机器学习工程师纳米学位

毕业项目报告-猫狗大战

杨振群

2018/10/10

I. 问题的定义

问题陈述

猫狗大战是一个经典的kaggle竞赛项目，项目目标是训练一个模型在给定的图像中对猫狗进行分类，这是一个图像分类问题，典型的计算机视觉问题。图像分类问题的一般解决方案是通过深度神经网络来进行特征提取，再将提取出来的特征值用于训练分类器模型（参考1）。项目使用的数据集是kaggle竞赛提供的数据（https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data），训练集包括12500张被标记为猫的图片和12500张被标记为狗的图片，测试集包括12500张未标记图片。

训练集的图片用来训练分类器模型，模型输出为给定一张图片预测出图像中是狗的概率。测试集的图片用来评价模型，模型评价指标是对数损失，即测试集样本正确标签的预测概率对数损失的平均值，数学表现形式如下所示。对于单个样本来说，对数损失是正确标签的预测概率的自然对数取负，正确分类标签给出的预测概率为100%的时候对数损失为零，预测概率越低对数损失越大。对数损失相对于正确率来说，是对模型分类能力更加准确的刻画，不仅仅需要模型分类正确，还需要模型对自己分类结果十分有把握。

Submissions are scored on the log loss:

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

where

- n is the number of images in the test set
- $\hat{y}_i$ is the predicted probability of the image being a dog
- y_i is 1 if the image is a dog, 0 if cat
- $log()$ is the natural (base e) logarithm

项目希望得到的结果就是训练得到一个对数损失尽量小的分类器模型，项目预备作为对标的基准结果为该kaggle的leaderboard（https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/leaderboard），目前对数损失第1名得分为0.0033，第50名得分为0.04842。

方案设计

项目计划分成以下几个阶段来开展，使用aws的deep learning ami作为基础环境，使用jupyter notebook作为编程环境，使用keras作为编程基础，使用tableau作为可视化工具，使用github来作为代码文件管理系统（https://github.com/danielyang123321/DogCat_Capstone_Daniel），项目中所列示的文件都存储在github中。

1. 数据预处理

使用百度人工智能图像识别服务替代众包人工审核，对训练集的图片进行标签预测，对识别出来的错误标签进行剔除或者修正，从而提高训练集的标签质量。将图像统一成同样大小的尺寸，使用预训练深度神经网络模型对训练子集和验证子集的图片进行计算得到特征数据，将样本特征数据和样本的标签按一定比例分割为训练子集和验证子集。

2.分类器模型搭建

搭建的分类器模型由全连接层以及激活层组成，对分类器模型进行编译，损失指标采用categorical_crossentropy, 优化程序采用rmsprop,额外观察的评价指标为 accuracy。

3.分类器模型训练

预备使用keras自带的fit函数以及训练子集的数据对分类器模型进行训练，使用ModelCheckpoint来追踪训练过程，batch size选择为20个图片作为一组来更新参数，epoch初步选择为20个回合，将损失指标最小的模型作为最终的分类器模型。

4.分类器模型评估以及调优

模型的评估将在验证子集上开展，评价指标为对数损失，验证集表现良好之后，在测试集上面进行开展。预备提高分类表现的手段包括：对训练子集的图片进行增强从而提高旋转不变性和平移不变形，调整模型编译以及训练使用的参数，考虑使用随机森林等其它分类器模型进行尝试，从只使用一个神经网路提取的特征扩展到使用多个神经网络提取的特征。

5.可视化

项目的可视化主要是模型训练过程中loss指标的变化。

II. 分析

数据的探索

对训练集的图片进行了一些预览，可以发现猫狗在图片中的姿态以及拍摄角度非常多样，背拍、侧拍、俯拍的情况都存在，猫狗在图片的干扰也非常多样，铁丝网遮挡、人和狗多主体、狗和狗多主体的情况都存在，示例如下图所示。






使用了jupyter notebook（DogCat_Capstone_01.Explore Dataset.ipynb）对训练数据集进行了图片维度提取，并观察了宽度、长度和深度的分布情况，图片深度一致为3，宽度和长度平均为360*400，存在分辨率较小和分辨率较高的两种极端情况。

	width	height	depth
count	25000.000000	25000.00000	25000.0
mean	360.478080	404.09904	3.0
std	97.019959	109.03793	0.0
min	32.000000	42.00000	3.0
25%	301.000000	323.00000	3.0
50%	374.000000	447.00000	3.0
75%	421.000000	499.00000	3.0
max	768.000000	1050.00000	3.0

使用了tableau public对提取的训练集维度信息（train_dimension.csv）进行了可视化（https://public.tableau.com/profile/danielyang#!/vizhome/CatDogTableau/dimension_distribution_density），长度或者宽度小于100的占比不到1%，长度或宽度大于600的占比不到1%。预备剔除所有长宽均小于100的图片。

Height Bucket	Width Bucket 0	100	200	300	400	500	700	总和
0	0.30%	0.17%	0.02%					0.49%
100	0.29%	3.58%	1.62%	0.18%	0.03%	0.01%		5.72%
200	0.02%	2.32%	5.34%	2.84%	1.47%	0.48%		12.46%
300	0.00%	0.37%	7.43%	5.35%	6.89%	3.86%		23.90%
400		0.05%	1.98%	21.79%	9.19%	3.03%		36.04%
500		0.04%	0.57%	17.22%	3.56%			21.38%
1000							0.01%	0.01%
总和	0.62%	6.54%	16.94%	47.38%	21.13%	7.38%	0.01%	100.00%

<1% 1%-5% 5%-10% >10%

使用了jupyter notebook（DogCat_Capstone_01.Explore Dataset.ipynb），通过百度动物识别服务对训练数据集上进行了标签预测，提取出来了标签预测数据（train_lablel.csv），其中18张照片由于尺寸较小无法进行识别。对于预测标签数据在tableau中进行了分析（https://public.tableau.com/profile/danielyang#!/vizhome/CatDogTableau/train_label_review），存在可能不一致的情况，涉及图片数量共538张，占比小于2.5%。

Type	train_file_..	baidu_pre..	记录数	%
Match	cat	cat	12,201	48.84%
		合计	12,201	48.84%
	dog	dog	12,243	49.01%
		合计	12,243	49.01%
	合计		24,444	97.85%
NoMatch	cat	dog	77	0.31%
		TBD	214	0.86%
		合计	**291**	**1.16%**
	dog	cat	32	0.13%
		TBD	215	0.86%
		合计	**247**	**0.99%**
	合计		**538**	**2.15%**

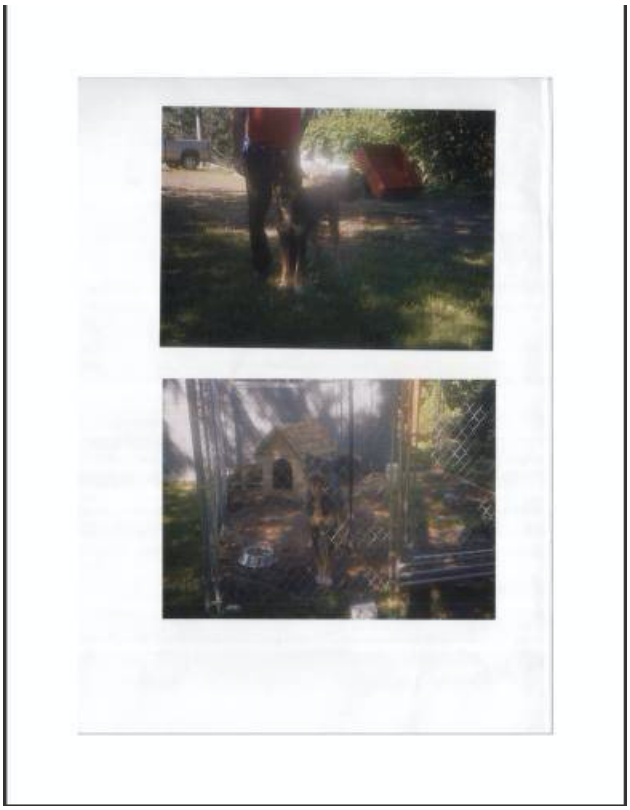
在标签很有可能存在问题的538张照片中，随机挑选了一些进行人工查看，发现其中一些确实存在问题，检查样本如下所示。鉴于标签存疑的图片数量比例小于2.5%，预备不进行人工复核清洗，直接剔除这538张图片，从而达到提高训练集标签质量的目的。

样本1-cat.92.jpg: 图片被标记为猫，但实际是一只玩具猫。

样本2-cat.2835.jpg: 图片被标记为猫，但可能是一只婴儿猫，尚不具备成年猫的特征。

样本3-cat.241.jpg: 图片被标记为猫，但镜头非常模糊

样本4-dog.7076.jpg: 图片被标记为狗，但镜头非常模糊



样本5-dog.11731.jpg: 图片被标记为狗，但明显能看出来是一只猫

算法和技术

项目使用到的技术是用于提取特征的预训练卷积神经网络和用于对提取的特征数据进行分类的全联接神经网络，整个方案相当于迁移学习（参考2）。预备使用的预训练卷积神经网络将是在图像分类领域表现良好的模型，与此项目的领域相似性是迁移学习的基础，这种预训练网络的前端神经元提取的特征对于图像分类问题来说是比较通用的。

keras中提供支持了很多预训练模型（https://keras.io/applications/），其中包括2014年的VGGnet（参考3），2015年的resnet（参考4），2015年的inception（参考5），2016年的xception（参考6），2016年的inception-resnet（参考7）,2016年的densenet（参考8），2017年的nasnet(参考9)，2017年的mobilenet（参考10）。

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
InceptionV3	92 MB	0.779	0.937	23,851,784	159
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
NASNetLarge	343 MB	0.825	0.960	88,949,818	-
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
ResNet50	99 MB	0.749	0.921	25,636,712	168
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
Xception	88 MB	0.790	0.945	22,910,480	126

从上图中的准确率可以看出，表现最好的预训练模型是NASNetLarge，在imagenet数据集上面的第一预测准确率为82.5%，前五预测准确率为96%，参数共计有8千多万个。预备将采用这个预训练模型来提取特征数据，对于每张图片将可以提取出来4032个特征值用于进一步分类。这个预训练模型需要设置的参数如下：

- include_top：预备设置为false，不保留顶层的全连接网络。
- weights：预备设置成’imagenet’，代表加载预训练权重。

- pooling：预备设置为max，代表了采用全局最大值池化方式。
- input_shape：预备采用模型默认的（331，331，3）.

基准结果

预备使用kaggle这个项目竞赛的leaderboard最佳分数作为基准阈值（https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/leaderboard）。目前猫狗大战在kaggle leaderboard上面的对数损失第1名得分为0.0033，第50名得分为0.04842，我们将以进入前50名作为这个项目的基准阈值，即对数损失小于0.04842。

III. 方法

数据预处理

使用了jupyter notebook（DogCat_Capstone_02.Preprocess Dataset），利用在数据探索阶段提取的图片维度数据（train_dimension.csv）以及图片标签预测数据（train_lablel.csv），对长度或宽度小于100的图片以及标签存疑的图片进行了剔除，共计剔除了711个样本，剩余24289个样本，最终选取的样本文件数据以及标签数据存储在train_files_final和train_targets_final中。

使用了jupyter notebook（DogCat_Capstone_02.Preprocess Dataset），对于最终选取的24289个样本，将图片按照统一的大小转换成张量数据（331，331，3），使用预先训练模型nasnetlarge对图片张量数据进行特征提取，nasnetlarge的pooling方式选择为max，每个样本可以提取出来4032个特征值，提取出来的特征数据以及标签数据存储在train_features_final.csv和train_targets_final.csv, test_features.csv。

执行过程-v0

使用了jupyter notebook（DogCat_Capstone_03.Model Training_v0），将预处理过程中的数据文件（train_features_final.csv和train_targets_final.csv）进行加载，利用scikitlearn对于提取出来的特征数据train_features_final和标签数据进train_targets_final行了训练集和验证集的分割，分割比例设置为30%，分割完之后生成了训练子集的特征数据和标签数据（X_train，y_train）以及验证子集的特征数据和标签数据（X_valid，y_valid）。

使用了jupyter notebook（DogCat_Capstone_03.Model Training_v0），搭建了一层全连接神经网络，使用特征数据的维度-4032作为输入，以标签数据的维度-2作为输出，共计有8064个参数。对分类器模型进行编译，损失指标采用categorical_crossentropy, 优化程序采用rmsprop,额外观察的评价指标为 accuracy。

使用了jupyter notebook（DogCat_Capstone_03.Model Training_v0），使用keras自带的fit函数以及训练子集的数据对分类器模型进行训练，使用ModelCheckpoint来追踪训练过程，batch size选择为20个图片作为一组来更新参数，epoch初步选择为20个回合，将损失指标最小的模型作为最终的分类器模型。在验证集上得到的最佳损失指标为0.01691。

使用了jupyter notebook（DogCat_Capstone_03.Model Training_v0），将具有最小损失指标的分类模型应用在预处理过程中提取出来的测试特征数据（test_features.csv）上，并将提取出来的预测结果数据（submission_v0.csv）提交至kaggle上，kaggle给出的损失指标为0.21864，这个结果和验证集上的表现相差巨大。

完善过程-v1

在执行过程-v0中，验证集损失指标和测试集损失指标相差巨大，观察了预测结果数据，发现模型的输出比较武断，给出的概率大部分都是1或者0，这种预测概率在预测错误时候带来的对数损失会比较大，因此考虑在分类模型之后增加一个预测概率修正函数predict_adjust。

使用了jupyter notebook（DogCat_Capstone_03.Model Training_v1），将调整函数的epsilon常数设置为0.005，利用同样的数据和训练方法，将具有最小损失指标（0.01887）的模型应用在测试数据集上，并将提取出来的调整预测结果数据提交至kaggle上，kaggle给出的损失指标为0.04414。验证集和测试集之间存在的巨大差异至此得到了解决，这样的对数损失在kaggle public leaderboard中可以排到前20名，达到了项目的基准阈值（0.04842），与最佳对数损失指标0.03302之间仅相差0.01112。

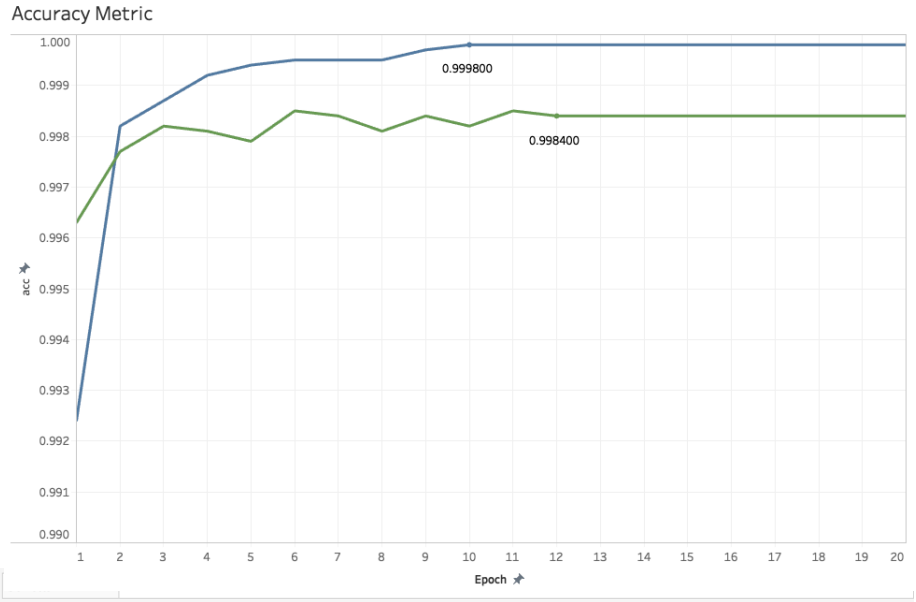
IV. 结果

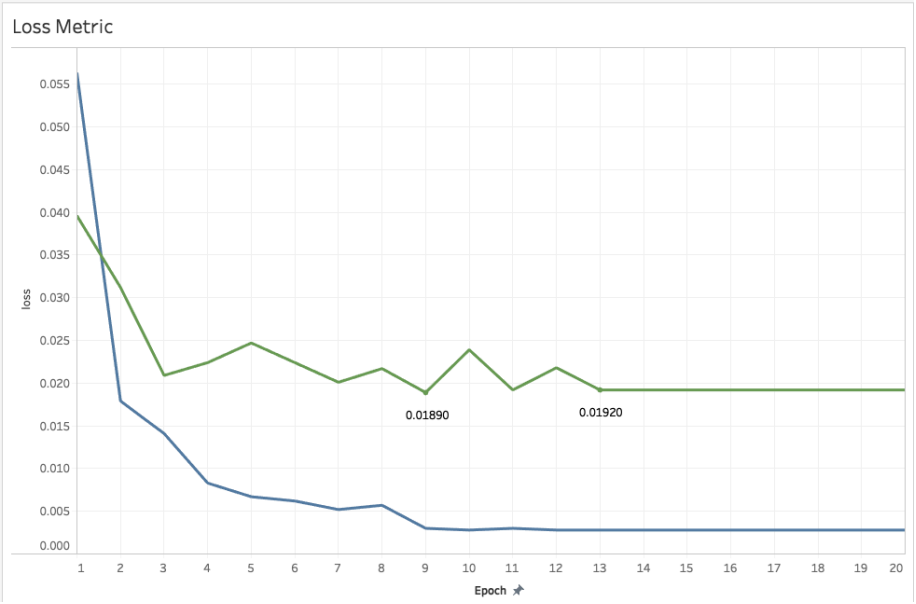
改善后的模型v1是由预训练的naslarge卷积神经网络、8066个参数的全连接分类神经网络、预测概率调整函数组成，整个模型的组成非常简洁，其中预训练的naslarge卷积神经网络是核心部件，承担了图像特征数据提取的重任。模型的输出跟期待的结果一致，在验证集上面的损失质保为0.01887，在测试集上得到的损失指标为0.04414，我们定义的基准损失指标为0.04842，以此推断，模型可以很稳定地解决猫狗图像分类问题。

V. 项目结论

结果可视化

其中对分类模型v1的训练过程中的损失指标、准确率指标整理成了excel文件（Metrics in training.xlsx），并使用了tableau进行了可视化（https://public.tableau.com/profile/danielyang#!/vizhome/CatDogTableau_metric/AccuracyMetric），损失指标和准确率指标在13个epoch后趋于稳定，其中损失指标在验证集上远远低于基准阈值。





对项目的思考

项目比我预期的要进行得顺利很多，阅读了这个项目的参考论文，发现前辈们在探索解决方案的时候需要尝试很多不同的优化方法，而在这个项目中我几乎是一次尝试即可达到目的。我认为这是得益于预训练卷积神经网络的进化，这次做项目的时候keras已经开始支持naslarge这个预训练神经网络模型，这个cnn在提取图像特征上的能力相对于之前的模型来说得到了提高，高质量的特征数据为分类模型的训练打下了很好的基础。这个项目给我最大的感悟是，特征提取的质量对于图像分类任务来说是极其关键的。

需要作出的改进

预备提高分类模型表现的手段在项目中并没有使用到，但这些改进可能可以进一步提高模型的分类能力，这些手段包括：对训练子集的图片进行增强从而提高旋转不变性和平移不变形，考虑使用随机森林等其它分类器模型进行尝试，从只使用一个神经网路提取的特征扩展到使用多个神经网络提取的特征。

项目参考

1. Yilun Wang, Michal Kosinski (2017). Deep neural networks are more accurate than humans at detecting sexual orientation from facial images, *Journal of Personality and Social Psychology*, https://www.gsb.stanford.edu/sites/gsb/files/publication-pdf/wang_kosinski.pdf

2. Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson (2014). How transferable are features in deep neural networks, *ARXIV*, https://arxiv.org/pdf/1411.1792.pdf

3. Karen Simonyan, Andrew Zisserman (2014). very deep convolutional networks for large-scale image recognition, *ARXIV*, https://arxiv.org/pdf/1409.1556.pdf

4. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (2015). Deep Residual Learning for Image Recognition, *ARXIV*, https://arxiv.org/pdf/1512.03385v1.pdf

5. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna (2015). Rethinking the Inception Architecture for Computer Vision, *ARXIV*, https://arxiv.org/abs/1512.00567

6. François Chollet (2016). Xception: Deep Learning with Depthwise Separable Convolutions, *ARXIV*, https://arxiv.org/abs/1610.02357

7. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, *ARXIV*, https://arxiv.org/abs/1602.07261

8. Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger (2016). Densely Connected Convolutional Networks, *ARXIV,* https://arxiv.org/abs/1608.06993

9. Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V. Le (2017). Learning Transferable Architectures for Scalable Image Recognition, *ARXIV*, https://arxiv.org/abs/1707.07012

10. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, *ARXIV*, https://arxiv.org/abs/1704.04861