机器学习纳米学位
开题报告

叶剑 Udacity

2018-03-08

一、项目背景

猫狗大战，之前是 2013 年 Kaggle 的一道竞赛题目，要求写一个算法区分图片里面是猫还是狗，是图像识别问题中典型的一种。图像识别问题（CAPTCHA 或 HIPS）可以用来区分人类和机器，以达到减少垃圾邮件，网站密码攻击等目的。对于图片里面是猫还是狗的问题，人很容易识别，而机器识别起来却比较困难。究其原因，人类从出生之日起，就有无数次机会看到猫或狗，从而区分它们；但是机器没有这种多次识别猫或狗的经验，所以无从识别。 但是如果机器能够像人类一样，能够找到海量的猫或狗的照片加上一个智能的算法训练，并一一识别区分它们，经过训练后的机器理论上应该也和人类一样，能够识别图片里面的猫和狗。早在 2007 年就已经有论文[6]指出机器识别图片中的猫或狗正确率超过 80%。而 kaggle2013 年的竞赛 kaggle Public Leaderboard [7]上面，榜首机器识别图片中的猫或狗正确率更是达到了 98.533%。所以从理论和实践都说明机器学习来区分图片里面是猫还是狗的问题，应该并且能够得到解决。利用图像识别来区分人和机器已经变得不再安全了。

二、问题描述

该项目要求给出一张彩色图片，机器通过使用深度学习方法识别这张图片是猫还是狗，是一个二分类问题。识别图片是猫还是狗的 score(log loss)要在 kaggle Public Leaderboard[7] 前 10%，也就是 score 要低于 0.06，而且每次给出不同的猫或狗的图片，score 都达到上面的要求。

三、数据或输入

该项目数据集可以从 kaggle 上下载。Dogs vs. Cats

通过观察，训练数据包括 25000 张猫和狗的图片，这个训练集的所图面的文件名都含有 cat 或 dog 的标签，前面一半为 12500 张猫的图片，后面半为 12500 张狗的图片，而且图片的大小是不一样的，而且很可能有异常数据，就是既不是猫也不是狗的图片。

对于图片大小不一，加载图片时把图片缩放为统一的 299x299 的图片。

对于异常数据的检测，可以使用 ImageNet 提供的预处理模型 InceptionV4[3]、Xception[2]和 ResNeXt[1]等进行检测，取并集，然后将检测到的异常在训练数据里面移除，异常数据的移除可以提高算法模型训练的效果。

训练数据用来训练算法模型，分别取 1000 张猫的照片和 1000 张狗的照片作为验证集，其余 23000 张图片打乱顺序作为验证集。训练集和验证集的比例大概(之所以大概因为有移除小部分异常图片)为 23:2。

通过观察，测试数据包括 12500 张图片，里面的所有图片以数字 id 命名。测试数据用来验证训练后的模型的预测准确率，也就是说，对测试集的每一张图片，要预测图片里面是一条狗的概率。

四、解决方法描述

首先，要对训练数据进行必要的预处理，然后建立卷积神经网络模型，用预处理后的训练数据对模型进行训练，然后用测试数据对模型进行预测并评估模型的准确性。

五、评估标准

对模型的评估方案使用 score(log loss)，score(log loss)越低，方案越好，目标是 score(log loss)低于 0.06

Score(log loss)的计算方式如下：

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

n 是测试集中的图片数，$\hat{y}_i$是第 i 张图片预测为一条狗的可能性，yi 是测试集的正确结果，yi=1 代表图片是一条狗，yi=0 代表图片是一只猫。Log（）是自然对数（底数为 e 的对数）。

提交文件只有 2 列，图片 id 和图片为狗的概率，格式如下：

```
id,label
1,0.5
2,0.5
3,0.5
...
```

六、基准模型

该项目最终目的是根据训练对测试集数据进行分类，是一个基于监督学习的二分类问题，基于图片分类的特殊性，一般分类算法并不适应，目前图片分类问题运用最广的是卷积神经网络 CNN，CNN 是一种空间上参数共享的网络，使用共享权重的卷积层代替一帮的全连接层，通过卷积操作，逐渐挤压空间的维度，同时不断增加深度信息，深度信息体现复杂语意，最终用一个分类器，实现图片分类。 本项目使用 CNN 作为基准模型。在 CNN 的基础上，运用现有的成熟模型 ResNe50 等，进行迁移学习，并不断对模型进行优化，直到模型结果达到基准阈值 0.06.

七、项目设计

下载好训练和测试数据后，计划通过以下步骤解决该问题：

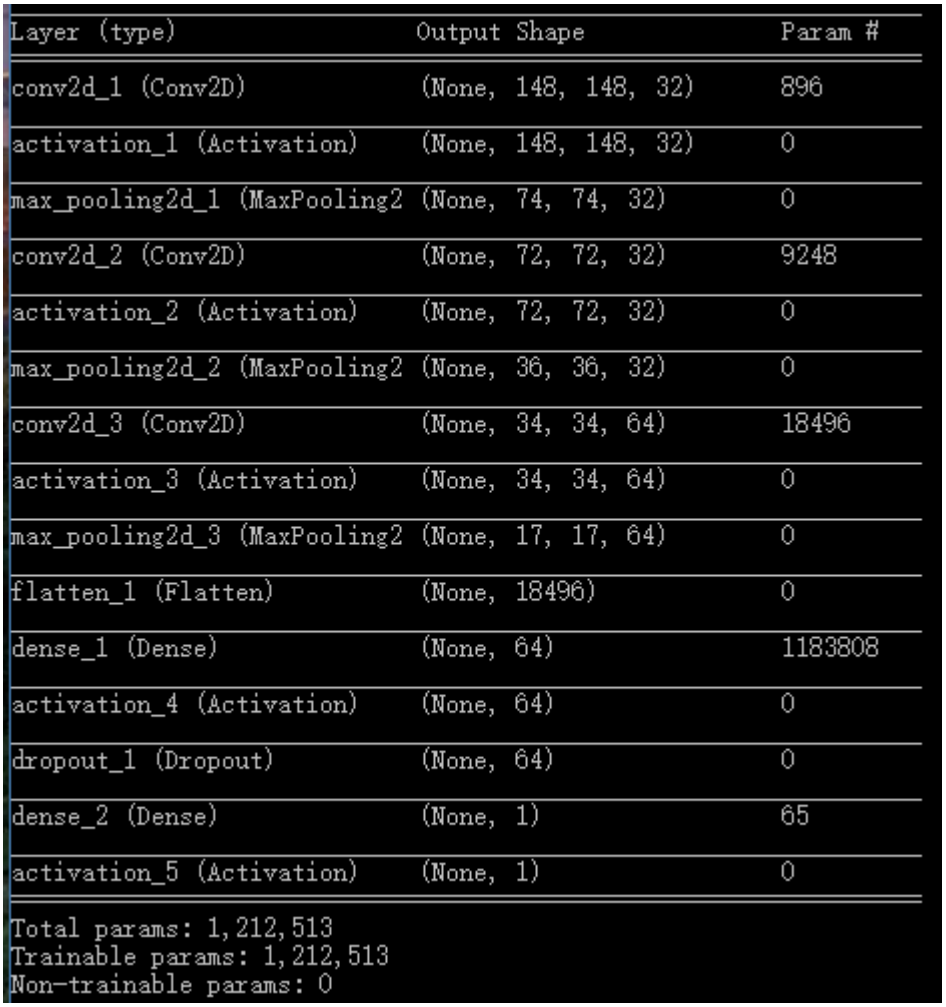
1. 数据预处理

   训练集的图片猫狗各 12500 张，文件名是以 type.num.jpg 这样的方式命名的，比如 cat.0.jpg，dog.0.jpg 等。 打算从训练集中分别抽取一部分猫狗图片作为验证集，比如

猫 1000 张，狗 1000 张，放到一个独立的验证集目录。使用 keras.preprocessing.image.ImageDataGenerator 进行随机转换和标准化操作，使用.flow_from_directory()训练集和验证集各自的文件夹中直接从 jpg 生成批量的图像数据及其标签。

2. 模型搭建

开始使用 keras.models.Sequential 搭建一个小型的卷积神经网络，包括卷积、relu 激活、最大池化，全连接层和输出层等。模型 summary 如下：



Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 148, 148, 32)	896
activation_1 (Activation)	(None, 148, 148, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 74, 74, 32)	0
conv2d_2 (Conv2D)	(None, 72, 72, 32)	9248
activation_2 (Activation)	(None, 72, 72, 32)	0
max_pooling2d_2 (MaxPooling2	(None, 36, 36, 32)	0
conv2d_3 (Conv2D)	(None, 34, 34, 64)	18496
activation_3 (Activation)	(None, 34, 34, 64)	0
max_pooling2d_3 (MaxPooling2	(None, 17, 17, 64)	0
flatten_1 (Flatten)	(None, 18496)	0
dense_1 (Dense)	(None, 64)	1183808
activation_4 (Activation)	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65
activation_5 (Activation)	(None, 1)	0

Total params: 1,212,513
Trainable params: 1,212,513
Non-trainable params: 0

3. 模型训练

使用 modle.complie binary_crossentropy 和数据准备中生成的图像数据　来训练模型，epochs 先设置小一点 10，bache_szie=16

使用 keras.models.Sequential 搭建的模型训练效果可能并不理想，运用迁移学习，使用 keras.applications.resnet50.ResNet50 建立 50 层残差网络模型，预先在 imagenet 数据集上进行训练，该模型已经学到了与我们分类问题相关的特征，在该模型的基础上，我们可以得出更好的结果，保存权重到本地文件。

4. 模型调参

   实例化 keras.applications.resnet50.ResNet50 的卷积并加载其权重

   将先前定义的全连接模型添加到顶部，并加载其权重

5. 模型评估

   Model.fit_generator 进行模型评估

6. 可视化

   基于前面的模型构建和训练结果，打算用 flask 做 web 展示。

参考文献

[1] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. arXiv:1611.05431
[2] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. arXiv:1610.02357
[3] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. arXiv:1602.07261
[4] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575
[5] Convolutional Neural Networks (CNNs / ConvNets). (Web)
[6] Philippe Golle. Machine Learning Attacks Against the Asirra CAPTCHA. (Web)
[7] kaggle Public Leaderboard. (Web)