

猫狗大战

机器学习纳米学位

叶剑

2018-05-26

目录.....	2
1. 项目定义	2
1.1 项目概述	2
1.2 问题陈述	2
1.3 评价指标	3
2. 分析	3
2.1 数据描述	3
2.2 算法与方法	5
2.3 基准指标	5
3. 方法	5
3.1 数据预处理	5
3.2 实施	6
3.3 改进	7
4. 结果	7
4.1 模型评估与验证	7
4.2 方案是否足以解决问题的理由	8
5. 结论	9
5.1 自由形态的可视化	9
5.2 思考	10
5.3 改进	11
参考文献	11

目录

1. 项目定义

1.1 项目概述

猫狗大战，之前是 2013 年 Kaggle^[4]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。

1.2 问题陈述

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，是一个监督学习的二分类问题。输入图片像素值向量 x ，输出样本 x 表示狗的概率 y ，那么样本表示猫的概率就是 $1-y$ 。

图片数据向量表示猫或狗的特征并不容易提取，而且不容易让人理解，人类本身也无法准确的描述分辨猫或狗的核心特征。但是如果不提取特征，以整个图片向量作为特征向量，以图片像素向量的高维度性，使用传统的机器学习方法将带来维度灾难，表现在对样本要求的指数级别增长和对模型规模和计算量的指数级别增长上。

针对维度灾难，卷积神经网络是一种空间上共享参数的神经网络，本质上也是一种深度神经网络，他使用共享权重的卷积层代替了一般的全连接层。卷积神经网络总的想法是形成一个金字塔结构，金字塔部是一张大而浅的图片，仅仅包括红绿蓝 3 色通道的深度，通过

卷积操作逐渐压缩空间的维度，同时不断增加深度信息，使深度信息大体可以表示出复杂语义，在金字塔顶端，定义一个分类器，所有空间信息被压损成一个表示，仅映射到图片内容的参数被保留。基于卷积操作的卷积神经网络可以使用较少的输入参数和计算量得到较高的准确率。

图像分类问题的代表性数据集位 ImageNet，从超过百万张图片的训练数据中学习识别 1000 中图像分类，包含猫和狗等动物。通常情况下，在 ImageNet 上表现优异的模型，对其他图像分类问题具有良好的泛化性能。由于猫狗大战是 ImageNet 的子问题，适合使用迁移学习方法，使用与训练模型在 ImageNet 中学到的经验参数表示猫狗图片的低维特征向量，并对这些特征向量进行分类。

1.3 评价指标

二分模型输出单元通常使用 Sigmoid 激活函数，损失函数使用对数损失函数，计算 log loss 方法如下：

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

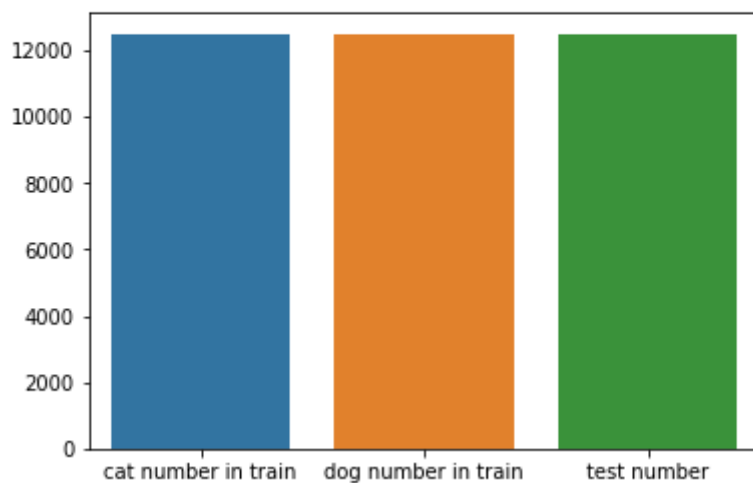
n 是测试集中的图片数， \hat{y}_i 是第 i 张图片预测为一条狗的可能性， y_i 是测试集的正确结果， $y_i=1$ 代表图片是一条狗， $y_i=0$ 代表图片是一只猫。Log () 是自然对数（底数为 e 的对数）。对模型的评价使用 log loss，log loss 越低，方案越好，模型在测试数据集的分类结果可以上传到 Kaggle 进行评价，排名。Kaggle 会给出分类结果的 score 也就是 log loss，我们依据 Kaggle 给出的 score 评价模型的好坏。

2. 分析

2.1 数据描述

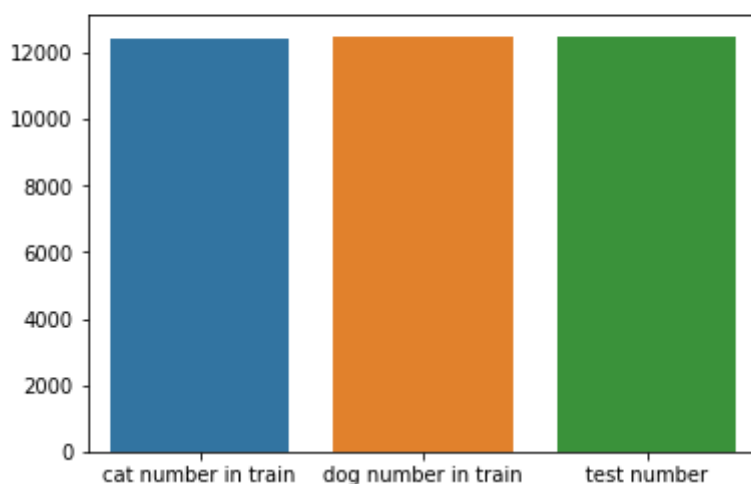
该项目使用了 Kaggle 竞赛提供的数据集，训练数据包括 25000 张图片，猫狗各占一半，在文件名中标注了该图片是猫还是狗。测试集包括 12500 张图片，没有标注类别，图片以图片数作为 ID 命名。具体图像分布如下图：

训练集总量: 25000
训练集猫的数量: 12500
训练集狗的数量: 12500
测试集总量: 12500



图片数量庞大, 虽然大部分图片数据是正常的猫狗图片, 但是仍然存在许多非猫非狗的异常样本, 这里可以使用预训练模型的分类型模型对训练集进行初步分类, 识别非猫非狗的异常数据, 然后删除异常数据, 在提出异常数据的基础上进行模型训练。这里只会对训练集进行异常数据清理, 测试集数据不在清理范围内。清理异常数据后数据分布如下:

训练集总量: 24933
训练集猫的数量: 12444
训练集狗的数量: 12489
测试集总量: 12500



2.2 算法与方法

该项目基于 Python3.6，使用基于 Tensorflow 的 Keras API 进行模型实现与训练。TensorFlow 是 Google 提供的深度学习开源框架，主要完成了图的梯度计算和反向传播过程，使开发者可以专注与模型搭建与训练。支持 GPU 加速计算。GPU 架构计算性能远超 CPU，使用 NVIDIA GPU 加速计算，还需要安装 CUDA 以及 cnDNN。Keras 对 TensorFlow 进行了进一步封装，实现了常用的层次结构，调用接口易于使用，并提供了一些搭建好的预训练模型，以提高效率，避免重复造轮子。由于基于 GPU 的计算环境本地搭起来费时费钱，该项目使用 AWS 提供的云服务器 p2.xlarge 实例，支持 GPU 但是也不太贵的那种。该项目使用 keras 提供的 ImageNet 数据集预训练模型 InceptionV3^[1]进行迁移学习，通过添加全局平均池化层、分类器等构建新的模型，通过调整模型参数不断优化模型。

2.3 基准指标

该项目定义基准指标为 Kaggle 给出的 score (log loss) 低于 0.06，在 kaggle public leaderboard 上位列前 10%。该项目不定义基准模型，评价只与基准指标对比。

3. 方法

3.1 数据预处理

由于训练集数据里面有一些非猫非狗的异常数据，或者一些很模糊或背景复杂的猫或狗的图片，这些数据测存在对模型的训练还是有一定的影响的，去掉这些异常数据有利于提高模型的预测准确性。

另外由于我构建的模型是使用迁移学习，基于预训练模型 InceptionV3，InceptionV3 默认输入的分辨率是 299*299，本项目会把所有的训练集和测试集中图片尺寸都设为 299*299。

最后，由于 InceptionV3 的权值是由 TensorFlow 训练得来的，所以使用 inception_v3.preprocess_input 把图片数据缩放到 (-1, 1) 的范围内。

3.2 实施

在进行模型训练之前，会把训练集中的数据混洗切分，把 train 文件夹中的数据随机分成训练集和验证集 2 部分，比例为 8:2。切分方法是 `sklearn.model_selection.train_test_split`。设置 `random.state` 保证每次切分效果一样。

然后运用 keras 上的预训练模型 InceptionV3 构建一个新的模型用于训练和预测，新的模型结构如下图：

batch_normalization_187 (BatchN	(None, 8, 8, 384)	1152	conv2d_187[0][0]
conv2d_188 (Conv2D)	(None, 8, 8, 192)	393216	average_pooling2d_18[0][0]
batch_normalization_180 (BatchN	(None, 8, 8, 320)	960	conv2d_180[0][0]
activation_182 (Activation)	(None, 8, 8, 384)	0	batch_normalization_182[0][0]
activation_183 (Activation)	(None, 8, 8, 384)	0	batch_normalization_183[0][0]
activation_186 (Activation)	(None, 8, 8, 384)	0	batch_normalization_186[0][0]
activation_187 (Activation)	(None, 8, 8, 384)	0	batch_normalization_187[0][0]
batch_normalization_188 (BatchN	(None, 8, 8, 192)	576	conv2d_188[0][0]
activation_180 (Activation)	(None, 8, 8, 320)	0	batch_normalization_180[0][0]
mixed9_1 (Concatenate)	(None, 8, 8, 768)	0	activation_182[0][0] activation_183[0][0]
concatenate_4 (Concatenate)	(None, 8, 8, 768)	0	activation_186[0][0] activation_187[0][0]
activation_188 (Activation)	(None, 8, 8, 192)	0	batch_normalization_188[0][0]
mixed10 (Concatenate)	(None, 8, 8, 2048)	0	activation_180[0][0] mixed9_1[0][0] concatenate_4[0][0] activation_188[0][0]
global_average_pooling2d_2 (Glo	(None, 2048)	0	mixed10[0][0]
dropout_2 (Dropout)	(None, 2048)	0	global_average_pooling2d_2[0][0]
dense_2 (Dense)	(None, 1)	2049	dropout_2[0][0]
Total params: 21,804,833			
Trainable params: 6,075,585			
Non-trainable params: 15,729,248			

该模型在不含池化层的 InceptionV3 基础上，添加一个全局平均池化层，然后添加 Dropout 层，最后添加 sigmoid 分类器。

模型锁住了 InceptionV3 前 280 层，单是对基础模型 InceptionV3 剩余的层进行训练。模型使用的优化函数是 `keras.optimizers.Adam`

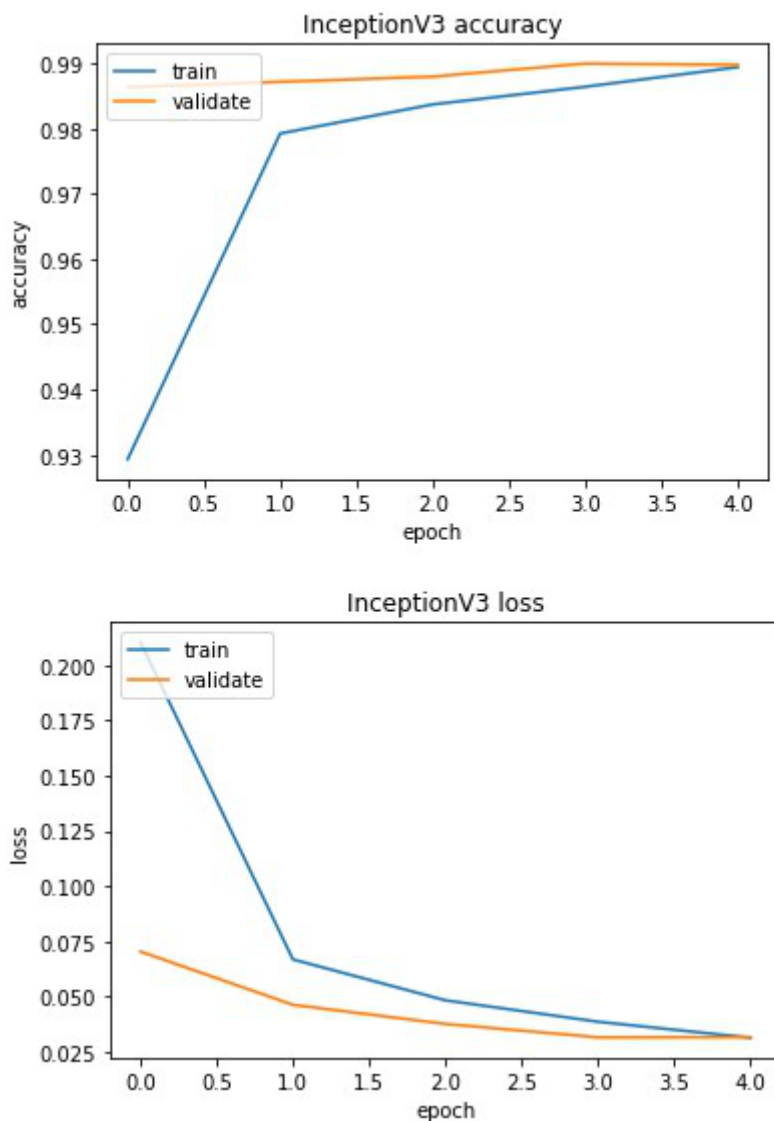
3.3 改进

为了是模型的 log loss 尽可能的低，对该模型进行了多次尝试，模型的优化器学习率最终调整为 $lr=1e-5$ ，每次更新后的学习率衰减值为 $decay=1e-6$ ，模型训练使用 keras callbacks ModelCheckPoint 保存模型权重数据，并使用 EarlyStopping 但验证集的 log loss 不再下降时自动终止训练。每次训练 128 各样本，总共训练 8 个周期。

4. 结果


4.1 模型评估与验证

根据之前定义的基准阈值，模型在测试集上面的预测结构要在 kaggle 上的 Public leaderboard 上进 10%，也就是说 log loss 要低于 0.06. 通过训练过程中的模型预测图可见，模型的稳定性很好，模型的训练集和验证集准确率都平稳收敛于 0.99 左右，log loss 收敛于 0.035 左右。模型并没由出现过拟合问题。模型训练预测图如下：



4.2 方案是否足以解决问题的理由

把模型对测试集的预测结果上传 kaggle，得到分数为 0.04947，低于我们之前定义的基准阈值 0.06，也就是说模型达到并超过了之前定义的基准。Kaggle 排名进入前 10%。这个结果证明，该模型能够完全胜任猫狗图像识别问题，准确率很高。



Dogs vs. Cats Redux: Kernels Edition

Distinguish images of dogs from cats

1,314 teams · a year ago

[Overview](#) [Data](#) [Kernels](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [My Submissions](#) [Late Submission](#)

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
predict002_inceptionV3.csv	just now	0 seconds	0 seconds	0.05142

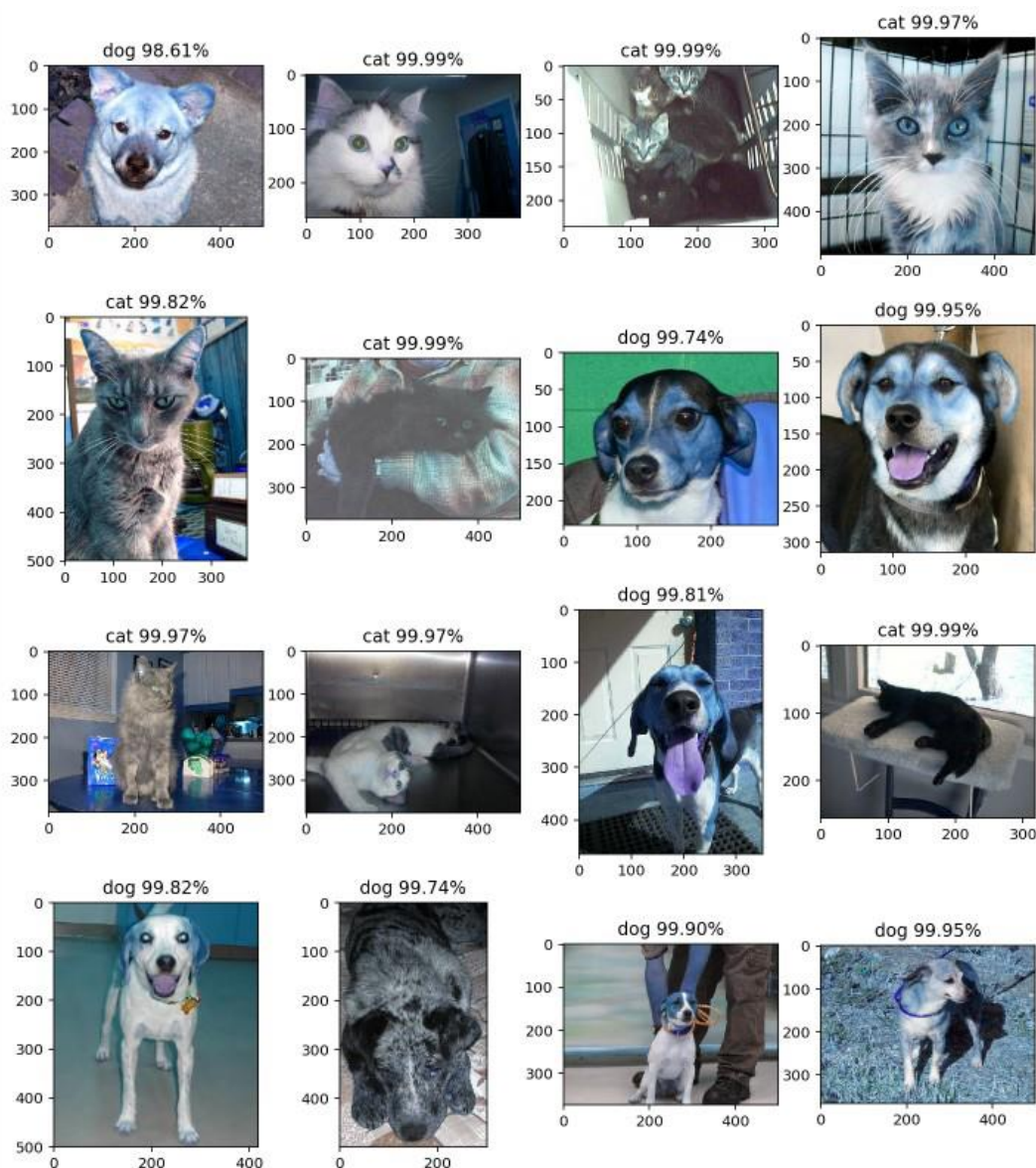
Complete

[Jump to your position on the leaderboard](#) ▼

5. 结论

5.1 自由形态的可视化

从测试集中随机挑选 30 张图片，然后用该模型进行预测，查看对图片的类别预测和图片预测概率，所有图片类别预测准确，而且每一张图片预测概率都接近 100%。



5.2 思考

该项目要求识别图片是猫还是狗，说实话，图片的参数多而且非常不易于理解，好在 `keras`、`sklearn` 等提供了强大的 API 支持，使这一任务变得简单许多。

该项目完成经历了数据获取、异常数据删除、训练数据混洗切分、图片的大小调整，入参范围调整，引入预训练模型 `InceptionV3`，基于 `notop` 的预训练模型构建一个新的模型，微调模型参数，在新模型上进行训练，预测测试数据等一系列过程。

该项目最终需要借助 `AWS` 的云服务器 `GPU` 去跑，所以在环境配置设置上花了至少由一周的时间，项目的实际开发时间在 2 周左右吧，实际上该项目由于工作原因我本来已经中断 2 个多月没有弄了，眼看要过了毕业 `deadline`，后来被延期一个月，这才有时间坚持到现在，

看来坚持到底，确实是达到目标的不变真理。

5.3 改进

该模型虽然达到了 kaggle 的前 10%，但是远远不是最好的成绩，所以在该猫狗分类问题上，该模型还有许多改进的地方。

可以使用 K Fold 验证法，消除部分混洗、分割造成的偶然性。

使用多模型融合，相当与不同的模型从不同的角度去预测，综合的预测结果一般会优于单模型预测

参考文献

- [1] [Rethinking the Inception Architecture for Computer Vision](#)
- [2] keras 中文文档 ([web](#))
- [3] Convolutional Neural Networks (CNNs / ConvNets). ([Web](#))
- [4] kaggle Public Leaderboard. ([Web](#))