

# 猫狗大战

骆炜

2018 年 4 月 8 日

## 目录

<b>1</b>	<b>项目背景</b>	<b>1</b>
<b>2</b>	<b>问题描述</b>	<b>2</b>
<b>3</b>	<b>数据或输入</b>	<b>2</b>
3.1	数据平衡性分析 . . . . .	2
3.2	数据清洗 . . . . .	2
3.3	数据扩充 . . . . .	3
<b>4</b>	<b>解决方法描述</b>	<b>3</b>
<b>5</b>	<b>评估标准</b>	<b>3</b>
5.1	Loss Function . . . . .	3

## 1 项目背景

本项目基于 Kaggle 公开训练的和测试数据集实现对图像中猫狗进行图像识别。图像识别是计算机视觉传统工作之一，也是近年来机器学习相关算法攻克的热点。虽然神经网络早在 20 世纪 40 年代，已有研究学者受到生物学研究成果的启发，建立了相关理论模型，然而受制于当时计算机技术和训练方法等制约，难以实现有效的训练，大多数的研究停留在理论上。传统的神经网络认为每个神经元都需要对输入（比如图像）进行感知，即进行全连接，并且只是进行简单的映射而没有对物体进行抽象处理。而随着研究学者的不断尝试和深入研究，认为可以通过卷积、池化等方式降低计算的难度

并且将图像的特征连接（共享）起来，大大提高网络的可用性。其中典型的代表是由 LeCun 于 1998 年提出的 LeNet5 [1]。虽然受限于当时的计算能力和计算方法，该网络和现今上百层的神经网络相比实在是非常简略，但是该网络为后续的图像识别提供了一个基础的样式和模板。随着图像数据的丰富，为研究学者提供了丰富的训练样本，进而可以训练越来越复杂而性能强大的神经网络，例如谷歌的 Inception V3/V4 和 Xception, ResNeXt [2] 等等。本项目拟基于最新的模型结构和一些实用的方法去对 Kaggle 公开的猫狗数据集进行训练和测试。

## 2 问题描述

猫狗大战是典型的二分类问题。所用的算法需要通过卷积神经网络，对图片中的图像特征进行提取，找出所分类的对象位置及其所属分类。

## 3 数据或输入

数据的来源主要由 Kaggle 提供，包括 25000 张猫狗的训练照片和 12500 张用以测试的测试照片。最后根据识别结果实现对图片中猫狗的分类（狗 = 1，猫 = 0）。另外，根据 Github 的建议，也可以使用 Oxford-IIIT 的数据集进行训练。除了选择的图像识别网络架构外，对数据的预处理是非常重要的。针对图像训练输入较为局限的问题，拟采用数据增强方法对原有数据源进行扩充，避免出现拟合现象。

### 3.1 数据平衡性分析

从原始的数据分类是来说，在训练集中分别给出了 25000 张照片，原始的标记各一半，平衡性相较于大多数分辨实验中已经算是比较好的。

### 3.2 数据清洗

根据对数据进行观察，发现并不是所有标记数据都能够正确反映所属的类别，亦或存在空/错误的图像。如果将这些照片选入进行训练，必然会让神经网络的训练受到干扰。因此有必要在展开训练前对数据进行筛选或是清洗。可采用的方法为通过已经训练完成的模型（例如基于 cifar10 或是其他

涵盖猫狗这两个分类的数据) 预先进行分类筛查。通过至少两个模型的筛查后, 剔除所检测出的非猫狗图像。

### 3.3 数据扩充

对于神经网络训练而言, 由于参数众多, 对于所需的训练数据的数量有较大的要求。除了从网上搜集更多的带有标记的数据以外, 可以使用数据增强的方法, 使用已有数据集中的数据, 通过旋转、剪接、高斯模糊等手法对数据进行扩充。另外, 可以考虑使用 GAN 技术对图像进行扩充, 最后综合原数据、数据增强产生的数据和 GAN 生成的数据对所选的模型进行训练。

## 4 解决方法描述

近些年来, 图像识别技术得到突飞猛进地发展。在每一年顶级会议上(如 CVPR、ICCV 和 NIPS 等), 均有国内外学者提出更新更快的网络结构来提升识别的准确率或是能够在较为轻量的平台上实现。本项目拟采用至少 2 种常用的网络架构, 例如 ResNet 或是 MobileNetV2 等, 对输入图像进行识别。拟使用迁移学习的方法, 基于他人已训练的通用性很强的模型进行修改和训练, 自定义符合本项目二分类要求的网络结构。

## 5 评估标准

评估标准拟采用在测试集中随机抽取图像后进行分类判断, 所采用的分类辨识器可以使用 Multiclass Support Vector Machine [3] 或是更常见的 Softmax 分类器 [4]。本项目所选用的基准模型为 Kaggle 的评分标准, 至少需要达到 Leaderboard 的前 10% 以上的水平。

### 5.1 Loss Function

在使用或是设定评价函数的时候, 通常是通过将 softmax 的输出用交叉熵的方法计算 loss。普通的交叉熵的公式为公式1。其中  $\{p_i\}$  是预测的分布, 而  $\{q_i\}$  是真实的分布。

$$S(q|p) = - \sum_i q_i \log p_i \quad (1)$$

所采用的 loss function 如公式2所示。和单一的使用交叉熵方法相比，该方法能避免模型盲目增大  $e^{z_j}$  的同时还能防止过拟合的出现。

$$loss_j = -(1 - \epsilon) \log(e^{z_j}) - \epsilon \sum_{i=1}^n \frac{1}{3} \log(e^{z_i}/Z), Z = e^{z_1} + e^{z_2} + e^{z_3} \quad (2)$$

## 参考文献

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84-90, May 2017.
- [2] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.
- [3] S. Y. Kung. Support vector learning models for outlier detection. *Kernel Methods and Machine Learning*, page 380-394, 2014.
- [4] Xingqun Qi, Tianhui Wang, and Jiaming Liu. Comparison of support vector machine and softmax classifiers in computer vision. *2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Dec 2017.