

表情识别中期报告

2020010768 无 05 付宇辉

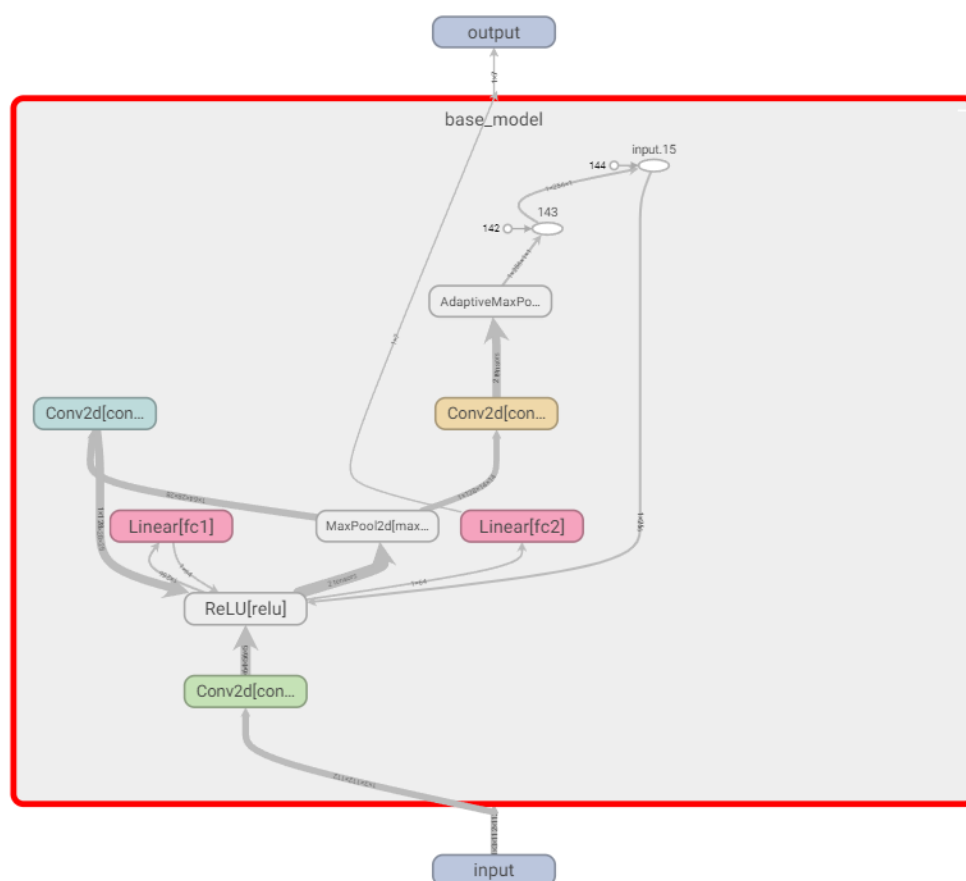
注：以下实验环境均为在 torch1.9.0, cuda=11.1

一、运行基准模型

1. 代码结构

└─files	
	dataset.py 自定义数据集，用于加载数据
	main.py 训练，验证，测试模型
	model.py 定义模型结构

2. 基准模型的结构



3. 运行结果

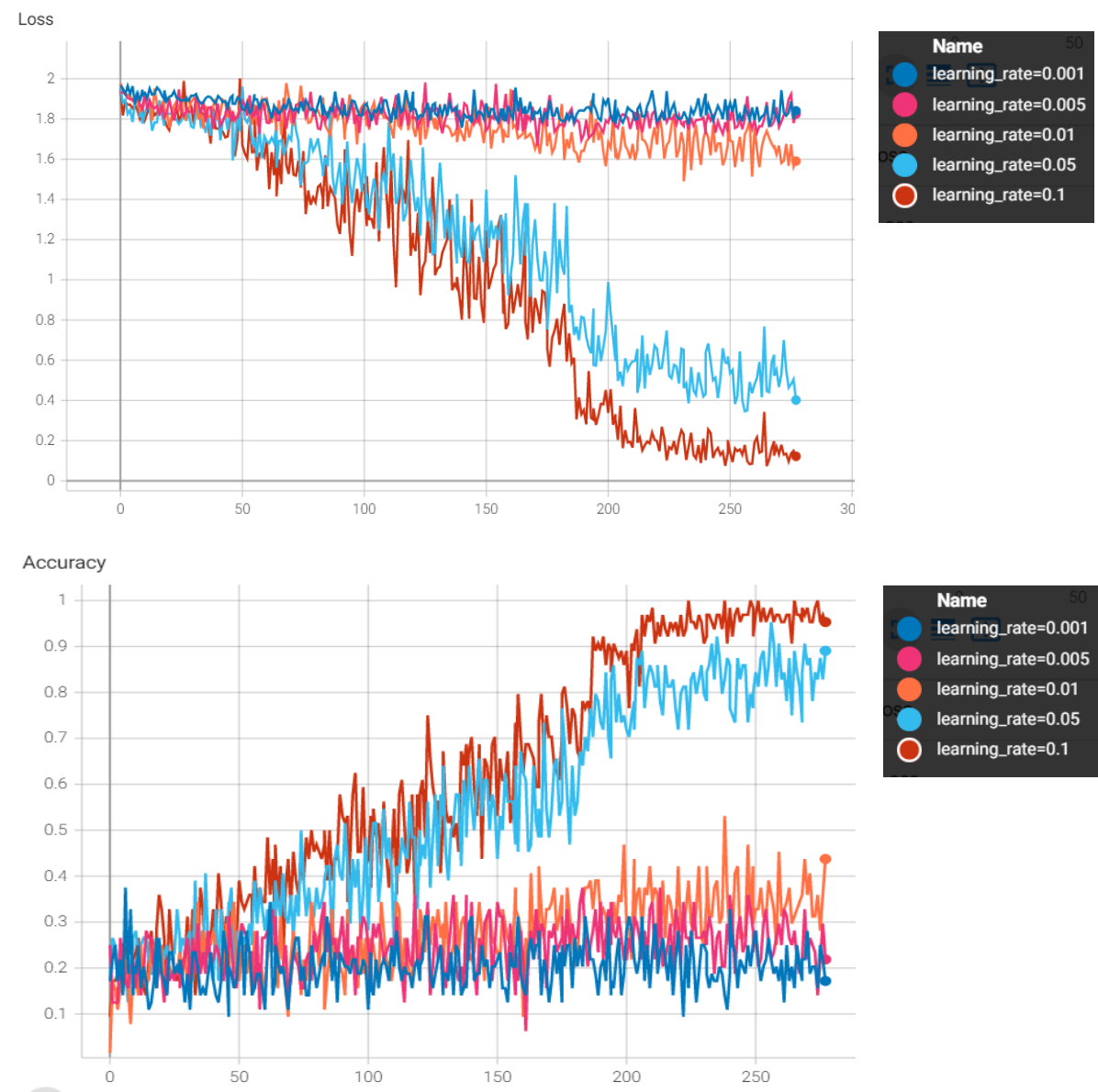
在全部使用默认参数的情况下的输出结果为：

val accuracy:28.952381134033203%

test accuracy:32.21757125854492%

二、实验结果分析

1. 对于不同的 learning_rate，在训练集上其 loss 和 accuracy 的变化曲线如下：



只改变 learning_rate，其余参数全为默认的情况下的输出结果为：

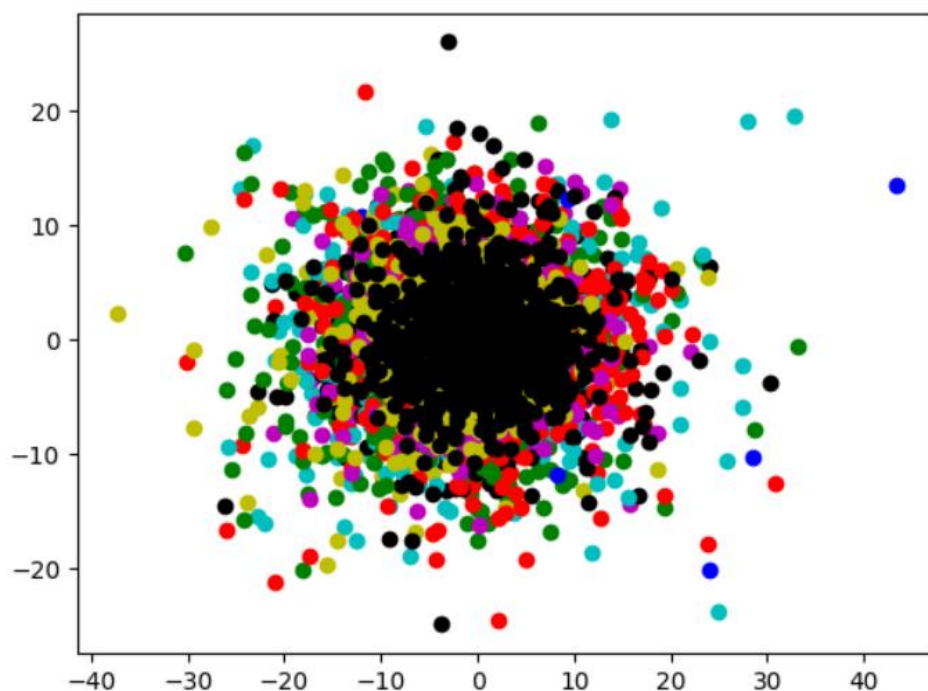
learning_rate	val accuracy	test accuracy
0.001	17.06122398376465%	17.06122398376465%
0.005	22.91156578063965%	26.79218864440918%
0.01	28.952381134033203%	32.21757125854492%
0.05	49.59183883666992%	52.23152160644531%
0.1	49.755104064941406%	53.528587341308594%

2. 样本的特征分布

加载在 `learning_rate = 0.1` 下训练得到的模型, 其余超参数均为默认的情况下, 在对测试集的预测过程中, 选取了如图所在的特征, 进行了 PCA 降维分析, 降维到 2 维并进行了可视化分析。

```
def forward(self, x):  
    x = self.relu(self.conv1(x))  
    x = self.max_pooling(x)  
    x = self.relu(self.conv2(x))  
    x = self.max_pooling(x)  
    x = self.conv3(x)  
    x = self.GAP(x).squeeze(dim=3).squeeze(dim=2)  
    # you can see this x as the feature, and use it to visualize something  
    self.feature = x  
    x = self.fc1(self.relu(x))  
    x = self.fc2(self.relu(x))  
    return x
```

看图, 发现模型并没有将特征很好地进行分离, 仍然混在一起。经过思考之后发现, 在特征提取之后仍然有两层非线性激活函数, 导致此时的特征并没有近似的线性可分的性质, 之后的实验会改变模型, 再次分析。



三、模型优化

接下来考虑使用的办法：

训练数据的使用：数据增强

网络结构的调整：使用 Resnet 的结构，进行实验

损失函数：交叉熵，加入正则化损失等等

训练方式：调整学习率，batch size，优化器换用 adam 等等，加大训练的 epoch 使其完全收敛等。