

频谱图分类模型训练实验报告

李林锋 2022013206

一、数据集处理

本次训练选择卷积神经网络（CNN）来对图像特征进行学习和分类。其中先对数据集进行处理，测试集的文件夹下面没有直接将图片放到分类的文件夹下，而是通过命名来进行训练的，所以通过名字将图片放入不同的文件夹下，形成与训练集相同的格式。例如：训练集图片 4630_0.png 放入 0 这类的文件夹下。

二、设备准备

由于训练需要用到 GPU，使用 CPU 计算效率太低，速度过慢，效果不佳，因此使用 GPU 版本。通过 `pip install tensorflow-gpu==2.10.1`，安装 tensorflow 的 gpu 版本，安装 CUDA 和 CDNN，从而设备能够调用笔记本本身的显卡 NVIDIA GeForce RTX 3050 Laptop GPU，准备完毕。

三、数据集准备

首先加载训练集数据，参数包括 `batch_size=12`，表示的是训练批次的数目为 12，每次选择 12 张图片进行训练，主要是由于电脑的 GPU 内存限制，所以每次加载图片数目不能过多，同时为了能够重复结果，生成随机数种子，显示设置图片大小，将 513*800 转化为 256*256，也是考虑内存限制进行的调整。同理，将测试集数据进行相同的操作。

四、CNN 模型定义

定义 CNN 模型，定义的伪代码如下：

```
定义函数 create_cnn_model(input_shape, num_classes):  
    初始化一个空的顺序模型 (Sequential)  
    # 第一卷积层  
    添加卷积层: Conv2D(32, kernel_size=(3, 3), 激活函数='relu', 填充方式='same', 输入形状  
=input_shape)  
    添加批量归一化层: BatchNormalization()  
    添加最大池化层: MaxPooling2D(pool_size=(2, 2))  
    添加 Dropout 层: Dropout(0.3)  
    # 第二卷积层  
    添加卷积层: Conv2D(64, kernel_size=(3, 3), 激活函数='relu', 填充方式='same')  
    添加批量归一化层: BatchNormalization()  
    添加最大池化层: MaxPooling2D(pool_size=(2, 2))  
    添加 Dropout 层: Dropout(0.4)
```

```

# 第三卷积层
添加卷积层: Conv2D(64, kernel_size=(3, 3), 激活函数='relu', 填充方式='same')
添加批量归一化层: BatchNormalization()
添加最大池化层: MaxPooling2D(pool_size=(2, 2))
添加 Dropout 层: Dropout(0.4)

# 第四卷积层
添加卷积层: Conv2D(128, kernel_size=(3, 3), 激活函数='relu', 填充方式='same')
添加批量归一化层: BatchNormalization()
添加最大池化层: MaxPooling2D(pool_size=(2, 2))
添加 Dropout 层: Dropout(0.5)

# 第五卷积层
添加卷积层: Conv2D(128, kernel_size=(3, 3), 激活函数='relu', 填充方式='same')
添加批量归一化层: BatchNormalization()
添加最大池化层: MaxPooling2D(pool_size=(2, 2))
添加 Dropout 层: Dropout(0.5)

# 第六卷积层
添加卷积层: Conv2D(128, kernel_size=(3, 3), 激活函数='relu', 填充方式='same')
添加批量归一化层: BatchNormalization()
添加最大池化层: MaxPooling2D(pool_size=(2, 2))
添加 Dropout 层: Dropout(0.5)

# 展平层
添加展平层: Flatten()

# 全连接层
添加全连接层: Dense(128, 激活函数='relu')
添加批量归一化层: BatchNormalization()
添加 Dropout 层: Dropout(0.8)

# 输出层
添加输出层: Dense(num_classes, 激活函数='softmax')

返回模型

```

随后，创建模型，编译模型，训练模型，同时进行回调设置，保存训练最好的结果。同时绘制出训练集与测试集的 accuracy 和 loss 图，如图 1 所示。

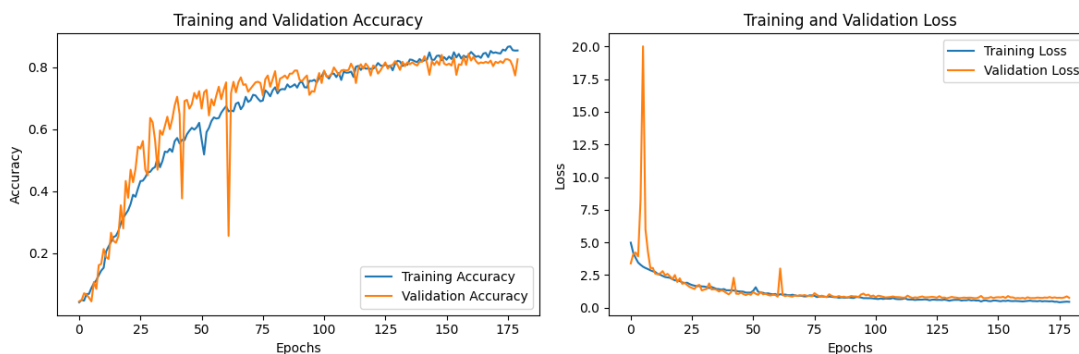


图 1

四、讨论

在训练过程中，需要判断卷积层中的各个参数的情况，所以需要尝试，但是由于频谱图本身便较为复杂，所以尝试使用较深的网络层，但是使用较深的网络层和卷积层的卷积核数目提升会存在严重的过拟合过程，所以可以尝试将数据标准化和 dropout。在为防止过拟合时可以得到训练情况如下（过拟合）：

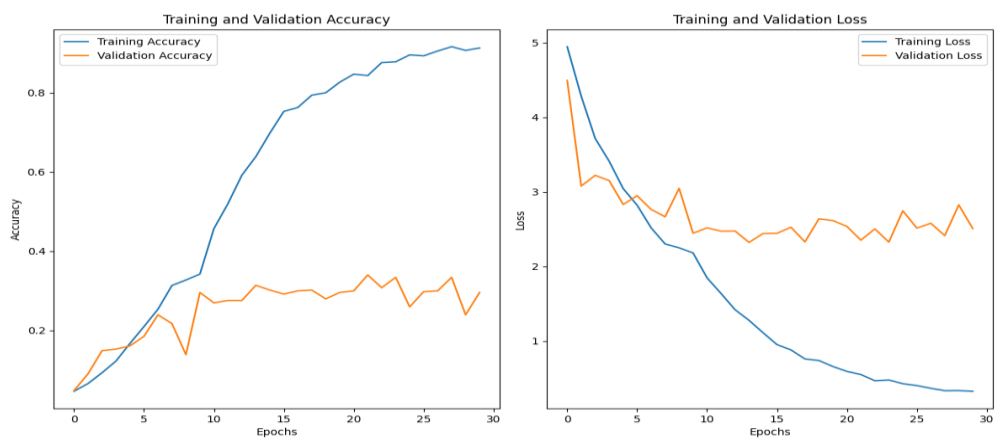


图 2

最后经过训练以后得到的训练集情况为 Loss: 0.718，Accuracy: 0.843，训练集的情况为 Loss: 0.692，Accuracy: 0.851。结果较好，训练 180 epoch 次数精确度上升速率逐渐降低，不加以更多的训练。