

机器学习纳米学位-猫狗大战-开题报告

开题报告

朱成亮

2019.01.11 初稿

2019.01.20 修改

项目背景

本项目源自 Kaggle 平台上的一个竞赛项目，目的是训练出一个模型来分辨图像中的猫和狗，是一个典型的图片二分类问题。

问题描述

这是一个典型的图片二分类问题，即输入一张图片，系统需要判别其属于已知两个分类狗和猫中的哪一类。

输入：一张彩色图片

输出：是猫还是狗

本项目是识别图片内容，属于图片处理或计算机视觉领域。近年来随着深度学习的研究发展，在图像处理领域取得了很大的成功，基于 CNN(卷积神经网络)构建的深度学习模型(比如 Xception, VGG19, NASNetLarge 等模型)对于图像识别分类有很高的准确率，可以很好的解决此问题。

数据集数据输入

项目的数据集来源于 Kaggle 竞赛的数据，此数据集可以从 Kaggle 上下载 [Dogs vs. Cats Redux: Kernels Edition](#)。数据下载后有训练集 train.zip、测试集 test.zip 以及测试结果 excel 表格 sample_submission.csv (用来上传至 Kaggle 打分)。

- 该数据集中训练集有 25000 张已经标记好的猫和狗的图片,猫狗各一半; 标记是图片文件名的一部分 type_number.jpg, 比如 cat.0.jpg; 可以使用 Keras 的 ImageDataGenerator 将不同种类的图片分在不同的文件夹中。
- 测试集有 12500 张未被标记的图片, 图片文件名是以数字命名。
- 随机观察了训练集中的部分图片, 有少部分异常图片, 需要从训练集中剔除这些图片, 如图 1 所示。
- 数据集中的图片大小是不同的, 而且所选的基准模型的输入要求是 299 x 299, 所以在输入前需要对图像的大小 resize。

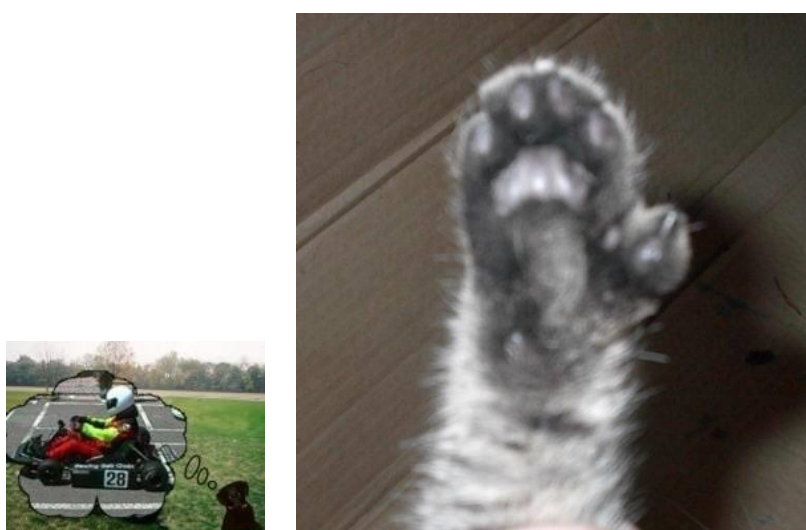


图 1

对于最终结果, 训练后的模型要预测图片里是狗的概率(1=狗, 0=猫), 将最终测试集的结果写入 sample_submission.csv 上传至 Kaggle 进行打分。

解决方案描述

卷积神经网络是一种多层神经网络, 擅长处理图像的相关机器学习问题。CNN 通过一系列方法, 成功将数据量庞大的图像识别问题不断降维, 最终使其能够被训练。

CNN 网络中的卷积层和池化层主要是对图片的特征进行抽取, 比如浅层的卷积层和池化层先抽取图片的一些简单几何信息, 比如边界、直线和角点等, 深层的卷积层和池化层抽取复杂的抽象信息如猫狗特征的信息, 最后全连接层是对分类的处理。

重新设计和训练一个好的模型不仅需要很强的理论和实践能力，而且还要耗费大量的时间和精力，对目前的我来说不具备可行性。因此本项目会综合已经在 ImageNet 训练好的模型比如 ResNet50, Inception V3 作为基准模型，然后通过迁移学习的方案来解决本项目图片二分类问题。

项目工具选择 Keras, Keras 是一个很流行的深度学习框架，Keras 的开发重点是支持快速的实验。能够以最小的时延把你的想法转换为实验结果，是做好研究的关键。而且 Keras Applications 模块提供了预先训练的 Xception, Inception V3 模型。

基准模型

图像识别在工业界已经有非常突出的研究成果和应用，近几年来产生了一些非常优秀的 CNN 的模型，比如 Inception V3, Xception, VGG19, NASNetLarge 等。本项目将会综合尝试使用 Xception, Inception V3 作为基准模型。

本项目最低要求是 Kaggle Public Leaderboard 前 10%，按目前提交的数据 LogLoss 要低于 0.06127。

评估指标

Kaggle 官方评估标准： $\text{LogLoss} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$,

n 测试集图片数量；

\hat{y}_i 图片预测为狗的概率；

y_i 如果是狗则为 1，是猫则为 0；

$\log()$ 自然对数；

对数损失越小，代表模型越好。

项目设计

本项目总体的设计思路是综合利用 Keras 里在 ImageNet 上预训练好的 Xception, Inception V3 模型通过迁移学习思路来设计新的模型，解决猫狗识别的图片二分类问题。

数据预处理

1. 使用 Keras 的 ImageDataGenerator 将猫狗的图片分别放入不同文件夹中；
2. Xception 和 Inception V3 基准模型的输入大小是 299 x 299，需要对图片进行 resize 操作；
3. 图片清理，考虑使用 Keras 预训练好的模型识别图片是不是属于猫狗，把不是猫狗的图片剔除。

模型设计

本项目使用的数据集相对来说是比较小的，而且猫狗图片与 ImageNet 类别的子集有很高的重叠性。

常见的 CNN 结构在前面的若干层都是卷积、池化层以及其变种，后面几层是全连接层，全连接层之前的网络层称为瓶颈层。CNN 结构中从开始到瓶颈层的过程可以看做是对图片的特征提取，因此这部分可以使用已经在 ImageNet 训练好的 CNN 模型的网络。

使用 Keras 里 Application 模块里预训练好的模型 Xception 和 Inception V3，直接使用其相应的结构和权重，冻结除了最后的全连接层和输出层以外的所有层，添加自己的全连接层和输出层到模型，然后使用 Kaggle 提供的猫狗数据集对新的模型进行训练。

模型训练

- 尝试不同的 dropout 对单一参数进行优化
- 因为是二分零问题，所以激活函数使用 Sigmoid
- 优化器使用 Adam，输出一个零维张量，表示某张图片中是狗的概率
- 因为用 sigmoid 作为激活函数，损失函数一般会选用 `binary_crossentropy`。
- 训练时，设置验证集大小为 20%（即 `validation_split = 0.2`）

模型评估

使用训练后的模型对 Kaggle 的测试数据集进行预测，预测结果写入 `sample_submission.csv`，然后上传至 Kaggle 进行打分。

数据可视化

尝试在项目中对其中的一些关键步骤和结果做可视化显示,直观展现项目实现过程和结果。会可视化的关键步骤和结果有:

- 图片的尺寸分布
- 异常图片
- 模型的流程和结构
- 可视化学习过程中的准确率曲线, 损失函数曲线等
- Kaggle 的验证结果

参考文献

- [1] Inception Net-V3 结构图 <https://www.jianshu.com/p/3bbf0675cfce>
- [2] Keras Document: <https://keras.io/applications/#available-models>
- [3] Transfer learning & The art of using Pre-trained Models in Deep Learning: <https://www.analyticsvidhya.com/blog/2017/06/transfer-learning-the-art-of-fine-tuning-a-pre-trained-model/>
- [4] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In arXiv, 2015.