

当代人工智能 lab5 实验报告

杨弘

July 14, 2022

1 github 地址

<https://github.com/ya-hong/Lab5>

2 模型设计

2.1 文本特征提取

使用预训练的 Bert 进行文本特征提取 (distilbert-base-cased)。使用输出结果中的 `last_hidden_state` 作为每个单词的特征向量。每个单词会得到 768 维的特征向量。

2.2 图像特征提取

使用 ResNet 提取图像特征。将此特征向量作为图片的 embedding 嵌入向量。

2.3 模型融合

使用 Bert 获取文本中每个单词的 embedding 嵌入表示。根据图片的 embedding 向量和单词 embedding 向量的相似度计算注意力。通过注意力对所有单词的特征向量进行加权求和得到图文特征向量。最后将图文特征向量通过全连接层得到三种分类的得分。

2.4 设计原因

主要参考了论文《TSAIE：图像增强文本的多模态情感分析模型》。并根据实际情况做了一些改动。

1. **调整图像特征向量的维度以匹配单词特征向量的维度** 由于单词特征向量是通过 Bert 得到的，所以有更好的表达能力。如果在 Bert 后面的输出再添加全连接层调整输出维度可能会使模型过拟合。而 ResNet 是自己实现并从零开始训练的，更容易调整输出维度。
2. **使用 ResNet 而非 VGGNet** VGGNet 参数过多训练过慢，而且容易过拟合。使用带 BottleNeck 的 ResNet 可以用很少的参数训练出 768 维的特征向量。
3. **最终的图文特征向量没有拼接图片特征向量** 论文的模型图片特征不仅用于增强文本特征，同时也参与了最终的分类得分计算。而我的模型中图片模型是没有预训练参数的，所以不把图片特征直接带入到最终的分类得分计算。

3 遇到的 bug

无法从 huggingface.co 下载模型的问题 代理软件与 python 有冲突。关闭代理软件或者直接使用 colab 的环境运行代码。

数据量超出显存大小 筛选长度较小的数据 只选择代码长度小于 400 的数据，减小一次训练的数据量。

减小 batch size 减小 batch size 也能减小一次训练的数据量。

梯度累积 单次训练不进行反向传播，累加记录的梯度值。达到一定训练次数后再进行反向传播。

训练部分参数 可以再 `torch.no_grad()` 下运行 bert, 只训练 bert 以外的部分。

loss 不下降，或者结果全为相同字符串 loss 不下降是学习率过低，结果为相同字符串是学习率过高。使用普通优化器很难精确找到合适的学习率，需要将优化器改为 AdamW。

验证集上 loss 正常，但预测字符串完全错误 在使用 mask 获取 Tensor 的有效部分时，没有将 mask 转化为 bool 类型导致输入错误。

4 验证结果

4.1 融合模型

训练

```
epoch: 1, loss: 1.0568655401468277, acc: 0.5299999713897705
epoch: 2, loss: 1.0259807094931603, acc: 0.5849999785423279
epoch: 3, loss: 1.016262798011303, acc: 0.5849999785423279
epoch: 4, loss: 0.9995096534490585, acc: 0.5849999785423279
epoch: 5, loss: 0.9832276731729508, acc: 0.5849999785423279
epoch: 6, loss: 0.9691289111971855, acc: 0.5899999737739563
epoch: 7, loss: 0.9554122760891914, acc: 0.6100000143051147
epoch: 8, loss: 0.9458006158471107, acc: 0.6150000095367432
epoch: 9, loss: 0.9360772550106049, acc: 0.6274999976158142
epoch: 10, loss: 0.9357915386557579, acc: 0.6225000023841858
epoch: 11, loss: 0.9174141719937324, acc: 0.6399999856948853
epoch: 12, loss: 0.9122666418552399, acc: 0.6349999904632568
epoch: 13, loss: 0.8987748622894287, acc: 0.6349999904632568
epoch: 14, loss: 0.8799556404352188, acc: 0.6524999737739563
epoch: 15, loss: 0.8708220794796944, acc: 0.6599999666213989
epoch: 16, loss: 0.8624436303973197, acc: 0.6574999690055847
epoch: 17, loss: 0.8574628531932831, acc: 0.6850000023841858
epoch: 18, loss: 0.8442770361900329, acc: 0.6875
epoch: 19, loss: 0.8405456334352494, acc: 0.6825000047683716
epoch: 20, loss: 0.8269083201885223, acc: 0.6949999928474426
epoch: 21, loss: 0.8153357356786728, acc: 0.6924999952316284
epoch: 22, loss: 0.8061919867992401, acc: 0.7099999785423279
epoch: 23, loss: 0.7960300207138061, acc: 0.7099999785423279
epoch: 24, loss: 0.7972919151186943, acc: 0.7074999809265137
epoch: 25, loss: 0.7854761019349098, acc: 0.7099999785423279
epoch: 26, loss: 0.7832262217998505, acc: 0.73499995470047
epoch: 27, loss: 0.7949161112308503, acc: 0.6974999904632568
epoch: 28, loss: 0.7869431897997856, acc: 0.7099999785423279
epoch: 29, loss: 0.7751833081245423, acc: 0.7049999833106995
epoch: 30, loss: 0.7719051361083984, acc: 0.7124999761581421
```

验证

loss: 0.8042360973358154, acc: 0.6734482717514038

4.2 文本模型

*****TRAIN*****

```
epoch: 1, loss: 0.992941380938126, acc: 0.5901400446891785
epoch: 2, loss: 0.9059958188786946, acc: 0.6487748026847839
epoch: 3, loss: 0.8668545997128881, acc: 0.6694865822792053
epoch: 4, loss: 0.8491705143465601, acc: 0.6694865822792053
epoch: 5, loss: 0.8360353344518913, acc: 0.6718202829360962
epoch: 6, loss: 0.830008877130484, acc: 0.6761960387229919
epoch: 7, loss: 0.8195344461443266, acc: 0.6788215041160583
epoch: 8, loss: 0.8068623985125832, acc: 0.6893232464790344
epoch: 9, loss: 0.8019146563152727, acc: 0.691948652267456
epoch: 10, loss: 0.7925625315485925, acc: 0.6910735368728638
epoch: 11, loss: 0.7891235165863082, acc: 0.6904900670051575
epoch: 12, loss: 0.7876254116362146, acc: 0.695157527923584
epoch: 13, loss: 0.7781510209119167, acc: 0.6971995234489441
epoch: 14, loss: 0.7752608009369537, acc: 0.695157527923584
epoch: 15, loss: 0.7795333892906838, acc: 0.6992415189743042
epoch: 16, loss: 0.7682457977303825, acc: 0.6980746984481812
epoch: 17, loss: 0.7569320745061966, acc: 0.7036172747612
epoch: 18, loss: 0.7638099008221332, acc: 0.695157527923584
epoch: 19, loss: 0.7551713912740193, acc: 0.7071178555488586
epoch: 20, loss: 0.7588324014406082, acc: 0.7030338644981384
epoch: 21, loss: 0.7483814258906499, acc: 0.7050758600234985
epoch: 22, loss: 0.7540308921103438, acc: 0.7050758600234985
epoch: 23, loss: 0.7500224034176963, acc: 0.7062427401542664
epoch: 24, loss: 0.7429185093770645, acc: 0.704492449760437
epoch: 25, loss: 0.7407245067412266, acc: 0.7047841548919678
epoch: 26, loss: 0.7487613182849617, acc: 0.6992415189743042
epoch: 27, loss: 0.741700294067868, acc: 0.7018669843673706
epoch: 28, loss: 0.734362312521929, acc: 0.7129521369934082
epoch: 29, loss: 0.7332270052566551, acc: 0.7074095606803894
epoch: 30, loss: 0.7269337578582096, acc: 0.7214118838310242
```

*****VALIDATE*****

4.3 图像模型

由于只用图像进行文本特征增强，所以不会有好效果

*****TRAIN*****

```
epoch: 1, loss: 1.0627044871918936, acc: 0.6021003723144531
epoch: 2, loss: 1.0382066599963844, acc: 0.6021003723144531
epoch: 3, loss: 1.0357490339802193, acc: 0.6021003723144531
epoch: 4, loss: 1.0356442088146054, acc: 0.6021003723144531
epoch: 5, loss: 1.0356449017446823, acc: 0.6021003723144531
epoch: 6, loss: 1.0356506731713249, acc: 0.6021003723144531
epoch: 7, loss: 1.0356553753309796, acc: 0.6021003723144531
epoch: 8, loss: 1.0356588465190963, acc: 0.6021003723144531
epoch: 9, loss: 1.035661675270686, acc: 0.6021003723144531
epoch: 10, loss: 1.035664052779644, acc: 0.6021003723144531
epoch: 11, loss: 1.0356660909524738, acc: 0.6021003723144531
```