基于卷积神经网络CNN的猫狗图像分类

1. 设计思路

（实验环境为windows10操作系统，英特尔十代i7CPU，英特尔核心显卡，python3.6，TensorFlow2.0）

卷积神经网络最常见的用途就是进行图像识别/分类任务，此处需要完成对于猫狗图像的二分类任务。考虑仿照LeNet结构构建卷积神经网络。

主要的实现步骤包括：使用keras库中的ImageDataGenerator模块进行图片数据的预处理和构成训练生成器；使用kerastuner库以小epochs的方式对模型超参数进行自动优化搜索；根据最优参数构建模型并增大epochs轮数完成模型训练。

1. 模型结构

|  |
| --- |
| 1: INPUT()  2: Conv2d(16卷积核，尺寸为3\*3，relu激活函数，输入尺寸150\*150像素，RGB三通道)  3: MaxPool2d(尺寸2\*2)  4: Conv2d(16卷积核，尺寸为3\*3，relu激活函数)  5: MaxPool2d(尺寸2\*2)  6: Conv2d(64卷积核，尺寸为3\*3，relu激活函数)  7: MaxPool2d(尺寸2\*2)  8: Conv2d(16卷积核，尺寸为3\*3，relu激活函数)  9: MaxPool2d(尺寸2\*2)  10: Flatten()  11: Dropout(0.2)  12: Dense(352个神经元，relu激活函数)  13: Dense(1个神经元，sigmoid激活函数)  14: OUTPUT() |

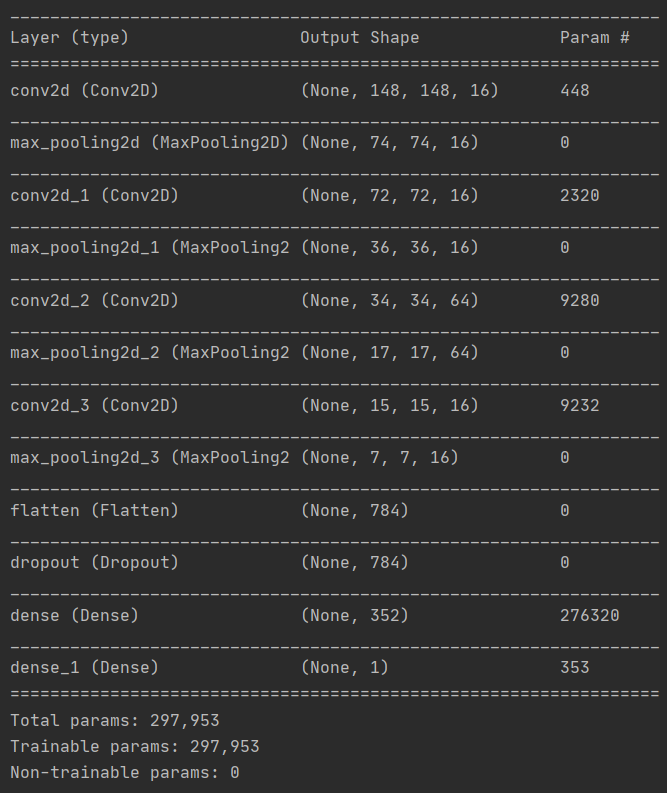


图 1模型的summary

1. 实验结果

使用上述模型结构，损失函数使用二元交叉熵，优化函数使用RMSprop，学习率设置为0.0005，mini\_batch设置为100。使用如上配置进行150轮训练之后得到如图2所示的结果。可以看出训练集和验证集上的准确率随着训练轮数的增加进行波动上升，当训练150轮后，验证集上的准确率基本稳定在0.8。

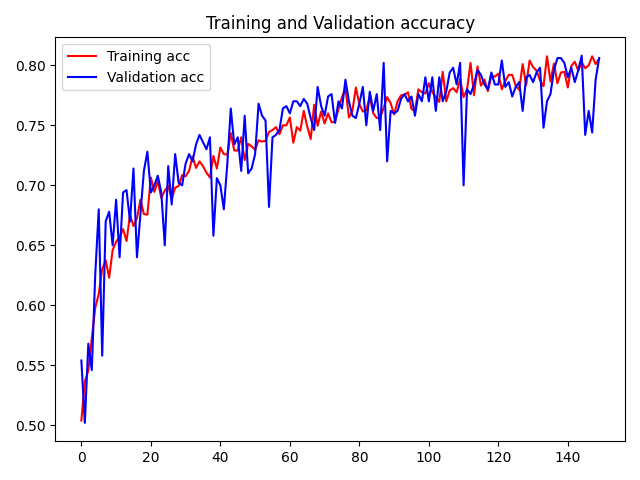


图 2 训练150轮训练集和验证集上的准确率

1. 总结与思考

刚开始做实验的时候遇到了问题—训练集准确率高达90%但验证集准确率只有不到70%。通过观察模型结构与查阅资料后发现可能遇到了过拟合问题。解决办法有：1.减小学习率；2.增加正则或者Dropout；3.在训练集上进行数据增强。经过上述的调整，最终解决了遇到的问题。

基于深度神经网络DNN的猫狗图像分类

1. 设计思路

深度神经网络换句话说就是全连接神经网络，由输入直接全连接到一层或多层神经元，最后从输出层输出数据。显然深度神经网络并不适用于对于图像的分类问题----样本输入是巨大的，用同样与卷积神经网络模型的输入的话那么就是150\*150\*3=67500个参数数值，这么多参数输入进下一层，如果第一个隐含层神经元数目较少的话则完全不能学到样本的数据特征，如果第一个隐含层神经元数目较多的话则直接会导致权重参数的爆炸增长。因此考虑将RGB三色三通道的图片转化为灰度图再输入全连接神经网络。使用多层全连接层来学习样本的特征。

1. 模型结构

|  |
| --- |
| 1: INPUT()  2: Flatten()  3: Dense(512个神经元，relu激活函数)  4: Dense(128个神经元，relu激活函数)  5: Dense(64个神经元，relu激活函数)  6: Dense(1个神经元，sigmoid激活函数)  7: OUTPUT() |

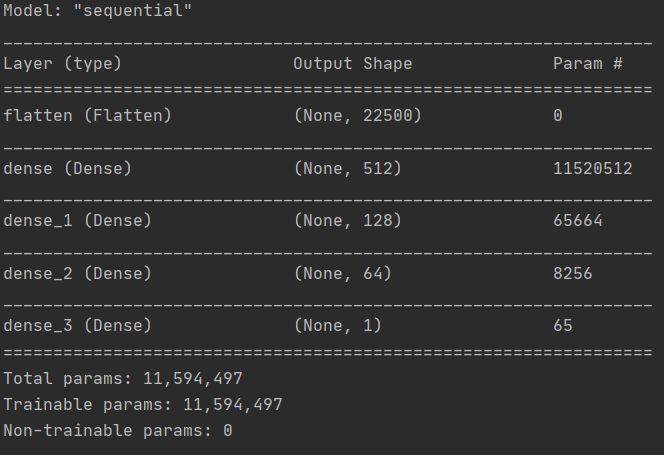


图 3模型的summary

1. 实验结果

使用上述模型结构，损失函数使用二元交叉熵，优化函数使用RMSprop，学习率设置为0.0001，mini\_batch设置为100。使用如上配置进行200轮训练之后得到如图4所示的结果。可以看出随着训练轮数的增加，训练集和验证集上的准确率增长较慢且波动较大，推测可能是全连接神经网络模型自身的问题，并且当训练了150轮之后，训练集上的准确率仍在缓慢增长，但验证集上的准确率已经基本问题，考虑这个阶段可能已经出现了模型过拟合。最终验证集上模型准确率在0.6左右。

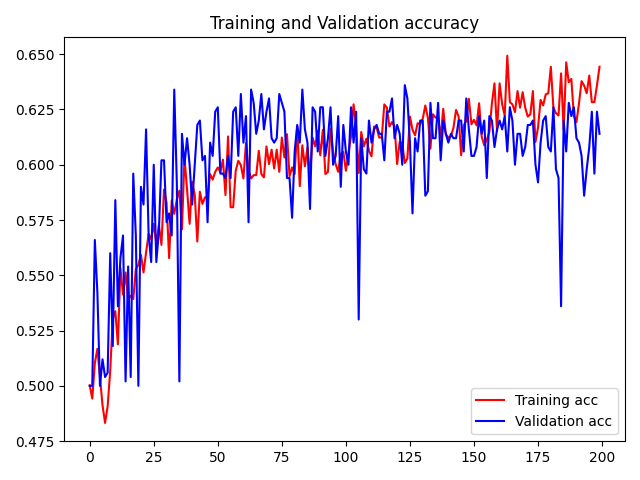


图 4训练200轮训练集和验证集上的准确率

1. 总结与思考

深度神经网络在图像识别任务上表现较差，这是因为该模型本来就不擅长进行这类任务。因此我们以后面对新的任务时，要首先考虑更适合用哪种模型去表达，之后才是模型训练，调参，模型应用等步骤。

基于循环神经网络RNN的猫狗图像分类

1. 设计思路

使用循环神经网络进行图片分类任务。首先确定图像输入尺寸为150\*150像素，因此在加载循环神经网络时（这里使用的是LSTM），可以使得每一层每一次输入图像的某一行的150个像素，然后循环输入图片的150行，以此来适配图像分类的任务。由于原生彩色图像中的三个通道对于循环网络来说将是一个负担以此还是采用灰度图进行训练。

1. 模型结构

|  |
| --- |
| 1: INPUT()  2: LSTM(64个神经元，tanh激活函数)  3: LSTM(64个神经元，tanh激活函数)  4: LSTM(32个神经元，tanh激活函数)  5: Dense(64个神经元，relu激活函数)  6: Dense(1个神经元，sigmoid激活函数)  7: OUTPUT() |

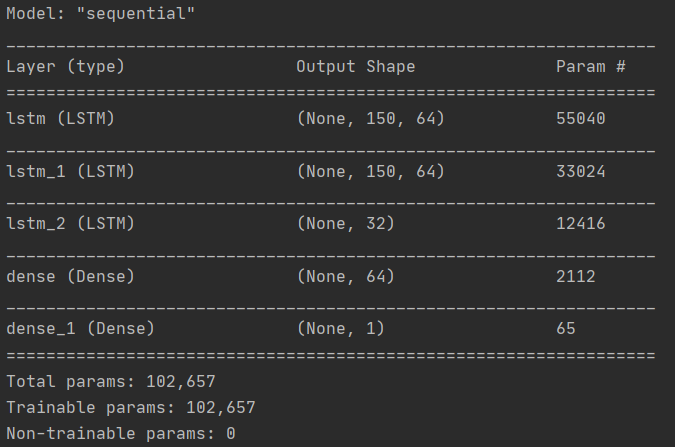


图 5模型的summary

1. 实验结果

使用上述模型结构，损失函数使用二元交叉熵，优化函数使用RMSprop，学习率设置为0.0001，mini\_batch设置为100。使用如上配置进行150轮训练之后得到如图6所示的结果。可以看到模型准确率波动较大，且训练较慢。最终模型基本收敛在100轮之后，验证集上平均准确率只有大约0.56。

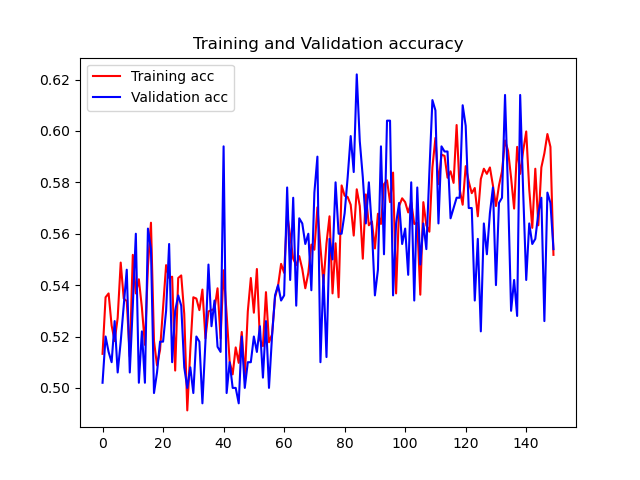


图 6训练150轮训练集和验证集上的准确率

1. 总结与思考

循环神经网络也不适合图片分类的任务（或者说直接使用最朴素的循环神经网络难以直接应用到图片分类任务上），因为按照我所设想的训练输入模式，模型的输入是按照图片的每一行逐像素进入模型，LSTM通过寻找上下文关系来构建模型，而图片的特征不仅仅是行像素上的关系，更多的是局部特征（结合了行和列）。因此LSTM较难学习到图片类别的有效特征。