训练过程在图 7 中用蓝色箭头表示。属于同一类视频 $\boldsymbol{x}_1$ 和音频 $\boldsymbol{x}_2$ 分别通过 MLP 和 ResNet 网络后得到输出 $\boldsymbol{g}_{\boldsymbol{W}_1}(\boldsymbol{x}_1)$ 和 $\boldsymbol{g}_{\boldsymbol{W}_2}(\boldsymbol{x}_2)$，结合标签 Y 带入对比损失函数进行反向传播 [6]，最终训练出的两个网络可以使得输出的视频和音频距离在二者相似时较小，而二者不相似时较大。同时还需要训练两个网络用于视频和音频的分类。

测试过程。训练过程在图 7 中用橙色和红色箭头表示，其中橙色方框表示视频数据集，红色方框表示音频数据集。二者先分别通过 ResNet 网络进行分类，对于同一类的视频 $\boldsymbol{x}_1$ 和音频 $\boldsymbol{x}_2$ 再分别通过之前训练好的模型获得输出 $\boldsymbol{h}_{\boldsymbol{W}_1}(\boldsymbol{x}_1)$ 和 $\boldsymbol{h}_{\boldsymbol{W}_2}(\boldsymbol{x}_2)$，进而获得一个距离矩阵，每一个类分别对应一个距离矩阵。距离矩阵是一个方阵，第 (i,j) 个元素表示此类中第 i 个视频和第 j 个音频的相似性。

2) 自适应矩估计优化器： 优化器我们使用了自适应矩估计（ADAptive Moment Estimation, Adam）[9]，它是一种只需要一阶梯度的高效随机优化方法。自适应矩估计根据损失函数对每个参数的梯度的一阶矩估计和二阶矩估计动态调整针对于每个参数的学习速率，它也是基于梯度下降的方法，通过引入动量来改善传统梯度下降，促进超参数动态调整，使得每次迭代参数的学习率都有一个确定的范围，面对大的梯度不会产生大的学习率，参数取值更加稳定。实现中调用了 `torch.optim.Adam`。

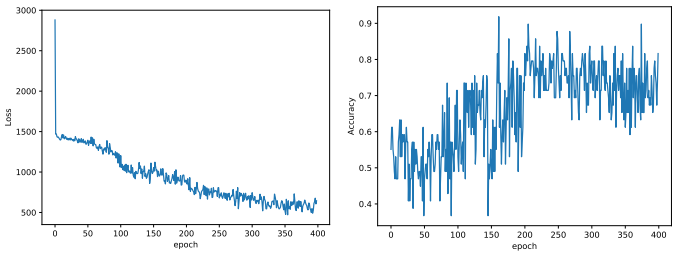
3) 匈牙利算法： 匈牙利算法（Hungarian algorithm）是一种在多项式时间内求解分配问题的组合优化算法。神经网络的输出结果是成对音频与视频的相似性，对应的距离矩阵是一个方阵，每个元素表示成对音频与视频相似性。匈牙利算法可以相应二部图中边权和最小的完全匹配，实现中我们调用了 `linear_sum_assignment` 来获得匹配的音频和视频。

在本问题中，我们先分别对视频和音频进行分类，再对同一类中的视频和音频使用匈牙利算法做匹配。对于提取视频特征，我们尝试了多种方法，如将图片序列拼接成三维矩阵，然后为了获取它的时序信息而采用 3DCNN 网络；或是针对 mask 图片序列，分别计算每张 mask 图片的质心，由此得到一个质心的运动轨迹向量，再将其送入 MLP 中提取特征。经过对比我们发现先提取质心特征的方法在多数类上的表现都要优于前



(a) Loss (b) Accuracy

图 8. 训练集上 Loss 和 Accuracy 变化情况



(a) Loss (b) Accuracy

图 9. 训练集上正负样本预测 Loss 和 Accuracy 变化情况

者，因此我们选用了提取质心 +MLP 的方法来进行训练。对于任务三，我们将距离与其他样本均过大的样本认为是没有匹配对象的样本，将匹配对象设为 -1。

III. 结果分析

A. 任务一：音频分类

训练集上 Loss 和 Accuracy 变化情况如图 8 (a) 和 (b) 所示。发现准确率开始时有较大的震荡，但全局上随损失函数的下降逐渐上升，最终稳定在了接近 100% 的准确率上。我们采用了交叉验证的方法证明了模型不是由于过拟合才有如此高的准确率。任务一分类的良好效果也很好地应用在了任务二和任务三上。

B. 任务二/三：完全/不完全匹配

训练集上正负样本预测 Loss 和 Accuracy 变化情况如图 9 (a) 和 (b) 所示。发现准确率震荡得同样很剧烈，最终随损失函数的下降能够稳定在 80%-90%。最后我们使用得到的网络产生距离输出，通过传统的匈牙利算法得到最终的匹配结果。

IV. 总结

本次实验中我们通过深度学习方法解决了音视频分类和音视频匹配问题。在匹配问题上我们先对音视频数据分类，对于同一类数据训练了一种度量学习模型，通过选择 ResNet 等合适的网络、对比损失函数和 Adam 等优化器获得了音视频的相似性，通过匈牙利算法计算了成对相似性之和最大的匹配，最终取得了较好的效果。

V. 文件清单

```
.
├── centroid.py ..... 质心函数
├── contrastive_loss.py ..... 对比损失函数类
├── dataset.py ..... 自定义数据集
├── main.py ..... 任务一主脚本
├── models.py ..... 各网络模型
├── requirements.txt ..... 项目依赖
├── resnet.py ..... ResNet 网络模型
├── spectrogram.py ..... 语谱图函数
├── task2.py ..... 任务二三训练脚本
├── test.py ..... 测试脚本
├── video.py ..... 视频分类训练脚本
└── *.pkl ..... 各模型文件
```

参考文献

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[3] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, vol. 2, 2006, pp. 1735–1742.

[4] S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[5] S. Clarke, T. Rhodes, C. G. Atkeson, and O. Kroemer, “Learning audio feedback for estimating amount and flow of granular material,” *Proceedings of Machine Learning Research*, vol. 87, 2018.

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[7] B. Kulis *et al.*, “Metric learning: A survey,” *Foundations and trends in machine learning*, vol. 5, no. 4, pp. 287–364, 2012.

[8] S. Kotsiantis, D. Kanellopoulos, P. Pintelas *et al.*, “Handling imbalanced datasets: A review,” *GESTS International Transactions on Computer Science and Engineering*, vol. 30, no. 1, pp. 25–36, 2006.

[9] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.