**卷积神经网络算法识别猫狗图片**

**一、收集数据集**

对收集到的数据集的来源进行说明,并对数据集进行描述。

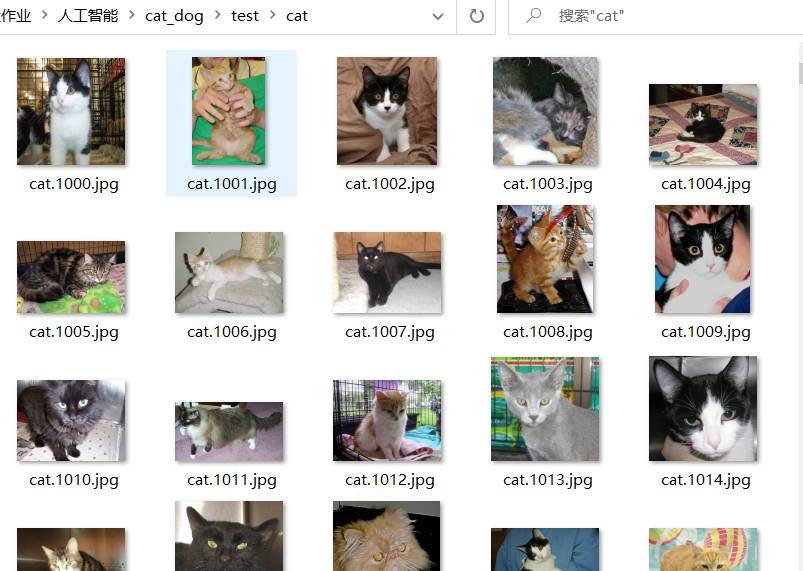
描述如:数据集一共有多少张图片,训练集、测试集的划分比例,图片类别等。

# 数据集在为kaggle比赛猫狗数据

链接为<https://gas.graviti.cn/open-data-set-detail>

其中一共有25000张图片，其中猫有12500，狗有12500，测试集有12500张图，图片类别为jpg格式。训练集、测试集的划分比例为1:1

此处发的数据集为处理过的数据，即有训练集，测试集和验证集，分别有猫狗图片各1000张。总为6000张猫狗图片。部分类似于如图



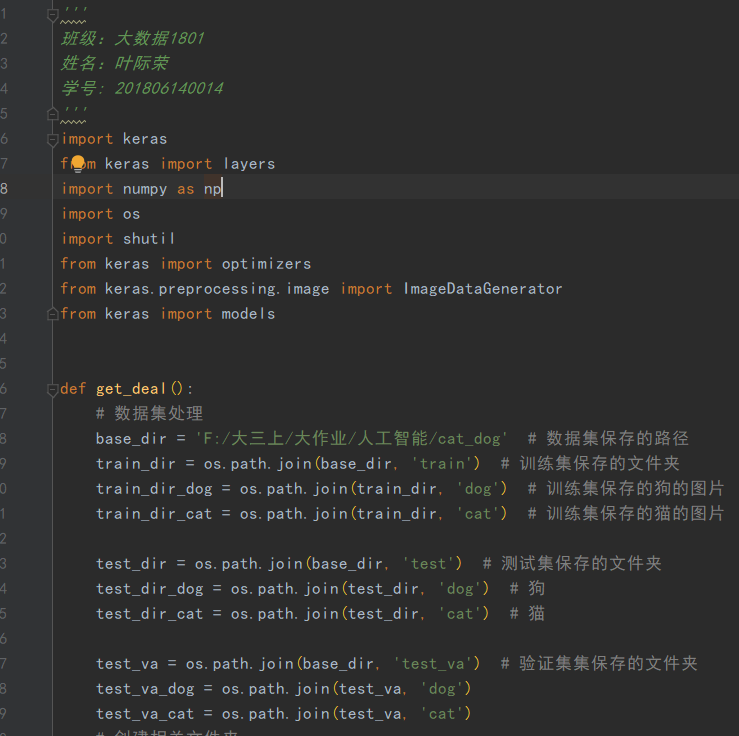
**二、数据集预处理**

**1、数据集预处理过程的实现思路与步骤**

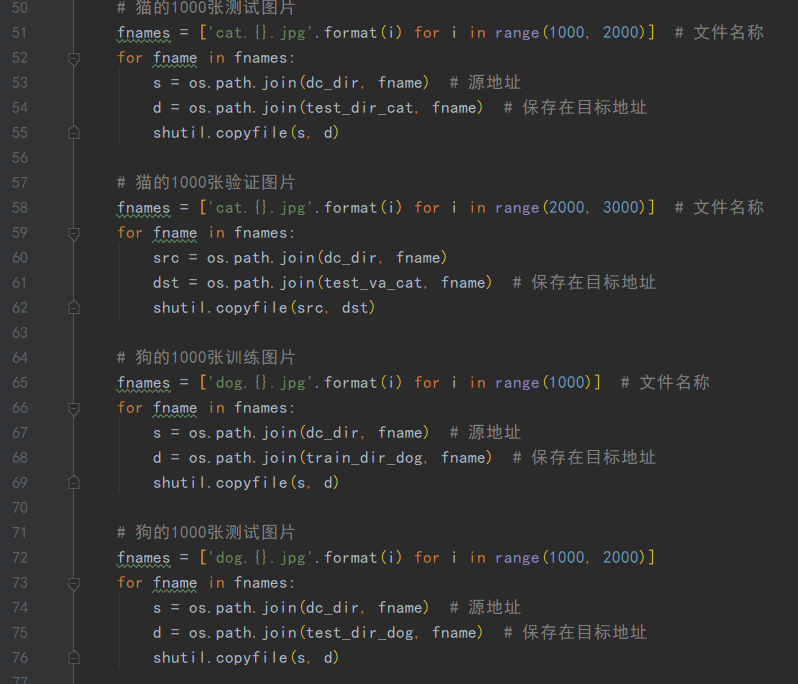
由于下载的数据集太大，故此先提取部分图片作为训练集和测试集。将图像复制到训练、验证和测试的目录。然后将图片归一化，使图片可以抵抗几何变换的攻击，它能够找出图像中的那些不变量，从而得知这些图像原本就是一样的或者一个系列的，同时加快训练网络的收敛性。

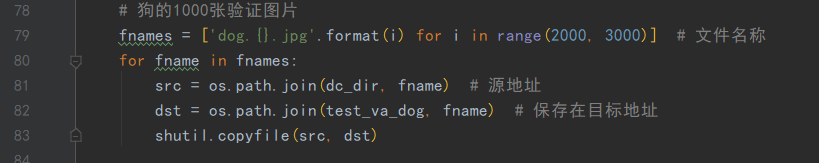
同时还需要对图像数据增强。因为处理图片时会丢失许多数据。所以需要对数据增强处理。

**2、数据集预处理过程的源代码、注释说明**





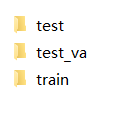




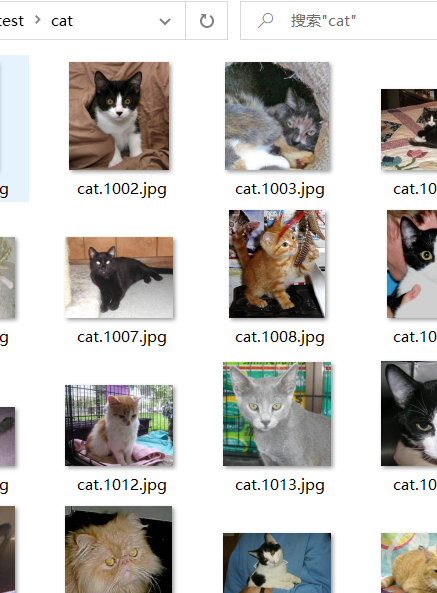
用代码对原数据集进行分类创建文件夹等

1. **预处理过程的运行结果及简要说明**

相关的文件夹



部分结果：



这里是把训练集和测试集和验证集创建好了并划分好了。

**三、模型的构建、训练、预测与评估**

**1、模型的构建、训练、预测与评估过程的实现思路与步骤**

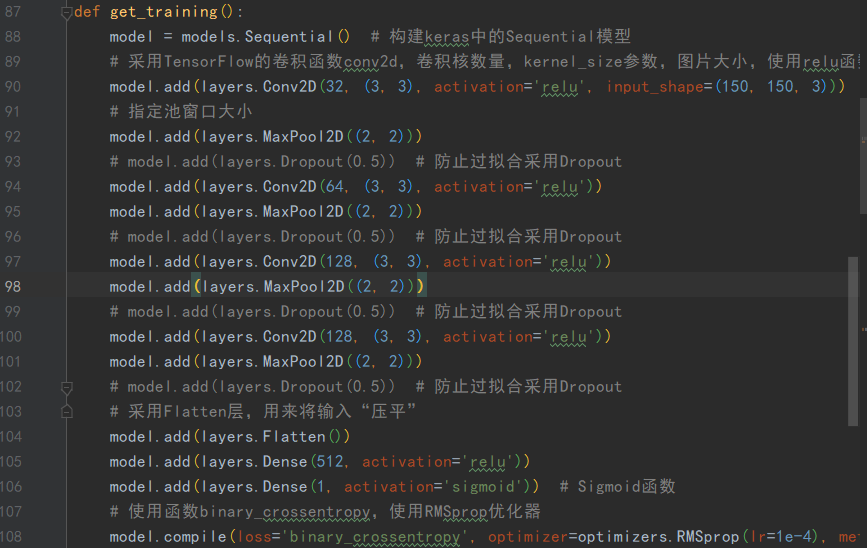
# 搭建keras中的sequential模型，设置好卷积层，最大池化层，全连接层等。将划分好的训练集测试集等进行处理即将在执行其他处理前乘到整个图像上并将这个值定为0~1之间的数。自己构建相关数据集。定好图片大小，使用的模型等。定好后训练模型。设置好steps\_per\_epoch：生成器返回steps\_per\_epoch次数据时计一个epoch结束，执行下一个epoch，epoch：数据迭代的轮数，validation\_data：验证集的生成器，validation\_steps：当validation\_data为生成器时，本参数指定验证集的生成器返回次数。

当进行模型训练的时候，往往可能错过模型的最佳临界点，即当达到最大精度的时候再进行训练，测试集的精度会下降，这时候就会出现过拟合。为了防止或减少过拟合，第二个模型采用Dropout。它的大致意思是在训练时，将神经网络某一层的单元（不包括输出层的单元）数据随机丢失一部分，实际上是等于创造出了很多新的随机样本，以增大样本量，减少特征量的方法来防止过拟合

同时采用vgg16网络进行使用，VGG16网络为13层卷积层+3层全连接层而组成，VGG16的突出特点是简单，卷积层均采用相同的卷积核参数，池化层均采用相同的池化核参数。

**2、模型的构建、训练、预测与评估过程的源代码、注释说明**

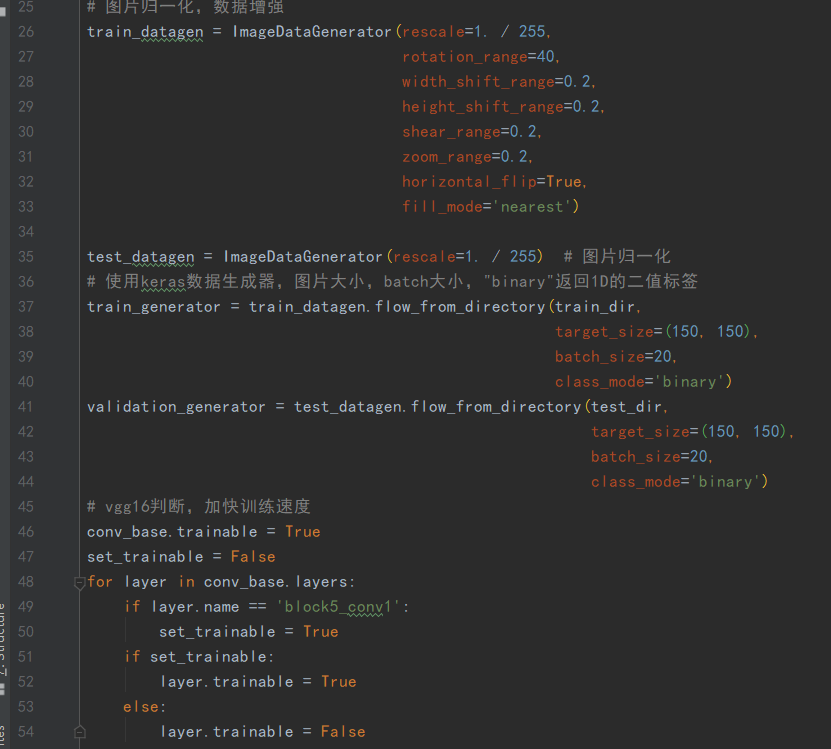
（1）第一个模型



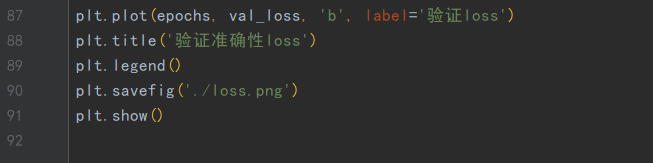


1. 第二个模型



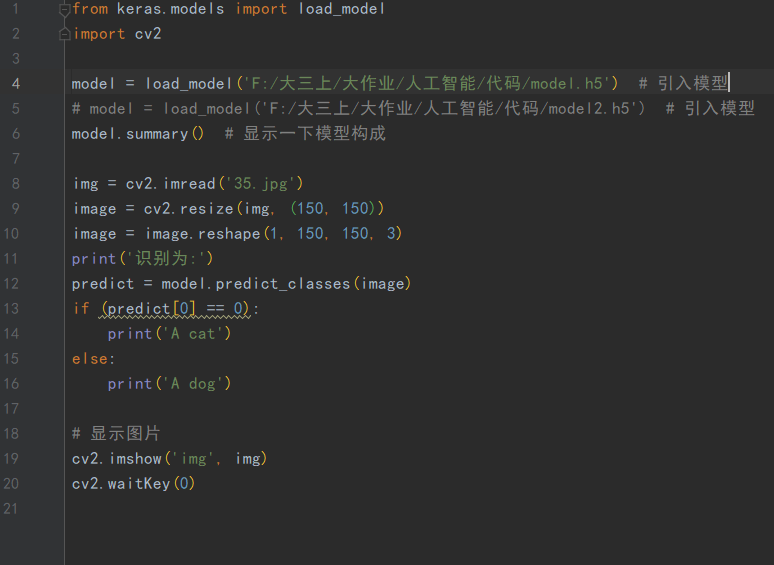






第一个模型只有图片归一化，第二个模型还有图片的增强等

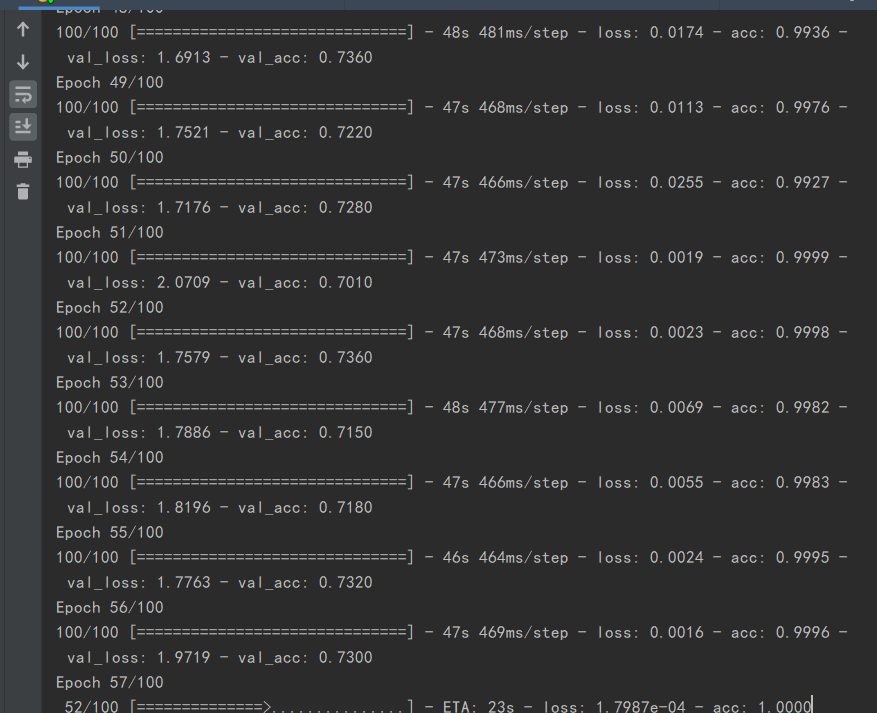
验证模型判断的代码：

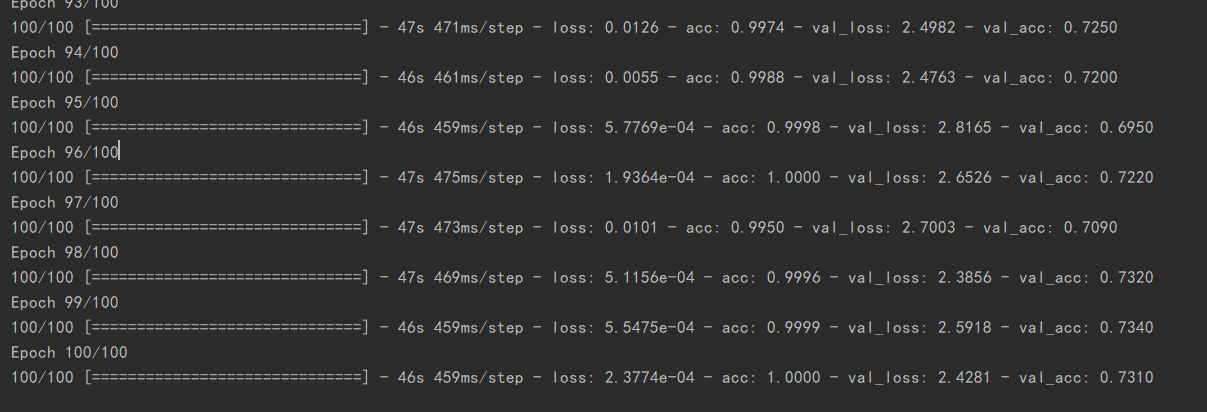


导入训练好的模型。对图片进行处理如大小等，进行判断。

**3、模型的构建、训练、预测与评估过程的运行结果及简要说明**

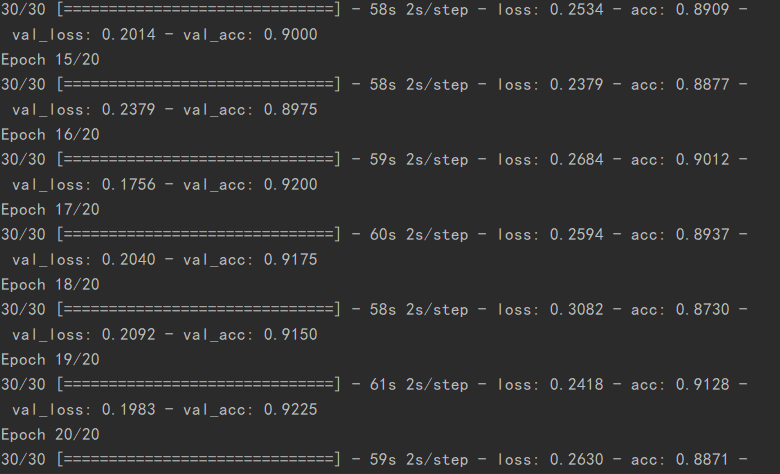
（1）第一个模型

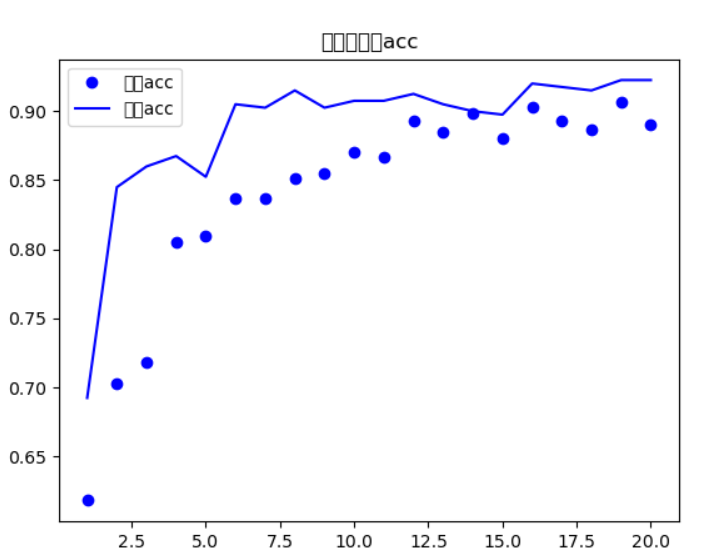


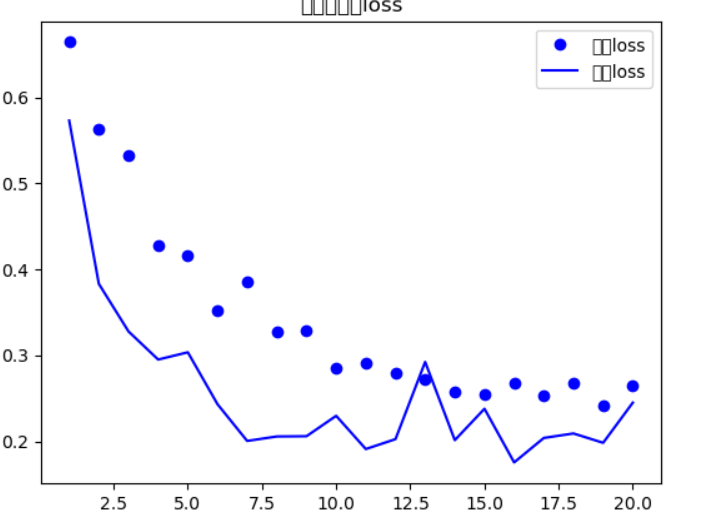


loss是训练集上的损失、val\_loss是测试集上的损失，最终模型的效果val\_acc。发现模型效果不是特别好。

（2）第二个模型



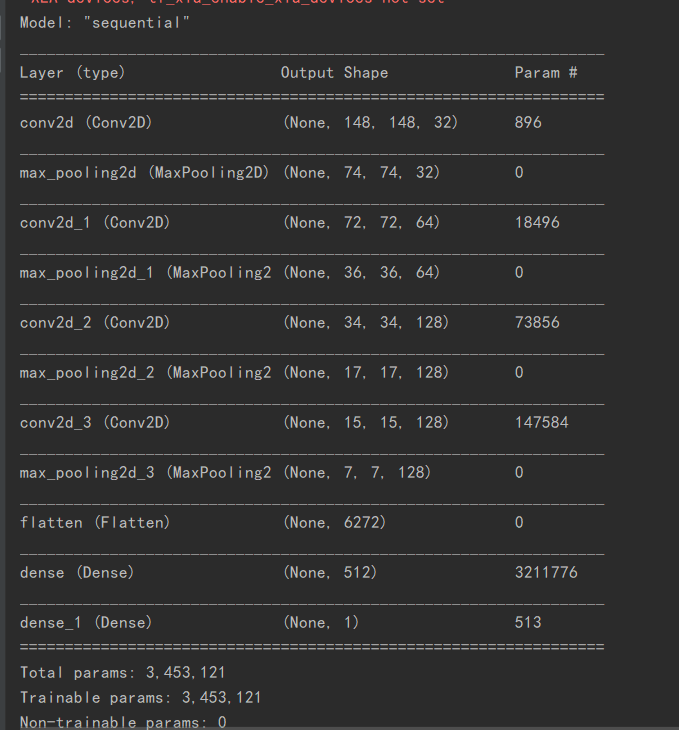


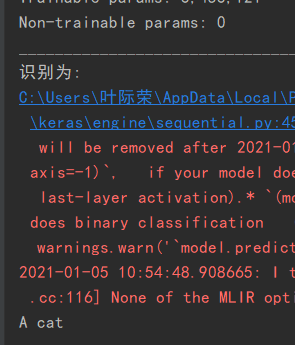


通过val\_acc发现模型效果还可以，图片发现震荡逐渐逐渐减少。

验证模型的结果：（随机设置的猫狗图片）

第一个模型：

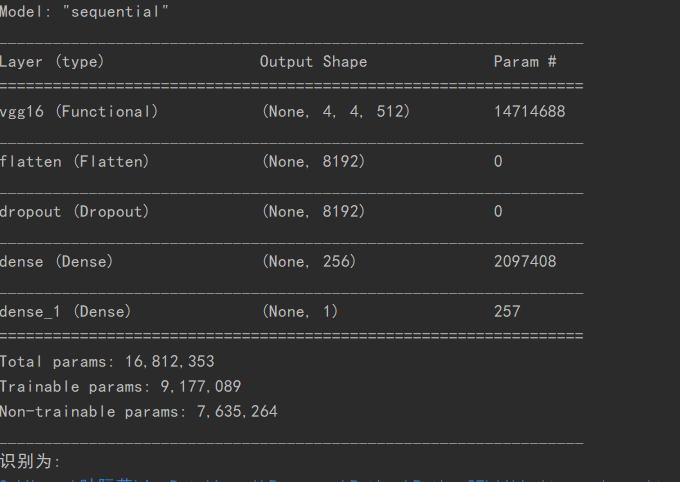


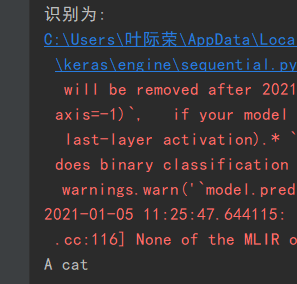


先是打印出相关模型的结构

结果会显示是猫或者是狗

第二个模型：





**四、思考与总结**

本次使用的是卷积神经网络，卷积神经网络有3个基本层，卷积层（卷积核），池化层，全连接层。

卷积过程是使用一个卷积核（如图中的Filter），在每层像素矩阵上不断按步长扫描下去，每次扫到的数值会和卷积核中对应位置的数进行相乘，然后相加求和，得到的值将会生成一个新的矩阵。卷积核相当于卷积操作中的一个过滤器，用于提取我们图像的特征，特征提取完后会得到一个特征图。池化操作相当于降维操作，有最大池化和平均池化，本次用的是最大池化。第二各模型中VGG16网络为13层卷积层+3层全连接层而组成。Conv2D卷积层即给图像矩阵四周都加上0。VGG16的特点是简单，第二各模型使用的即是vgg16

卷积神经网络的层级结构有数据输入层，卷积计算层，激励层，池化层，全连接层。

数据输入层：主要是对原始图像数据进行预处理，如归一化

卷积计算层：这一层就是卷积神经网络最重要的一个层次，计算矩阵

激励层：把卷积层输出结果做非线性映射

池化层：池化层的最主要作用就是压缩图像。

全连接层：两层之间所有神经元都有权重连接，通常全连接层在卷积神经网络尾部。

开始的时候找数据集比较麻烦，因为不知道能不能使用，然后对数据集的处理不太会，导致在这部分困扰很久。比如归一化，灰度等，因为划分训练集和测试集时出错了。不过后面还好ok了，使用卷积神经网络时代码开始不太会编写，通过百度等相关方法理解了并知道相关使用方法。使用VGG16方法时发现比较简单，所以做了第二个模型。VGG16具有很高的拟合能力；但同时缺点也很明显，训练时间过长，调参难度大。

先要理解卷积神经网络，理解后编写代码就简单多了，就不会说有些参数怎么设置，怎么用法不知道了。然后再对相关图片进行处理。但训练过程使用cpu实在是太慢了，所以想搞个gpu，现在还没有成功弄好gpu。