|  |
| --- |
| 哈尔滨工业大学(深圳) |
| **《人工智能》实验报告** |
|  |
| **实验二**  **深度学习实现花卉识别**   |  |  | | --- | --- | | 学 院: | 计算机科学与技术 | | 班 级: |  |   姓 名:   |  |  | | --- | --- | | 学 号: |  | | 学 期: | 2022春 | | 实验地点: |  | | 实验教师: |  |   目录  [1 实验内容 2](#_Toc97113272)  [2 实验记录 2](#_Toc97113273)  [2.1. 实验环境 2](#_Toc97113277)  [2.2. 数据集处理 2](#_Toc97113278)  [2.3. 模型构建与训练 2](#_Toc97113279)  [2.3.1. \*\*\*模型 3](#_Toc97113280)  [（1） 模型定义 3](#_Toc97113281)  [（2） 模型训练 3](#_Toc97113282)  [2.3.2. \*\*\*模型 3](#_Toc97113283)  [（1） 模型定义 3](#_Toc97113284)  [（2） 模型训练 3](#_Toc97113285)  [2.4. 实验结果 4](#_Toc97113286)  [2.5. 深度学习框架对比 4](#_Toc97113287)  [3 总结 4](#_Toc97113288)  [3.1. 问题及解决方法 4](#_Toc97113290)  [3.2. 实验的启发、总结及建议 4](#_Toc97113291)  [4 参考文献 5](#_Toc97113292)  *实验报告内容包含但不限于以下内容，如有补充请用红色\*标注。*  *注意标注每部分的作者，作者可标注在小标题上。* |

# 实验内容

*简要概述本次实验的内容，用到哪些框架。*

# 实验记录



## 实验环境

*阐述实验用到的环境，包括操作系统、开发软件、用到的库及版本号。*

谢俊安：

* 操作系统：MacOS Monterey12.3.1
* 开发软件：PyCharm
* 库及版本号：
  + Tensorflow 2.6.2

Cv2

Keras 2.6.0

numpy 1.19.5

matplotlib 3.3.4

scikit-learn 0.24.2

## 数据集处理

*阐述对数据集进行的预处理，包括但不限于数据集划分、数据扩充、数据重采样、数据增强等，写明处理的方法、理由及对实验实验结果的影响。*

数据集划分为训练集:测试集=8:2，相对是比较合理的划分方法，既给予了模型足够的训练样本，也获得了充足的测试样例来验证模型的正确性。

## 模型构建与训练

*以下部分以模型为单位写，每种模型均需要包括模型定义和模型训练部分。*

### \*\*\*模型

### 模型定义

*阐述本次实验使用的模型，详细说明模型结构及选型，包含但不限于卷积层、池化层、全连接层、损失函数等主要参数及选型理由，提供完整的模型结构图，截图模型定义代码。*

### 模型训练

*详细说明模型在训练集上的训练速度、收敛速度、收敛精度、超参数设置与调整、优化算法选择等，提供训练日志或loss曲线的截图。*

### \*\*\*模型

### 模型定义

*阐述本次实验使用的模型，详细说明模型结构及选型，包含但不限于卷积层、池化层、全连接层、损失函数等主要参数及选型理由，提供完整的模型结构图，截图模型定义代码。*

### 模型训练

*详细说明模型在训练集上的训练速度、收敛速度、收敛精度、超参数设置与调整、优化算法选择等，提供训练日志或loss曲线的截图。*

### \*\*\*模型

### 模型定义

*阐述本次实验使用的模型，详细说明模型结构及选型，包含但不限于卷积层、池化层、全连接层、损失函数等主要参数及选型理由，提供完整的模型结构图，截图模型定义代码。*

### 模型训练

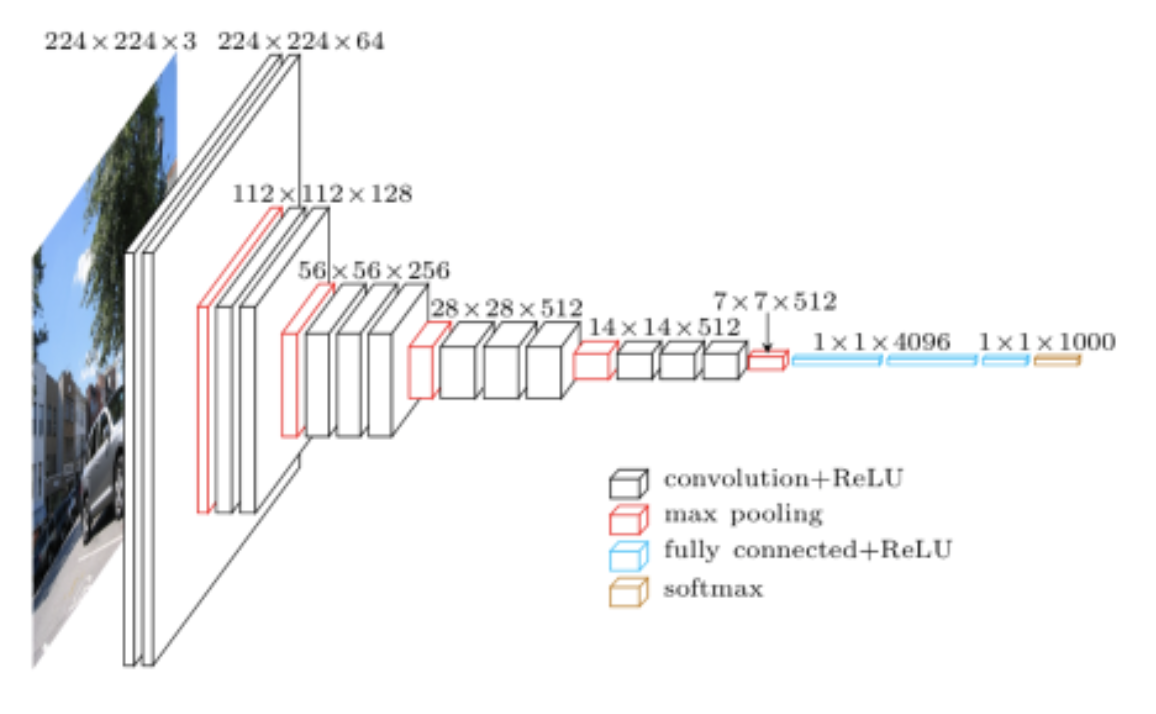
*详细说明模型在训练集上的训练速度、收敛速度、收敛精度、超参数设置与调整、优化算法选择等，提供训练日志或loss曲线的截图。*

### 利用VGG16网络架构进行的迁移学习（by 谢俊安）

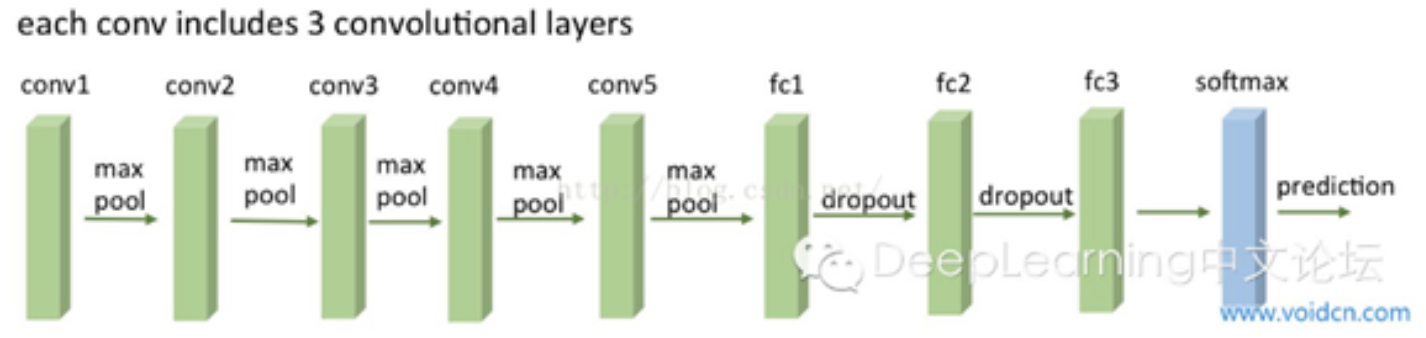
### 模型定义

本模块选用了VGG16网络架构和ImageNet预训练网络进行训练并与本小组自写的模型进行对比参照，VGG卷积神经网络模型在2014ImageNet图像分类与定位挑战赛 ILSVRC-2014中取得在分类任务第二，定位任务第一的优异成绩。

选用的变种是VGG16，这是一个拥有16层的模型，模型结构图如下：



其中具体包括5个卷积层和3个全连接层，VGG16 的第 3、4、5 块：256、512、512个 3×3 滤波器依次用来提取复杂的特征。本实验将顶层的3个全连接网络删除，并进行微调以适应本任务。模型的权重采用ImageNet数据集上预训练的权重。模型的默认输入尺寸是224x224，本任务调整为100\*100以在保持训练精度的情况下降低训练成本。



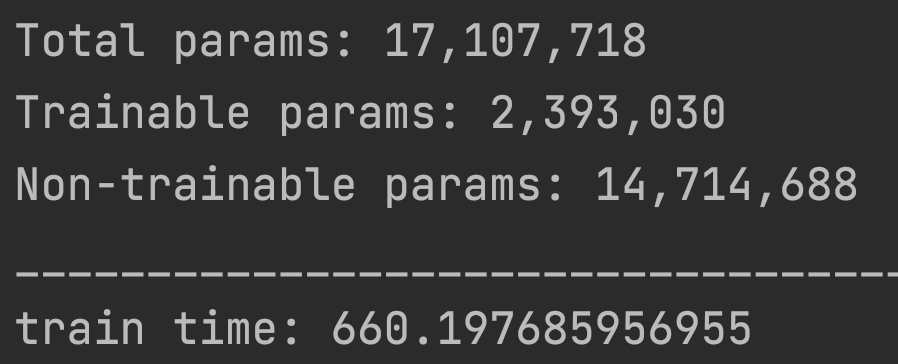
特别地，VGG 的的优越性是通过依次采用多个 3×3 卷积，模仿出更大的感受野效果。模型定义如下：



### 模型训练

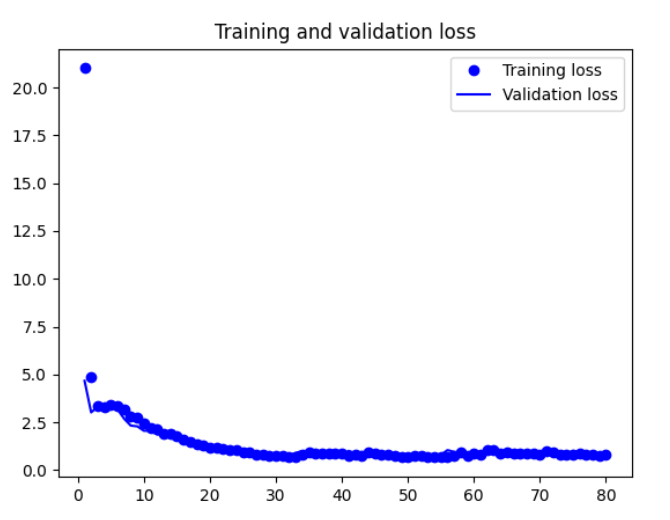
*详细说明模型在训练集上的训练速度、收敛速度、收敛精度、超参数设置与调整、优化算法选择等，提供训练日志或loss曲线的截图。*

* 1. 训练速度：

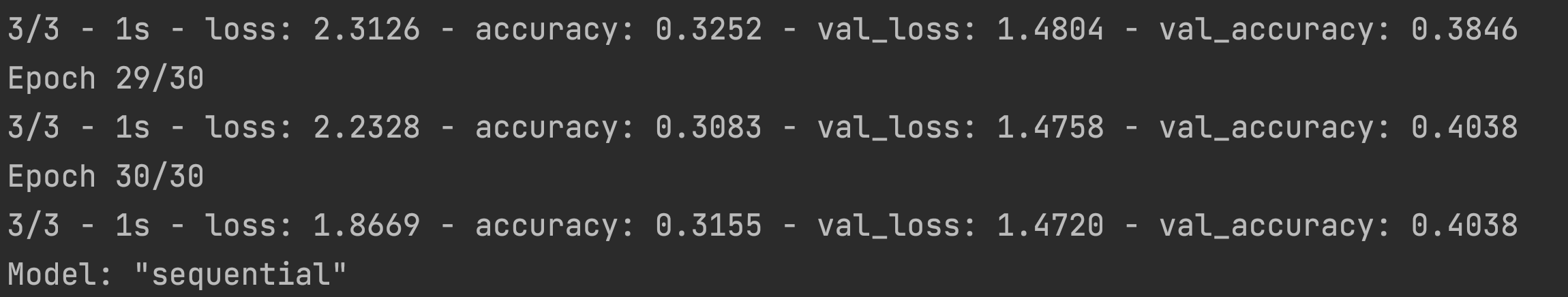


由图可知，模型对精度为100\*100的图片训练80个epoch共耗费660s，训练速度较快。

* 1. 模型基本在epochs=30时已经收敛

****

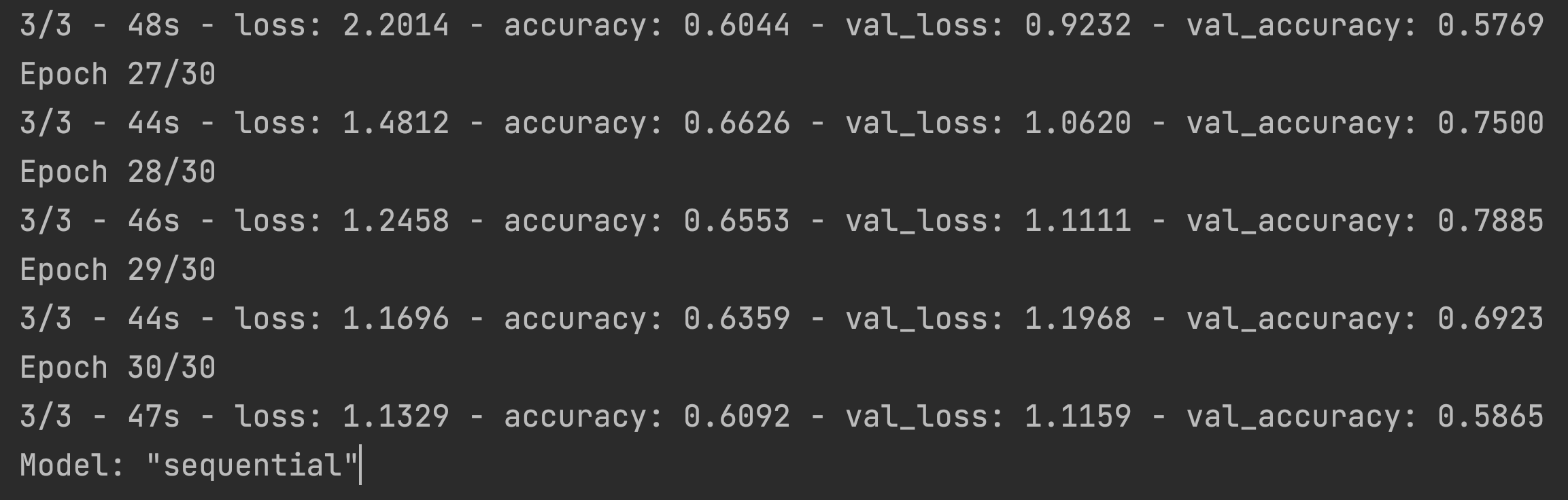
* 1. 超参数：
     1. 最终选择：图片像素为100\*100、epochs=80、batch\_size=200、优化器Adam学习率learning\_rate=0.01、添加l2正则化。
     2. 调整过程：
        1. 图片像素过低导致的精度过低：



分析：如图显示的为32\*32像素的图片样本，个人猜想原因是图片像素被压缩时，物件本身的特征也被平坦化了，导致模型的训练效果不佳。尝试将输入图片像素调整为224\*224.

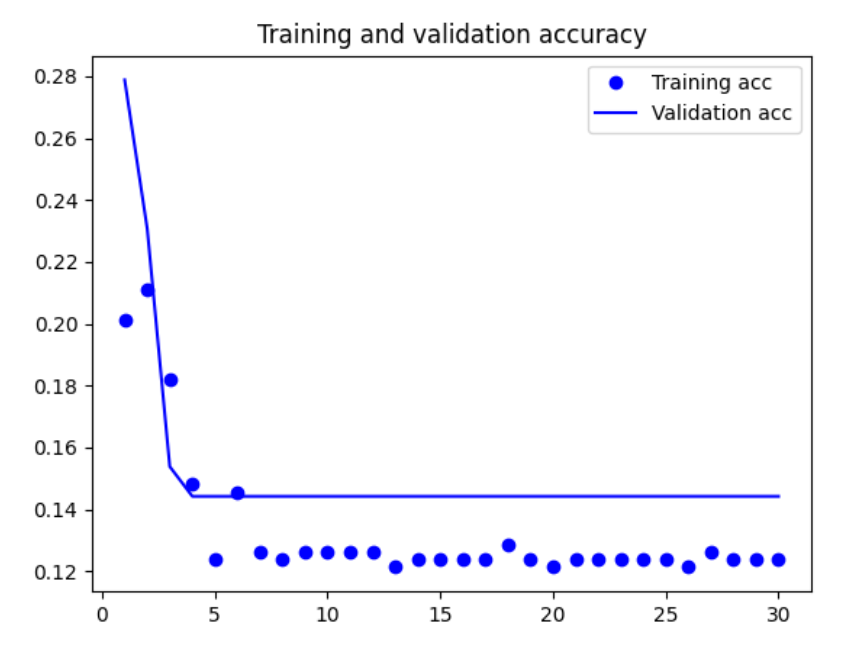
* + - 1. 对原有模型的理解不通透

具体描述：上述问题经调整图片像素之后有了小幅改进，但结果仍不理想（如下图）

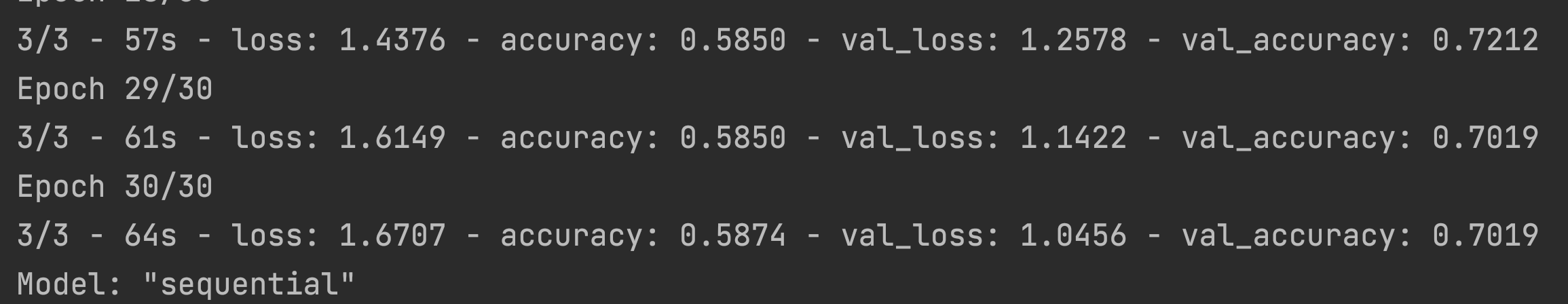


分析原因：没有理解使用的VGG16模型的特性，原模型的最后一个全链接层使用的激活函数是softmax 而不是rula

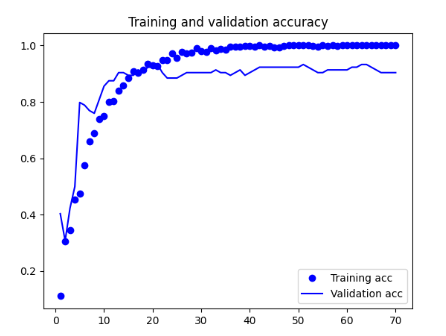
* + - 1. 越训练效果越差



原因及解决办法：样本像素太低，花种的特征已经消失。提高图片像素有很大提升。

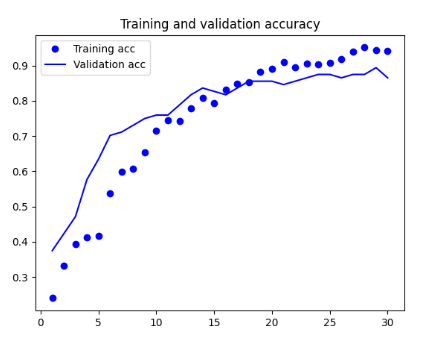


* + - 1. 训练次数过高



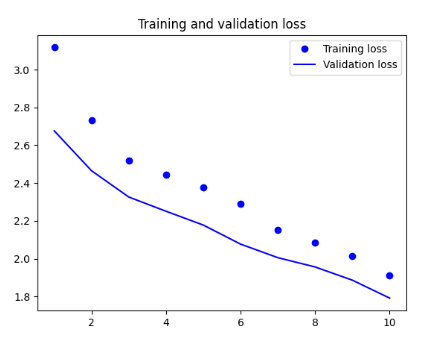
分析：实际上但epochs==30时模型精度已经收敛，趋于稳定，没必要再抬高训练次数，反而造成资源浪费。

* + - 1. 出现模型过拟合的现象



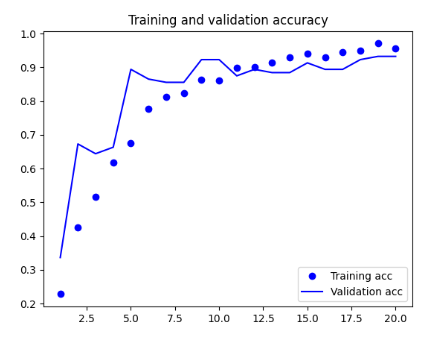
分析原因：学习率过低导致的模型泛化水平差，尝试提升模型学习率0.00001->0.001。

* + - 1. 模型未收敛



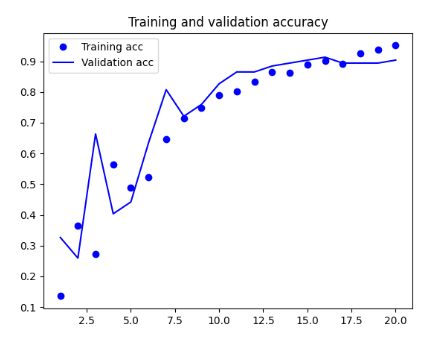
解决办法：提高学习率/增加训练次数

* + - 1. 模型收敛很不稳定



分析原因&解决办法：增加batchsize为原来的N倍时，按照线性缩放规则，学习率应该增加为原来的N。所以应该提升batchsize。

* + - 1. 模型仍旧过拟合

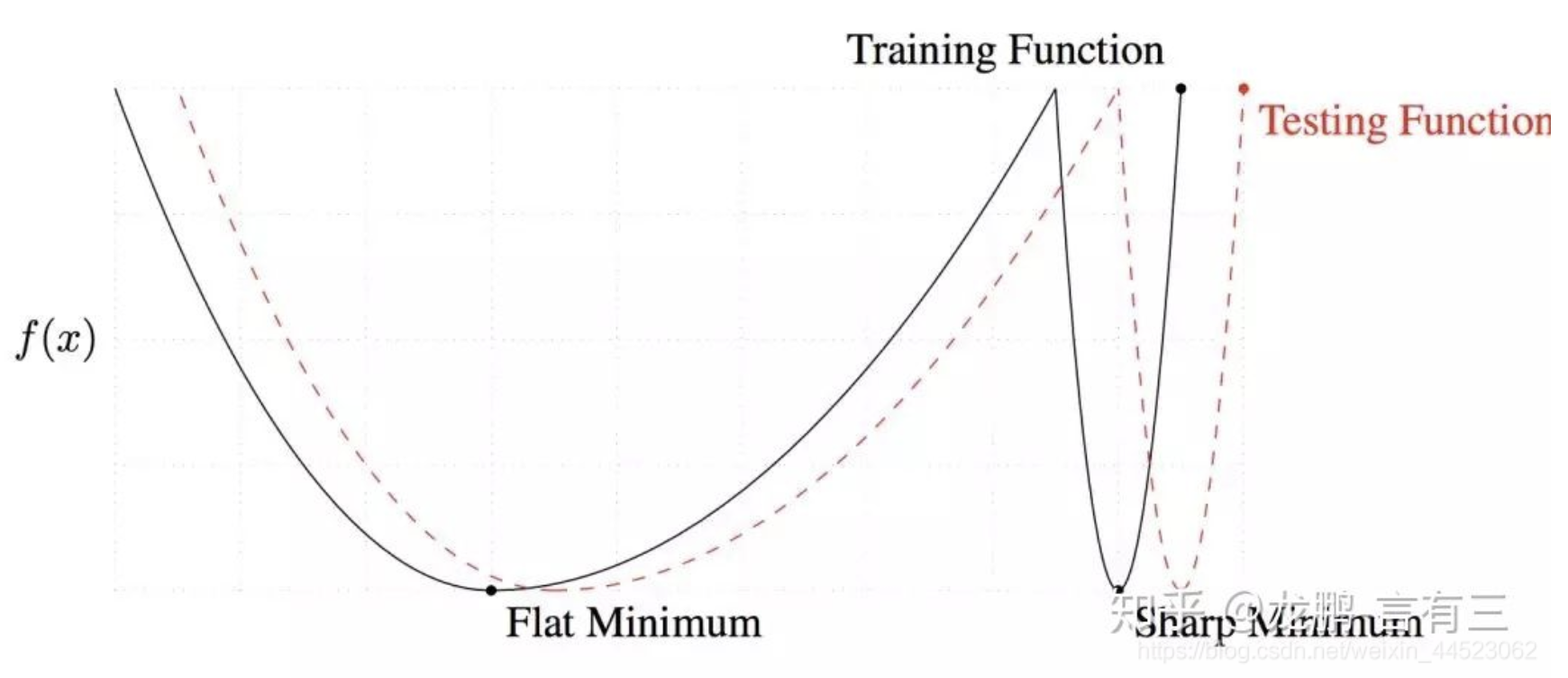


分析原因：

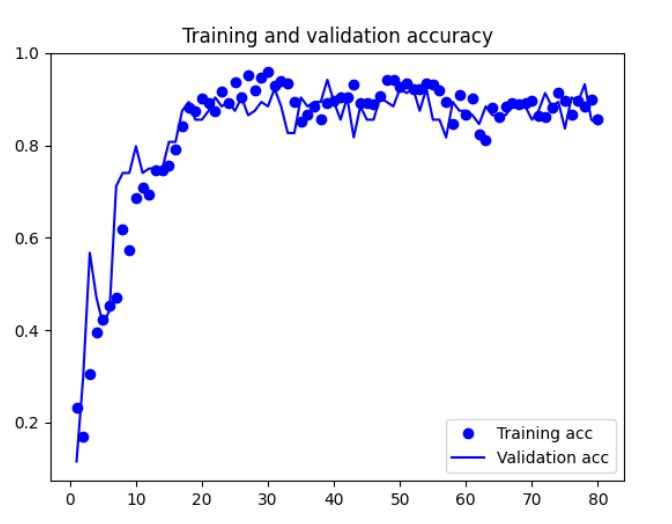
在一定范围内，一般来说 Batch\_Size 越大，其确定的下降方向越准，引起训练震荡越小。但是本次训练中batchsize调整到2000显然有些超出了“一定范围”，而batchsize过大会导致模型收敛到sharp minimum，比batchsize小的模型泛化能力更差。

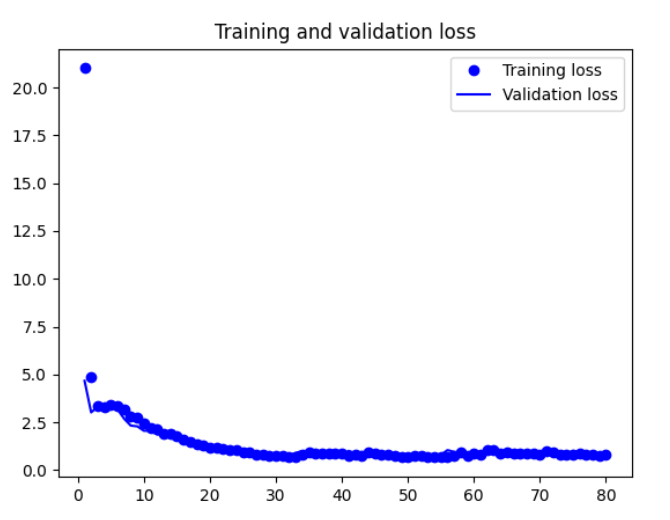
模型训练过度。

解决办法：尝试调整batchsize为200；添加l2正则化防止过拟合；进行dropout率为0.5的dropout防止过拟合。



* + - 1. 拟合效果良好

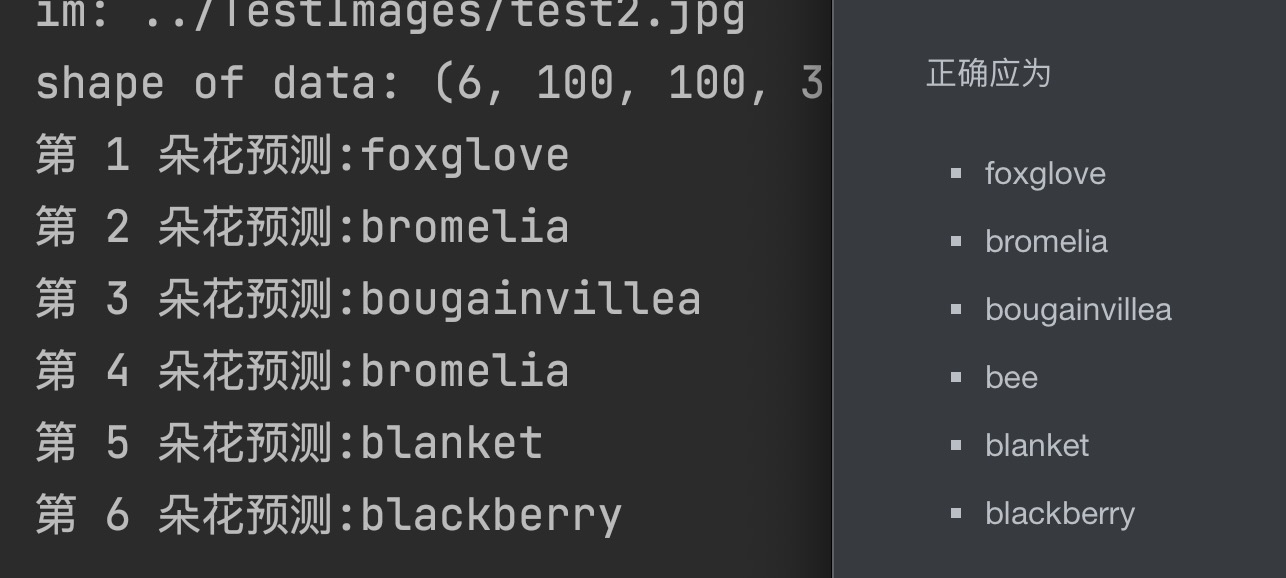
****

****

## 实验结果

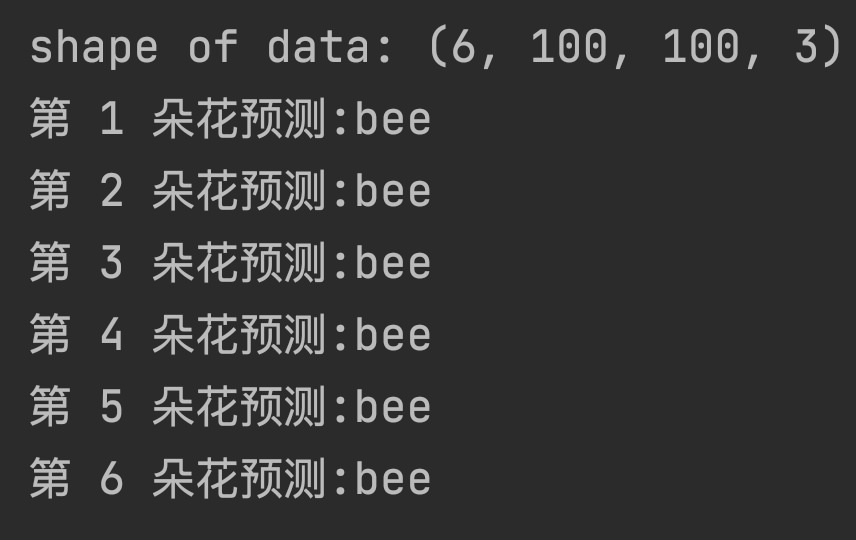
*阐述模型测试与评估结果，包括各种模型在本地预测的效果截图与说明、ModelArts在线部署的预测效果截图与说明。*

本地预测效果：

****

预测结果显示错误的是bee和bromelia花种，初步估计可能是由于训练集太小导致的模型特征提取不够充分，预测效果不太明显。

ModelArts在线部署的预测结果：



可以看到效果非常差，原因是因为ModelArts平台不能访问外网的数据，因此无法在调用模型的时候加载imagenet权重，因此训练效果比本地差。

## 深度学习框架对比

*1.总结对比本地用到的几种深度学习框架的优缺点，在相同模型结构、超参数等情况下，不同框架在开发效率、训练速度及精度、推理速度及精度等方面的区别。*

*2.总结对比云端不同框架的的区别。*

*3.总结对比本地、云端同一框架之间的区别。*

|  |  |  |
| --- | --- | --- |
| 框架 | 优点 | 缺点 |
| TensorFlow |  |  |
| PyTorch | 接口设计精简灵活  搭建网络和调试网络非常方便 |  |
| MindSpore |  |  |

# 总结



## 问题及解决方法

## 实验的启发、总结及建议

谢俊安：

前期盲目地乱套模型+调试参数浪费了大量的时间，但一直处于一知半解的状态。最后去阅读了所选用模型的源码，阅读了相关的论文。对模型本身有了一定的认知，再查看与理解其他项目的调用别的模型方式，才对整个迁移学习的过程有了一定的了解。

做完之后才觉得项目的难度其实也没有很高，需要操作的代码量也没有很大，但是最开始毫不了解的时候确实觉得非常难以下手。哪怕是调用模型的任务，因为网上其他项目的深度学习任务代码框架和老师给的不一样，所以想要真正理解并且完成此次任务还是需要去用心理解透彻之后，再回到框架自己认真书写。实际阅读并理解了一些VGG16的源码之后，感觉其实自己的工作量也并不小。

这门课是深度学习入门的第一堂课，本次实验之后学到了很多很多！

# 参考文献

谢俊安：

[1] VGG in Tensorflow (<https://www.cs.toronto.edu/~frossard/post/vgg16/>]

[2] Usage of VGG16 (<https://github.com/machrisaa/tensorflow-vgg>)

[3] Very Deep Convolutional NetWorks for Large-Scale Image Recognition (<https://kns.cnki.net/kns8/Detail?sfield=fn&QueryID=4&CurRec=1&recid=&FileName=XTYY202109045&DbName=CJFDLAST2021&DbCode=CJFD&yx=&pr=&URLID=>)