猫狗大战（最终篇）

骆炜

2018 年 6 月 23 日

目录

1 项目概览 **1**
1.1 目前现状 2

2 数据研究 **3**
2.1 预处理筛除 3
2.2 数据增强 4

3 算法与方法 **6**
3.1 生成特征向量 6

4 测试结果 **7**
4.1 测试结果 7
4.2 识别可视化 8
4.3 需要作出的改进 8

5 项目附件及其说明 **9**

1 项目概览

本项目基于 Kaggle 公开训练的和测试数据集实现对图像中猫狗进行图像识别。本项目涵盖数据处理，模型选择和搭建以及最终测试等主要步骤。本项目所涉及的为典型的二分类问题，需要通过训练集的图片对自己所设计的或是改进的神经网络进行训练，而后通过 Kaggle 官方的评价系统评判所选模型的准确性。相比于对目标图像的判别，这个项目更重视分类的准确性，需要达到 Kaggle 评分标准（loss 不高于 0.06127）。

1.1 目前现状

在图像分类上神经网络有着丰富的历史。从经典的 LeCun 于 1998 年提出的 LeNet5 开始 [1] ，计算机对于图片的识别能力得到了一步一步地提升。虽然受限于当时的计算能力和计算方法，该网络和现今上百层的神经网络相比实在是非常简略，但是该网络为后续的图像识别提供了一个基础的样式和模板。其经典的结构如图1所示。

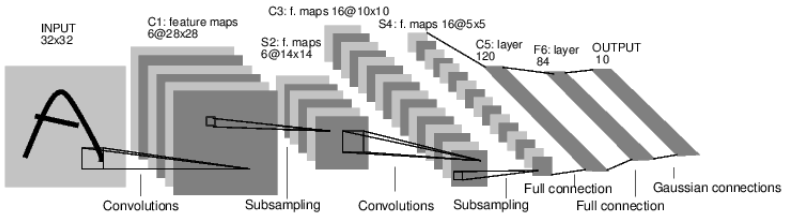

图 1: LeNet5 经典网络结构

后人在这个网络结构的启发下，不断提高对于图片的分类和识别的能力。传统结构容易出现，训练难度大，难以收敛，容易过拟合等问题。近些年来，随着图像数据的丰富，为研究学者提供了充足的带标记的训练样本，进而可以训练越来越复杂而性能强大的神经网络。此外，传统的网络在深度不断加深的情况下，容易出现梯度消失、准确率不高等问题。近些年的研究人员同时也不断提出新的网络结构，例如谷歌提出的 Inception V3/V4 [2] 和 Xception [3] 等等。其中 InceptionV3 和 V4 的结构如图2和3所示，V4 版本利用 TF 等工具，能更方便地进行内存优化训练，相比 V3 有进一步提高（虽然不是很明显）。

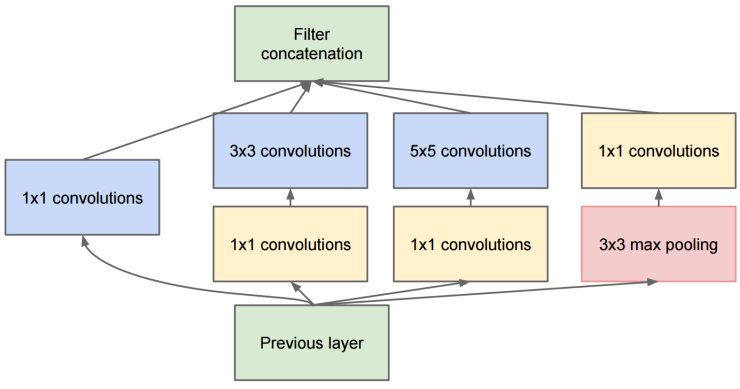

图 2: Inception V3 单元结构

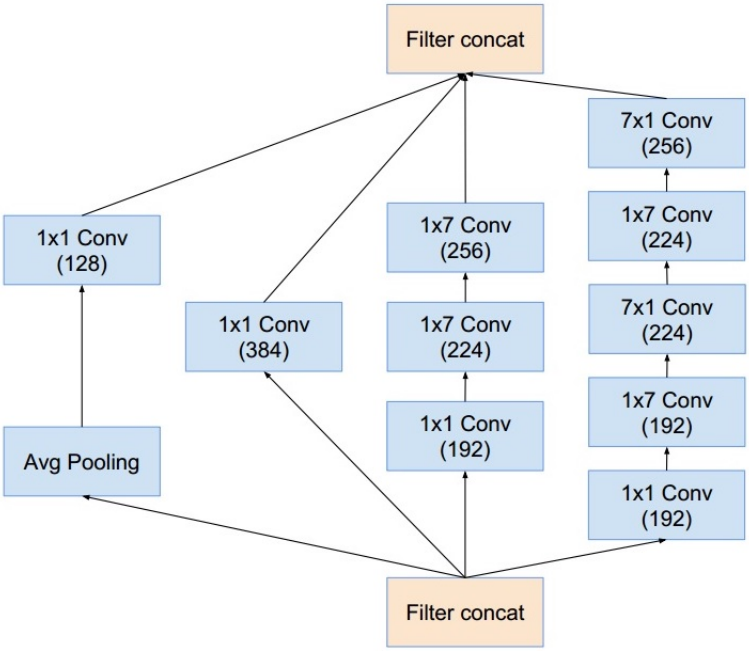

图 3: Inception V4 单元结构 (B 模块)

2 数据研究

本项目有 Kaggle 提供了 25000 张训练照片和 12500 测试照片。对于数据的研究主要针对训练照片展开，而测试照片将会原封不动保留到测试步骤。数据的总量较少，因此难以独立完整地训练复杂的卷积神经网络。常见的处理流程包括，数据预筛选，数据增强等。这些处理流程的目的是为了尽量减少错误的信息对模型的误导（例如将贴上错误标签的图像用于分类），或是增加可用的训练的数据等。

2.1 预处理筛除

首先，根据这篇Github 内容[1]所述的方法创建符号链接，避免复制一遍照片。其次，依据这篇博客[2]所提示的信息，我也对 Kaggle 的数据集进行了筛查。所用的方法和博客作者类似，通过已经在 ImageNet 数据集下进行

[1] https://github.com/ypwhs/dogs_vs_cats

[2] https://zhuanlan.zhihu.com/p/34068451

了训练的模型，对包括猫狗在内的多个类别能进行预先的判断。具体来说，结合时下比较精确的三个的网络模型（如图4所示），ResNet50，Xception 和 Inception-ResNetV2，共同对原数据集进行预筛查。每个模型最终的结果选择 TOP-10 作为判断依据，即如果前 10 个判断中有出现猫/狗相关的判断，即可认为这张图片中存在猫/狗，反之则需要添加到排除列表中。在 **pre_check_img.py** 中有代码的具体实现。最终代码将会将结果以 npy 文件和 txt 文件的方式存储结果。npy 文件能够方便的在接下来的编程中快速进行结果调用，而 txt 文件可以很直观的将结果显示出来，部分结果如图5所示。图6所示为两种典型的无法通过 3 种模型判断的照片。图6左为标记在猫分类下的训练照片（cat.7920）。虽然人的肉眼可以看到确实有一张黑猫躺在深蓝色的床单上，但是由于对比度比较低，三个模型都没法将其判别，这样的照片很难给之后所需要训练的模型提供正确有用的信息。这张图还有一个特点是右侧的狗非常显眼，如果这张图标记成猫进行训练，很可能对本项目所需判断的猫狗分类产生比较大的干扰，因此需要移除。图6右侧的图片被标记为狗（dog.8736），显然这是一个明显错误的标记，因此也需要将图片移出训练集。

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.715	0.901	138,357,544	23
VGG19	549 MB	0.727	0.910	143,667,240	26
ResNet50	99 MB	0.759	0.929	25,636,712	168
InceptionV3	92 MB	0.788	0.944	23,851,784	159
InceptionResNetV2	215 MB	0.804	0.953	55,873,736	572
MobileNet	17 MB	0.665	0.871	4,253,864	88
DenseNet121	33 MB	0.745	0.918	8,062,504	121
DenseNet169	57 MB	0.759	0.928	14,307,880	169
DenseNet201	80 MB	0.770	0.933	20,242,984	201

图 4: 当前 Keras 提供的基于 ImageNet 训练模型排名

2.2 数据增强

数据增强是一系列扩充图像数据的方法的统称，通过裁切、添加噪音、改变颜色等手段，对原始照片进行处理，进而生成更多的可用于训练的图像数据，部分地弥补数据量不足造成的过拟合问题，如图7。本项目使用 Github

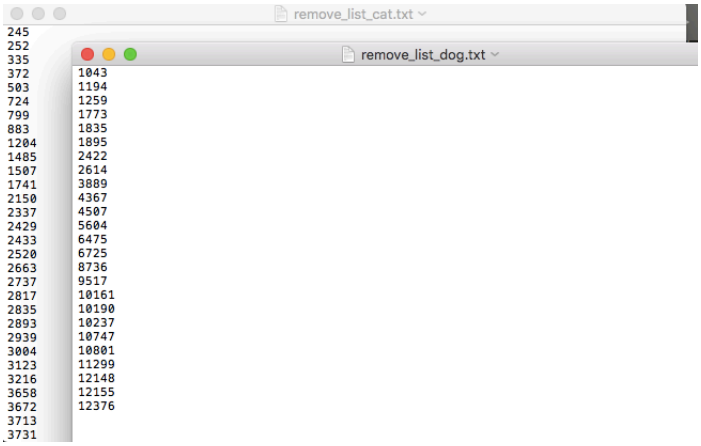

图 5: 部分需要移除的照片列表

图 6: 被移除的部分照片示例

中开源项目 Imgaug[3]进行数据增强的图片生成。Imgaug 集成了多种数据增强手段，可以方便的调用。本项目选用了四种手段进行预处理，每张照片选择两个处理手法进行组合，即一张照片最终可以生成六张照片。示例如图8所示。具体的代码实现可查阅两个 python 文件 **create_aug_fotos.py** 和 **data_aug_tool.py** 。最终通过数据增强算法将通过筛选后的训练文件生成约 15 万张猫狗照片进行后续的模型训练。

[3]https://github.com/aleju/imgaug

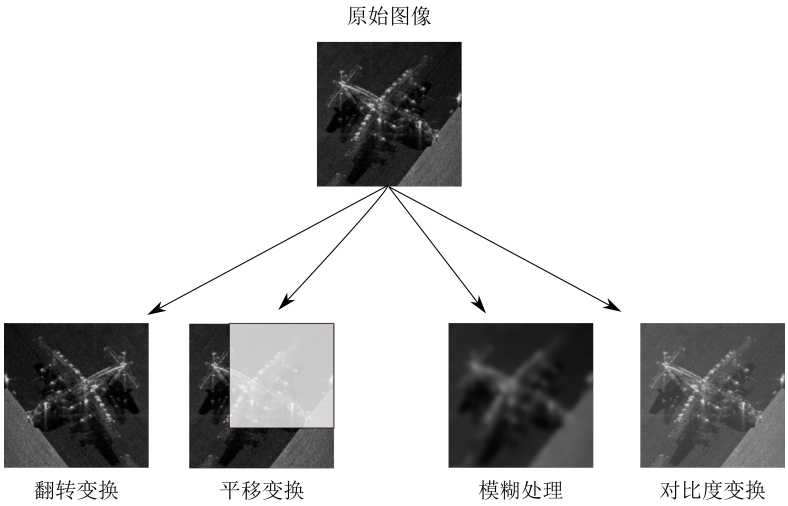

图 7: 数据增强常用的手法

3 算法与方法

由于 Keras 提供了非常便捷的功能，大部分算法部分的结构不需要自行重新编程。为了提高最终的识别精度，本项目结合了之前预筛选所选用的三种模型进行迁移学习，最后加入自定义的层进行训练。如图9所展现的就是三个模型融合的示意图。通过三个已经预训练好的模型，生成的特征向量的融合，接上相对较为简单的 Dropout 层，将三个模型的结果融合成最终的判断–猫/狗。

3.1 生成特征向量

使用特征向量和训练数据生成特征向量，有利于快速进行训练。每个选用的模型都将分别根据提供训练数据生成对应的特征向量。其中使用原始训练数据得到的特征向量.h5 文件大约 300 多 MB，而使用数据增强方式生成的照片进行生成的数据文件每个.h5 文件大约 1.5GB。具体代码详见 **Feature_Gen.py** 或是 jupyter notebook 文件。

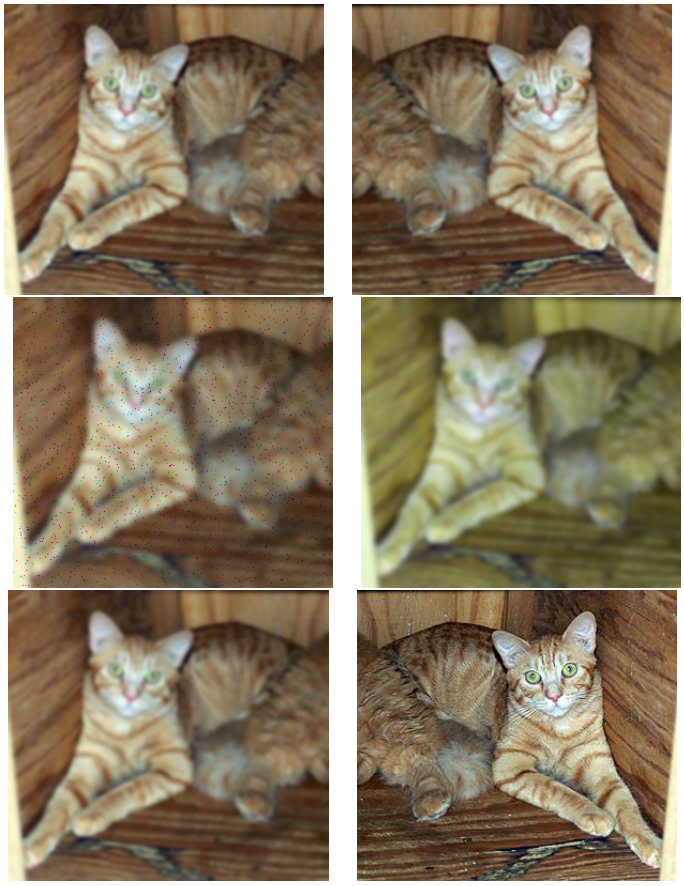

图 8: Imgaug 处理结果

4 测试结果

4.1 测试结果

在测试的时候，将预生成好的.h5 文件载入，连接上自定义的网络层进行训练，并将结果以 Kaggle 要求的 csv 文件方式保存。相较于直接导入数据进行训练，利用特征向量的方法在训练和测试结算所画的时间相对较小，能够很快的完成训练，如图所示。最后将生成的 csv 文件上传至 Kaggle 平台，得到如下测试结果，图10。满足了项目不大于 0.06 的需求。

最终结果为，以数据增强后训练的模型比直接使用原训练数据的模型有较好的测试结果（0.03681 vs 0.03728）。

通过 Tensorboad 可以很方便地观察训练过程，其结果如图11所示。由于本身融合模型已经能够对猫狗进行识别，因此训练的时候起始精度已经非

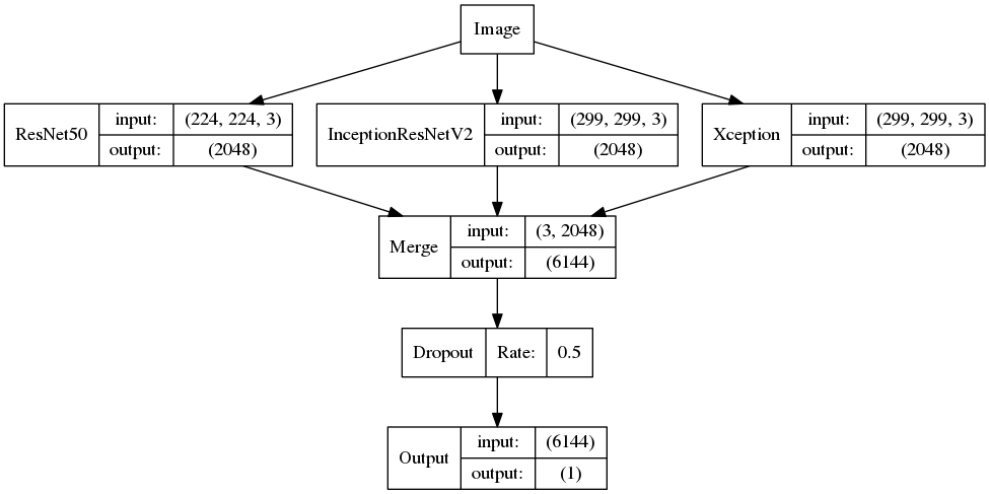

图 9: 自定义的模型结构示意图

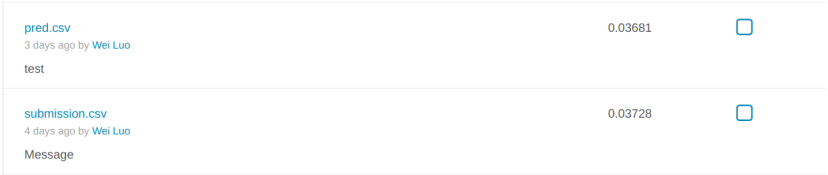

图 10: Kaggle 测试结果

常高，通过较少 epoch 的训练，已经能够达到很高精度。整体训练精度变化符合预期，说明模型设置没有明显问题。

4.2 识别可视化

根据这篇Github 内容中提示的方法，进行了猫狗识别的可视化。可视化的方法是通过热图的方法对相关区域进行高亮，越是对此类别识别相关的区域，越是会以暖色来显示。具体效果如图12所示。由图所示的结果，首先对于所随机抽样的的图片，模型都能做到准确识别。其次，通过观察我们可以发现，基本上模型用于判断是猫还是狗的区域是猫狗的脸部特征，也就是图片中庸红色或是暖色标记的区域。

4.3 需要作出的改进

之后结合模型结构优化，可能会取得更好的结果。例如，可以考虑除了去掉各个预训练模型的全连接层外，进一步扩展可训练的模型参数。基于数

图 11: 通过 Tensorboard 观察的训练过程

据增强的方法已经将可用于训练模型的图像数据扩充了很多，因此有希望训练参数更多的模型，这样原本模型的泛用性会进一步降低，可能会取得更好的结果。

5 项目附件及其说明

现在对 Github 提交的文件进行补充说明：

1. final_paper.pdf – 最终报告文稿
2. create_symbol_link/_2.py – 生成图像的符号链接
3. pre_check_img.py, remove_list_cat/dog.txt – 判断是否非猫非狗
4. create_aug_fotos.py data_aug_tool.py – 数据增强工具
5. Feature_Gen.py – 生成特征向量
6. TrainingandTesting.py – 训练并获得结果
7. submission_first/final.csv – kaggle 提交文件
8. Final_jupyter_notebook.ipynb – 解释性 jupyter notebook

参考文献

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84-90, May 2017.

[2] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[3] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

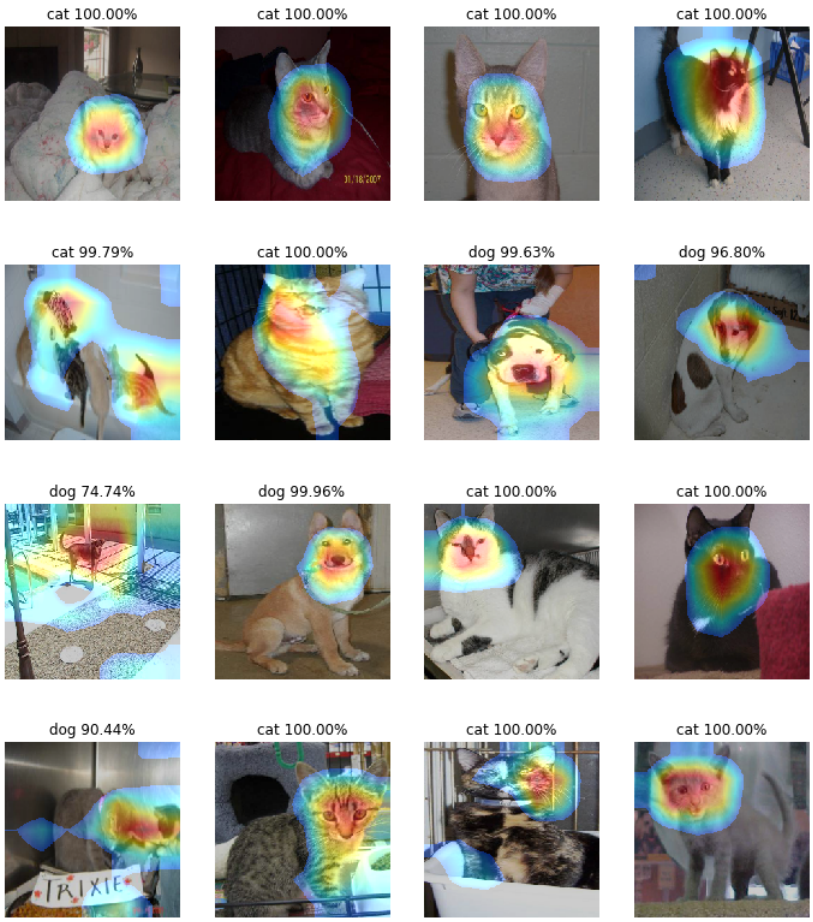
cat 100.00%
cat 100.00%
cat 100.00%
cat 100.00%
cat 99.79%
cat 100.00%
dog 99.63%
dog 96.80%
dog 74.74%
dog 99.96%
cat 100.00%
cat 100.00%
dog 90.44%
cat 100.00%
cat 100.00%
cat 100.00%

图 12: 识别可视化热图