机器学习进阶纳米学位
开题报告

曹小虎

2018 年 8 月 16 日

目录

1 背景 **2**
1.1 简介计算机视觉 2
1.2 相关的研究 3

2 问题描述 **3**
2.1 猫狗大战问题 3
2.2 个人动机 4

3 数据描述 **4**
3.1 简介 4
3.2 使用 7

4 解决方案 **7**
4.1 神经元 7
4.2 神经网络 8
4.3 卷积 9
4.4 卷积神经网络 10

5 基准模型 **11**

6 评估指标 **12**
6.1 一般对数损失函数 12

6.2 二分类任务对数损失函数 13

7 项目设计 13

7.1 探索DenseNet 13
7.2 下载预训练模型 16
7.3 项目目录结构 17
7.4 异常数据 18
7.5 模型建构 18
7.6 模型优化 19
7.7 模型评价 19

1 背景

1.1 简介计算机视觉

计算机视觉（Computer Vision）又称为机器视觉（Machine Vision），顾名思义是一门“教”会计算机如何去“看”世界的学科。作为人工智能的研究方向之一，它的主要目的是：让计算机能识别、分辨甚至理解不同的图像、视频及复杂场景。该方向的研究成果将会在自动驾驶、机器人、AR/VR、金融、安防以及医疗领域均得到应用。

表 2：计算机视觉技术在众多领域得到应用

应用领域	应用简述
自动驾驶（无人车、无人机）	计算机视觉技术在自动驾驶中解决的问题主要是障碍物检测和道路检测，尤其是在驾驶过程中的实时动态检测，对计算机判断的速度和准确性要求都很高。
机器人	训练机器人“能听会说能看”的能力显然离不开计算机视觉，除了需要 SLAM 即时定位与地图构建技术，在对象识别上主要依赖图像及场景的分类与分割等基本技术。
AR/VR	在物体跟踪、动作识别、三维环境建模等方面都需要用到计算机视觉技术。例如“虚拟试衣应用”主要利用计算机视觉技术对人体和服装进行建模，然后做在线试衣、穿搭，未来的游戏甚至可能将现实世界扫描、建模、完成对象识别，最后生成游戏地图。
金融	基于人脸识别、指纹识别以及指静脉、虹膜等生物识别技术对目标任务进行身份鉴定，从而在开户、支付等环节提供更为安全、便利、高效的服务。
安防	车牌、车辆识别在公安、交管领域应用较早也相对成熟，人脸识别目前应用还相对有限，更多在门禁系统、ATM 监控等简单任务，未来在刑事案件侦查、特定人员追踪、嫌疑人报警灯领域将发挥更大的作用。
医疗	计算机视觉在医疗领域的应用主要表现为对医学影像的识别，从而辅助医生对患者进行诊疗。

数据来源：东方证券研究所

1.2 相关的研究

类似人类自身这种与生俱来、再基本不过的“看”并且“理解”世界的能力——即使是专家学者也不能完全搞清楚其中深层次的内在机理。事实上单是这种教计算机像人类一样“看”并且“理解”现实世界的问题，虽然看上去简单，却还是花费了许多专家学者数十年时间。

近年来卷积神经网络(Convlutional Nerual Network)的深度学习模型在计算机视觉领域得到广泛应用，取得了令人惊喜的应用成果。

相比传统神经网络，卷积神经网络体积更小，能力更强。得益于现在强大的GPU并行运算能力，卷积神经网络已经由最开始的8层的AlexNet，到16层的VGGNet，再到152层ResNet，甚至更高，在ImageNet比赛中所取得的成绩也越来越优秀，自从2012年Hinton团队首次在该赛事中使用深度学习以来，Top5分类错误率已经从26.1%降低到不足3%。

表 1：2012 年以来 ImageNet 图像分类大赛冠军成绩

时间	机构	结构层数	Top5 错误率
2012.10.13	多伦多大学	8	16.4%
2013.11.14	纽约大学	7	11.7%
2014.08.18	Google	22	6.66%
2015.12.10	微软亚洲研究院	152	3.57%
2016.9.26	公安部三所		2.99%

数据来源：互联网，arXiv.org，东方证券研究所

国外计算机视觉相关的研究，除了耳熟能详的Google、微软、Facebook等工业界科技巨头有所涉及之外，一些著名高校也设有专门的实验室，如斯坦福、麻省理工以及伯克利等。

2 问题描述

2.1 猫狗大战问题

猫狗大战是Kaggle的一个图片分类竞赛项目。

它在计算机视觉领域是一个有监督学习的二分类问题。

我们要训练一个基于卷积神经网络的图片二分类模型，并且期望模型的图片分类能力(预测精确度)胜过人类。

2.2 个人动机

告别移动互联网时代，我们将要迎来人工智能时代。人工智能技术的将会分阶段、有步骤的在现实生活中得到应用。

计算机视觉技术让机器像人类一样能够感知现实世界，进而让计算机将原来只有人工的办法、花费大量人力才能完成的工作自动化起来，实现了生产效率的巨大飞跃。

> “计算机视觉是机器智能与企业转型的关键要素之一。”
>
> ——李飞飞

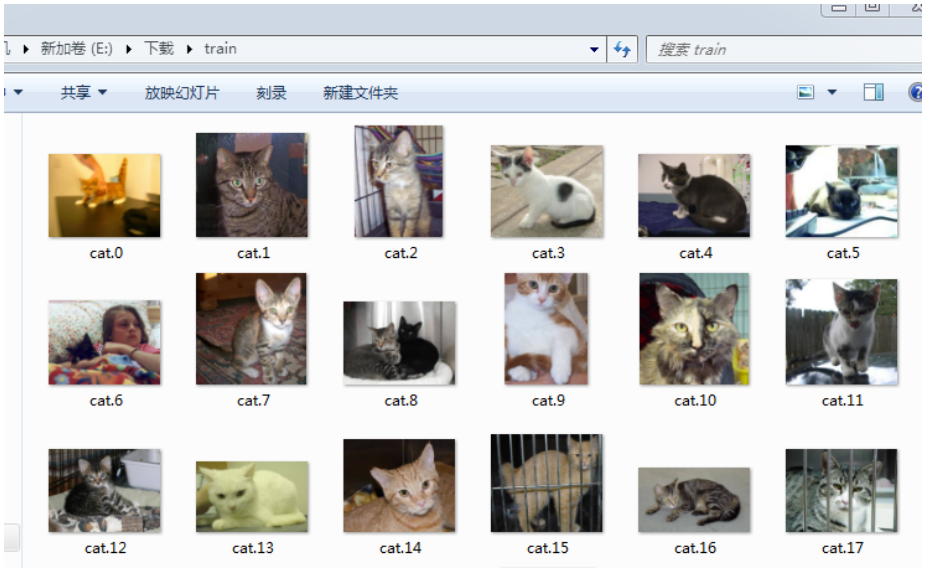
选择猫狗大战项目，将会帮助我深入理解图像分类这个问题领域中基于卷积神经网络的深度学习模型的工作原理。为自身进一步的学习研究打下良好基础。

3 数据描述

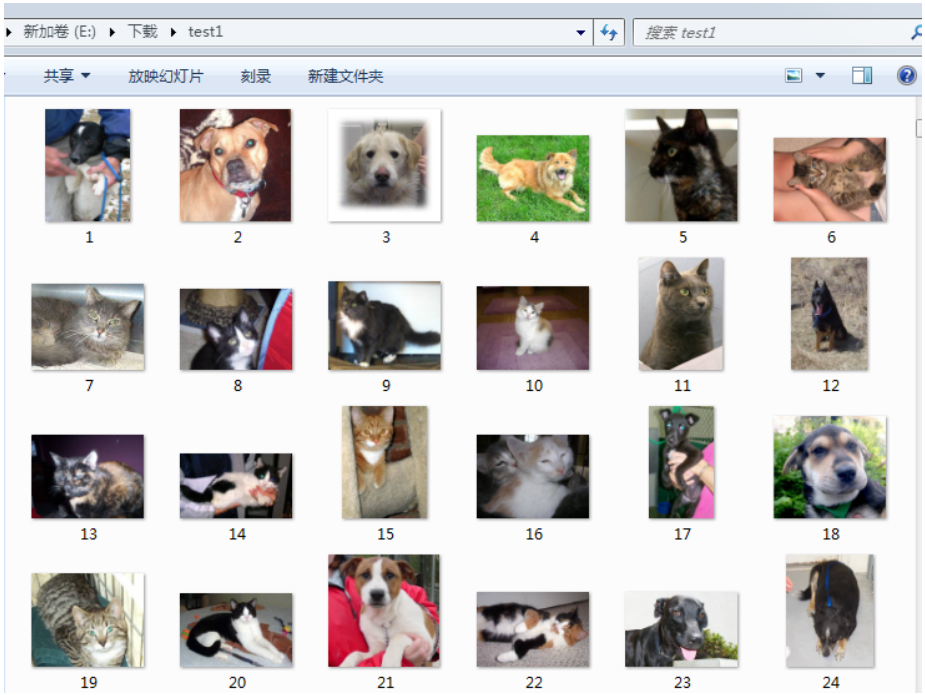
3.1 简介

首先我们需要在Kaggle网站注册自己的用户，然后才能下载到数据集。

在该项目中，kaggle上提供一个名称为train.zip的文件压缩包，其中包含了25,000张图片，其中猫和狗各12,500张，每一张图片都标记了内容类型。这个包里的图片被用于训练卷积神经网络的分类模型。



kaggle还提供了一个名称为test1.zip的文件压缩包，其中包含了12,500张没有标记内容类型的图片，用于测试训练好的分类模型的预测精度。



进一步观察，可以发现图片的像素大小不一致，随后在图片与处理的时候，我们可以对图片像素进行缩放。

也有像这样不够清晰的图像,就像下面这张图片：

对于这种肉眼看来比较模糊的图像，我认为应该保留，原因有二：其一，如果训练集里有几千张这种图片，凭借肉眼意义识别并且手工删除是一件耗时耗力、得不偿失的工作；其二，要想训练一个分类识别能力可以

与人相提并论的CNN模型，我们就有理由要求，凡是人类肉眼可以识别并进行分类的图片，机器模型也应该应该完成同样工作。

3.2 使用

在训练阶段，我们需要将train目录下的图片连带其标记输入到给定结构的卷积神经网络，训练它的图片分类能力。

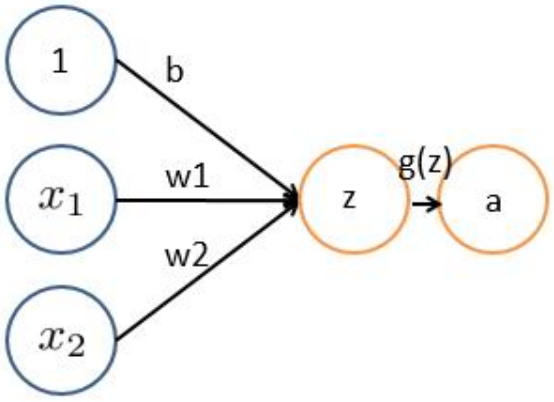
在测试阶段，我们拿test1目录下的图片来检测和评价卷积神经网络在进行图片内容类型预测时的精确程度。

4 解决方案

4.1 神经元

神经网络由大量的神经元相互连接而成。每个神经元接受线性组合的输入后，最开始只是简单的线性加权，后来给每个神经元加上了非线性的激活函数，从而进行非线性变换后输出。每两个神经元之间的连接代表加权值，称之为权重（weight）。不同的权重和激活函数，则会导致神经网络不同的输出。

举个手写识别的例子，给定一个未知数字，让神经网络识别是什么数字。此时的神经网络的输入由一组被输入图像的像素所激活的输入神经元所定义。在通过非线性激活函数进行非线性变换后，神经元被激活然后被传递到其他神经元。重复这一过程，直到最后一个输出神经元被激活。从而识别当前数字是什么字。



基本wx + b的形式，其中

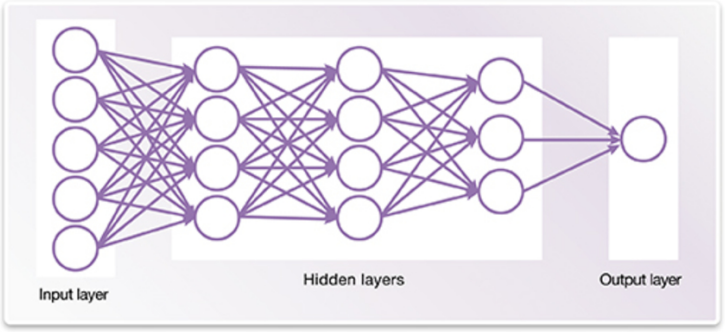
- x 表示输入向量
- w 为权重，几个输入则意味着有几个权重，即每个输入都被赋予一个权重
- b为偏置bias
- g(z) 为激活函数
- a 为输出

事实上，上述简单模型可以追溯到20世纪50/60年代的感知器，可以把感知器理解为一个根据不同因素、以及各个因素的重要性程度而做决策的模型。

一开始为了简单，人们把激活函数定义成一个线性函数，即对于结果做一个线性变化，比如一个简单的线性激活函数是g(z) = z，输出都是输入的线性变换。后来实际应用中发现，线性激活函数太过局限，于是人们引入了非线性激活函数。

4.2 神经网络

组织在一起，便形成了神经网络。下图便是一个三层神经网络结构:



上图中最左边的原始输入信息称之为输入层，最右边的神经元称之为输出层（上图中输出层只有一个神经元），中间的叫隐藏层。

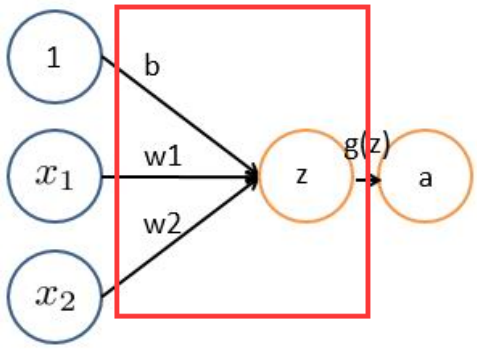
- Input layer，众多神经元（Neuron）接受大量非线形输入讯息。输入的讯息称为输入向量。

- Output layer，讯息在神经元链接中传输、分析、权衡，形成输出结果。输出的讯息称为输出向量。
- Hidden layer，简称“隐层”，是输入层和输出层之间众多神经元和链接组成的各个层面。如果有多个隐藏层，则意味着多个激活函数。

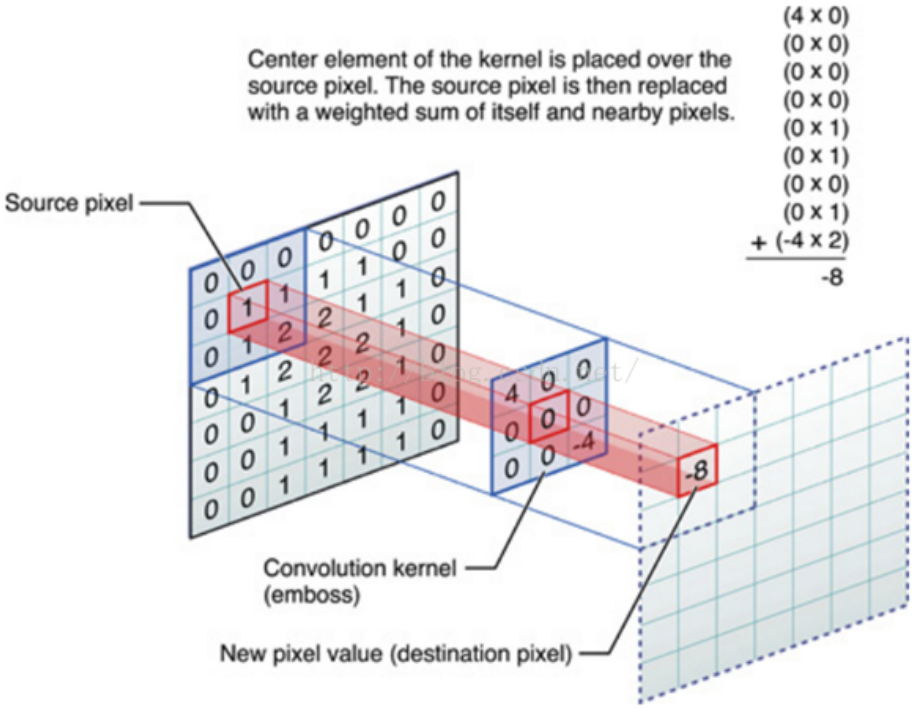
4.3 卷积

对图像（不同的数据窗口数据）和滤波矩阵（一组固定的权重：因为每个神经元的多个权重固定，所以又可以看做一个恒定的滤波器filter）做内积（逐个元素相乘再求和）的操作就是所谓的『卷积』操作，也是卷积神经网络的名字来源。

非严格意义上来讲，下图中红框框起来的部分便可以理解为一个滤波器，即带着一组固定权重的神经元。每一个这样的滤波器都被成为一个卷积核。滤波器的宽度和高度被称为“卷积核大小”(kernel size)。



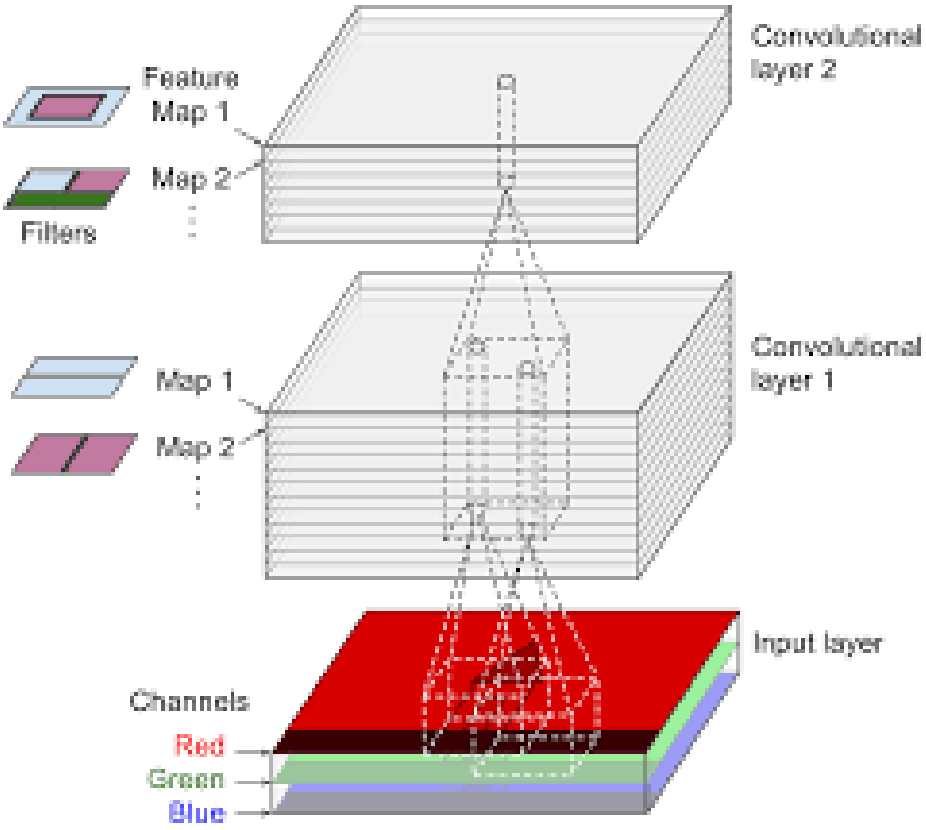
下面的图例说明了卷积核如何从原始像素矩阵中过滤信息:



上面图中的卷积核(滤波器)与数据窗口做内积，其具体计算过程则是：4*0 + 0*0 + 0*0 + 0*0 + 0*1 + 0*1 + 0*0 + 0*1 + -4*2 = -8。

4.4 卷积神经网络

卷积神经网络就中间隐藏层包含了卷积层的神经网络。

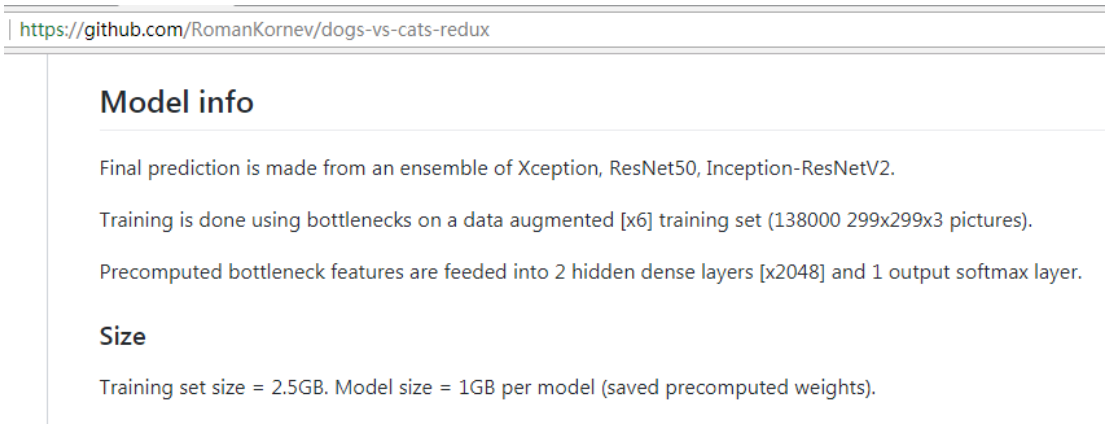


以图片分类任务为例，卷积层用(多个)不同的卷积核,对代表输入图片的数字矩阵进行卷积运算，继而从图片中提取到多个特征图(Feature Map)。就像人类的神经网络一样，由前一个卷积层提取图像的低层次特征，由后一个卷积层把这些低层次特征组合起来，提取出更高层次的特征。

5 基准模型

在所有参与Kaggle数据猫狗大战图像分类竞赛的选手中，我挑选到一个表现好的深度卷积网络模型作为参考基准。请点击链接查看有关github项目页面，本文只对模型本身进行最必要的介绍和说明。

模型的基本情况结构如下：



https://github.com/RomanKornev/dogs-vs-cats-redux

Model info

Final prediction is made from an ensemble of Xception, ResNet50, Inception-ResNetV2.

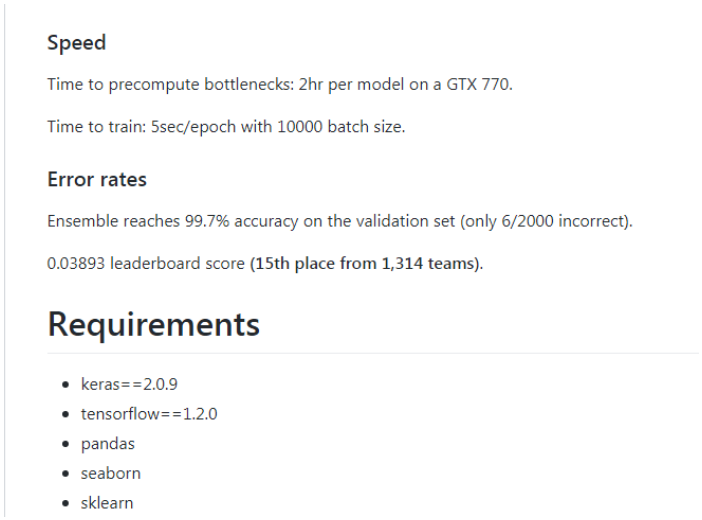
Training is done using bottlenecks on a data augmented [x6] training set (138000 299x299x3 pictures).

Precomputed bottleneck features are feeded into 2 hidden dense layers [x2048] and 1 output softmax layer.

Size

Training set size = 2.5GB. Model size = 1GB per model (saved precomputed weights).

模型在测试数据集上的预测分类表现情况如下：



Speed

Time to precompute bottlenecks: 2hr per model on a GTX 770.

Time to train: 5sec/epoch with 10000 batch size.

Error rates

Ensemble reaches 99.7% accuracy on the validation set (only 6/2000 incorrect).

0.03893 leaderboard score **(15th place from 1,314 teams)**.

Requirements

- keras==2.0.9
- tensorflow==1.2.0
- pandas
- seaborn
- sklearn

6 评估指标

6.1 一般对数损失函数

对数损失, 即对数似然损失(Log-likelihood Loss), 也称逻辑斯谛回归损失(Logistic Loss)或交叉熵损失(cross-entropy Loss), 是在概率估计上定义的.它常用于(multi-nominal, 多项)逻辑斯谛回归和神经网络,以及一些期望极大算法的变体. 可用于评估分类器的概率输出。

对数损失通过惩罚错误的分类,实现对分类器的准确度(Accuracy)的量

化. 最小化对数损失基本等价于最大化分类器的准确度.为了计算对数损失, 分类器必须提供对输入的所属的每个类别的概率值, 不只是最可能的类别. 对数损失函数的计算公式如下:

$$L(Y, P(Y|X)) = -\log P(Y|X) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

其中, Y 为输出变量, X为输入变量, L 为损失函数. N为输入样本量, M为可能的类别数, yij 是一个二值指标, 表示类别 j 是否是输入实例 xi 的真实类别. pij 为模型或分类器预测输入实例 xi 属于类别 j 的概率.

6.2 二分类任务对数损失函数

在处理二分类问题的时候, 则对数损失函数的公式简化为:

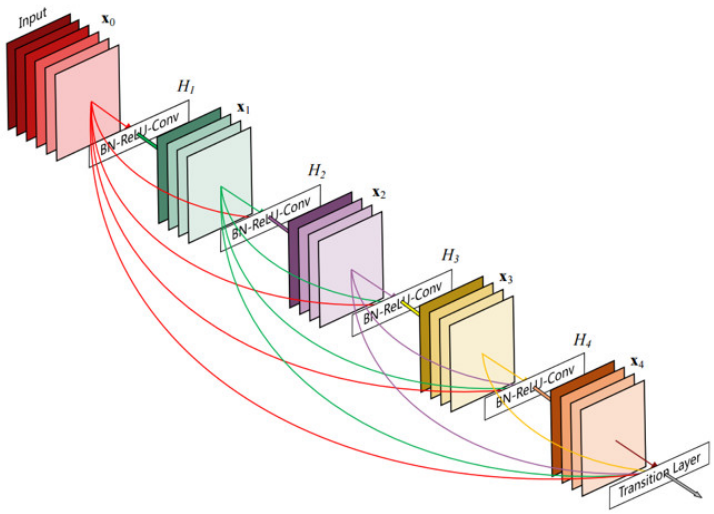
$$\text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]$$

在上面的表达式中，n为测试集中图片的数量,yi_hat为预测图中动物为狗的概率。如果实际图中的动物是狗，则yi=1, 否则yi=0； yi 为输入实例 xi 的真实类别, $\hat{yi}$ 为预测输入实例 xi 属于类别 1 的概率。

7 项目设计

7.1 探索DenseNet

在计算机视觉领域，卷积神经网络（CNN）已经成为最主流的方法，比如最近的GoogLenet，VGG-19，Incepetion等模型。CNN史上的一个里程碑事件是ResNet模型的出现，ResNet可以训练出更深的CNN模型，从而实现更高的准确度。ResNet模型的核心是通过建立前面层与后面层之间的“短路连接”（shortcuts，skip connection），这有助于训练过程中梯度的反向传播，从而能训练出更深的CNN网络。



DenseNet模型示意图如上图。它的基本思路与ResNet一致，但是它建立的是前面所有层与后面层的密集连接（dense connection）。在该网络中，任何两层之间都有直接的连接，即网络每一层的输入都是前面所有层输出的并集，而该层所学习的特征图也会被直接传给其后面所有层作为输入。

DenseNet的另一大特色是通过特征在channel上的连接来实现特征重用（feature reuse）。这些特点让DenseNet在参数和计算成本更少的情形下实现比ResNet更优的性能，DenseNet也因此斩获CVPR 2017的最佳论文奖。

它在ImageNet图片分类任务的验证集上，DenseNet的Top-1和Top-5错误率如图所示:

Model	top-1	top-5
DenseNet-121 ($k=32$)	25.02 (23.61)	7.71 (6.66)
DenseNet-169 ($k=32$)	23.80 (22.08)	6.85 (5.92)
DenseNet-201 ($k=32$)	22.58 (21.46)	6.34 (5.54)
DenseNet-161 ($k=48$)	22.33 (20.85)	6.15 (5.30)

Table 3: The top-1 and top-5 error rates on the ImageNet validation set, with single-crop (10-crop) testing.

在ImageNet图片分类任务的验证集上，DenseNet与ResNet的Top-1错误率变化走势对比如图所示：

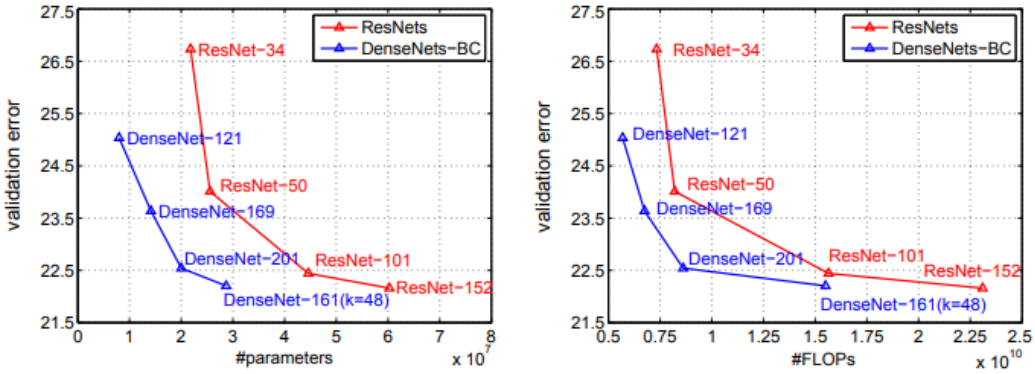


Figure 3: Comparison of the DenseNets and ResNets top-1 error rates (single-crop testing) on the ImageNet validation dataset as a function of learned parameters (*left*) and FLOPs during test-time (*right*).

下面的表格，对比了ResNet各种变体和DenseNet各种变体在C10数据集和C100数据集上的错误率，同时也显示了各自的网络深度和参数总量:

Method	Depth	Params	C10	C10+	C100	C100+
Network in Network [22]	-	-	10.41	8.81	35.68	-
All-CNN [31]	-	-	9.08	7.25	-	33.71
Deeply Supervised Net [20]	-	-	9.69	7.97	-	34.57
Highway Network [33]	-	-	-	7.72	-	32.39
FractalNet [17]	21	38.6M	10.18	5.22	35.34	23.30
with Dropout/Drop-path	21	38.6M	7.33	4.60	28.20	23.73
ResNet [11]	110	1.7M	-	6.61	-	-
ResNet (reported by [13])	110	1.7M	13.63	6.41	44.74	27.22
ResNet with Stochastic Depth [13]	110	1.7M	11.66	5.23	37.80	24.58
	1202	10.2M	-	4.91	-	-
Wide ResNet [41]	16	11.0M	-	4.81	-	22.07
	28	36.5M	-	4.17	-	20.50
with Dropout	16	2.7M	-	-	-	-
ResNet (pre-activation) [12]	164	1.7M	11.26*	5.46	35.58*	24.33
	1001	10.2M	10.56*	4.62	33.47*	22.71
DenseNet ($k = 12$)	40	1.0M	**7.00**	5.24	**27.55**	24.42
DenseNet ($k = 12$)	100	7.0M	**5.77**	**4.10**	**23.79**	**20.20**
DenseNet ($k = 24$)	100	27.2M	**5.83**	**3.74**	**23.42**	**19.25**
DenseNet-BC ($k = 12$)	100	0.8M	**5.92**	4.51	**24.15**	22.27
DenseNet-BC ($k = 24$)	250	15.3M	**5.19**	**3.62**	**19.64**	**17.60**
DenseNet-BC ($k = 40$)	190	25.6M	-	**3.46**	-	**17.18**

综合来看，DenseNet的优势主要体现在以下几个方面：

- 由于密集连接方式，DenseNet提升了梯度的反向传播，使得网络更容易训练。由于每层可以直达最后的误差信号，实现了隐式的“deep supervision”；
- 参数更小且计算更高效，这有点违反直觉，由于DenseNet是通过concat特征来实现短路连接，实现了特征重用，并且采用较小的growth rate，每个层所独有的特征图是比较小的；
- 由于特征复用，最后的分类器使用了低级特征。

要注意的一点是，如果实现方式不当的话，DenseNet可能耗费很多GPU显存，有关高效的实现更多细节可以见这篇论文Memory-Efficient Implementation of DenseNets。

7.2 下载预训练模型

有Github如下链接可以下载ImageNet数据集上预先训练好的DenseNet-

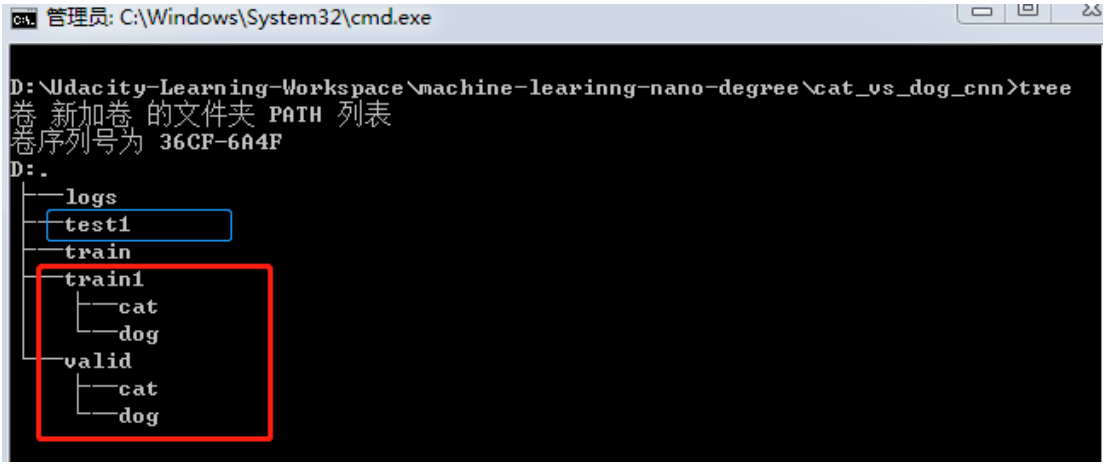
161模型文件。

7.3 项目目录结构

按照以下步骤搭建项目的工作目录。

- 新建目录cat_vs_dog_cnn
- 从kaggle网站下载train.zip和test1.zip两个压缩包，保存到目录cat_vs_dog_cnn
- 在cat_vs_dog_cnn目录，解压缩train.zip压缩包到名字是train的文件夹
- 在cat_vs_dog_cnn目录新建名称为train1的训练集目录，在train1中分别新建名称为cat和dog的目录
- 在cat_vs_dog_cnn目录创建名称为valid的验证集目录，在valid目录中新建名称为cat和dog的目录
- 把cat_vs_dog_cnn/train目录中的前3000张猫图复制到cat_vs_dog_cnn/valid/cat目录下
- 把cat_vs_dog_cnn/train目录中的前3000张狗图复制到cat_vs_dog_cnn/valid/dog目录下
- 把cat_vs_dog_cnn/train目录中的剩下的猫图复制到cat_vs_dog_cnn/train1/cat目录下
- 把cat_vs_dog_cnn/train目录中的剩下的狗图复制到cat_vs_dog_cnn/train1/dog目录下
- 在cat_vs_dog_cnn目录，解压test1.zip压缩包到名字是cat_vs_dog_cnn/test1的文件夹作为测试集目录

以下是目录结构:



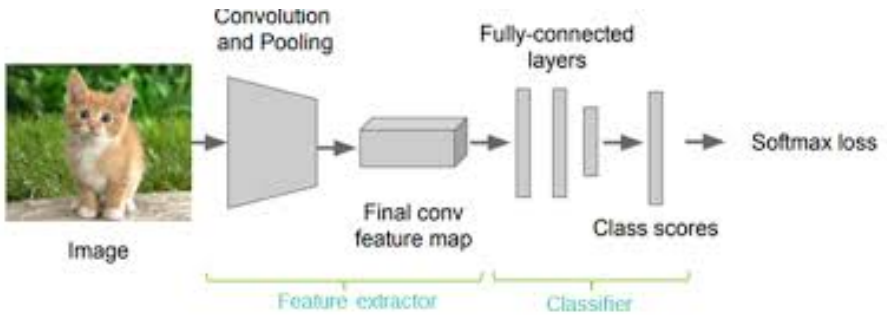
7.4 异常数据

在描述比赛用到数据的前面章节我们了解到训练集中图片的两种情况：

- 存在不同尺寸的图片: 这种情况可以通过图片缩放来统一输入图片的大小
- 显示模糊的图片: 只要人类肉眼能够区分猫狗，我们就有理由把它留在训练集中
- 如果发现非猫非狗的图片，就一定要把他们从训练集、验证集还有测试集图片中清理出去

7.5 模型建构

如下所示，一个CNN模型大致分为以下两大部分：



拿预先训练好的DenseNet-161模型为例，我们要复用它的部分模型架构和各层权重。具体分为以下几步：

- 保留Densenet-161模型中的卷积层、池化层，这些层主要是用于提取图像高层次特征特征(feature map)
- 移除在DenseNet-161上包含有全连接层、最终用于输出最终分类结果的分类器
- 顺序连接好自定义的包含有全连接层的分类器
- 冻结DenseNet-161原有的feature extractor部分，用猫狗大战项目的图片数据集训练自定义的分类器(custom classifier)

7.6 模型优化

在训练自定义分类器的过程中，我们可以尝试以下方法提高模型表现：

- 使用全连接层：我们可以改变全连接层的个数，观察不同情况下模型在验证集上的分类效果。它的作用是学习从前边feature extractor得到的关于输入图片的各个局部特征的综合起来(线性组合)，作为分类器(classifier)做出最终分类决定。
- 正则化：Regilarization(Dropout)技术,防止模型过拟合，提升模型在验证集和测试集上的泛化能力
- 数据增强技术：通过数据增强(Data Augmentation)，可以在修改现有训练集图片的基础上获得更多训练图片，从而提升模型的泛华能力
- 我们可以参考参考文献[3]中描述的方法，把多个一直表现优秀的分类模型融合在一起，训练一个自己的最终模型。

7.7 模型评价

模型训练完毕以后，我们就可以在测试集上验证模型的分类预测能力了。本项图片分类比赛最终评估的是测试集上的对数损失函数值。

参考文献

[1] 康奈尔大学博士后黄高博士（Gao Huang）、清华大学本科生刘壮（Zhuang Liu）、Facebook 人工智能研究院研究科学家 Laurens van der Maaten 及康奈尔大学计算机系教授 Kilian Q. Weinberger所作论文Densely Connected Convolutional Networks

[2] Authors: Sihan Li, Jiantao Jiao, Yanjun Han, Tsachy Weissman，Paper: Demystifying ResNet

[3] 介绍多个成熟模型迁移学习并且取得不俗战绩的Github仓库: Dogs vs Cats - Predict whether a given image is of a cat or a dog with 99.7% accuracy

[4] 介绍DenseNet模型的Github仓库: DenseNet-Keras with ImageNet Pre-trained Models

[5] Authors: Takuya Akiba, Shuji Suzuki, Keisuke Fukuda，Paper: Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes

[6] Authors: Hussam Qassim, David Feinzimer, Abhishek Verma，Paper: Residual Squeeze VGG16

[7] Authors: Hao Peng, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, Zhi Jin(Software Institute, Peking University, 100871, P. R. China)，Paper: A Comparative Study on Regularization Strategies for Embedding-based Neural Networks

[8] 有关迁移学习，Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson，Paper: How transferable are features in deep neural networks

[9] Authors: Tao Tan, Zhang Li, Haixia Liu, Ping Liu, Wenfang Tang, Hui Li, Yue Sun, Yusheng Yan, Keyu Li, Tao Xu, Shanshan Wan, Ke Lou, Jun Xu, Huiming Ying, Quchang Ouyang, Yuling Tang, Zheyu Hu, and Qiang Li，Paper: Optimize transfer learning for lung diseases in bronchoscopy using a new concept: sequential fine-tuning