**自然语言处理结课作业**

使用CNN和Tensorflow 2.0实现猫狗识别

姓 名 ： 侯宇鹏

学 号 ： 2271982

专 业 名 称 ： 计算机技术

授 课 老 师 ： 肖桐

学 院 ： 计算机科学与工程学院

二○二二年秋季

### 一、实验背景

由于我在本科阶段没有接触太多关于深度学习的内容，因此希望从比较简单基础的CNN入手，并结合老师上课讲的关于tensorflow的一些内容，做一个入门的实验——猫狗分类（二分类实验）,部分内容参考了CSDN和B站的视频讲解，虽然过程磕磕绊绊，但也借此机会了解了CNN的理论背景和tensorflow的相关安装问题，由于本人计算机无独立显卡，故使用CPU版本的tensorflow。

CNN的结构可以分为3层：卷积层(Convolutional Layer) - 主要作用是提取特征。池化层(Max Pooling Layer) - 主要作用是下采样(downsampling)，却不会损坏识别结果。全连接层(Fully Connected Layer) - 主要作用是分类。对于这三个部分的具体设计会在第四部分详细介绍。

### 二、实验目标

用CNN和tensorflow2.0.0实现简单的猫狗分类，在过程中使用25000张图片进行训练，训练过程中和训练结束时分别将模型进行保存，并生成中间结果，最后在测试集上进行测试，得出精确度，同时可以在互联网上随便找一些猫狗的照片让模型去判断是猫还是狗。

### 三、实验环境

Windows11

Python3.9.12

Tensorflow2.0（CPU版本）

### 四、实验内容

数据集：

训练集共25000张图片：12500张狗的图片和12500张猫的图片

测试集共12500张图片，其中猫和狗的图片是乱序的，最终程序会随机选取测试集中的一张图片送给模型进行预测。

##### 1.Inputdata模块

用于读取用于训练的猫狗图片，由两个方法组成。

1. get\_files方法，需要传入一个参数：存放训练图片的文件地址，之后返回值为两个列表：一个是乱序后的图片路径列表，一个是乱序后的标签列表。这两个列表作为输入值传入后面的get\_batch方法中。



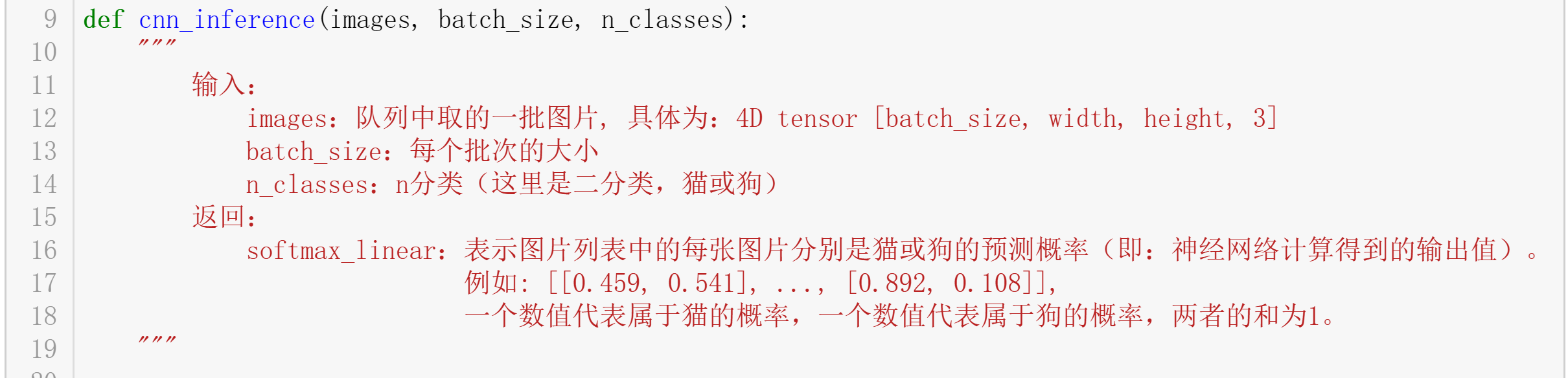
②get\_batch方法，需要传入由上述方法的返回值的两个列表image和batch，图片的高度和宽度，每批训练的图片数量batch\_size，capacity为队列中的最大容量，返回值为两个张量：分别是四维张量image\_batch和一维张量label\_batch



##### 2.model模块

用于进行训练，由五个方法构成

①cnn\_inference方法：输入为队列中取得的一批图片image，每一批图片的大小batch\_size，分类数n\_class，这里是2分类，n\_class就是2；返回值为softmax\_linear，表示图片列表中每张图片分别是猫和狗的预测概率。



模型结构：

第一层卷积核是3\*3的，有三个通道，共16个，第一层池化层进行maxpool，再用lrn（）方法进行局部归一化；

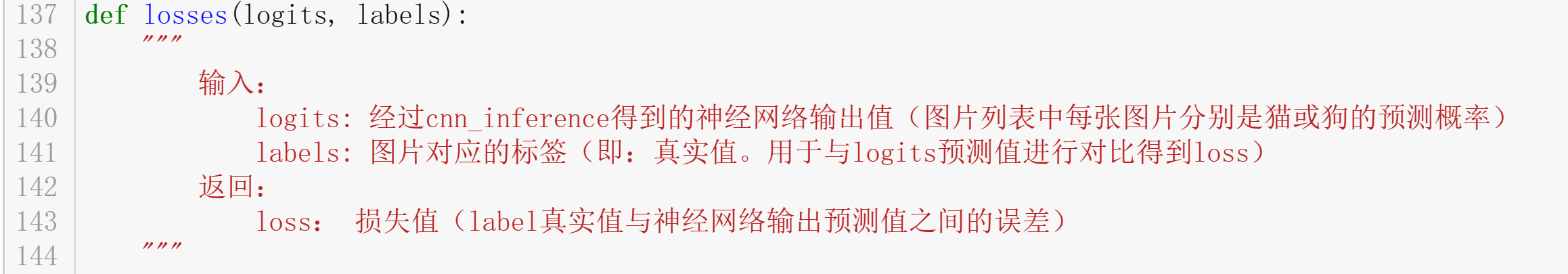
第二层的卷积核是3\*3的，但由16个通道，因为对应第一层卷积核输出的特征图的数量，共16个；

第三层全连接层将池化层的pool2调整成shape大小，然后将reshape和weight矩阵相乘再加上偏差后传入relu激活方法进行非线性话化处理；

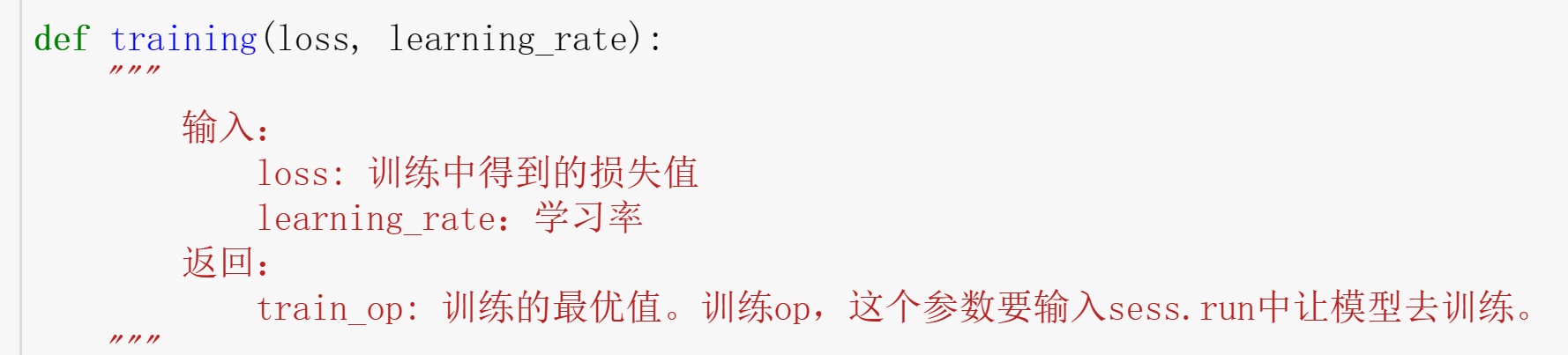
第四层全连接层与第三层相似；

第五层为输出层，将全连接层进行输出；

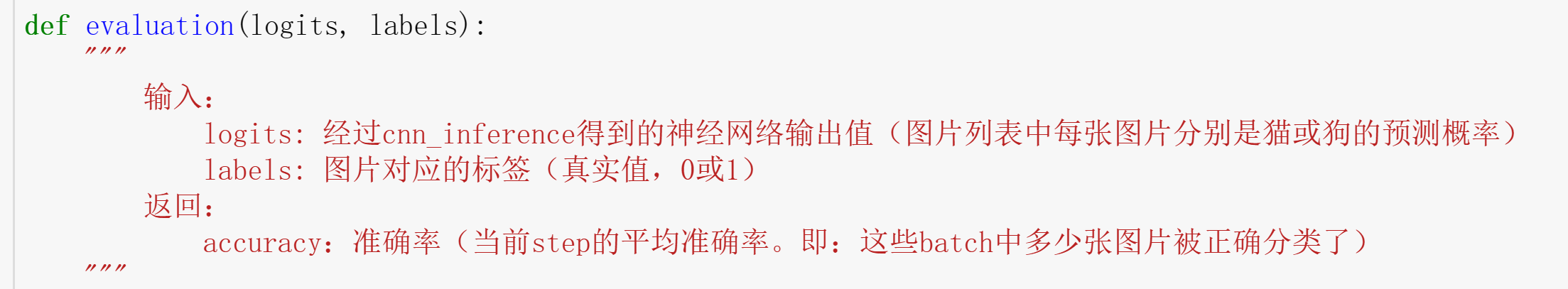
losses方法：传入cnn处理过的神经网络的输出值和对应的图片标签labels，最终返回损失之loss，进行归一化和交叉熵处理。



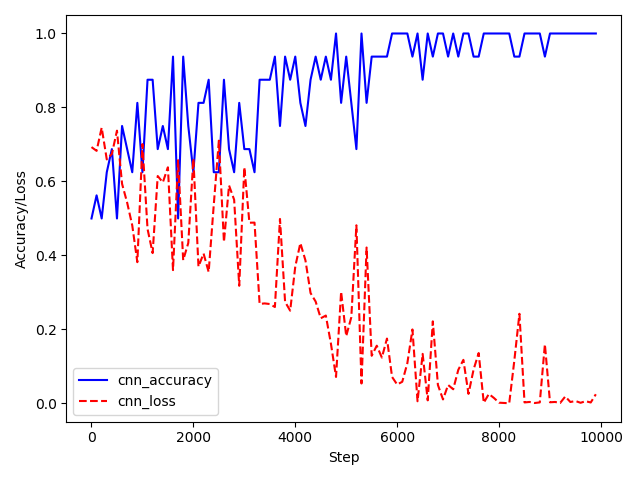
traning方法：传入loss损失值和学习率learing\_rate，使用了Adam优化器进行反向传播，使得loss损失值达到最小化。



evaluation方法：输入经过cnn\_inference方法得到的神经网络输出值和图片对应的标签labels，返回每一批训练数据的平均准确率

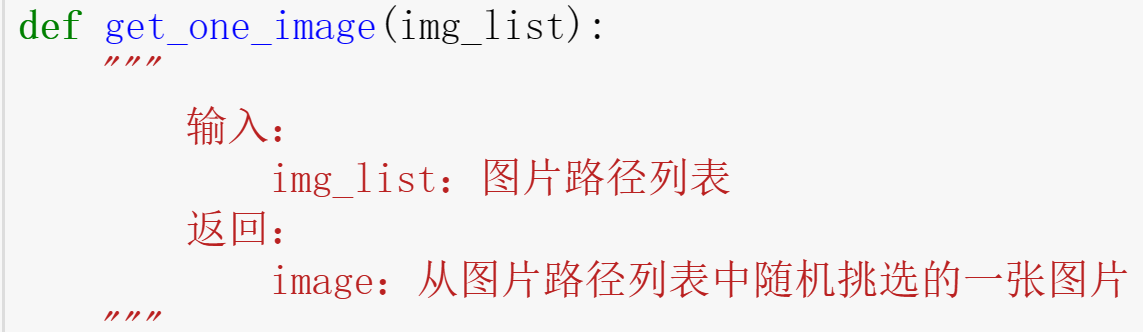


⑤ Training模块：定义全局变量并进行初始化，设置训练集的文件路径和模型被保存的路径，并调用对应模块的方法来得到相应的返回值，所有的操作都需要在tensorflow的session里进行，定义最大训练次数MAX\_STEP为10000次，当队列中所有的数据都被取出时，停止训练，每训练50次会在控制台输出一次准确率和损失值，并把训练的数据存入文件夹中，每训练100次，向列表中传入画图需要用到的准确率和损失值，每训练5000次保存好训练好的模型，训练结束后将训练图显示出来，如图所示，红色虚线表示的时损失值，蓝色曲线表示的是准确率，两者和为1.

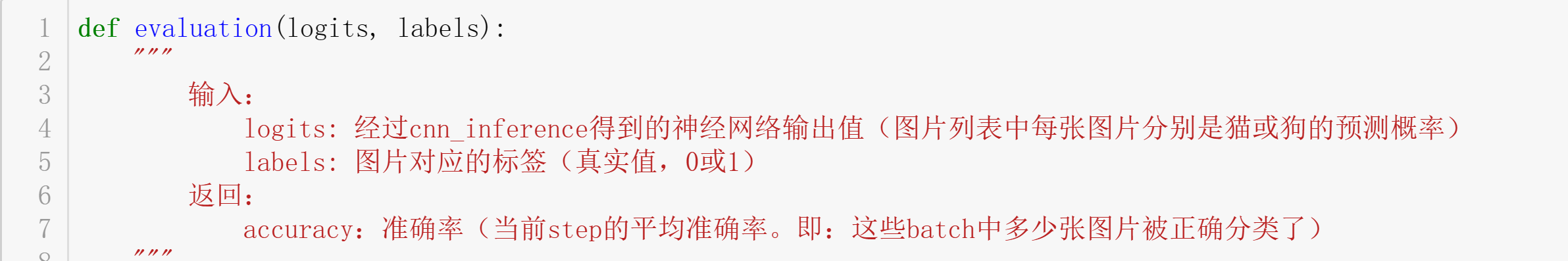


Test模块对图片进行预测

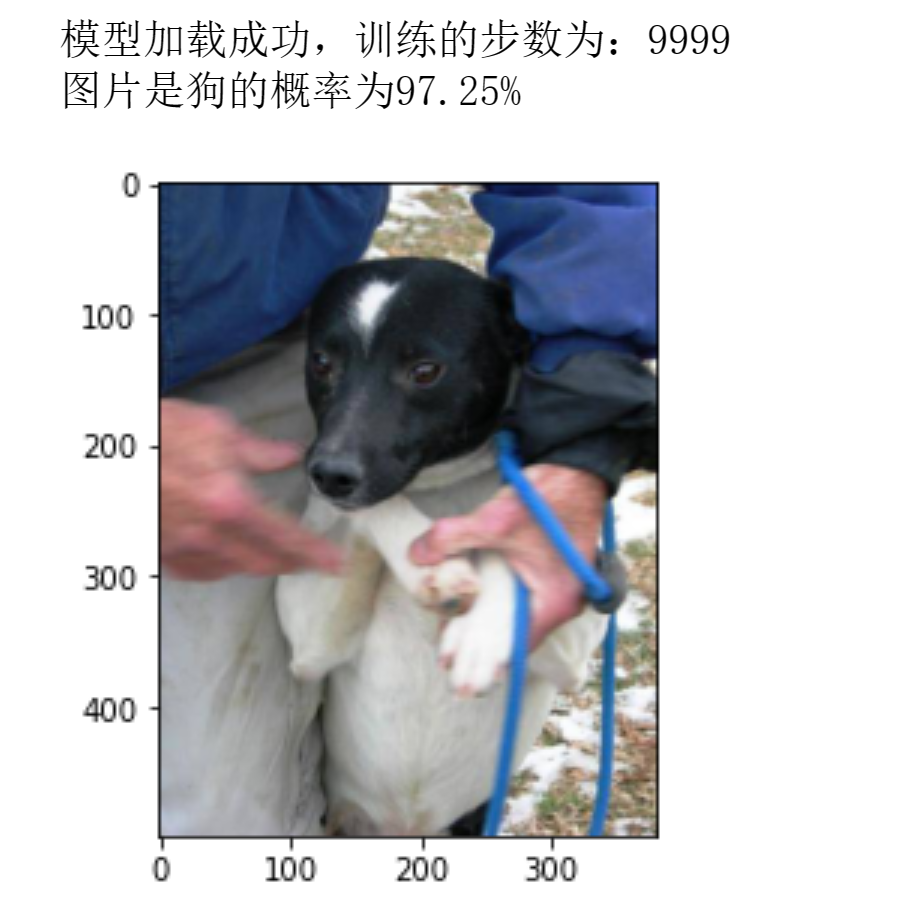
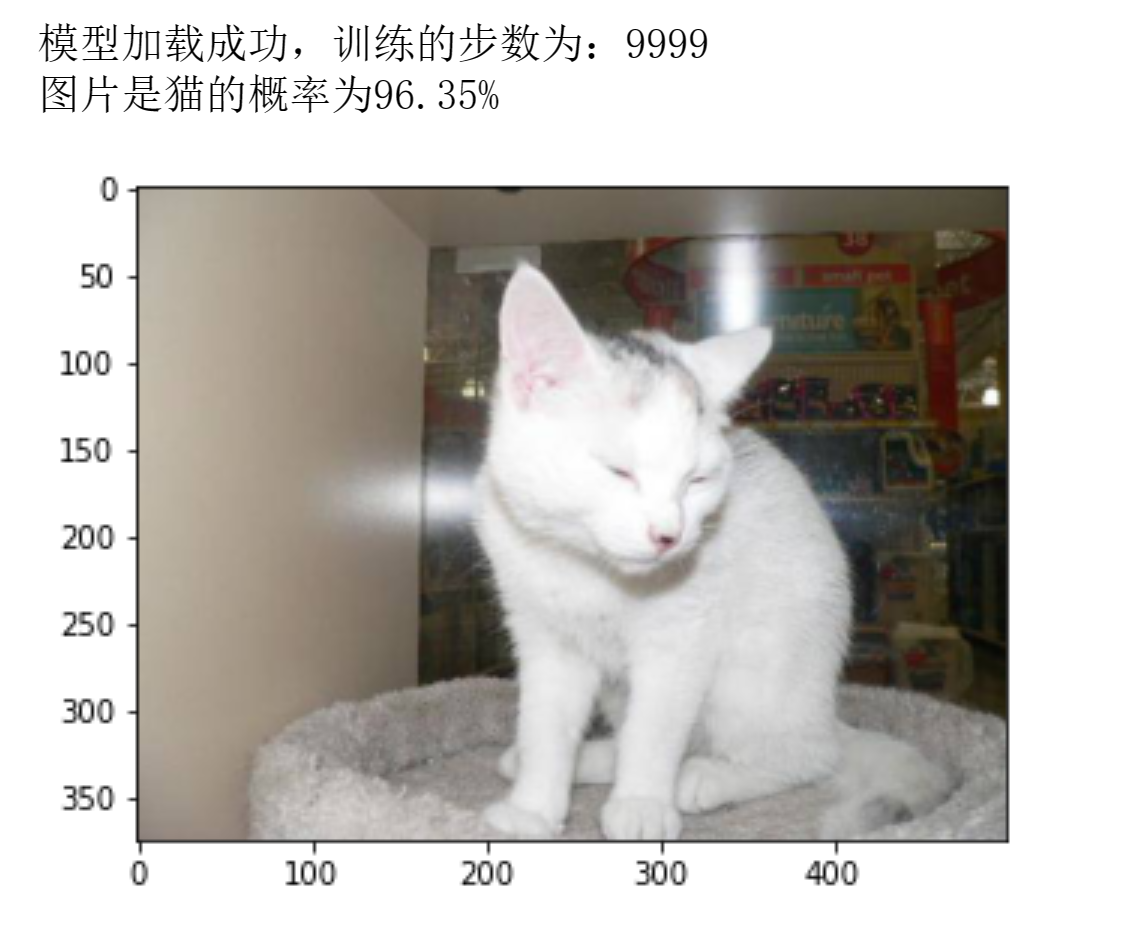
get\_one\_image方法：传入图片路径的列表，并随机挑选一张图片进行返回。



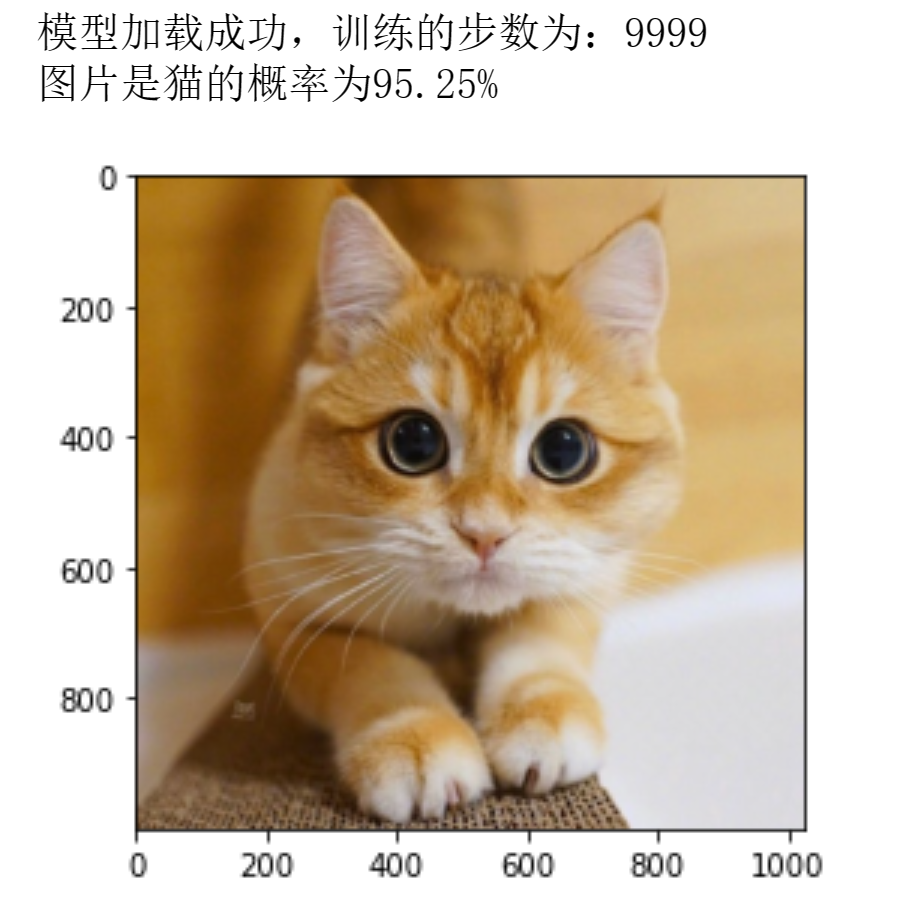
evaluate方法会加载训练好的模型，并对传入的图片进行预测，得到是猫还是狗。



### 五、实验结果

准确率还是比较高的，接下来选取百度上的一张猫的图片，进行识别，识别准确了达到了95%还是比较高的，该图片选自知乎专栏，读者可自行考证：https://zhuanlan.zhihu.com/p/146833924



### 六、实验结论

该CNN+Tensorflow的方法在接受25000张图片的训练后达到了不错的性能，至少在猫狗分类上的性能是喜人的，但是如何把该模型进行修改，使其能够在其他分任务上有一个好的性能，是一个值得继续深入工作的事情。

### 七、实验收获

整个实验过程可以说是磕磕绊绊，最让我印象深刻的是是在Tensorflow的安装上，在1.x版本和2.版本的两种选择中，如果选错会导致一些引用失败，如报错：No module named 'tensorflow.combat',这个错误是因为Tensorflow版本问题导致的，然而我一直在研究是不是因为自己没有实际安装上，因此反复卸载安装了好几遍，后来以为是ANACONDA虚拟环境配置有问题，结果忙活了一大顿也没解决，最后才发现是方向错了，通过CSDN和知乎的一些回答得知，过新的版本会导致引用tensorflow的时候无法识别，才知道需要修改引用代码才能成功引入tensorflow，这种不是程序逻辑上的问题有时候更让人头疼，且不好解决。

最后，很幸运能选修肖老师的自然语言处理，让我能入门深度学习和自然语言处理，对这个广袤无垠的研究方向能有一些认识和了解。

### 八、参考

[1]<https://blog.csdn.net/nanhuaibeian/article/details/101862790>

[2]<https://blog.csdn.net/Elenstone/article/details/105123255>

[3]<https://zhuanlan.zhihu.com/p/67206089>

[4]<https://link.csdn.net/?target=https%3A%2F%2Fwww.zybuluo.com%2Fhanbingtao%2Fnote%2F485480>