机器学习工程师纳米学位项目报告

猫狗大战

刘佳业

2018年05月15日

# 零、需求

资料：

https://github.com/nd009/capstone/blob/master/capstone\_report\_template.md

要求

<https://review.udacity.com/#!/rubrics/273/view>

### 提交文件

你的提交最少需要包括下列要求的文件。如果你用 zip 压缩包来提交，请注意总的文件大小。你的提交需要包括：

* 一个项目报告（**只能是pdf格式**），里面包含了项目开发的五个主要阶段。建议的报告长度是9到15页。\*\*请不要通过 Jupyter notebook 导出pdf来提交报告。
* 所有你项目中用到的，能够重现你运算过程的 python 代码。你的代码应该整洁且文档完备。我们建议用Jupyter notebook 作为开发环境。
* 一个 README 文档。包含你项目用到的软件以及库的描述，包括必要的辅助材料来源。如果你的项目需要配置或者启动，也请在 README 报告中给出说明。
* 任何额外的支持材料，例如数据集、图片或者项目开发和实现所必需的输入文件。如果这些文件太大，无法提交。也请在你的 README 文件中**提供一个合适的获取方法**。
* 请在 README 文档中说明运行你的代码所需要的（大概）时间。提交你运行代码成功的截图，也有助于项目评审师复现你的代码。

# 1、问题定义（1-2页）

## 1.1、项目概述

这是一个图像识别的项目，识别图像中的中物体是猫或者是狗（图片中会有许多其他各种物品，但我们只关心其中的猫或者狗），因此，这是一个二分类问题。

该项目源自于Kaggle平台的一个娱乐竞赛项目，因此训练集图片，测试集图片，皆由Kaggle提供，分类结果的评判标准也是以Kaggle平台为准。

本项目将使用卷积神经网络（CNN）作为基础模型，搭建一个具有学习能力的深度学习网络，对输入的训练集图片进行学习及调参，并最终得到一个分类算法模型，可以对训练集的图片进行分类。

CNN是一种类似于滤波器一样的东西，只是普通的滤波器是一维的，而CNN可以是多维的。

项目的输入数据包括12500张标记好的猫的照片和12500张标记好的狗的照片，这25000张图片构成训练图片集，另外有12500张未分类的猫和狗的图片，是测试集，用于最终评估分类算法的性能。

## 1.2、问题陈述

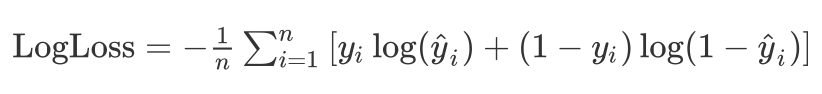
Kaggle竞赛项目中所用到的图片，都是从现实生活中采集的包含猫或者狗的图片，图片的分辨率，图片中的场景、光线、背景各不相同。猫或狗的颜色，姿态，数量也各不相同。这些都有利于提高算法模型的适用范围。

项目过程中，使用了2个步骤进行模型的搭建和模型的学习。其一是使用现成的算法模型Xception提取图片特征；其二是搭建神经网络对提取的特征进行分类，并训练该部分神经网络。

同时，分类算法需要给出分类的自信度，取值为-0-1：1代表100%是狗；0代表0%是狗，也就是100%是猫（这是个二分类问题，不存在其他第三种状态）。

## 1.3、评价指标

Kaggle本次猫狗大战竞赛，使用的评价指标为对数损失函数（越低越好），计算方法如下：



可以看出，该函数不但与正确率相关，也与分类的自信度相关，在当前算法正确率越来越接近100%，正确率对算法的优劣的区分度越来越弱的情况下，自信度就更加重要了。

# 2、分析

## 2.1、数据的探索

输入数据为Kaggle上下载的图片数据，包括：25000张标记好的图片作为训练数据集（训练过程中需要将其拆分为2部分，作为训练数据集和验证数据集），这25000张图片又分为12500张标记为猫的图片和12500张标记为狗的图片。另外，还有12500张图片是未标记的，将作为测试数据集，用于最终评估算法模型的优劣。

随机浏览这些图片可以罚下，在这总共37500张图片里面，图片的分辨率，图片中的场景、光线、背景各不相同。猫或狗的颜色，姿态，数量也各不相同。

另外，由于猫狗所占比重一样，构建训练模型时不再调整类别权重。

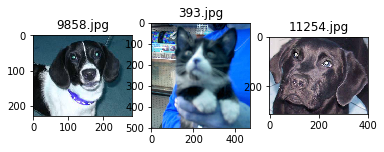
## 2.2、探索性可视化

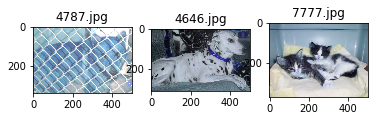
每次随机抽取18张图片，其中6张是标记为猫的，6张标记为狗的，以及6张未标记的测试集图片：

**图1、训练集图片**



**图2、测试集图片**

****



可以看到，从动物形态，图片背景，照片光线等角度看，数据的多样性还是比较完备的。

## 2.3、算法和技术

**2.3.1 算法**

[**https://en.wikipedia.org/wiki/Convolutional\_neural\_network**](https://en.wikipedia.org/wiki/Convolutional_neural_network)

[**https://www.cnblogs.com/charlotte77/p/7759802.html**](https://www.cnblogs.com/charlotte77/p/7759802.html)

[**https://blog.csdn.net/v\_july\_v/article/details/51812459**](https://blog.csdn.net/v_july_v/article/details/51812459)

[**http://cs231n.github.io/convolutional-networks/#overview**](http://cs231n.github.io/convolutional-networks/#overview)

[**https://blog.csdn.net/yjl9122/article/details/70198357**](https://blog.csdn.net/yjl9122/article/details/70198357)

这是一个图片识别问题，也是一个图片分类的问题，在这类问题中，卷积神经网络有较好的优势。

卷积神经网络（Convolutional Neural Network, CNN）是一种前馈人工神经网络，一般情况下，卷积神经网络由一个或者多个卷积层，池化层，激活函数以及顶层全连接层组成。它通过较小的卷积单元（也代表较少的参数），在图片上提取特征，然后又可以使用多个这种较小的卷积单元，以及多层卷积单元，实现对图片不同特征以及非常抽象的特征的提取。

卷积层（Convolutional layer）是卷积神经网络的核心模块，在卷积神经网络中，一般第1、2层卷积层可以提取简单的图片特征，如边缘、线条、角等，更多层、更深层的网络，提取的特征更复杂，更加抽象。

**图3、卷积**



在大量的图片上使用卷积神经网络及反向传播算法，更新这些神经元（包括卷积层）的权重与偏置，使得神经单元的参数能高效和准确地提取图片特征，可实现更准确的分类。

在深度学习快速发展的今天，卷积神经网络越来越复杂，层数也越来越多，达到了几十层，在这些网络里面应用的技术也有长足的发展。

相较于全连接网络，

**2.3.2 技术**

## 2.4、基准模型

以Kaggle的Public LeaderBoard排名10%作为基准，也就是在测试集上的logloss=0.06127(这是当 前Public LeaderBoard上排名为131/1314的分数)。

我在此将会沿⽤用该logloss算法评估我的分类器器性能，期望是使算法模型能在kaggle的Public LeaderBoard排名10%以内，即logloss<0.06127

# 3、 方法

## 3.1、数据预处理

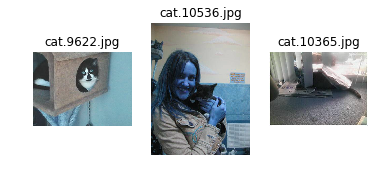
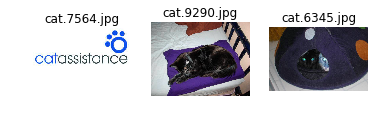
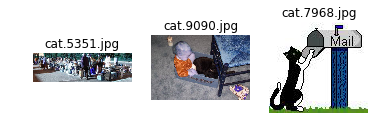
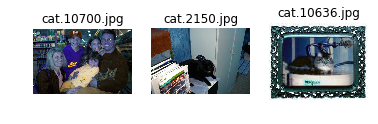
**3.1.1 异常数据处理**

使用预训练网络ResNet50（在imagenet数据集上训练过），以及已有的权重，仅对训练数据集中已经标记的猫或狗的图片进行预测，而不对测试数据集进行预测。

在imagenet中，猫的类别有7种（分别标记为：['n02123045', 'n02123159', 'n02123394', 'n02123597', 'n02124075', 'n02125311', 'n02127052']），而狗的标记有118种，那么我们预测出的结果的Top\_x（x可以是3，5，10，或者100等），如果没有出现对应的猫或者狗，我们就打印出来，再人工辨别。

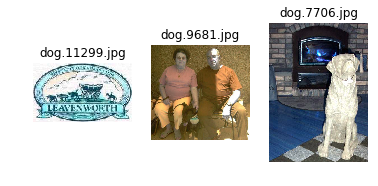
最终，对于猫的异常图片识别，使用了Top100，在Top100里，仍有49张图片预测没有出现猫：

图4、错误标记的猫（举例）



对于狗的异常图片识别，使用了Top20，在Top20里，仍有20张图片预测没有出现：

图5、错误标记的狗（举例）



· 5336jp9 
9 043 jpg 

( 9 , 12155 jpg 
q11437jpq 

cbg.1259.jpg 
camera shy 

这些判定异常的49张猫的图片中，再根据肉眼判断，最终有13张得以保留，其余36张删除。

这些判定异常的40张狗的图片中，再根据肉眼判断，最终有7张得以保留，其余33张删除。

总共删除训练数据69张

**3.1.2 图片输入处理**

由于在项目中是用Xception及其在imagenet的预训练权重提取图片特征，而Xception输入的图片尺寸需要统一为 (299,299,3)，将数据从(0，255)缩放到 (-1，1)，这里使用keras模块中的自带的keras.applications.xception.preprocess\_input进行输入数据的预处理。

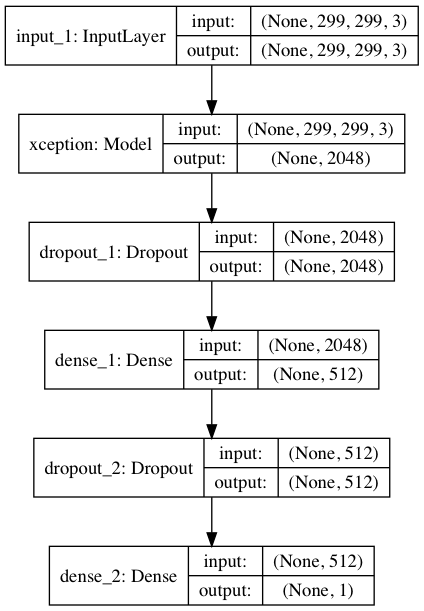
## 3.2、执行过程

**3.2.1 整体方案**

整体模型搭建如图5，具体如下：

1. 先将图片预处理为维度：(299,299,3)；
2. 然后使用一个预训练的Xception模型（也使用其在imagenet上的预训练权重），对图片进行特征的提取；
3. 使用GlobalAveragePooling2D进行池化，是的输出特征的维度变为（nb\_samples, 2048）;
4. 加入Doupout层，防止过拟合；
5. 全连接层，激活函数为'relu'，加快收敛速度，输出维度为（nb\_samples, 512）；
6. Doupout层，防止过拟合；
7. 全连接层，激活函数为'sigmoid'，输出维度为（nb\_samples, 1），也就是每个图片的输出都是（0,1）区间的一个数，就是进行分类。

**图5、图片分类算法模型**

****

**3.2.2 特征提取**

因为Xception的权重在项目中不进行调整，而且Xception的参数多，层数也多。那么在训练过程中，就使用Xception模型将图片的特征提取出来，再将这些特征提供给后面的神经元进行训练。

又因为可能会多次使用这些特征用于训练，因此将提取的特征保存为h5文件，然后就可以随时获取这些特征并进行训练了。

另外，对于Xception模型的输出，如果使用Flatten输出，维度是：(samples,204800)，这个比较大，此处不太必要。最终选择池化层GlobalAveragePooling2D进行输出并保存为h5文件。

**3.2.3 分类模型及训练**

读取由Xception提取的特征后，在上面应用：Doupout层，Dense层，Doupout层，Dense层最后一层使用激活函数'sigmoid'，输出（0, 1）区间的单个值，正好用于分类及分类自信度的表达。

使用随机梯度下降法进行训练，损失函数就是交叉商损失函数。

同时，对于整个训练集，切割其中的20%作为验证数据集，剩下的80才是真正的训练集。

## 3.3、完善

对训练数据集图片进行数据增强：

1. 向左旋转90度；
2. 向右旋转90度；
3. 调整亮度，使用指数型放大公式：

其中，A为原图像素的RGB值，A’为变更后的RGB值。γ为放大指数，取值0.5-2之间。

当γ取值在(0.5, 1) 区间时，图片的亮度降低，但对比度增加；

当γ取值在(1,2) 区间时，图片的亮度增加，但对比度降低；

图6、图片亮度处理(γ分别取值0.5和2)





1. 复制原图，单纯增加图片数量；

# 4、结果

## 4.1、模型的评价与验证

## 4.2、合理性分析

我没有试过这样做

# 5、项目结论

## 5.1、结果可视化

## 5.2、对项目的思考

## 5.3、需要作出的改进

