机器学习纳米学位

毕业项目开题报告

优达学城

2017 年 3 月 6 日

项目概述

首先，在这里我要先感谢 Viewer 对我的第一份 behavioral-cloning 的耐心和建议，第一次的提交后，我充分意识到了自己的不足，再经过考虑后，我决定将项目换成猫狗大战（dog_vs_cat）。

这个项目简单描述就是基于图片的猫狗分类，但这其中涉及的许多的领域，包括计算机视觉中的图像处理，深度学习中的卷积神经网络，以及其他与机器学习相关的邻域。

所以做这个项目可以提高我对深度学习的理解，也有利于我对计算机视觉的掌握，更不用说我现在正在学习的机器学习纳米学位[0]了。

本次的数据集来自 kaggle 上的猫狗大战项目[1]，详见引用。

问题陈述

在现实生活中，我们人类能够很清楚就能理解的东西，在计算机看来就不一

样了，比如我们能很清楚的看到外面草坪上有一只猫或一只狗，但这在计算机眼里就是一连串的 1 和 0，所以我在此要解决的问题就是让计算机'看到'猫和狗，并对猫和狗进行分类，所以这个问题也被一些大神戏称为猫狗大战，

要解决这个问题呢，首先要有数据，在机器学习中，数据是必不可少的东西，有了数据以后就可以构建一个可以解决此问题的模型，最后还需要强大的运算能力对模型进行训练。在训练结束后就是对我们的模型进行评估了，评估通过的话，我们的问题就解决了。

评价指标

此次项目的评价指标为猫狗预测概率值的 log loss。

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

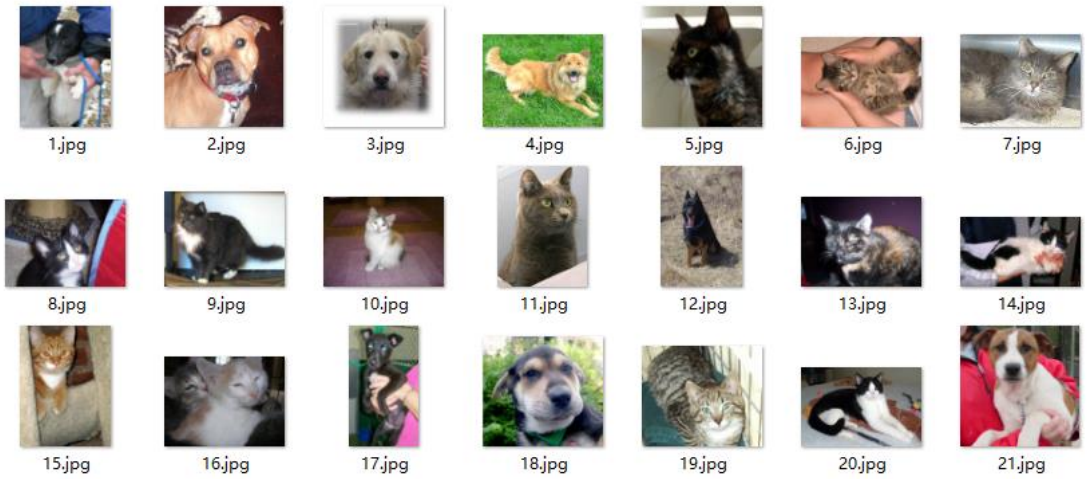
在这里，n 为测试集的图片数量，$\hat{y}_i$是图片预测为狗的概率值，y_i为 1 如果图片为狗，为 0 如果图片为猫，$log()$是以 e 为底的自然对数。Logloss 越小则越好。这个根据公式便可看出。

数据的探索

本次使用的数据在上面已经说了，是来自 kaggle 的猫狗大战数据集，它分为训练集和测试集，训练集有 25000 张图片，其中猫狗各占一半，

至于其准确度，我稍微看了一下，除了一些拍得比较差的有点难区分外，我还是没找到什么标记错误的。

而测试集有 12500 张图片，从下图就可以看出测试集的猫狗都是没有分类的，所以我也没有去数猫狗各有多少张了。



另外一些异常点，我在下面举几个例子，


这里有 4 条狗

 这个不知道是什么的东西被标记为狗，

 这只猫的镜头都被主人抢去了，

 这只猫太漂亮了，在整体数据里感觉不协调。

这些异常点比较少，不必针对他们进行格外处理，可以通过数据提升使其对整体的影响降到最低。

探索性可视化

对于数据集，有一个缺陷就是每张图片的大小不一，这是需要处理的。对于猫狗图片的一些探索性可视化，我实在找不到什么，我想到了把图像的 size 做一个图表，但又觉得没必要，所以此处没有可视化，不过你可以参考上面数据的探索。

另外一件重要的是文件名，因为数据预处理的时候，需要用到，从上面的图

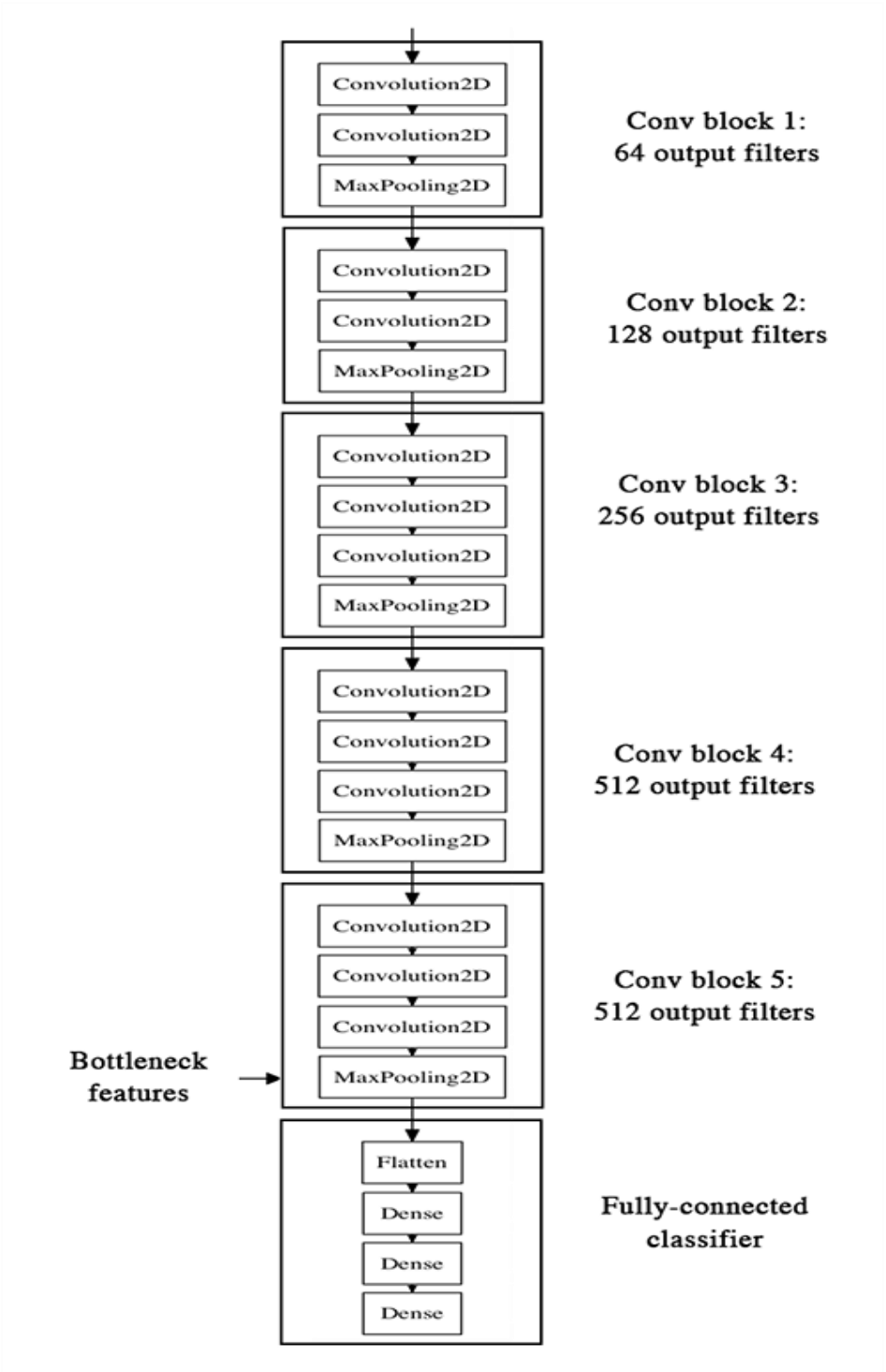
片我们可以看到训练集的文件名为‘cat/dog’+‘.’+‘number’+‘.’+‘jpg’的形式，我们就可以据此来进行标签分类，而测试集文件名就是简单的‘number’+‘.’+‘jpg’，它的标签是需要我们进行预测的。

算法和技术

此次项目我用到的算法主要是卷积神经网络[2]，所谓卷积神经网络，它也是神经网络的一种，由一个或多个卷积层和顶端的全连通层（对应经典的神经网络）组成，同时也包括关联权重和池化层，常用于处理图像，所以这次根据图像区分猫和狗用它再适合不过了。但其本质上还是属于机器学习中的监督学习，所以我们上面需要输入猫狗的图像和区分它们的标签这两种数据。得到数据之后，我们就对其进行卷积，卷积有不同 filter 大小，深度和它的 stripe 步长，我们增加增加它的‘深度’，再有池化，池化也有许多类型，如 maxpool，averagepool 等，卷积和池化都弄好后，我们就将它 flatten 拉平，最后经过全连接（即经典的 Y=wX+b）得出我们想要的结果。

另外我还使用了迁移学习[3]，所谓迁移学习呢，就是把已经与训练好的模型迁移到一个新的模型中去，因为大多数数据都存在一定相关性，所以运用已经训练好的模型的参数，这样就可以节约大量训练时间，避免从 0 开始训练。

对于此次分类，我用的 base_model 是 VGG16[4]，它是一个非常强大的卷积神经网络，详见引用，该模型的基本构架如下，



在这里我只保留它的卷积层，然后全连接是自己写的，作为我的基准模型。

如上图，VGG16 有 5 个卷积区，每个卷积区最后一个 max pool，共有卷积层 13 层， output filter 由 64 逐渐增加到 512，最后 flatten 后加上全连接。进行了这么多层的卷积，足以解决比猫狗分类大得多的问题，但是导师说得好，杀

鸡得用牛刀，这就是我选择它的原因。

基准模型

这次项目我用来衡量解决方案性能基准是提交 kaggle 猫狗大战比赛后由其给出的 logloss 值，就取 0.1 左右吧，0.1 已经是一个很合适的值了，我训练时看 logloss 为 0.1 就已经是 99%的准确率了，虽说这我有可能过拟合，但如果测试集，即 kaggle 给出的 logloss 有 0.1，那么也能一定程度代表我的准确度已经达到了 99%，所以就这么决定了。

设计大纲

一、 对输入的数据进行分析，根据其特征进行预处理

二、 构建一个基本模型的框架

三、 运用模型对数据进行训练

四、 对训练出来的模型，进行评估，并对预测进行可视化

五、 对模型进行完善，也可进行数据提升

六、 得出最佳模型，将结果提交 kaggle

七、 结束

资料及来源

【0】 udacity

【1】 kaggle

【2】 CNN

【3】 Transfer learning

【4】 VGG16