

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	3
1.1 项目概述	3
1.2 问题陈述	3
1.3 评价指标	3
2 分析	3
2.1 数据的探索	4
2.2 探索性可视化	4
2.3 算法和技术	7
2.4 基准模型	9
3 方法	9
3.1 数据预处理	9
3.2 执行过程	9
3.3 完善	12
4 结果	13
4.1 模型的评价与验证	13
4.2 合理性分析	14
5 项目结论	14
5.1 结果可视化	14
5.2 对项目的思考	14
5.3 需要做出的改进	15
参考文献	16

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 Neural 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。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：

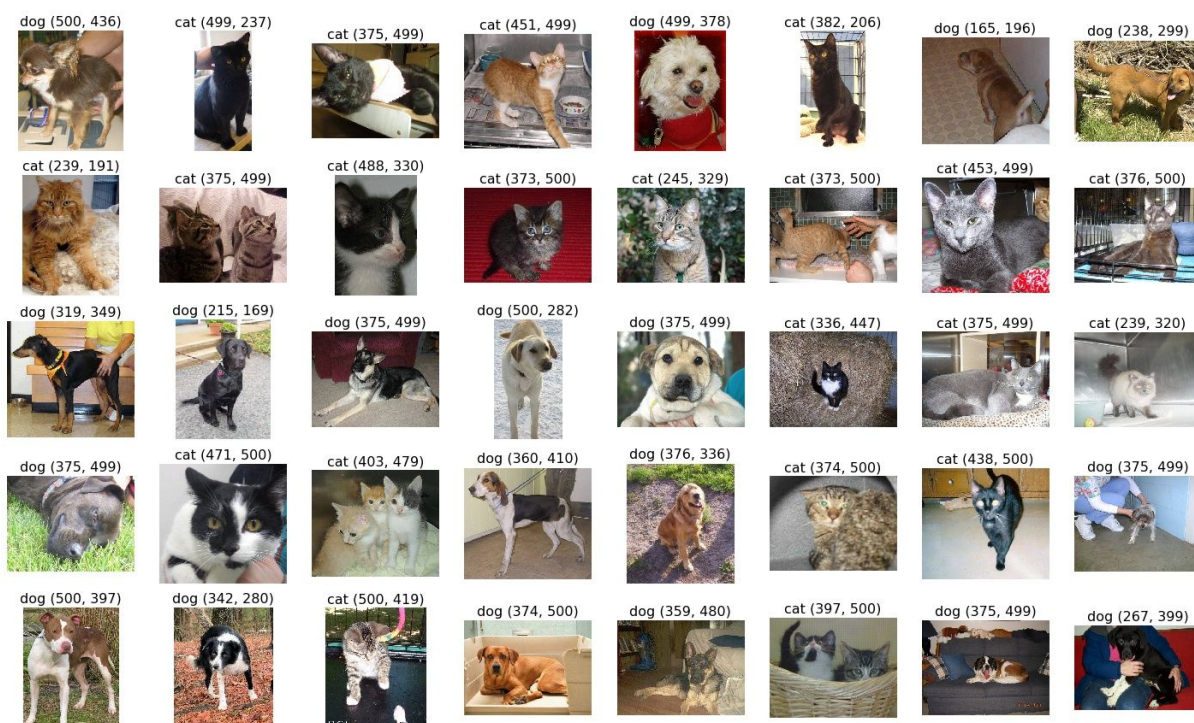


Fig 1. 随机抽取训练集数据进行预览

Fig 1中，各图片标题表示“标签（高px，宽px）”。可见图片高宽多在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^[5]，可以作为参考，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^[6]、InceptionV3^[7]和Xception^[8]这3种，将预测排名前60的标签中均不含猫或狗，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。



Fig 2. 随机抽取异常数据检视详情

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。

Sample size distribution

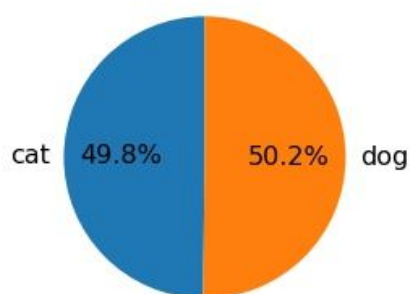


Fig 3. 剔除异常数据后各分类的占比

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。

再观察剩下的图片中的长宽分布：

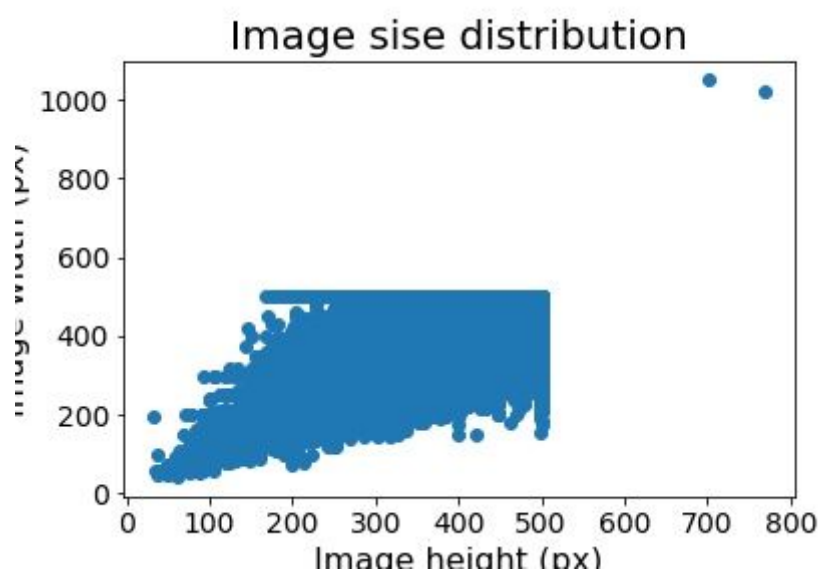
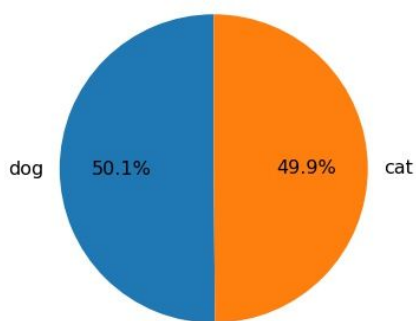


Fig 4. 图片长宽分布(px)

可见图片长宽范围约为50至500，但有两幅图长宽达800~1000像素，也可能会在图象变换时造成一些扭曲，所以也予剔除。

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19894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 5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。

Sample size distribution of training set



Sample size distribution of validation set

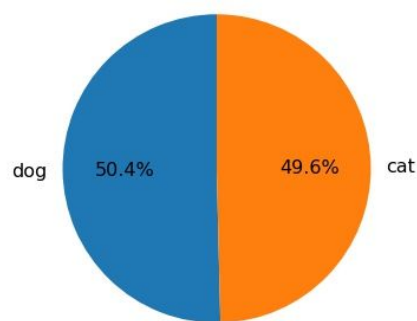


Fig 5. 训练集和验证集中各分类的占比

训练集和验证集的图像大小、高宽比的分布也较为一致。高宽比约为2:1至1:2之间，集中趋势为1:1，这对图像预处理时统一大小比较有利，多数预训练模型的要求都为1:1的比例。

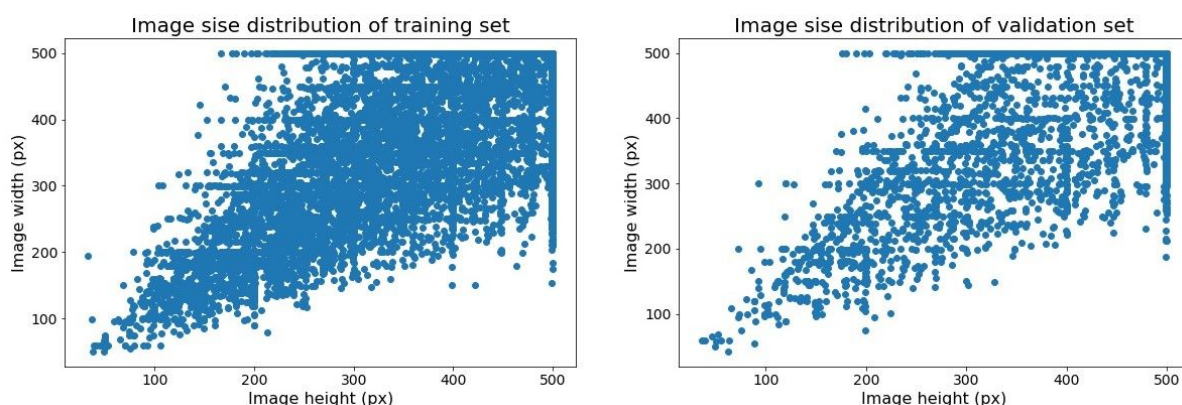


Fig 6. 训练集和验证集图片的高宽像素分布

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。^[9]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^[10]，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。

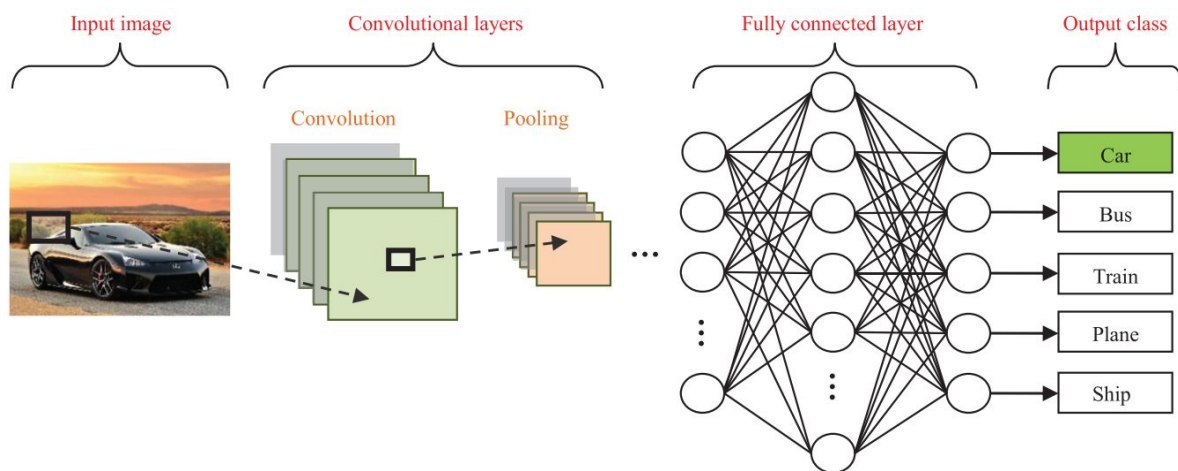


Fig 7. CNN基本结构示例^[3]

如Fig 7所示，是一个典型CNN的基本结构示例。CNN结构虽多种多样，但有一定的共同特征。输入的图片会经过卷积层与池化层模块的处理，这些模块一次堆叠形成深度模型；接着，处理过的特征再进入一个或多个全连接层，最后一个全连接层配合适当的激活函数，输出签标类别。^[3]

在迁移学习任务中，通常是去除顶部全连接层而利用其他预训练模型的结构，再根据任务所需，重新构建合适的全连接层和输出层。而原始结构可以锁定做为特征，不进行训练，以此节约计算资源，也可以打开参加训练，以获得更高的精确度。^[9]

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，也先锁定预训练模型的特征，仅从顶部输出层开始训练，必要时再打开全部的层一起训练。

本项目的预训练模型拟采用ResNet50或ResNet152，即深度残差学习网络^[6]。此前已有证据表明，深度学习进行图像识别的任务中，网络的深度是非常重要的因素^[11]。但简单堆积层数会带来准确率退化的问题，使得训练更困难。深度残差模型则提供了一种方案，即令一组非线性层去拟合残差映象（residual mapping），而非恒等映象(identity mapping)。理想情况下，与拟合恒等映象相比，将残差向0拟合会更简单。这就使得深度学习网络的训练较以前的模型更容易训练^[6]。

ResNet50是一个由50层组成的深度残差学习网络，曾在猫狗大战项目中取得较好的准确率。ResNet152则更深，有152层，也曾在ILSVRC 2015图像分类任务中夺得头筹。^[6]本项目在训练过程中，将根据训练情况以及设备性能等因素进行适时调整。

模型的优化主要采用Adam算法^[12]，即自适应矩估计（adaptive moment estimation）优化器，它可以为不同参数计算自适应学习率。它有很高的计算效率和较低的内存要求，因此在许多深度学习实验中都广受欢迎。

本项目还会采用随机丢弃法^[13]，这是一种很常用的避免过拟合的方法^[3]。训练过程中，当训练数据递呈给神经网络时，每个隐藏单元都有一定概率被忽视，使得隐藏单元不能互相依赖，避免了复杂的训练数据共适应现象（coadaptation）。在测试期间，所有的隐藏单元都会被用到，但输出结果会乘以之前的丢弃概率进行补偿。^[3,14]实验表明，该方法确实有显著的调控效应，减少了过拟合的情况。^[14]

数据增强也是较流行的避免过拟合的方法。在图像识别任务中，是指在预处理阶段将图片进行随机翻转、旋转、缩放、调节各颜色通道的强度等等，从而人为制造更大量的训练数据，使模型具有更好的效果。^[3]但数据量太大也会消耗更多的运算资源，所以本次也会根据项目情况和设备性能进行调整。

dropout

在本项目中将调整数据增强、模型结构、随机丢弃概率、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

本项目将采用较ResNet更早一些流行的AlexNet迁移学习模型为基准模型^[15]。AlexNet是大卷积核的CNN，拥有5个卷积层、3个全连接层。该模型在2010年的ILSVRC竞赛中夺冠，其top-1和top-5错误率分别为37.5%和17.0%，可以选为合适的基准模型。

在本项目中，要将AlexNet去除顶部全连接层，再增加2个全连接层，以及最后一个1节点的全连接层，激活函数 *sigmoid*。用Adam优化器以0.0001的学习率进行训练，以此训练结果为基线参照。

此处得测试集的logLoss为0.32677。本项目的模型经调整训练，应使测试集损失达到Kaggle竞赛排名前10%，即0.06127^[1]。

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ResNet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增强，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1~3种，使模型具有更好的泛化能力，并根据设备性能进行调整，避免耗费太多计算资源导致崩溃。以此做为模型输入。

验证集与测试集图片仅根据预训练模型统一大小并标准化，同样用相应预训练模型的preprocess_input()函数进行处理，不做数据增强处理。

3.2 执行过程

首先将采用ResNet152预训练模型，去除顶部全连接层后，锁定预训练层作为特征，再加上一些全连接层进入训练，如前所述，最后一层为1个节点的全连接层，sigmoid激活。

在模型在训练过程中，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. 第一阶段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
AlexNet+ Dense 512+ Dropout 0.75+ Dense 256+ Dropout 0.75; lr=0.0001 ; epoch = 20/40	基线	0.1459	0.2594	0.32677
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;	增加丢弃层	0.0915	0.0425	0.08973

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 = 0.00005; epoch = 22/40	添加数据增强方法	0.0930	0.0437	0.08328
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

在后续调试过程中，无论降低学习率还是修改模型结构、调整数据增强方法等，均未能达到更好的效果。

接下来将打开底部各层进入训练。但如此则ResNet152会占用大量计算资源，由于设备条件的限制，下面改用ResNet50模型进行迁移学习。同时也可以丰富数据增强的方法做为补偿，故本阶段的训练均采用水平翻转+裁剪比例0.2、缩放比例0.2的方法。

Table 2. 第二阶段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
ResNet50+ Dropout 0.75+ Dense 1024+ Dropout 0.75+ Dense 500+ Dropout 0.75; lr = 0.0001; epoch = 11/40	更换预训练模型 增加可训练层数 增加全连接层 增加数据增强方法 提高学习率	0.0316	0.0342	0.08348
ResNet50+ Dropout 0.75+ Dense 1024+ Dropout 0.75+ Dense 500+ Dropout 0.75; lr = 0.00001; epoch = 16/40	降低学习率	0.0130	0.0294	0.07761
ResNet50+ Dropout 0.75+ Dense 1024+ Dropout 0.75+ Dense 500+ Dropout 0.75; lr = 0.000005; epoch = 23/40	降低学习率	0.0080	0.0304	0.06817
ResNet50+ Dropout 0.75+ Dense 1024+ Dropout 0.5+ Dense 500+ Dropout 0.5; lr = 0.000001; epoch = 36/40	降低学习率 降低丢弃概率	0.0130	0.0157	0.06234
ResNet50+ Dropout 0.75+ Dense 1024+ Dropout 0.75+ Dense 500+ Dropout 0.75; lr = 0.000001; epoch = 36/40	提高丢弃概率	0.0210	0.0180	0.05916

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，故开放所有预训练层数参加新数据的训练。由于继续选用ResNet152时，出现运算资源不足，故改用更为简单的模型。

在初始尝试期间，用过AlexNet^[15]、VGG19^[11]等早期成名的模型。AlexNet是最早在ImageNet图像识别竞赛中成名的模型之一，也是开启了深度学习复兴热潮的先驱者。但这些模型虽然结构简单、训练较快，但效果都不如意，故选择ResNet50进行后续的深入调整。

如上文所述，当运算资源足够时可以增加数据增强的方式，人为扩大训练素材，也有利于使模型取得更好的效果。而接续的全连接层也可以有所提高，预计深度的增加也会对结果有利。

开始时选用了相对较高的学习率，但发现loss下降的过程中震荡较大，由于过早停止方法的运用，使其迅速结速了训练，最后的效果也并不理想。此后逐步降低学习率，最终达到更理想的测试集损失。

其间亦曾怀疑是否过于担心过拟合问题而尝试下调丢弃概率，虽测试集损失有所降低，但仍然不满足要求。在回升丢弃概率后，虽训练集和验证集损失有所回升，但测试集损失下降，也印证了此前的确仍存在过拟合的情况。最终测试集损失达到Kaggle排名前10%。

4 结果

4.1 模型的评价与验证

最终的模型采用ResNet50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ResNet50顶部全连接层的基础上，增加了一个全局平均池化层、三个丢弃概率为0.5的丢弃层、两个中间全连接层，和一个节点为1的输出全连接层，其激活函数为 *sigmoid*，用于最后的二元分类。

ResNet深度残差模型对传统深度卷积模型做了改进，使得深度相对较大的模型也更易于训练^[6]，为项目提供了很好的基础。而后自定义的各层经过调整，除最后用于二分类的全连接层外增加两个隐藏层是较好的方法，太复杂的模型极容易出现过拟合。

而随机丢弃节点的丢弃概率较难把握，在早期亦尝试过0.25、0.3、0.5等等，由于担心过拟合问题，更多地选择了较高的0.75。但经过多次尝试，确实提高丢弃概率能取得满意的结果，所以最终模型的三个丢弃层都用0.75。

在此模型结构中，亦曾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，所以第一阶段时，始终不愿降得更低。但后来大胆尝试的实验结果表明，学习率仍可降低，并达到了更好的效果。

4.2 合理性分析

最终模型得到测试集的损失为0.05916，达到Kaggle竞赛排名前10%水平，且比基准模型的0.32677降低了81.9%，而验证集的损失也从0.2594降至0.0180，降低了93.1%，性能有了明显的提升。

最终验证集的准确率也达到了99.46%，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。

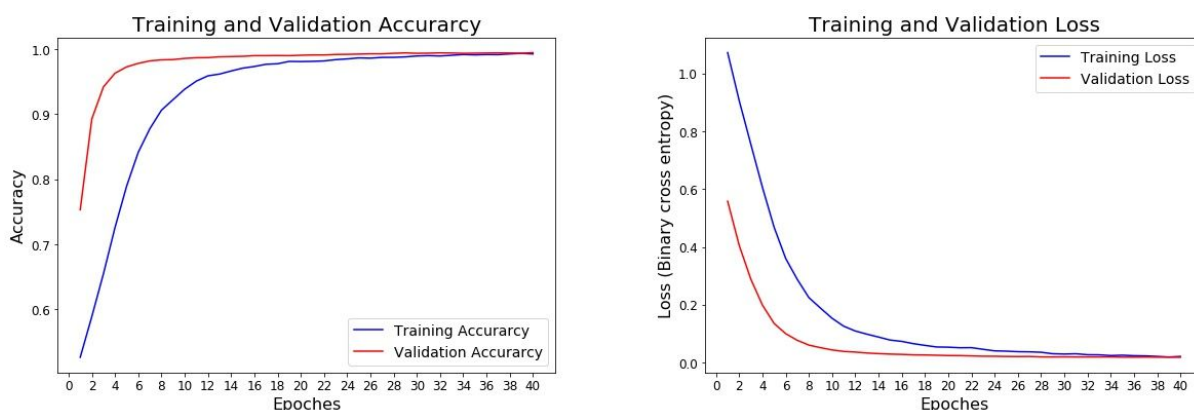


Fig 8. 最终模型训练过程中准确率和损失的变化

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，似有欠拟合的迹象。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，又似有过拟合的迹象。

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

理论上，模型结构更复杂、层数更多时，应该能使训练集的表现更好，在实验中也确实观察到这些迹象，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。

本项目最后采用了ResNet50模型，打开全部结构参与新数据的训练，取得了更好的效果，但测试集与训练集仍有一段差距，仍可能存在过一定的拟合问题。除了运用随机丢度法之外，数

据增强是更常用的方法，但这对运算资源例如内存有较高的要求，也会增加训练时间^[3]。所以在一定的硬件条件下，需要和其他方法做出权衡，例如本项目中牺牲了预想中更深的ResNet152模型。

所以条件充足时，可以增加训练层数，构建更深的模型寻找一些外部数据集，同时扩大训练样本，并实行充分的数据增强方法，进一步降低测试集损失。

影响模型效果的原因中，还可能是训练集仍存在一定的噪声，例如图片背景的干扰，或图片主体是否清晰，是否有多个主体干扰了图像特征的提取等等。从数据预处理阶段筛选出的异常数据来看，这种情况是存在的。当时采用了三种模型未加新数据训练的结果，来找出可能的异常值。可想而知，这种方法的准确率并不能十分肯定。

所以有一种可能的策略便是，采用本次训练所得的模型，再次对训练集进行一次预测，查看是否有预测不准确的数据集，经过人眼检查之后剔除，重新训练。

但可想而知，对训练集太过挑剔时，可能会使模型不适用于预测一些主体丰富的生活图片，而要求其面对的未见过的图片也必须主体清晰。这种应用场景也不陌生，目前许多手机app上即时识别的应用，也要求拍摄者将主体限定在一定的视野内，保持主体清晰明确。

但这种场景可能对识别速度及便携性有更高的要求，可能需要对模型做出更多的调整，或许可以选用MobileNet^[16]进行迁移学习迁移学习训练。

参考文献

- [1] KAGGLE. Dogs vs. Cats Redux: Kernels Edition[EB/OL]. Kaggle, [2019-04-05]. <https://kaggle.com/c/dogs-vs-cats-redux-kernels-edition>.
- [2] PEREZ L, WANG J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning[J]. 2017.
- [3] RAWAT W, WANG Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review[J]. Neural Computation, 2017, 29(9): 2352–2449.
- [4] HEKLER A, UTIKAL J S, ENK A H, 等. Pathologist-level classification of histopathological melanoma images with deep neural networks[J]. European Journal of Cancer (Oxford, England: 1990), 2019, 115: 79–83.
- [5] 262588213843476. text: imagenet 1000 class idx to human readable labels (Fox, E., & Guestrin, C. (n.d.). Coursera Machine Learning Specialization.)[EB/OL]. Gist, [2019-04-16]. <https://gist.github.com/yrevar/942d3a0ac09ec9e5eb3a>.
- [6] HE K, ZHANG X, REN S, 等. Deep Residual Learning for Image Recognition[J]. arXiv:1512.03385 [cs], 2015.
- [7] SZEGEDY C, VANHOUCKE V, IOFFE S, 等. Rethinking the Inception Architecture for Computer Vision[J]. arXiv:1512.00567 [cs], 2015.
- [8] CHOLLET F. Xception: Deep Learning with Depthwise Separable Convolutions[J]. arXiv:1610.02357 [cs], 2016.
- [9] SHIN H, ROTH H R, GAO M, 等. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning[J]. IEEE Transactions on Medical Imaging, 2016, 35(5): 1285–1298.
- [10] DENG J, DONG W, SOCHER R, 等. ImageNet: A large-scale hierarchical image database[C]//2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL: IEEE, 2009: 248–255.
- [11] SIMONYAN K, ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. arXiv:1409.1556 [cs], 2014.
- [12] KINGMA D P, BA J. Adam: A Method for Stochastic Optimization[J]. 2014.
- [13] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, 等. Dropout: A Simple Way to Prevent Neural Networks from Overfitting[J]. Journal of Machine Learning Research, , 15(1): 1929–1958.
- [14] HINTON G E, SRIVASTAVA N, KRIZHEVSKY A, 等. Improving neural networks by preventing co-adaptation of feature detectors[J]. arXiv:1207.0580 [cs], 2012.
- [15] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet Classification with Deep Convolutional Neural Networks[J]. Commun. ACM, 2017, 60(6): 84–90.
- [16] HOWARD A G, ZHU M, CHEN B, 等. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. arXiv:1704.04861 [cs], 2017.