

开题报告

猫狗大战

万嘉俊

06/23/2018

项目背景

毕业项目猫狗大战是 Kaggle 上的一个比赛，根据给定的图片模型识别为猫还是狗的准确率来排名，是一个计算机视觉问题。计算机视觉的本质上是深度学习的应用，深度学习在近年来取得了巨大的突破，更强的计算能力，更多的数据，更好的算法模型使得近十年来深度学习在人工智能领域取得重要的突破。不仅在计算机视觉领域，在自然语言处理等等领域都取得了巨大成功。

深度学习在计算机视觉领域最具影响力的突破发生在 2012 年，欣顿的研究小组采用深度学习赢得了 ImageNet 图像分类比赛的冠军。排名第 2 到第 4 位的小组采用的都是传统的计算机视觉方法、手工设计的特征，他们之间准确率的差别不超过 1%。欣顿研究小组的准确率超出第二名 10% 以上。这个结果在计算机视觉领域产生了极大的震动，引发了深度学习的热潮¹。

问题描述

本项目的目标是构建一个模型能够识别出没有标签的猫和狗的图片并对其进行分类，是一个二分类问题。我们的训练数据是有标签的，这属于监督学习，使用有的标签的图片训练模型，训练后对没有标签的图片进行分类。为了使项目识别准确率更高，使用深度学习构建神经网络的方法来识别图片。这个问题是可量化的、可衡量、可复制的因为通过训练模型，所有的测试图片模型都能将输入的图片数据转换为猫或狗的概率。

输入数据

输入的数据是 Kaggle 猫狗大战比赛上的，包含两个 zip 文件，分别是 train.zip 和 test.zip。train.zip 用来训练模型，test.zip 用来测试模型。

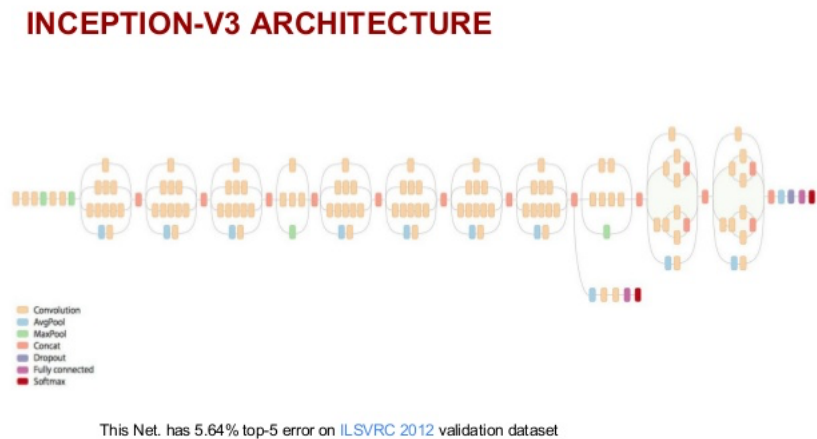
训练数据共有 25000 张图片，猫和狗各有 12500 张，每张图片都有猫或狗的类别标签。测试数据共有 12500 张无标签的图片。将训练集的图片按 4:1 分成训练集和验证集。其中图片大小不一，清晰程度不一，也有个别的异常值例如遮挡与非猫狗图片。在代码实现中，我们使用

keras.preprocessing.image.ImageDataGenerator² 加载图像。像素值放缩到 0 和 1 之间，再将照片 resize 到 224*224 供之后的模型使用。

解决方法

相比传统模型而言，深度学习中的卷积神经网络（CNN）能够有效提升图像识别的性能，特别是以 CNN 为基础的多层神经网络模型在 ImageNet 数据集图像识别竞赛上取得了很突出的成果如 Xception, InceptionV3, VGG, Resnet50 等等，对这些模型进行迁移学习引用到此项目上上可以取得很好的效果⁴。

InceptionV3³ 模型是此项目中迁移学习用到的模型，它的模型结构如下：



* Photo Credits - Google Research

29

这其中有 60 多层网络，结构十分复杂。复杂的结构体现了它优异的表现，单个模型只有 5.64% 的 top-5 error，四个这个模型的整合在 ImageNet 验证集上只有 3.58% top-5 error，表现十分优秀²。

Network	Models Evaluated	Crops Evaluated	Top-1 Error	Top-5 Error
VGGNet [18]	2	-	23.7%	6.8%
GoogLeNet [20]	7	144	-	6.67%
PReLU [6]	-	-	-	4.94%
BN-Inception [7]	6	144	20.1%	4.9%
Inception-v3	4	144	17.2%	3.58%*

与其他的一些模型相比，InceptionV3 表现十分出色，这也是我迁移学习 InceptionV3 的原因之一。

评估标准

使用模型在测试集中的得分对数损失(LogLoss)来对指标评估采用来衡量：

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

\hat{y}_i 表示我们预测出来的结果， y_i 表示图片的正确归类，1 对应狗，0 对应猫，n 表示测试集样本个数。

对数损失越小，代表模型的性能越好。最后，需要进入 Kaggle 猫狗大战的前 10%，LogLoss 分数大概在 0.06127 以下。

基准模型

基准模型的基准阈值是猫狗大战的前 10%，LogLoss 分数大概在 0.06127。我们设计的模型需要比基准模型更好，也就是说 LogLoss 要在 0.06127 以下。

项目设计

1. 从 Kaggle 下载数据，将猫狗图片分类，训练集内图片分开，将猫放入一个文件夹，狗放入另一个文件夹。对于数据集中的异常值我没有选择删除是因为考虑到测试集也会有异常值，如果删掉了训练集的异常值，模型就不能很好的学会这些异常值的处理，所以在测试集上无法做出有利于分数的预测。使用 `keras.ImageDataGenerator` 准备数据，对图片进行 `resize`，保持输入图片信息大小一致。之后训练导出载入 InceptionV3 特征向量。
2. 在 `keras.applications`² 引用带有预训练权重的 Keras 模型，利用这些模型来搭建整合最终的模型。使用 InceptionV3 去掉了全连接层预训练模型，然后添加自己的全连接层与输出层到网络搭建最终模型。
3. 进行模型训练，按照训练集和验证集 4:1 分配数据。不对 InceptionV3 已经训练好的层级做训练，仅仅是利用预训练模型来导出特征权重，然后添加自己的全连接层与输出层到网络搭建最终模型，并针对后续添加的全连接层与输

出层做训练。

4. 通过调节 dropout 和 optimizer 如 SGD, Adagrad, Adam, Adadelata 等不同的参数来达到最低的 LogLoss 值。

5. 提交最优模型到 Kaggle, 获取测试集中的 LogLoss 分值对模型评估, 检查是否符合标准。

6. 进行数据探索并且可视化模型训练过程的准确率曲线, 损失曲线, 不同权重参数对准确曲线, 损失曲线的影响。

参考文献

[1] 图像识别中的深度学习

<https://www.cnblogs.com/nowornever-L/p/6392577.html>

[2] Keras Document

<https://keras.io/>

[3] Christian Szegedy, Google Inc., Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna University College London, Rethinking the Inception Architecture for Computer Vision

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf

[4] Lisa Torrey and Jude Shavlik, University of Wisconsin, Madison WI, USA

<http://ftp.cs.wisc.edu/machine-learning/shavlik-group/torrey.handbook09.pdf>