AI算法工程师纳米学位毕业项目
猫狗大战

沈捷

2019年6月26日

1 问题定义

1.1 项目概述

本项目是一个图像分类问题，即训练一个神经网络，使其能够在猫和狗的照片中将二者区分开。这些照片由Kaggle竞赛“猫狗大战”（Dogs vs Cats）题目提供。这项赛事亦是图像分类问题中的著名竞赛题，曾经激发了世界上许多深度学习领域的人才贡献大量方案，极大地促进了该领域的发展[1]。虽然正式赛已经结束许久，但仍有学者以这些数据集为素材进行图像分类问题的研究，有着旺盛的生命力[2]。

图像分类问题即是对图像进行简单的分类，区分成两个以上的预设类别。这是计算机视觉的基础问题，将为其他更复杂的问题铺设道路，例如定位、物体检测、分割等等。[3] 但图像分类本身也有其应用场景，例如医学影像学上研究最广泛的区分各种类型的肿瘤的影像，准确率已可与人类病理学家相媲美。[4]

图像分类问题中，应用最成功的模型是深度卷积神经网络（Deep Convolutional Neurol Network，DCNN），它在2012年的ImageNet图片分类项目中备受瞩目，并衍生出许多成熟的图像分类预训练模型，使研究人员能在此基础上进行迁移学习、改造应用。本项目即将采用迁移学习的方式，构建一个CNN模型对图像进行分类。

1.2 问题陈述

本项目是要将日常生活照片中的猫和狗进行区分，显然是一个二分类问题。通过输入图像特征，获得一个概率，通过概率来判断属于哪一个类。

1.3 评价指标

模型将采用对数损失函数（log loss）进行评价。对数损失函数需要输入每个分类的预测概率与标签，对错误的分类进行惩罚，从而对准确率（Accuracy）进行量化[5]。损失越少，准确率越高。对数损失函数公式[1]：

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

其中：
n 是样本数量
$\hat{y}_i$ 是图像 i 为狗的预测概率值
y_i 是图像 i 的标签，y_i =1是狗，y_i =0是猫
$log()$ 是以自然数 e 为底的对数函数
此外，预测结果将上传到Kaggle进行排名对比，目标是要达到Kaggle排名的前10%。

2 分析

2.1 数据的探索

在正式分析之前，要对数据进行一定的探索。本将数据集由Kaggle竞赛题提供，分为训练集和测试集。其中训练集共25000张图片，标记为猫和狗的各12500张，标签就在文件名中，文件名格式为“标签.编号.jpg”的格式。测试集共12500张图片，文件名只有编号，没有标签。

随机抽取部分训练集样本进行查看：

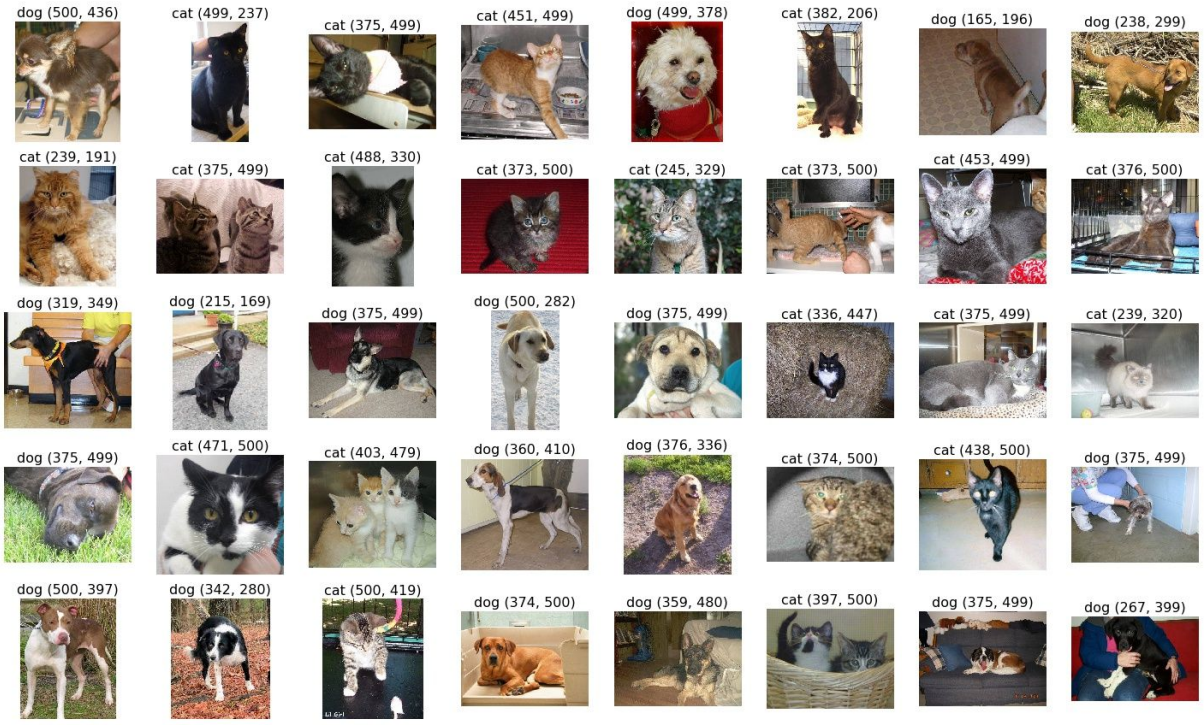

Fig 1. 随机抽取训练集数据进行预览。各图片标题表示“标签（高px，宽px）”。

可见图片长宽多在300~500像素之间，目标主体基本清晰。只不过有的主体占图片的比例较小，有的图片上有两个或以上的目标，可能会对模型训练产生影响，且尚不知是否有分类错误的训练样本。

所以在数据探索步骤，打算采用预训练模型进行初始预测，ImageNet的1000个标签中，有118个狗的品种和7个猫的品种[6]，可以做为参考，找到那些预测与标签不符的图片，再做人工确认。对可能影响模型训练的异常图片将剔除，再检查剔除之后的样本分布。

2.2 探索性可视化

做初始预测的预训练尝试了ResNet50、InceptionV3和Xception这3种，将预测排名前60的标签中匀不含猫或狗，或者预测与实际标签的猫或狗分类不符者，视为异常值。将异常图片人工审查后，最终选用ResNet50模型预测的结果，相对来说预测错误的图片较少。

最终得到130张异常图片，如下：

Fig 2. 随机抽取异常数据检视详情。各图片标题为文件名。

可见这些图中，有的目标主体太小，有些有遮挡，有些模糊，有些曝光度太高以至特征不易辨识，甚至一张图片上同时有猫和狗，或两者都不是，这都是对模型训练的干扰，这部分图片将被剔除。

剔除后，再进行训练集标签分布情况的检查，绘制饼图查看各分类所占的比例。

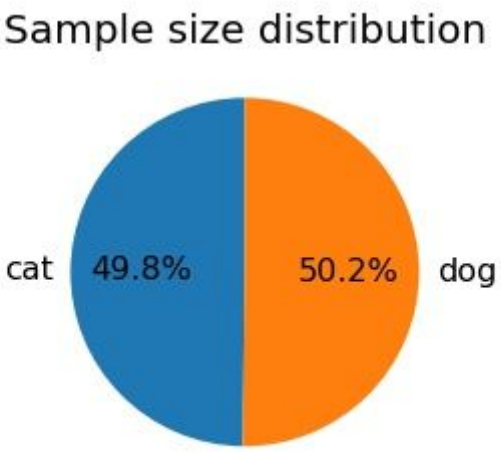

Fig 3. 剔除异常数据后各分类的占比。

从Fig 3可见，两种分类的比例几乎未受影响，基本维持1:1的均衡比例。

将剩下的训练集数据按4:1比例切分为训练集和验证集，得训练集19896个样本，验证集4974个样本。再次各绘制饼图检查样本分布。从Fig 4可见，两个数据集的表现基本一致，两个类别仍然维持均衡，无需特别处理。

Fig 4. 训练集和验证集中各分类的占比。

2.3 算法和技术

本项目采用迁移学习的方法。迁移学习是一种常用的深度学习模型搭建方法，尤其见于图像识别任务中。[7]图像识别所需的深度学习模型通常较复杂，也需要花费较长的时间去训练。而迁移学习则是使用一些在其他相关任务中已训练好的公共模型，经过一些调整，进而用于当前任务。

尤其得益于ImageNet项目，这些公开模型经过了大量被标记图片的训练，获得了很好的图片特征提取能力，在该项目中可以识别1000种图像分类。在本项目中，可以选用一种预训练模型，去除顶部全连接层，再根据需要增加其他层，最后加上一个识别2种分类的全连接层，用本项目的猫狗图片重新训练这些新增的层，则可以节省重新构建模型的工作量，也减少了训练时间。

在本次任务中，由于是二分类问题，所以在去除顶层之后的预训练模型之上，加一个全局池化层，最后加一个全连接层，节点设为1，采用激活函数 $sigmoid$，将线性预测值映射到 $(0, 1)$ 之间，即为概率（狗 = 1， 猫 = 0）。若在调整过程中增加其他全连接层，则激活函数选用 $relu$ 。

在训练过程中，将调整数据增强、模型结构、随机丢弃节点数、epoch、学习率等等参数，对模型进行调整。并采用过早停止方法获得最佳模型，以验证集准确率为指标，耐受度根据训练早期表现进行调整，一般为5，但如果损失波动过大，可考虑放宽至10以便观察，当超过耐受周期仍无改进时则停止训练，获得当前模型。最后也将训练过程的准确率和损失变化绘制曲线进行可视化，进行确认。

本项目将采用keras框架，以tensorflow为后端，进行整个图像预处理、模型搭建、训练和预测的任务。

2.4 基准模型

本项目尝试了多种基础预训练模型，如ResNet50、ResNet152。

ResNet50是一个由50层组成的深度学习模型，曾在猫狗大战项目中取得较好的准确率。该神经网格中的各层用于检测图片中的轮廓、曲线、直线等特征，再加上新训练集的训练，应当能适用于本项目的任务。[7]

ResNet152与ResNet50相似，但它是一个更深层的残差学习网络，有152层，并且解决了较深的模型更难训练的问题，它更容易优化，复杂度低，在ILSVRC 2015图像分类任务中夺得头筹。[8]

本项目的基准模型将设为，去除顶部全连接层的ResNet152预训练模型，加一个1节点的全连接层，激活函数 $sigmoid$ 。训练数据不做增强，学习率也使用Adam优化器的默认值0.001，以此训练结果为基线参照。

3 方法

3.1 数据预处理

前期已经采用ResNet50对所有训练数据进行预测，找出预测前60的标签中均无猫狗分类，或猫狗分类不准确的数据，经人工检查判断为可能会对训练产生干扰的数据共130个，予剔除。再将题目提供的训练集按4:1比例切分为训练集和验证集，得训练集19896个样本，验证集4974个样本，检查二者类别分布都基本维持1:1的比例。

接下来，图片会根据所选用的预训练网络的要求统一图片大小，对ResNet152来说则是 224×224 大小；再分解为RGB三个颜色通道的色值，均经过与预训练模型一致的标准化处理，以模型的preprocess_input()函数执行。

再将训练集进行一定的图像培强，即随机水平或垂直翻转、随机旋转一定角度、按一定比例缩放、裁剪等等，任选1~2种，使模型具有更好的泛化能力，也不必选用太多耗费更多计算资源。以此做为模型输入。

验证集图片仅根据预训练模型统一大小并标准化，不做数据增强处理。

3.2 执行过程

模型在训练过程中，尝试调整模型结构、数据增强方法、学习率等参数，并采用过早停止方法，历次各训练结果如下：

Table 1. 历次训练验证情况汇总

方案	变化	训练集损失	验证集损失	测试集损失
ResNet152; lr=0.001； epoch = 10/40	基线	0.0203	0.0436	0.10487
水平翻转； ResNet152+ Dense(500)+ Dropout 0.75; lr = 0.00005; epoch = 25/40	增加全连接层和丢弃层 增加数据增强方法 降低学习率	0.0175	0.0300	0.10795
水平翻转； ResNet152+ Dropout 0.75+ Dense(500)+ Dropout 0.75; lr = 0.00005; epoch = 22/40	增加丢弃层	0.0915	0.0425	0.08973
水平翻转+ 裁剪比例0.2； ResNet152+ Dropout 0.75+ Dense(500)+	添加数据增强方法	0.0930	0.0437	0.08328

Dropout 0.75; lr = 0.00005; epoch = 22/40				
水平翻转+ 裁剪比例0.2； ResNet152+ Dropout 0.75+ Dense(500)+ Dropout 0.75; lr = 0.00001; epoch = 40/40	降低学习率	0.1293	0.0451	0.08889
水平翻转+ 裁剪比例0.2； ResNet152+ Dropout 0.3+ Dense(500)+ Dropout 0.3; lr = 0.00001; epoch = 40/40	降低丢弃比例	0.0348	0.0356	0.09875
水平翻转+ 随机旋转30度； ResNet152+ Dropout 0.75; lr=0.00005; epoch = 40/40	更改数据增强方法 删除一个全连接层和丢弃层 恢复原先的学习率	0.0936	0.0422	0.08057
水平翻转； ResNet152+ Dropout 0.75; lr = 0.00005; epoch = 40/40	减少数据增强	0.0774	0.0398	0.07691

3.3 完善

训练时，初始方案的训练集和验证集表现较好，但过程中波动较大，有过早停止。而且从测试集损失来看，是存在过拟合。

在后续训练调整参数的过程中，首先尝试了降低学习率，找到了使曲线波动更小的学习率，约为 $e-5$ 的数量级，在此基础上构建模型，即在基准模型的基础上再添加一个500节点的全连接层和0.75丢弃比例的丢弃层，并添加水平翻转的数据增强方法，以0.00005的学习率进行训练。此模型的训练集和验证集表现更佳，但测试集比基准模型更差一些，故仍然认为有过拟合。

此后又尝试了增加丢弃层、添加一个数据增强方法等方案，测试集的表现有所改善，却尚有提高空间。此时由于训练集和验证集的表现不如之前，也曾考虑是否出现欠拟合，而且训练过程

中验证集的损失变化波动较大，所以尝试进一步降低学习率、减少丢弃比例等，则测试集的表现更加不理想，故而放弃此方向的调整。

仍然考虑模型是过拟合的情况，此后恢复了原先的学习率，并删除了一个全连接层和一个丢弃层，简化模型，而剩下的一个丢弃层的丢弃比例也调高为原先的0.75。也尝试了更换一个数据增强方法。此时测试集损失降低，往好处发展。

最后尝试减少数据增强，是因为考虑到数据增强方法太多的话，可能会消耗更长训练时间、更多的训练周期，却意外发现训练集、验证集和测试集的表现都有较明显的改善。

此后又另外尝试过调高学习率，效果均无改善，故以此为最后方案。

4 结果

4.1 模型的评价与验证

最终的模型采用ResNet152预训练模型进行迁移学习，经过模型结构及各项参数的调整，得到的最终模型，是在去除ResNet152顶部全连接层的基础上，增加了一个全局平均池化层、一个丢弃比例为0.75的丢弃层，和一个节点为1的全连接层，其激活函数为 *sigmoid*，用于最后的二元分类。
ResNet152在ILSVRC 2015图像分类项目中表现最好[8]，为项目提供了很好的基础。而后自定义的层经过调整，只添加最后用于二分类全连接层而不再添加其他全连接层，是较好的方法，太复杂的模型极容易出现过拟合。

而随机丢弃节点的比例，在早期亦尝试过0.25、0.5等等，最后仍然由于过拟合问题选择了较高的0.75。

在此模型结构中，亦增尝试在训练时调整学习率，在0.001、0.0005、0.0001、0.00005、0.00001中来回选择，验证集的差异均不大，甚至当学习率较高时，验证集的损失波动更大，可能难以达到损失低谷；而学习率太低，则损失降低太慢，需要花费更多的训练周期，也有可能停留在局部最低点而受限，所以最后选择了0.00005。

4.2 合理性分析

最终模型得到测试集的损失为0.07961，比基准模型的0.10487降低了24%，而验证集的损失也从0.0436降至0.0398，降低了9%，性能有了明显的提升。

最终验证集的准确率也达到了98.55%，对猫狗二分类的预测基本可以满足日常应用场景。

5 项目结论

5.1 结果可视化

将模型训练过程所历的epoch做为横坐标，训练集和验证集的历次准确度为纵坐标，绘制曲线；同理，也以训练集和验证集的历次损失作曲线。

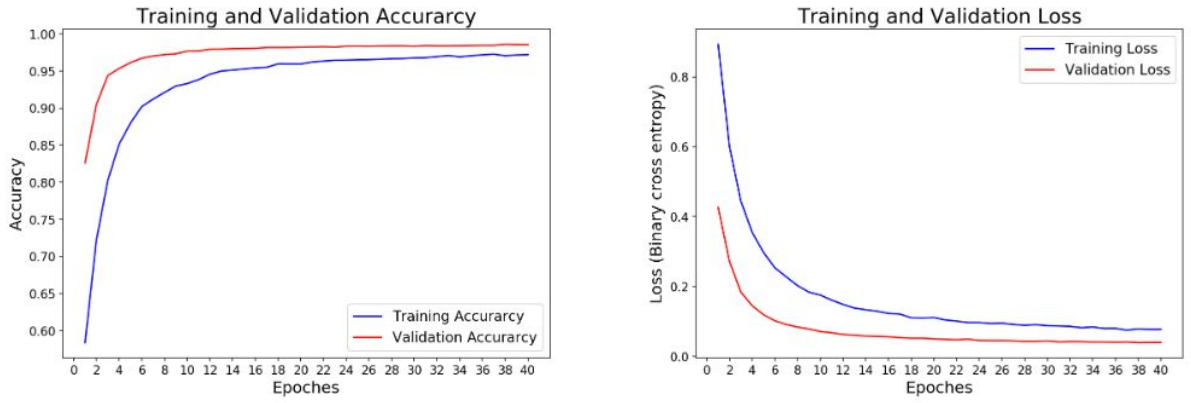

Fig 5. 最终模型训练过程中准确率和损失的变化。

可见随着训练的进行，准确度逐渐升高，损失逐渐降低，最后趋于稳定收敛，并各自达到较理想的结果。

5.2 对项目的思考

本项目采用迁移学习的方法，利用预训练的ResNet152网络，尝试解决一个图像二元分类的问题，这也是类似问题中最常见的一种解决方案。

在训练过程中，最令人困惑的是，模型到底属于过拟合还是欠拟合。多数情况下，训练集的损失和准确率情况均不佳，似有欠拟合的征象。但验证集的表现又非常好，早已可以排进Kaggle竞赛项目中前10%的位置，然而测试集往往不如意，又似有过拟合的征象。

在项目早期还曾考虑，是否在将数据切分成训练集和验证集时不慎造成泄露，使验证集数据参与了训练。在排除了这个原因之后，一直未能脱离这种模式。一个可能的解释是，验证集数据较简单，特征明显，所以表现良好。而测试集样本量较验证集要大得多，代表着真实世界中更复杂的情况，所以可能出现更多的偏差。诚然有过增加验证集样本量的想法，但训练同样需要更大的数据来支持。权衡之下决定不做调整。

从早期增加模型复杂度，例如增加一个全连接层的尝试看来，调整之后的确使训练集损和验证集的损失更低，但测试集损失却反而升高，表现出明确的过拟合征象。

此后基本往防止过拟合的方向调整，虽过程中有所迟疑，不过最后模型的结构和参数都证明了此前模型是过拟合的猜想。

5.3 需要做出的改进

在观察到模型结构更复杂、层数更多的情况下，的确能使训练集的表现更好，但同时也带来了训练时间增加、极易过拟合等问题，则需要更多的训练数据的支持，也要花更多时间来调试。条件充足时，可以寻找一些外部数据集，扩大训练样本，再增加模型层数或节点数，进一步提高准确率、降低损失。

本次训练图片均是生活场景下的图片，所以非常贴合平常应用；而且预测一张图片所需时间很短，所以日后可尝试做成手机App，将手机摄像头获取的图片进行即时预测，也是有趣的应用。

参考文献

1. Kaggle. Dogs vs. Cats Redux: Kernels Edition. *Kaggle* Available at: https://kaggle.com/c/dogs-vs-cats-redux-kernels-edition. (Accessed: 5th April 2019)
2. Perez, L. & Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. (2017).
3. Rawat, W. & Wang, Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Comput.* **29**, 2352–2449 (2017).
4. Hekler, A. *et al.* Pathologist-level classification of histopathological melanoma images with deep neural networks. *Eur. J. Cancer Oxf. Engl. 1990* **115**, 79–83 (2019).
5. klchang. 对数损失函数(Logarithmic Loss Function)的原理和 Python 实现. 博客园 Available at: https://www.cnblogs.com/klchang/p/9217551.html. (Accessed: 5th April 2019)
6. 262588213843476. text: imagenet 1000 class idx to human readable labels (Fox, E., & Guestrin, C. (n.d.). Coursera Machine Learning Specialization.). *Gist* Available at: https://gist.github.com/yrevar/942d3a0ac09ec9e5eb3a. (Accessed: 16th April 2019)
7. Scott, M. What Is Transfer Learning? | NVIDIA Blog. *The Official NVIDIA Blog* (2019).
8. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *ArXiv151203385 Cs* (2015).