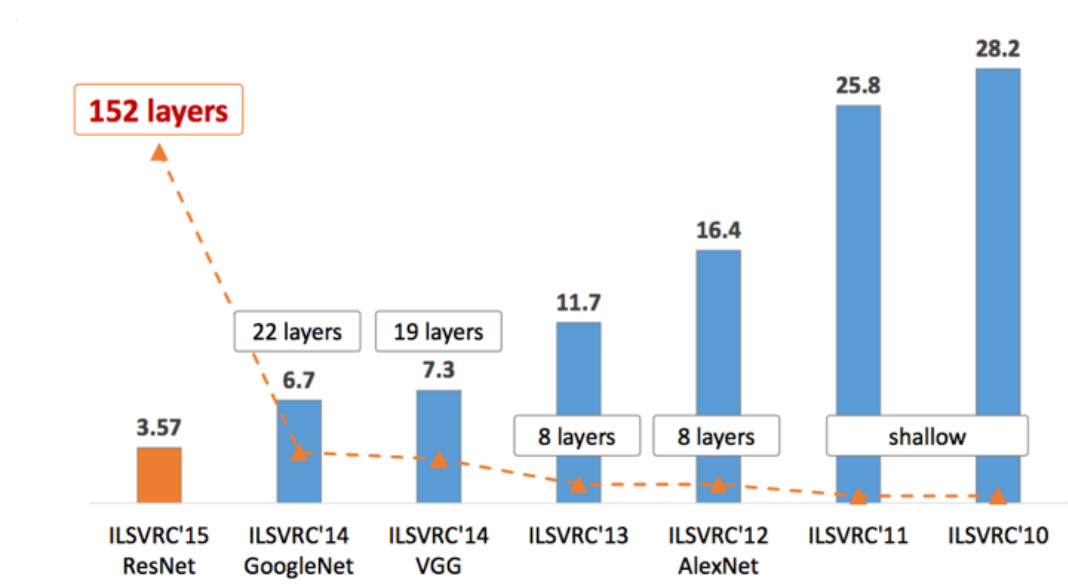


Machine Learning Engineer Nanodegree

猫狗大战项目开题报告

项目背景

猫狗大战是kaggle.com在2013年举办的一场竞赛，即通过计算机识别一张图片是猫还是狗。随着近几年机器学习的发展，特别是计算硬件性能及大数据量级的提高，现在深度学习方法识别猫狗准确率非常高。



如图所示，它们分别获得了ILSVRC比赛分类项目的2012年冠军（AlexNet, top-5错误率16.4%，使用额外数据可达到15.3%，8层神经网络）、2014年亚军（VGGNet, top-5错误率7.3%，19层神经网络），2014年冠军（InceptionNet, top-5错误率6.7%，22层神经网络）和2015年的冠军（ResNet, top-5错误率3.57%，152层神经网络）[1]。

此报告就是基于深度学习和卷积神经网络尝试识别猫狗。

问题描述

问题：使用深度学习方法识别一张图片是猫还是狗。

输入：一张彩色图片

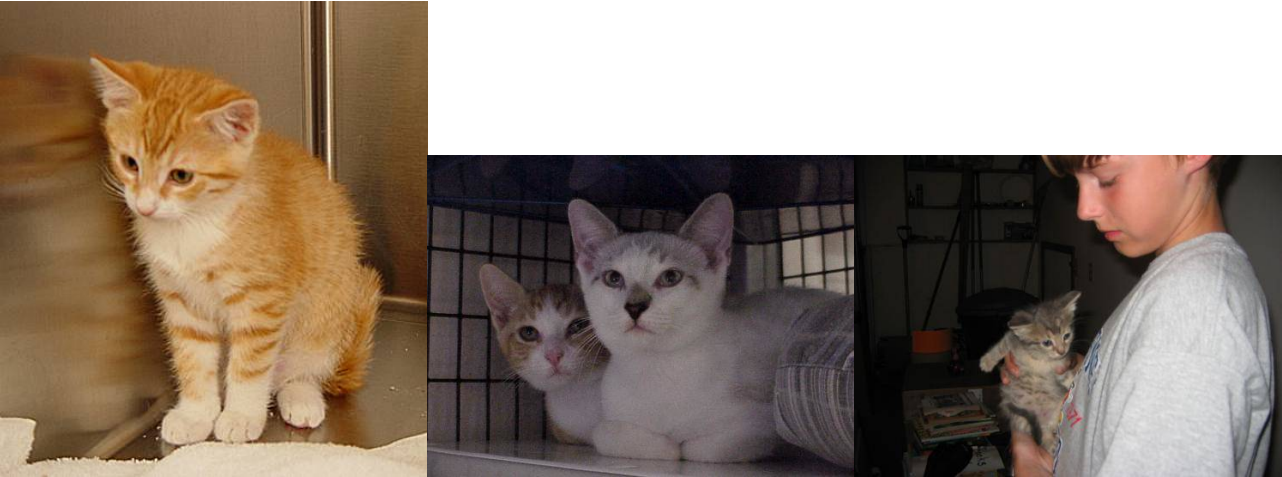
输出：是猫还是狗

在这个项目中，该问题是一个二分类问题。最后输出图片是狗的概率[0,1]

数据和输入

项目数据集可以从[kaggle](#)上下载。
此训练集共有25000张jpg图片，猫狗各12500张，通过文件名区分，图片尺寸不定大小不定。测试集共有12500张jpg图片，没有区分是猫还是狗。

图片场景有：单独出现，多个出现，有人类入镜等。例如：



-----cat.247.jpg-----cat.81.jpg-----cat.1102.jpg-----

还有极少的异常值。例如：



-----dog.1043.jpg-----dog.1773.jpg-----

由于图片大小不一致，这里将会采用Keras的ImageDataGenerator函数进行统一的预处理，生成批次的带实时数据增益的张量图像数据。
另外需要将数据集划分出训练集和验证集，比例暂定8:2。

基准模型

使用ResNet50[2]的迁移学习模型作为基准模型。

下图为使用基准模型在kaggle 上的跑出的分数截图

| | | | |
|--|--|--------------|--------------------------|
| 1 submissions for fanyaohua | | Sort by | Most recent |
| All Successful Selected | | | |
| Submission and Description | | Public Score | Use for Final Score |
| pred.csv a few seconds ago by fanyaohua resnet50 | | 0.07635 | <input type="checkbox"/> |

LogLoss分数为0.07635

评估标准

评估标准为 *LogLoss*，使用 kaggle 官方的二分类 *LogLoss* 公式：

Submissions are scored on the log loss:

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

where

- n is the number of images in the test set
- \hat{y}_i is the predicted probability of the image being a dog
- y_i is 1 if the image is a dog, 0 if cat
- $\log()$ is the natural (base e) logarithm

A smaller log loss is better.

基准阈值为kaggle 排行榜前 10% ，也就是在 Public Leaderboard 上的 *LogLoss* 要低于 0.06127。

所以基准模型跑出的分数还不够。

项目设计

本项目使用Keras, TensorFlow为后端。设计流程为下：

1. 下载数据集 [data](#)

2. 数据预处理

调用Keras的数据预处理API [ImageDataGenerator](#)进行图像预处理,需要先对数据集按猫狗分目录以便后续方法 `flow_from_directory` 调用

3. 导出特征向量

使用Keras的预训练模型提取特征，导出多个不同预训练模型的特征向量综合

4.载入特征向量

载入上一步生成的特征向量，合成一条特征

5.构建模型

调用Keras的API构建模型

6.训练模型

分割训练集验证集，然后进行训练

7.预测测试集

对测试集进行预测，导出csv，然后上传到kaggle相关页面查看得分

8.继续优化

可更换预训练模型，或者对预训练模型进行微调（fine-tune），或者进行数据增强（data augmentation）等

9.部署应用

web应用。用户提交图片，后端分析图片预测猫狗概率输出

参考文献

[1] #Deep Learning回顾#之LeNet、AlexNet、GoogLeNet、VGG、ResNet: <https://www.cnblogs.com/52machinelearning/p/5821591.html>

[2] Deep Residual Learning for Image Recognition: <https://arxiv.org/abs/1512.03385>