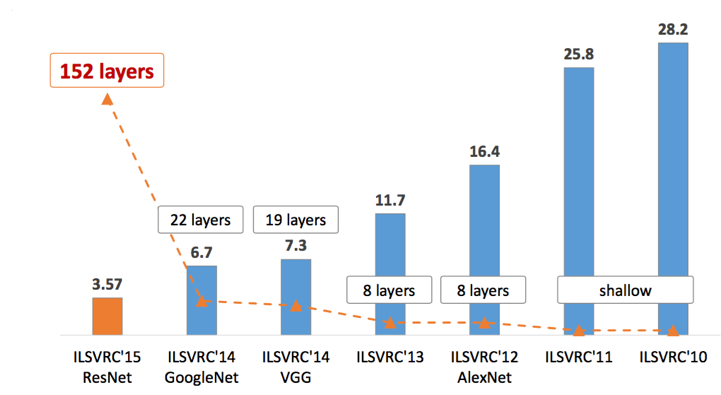
**Machine Learning Engineer Nanodegree**

**猫狗大战项目开题报告**

**项目背景**

猫狗大战是[kaggle.com](https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition)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[1], 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[2]，top-5错误率7.3%，19层神经网络），2014年冠军（GoogleNet[3]，top-5错误率6.7%，22层神经网络）和2015年的冠军（ResNet[4]，top-5错误率3.57%，152层神经网络）。

此报告就是基于深度学习和卷积神经网略尝试识别猫狗。

**问题描述**

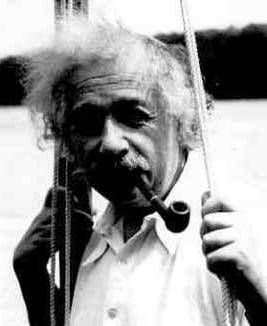
问题：使用深度学习方法识别一张图片是猫还是狗。  
输入：一张彩色图片  
输出：是猫还是狗

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](https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data)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。

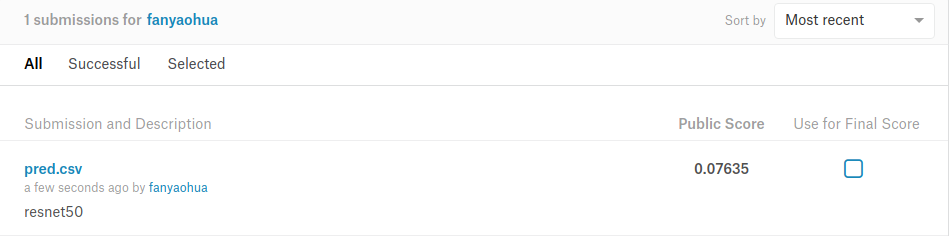
图片场景有：单独出现，多个出现，有人类入镜等。例如：  
  
 cat.247.jpg cat.81.jpg cat.1102.jpg

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

  
logloss分数为0.07635

基准阈值为 kaggle 排行榜前10%，也就是第131/1314名，也就是在 Public Leaderboard 上的 logloss 要低于 0.06127。

**评估标准**

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

其中

* n是测试集中的图像数量
* 是图像是狗的预测概率
* ：如果图像是狗，则为1;如果是猫，则为0
* log()是自然对数e

较小的log loss 更好

**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](https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data)

2.数据预处理  
调用Keras的数据预处理API [ImageDataGenerator](https://keras.io/zh/preprocessing/image/)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] ImageNet Classification with Deep Convolutional Neural Networks. <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>  
[2] Very deep convolutional networks for large-scale image recognization. <https://arxiv.org/pdf/1409.1556v6.pdf>  
[3] Going Deeper with Convolutions. <https://arxiv.org/abs/1409.4842>  
[4] Deep Residual Learning for Image Recognition. <https://arxiv.org/abs/1512.03385>