视听系统导论课程大作业报告

吴昆　无 58　2015010625

张蕴琪　无 51　2015011229

马洋　无 58　2015011181

组员	分工
吴昆	文献调研，提出本文所述模型，复杂度分析，撰写中期报告，撰写最终报告
张蕴琪	文献调研，尝试其它模型，调参数，准确率测试，撰写最终报告
马洋	尝试模型，调参数

一、问题分析与网络模型

网络模型图见最后一页。

题目给了我们视频和音频的语义矩阵（大小为每一帧语义向量维数×帧数），要求我们输出视频和音频的距离（值越小相似度越大）。

这个问题的核心是对提取得到的语义向量进行各种操作。我们知道，两个语义向量 vi,vj，要求相互关系，有三种常见的办法:dot(vi,vj),vi*M*vj 和 concat(vi,vj)。我们在这里，使用 concat，而位置则在输入一开始。输入伊始，我们就把视频和音频的语义向量 concat 起来。

注意到视频和音频的特征提取器使用的是 inception v3 和 vgg-like，都是或改进自对图片进行处理的网络，因此 120 帧之间的关系是独立的。我们在这里要通过一些办法，得到全视频的语义向量，经过 conv 滤波器组、或者 lstm 后沿着 time_step 方向做 max_pool 可以实现这一点。conv 是为了获得窗口中的整体特征，也不断得到更抽象的特征，而 Maxpool 则是为了得到表示这个窗口整体特征的语义向量，也算借鉴 CV 里的做法。

因此，我们采用将 conv 和 time_step 方向的 mp 的两层不断堆叠的网络，最后将输出展平，送到一个全连接层，得到最后的距离输出。

在网络输入处，音频和视频的 concat 和 conv 层之间，增加了一层 time_step 方向 kernel_size 不同的 conv 层，这是学习 NLP 中 encoding 层由 word embedding 得到语句层面特征的做法，即可以综合考虑不同窗口大小的邻居帧的信息，这在 NLP 中符合现实也在视频中也符合现实。

使用 Binary Cross Entropy 作为目标函数。

二、实现与骨架代码修正

我们改用了 pytorch 原生的 Adam 优化器，参数如 learning_rate 使用默认的，一般来说如果网络鲁棒，这些参数对结果影响不大。将 weight_decay 设置成 1e-4，等效于 L2 正则项，于是不需要另写，而 1e-4 是 L2 正则项常见的值。

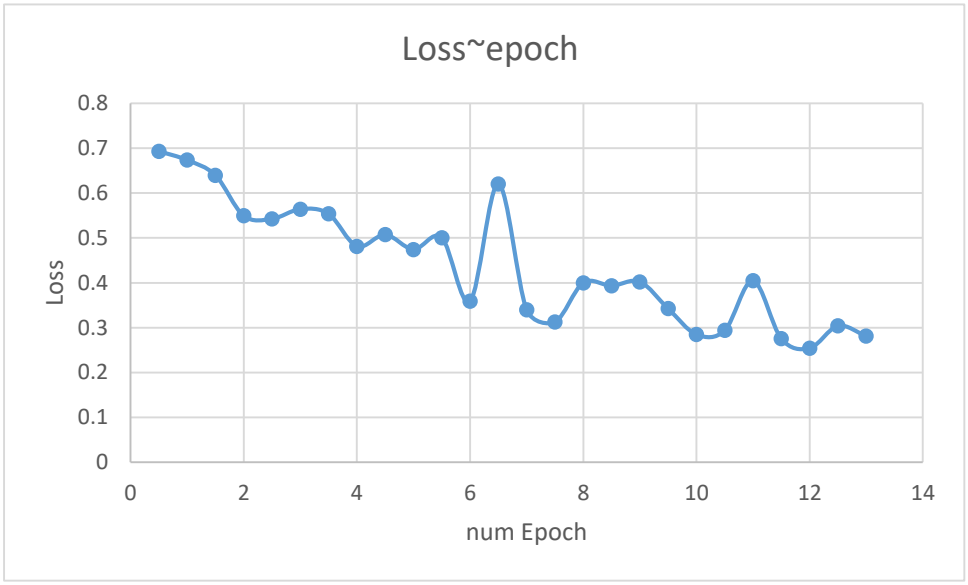
正例和负例的取法，按照所给骨架文件的默认，按 1:1 取，这个比例非常科学。

Pytorch 的 BCE loss 有 bug，在测试 size 一致的地方，自己按源码实现了自己的 MyCrossEntropyLoss 类。

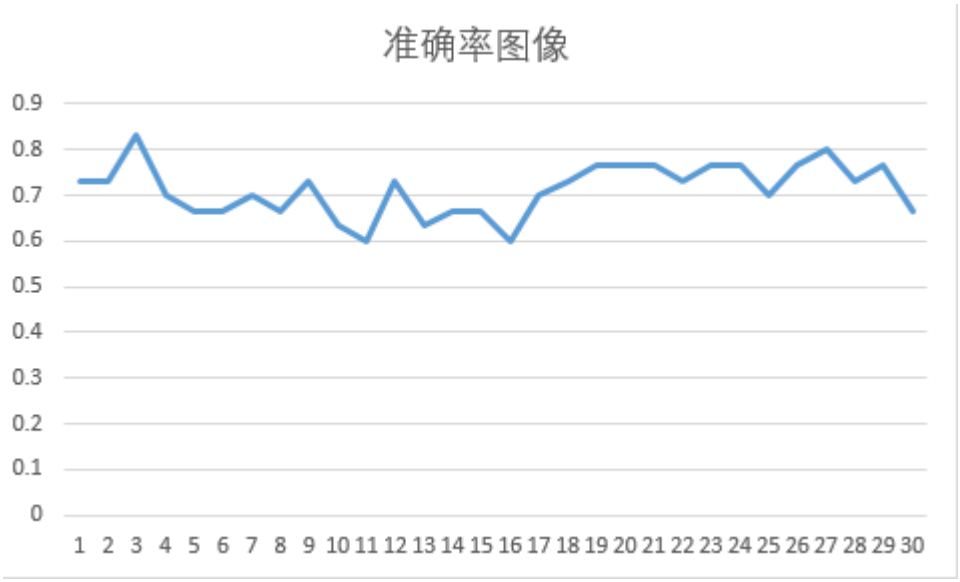
三、参数选择和准确率

调参从两个方面入手，第一个是加深，在后部增加卷积层或全连接层，但是经过尝试之后发现对准确率没有提高；第二个是第一层卷积层的卷积核大小，原来为 64x{5,7,9,11},改为 64x{13,13,15,15}。经过测试后发现，将滤波器核增大能够提高准确率。修改后的准确率图像如下图所示：

横坐标为 x10epoch，从 10~300 个 epoch 的准确率如图所示，可见相当稳定，没有发生过拟合的现象。



可以看到训练过程中 loss 稳步下降。



Adam 优化器，调 learning rate 和的意义不大，所以就没有调。

为了方便调试，我们在测试时修改 evaluate.py 的代码，使它能够一次性测试任意个 breakpoint 的准确率，但提交的代码没有体现这一修改。

可以看出模型基本上在 4、5 个 epoch 就达到了足够的准确度。随着迭代次数的增加，准确率现在 eopch30 处达到峰值 0.833，之后由于过拟合等问题出现了准确率的下滑和波动。

四、参数量

由于使用主要为 conv 层，最后的全连接层的输入大小仅为 64*15，所以网络的参数量不算大。原型网络的实测 checkpoint 文件大小为 11248843B(10.5M 左右)。滤波器数量增大一倍后，变为 18.5M.

五、时间和空间复杂度

考察 inference，一个样本输入到神经网络后的计算量。网络主要由 CNN 构成。对于一个(F_n,T_s)的样本，CNN 核为(F_n,k,n)（F_n特征向量数，T_s帧数，k 卷积核大小，n 卷积核数量），计算复杂度为 O(F_nT_sn)，由于之后核逐渐减小，因此总的复杂度不会超过 O(F_nT_sn)。由于之后的 CNN 层在 Ts 和 n 方向均比之前缩小 4 倍以上，因此计算元素乘累加作为一次操作，计算总数在第一层的两倍以内，当然考虑到 SIMD，这样的估算并不科学。全连接层的 64*15 的 64 和 15 分别对应的是 T_s和 F_n逐渐缩小，计算复杂度为 O(T_sF_n)。

因此 inference 总的时间复杂度为 O(F_nT_sn)，考虑到全连接和 CNN 都是并行度很高的算法，而且每一层的运算量很容易切分成工作量均等的多块，因此计算可以很好地在 GPU 上加速。

Training 部分，由于全连接层和 CNN 的梯度回传算法均是矩阵转置做乘法，考虑到这里全连接层只有一行/列，以及 CNN 核是稀疏阵，因此计算的时间复杂度依然是 O(F_nT_sn)。而且并行性很好。

类似以上的分析，可以得到空间复杂度也是 O(F_nT_sn)。

六、优点和不足处分析

我们的模型没有使用 LSTM，所以速度很快。同时，300epoch 的准确率告诉我们，网络没有出现过拟合的现象，这是很不错的。

不足之处主要有两个，首先模型主要使用 CNN，但是目前的结构过于简单，近似于 VGG 等模型的前几层，因而准确率一般。在尝试中没有发现准确度更好的模型，这是我们经验不足所致。第二是没有自己建立数据集，来进一步提高并验证模型的准确度。

七、参考文献

复现了[6]，尝试在[1][2]的启发下写加上 attention 的 LSTM，但效果均不好。

[1] Oriol Vinyals, Alexander Toshev, Samy Bengio, et al. Show and Tell: A Neural Image Caption Generator. Arxiv, 2014.

[2] Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. Arxiv, 2016.

[3] Louis-Philippe Morency, Tadas Baltrusaitis. Tutorial on Multimodal Machine Learning. ACL, 2017.

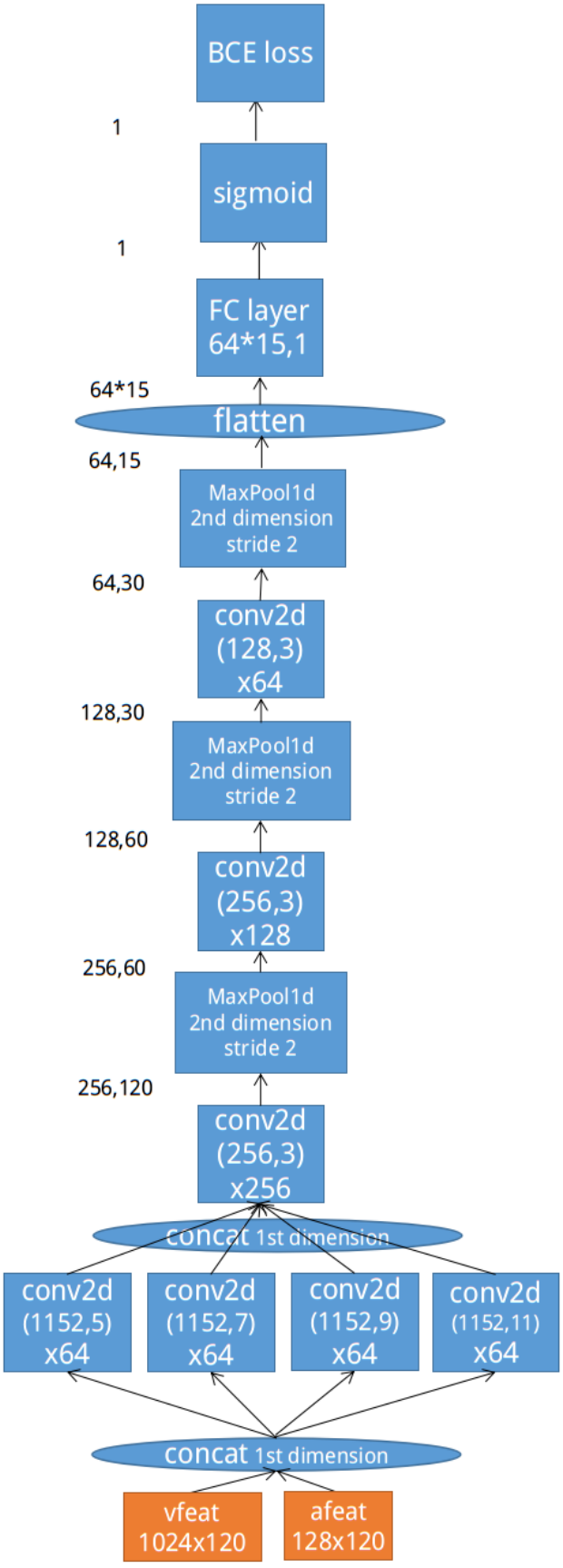
[4] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, et al. Multimodal Deep Learning. NIPS, 2010.

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013.

[6] Cunchao Tu, Han Liu, Zhiyuan Liu, et al. CANE: Context-Aware Network Embedding for

Relation Modeling. ACL, 2017.

附页：模型图



左侧数为该位置的 tensor 形状
第一维为语义向量长度，第二维方向为 time_step 方向
只研究了一个输入样本的情形，mini-batch 不本质