**实验4 迁移resnet50猫狗图像分类**

计科2002班 202013407043 张艺汶

**一、实验目的**

1. 掌握迁移学习方法快速训练resnet50的分类模型；

2. 了解使用pytorch对resnet50进行部分参数进行微调的方法。

**二、实验内容**

1. 通过代码自动下载模型并直接调用；

2. 对当前迁移过来的模型进行全连接层的调整；

3. 模型训练及测试结果。

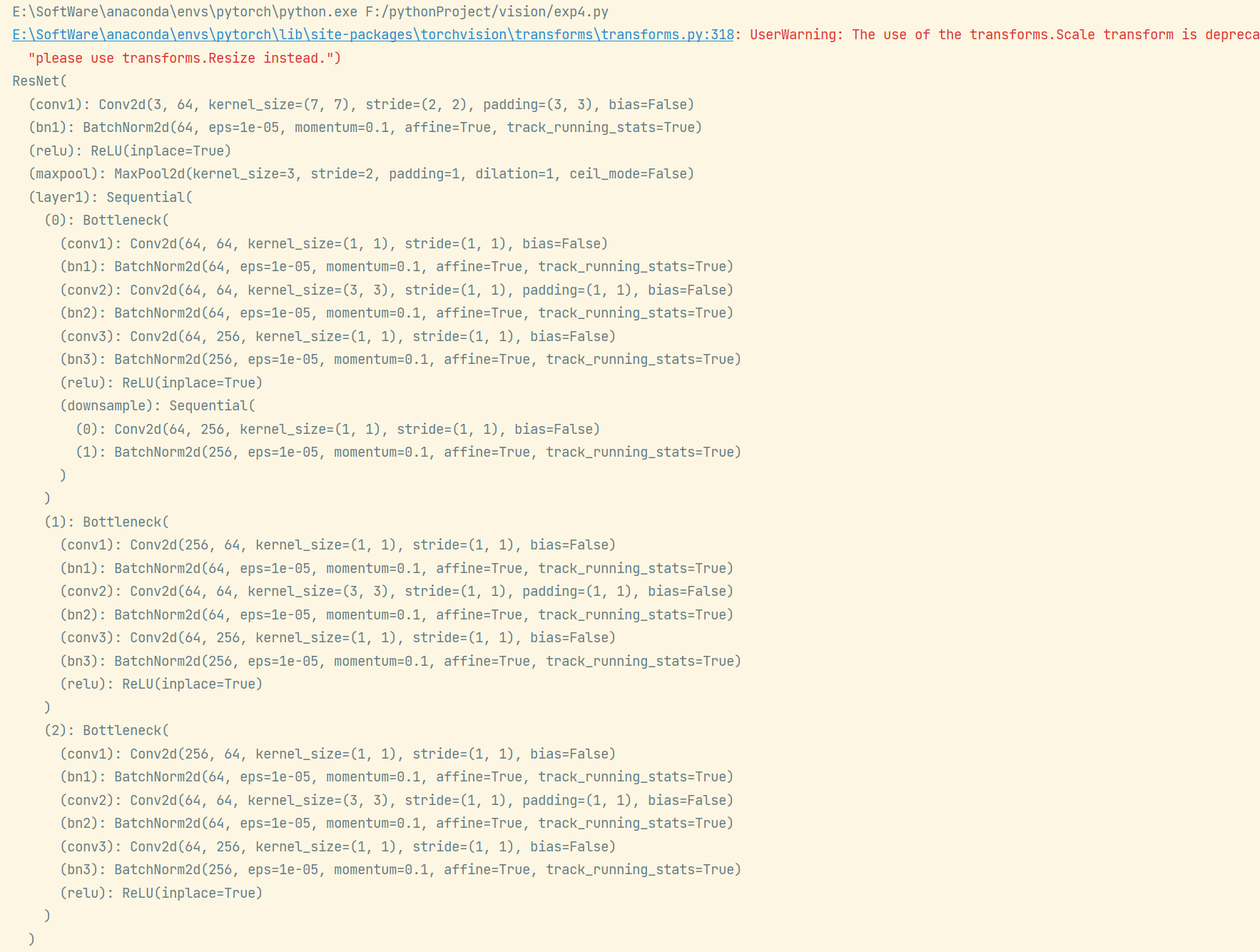
**三、实验指导**

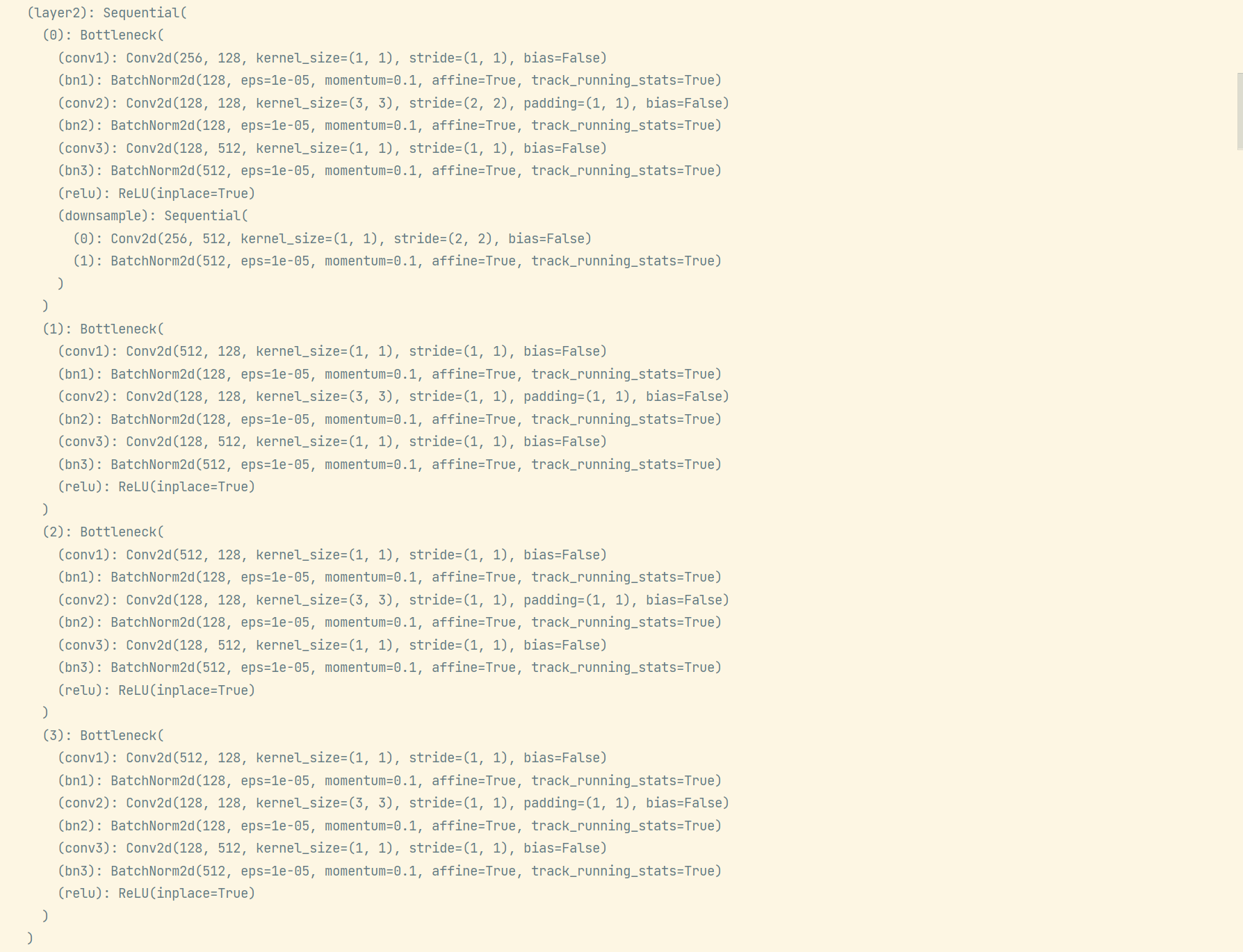
**1.通过代码自动下载模型并直接调用**

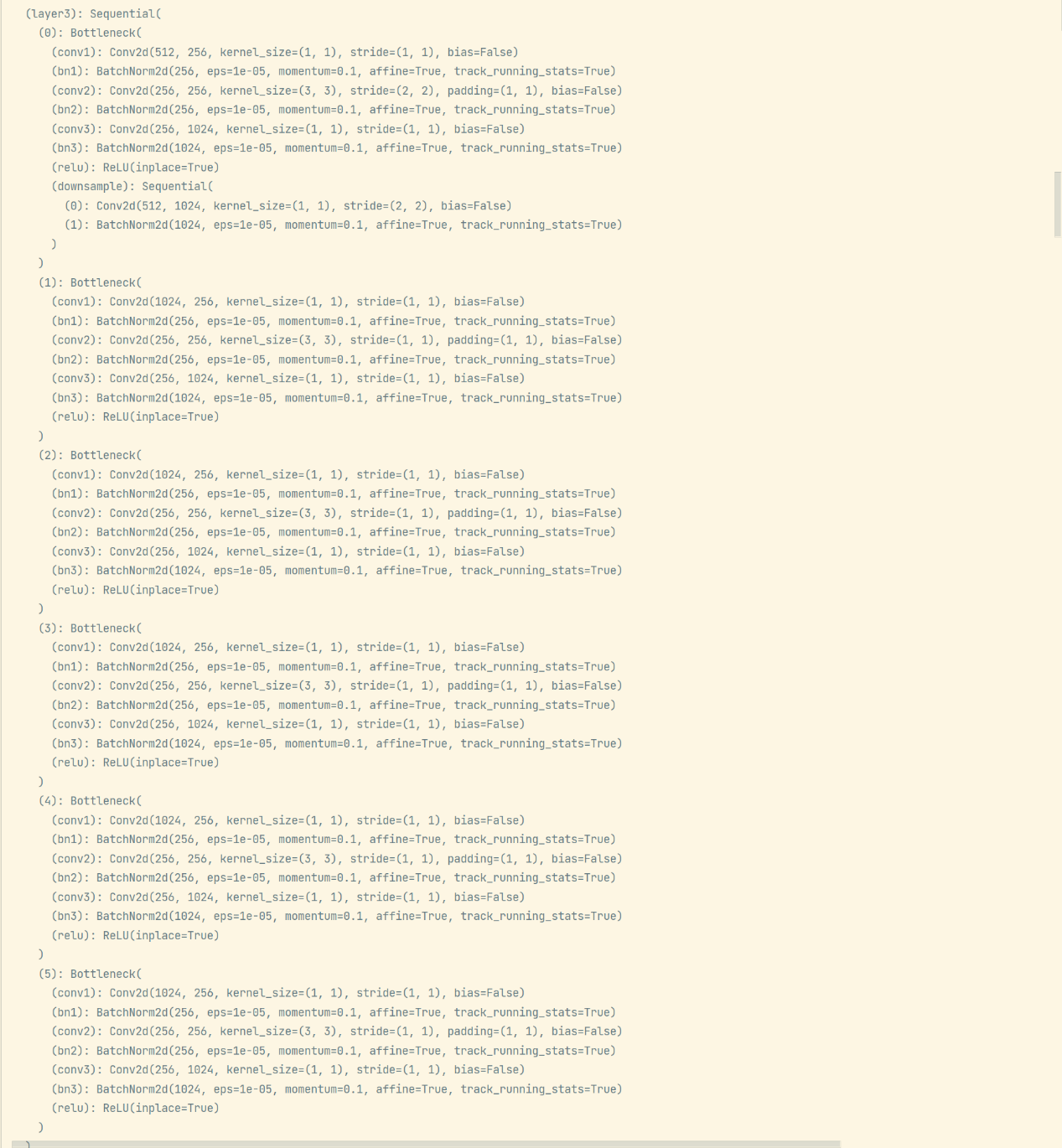
首先需要下载已经具备最优参数的模型，这需要对我们之前使用的 model = Models()代码部分进行替换，此时不需要自己搭建和定义训练模型了，而是通过代码自动下载模型并直接调用（此时干脆把前面对数据集进行的处理再重新做一遍）：

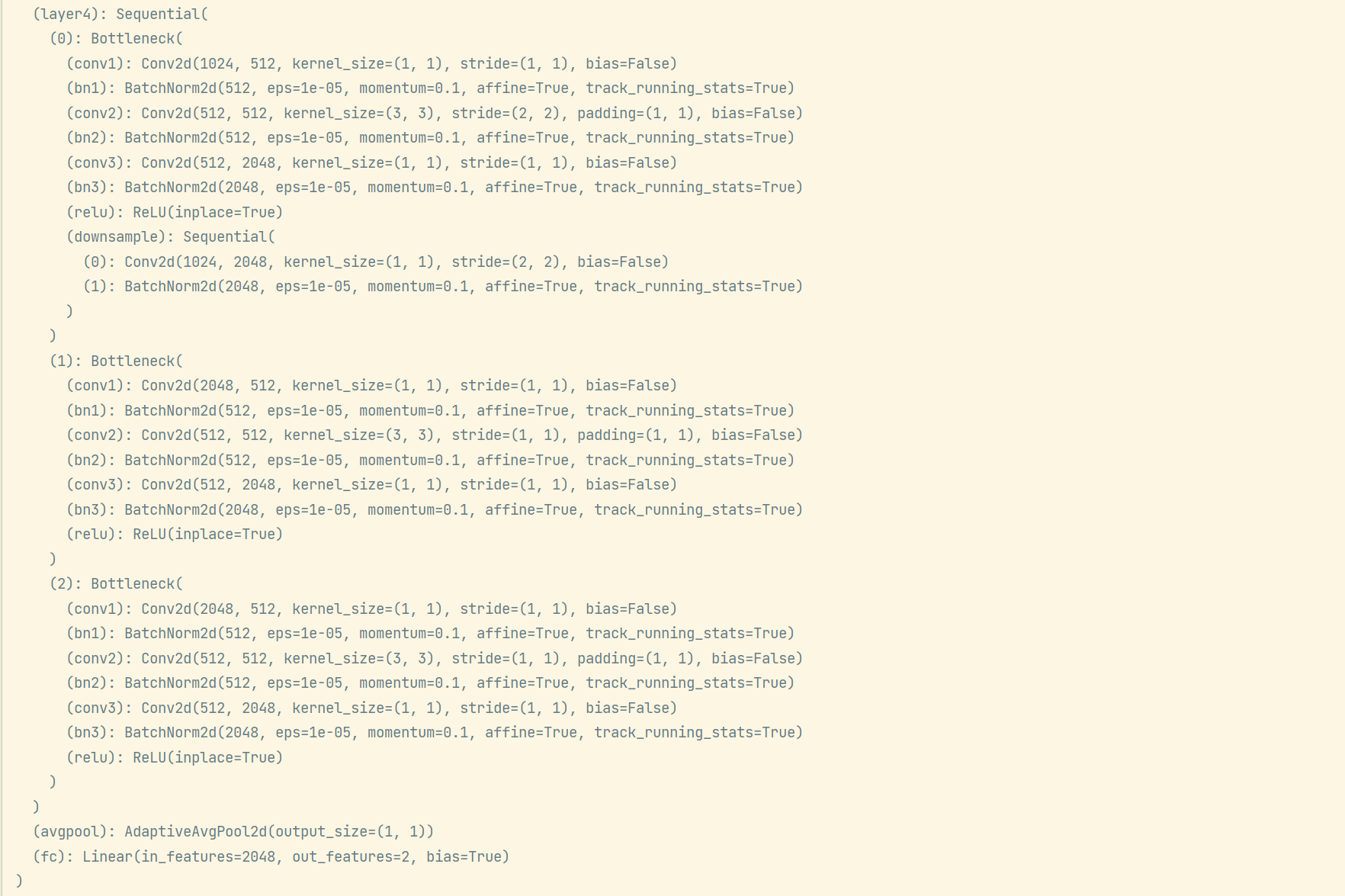


运行结果：







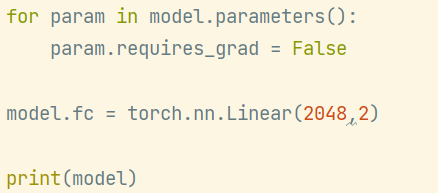


**2. 对当前迁移过来的模型进行全连接层的调整**

尽管迁移学习要求我们需要解决的问题之间最好具有很强的相似性，但是每个问题对最后输出的结果会有不一样的要求，而承担整个模型输出分类工作的是卷积神经网络模型中的全连接层，所以在迁移学习的过程中调整最多的也是全连接层部分。

其基本思路是冻结卷积神经网络中全连接层之前的全部网络层次，让这些被冻结的网络层次中的参数在模型的训练过程中不进行梯度更新，能够被优化的参数仅仅是没有被冻结的全连接层的全部参数。

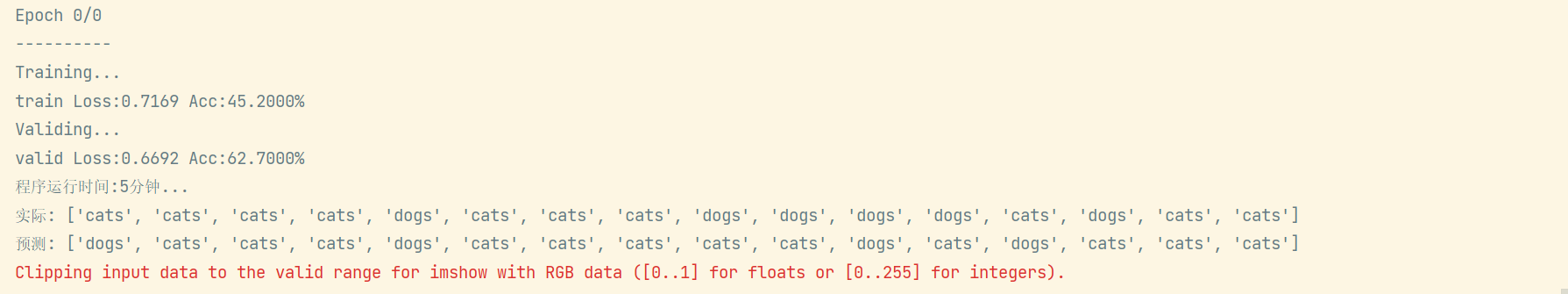
首先，迁移过来的 resnet50 架构模型在最后输出的结果是 1000 个，在我们的问题中只需两个输出结果，所以全连接层必须进行调整：



**3.模型训练及结果**



运行结果：



**4.举例说明**

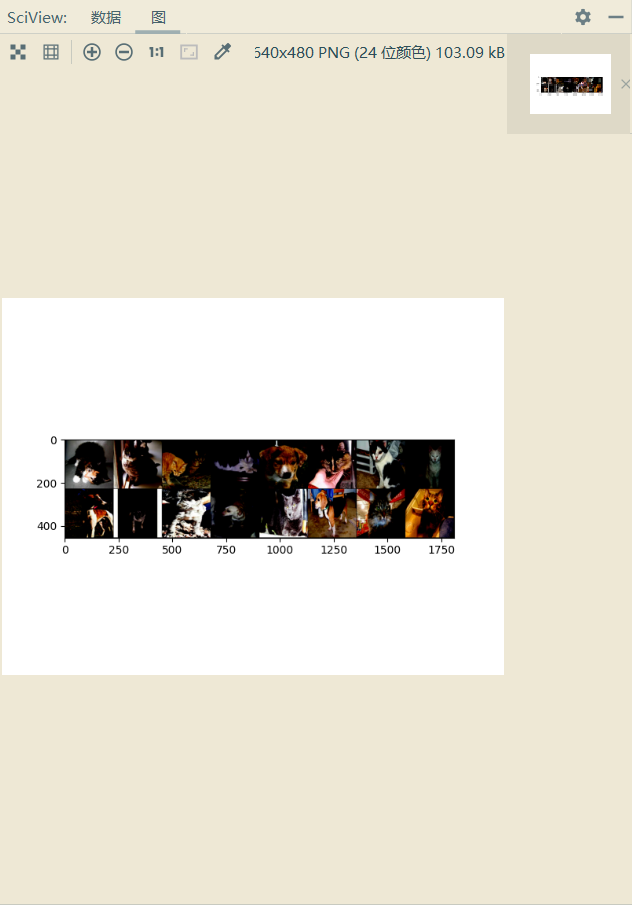


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

实际: ['cat', 'dog', 'dog', 'cat', 'cat', 'dog', 'cat', 'cat', 'dog', 'dog', 'dog', 'cat', 'cat', 'dog', 'cat', 'dog']

预测: ['cat', 'dog', 'dog', 'cat', 'cat', 'dog', 'cat', 'cat', 'dog', 'dog', 'dog', 'cat', 'cat', 'dog', 'cat', 'dog']

运行结果：



**四、实验小结**

首先使用torchvision.transforms包进行图片变换：

1. Compose函数用于串联多个图片变换的操作；
2. ToTensor函数用于变换图像数据为tensor并shape转化为[channel, h, w]
3. Normalize函数用于逐channel的对图像进行标准化（均值变为0，标准差变为1），可以加快模型的收敛

接着使用torchvision.datasets和torchvision.utils对数据进行处理：

1. DataLoader函数用于加载数据以及给数据加上标签

再使用torchvision.models中的resnet50模型进行猫狗图像分类，其基本思路是冻结卷积神经网络中全连接层之前的全部网络层次，让这些被冻结的网络层次中的参数在模型的训练过程中不进行梯度更新，能够被优化的参数仅仅是没有被冻结的全连接层的全部参数：

（1）train函数表明是否为训练集

然后使用torch包对模型进行训练：

1. CrossEntropyLoss为损失函数，描述了两个概率分布之间的距离
2. Adam提供一个能够分析数据的框架,同时让数据的审阅者和其他接收者们清楚地了解数据从收集到分析再到结果的系谱
3. backward属于torch.autograd函数库,在深度学习过程中对函数进行反向传播,计算输出变量关于输入变量的梯度
4. optimizer.step用于更新模型
5. sum对输入的tensor数据的某一维度求和
6. item从只包含一个元素的tensor中提取值

最后进行数据测试，对数据进行独热编码并把原始图像保存起来，用torchvision包显示结果：

1. make\_grid用于把几个图像按照网格排列的方式绘制出来
2. transpose的作用是调换数组的行列值的索引值
3. max用于返回输入张量给定维度上最大值,并同时返回每个最大值的位置索引
4. imshow和show用来显示图像

通过本次实验，我学习了迁移学习，掌握了迁移学习方法快速训练resnet50的分类模型并且了解使用了pytorch对resnet50进行部分参数进行微调的方法。因为通过迁移学习可以节省大量的时间和精力，而且最终得到的结果不会太差，并且我还认识到迁移学习非常强大，能快速解决同类问题，对类似的问题不用再从头到尾对模型的全部参数进行优化。