多模态情感分析

github地址：https://github.com/LL1122LL/multimodel-sentiment-analysis

实验目的

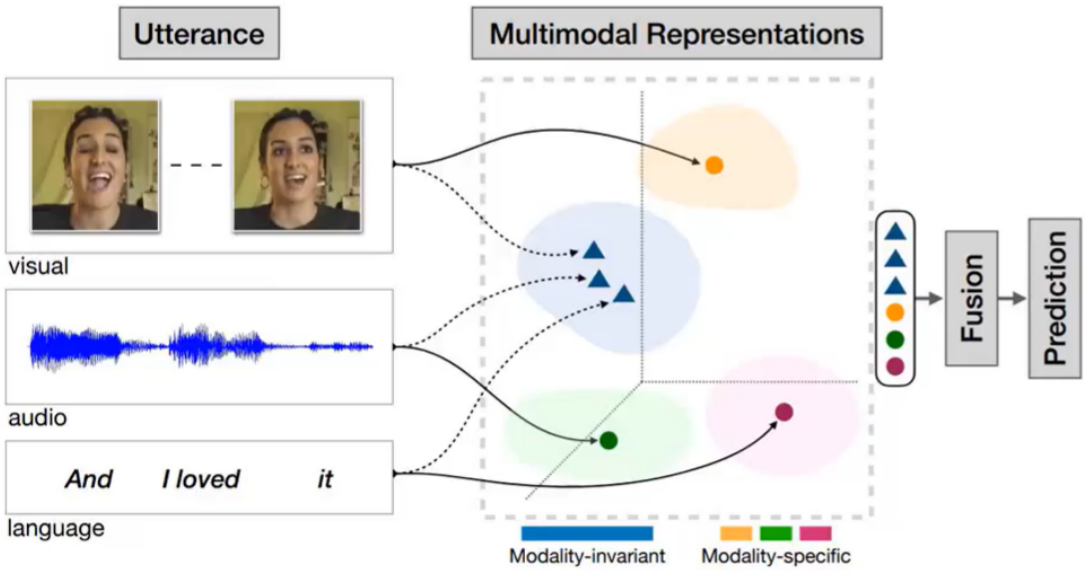
给定配对的文本和图像，预测对应的情感标签（三分类任务：positive, neutral, negative）。

实验任务

- 设计一个多模态融合模型。
- 自行从训练集中划分验证集，调整超参数。
- 预测测试集（test_without_label.txt）上的情感标签。

任务概括

多模态情感分析主要集中在单模态表征学习和多模态融合两个方面，如下图所示，蓝色代表不同模态内包含的相同的语义信息，其他颜色代表其单模态学习到的特定的语义信息。



多模态的特点：空间差异性难以代表特定模式的差异，直观的，不同模态具有不同语义信息，文本是人类产生的信号，具有高度和密集的语义信息，相反，图像是具有大量空间的冗余信号，包含了低阶的语义和单元特征

因此，我们在本实验中，我们需要做的，是在imagae_model部分提炼出少的冗余的特征向量，且其特征向量内共同语义和独特语义的部分要尽可能的分离。

为了验证这个思想

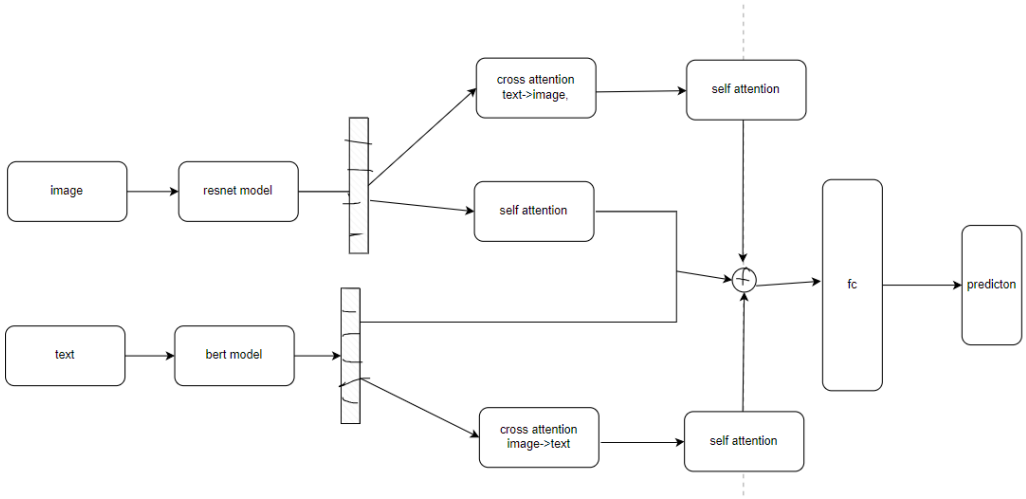
我这里给出只通过图像或者文本的实验结果

	accuracy
text	0.71
image	0.65

发现图像的高阶语义特征确实不如text明显。

模型设计介绍

cross-attention ，即相比self-attention的(q,k,v)都是同一个向量，其q是一种类型的特征向量，k、v是另一种特征向量。



得到表征向量

通过resent模型得到图像表征，通过bert模型得到文本表征。

对于img的表征向量进行self-attention，得到有用的单模态表征。通过cross-attention + self-attention操作，得到其对于情感分析有用的共同的语义部分(cross-attention)，并且得出共同语义部分中对情感预测比较有用的部分(self-attention)

对于文本的表征向量，一条路线是不做处理，另一条则也是使用通过cross-attention + self-attention操作。

向量降维处理

现在得到的表征向量shape都是(seq_len,batch_size,hidden_dim),在相加前需要将shape转为(batch_size,hidden_dim)。

在文本模块得到的表征向量，我们取seq_len的最后一个值作为全局特征，因为文本具有较好的时序性，最后一个token可以一定程度代表整个句子的语义信息。

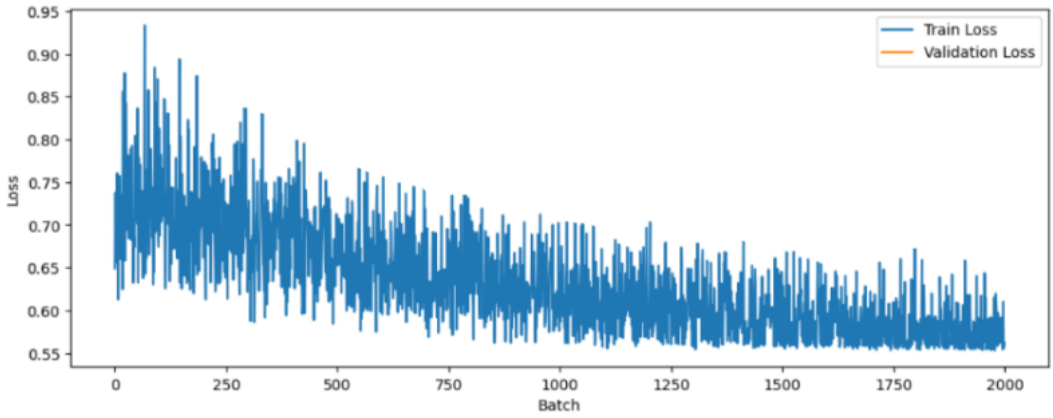
在图像模块方面，由于这是静态图，主要包含的空间信息，且不同图片因为构图的不同，其空间位置的相对关系也会不一样，因此这里选择在dim=0处取平均值。

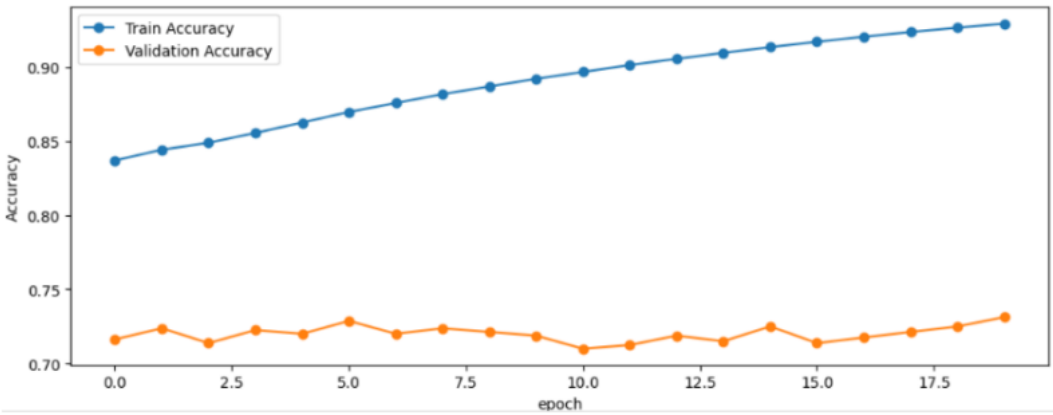
预测

将四组降维后的表征相加，得到多模态特征向量，然后进入全连接层预测。

实验结果

训练与验证





最后选择在第5个epoch所训练出的参数，其acc为0.73.

刚开始的剧烈振荡是因为预训练模型的下游进行调试和数据的label数量不平衡(nuetral标签的数据量要大大少于其他两者，positive标签的数据量要明显高于两者)所造成的。

最开始的metric_report

	precision	recall	f1-score	support
0	0.75	0.82	0.79	483
1	0.00	0.00	0.00	83
2	0.59	0.69	0.64	234

拥有较高准确率时的metric_report

	precision	recall	f1-score	support
0	0.80	0.74	0.77	483
1	0.61	0.20	0.31	83
2	0.54	0.76	0.63	234

0代表positive标签，1代表neutral标签，2代表negative标签。由于nuetral标签数据量极少，因此只能通过加大epoch的方法来学习neutral对应的高阶语义信息。其所带来的问题则是，会引起negative标签对应的准确率的下降。因此会出现最后依旧有小幅度振荡且loss值不低的现象。

消融实验

feature	accuracy
TEXT Only	0.7
Image Only	0.61
Both	0.73

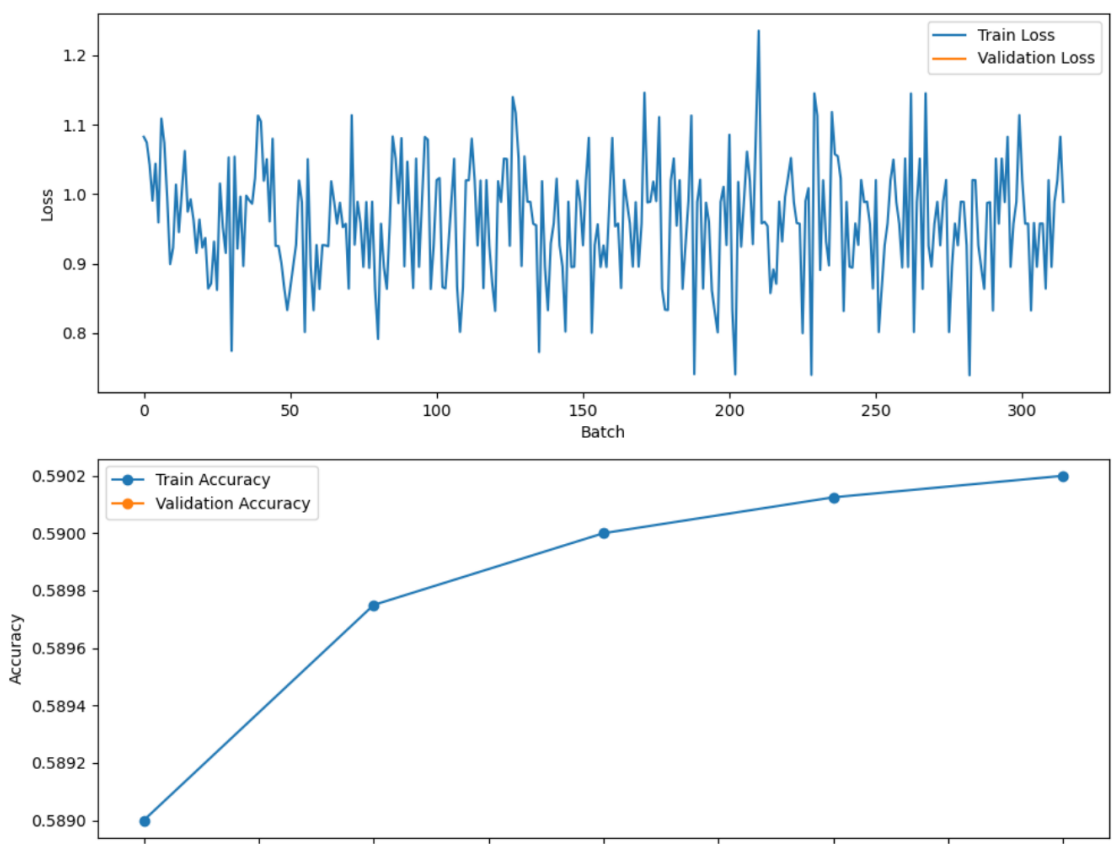
发现多模态确实要要优于两者。

奇怪的是，在这里我们可以发现消融实验下的Image Only的准确率是要低于先前给出的仅凭image预测的准确率的。

这里是因为在self-attention中，我使用的归一化方式是nn.LayerNorm,而用于图像的分类方法，其主要是batchNormalize的归一化方式。

实验中遇到的问题

遇到了准确率一直为0.59，且loss曲线振荡极大的情况



经过打印，发现在训练过程和验证过程中，在第一个epoch的前几个batch内，其预测label有positive，neutral，negative。但在剩余的过程中，其预测输出的label几乎都是positive。

然后我们再查看所给的训练集

	positive情感	neutral情感	negative情感

	positive情感	neutral情感	negative情感
train loader	1905	336	959
test loader	483	83	234

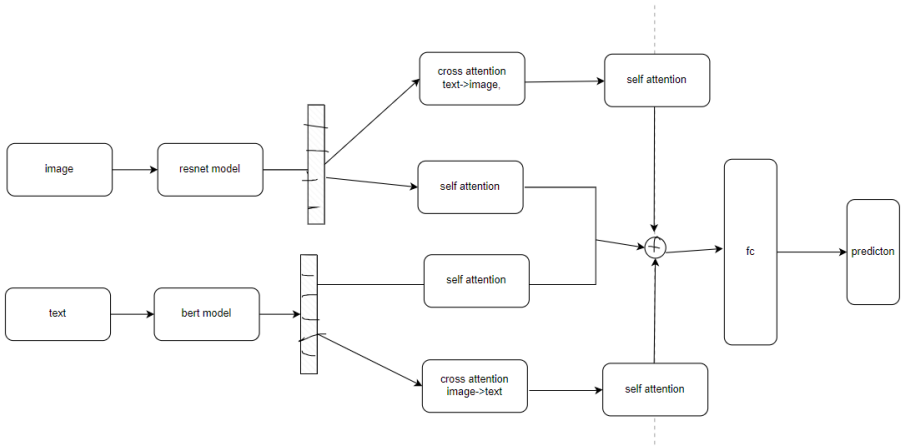
发现在test loader 中，positive样本的百分比为60%，与预测的accuracy相近。并且在以batch为横轴的loss曲线中，其振荡幅度极大，因此推测是由于每个内，因为positive情感样本个数较多，导致每一个batch内的训练后，过于注重positive情感，无法很好的辨别其他情感，使得模型很不稳定。

通过两个措施解决了这个问题：

- 降低了模型的学习率参数；
- 降低了模型的复杂度

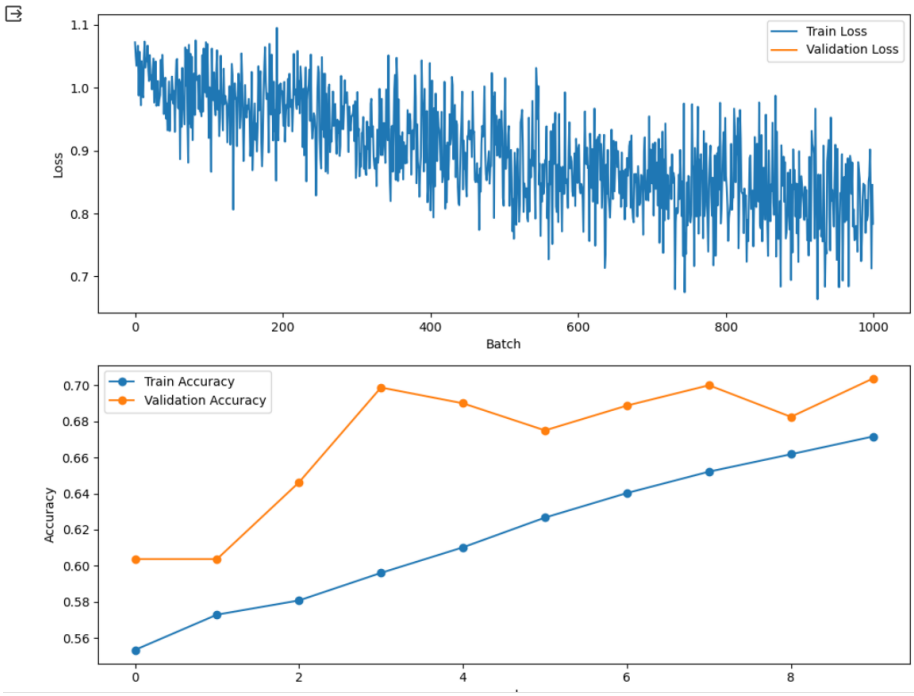
在降低了学习率后，(从1e-4到5e-6)，其准确率有明显提升(从0.59到0.67),但还是较低.**同时loss的振荡幅度依旧很大，且收敛值为1。**

接下来准备降低模型复杂度，下图是原先的模型，且这里在对img的向量进行reshape时，即从(seq_len,batch_size,hidden_dim)到(batch_size,hidden_dim)，我是直接选取src_len的最后一个token的特征代表整个句子的高阶语义信息的。



由于最高的准确率0.67是要比单独使用text数据预测还要低的，所以我这里最初猜测是text和image的对齐方式不够好，才导致拉低了text的预测能力。

在给它img图像添加绝对位置编码或者可学习的位置编码后，其准确率有一定的提高，但振荡幅度依旧很大，两种编码方式都如下图所示(这里展示的时是可学习的位置编码)。



上述图像看出，这种方式很容易形成过拟合。

因此在这里认为位置编码无法帮助小型数据集的图像数据与text文本对齐，然后我在尝试直接对图像的相关特征向量取平均值后，其振荡幅度明显减小，准确率上升(即实验结果部分的图)。

不足之处：

由于colab的限制，我无法将batchsize设置的很大，这就加大了loss曲线的振荡调试。

由于我的一个模型每10个epoch需要运行20分钟，时间上的消耗导致无法对模型参数进行更加细致的微调。