

# 机器学习（进阶）纳米学位毕业项目开题报告

## 猫狗大战

---

### 项目背景：

近年来，深度学习的发展突飞猛进，使用深度学习技术解决生活中方方面面的问题也成为可能。计算机视觉作为深度学习最重要的领域之一，应用场景非常丰富，可以用于制造业、检验、文档分析、医疗诊断，和军事等领域中各种智能/自主系统中。

该篇文章讨论的课题是：利用深度学习的方法区分猫和狗的问题，该问题是 kaggle 平台提出的一个比赛问题，旨在使用机器学习，特别是深度学习来实现分类问题。深度学习的方法可以有效提取出图片的特征，并对不同类型的分类进行概率输出，故有很高的可行性。而且解决分类问题，对于各个领域的智能识别都具有重要的意义。

---

### 问题描述：

猫狗大战的项目主要是解决猫/狗两个种类的分类问题，是一个有监督的二分类问题。即给定一张图片，模型输出图片上的动物分别是猫和狗的概率。

实现方法：使用卷积神经网络的方法建立模型，顶层使用 softmax 分类器，分别输出猫和狗的概率。

---

### 数据及输入：

本项目的全部数据均由 kaggle 平台的猫狗大战项目提供（下载地址：<https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data>）。

命名格式：其中训练数据集中分别包含了 12500 张猫和狗的图片，格式为 jpg，每张图片使用‘类别’+‘序号’的格式进行命名（如‘cat.0’）。测试集有 12500 张图片，格式为 jpg，以序号命名（如‘23’）。

### 图片特点：

拍摄场景：绝大部分图片都是在真实的生活场景下拍摄的，拍摄使用的光源包括自然光（如图 1）和闪光灯（如图 2）。

拍摄特点：大部分图片为猫狗的特写，小部分有人和猫狗同时出现。其中大部分图片为单一主体（即一只猫或一只狗出现在图片中），少部分会出现多个主体的情况（如图3），也有如图4，出现人物占图片大部分。

姿态特点：宠物拍摄时，姿态各异，面部或身体的角度，部分正对镜头（如图5），部分有一定角度的旋转（如图6）

模糊图片和异常图片：少部分图片有模糊的情况（如图7），并且有一些异常图片（如图8）



图 1 自然光环境



图 2 闪光灯



图 3 多个主体



图 4 人物



图 5 正面

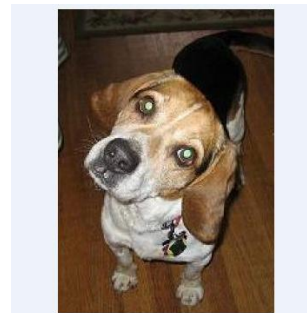


图 6 旋转



图 7 模糊



图 8 异常图片

图片尺寸分布：由下图可以看出，训练集图片的宽分布在 $[32, 768]$ 范围内，其中大部分集中在 100-500 之间；长分布在 $[42, 1050]$ 范围内，其中大部分集中在 200-500。

测试集图片的宽分布在[44, 500]范围内，长分布在[37,500]范围内，其中长宽的大部分都集中在 100-500 之间。

	width	length
count	25000.000000	25000.000000
mean	360.478080	404.09904
std	97.019959	109.03793
min	32.000000	42.000000
25%	301.000000	323.000000
50%	374.000000	447.000000
75%	421.000000	499.000000
max	768.000000	1050.000000

图 9 训练集图片尺寸分布

	width	length
count	12500.000000	12500.000000
mean	359.930720	404.224480
std	96.757411	109.330874
min	44.000000	37.000000
25%	300.000000	329.000000
50%	374.000000	447.000000
75%	418.000000	499.000000
max	500.000000	500.000000

图 10 测试集图片尺寸分布

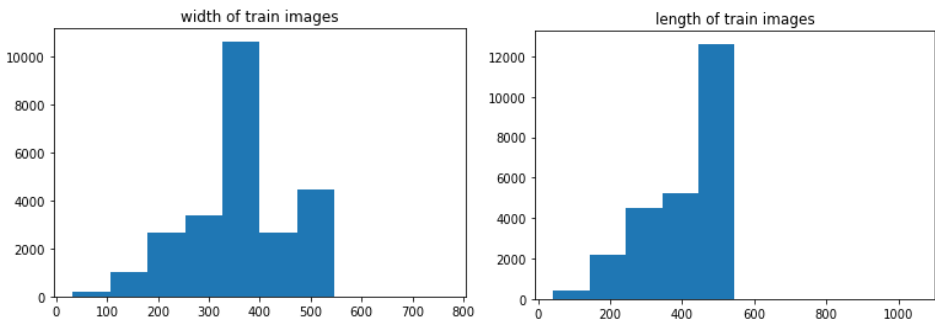


图 11 训练集长宽分布

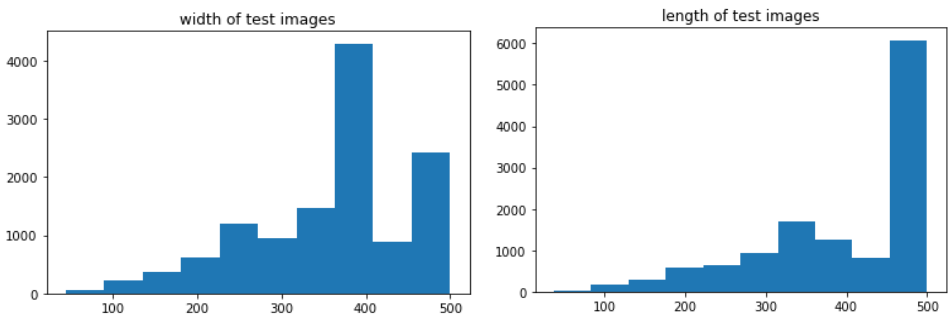


图 12 测试集长宽分布

解决方法描述：

确定使用卷积神经网络的方法来解决该项目的问题，可以采用以下两个方法：

1. 通过建立卷积神经网络，对图片的特征进行提取，在模型的顶层添加一层全连接层，两个输出节点，分别输出图片对于猫和狗的预测概率。
2. 通过迁移学习的方法进行模型搭建。利用 keras 提供的预训练模型如 VGGNet/Xception 等结构进行特征提取，去掉预训练模型的输出层，添加符合该项目的全连接层，以输出概率。

其中方法 1 在训练的过程中会更耗费时间，因为所有的权重都是从随机权重开始训练。方法 2 利用在 imageNet 的大数据集上训练的模型作为基础来建立的模型，可以只对顶层的全连接层进行训练，训练难度降低。综合考虑，选择方法 2。

---

评估标准：

根据 kaggle 官方的标准，应该尽量降低 log loss 的值，计算公式如下：

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

其中 n 是测试集的图片数量

$\hat{y}_i$  是预测为狗狗的概率值

$y_i$  表示如果该图片为狗狗则为 1，为猫则为 0

$\log()$  表示以 e 为底的自然对数

---

基准模型：

kaggle Public Leaderboard 前 10%，即最低应该达到 0.06127

---

项目设计：

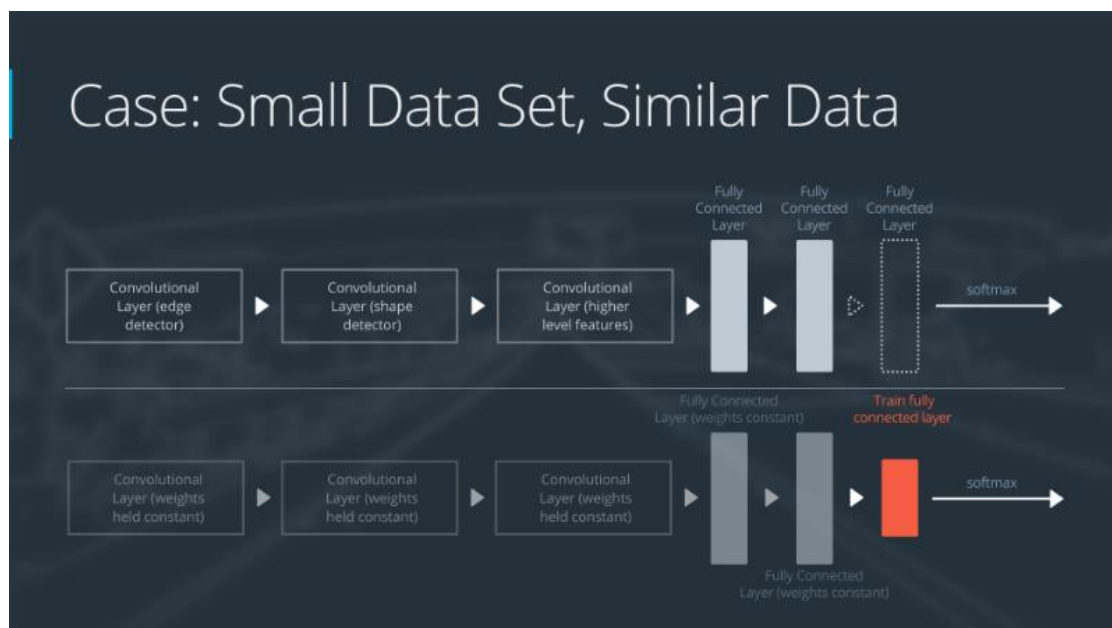
步骤 1：图片预处理。

➤ 图片异常值处理。

- 训练集和验证集划分，按照训练集占 80%，验证集占 20%的比例。
- 剪裁：训练集中的图片的大小不定，故将图片进行剪裁到固定大小，如采用 Xception 进行迁移学习，则将图片剪裁到（299\*299）。
- 归一化：将图片的每个像素值变成[-1,1]之间的浮点数（Xception 在预训练时将像素值预处理到[-1,1]）。
- 数据增强：通过水平翻转、旋转、缩放对图片进行实时增强。

## 步骤 2：模型建立。

- 采用迁移学习的方法，采用 keras 提供的 Xception 预训练模型（模型权重大小为 88M，TOP-1 准确率最高，TOP-2 准确率第二，综合以上故采用该模型），加载预训练权重并去掉顶层的输出层，添加符合该项目的输出层。（由于预训练的模型和该项目的分类数据属于相似数据，且该项目数据属于小数据集，所以只训练顶层权重即可）



（图 13 图片来自 ‘机器学习（进阶）纳米学位课程课件’）

## 步骤 3：调整超参数

主要选择模型的正则化参数 $\lambda$ 和 dropout 层比例。计划采用开源调参工具 hyperas  
（<https://github.com/maxpumperla/hyperas>）

采用早期停止的方式控制训练过程，通过设置 patience 的值来判断模型是否已经收敛，记录 valid loss 最优时的模型权重。

#### 步骤 4：模型评估

可视化训练过程中的训练损失、验证损失、验证集准确率的变化趋势。

抽样可视化预测结果和实际图片的对应情况，观察是否一致。

根据‘评估标准’中的公式计算 log loss 的值。

## 参考文献

1. Francois Chollet Xception: Deep Learning with Depthwise Separable Convolutions
2. Sun 05 June 2016 By Francois Chollet In Tutorials. Building powerful image classification models using very little data
3. Keras 中文文档
4. A survey on transfer Learning. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 10, OCTOBER 2010
5. Wikipedia(2016). Convolutional neural network —— Wikipedia, the free encyclopedia. [Online; accessed 1-November-2016]