

Machine Learning Engineer Nanodegree

Capstone Proposal

Chen Yifan

July 27th, 2018

Domain Background 项目背景

计算机视觉是一门研究如何使人工智能系统从图像中‘感知’的科学，即利用计算机代替人眼对目标进行识别、跟踪和测量，并进一步做图像处理，用计算机处理成为更适合人眼观察或传送给仪器检测的图像。

作为一门学科，计算机视觉可以看作是研究如何使计算机从图像或视频中获取高级、抽象的信息的科学；计算机视觉包含如下一些分支：画面重建，事件监测，目标跟踪，目标识别，机器学习，索引创建，图像恢复等

计算机视觉需要解决的经典问题是如何判定一组图像数据中是否包含某个特定的物体，图像特征或运动状态。通过从图像中提取各种复杂的特征，如线，边缘形状等，机器能判定鉴别某个物体的类别

Problem Statements 问题描述

《Dogs vs. Cats》是kaggle的一个竞赛项目，目标是建立一个模型，将给定的图片分辨为猫或狗。这是一个图像分类问题，是典型的计算机视觉问题。通过大量已经标注为猫或狗的图片对模型进行训练，用训练好的模型预测未知的图片，模型的输出为图片是狗的概率，概率为1.0表示狗，0表示猫

Datasets and Inputs 数据和输入

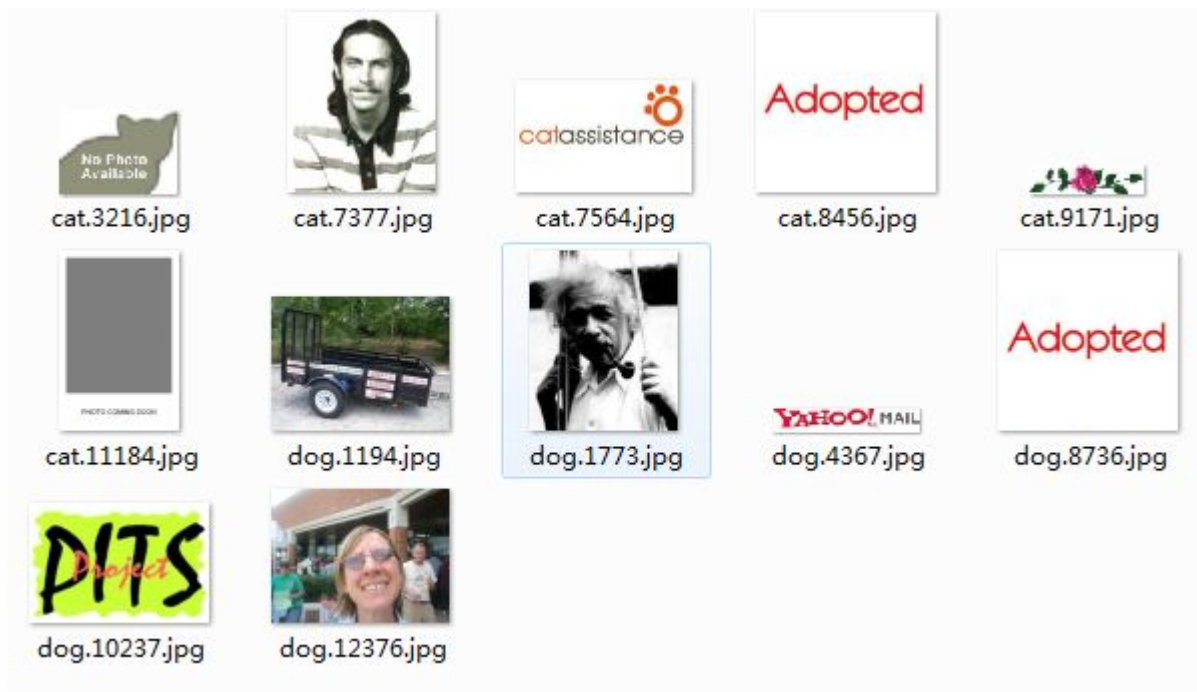
本项目采用Kaggle竞赛项目《Dogs vs. Cats Redux: Kernels》的数据集，这个数据集中的图片是从真实世界采集的猫和狗的图片，图片中猫和狗品质多样，颜色和姿态各异，背景多变。

图片分为train和test两个数据集；train中共有25000张图片，其中猫和狗各有12500张；所有图片都已经在文件名中标注为猫或狗，如cat.xxx.jpg, dog.xxx.jpg；train中的图片将被分为训练集和验证集用来对模型进行训练及验证

test中共有12500张猫狗的图片，test中所有图片均没有标注；test用来测试模型的性能

异常值的存在会让训练误差增大，在训练模型前对训练集的异常值进行清理是很有必要的；使用在ImageNet上有着很高准确率的预训练模型对训练集的图片进行预测，通过比较预测结果与图片的标注类型是否一致，即可以筛选出异常值，异常值有如下几种类型：

- 图片既不是猫也不是狗



- 图片标注为猫但实际是狗，或标注为狗但实为猫



Solution Statement 解决方法描述

如今，深度学习几乎成了计算机视觉研究的标配，因此本项目也计划采用深度学习模型作为解决方案。原因有，一是深度学习在识别精度方面高出传统算法许多；二是深度学习算法通用性很强，在传统算法里，针对不同的物体，需要定制不同的算法，而深度学习在不同的物体检测上都可以取得很好的效果；三是深度学习获得的特征有很强的迁移能力，例如在ImageNet上学习到的特征在场景分类上也能取得很好的效果。基于深度学习模型的通用性，我计划引用在ImageNet上取得很好效果的模型来对猫狗进行分类；使用基于tensorflow后端的Keras来搭建模型。Keras的Applications模块提供了带有预训练权重的深度学习模型，这些应用于图像分类的预训练模型，其权重训练自ImageNet[1]。模型输出的是各物体的预测概率，如输入一张狗的图片，模型的输出结果是属于某个狗品种的概率，而本项目预测的是属于狗还是猫的概率，不用理会猫狗是何品种，所以本方案仅使用预训练模型从图片中提取特征，再将提取的特征输入自定义的深度学习模型进行训练和预测，最终的输出为图片属于狗的概率

Benchmark Model 基准模式

Kaggle竞赛项目《Dogs vs. Cats Redux: Kernels》排名前10%的成绩达到了0.06114, 排名100的成绩是0.05629, 最好成绩是0.03302; 我希望最终的成绩能进入排名前100以内, 即 logloss 分数小于 0.056.

Evaluation Metrics 评估标准

本次猫狗大战(Dogs vs. Cats Redux: Kernels Edition)中使用logloss[2]作为评价指标, 分数计算如下:

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

LogLoss是一个连续值, 取值范围 是0至无穷大, 相比Accuracy, LogLoss能对模型提供更细致的评价. 在深度学习成为主流的今天, 模型对图像分类的准确率都非常高, 模型之间的性能差异较小, 使用LogLoss作为评价指标能以更细微的视角观察到模型性能之间差异

Project Design 项目设计

本项目计划使用如下策略: 使用ImageDataGenerator生成图像, 使用Keras提供的带有预训练权重的模型从图像数据中导出特征向量, 为了减少数据量以及避免过拟合, 将预训练模型的输出数据实施池化后再保存为文件; 构建自己的模型, 载入预训练模型导出的特征向量进行训练, 使用训练好的模型对测试集进行预测, 然后提交到kaggle.

项目实施的第一步是准备数据和对数据进行预处理; 数据来自kaggle上的一个竞赛(Dogs vs. Cats Redux: Kernels Edition), 训练集有25000张, 猫狗各占一半; 测试集12500张, 没有标是猫还是狗. 训练集中的每张图片都在文件名中标注为猫或狗, 图片是以 type.num.jpg这样的方式命名, 比如 dog.999.jpg. 对数据集进行预处理以便将图像转换成模型能用的图像数据, 并给训练集中的每个图像数据创建标签, 本项目将使用 Keras 的 ImageDataGenerator 将图像转换成模型能用的数据, 并为训练集的数据创建相应的标签. 使用 ImageDataGenerator 需要将不同种类的图片分在不同的文件夹中, 因此我们将数据集的路径结构做如下转换: 转换前:

```
|-- test [12500 images]
|-- train [25000 images]
```

转换后:

```
|-- pre-test
    |-- test [12500 images]
|
|-- pre-train
    |-- cats [12500 images]
    |-- dogs [12500 images]
```

二、提取特征向量; 使用带有预训练权重的模型, 从训练集和测试集中导出特征向量并保存为文件;;为了比较不同预训练模型的效果, 将使用 Top5准确率最高的3个预训练模型分别导出特征向量并保存

三、构建自己模型: 我计划使用基于tensorflow后端的Keras来构建模型, 首先载入从预训练模型中提取的特征作为输入张量, 然后依次加入Dropout层, Dense层和Activation层; Dropout层在训练过程中每次更新参数时按设置的概率随机断开输入神经元, 可有效防止过拟合; 全连接层Dense和Activation层实现操作: $\text{output} = \text{activation}(\text{dot}(\text{input}, \text{kernel}) + \text{bias})$, 其中kernel为本层的权值矩阵, bias为偏置向量, activation是逐元素计算的激活函数, 本项目要求的输出是图片是狗的概率, 因此本模型使用sigmoid为激活函数

四、是训练模型, 将验证集的大小设置为20%, 即从训练集中划分出20%的数据用来验证模型的性能

五、是对测试集进行预测, 将预测结果写入csv并提交至Kaggle

数据集下载链接: [Dogs vs. Cats Redux: Kernels Edition](#)

参考文献：

[1][Applications- Keras Documentation](<https://keras.io/applications/#applications>)

[2][Log Loss - Deep Learning Course Wiki](http://wiki.fast.ai/index.php/Log_Loss)