

深度学习大作业： Tiny-Caltech35 分类

实验环境

Ubuntu 20.04 (Windows WSL2)

pythoch 1.8.0 + torchvision 0.9.0 + cuda11.1

基准模型运行

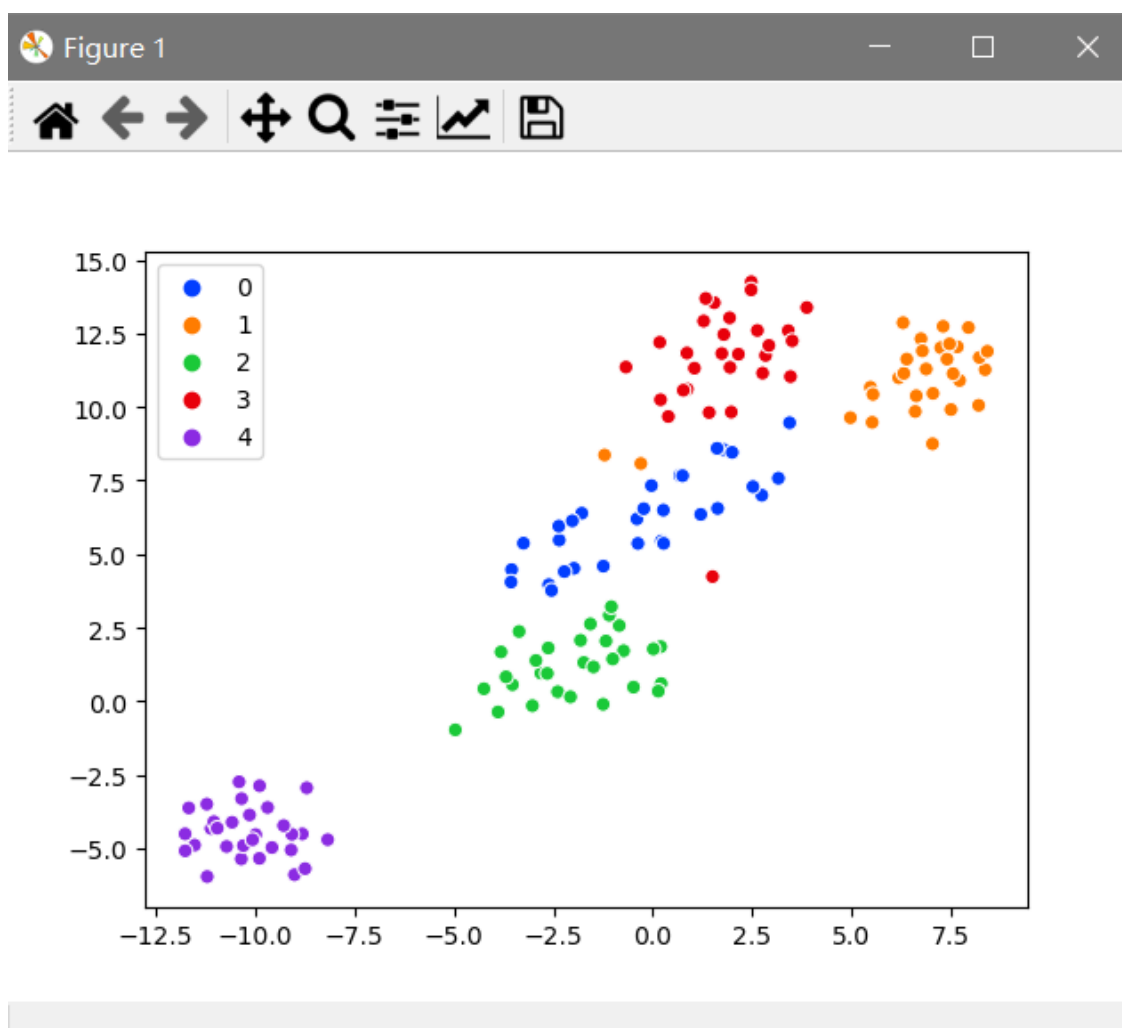
1. dataset.py 定义了读取数据集的方法
2. model.py 定义了网络结构与前向和反向传播算法
3. main.py 调用了前两个文件中定义的类，并定义了训练和测试的函数，但是默认代码运行在cpu上，因此需要自行修改至GPU
4. 基准模型准确率如下所示，有明显的过拟合

5.

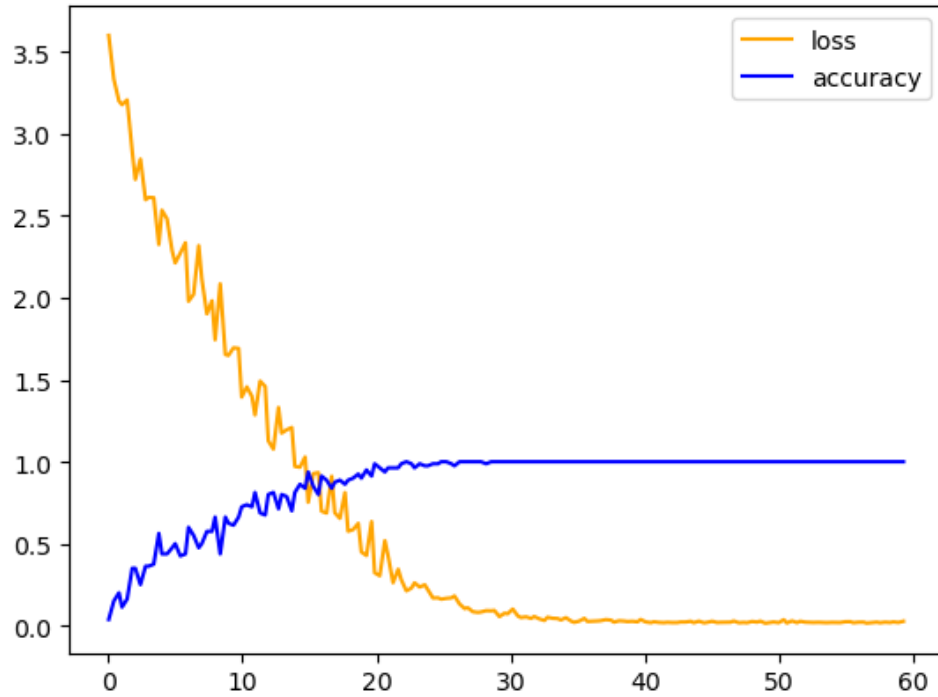
train	val	test
100%	51.4%	45.6%

可视化

1. T-SNE可视化样本分布：对经过网络后的35维输出进行TSNE降维以及可视化

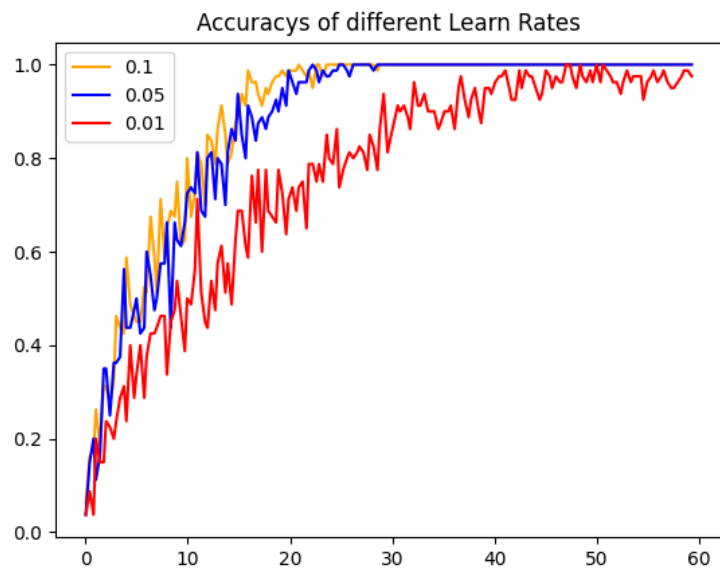


2. loss与准确率变化

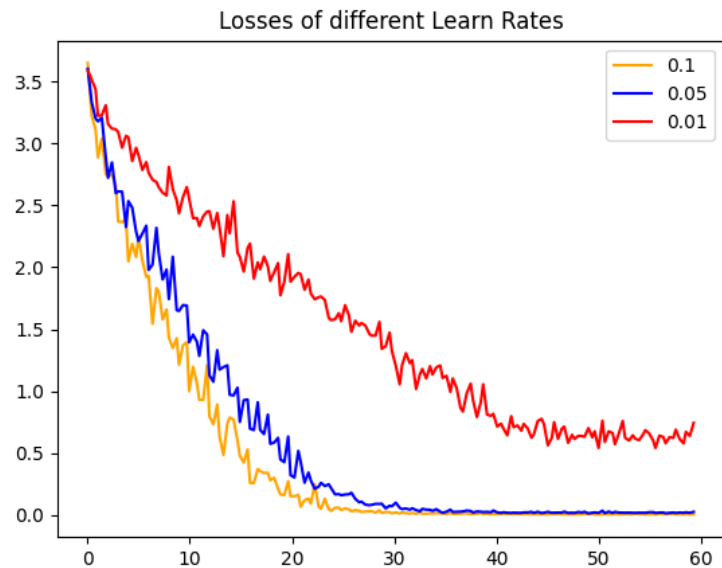


3. 超参数对性能影响

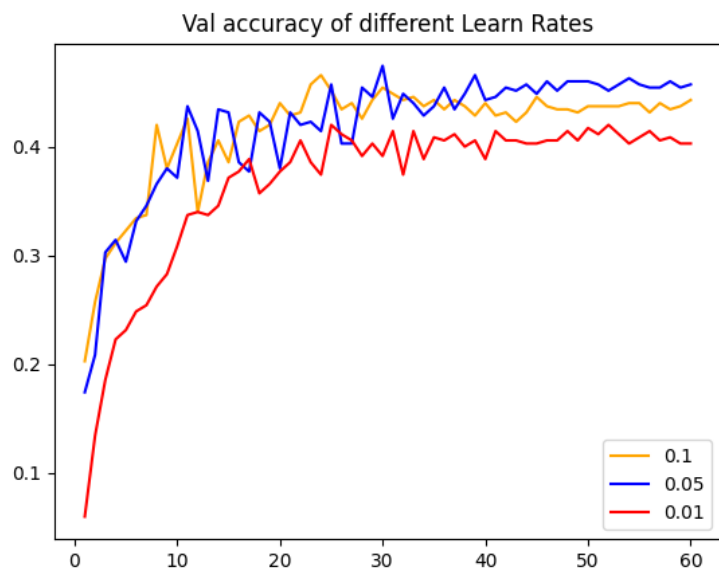
1. 准确率与学习率关系



2. loss与学习率关系



3. 验证集准确率与学习率关系



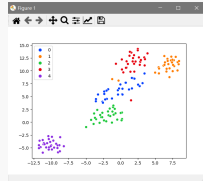
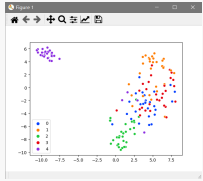
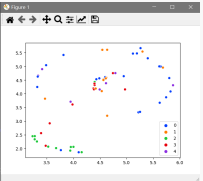
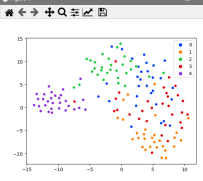
调整网络结构

网络结构	Base Model	自定义网络	ResNet18	Vgg19
val accuracy	51.4%	41.7%	77.43%	79.1%
test accuracy	45.6%	39.2%	76.43%	82.5%

自定义网络：将Base Model中第一层卷积层的卷积核大小改为 3×3 ，stride改为1，padding改为1

ResNet与VggNet均采用了预训练权重，从结果来看，Vgg19的性能最好。

调整损失函数

损失函数	CrossEntropy	MSELoss	L1Loss	CrossEntropy+L2 正则化 (系数 0.0001)
train accuracy	100%	78.0%	6%	91.0%
val accuracy	51.4%	44.5%	3.4%	42.5%
test accuracy	45.6%	43.5%	3.1%	39.7%
t-SNE				

从结果来看，CrossEntropy在这样的多分类问题中性能确实是最好的，MSELoss稍差，而L1Loss则几乎无法进行梯度下降，具体原因我也不知道。最后一项我选择了CrossEntropy+L2正则化损失，希望用权重作为损失项来降低过拟合，从准确率来看过拟合有所缓解，但是总体性能并无改善，从t_sne可视化特征也可以看出特征的提取情况。

添加数据，再次进行训练

数据集	train	val	test
基础训练集	100%	51.4%	45.6%
训练集+额外数据	97.0%	98.0%	50.8%

加入额外数据集后，性能有明显提升，但是该数据集的局限性就是样本数太少，很容易造成过拟合。

标签加入噪声

噪声比例	train	val	test
0	100%	51.4%	45.6%
10%	82.0%	44.5%	39.6%
20%	76.0%	40.8%	39.2%
40%	53%	29.7%	30.9%
60%	57%	13.7%	14.5%

在噪声标签比例达到60%左右模型就已经几乎无法进行分类了。对于对抗噪声的方法，我尝试过添加dropout，可以提升2-3%的准确率；也尝试了正则化，但是没有明显的效果。

