**实验三 猫狗分类**

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。

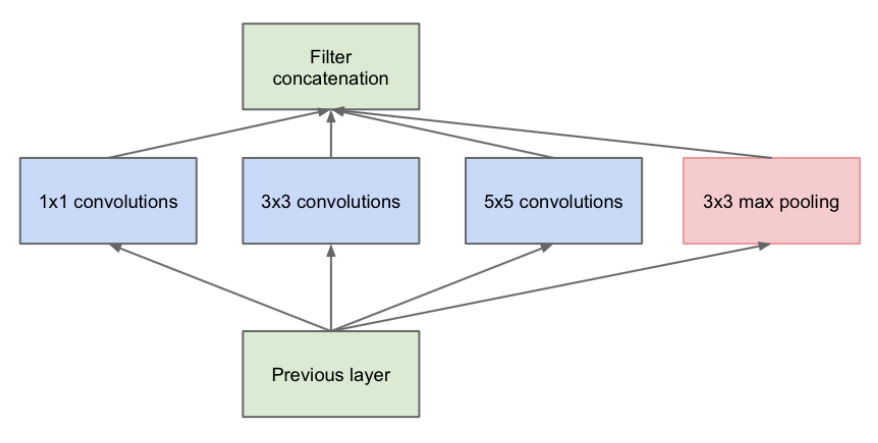


图1 Inception Module（V1版本）

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。

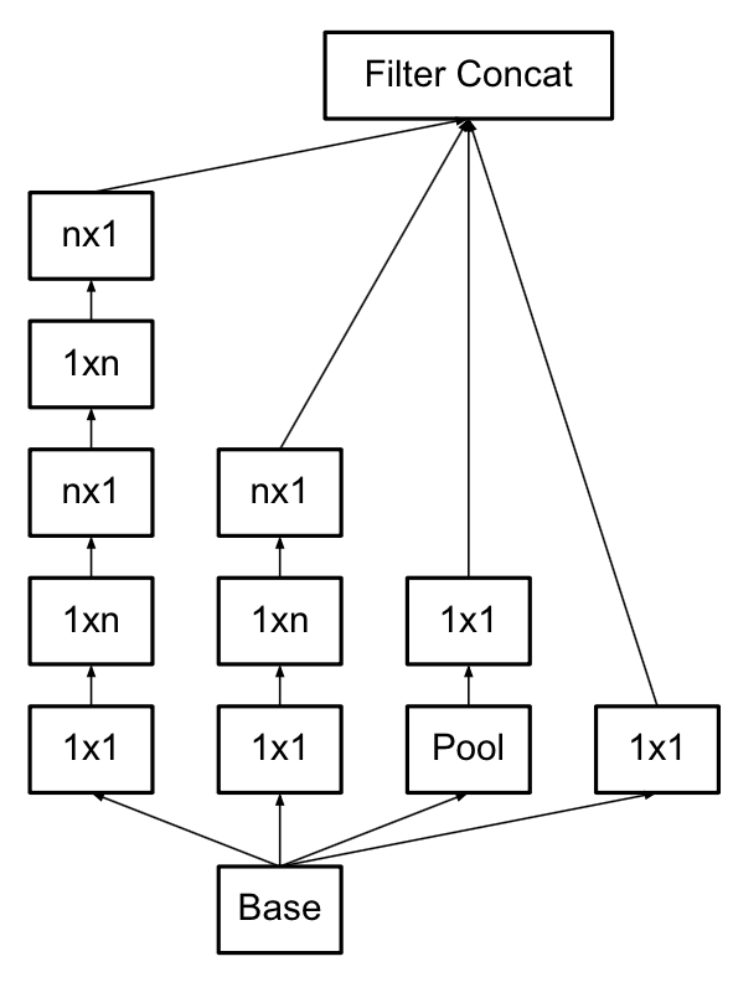


图2 Inception Module（V3版本）

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. Slim工具

Slim是 TensorFlow 中一个用来构建、训练、评估复杂模型的轻量化库，TF-Slim 模块可以和 TensorFlow 中其它API混合使用。参见：

https://github.com/tensorflow/models/tree/master/research/slim。

这里简要介绍Tensorflow Slim的代码结构：

1. datasets/：定义一些训练时用的数据集，预先定义了4个数据集：MNIST、CIFAR-10、Flowers、ImageNet，如果需要训练自己的数据，则可以在datasets文件夹中定义。
2. nets/：定义了一些常用的网络结构如AlexNet、VGG16、Inception系列等。
3. preprocessing/：定义了一些图片预处理和数据增强方法。
4. train\_image\_classifer.py：训练模型的入口代码。
5. eval\_image\_claasifer.py：验证模型的入口代码。

五、实验步骤与方法

对猫狗照片识别分类的卷积神经网络模型可以自行设计，本实验指导书给出的方法是利用Slim工具包中预定义好的网络结构并进行微调的方法实现，具体是基于nets文件夹中预定义好的Inception V3进行微调。

实验可简单地划分为数据准备、模型训练和模型验证三个步骤。

1. 数据准备

首先要将自己的数据集划分为训练集和验证集，根据资源情况可以自行调整训练集和验证集大小

基于TensorFlow提供的API函数自行编写将RGB图转化为tfrecord格式以便模型训练。

def \_int64\_feature(value):

if not isinstance(value, list):

value = [value]

return tf.train.Feature(int64\_list=tf.train.Int64List(value=value))

def \_bytes\_feature(value):

return tf.train.Feature(bytes\_list=tf.train.BytesList(value=[value]))

def \_convert\_example(filename, image\_buffer, label, text, height, width):

colorspace = 'RGB'

channels = 3

image\_format = 'JPEG'

example = tf.train.Example(features=tf.train.Features(feature={

'image/height': \_int64\_feature(height),

'image/width': \_int64\_feature(width),

'image/colorspace': \_bytes\_feature(colorspace),

'image/channels': \_int64\_feature(channels),

'image/class/label': \_int64\_feature(label),

'image/class/text': \_bytes\_feature(text),

'image/format': \_bytes\_feature(image\_format),

'image/filename': \_bytes\_feature(os.path.basename(filename)),

'image/encoded': \_bytes\_feature(image\_buffer)}))

return example

然后在slim/datasets下新建一个dogsVScats，复制flowers.py的内容进来并改变其中的\_FILE\_PATTERN=’dogsVScats\_%s\_\*.tfrecord’

SPLITS\_TO\_SIZES=(‘train’:20000,’validation’:5000)

\_NUM\_CLASSES=2

并在同目录的dataset\_factory.py中对dogsVScats数据集进行注册。

1. 模型训练

在slim文件夹下新建一个dogsVScats目录。在这个目录下：

a.新建一个data 目录，并将第3.2中准备好的5个转换好格式的训练数据复制进去。

b.新建一个空的train\_dir目录，用来保存训练过程中的日志和模型。

c.新建一个pretrained目录，在slim的GitHub页面找到Inception V3 模型的下载地址http://download.tensorflow.org/models/inception\_v3\_2016\_08\_28.tar.gz 下载并解压后，会得到一个inception\_v3.ckpt 文件，将该文件复制到pretrained 目录下。

d.在训练文件夹准备完成后，即可利用slim工具包中的训练模型入口代码train\_image\_clssifier.py进行训练：

python train\_image\_classifier.py

--train\_dir=dogsVScats/train\_dir

--dataset\_name=dogsVScats

--dataset\_split\_name=train

--dataset\_dir=dogsVScats/data

--model\_name=inception\_v3

--checkpoint\_path=dogsVScats/pretrained/inception\_v3.ckpt

--checkpoint\_exclude\_scopes=InceptionV3/Logits,InceptionV3/AuxLogits

--trainable\_scopes=InceptionV3/Logits,InceptionV3/AuxLogits

--max\_number\_of\_steps=100000

--batch\_size=32

--learning\_rate=0.001

--learning\_rate\_decay\_type=fixed

--save\_interval\_secs=300

--save\_summaries\_secs=2

--log\_every\_n\_steps=10

--optimizer=rmsprop

--weight\_decay=0.00004

1. 模型验证

在模型训练完成后，即可利用slim工具包中的验证模型入口代码eval\_image\_clssifier.py进行验证：

python eval\_image\_classifier.py

--checkpoint\_path=dogsVScats/train\_dir

--eval\_dir=dogsVScats/eval\_dir

--dataset\_name=dogsVScats

--dataset\_split\_name=validation

--dataset\_dir=dogsVScats/data

--model\_name=inception\_v3