**实验五 猫狗分类 作业报告**

1. 实验目的
2. 进一步理解和掌握卷积神经网络中卷积层、卷积步长、卷积核、池化层、池化核、微调(Fine-tune)等概念。
3. 对已经在ImageNet数据集上训练好的模型进行微调以实现猫狗分类任务。
4. 实验要求

基于Python语言和TensorFlow框架，利用Google公司发布的Tensorflow Slim工具包和迁移学习（Transfer learning）方法，对已经在ImageNet数据集上训练好的模型进行微调，实现一个猫狗分类神经网络模型的训练、测试和导出。

1. 实验原理

迁移学习(Transfer learning) 顾名思义就是把已训练好的模型参数迁移到新的模型来帮助新模型训练。考虑到大部分数据或任务都是存在相关性的，所以通过迁移学习我们可以将已经学到的模型参数（也可理解为模型学到的知识）通过某种方式来分享给新模型从而加快并优化模型的学习效率，而不用像大多数网络那样从零学习。

迁移学习有多种具体方式，如Transfer Learning、Extract Feature Vector、Fine-tune等，本实验指导书中用到的方法为Fine-tune。

Fine-tune具体是指冻结预训练模型的部分卷积层（通常是靠近输入的多数卷积层），训练剩下的卷积层（通常是靠近输出的部分卷积层）和全连接层。

本质上来讲：这三种迁移学习的方式都是为了让预训练模型能够胜任新数据集的识别工作，能够让预训练模型原本的特征提取能力得到充分的释放和利用。但是，在此基础上如果想让模型能够达到更低的Loss，那么光靠迁移学习是不够的，靠的更多的还是模型的结构以及新数据集的丰富程度。

Inception为Google开源的CNN模型，至今已经公开四个版本，每一个版本都是基于大型图像数据库ImageNet中的数据训练而成。因此我们可以直接利用Google的Inception模型来实现图像分类。Inception的核心就是把GoogleNet的某一些大的卷积层换成1\*1, 3\*3, 5\*5的小卷积，这样能够大大的减小权值参数数量。

本实验指导书主要以Inception\_v3模型（https://arxiv.org/abs/1512.00567）为参考结构。Inception网络以Inception Module为基础。

Inception Module的作用就是代替人工来确定卷积层中的过滤器类型，或者确定是否需要创建卷积层或是池化层。我们可以应用各种类型的过滤器，只需要把输出连接起来。之后通过使用 1×1 卷积来降低计算成本。

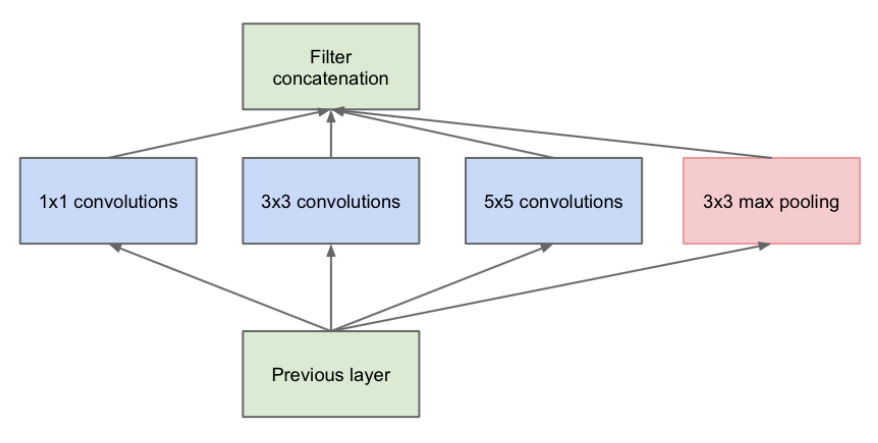


图1 Inception Module（V1版本）

Inception V3将n\*n卷积层分解成1\*n卷积层和n\*1卷积层，这样既可以加速计算(多余的计算能力可以用来加深网络)，同时又可以将一个卷积层拆成两个卷积层，从而加深网络，增加了网络的非线性，进一步提升了模型的性能。

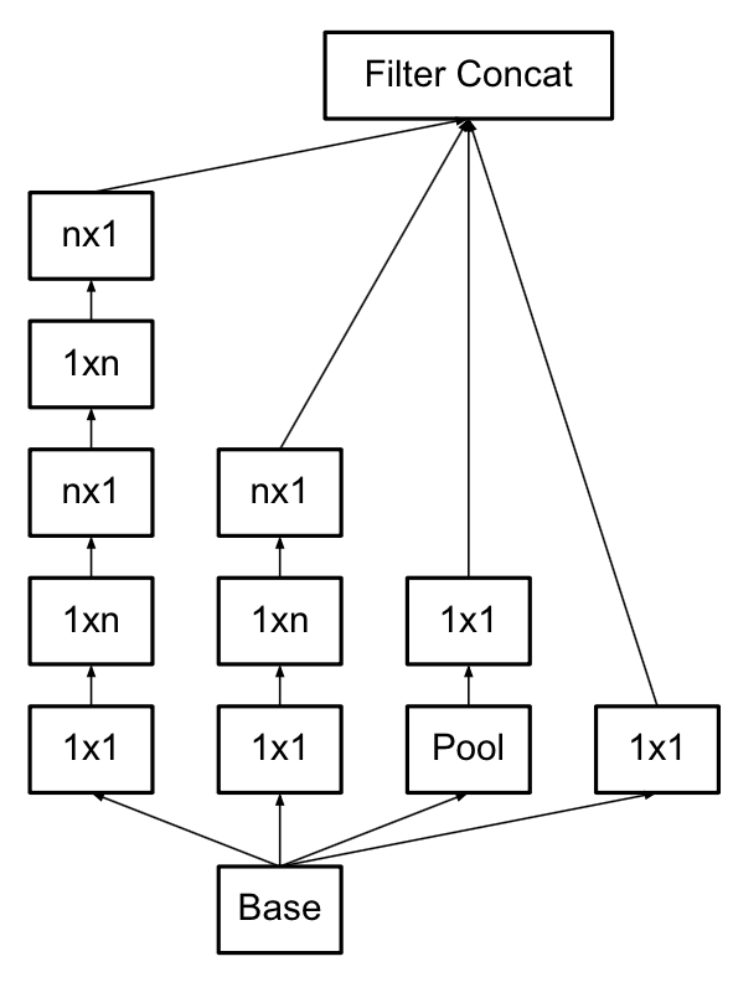


图2 Inception Module（V3版本）

1. 实验所用数据集及工具
2. 数据集

本实验使用实验数据基于kaggle Dogs vs. Cats 竞赛提供的官方数据集，数据集可在百度网盘中进行下载：

链接：https://pan.baidu.com/s/13hw4LK8ihR6-6-8mpjLKDA 密码：dmp4。

数据集的目录划分如下：

dataset

|— train

|— dogs

|— cats

|— validation

|— dogs

|— cats

将数据集划分为训练集（training dataset）和验证集（validation dataset），均包含dogs和cats两个目录，且每个目录下包含与目录名类别相同的RGB图。数据集共25000张照片，其中训练集猫狗照片各10000张，验证集猫狗照片各2500张。（注：可根据计算资源情况自己调整训练集和验证集的大小，但最好按比例调整）

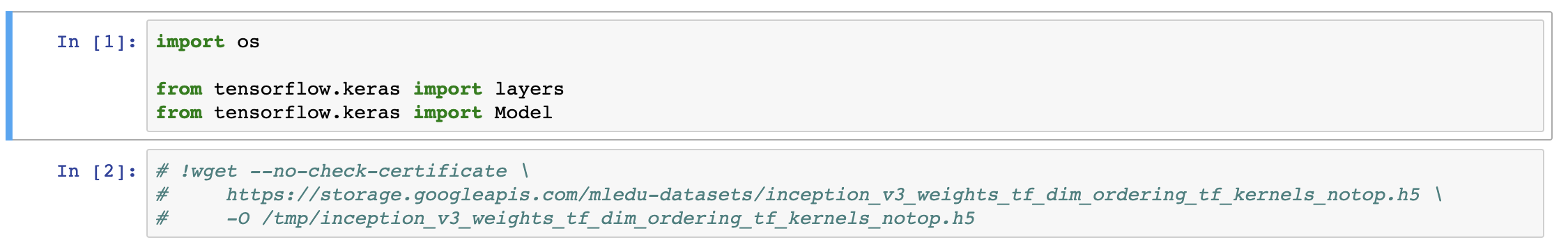
原始数据集如图3、4、5、6所示。

|  |  |
| --- | --- |
|  | C:\Users\63039\AppData\Local\Microsoft\Windows\INetCache\Content.Word\dog.9496.jpg |
| 图3 | 图4 |

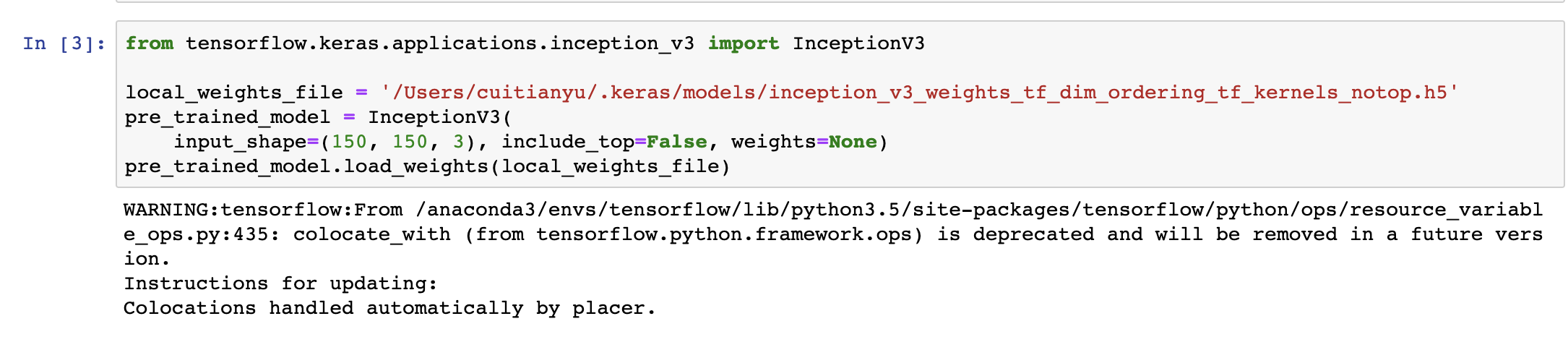
|  |  |
| --- | --- |
|  | C:\Users\63039\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cat.7013.jpg |
| 图5 | 图6 |

1. 实验过程

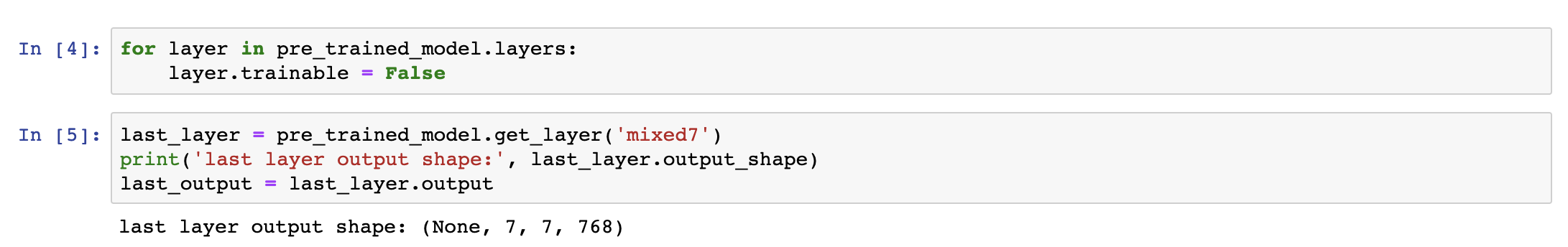
本次实验使用Tensorflow的高级封装Keras实现，获取预训练的Inception模型



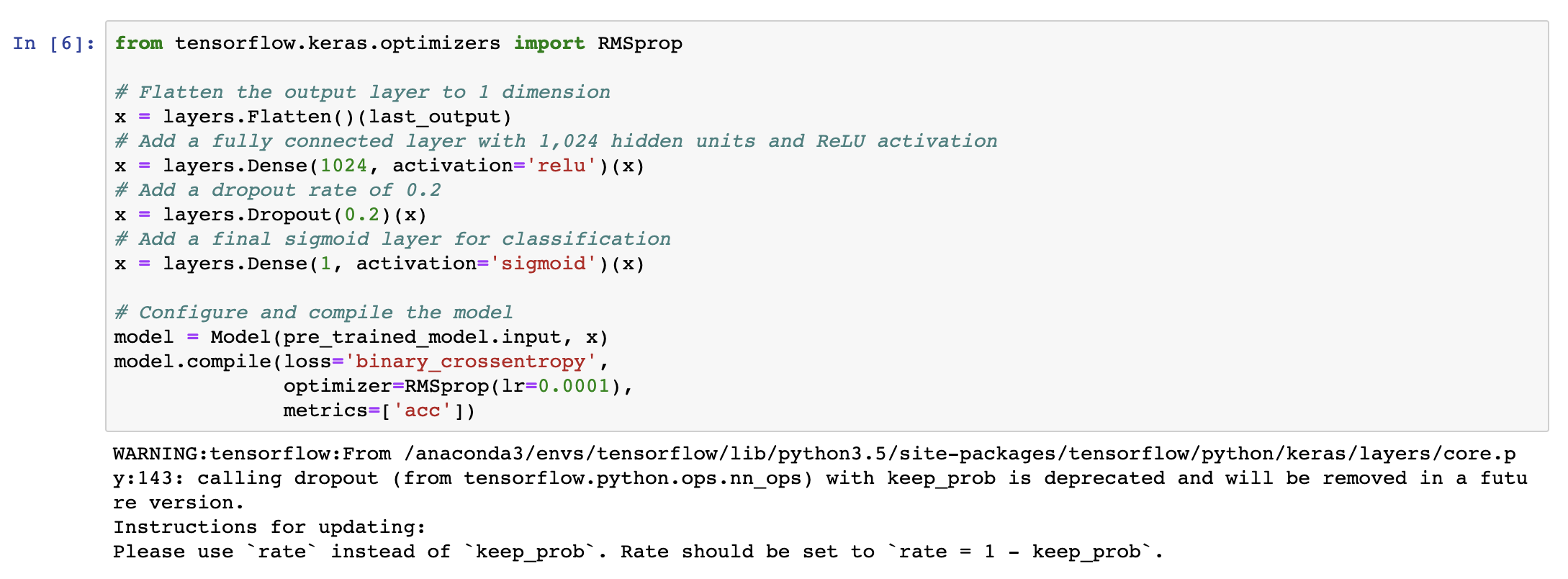
构建Inception模型并读取预训练模型参数。



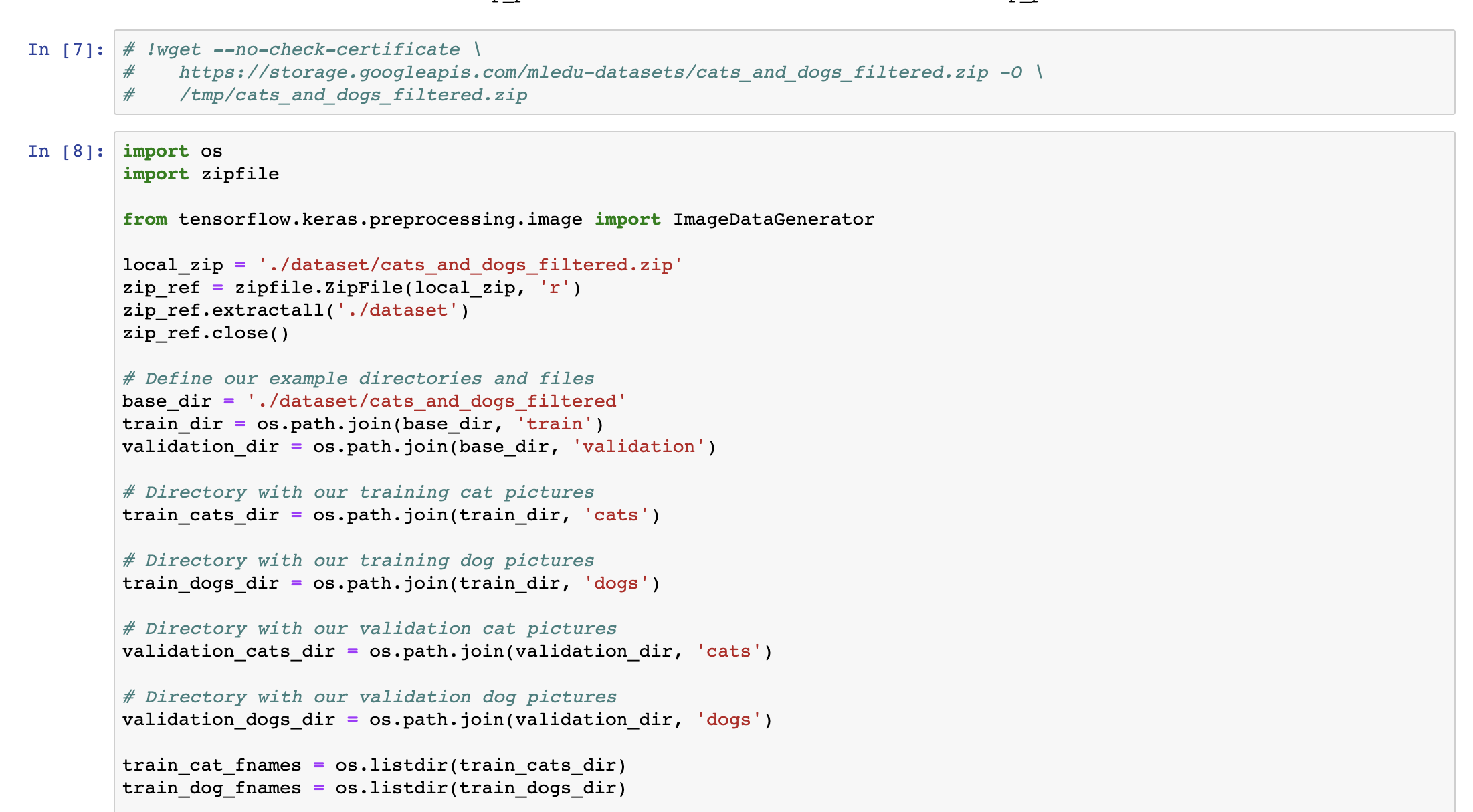
设置模型参数为不可训练，冻结模型



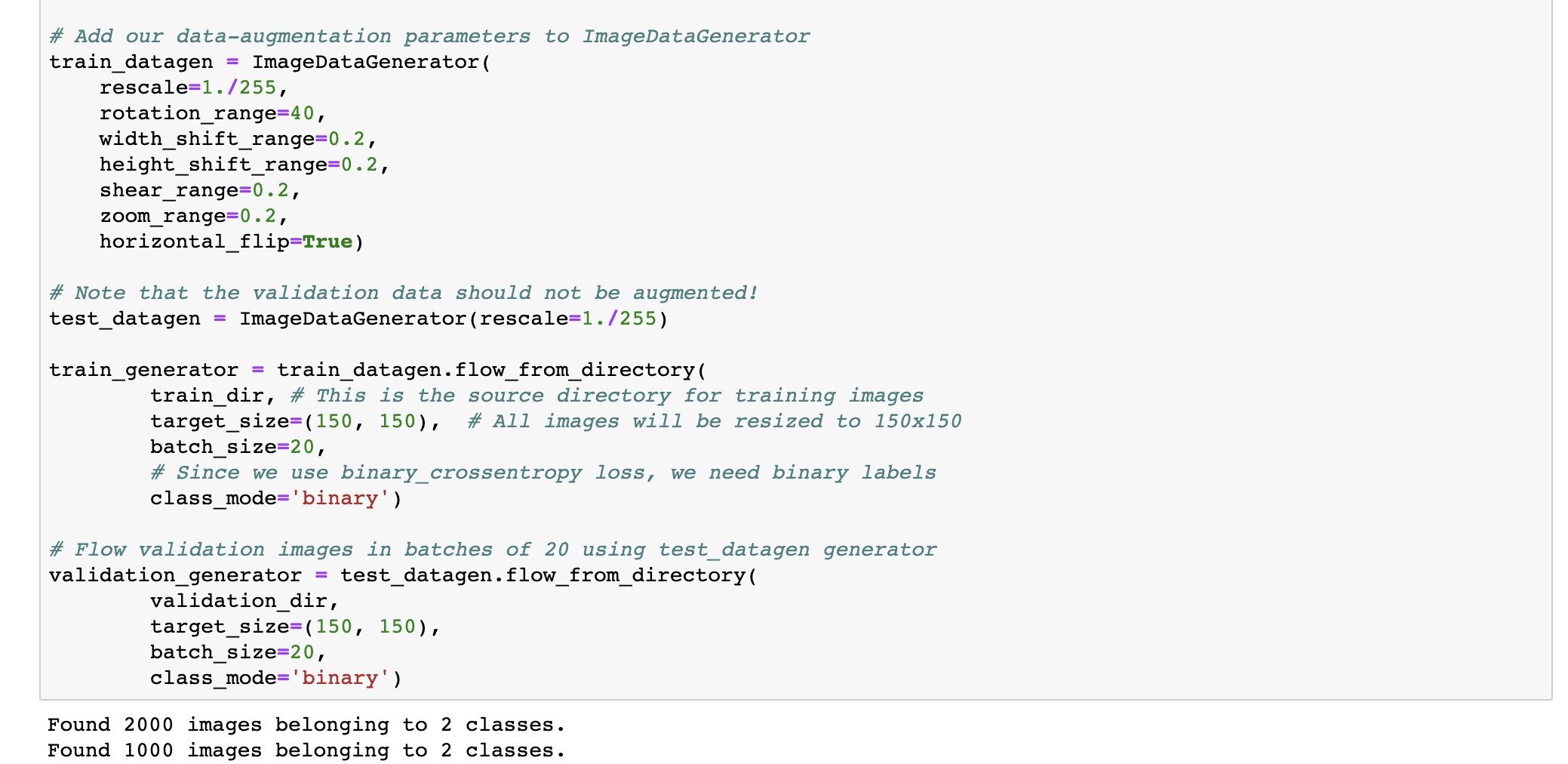
添加Flatten层，全联接层和Dropout层，定义优化器，损失函数和评估方法



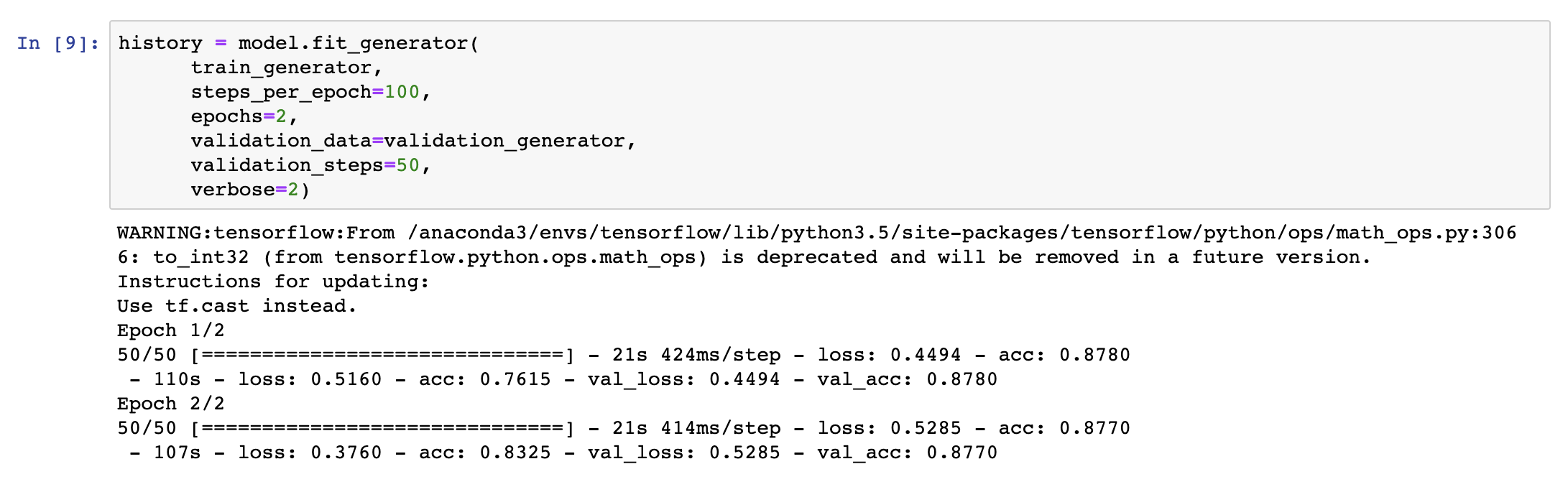
获取数据集



定义图片输入函数generator



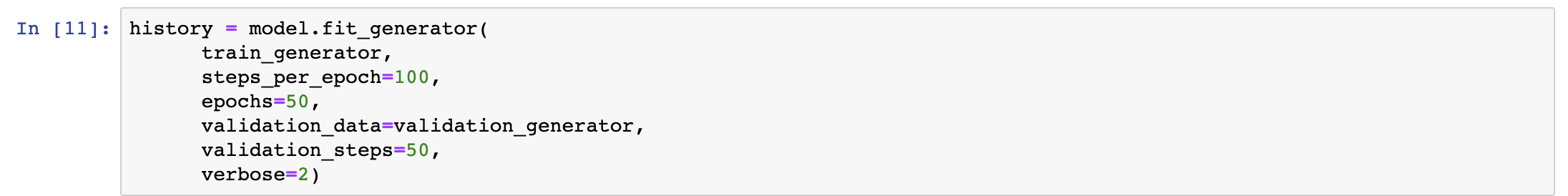
开始训练，得到训练和验证结果



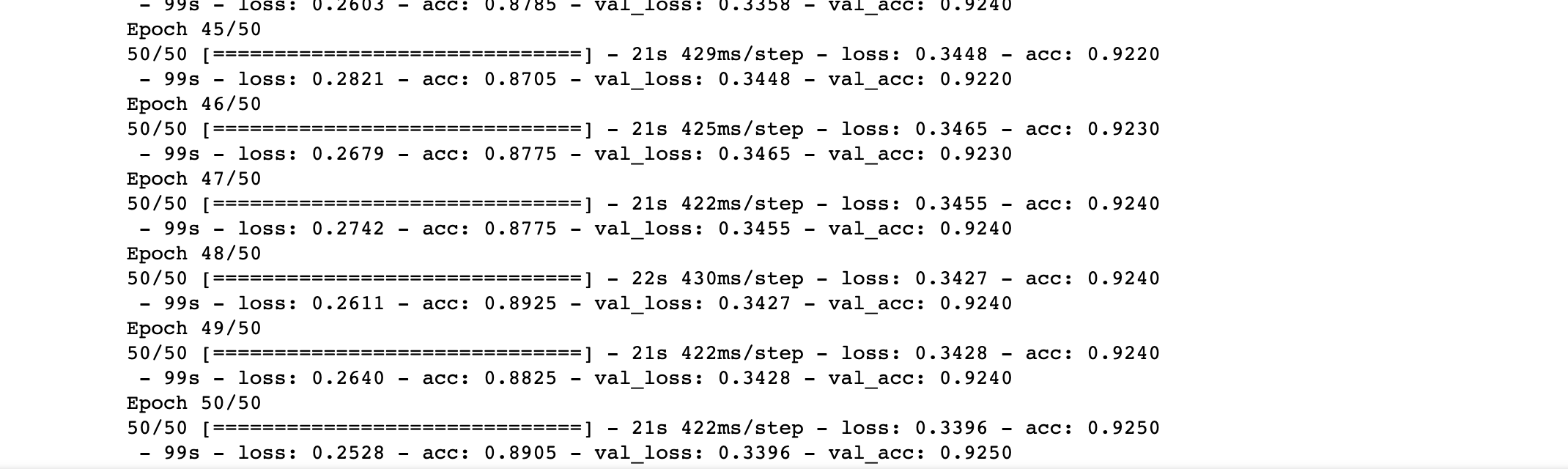
模型微调，解冻Inception模型其中一层



再次训练



结果展示



绘制loss和acc变化曲线



结果展示

