**实验6、实验7迁移学习**

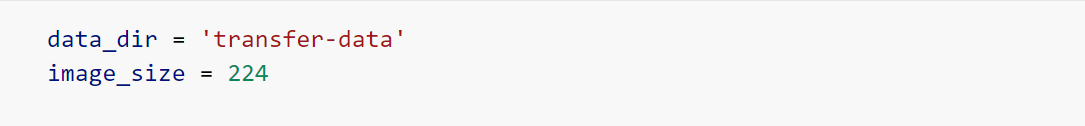
1. **实验内容**

实验内容：使用PyTorch 的数据集套件从本地加载数据的方法，迁移训练好的大型神经网络模型到自己模型中的方法，迁移学习与普通深度学习方法的效果区别，两种迁移学习方法的区别

实验环境：Python-3.11;PyTorch-cu121;CUDA-12.1;VS code/JupyterNotebook

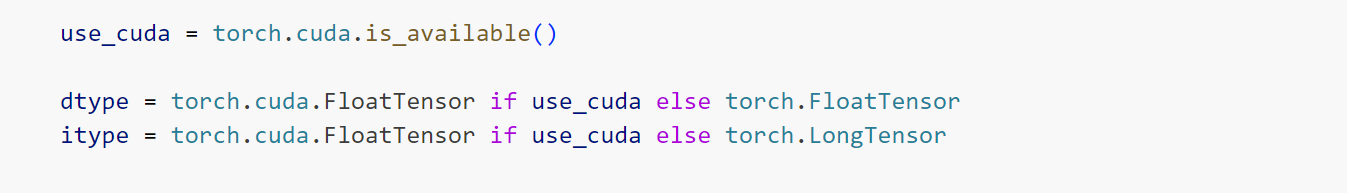
1. **实验过程和结果**
2. **从图片文件中加载训练数据**

****

****

****

以上代码引入了必要的库，设置了数据集路径，运用datasets中的ImageFolder加载训练和验证数据集，针对图像的三个颜色通道通过设置均值和标准差对图片进行标准化，并且对图像进行了随机翻转，这样的操作可对图像进行增强便于机器进行特征识别，图像随机翻转使训练的模型鲁棒性更高。



该段代码对是否使用cuda变量定义，针对使用cuda情况改变数据类型

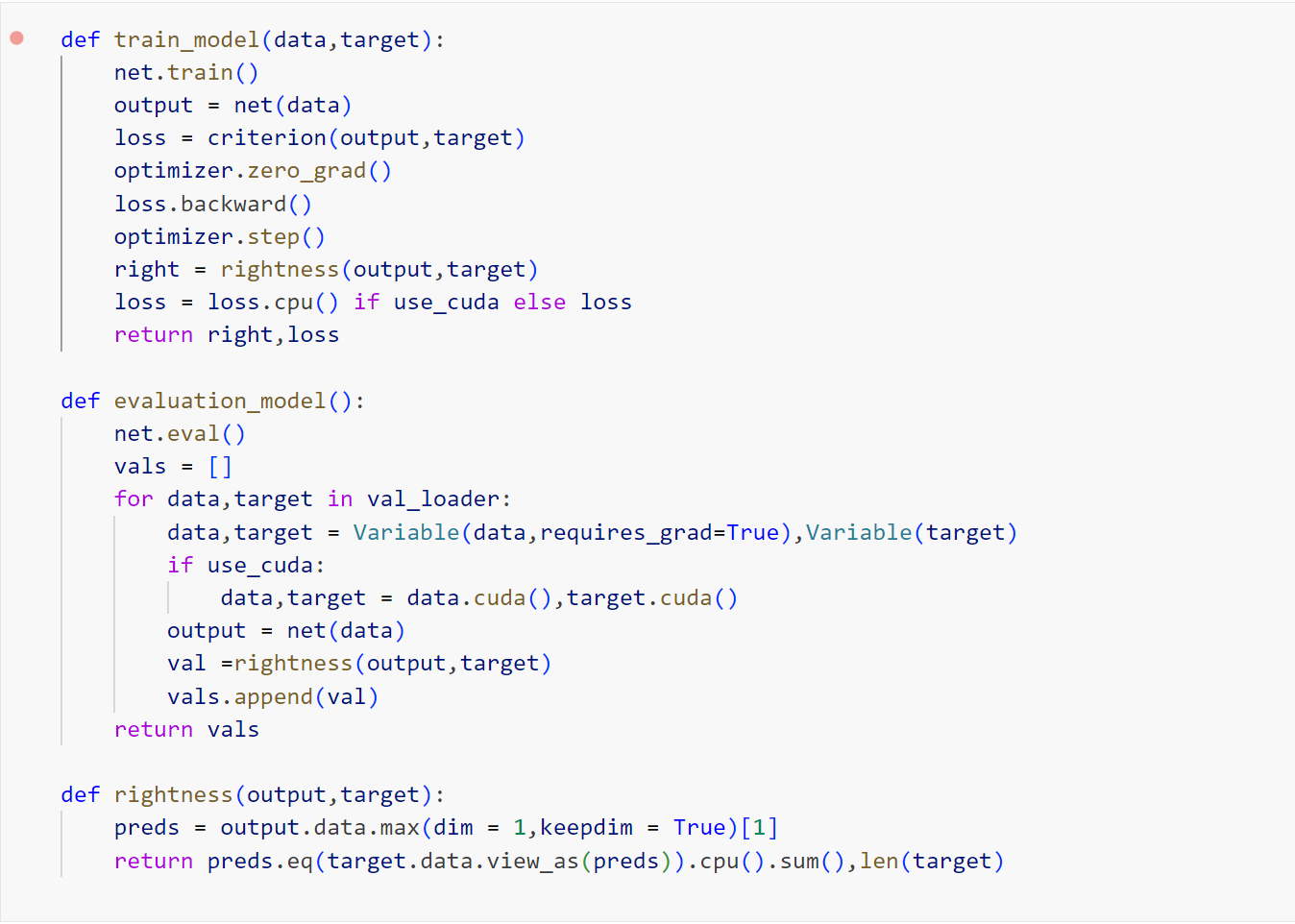




该段代码定义了imshow函数，用于输出数据集图像。数据集中图像大小为3\*224\*224而plt中imshow函数要求图片为224\*224\*3，且其颜色取值如果为整数应在[0,255]范围中，浮点数应在[0,1]中，代码首先对图片的维度次序进行了转换，再通过去标准化将图片颜色取值设定到规定范围中，通过plt.imshow函数绘制图片。

1. **训练一个普通的卷积神经网络**

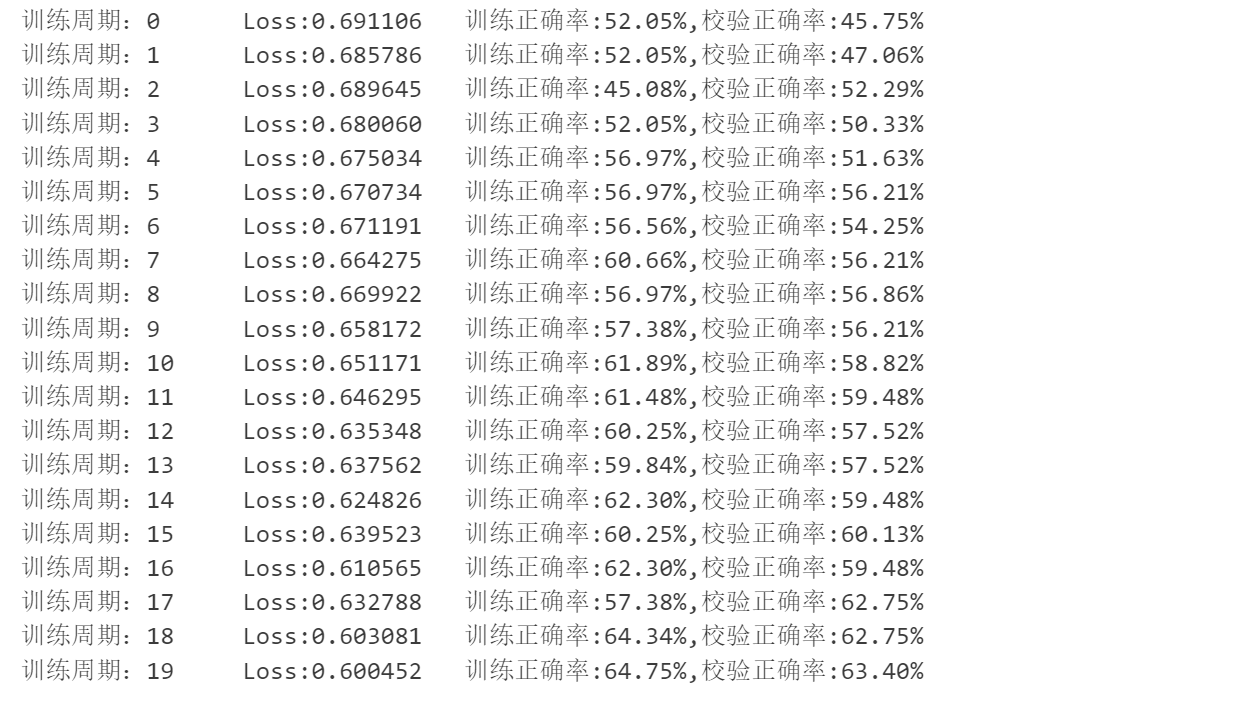
****

****

上面代码参照实验四通过类定义的方式构建了一个卷积神经网络，定义函数rightness输出预测正确个数和总数，train\_model对模型进行一次训练返回正确率和损失值，evaluation\_model使用校验集对模型有效性进行验证。

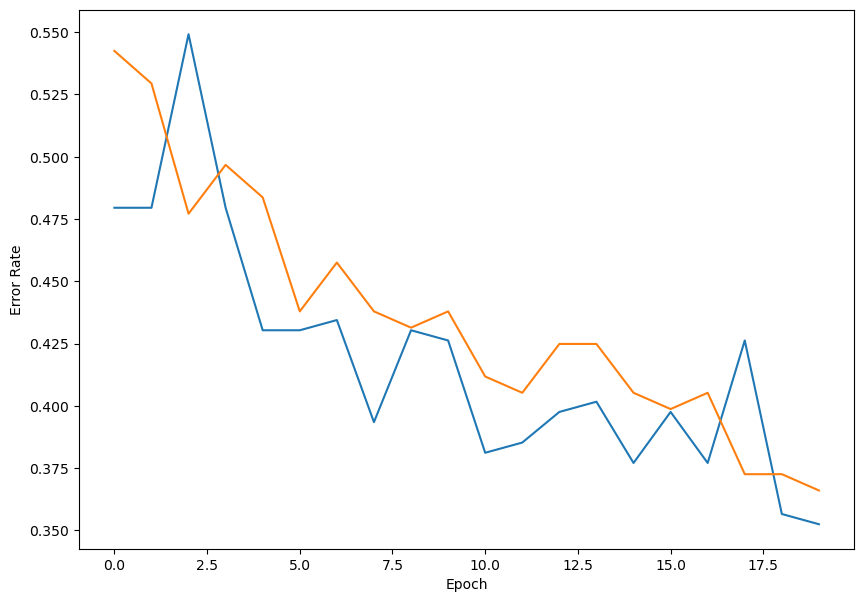


这段代码正式开始训练神经网络，与上一实验相同，每轮分成若干撮训练，每撮计算正确率和损失值追加入列表中，每轮进行一次校验集验证，验证结果追加入列表，每轮计算一次平均正确率和损失值输出并保存到record中用于绘制训练效果图。其中每轮训练结束进行一次判断验证正确率和最优的正确率，将20轮训练中正确率最高的模型赋值给best\_modeel。

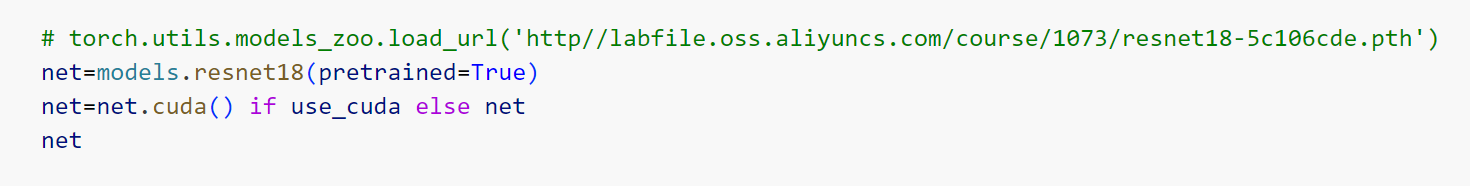
****

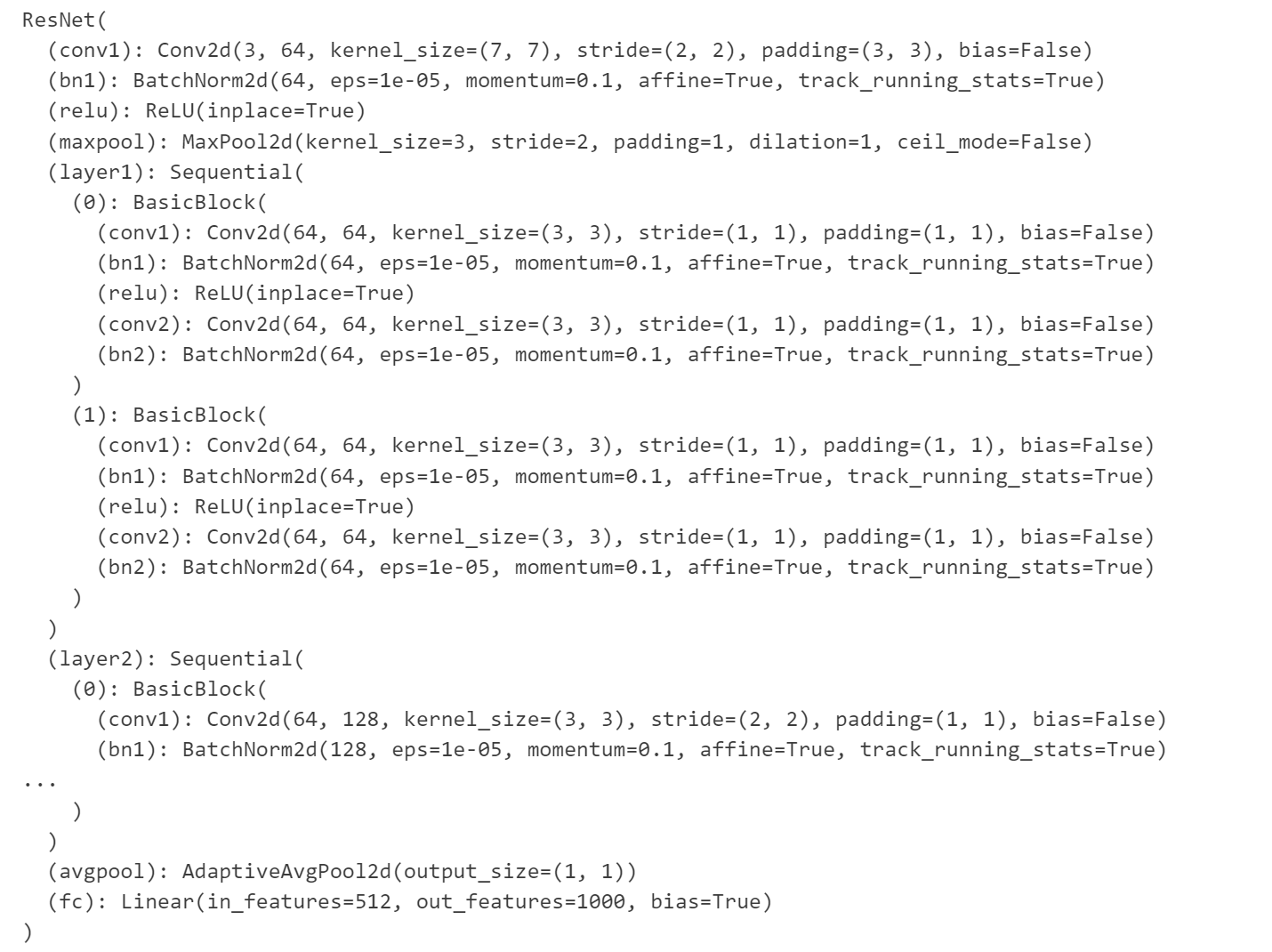
****

训练过程中采用了dropout，在eval模式下禁用dropout再次使用验证集对模型正确率进行测试，输出正确率为63.398%。对训练过程中record中保存的每轮训练的训练正确率和校验正确率通过绘图输出。

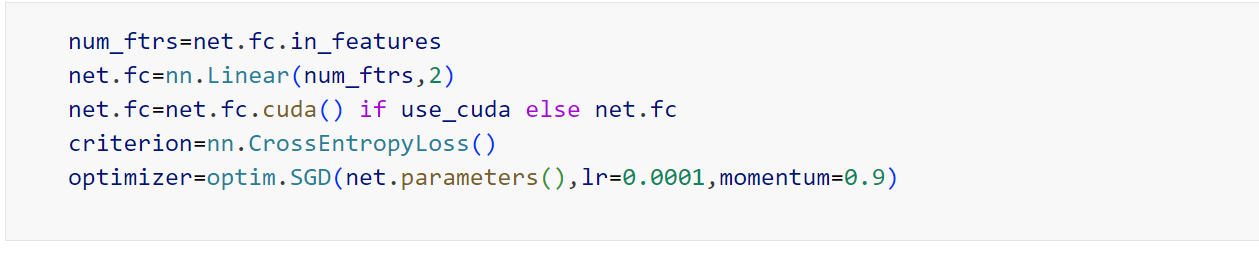


1. **加载已经训练好的ResNet进行迁移学习**

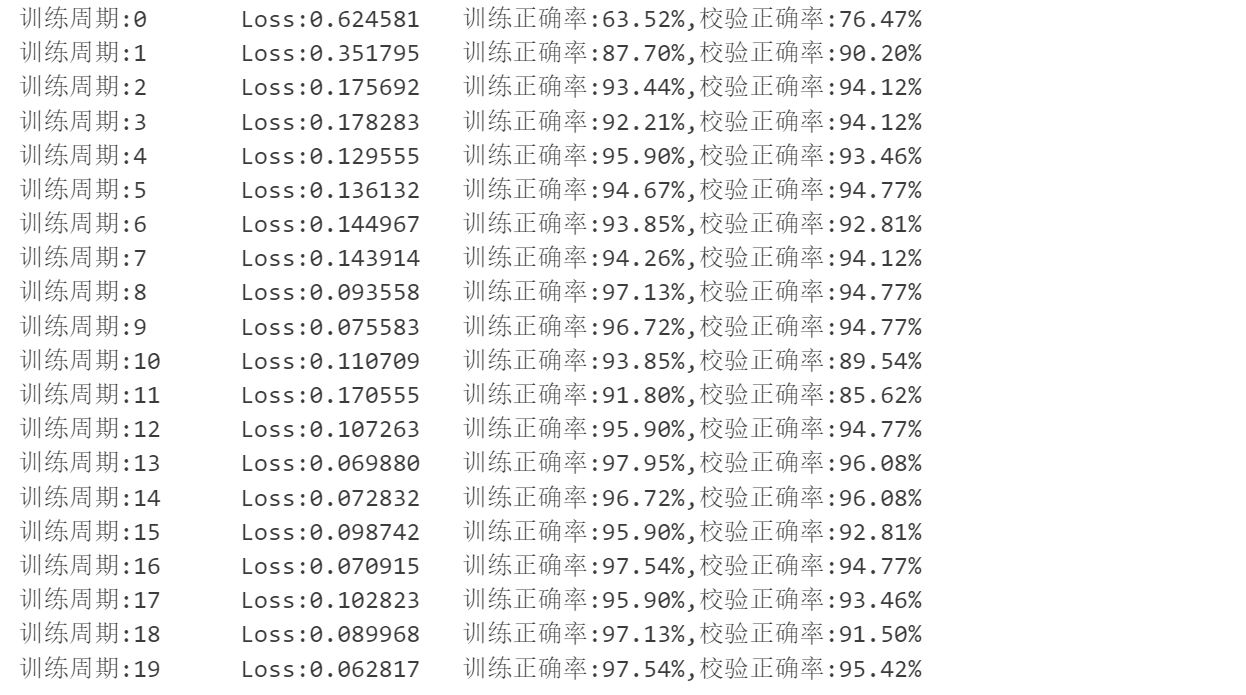
****

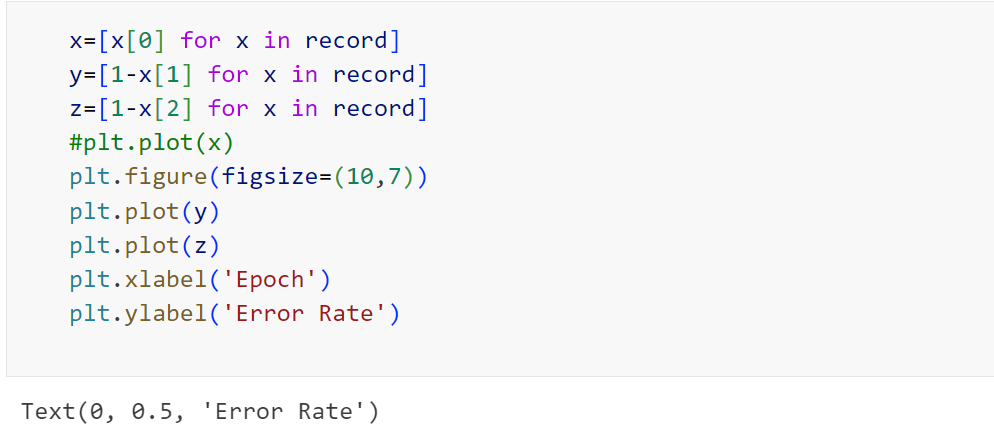
****

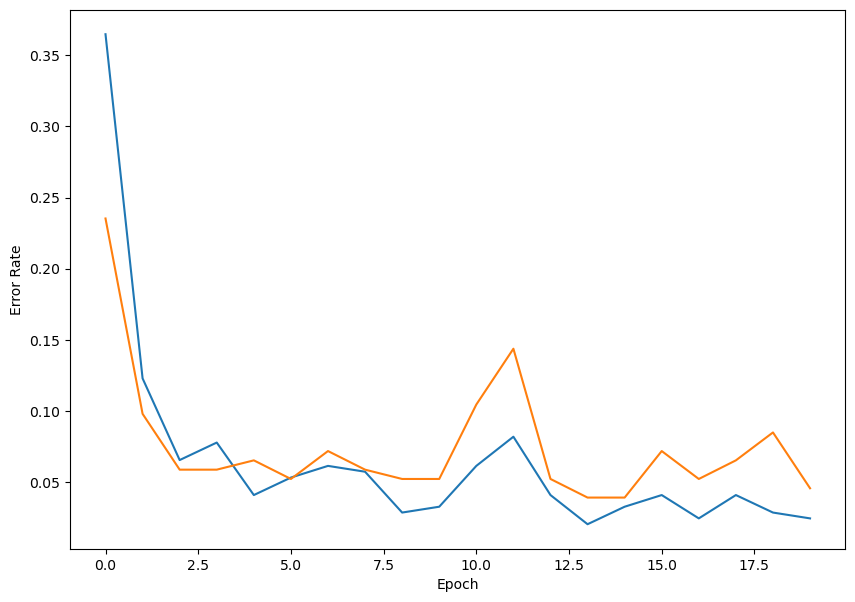
* 1. **预训练模式**

****

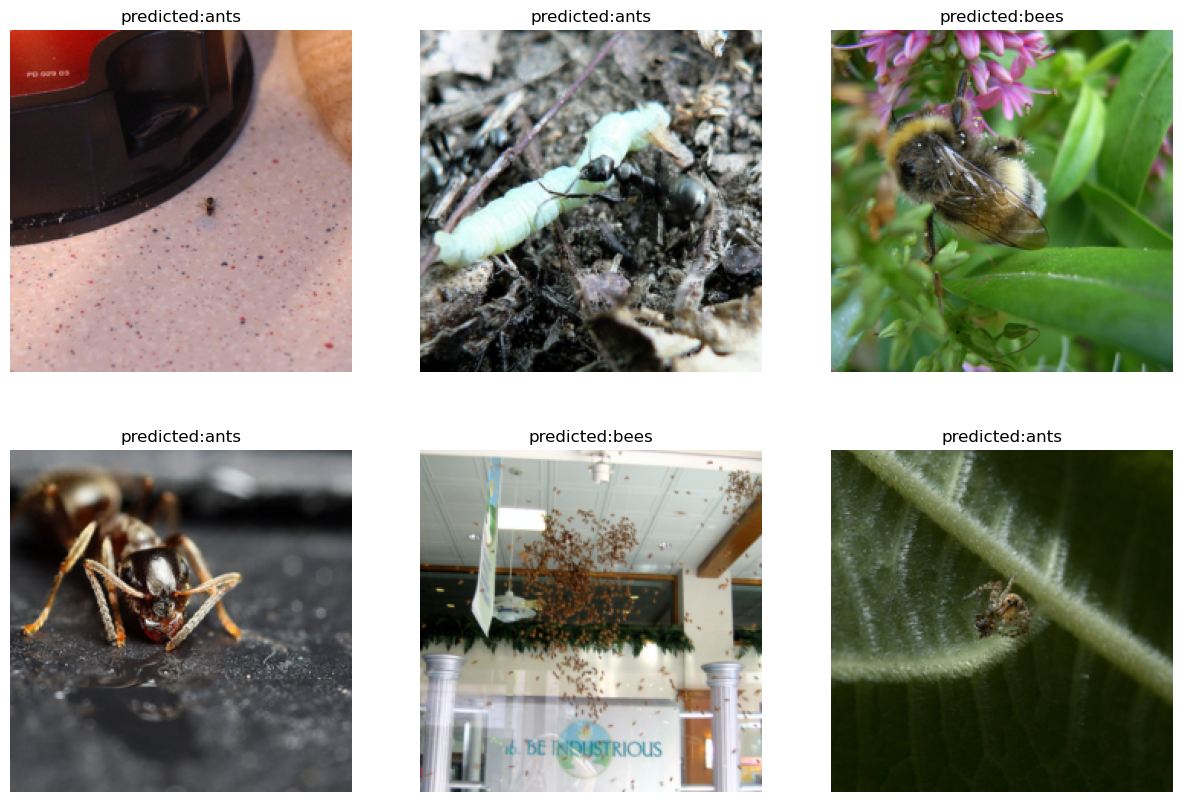


****

****





****

第三部分代码对已经训练好的残差网络进行迁移学习。通过调用输出ResNet18的结构，其中最后的全连接层为隐层512输出层1000的前馈层，而我们只需要辨别蜜蜂和蚂蚁，因此将最后的前馈全连接层修改为512隐层和2输出神经元。通过将resnet18首先实例化，再将变量的fc层直接修改，就完成了迁移嫁接过程。

第一次尝试采用预训练模式，将预训练好的ResNet参数直接迁移作为新网络训练的起点，便于新的网络训练中可以更快的收敛，参数达到局部最优，这种方式灵活性和适应性强。训练的过程和其他网络无异。

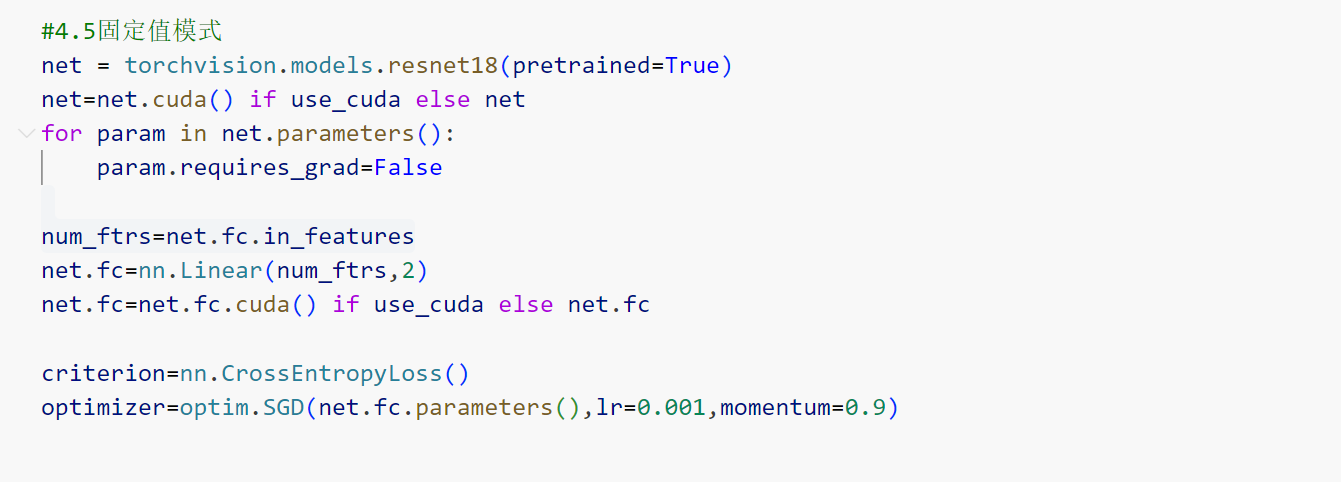
同样输出每轮训练中的损失值和训练校验正确率，可见通过迁移学习得到的模型正确率远高于自搭建的卷积神经网络，训练过程中正确率的收敛速度也更快。

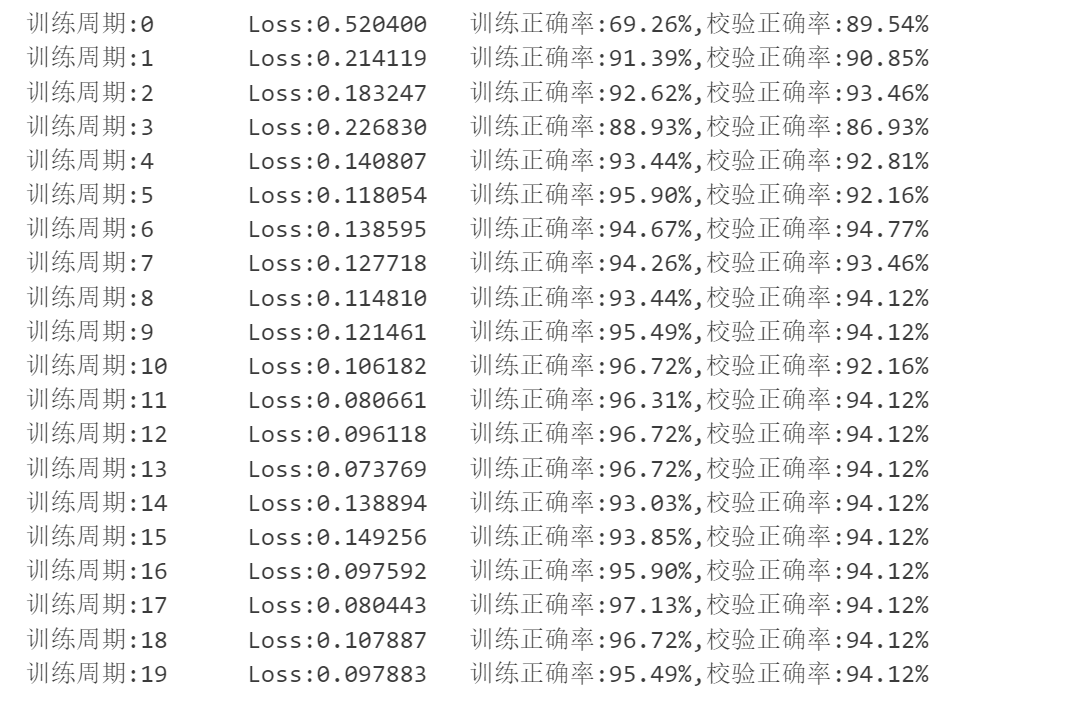
最后定义了一个可视化模型函数visualize\_model，其实际作用是从验证集中抽取若干数据送入模型中，以子图形式显示图像和模型对它的预测值。指导书代码出现了一些错误，图像不能以子图形式显示，我将前一部分imshow的代码稍作修改替换掉了子图绘制部分的函数，保证图像能以正常的色彩值输出。

第二次代码如下，采用了固定值模式，就是训练过程中不改变迁移部分的参数，只改变更改大小的前馈全连接层参数的值，当使用反向传播算法的时候，误差反传过程会在迁移模块中停止，从而不改变迁移模块中的权重数值。采用这种方式，可保留被迁移部分的知识不被破坏。实现的方法很简单，将模型实例化后将参数全部设定为requires\_grad=False,这样自动微分和反向传播就不会改变参数值，再将前馈全连接层惊醒替换，送入训练的轮回中训练。

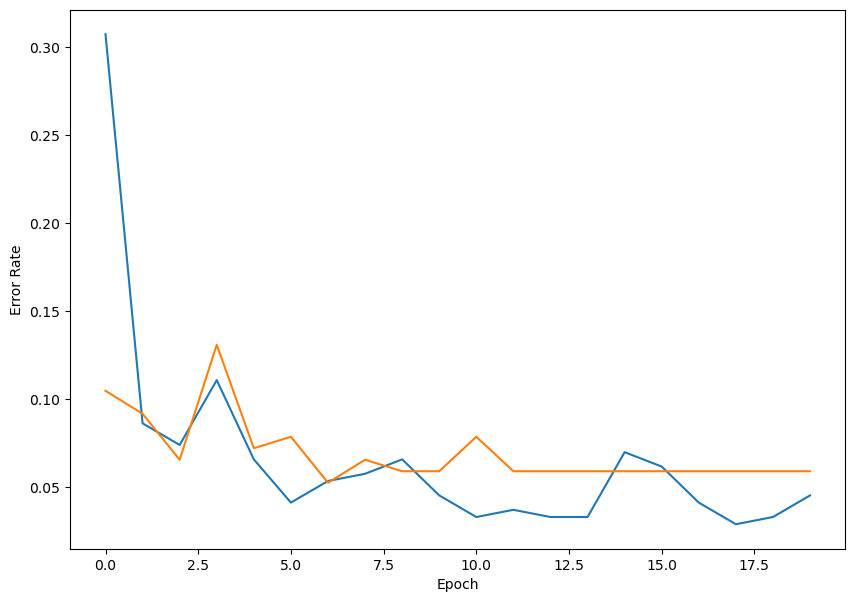
虽然固定值模式不如预训练模式灵活，通过训练结果比较可以看出固定值模式训练的正确率变化更加平缓，但预训练模式下的正确率能够达到更高的数值。

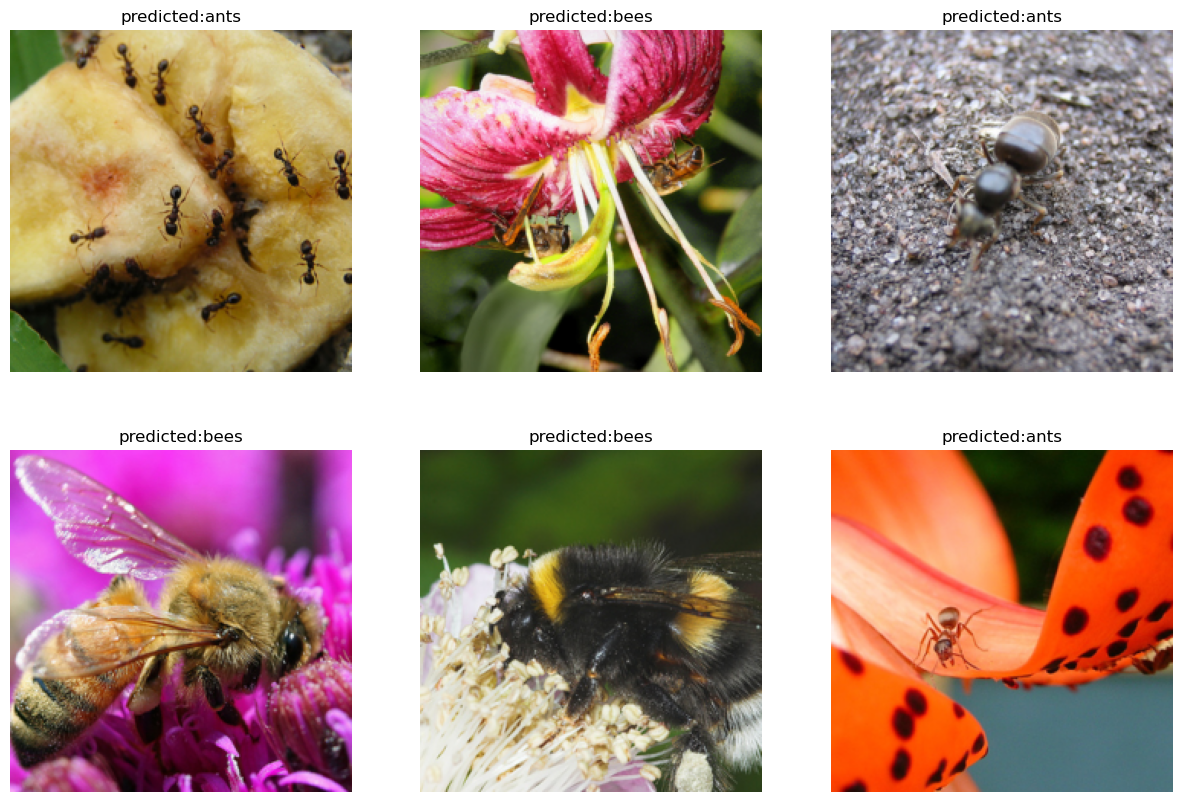
* 1. **固定值模式**

****

****

****

****

****

1. **心得体会**

本次实验练习了如何进行迁移学习，实验大致围绕一个主要问题分为三部分，针对蚂蚁蜜蜂分类问题采用卷积神经网络、迁移学习预训练的残差神经网络和固定值的残差神经网络进行分类。

迁移学习主要有预训练和固定值两种迁移方式。预训练模型是将网络的结构和参数迁移到新的任务中，参数的初始化取值设定为源网络的数值，在新的网络中，所有的参数都会被重新训练。固定值模型是将源网络的结构和参数迁移到新的任务中后，新网络中参数的取值设定为原网络的数值，不参与反向传播和训练。而在实际应用过程中，预训练方式往往能取得更好的迁移效果。

实验中迁移过程十分简便，将预训练的模型实例化后直接更改模型某一层的属性值。迁移学习通过利用已训练好的模型，构建用于解决特定新问题的方式大大减少了个人训练模型的时间和资源开销。