

机器学习纳米学位

猫狗大战

董建华 优达学城
April 11th, 2018

I. 问题的定义

项目概述

- 解决猫狗大战问题涉及到图片识别领域，需要用到深度学习、卷积神经网络等知识，并要借用到等成熟模型在 ImageNet 数据集上的结果，lecun 的 LeNet¹开创了 CNN 卷积网络，由于计算资源不允许没有普及，直到 Alexnet²出现，添加了 dropout 和 relu 结合现在计算资源的发达，CNN 发展速度也越来越快，Imagenet 是目前世界上图像识别最大的数据库，并举办了相应比赛来促进 CNN 模型构造理论的交流和发展，有很多优秀的模型都通过这个数据集得到了优秀展现从而得到普及，VGG³、Inception、ResNet⁴这三个模型都是出自 Imagenet 数据集比赛上的冠军模型。目前的在 Imagenet 上最好的模型是这几个：ResNet50、Inception v3⁵、Xception⁶、Inception-ResNet v2⁷等
- 项目的出发点：实现识别猫狗的最优模型，预测结果 loss 在 kaggle 上达到 0.04 以内。
- 相关数据集：Kaggle 竞赛上的数据集以及牛津大学的宠物数据集

问题陈述

此问题很符合二分类问题，因为模型预测结果只有两个：非猫即狗。

- 问题一：保证数据准确有效，不能是一些无关或者相关度小的图片
- 问题二：模型训练如果从零开始训练，数据集不够庞大
- 问题三：如何构造整个网络模型进行训练

评价指标

猫狗识别是二分类问题，输出层使用 sigmoid 函数，通过交叉熵来计算损失，相比于方差代价函数它可以克服因为使用 sigmoid 函数更新权重过慢的问题，最终和 kaggle 在的得分做比较。

$$C = \frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)]$$

y：数据真实值 a：模型预测值 n：测试样本数量

II. 分析

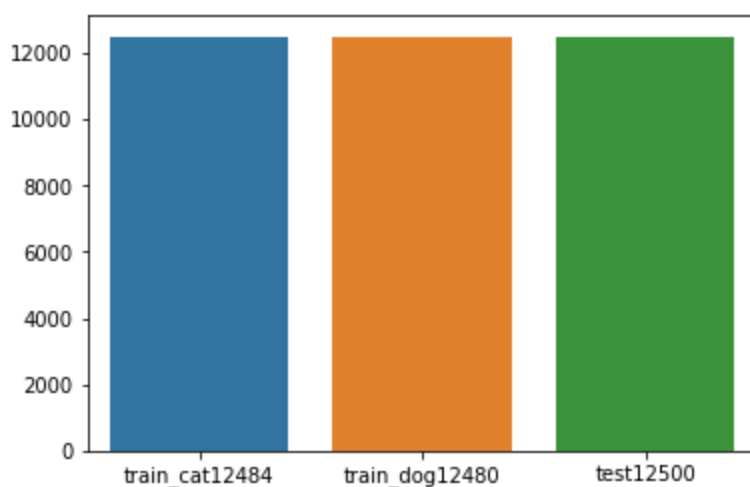
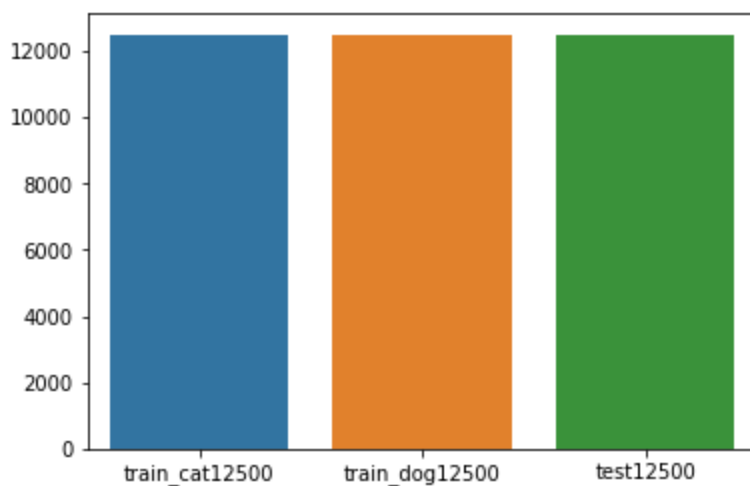
数据的探索

项目使用了 Kaggle 上的数据集，训练集一共有 25000 张猫狗的图片,其中猫狗照片比例一比一，以及测试集 12500 张，经过简单筛选可以看出训练数据中存在非猫非狗的图片、非实物猫狗的图片以及猫狗占比很小的图片，所以要对训

训练数据进行清洗，去除这些异常值。去除异常图片后训练集猫狗比例接近一比一，测试集保持不变。

探索性可视化

因为特征简单，训练集中只有两类，所以进行了简单可视化，横轴为训练集中猫的数量和狗的数量以及测试集的总数量。原始数据中训练集中猫的数量为 12500、狗的数量为 12500，测试集总数量为 12500；去除异常值图片后训练集中猫的数量为 12484、狗的数量为 12480，测试集总数量为 12500。



算法和技术

- 问题一：通过预训练模型清洗数据，筛除没有猫狗的图片、非实物猫狗的图片以及猫狗占比很小的图片
- 问题二：通过对相近模型的迁移学习预训练自己数据集提取特征，达到加速训练的目的
- 问题三：通过对 Keras 中已有的 Inception-ResNet v2 Xception Inception v3 等成熟模型进行迁移学习，将一张图片通过若干个模型提取出特征融合为一个特征向量，最后搭建全连接层进行训练。

基准模型

通过 kaggle 竞赛上的得分来衡量解决方案优劣，竞赛上的排名靠前得分多在 0.04 以内，所以使用 0.04 作为基准结果/阈值

III. 方法

数据预处理

由于训练数据中存在非猫非狗的图片、非实物猫狗的图片以及猫狗占比很小的图片，所以在对数据进行前处理之前会先进行清洗去除这些异常值，通过图片处理 resize 尺寸为 (299,299) ,来匹配用到 Keras 中三个预训练模型

(Inception-ResNet v2、Xception、 Inception v3) 分别进行异常值检测，接着在 Filter_data.ipynb 中，汇总异常图片剔除掉三者都不能识别的猫狗图片以及部分两个模型同时不能识别的图片，最后在 Get_gap.ipynb 中，将剩下的

猫狗图片生成软连接分成两个类别，再通过 Keras 中的图片预处理模块将训练集中 cat lable 归为 0，dog lable 为 1，并通过加载预训练模型权重提取特征后保存为 h5 文件，在 cat vs dog.ipynb 中，读取这些数据进行特征融合，在模型训练中选择用 0.2 的比例分割数据进行训练和评估。

最终筛选的图片：



执行过程

- 通过预训练模型对数据进行异常值清洗，使将来的预测模型更准确

- 数据建立软连接（排除异常图片），接着通过迁移学习提取图片特征，保存为数据文件
- 读取多个数据文件，融合多个模型的特征得到需要使用的预训练数据集，设置全连接层和 dropout 层搭建一个新的模型进行编译和训练
- 通过可视化，伪代码来分析训练过程

完善

介绍网络模型：

首先是 Resnet50 网络

它首次提出运用了“残差网络”，解决了传统的卷积层层叠网络当层数加深时，网络的表现越来越差的问题。这其中很大程度上的原因是因为当层数加深时，梯度消散得越来越严重，以至于反向传播很难训练到浅层的网络，“残差网络”的出现使得梯度更容易地流动到浅层的网络当中去。

接着说 inception V3 网络，

这一系列要围绕着 Going deeper with convolutions 说起：

Inception v1：提出了 Inception module 结构（ 1×1 ， 3×3 ， 5×5 的 conv 和 3×3 的 pooling 组合在一起），最大的亮点就是从 NIN（Network in Network）中引入了 1×1 conv，代表作 GoogleNet

Inception v2：在 v1 基础上加入了 BN 层，并用 2 个连续的 3×3 conv 替代 inception 模块中的 5×5 ，从而实现网络深度的增加，网络整体深度增加了 9 层，缺点就是增加了 25% 的 weights 和 30% 的计算消耗。

Inception v3：则是在 v2 的基础上，提出了卷积分解，将 7×7 分解成两个一维的卷积（ $1 \times 7, 7 \times 1$ ）， 3×3 也是一样（ $1 \times 3, 3 \times 1$ ），既可以加速计算（多余的计算能力可以用来加深网络），又可以将 1 个 conv 拆成 2 个 conv，使得网络深度进一步增加，增加了网络的非线性，精细设计了 $35 \times 35 / 17 \times 17 / 8 \times 8$ 的模块。增加了网络宽度，网络输入从 224×224 变为了 299×299

Xception (Extreme Inception)

在 Inception V3 的基础上提出的，基本思想就是通道分离式卷积（空间变换、通道变换），最终模型参数较 Inception V3 有微量的减少，精度有所提高。

最后是 Inception-ResNet-v2

Inception 模型的变体，结合了残差网络的理念，但比之前的 Inception V3 这些要深得多，这个架构比之前的前沿模型更加准确。

最初的模型：

使用 Inception-ResNet v2、Xception、Inception v3、三个模型进行特征提取并融合，然后调整了 optimizer、batch size 和 poach：

首先使用了 SGD 做优化器，

当 batch_size 设为 128、nb_epoch 设置为 100 的时候，val_loss 在 0.0106 趋于稳定，上传 kaggle 成绩 0.03616

当 batch_size 设为 64、nb_epoch 设置为 50 的时候，val_loss 在 0.0106 趋于稳定，上传 kaggle 成绩 0.03614

接着使用了 adadelta 优化器

当 batch_size 设为 128、nb_epoch 设置为 12 的时候，val_loss 在 0.0115 趋于稳定、上传 kaggle 成绩 0.03596

当 batch_size 设为 64、nb_epoch 设置为 6 的时候，val_loss 在 0.0112 趋于稳定、上传 kaggle 成绩 0.03591

最后使用了 Adam 优化器

当 batch_size 设为 128、nb_epoch 设置为 6 的时候，val_loss 在 0.0104 趋于稳定、上传 kaggle 成 0.03632

当 batch_size 设为 256、nb_epoch 设置为 9 的时候，val_loss 在 0.0105 趋于稳定、上传 kaggle 成绩 0.03612

综上、三个优化器的最佳成绩差异不大，SGD 训练较慢，adadelta 训练不稳定，所以 optimizer 最终选择了 Adam, batch_size 设置为 256, nb_epoch 设置为 9 作为模型参数。

另外的做了新的尝试：

增加了 ResNet50 预训练特征，尝试使用 ResNet50、Inception-ResNet v2、Xception、Inception v3 四个模型进行特征提取并融合，optimizer 继续使用 Adam，然后调整了 batch size 和 epoch，最终模型参数 batch_size 设置为 256, nb_epoch 设置为 6 的时候，val_loss 在 0.0110 趋于稳定，上传 kaggle 成绩为 0.03620，和三模型成绩相比差异不大。

最终的模型：

最终决定选择 Inception-ResNet v2、Xception、Inception v3 三个模型进行特征提取并融合，搭配一个全连接层，采用二分类，优化器用 Adam，batch_size 设置为 256, nb_epoch 设置为 9 作为模型参数。

IV. 结果

模型的评价与验证

最初的三模型融合下，三种优化器在 kaggle 上的最佳成绩差异不大，SGD 训练较慢，adadelat 训练不稳定，Adam 高效稳定。然后四模型结果据上描述没有改善，最终选择了三模型融合以及合适的参数，重新编译模型进行多次验证训练，Adam 的成绩均稳定在 0.036，同时我也尝试了改变三个模型特征的融合顺序，对结果没有影响。

Kaggle 成绩截图：

my-sub-mix-4-adam-256.csv 16 minutes ago by aidtnn	0.03620
my-sub-mix-3-adam-256.csv 34 minutes ago by aidtnn adam-256	0.03612
my-sub-mix-3-adam-128.csv 41 minutes ago by aidtnn update adam-128	0.03632
my-sub-mix-3-adam-128.csv 43 minutes ago by aidtnn adam-128	0.03672
my-sub-mix-3-sgd-64.csv an hour ago by aidtnn update sgd-64	0.03614
my-sub-mix-3-sgd-64.csv an hour ago by aidtnn sgd-64	0.03821
my-sub-mix-3-sgd-128.csv an hour ago by aidtnn SGD-128	0.03616
my-sub-mix-3.csv 7 days ago by aidtnn train by "gap_InceptionV3","gap_InceptionResNetV2","gap_Xception"	0.03591
my-sub-mix-3.csv 10 days ago by aidtnn train by "gap_InceptionV3","gap_InceptionResNetV2","gap_Xception"	0.03596

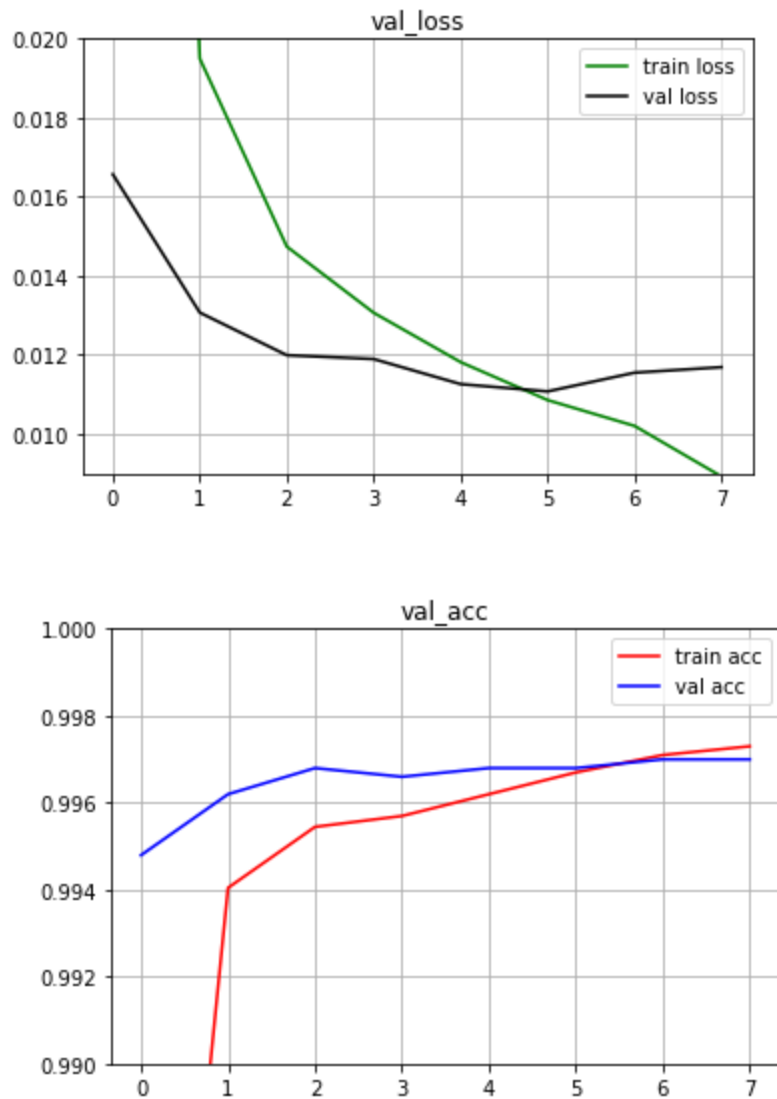
合理性分析

Kaggle 上排名第一的 loss 为 0.03302，排名第 15 的 loss 为 0.03994，我的最终结果是 0.03612，放在 kaggle 上可以排名第八，定义的阈值为 0.04，最终的结果好于设定的阈值，能解决项目问题。

V. 项目结论

结果可视化

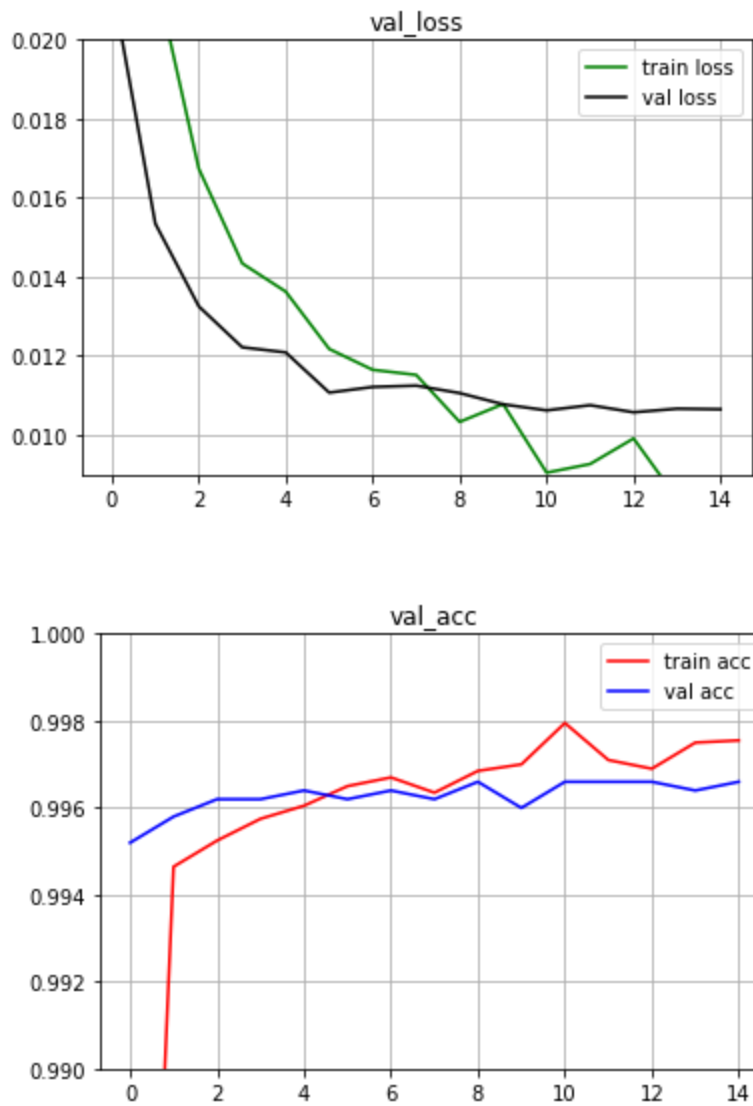
四模型融合 Adam 下的 loss 和 accuracy 曲线：



可以看出 val_loss 在 5-7 个 poach 趋于稳定，所以最终模型选择了 batch

size : 256 epoch : 6

三模型融合 Adam 下的 loss 和 accuracy 曲线：



可以看出 val_loss 在 9-15 个 poach 趋于稳定，所以最终模型选择了 batch size : 256 epoch : 9

随机测试了 20 张图片，并通过 open CV 和 grad-cam 生成了热力图，
可以从中看出，图片识别过程中模型重点寻找的是动物的面部，其次是毛发和尾巴等。

2591&cat 100.00%



4684&dog 97.45%



5098&dog 99.99%



3990&dog 100.00%



9073&dog 99.99%



954&cat 99.98%



101&cat 100.00%



5488&dog 99.99%



5271&dog 99.99%



9207&dog 99.81%



10467&cat 100.00%



1203&dog 99.95%



6493&dog 99.94%



984&dog 99.96%



4465&dog 99.93%



4610&cat 99.99%



11753&dog 99.96%



2869&cat 99.91%



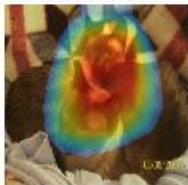
9473&dog 99.90%



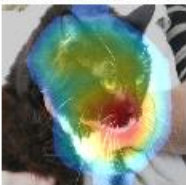
781&dog 99.97%



10834&cat 99.70%



12116&cat 100.00%



4554&dog 99.98%



11622&cat 99.99%



6754&dog 99.99%



3403&cat 99.98%



1572&dog 99.77%



8327&dog 99.52%



11523&cat 100.00%



11036&dog 100.00%



10466&dog 99.96%



6199&dog 99.92%



4963&dog 99.81%



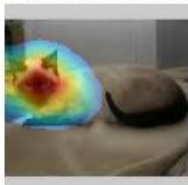
3837&cat 99.98%



1144&dog 99.69%



9596&cat 99.78%



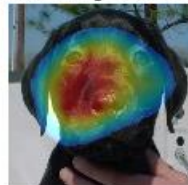
12118&cat 99.99%



816&dog 99.98%



3606&dog 100.00%



4623&dog 99.99%



对项目的思考

整体的解决方案就是通过预训练模型清洗数据，接着进行迁移学习提取特征和特征融合来加强训练，最后搭建模型训练，快速准确的生成解决项目问题的模型。有意思的地方是，清洗图片时会发现很多名人照片位于训练集中，困难的地方是选取不同的预训练模型进行数据清洗和特征提取。最终结果符合我对这个问题的期望，可以在通用的场景下解决这些类型的问题。

需要作出的改进

在模型搭建过程中，尝试增加了一个预训练模型，结果来看 loss 成绩差异不大，但达到最优模型的训练代数有了一定的减少，所以未来可以考虑减少或者增多预训练模型的个数并进行一些组合，或许能训练出更好的模型来，这个项目的执行流程也能应用到将来遇到的迁移学习相关的任务中去。

References:

- ¹ Lecun Gradient-based Learning Applied to Document Recognition 1998
- ² Alex ImageNet Classification with Deep Convolutional Neural Networks 2012
- ³ Karen Simonyan Very Deep Convolutional Networks for Large-Scale Image Recognition 2014
- ⁴ Kaiming He Deep Residual Learning for Image Recognition 2015
- ⁵ Christian Szegedy Rethinking the Inception Architecture for Computer Vision 2015
- ⁶ Francois Chollet Xception: Deep Learning with Depthwise Separable Convolutions 2016
- ⁷ Abhinav Shrivastava Beyond Skip Connections: Top-Down Modulation for Object Detection 2017