

机器学习纳米学习 -- 猫狗大战

开题报告

苗沛

2018.09.10

项目背景

Cats vs. Dogs（猫狗大战）是Kaggle上的一个竞赛题目，利用给定的数据集，用算法实现猫和狗的识别。这个是计算机视觉领域的一个问题。

kaggle一共举行过两次猫狗大战的比赛，第一次是在2013年，那个时候使用的是正确率作为评估标准，而在2017年第二次举办猫狗大战的比赛时，使用的是log损失函数。这么做是因为现在深度学习发展十分的迅速，而深度学习尤其适合处理图像方面的问题，如果依旧是使用正确率作为评估标准，那么大多数选手的模型都是99%的正确率，不能明显地区分开。如果使用log损失函数，不仅仅需要分类正确，还需要对结果有一个较高的可信度，这样就能明显地区分各个模型的分类效果，尤其是Top模型的分类效果。

因此参赛者需要训练一个机器学习模型，输入测试集中的图片，输出一个概率，概率越接近1，表示该图片分类结果是狗的概率越高；概率越接近0，表示该图片分类结果是猫的概率越高。

问题描述

从问题的描述可以发现，kaggle猫狗大战竞赛是一个典型的“单标签图像分类”问题，即给定一张图片，系统需要预测出图像属于预先定义类别中的哪一类。在计算机视觉领域，目前解决这类问题的核心技术框架是深度学习（Deep Learning），特别地，针对图像类型的数据，是深度学习中的卷积神经网络（Convolutional Neural Networks, ConvNets）架构。卷积神经网络是一种特殊的神经网络结构，即通过卷积操作可以实现对图像特征的自动学习，选取那些有用的视觉特征以最大化图像分类的准确率。

输入数据

数据集来自 kaggle 上的一个竞赛：[Dogs vs. Cats Redux: Kernels Edition](#)。

下载kaggle猫狗数据集解压后分为 3 个文件 train.zip、test.zip 和 sample_submission.csv。

train 训练集包含了 25000 张猫狗的图片，每张图片包含图片本身和图片名。命名规则根据“type.num.jpg”方式命名。

test 测试集包含了 12500 张猫狗的图片，每张图片命名规则根据“num.jpg”，需要注意的是测试集编号从 1 开始，而训练集的编号从 0 开始。

sample_submission.csv 需要将最终测试集的测试结果写入.csv 文件中，上传至 kaggle 进行打分。

训练集中大部分图片是正常的，有少部分异常图片和低分辨率图片，对于训练集来说这些异常数据是要剔除掉的。

训练集中的图像大小是不固定的，但是神经网络输入节点的个数是固定的。所以在将图像的像素作为输入之前，需要将图像的大小进行resize。

使用深度学习方法识别一张图片是猫还是狗，这是一个二分类问题。1表示分类结果是狗，0表示分类结果是猫。

- 输入：一张彩色图片
- 输出：狗的概率

解决办法

项目要求使用深度学习的方法解决问题，这里拟使用卷积神经网络（CNN）。卷积神经网络(Convolutional Neural Network, CNN)是深度学习技术中极具代表的网络结构之一，在图像处理领域取得了很大的成功，在国际标准的ImageNet数据集上，许多成功的模型都是基于CNN的。CNN相较于传统的图像处理算法的优点之一在于，避免了对图像复杂的前期预处理过程（提取人工特征等），可以直接输入原始图像。CNN网络对图片进行多次卷积和池化层处理，在输出层给出两个节点并进行softmax计算得到两个类别各自的概率。

基准模型

项目使用ResNet50, Xception, Inception V3 这三个模型完成。本项目的最低要求是 kaggle Public Leaderboard 前10%。在kaggle上，总共有1314只队伍参加了比赛，所以需要最终的结果排在131位之前，131位的得分是0.06127，所以目标是模型预测结果要小于0.06127。

评估指标

kaggle 官方的评估标准是 LogLoss，下面的表达式就是二分类问题的 LogLoss 定义。

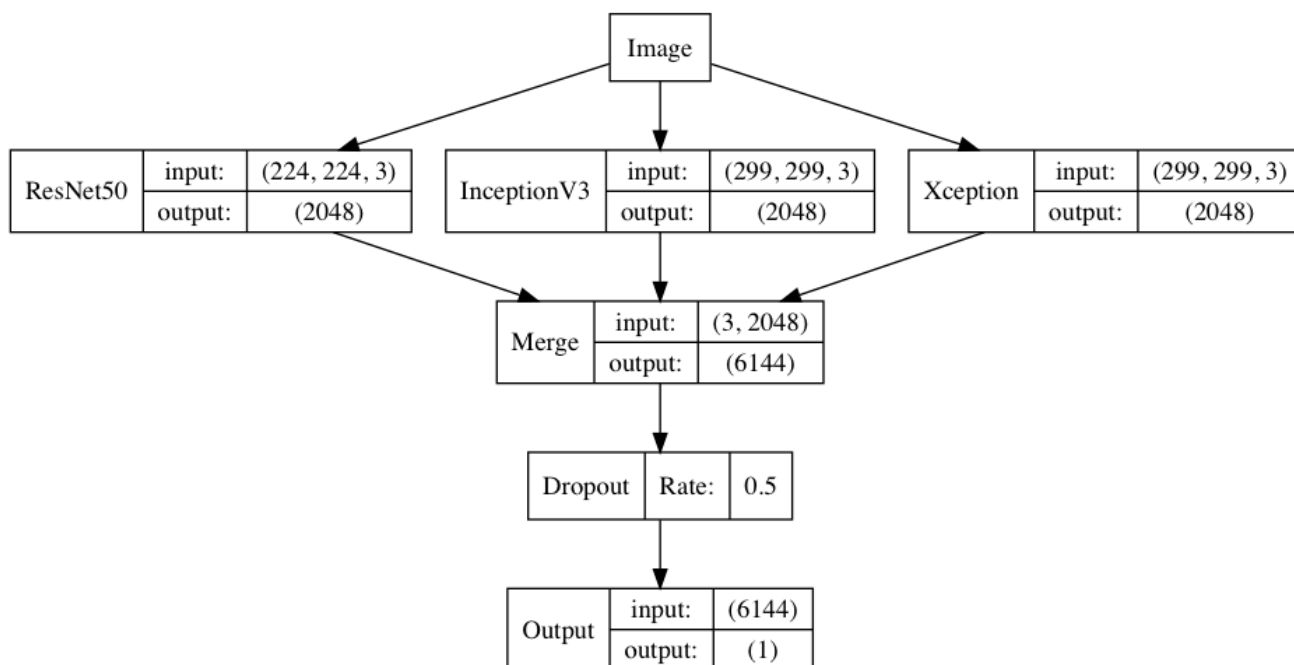
$$LogLoss = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

其中：

- n 是测试集中图片数量
- \hat{y}_i 是图片预测为狗的概率
- y_i 如果图像是狗，则为1，如果是猫，则为0
- $\log()$ 是自然（基数 e ）对数

对数损失越小，代表模型的性能越好。上述评估指标可用于评估该项目的解决方案以及基准模型。

设计大纲



1. 数据预处理

- 从kaggle下载好图片
- 将猫和狗的图片放在不同的文件夹以示分类，使用创建符号链接的方法
- 对图片进行resize，保持输入图片信息大小一致

2. 模型搭建

Kera的应用模块Application提供了带有预训练权重的Keras模型，这些模型可以用来进行预测、特征提取和微调调整。

- Xception 默认输入图片大小是 299*299*3
- InceptionV3 默认输入图片大小是 299*299*3
- ResNet50 默认输入图片大小是 224*224*3

在Keras中载入模型并进行全局平均池化，只需要在载入模型的时候，设置 `include_top=False` , `pooling='avg'` . 每个模型都将图片处理成一个 `1*2048` 的行向量，将这三个行向量进行拼接，得到一个 `1*6144` 的行向量，作为数据预处理的结果。

3. 模型训练&模型调参

载入预处理的数据之后，先进行一次概率为0.5的dropout，然后直接连接输出层，激活函数为Sigmoid，优化器为Adam，输出一个零维张量，表示某张图片中有狗的概率。

4. 模型评估

- 使用 `Logloss` 进行模型评估,上传Kaggle判断是否符合标准

5. 可视化

- 进行数据探索并且可视化原始数据
- 可视化模型训练过程的准确率曲线，损失函数曲线等

参考文献

- [1] Karen Simonyan and Andrew Zisserman. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE- SCALE IMAGE RECOGNITION. At ICLR,2015.
- [2] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
- [3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.
- [4] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.
- [5] Building powerful image classification models using very little data. <https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>
- [6] Dogs vs. Cats: Image Classification with Deep Learning using TensorFlow in Python. <https://www.datasciencecentral.com/profiles/blogs/dogs-vs-cats-image-classification-with-deep-learning-using>
- [7] ImageNet: VGGNet, ResNet, Inception, and Xception with Keras. <https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/>
- [8] The residual module in ResNet as originally proposed by He et al. in 2015.
- [9] Going Deeper with Convolutions. arXiv:1409.4842, 2014.
- [10] Xception: Deep Learning with Depthwise Separable Convolutions. arXiv:1610.02357, 2016.
- [11] An Analysis of Deep Neural Network Models for Practical Applications. arXiv:1605.07678, 2017 .