机器学习纳米学位

毕业项目

Dogs vs. Cats

毛世杰
2017年5月27日

I. 问题的定义

项目概述

计算机视觉是机器学习的一个很重要的应用领域。计算机视觉是一门研究如何使机器“看”的科学，更进一步的说，就是是指用摄影机和计算机代替人眼对目标进行识别、跟踪和测量等机器视觉。作为一个科学学科，计算机视觉研究相关的理论和技术，试图建立能够从图像或者多维数据中获取‘信息’的人工智能系统。这里所指的信息指Shannon定义的，可以用来帮助做一个“决定”的信息。因为感知可以看作是从感官信号中提取信息，所以计算机视觉也可以看作是研究如何使人工系统从图像或多维数据中“感知”的科学。
图像识别（image recognition）是计算机视觉的一个领域。我们的大脑做图像识别是很容易。人类可以很容易的区别一只狮子和一只美洲虎，识别一个标志，或认出一个人的脸。但对计算机视觉来说这些都是一个难题。近几年机器学习在解决这些困难问题上取得了巨大进步。特别地，我们发现一种称为深度卷积神经网络（CNN）的模型在困难的视觉识别任务上有很好的性能。研究人员不断地发明新的CNN模型。从早期的LeNet，AlexNet 到最新的 Inception v4，ResNeXt，这些模型在ImageNet数据集上取得了越来越好的正确率。
Dogs vs. Cats最早来源于Kaggle的一个竞赛。现在已经演变为流行的图像识别初学者练习项目。

问题陈述

Dogs vs. Cats这个项目是一个监督学习的问题。训练数据同时包含输入图片和输出的标记数据。我们需要训练一个二分类分类器。分类器的目标是判断你输入的图片是Dog还是Cat。

项目包含以下几个部分：

1. 从kaggle下载Dogs vs. Cats数据集
2. 预处理数据：
    1. 把train文件夹中的数据分割为训练数据（train dataset）和校验数据（validation dataset）
3. 构建分类器
    1. 分类器使用ResNet-50 CNN模型
    2. 针对Dogs vs. Cats做少量调整
4. 训练分类器
    1. 使用Dog vs. Cats的数据集训练CNN模型
    2. 使用迁移学习的方法提高训练的速度和预测的精度
5. 使用分类器判断测试集合的图片是Dog还是Cat

模型评估

项目使用准确率（Accuracy）和log loss作为最终的模型评估方法。准确率和log loss是分类器常用的评价指标。

$$accuray = \frac{n}{N} * 100\,\%$$

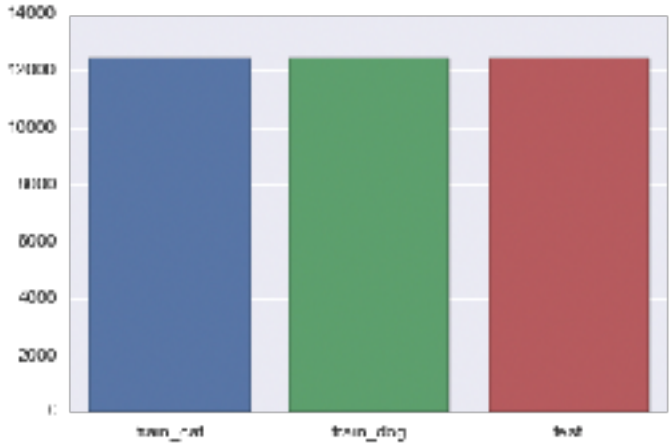
- n是分类器判断正确的dog和cat的图片数据
- N是数据集总体的图片数量

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(\hat{y}_i) + (1 - y_1) log(1 - \hat{y}_i)))$$

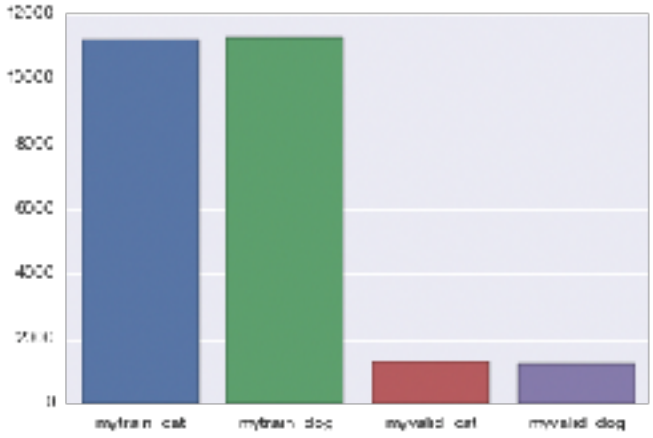
- N是测试集中图片的数量
- y_i是图片的真实Label，Dog为1，Cat为0
- $\hat{y}_i$是分类器预测图片为Cat的概率

II. 分析

数据的探索



原始数据集数量



分拆后的数据集数量

训练数据包含25,000张猫和狗的图片，每张图片的文件名标示了这是猫还是狗。测试数据包含12,500张猫和狗的图片。

数据集包含两个文件夹train和test：

- train文件夹中文件名为[doglcat].n.jpg，文件名标示了样本为dog和cat。
- test文件文件名为n.log，没有样本的类别信息。

把train文件夹拆分为两个文件夹：

Dog

Cat

- mytrain：包含两个文件夹
  - cat：11250张猫的图片
  - dog：11250张狗的图片
- myvalid：包含两个文件夹
  - cat：1250张猫的图片
  - dog：1250张狗的图片

探索性可视化

数据集中图片的分辨率，清晰度等规格都有所不同。
测试数据中也包含一些很难判别的图片：左边的图片同时包含猫和狗；右边的图片不是真实图片，太抽象。增加了分类判别的难度。

2420.jpg

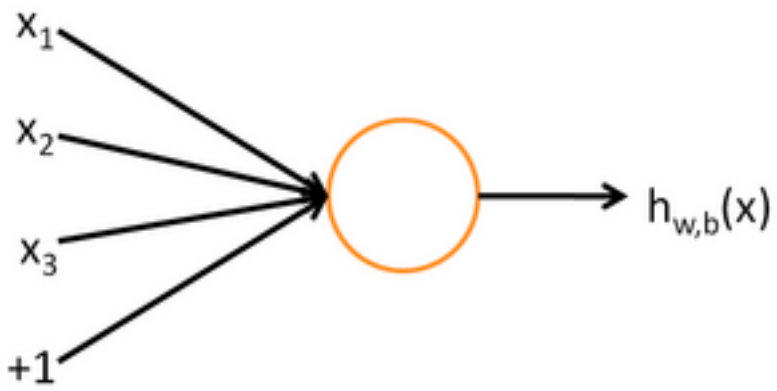
12324.jpg

原始数据集中猫和狗的图片数量是相等的。数据是平衡的，更方便训练分类器。
拆分validation数据后，依然保持数据集是平衡的。

算法和技术

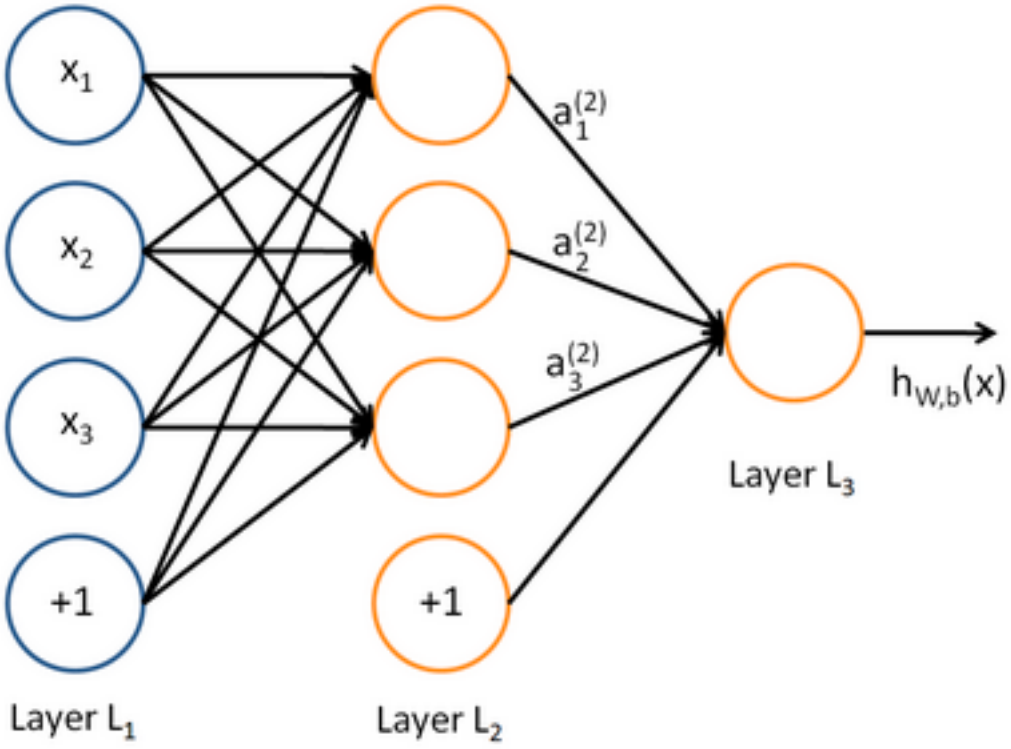
使用的分类器是ResNet-50。ResNet的全称是Deep Residual Networks。50表示模型网络结构包含50层。ResNet在图像识别领域的ImageNet 2015年的竞赛上取得了第一名的成绩，识别准确率较高。ImageNet是一个图像数据库，包还有很多类别的图片，每个类别都上万的图片数据。

ResNet是卷积神经网络（CNN）模型的一种。一个神经网络包含多个神经层，每个神经层包含多个神经元。每个神经元都包含一系列可以学习的权重（weights和biases）。每个神经元都可以接受一组输入，并计算输入值和对应权重的线性加权和，然后通过激活函数引入非线性，输出最终的值。



$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$$

最基础的神经网络是全连接网络。全连接网络中，每一层神经网络的神经元和下一层的神经元完全联接。这种结构在输入数据，比如图片数据，维度较大时，需要学习的权重数量急剧增加，学习成分很高。全联接网络结构如图所示：

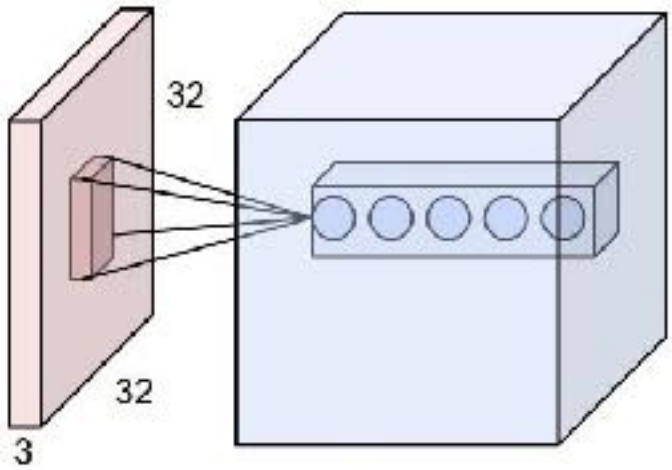


其对应的公式为：

$$
\begin{aligned}
a_1^{(2)} &= f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \\
a_2^{(2)} &= f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}) \\
a_3^{(2)} &= f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}) \\
h_{W,b}(x) = a_1^{(3)} &= f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})
\end{aligned}
$$

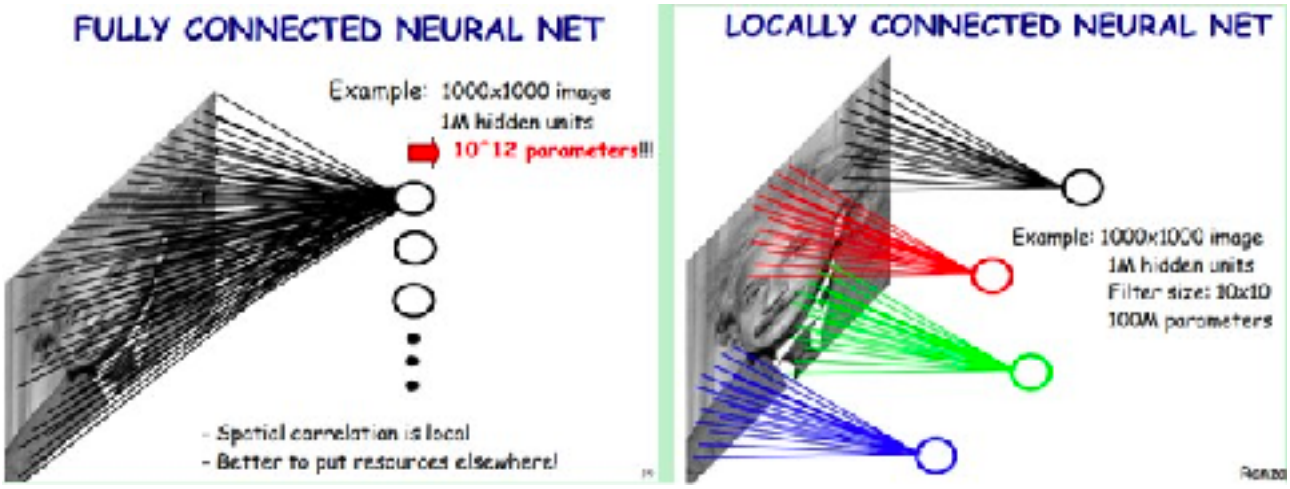
卷积神经网络

CNN网络中，每一层的神经元只和上一层神经元的一小部分联接。如图所示：



局部感知

卷积神经网络有两种方法可以降低参数数目，第一种方法叫做局部感知野。一般认为人对外界的认知是从局部到全局的，而图像的空间联系也是局部的像素联系较为紧密，而距离较远的像素相关性则较弱。因而，每个神经元其实没有必要对全局图像进行感知，只需要对局部进行感知，然后在更高层将局部的信息综合起来就得到了全局的信息。网络部分连通的思想，也是受启发于生物学里面的视觉系统结构。视觉皮层的神经元就是局部接受信息的（即这些神经元只响应某些特定区域的刺激）。如下图所示：左图为全连接，右图为局部连接：在上右图中，假如每个神经元只和10×10个像素值相连，那么权值数据为1000000×100个参数，减少为原来的千分之一。而那10×10个像素值对应的10×10个参数，其实就相当于卷积操作。



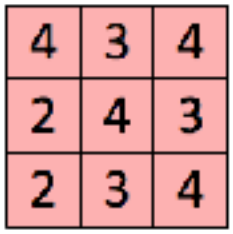
参数共享

但其实这样的话参数仍然过多，第二种方法即权值共享。在上面的局部连接中，每个神经元都对应100个参数，一共1000000个神经元，如果这1000000个神经元的100个参数都是相等的，那么参数数目就变为100了。
怎么理解权值共享呢？我们可以这100个参数（也就是卷积操作）看成是提取特征的方式，该方式与位置无关。这其中隐含的原理则是：图像的一部分的统计特性与其他部分是一样的。这也意味着我们在这一部分学习的特征也能用在另一部分上，所以对于这个图像上的所有位置，我们都能使用同样的学习特征。

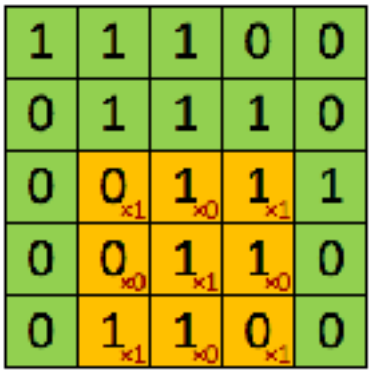
更直观一些，当从一个大尺寸图像中随机选取一小块，比如说 8×8 作为样本，并且从这个小块样本中学习到了一些特征，这时我们可以把从这个 8×8 样本中学习到的特征作为探测器，应用到这个图像的任意地方中去。特别是，我们可以用从 8×8 样本中所学习到的特征跟原本的大尺寸图像作卷积，从而对这个大尺寸图像上的任一位置获得一个不同特征的激活值。

如下图所示，展示了一个33的卷积核在55的图像上做卷积的过程。每个卷积都是一种特征提取方式，就像一个筛子，将图像中符合条件（激活值越大越符合条件）的部分筛选出来。

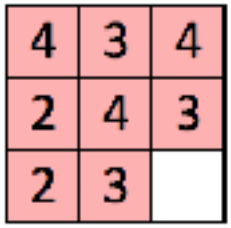
Image



Convolved Feature



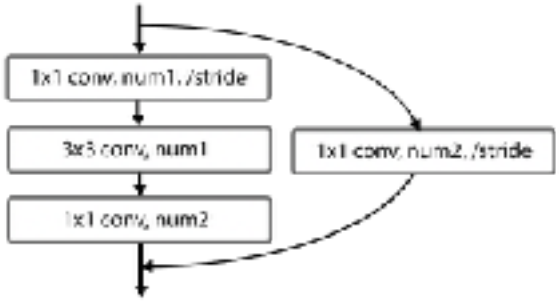
Image



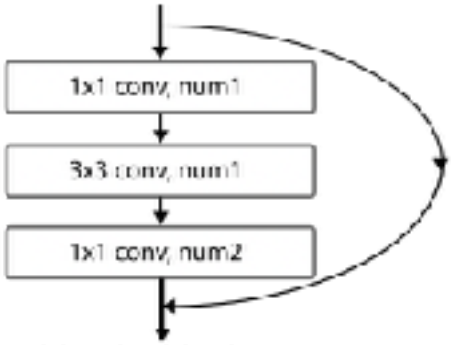
Convolved Feature

Resnet-50

Resnet-50包含一个7x7的卷基层，4个convolutional blocks，12个identity blocks，和一个1000维度的全联接层。
convolutional blocks和identity blocks的结构如下：

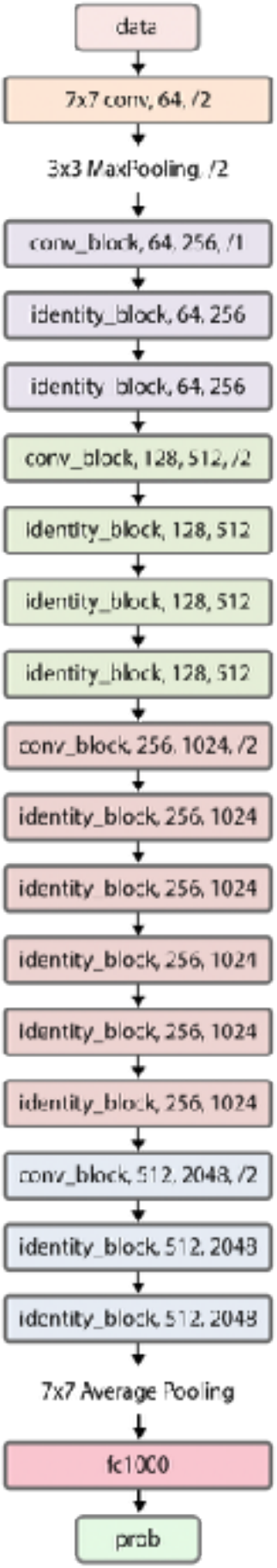


convolutional blocks

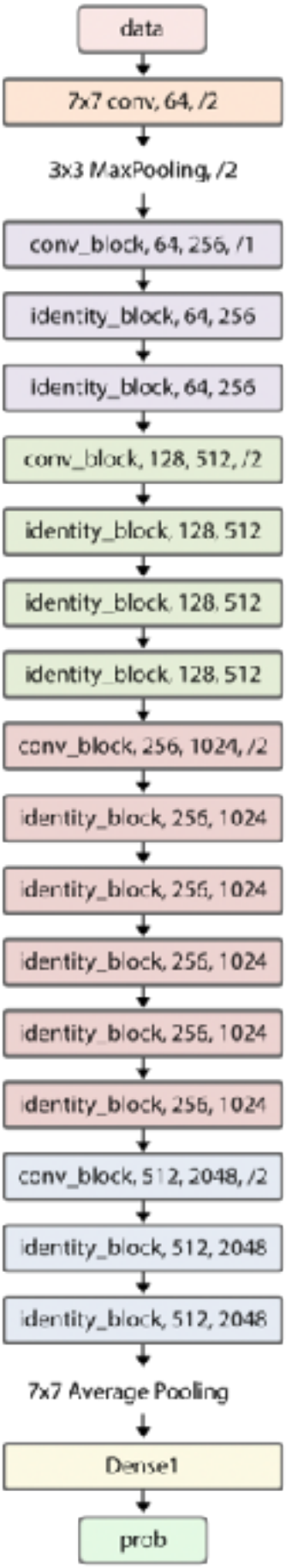


identity blocks

在Dog Cat项目中，预测的目标是2分类问题，所以在Resnet-50的最后1000维度的全联接层替换为1维度的全联接层。最终的网络结构如下：



ResNet-50



ResNet-50 Dogs vs. Cats

基准模型

Benchkmark： Golle, Philippe. "Machine learning attacks against the Asirra CAPTCHA." Proceedings of the 15th ACM conference on Computer and communications security. ACM, 2008.
Golle, Philippe构建了一个SVM分类器。在Asirra CAPTCHA上识别猫和狗，取得了82.7%的精确度。

不过上述基准实现的算法比较久远了。本项目的目标是在Dog vs. Cats数据集合上取得超过90%的准确率。

III. 方法

数据预处理

数据预处理包含如下几个步骤：
1. 把下载数据的train目录，拆分为训练数据和验证数据
2. 所有图片大小调整为224x224
3. 把RGB数据，从0-255归一化为0-1

执行过程

执行过程包含如下几个主要步骤：
1. 构建ResNet-50网络结构
    1. 定义identity block
    2. 定义convolution block
    3. 构建ResNet-50网络结果（除了最后一层全联接层）
    4. 加载在ImageNet上预训练的权重
    5. 添加最有一个层全联接层（维度1）
    6. 编译模型
2. 使用训练数据训练模型，并查看在验证集上的效果
3. 保存在验证集合上效果最好的模型
4. 使用最好的模型预测测试数据的Label

网络结构使用keras构建。keras是一个构建在tensorflow或者theano上高层深度学习库。keras提供大量神经网络组件与优化方法等工具。keras使得构建神经网络更简单。

完善

最开始的模型，使用层数更少、更简单的CNN模型；随机初始化模型权重，直接使用Dogs vs. Cats训练模型；但最终的效果不是很好。

然后使用了完整的ResNet-50 CNN模型，但是训练成本很高，要让模型收敛需要大量的计算，目前的单GPU主机训练时间过长，基本不可行。

最终的方案是使用迁移学习的方法。加载在ImageNet上预训练的权重，在修改最后的全联接层，使用Dogs vs. Cats数据集训练最后一层全连接层，取得了最终的结果。

IV. 结果

模型的评价与验证

前文提到，项目的目标是取得90%以上的准确率。最终使用ResNet-50加上在ImageNet上预训练的权重，达到并远超过了这个目标。具体的训练过程如下：

训练数据准确率　　**验证数据准确率**　　**验证数据loss**

从图中可以看出，模型在前10个epoch快速的收敛，学习得很快，准确率快速提升，loss也快速的减小。随后逐渐稳定，不再提高。最终的在验证集合上的准确率稳定在98%左右，loss稳定在0.06左右。

合理性分析

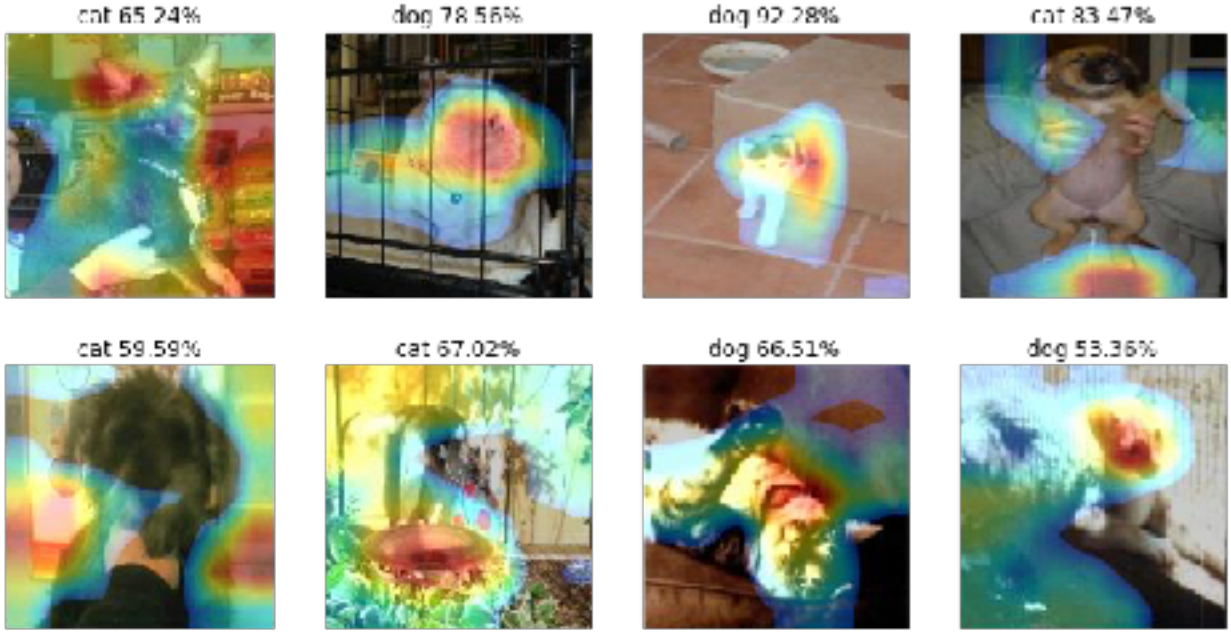
如前文所示，ResNet-50的网络结果足够复杂，对于Dog vs. Cat来说，已经完全足够。再加上在ImageNet上的预训练权重，模型可以快速的学习、收敛。

最终的准确率稳定在98%左右，loss稳定在0.06左右，是完全可信和合理的。

V. 项目结论

结果可视化

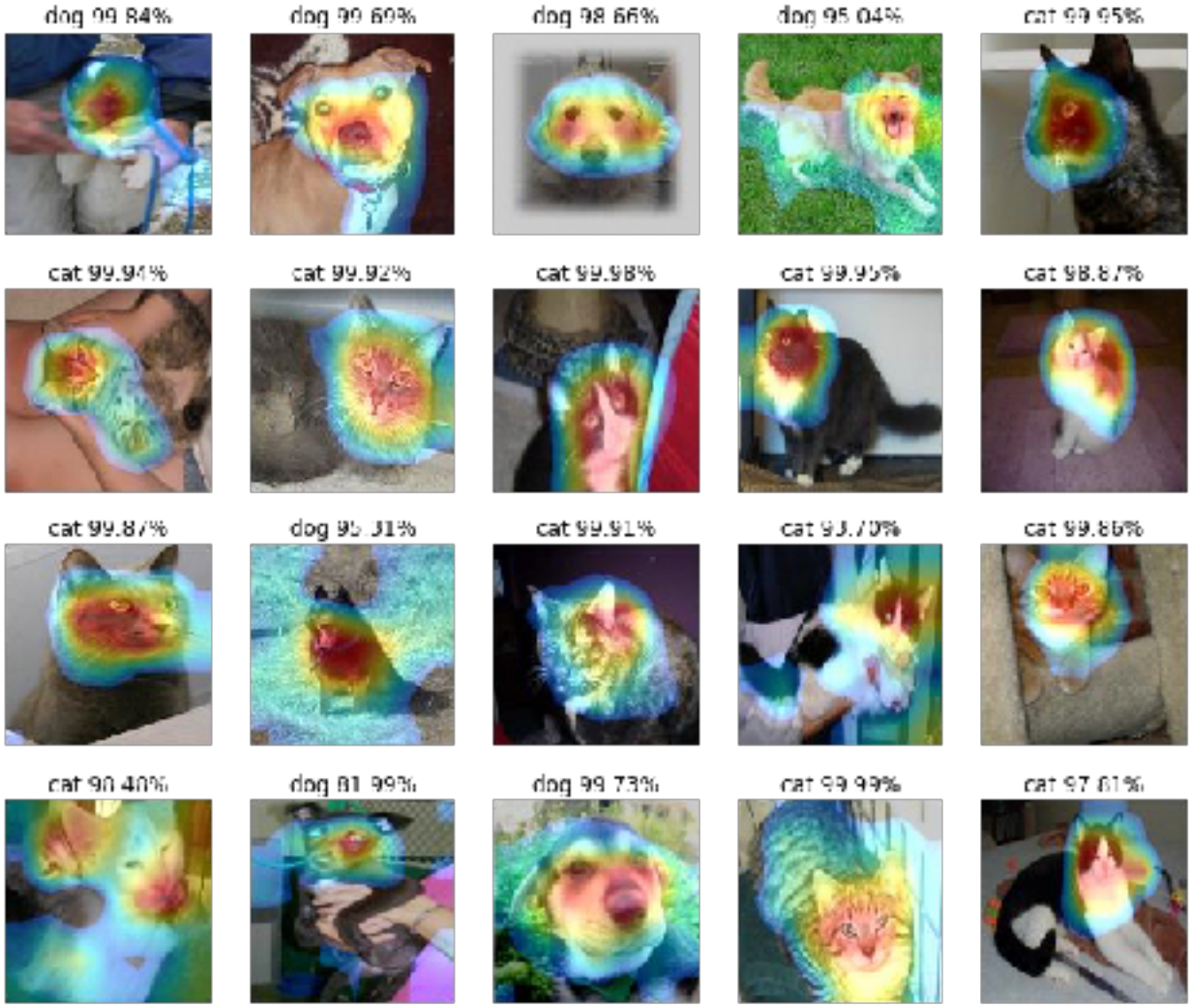
一些错误的预测结果：



模型的总体准确率虽然较高。但对于有些图片，例如分辨率不足、拍照角度、图片亮度、主体不够突出等情况下，依然不能很好地识别。

但是常规的猫和狗的图片都能够正确地识别：

使用feature heatmap显示的分类器捕获的特征：



对项目的思考

项目中所有的关键步骤和其他的CNN项目一致。找到合适的标注数据。设计合适的网络结构。使用合适的迁移学习方法减少训练时间。

首先是合适的标注数据，本项目的标注数据是kaggle预先提供的，这一部分没有花太多功夫，直接下载就好。其他项目在这个步骤需要花费大量的精力和成本。在很多项目中标注数据的获取是最重要的、成本最高的一个环节。

然后是数据的预处理，本项目的数据规格差异不大，仅需要做少量的工作即可：调整图片大小等。

然后因为使用深度学习方法，没有特别的特征工程环节，直接输入图片数据到网络模型训练即可。这一步，深度学习的方法节省了大量的工程开发时间。

网络结构的设计相当于传统的特征工程环节。从0开始设计一个有效的网络是高成本的事情。本项目使用了知名的、经过广泛验证的ResNet-50 CNN模型，并针对Dogs vs. Cats项目做少量的修改即可。

本项目很重要的一个部分是使用ImageNet上的预训练的权重。如果没有这部分，模型的训练时间会大幅增加，模型的精度也会受到影响。

最终模型完成了预测，达到预期效果。

需要作出的改进

为得到更高的准确率，我认为有以下几个事情值得尝试：

1. 使用ResNet-101，ResNet-152等更复杂的网络结构
2. 获取更多猫狗图片做训练数据
3. 进一步微调参数等