沈龙飞

猫狗大战

机器学习工程师纳米学位毕业项目

# 定义

## 1.1项目概述

这是一个Kaggle上的竞赛项目，目标是训练一个深度学习模型，在给定的图像中分辨猫和狗。属于图像分类问题。

本项目中使用的模型是**卷积神经网络**（Convolutional Neural Network, **CNN**）是一种前馈神经网络，它的人工神经元可以响应一部分覆盖范围内的周围单元，对于大型图像处理有出色表现。

卷积神经网络由一个或多个卷积层和顶端的全连通层（对应经典的神经网络）组成，同时也包括关联权重和池化层（pooling layer）。这一结构使得卷积神经网络能够利用输入数据的二维结构。与其他深度学习结构相比，卷积神经网络在图像和语音识别方面能够给出更好的结果。这一模型也可以使用反向传播算法进行训练。相比较其他深度、前馈神经网络，卷积神经网络需要考量的参数更少，使之成为一种颇具吸引力的深度学习结构[。](https://zh.wikipedia.org/wiki/卷积神经网络" \l "cite_note-)

项目选择的数据集是Kaggle上竞赛提供的数据，训练集包括12500张被标记为猫和12500标记为猫的图片。测试机是包含12500未标记的图片。对于每一张测试机中的图像，模型需要预测出是狗图像的概率（1 代表狗，0 代表猫）。

## 问题陈述

Kaggle竞赛数据是从真实数据中采集到的包含猫或狗的图像。图像的分辨率参差不齐，

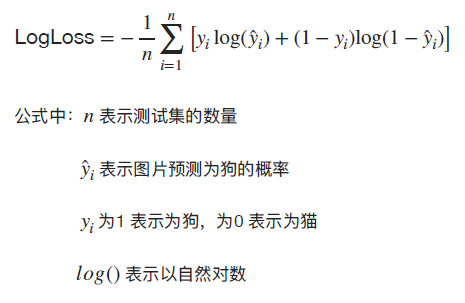
图片中猫和狗的主体也有很大的差异。同时猫和狗的不同品种造成的本身的形体差异。在加上拍摄角度与所处角度环境的关系。照成图片数据中的异常噪音非常大。

在图像分类问题中，需要出现两个过程：体征提取和分类。在特征提取环节，模型需要提取边缘，轮廓，色彩组合等的不同抽象级别特征。在分类部分，需要对提取出的特征进行准确分类。以得出分类结果。

## 评价指标

在kaggle的比赛中提出了标准的评价公式，本项目也采用相同的评价指标。

公式如下：

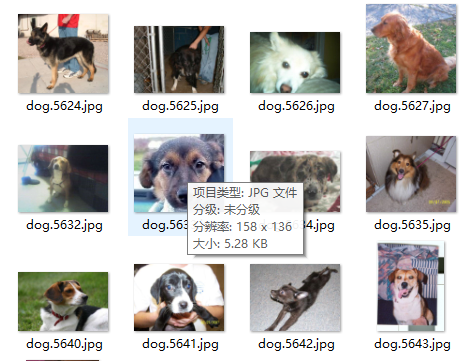


# 分析

## 2.1 数据可视化

竞赛数据集中有两个子集，一个训练数据集，一个是测试数据集，训练数据集中的文件名称即为标签。说明了图片数据猫还是狗。





训练集中共有25000张图片，猫狗各占一半，因此最后训练的时候不需要调整类别权重。

同时猫狗形态各异，要额外注意欠拟合和过拟合现象。

然而在整个图片中同时也存在着少量不靠谱的数据，如下：



## 算法和技术

### 深度学习简介

**深度学习**（deep learning）是机器学习的分支，是一种试图使用包含复杂结构或由多重非线性变换构成的多个处理层对数据进行高层抽象的算法。

这些图片的总体数量大概在100张以内,以两万张的总量来算,比例不及1%.在训练过程中可以有两种方案, 第一种去除异常值进行训练.第二种进行多次随机抽样的训练,取最好的结果.

在这里由于猫和狗的样本数量刚好相等.所以采用第二种方法,进行随机抽样训练,取最好的结果.

深度学习是机器学习领域中对模式（声音、图像等）进行建模的一种方法，它也是一种基于统计概率的模型。在对各种模型进行建模之后，便可以对各种模式进行识别了，例如待建模的模式是声音的话，那么这种识别便可以理解为语音识别。

深度学习是从很早之前的人工神经网络模型发展而来。这种模型一般采用计算机科学中的图模型来直观的表达，而深度学习的“深度”便是指的是图模型的层数以及每一层的节点上核量，相对之前的神经网络而言，有了很大的提升。

为了进行某种模式的识别，通常的做法首先是以某种方法，提取这个模式中的特征，这个特征有时候是人工设计或制定的，有时候是在给定相对较多的数据前提下，有计算机自己在模型中总结得出的。深度学习提出了一种让计算机自动学习出模式特征的方法，并将特征学习融入到了简历模型过程中，从而减少了不少认为设计特征的不完备性。而目前以深度学习为核心的某些机器学习应用，在满足特定条件的应用场景下，已经达到了超越现有是废的识别或分类性能。

由于深度学习能够自动学习，在学习时要额外注意损失函数的结果，是否存在过拟合的想象。再者，在模型训练过程中，需要大量的数据进行总结归纳，会造成训练耗时边长。

我们先了解一下神经元。神经元的结构如下图，也叫做感知机。



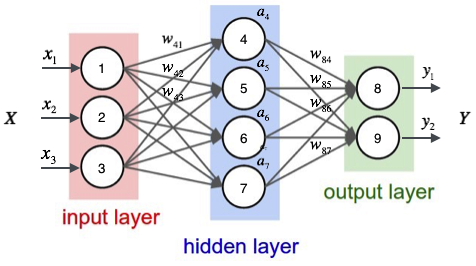
一个感知机包含如下部分：

输入权值（inputs），一个感知机可以接受多个输入（）在每个输入上又一个权值 ，此外还有一个偏置项 ,也就是上图的

激活函数感知器的激活函数可以有很多选择，比如我们可以选择下面的阶跃函数 来作为激活函数：

输出，感知机的输出有如下公式来计算：

深度学习使用的是神经网络模型，神经网络其实是按照一定规则连接起来的多个神经元。



如上图所示：

* 神经元按照层来布局，最左侧叫输入层（input layer），负责接收输入数据；最右边的层叫输出层（output layer），我们可以从这层获取神经网络的输出数据。输入层和输出层之间的层叫做隐藏层（hidden layer），因为他对外部调用者来说是不可见的。
* 同一层之间的不同神经元没有连接。
* 第N层每个神经元和低N-1 层所有的神经元相连（Full Connected），第N-1 层神经元的输出就是低N层神经元的额输入
* 每一个连接都有一个权值

### 卷积神经网络

卷积神经网络（Convolutional Neural Network, CNN）是一种[前馈神经网络](https://zh.wikipedia.org/wiki/前馈神经网络)，它的人工神经元可以响应一部分覆盖范围内的周围单元，对于大型图像处理有出色表现。

### 卷积层的基本原理

从网上的找来一个图片。如下图所示，左侧输入的汽车图片，人可以分辨数来，但是我们怎么让计算机也能像人一样识别出这是一个图片呢。

左边的层次：

一张图片

中间的层次：

CONV ： 卷积计算层

RELU ： 激活函数

POOL : 池化

最右边：

FC： 全连接层



这是一个简单的卷积神经网络的数据运算逻辑。

先从输入图像开始看起，在计算机里面，图像的表示方式可以有很多种，我们这里可以将图片转为RGB三通道的矩阵，图片中的每一个像素点可以认为是一个三个数值的向量。合并到一起之后就变成一个，三维矩阵，包含RGB三色通道的一个三维矩阵。这样我们就可以从纯数学的角度去理解，也有助于计算机能对这些特征进行计算。

中间层次：

卷积层对图片上每个滤波器大小的特征模块进行卷积以增强或者减弱特征部分。

RELU给所有特征进行计算变成下一层的入参，在最后进行梯度下降时候能够快速收敛以迅速算出最优解。

池化：取区域平均或者最大，以避免在卷积之后的特征丢失现象。、

右边层次：

在对图片中所有的像素点进行了卷积和池化之后，可以认为对图片的 特征提取完成，然后进行全连接层进行计算以输出自己想要的分类结果。

### 2.2.4 技术实现

项目实现CNN使用的是Facebook 开源的PyTorch。由于使用CNN处理分辨率全彩色图片所需要的计算量非常大，项目使用了GPU进行计算加速。在安装PyTorch的同时还安装了CUDA和cuDNN。

在图片处理的CNN问题中，主要包含的步骤如下：

1. 读取图片
2. 标准化图片
3. 卷积和池化（可以出现多次卷积和池化）
4. 归一化（隐藏层）
5. 输出结果

整个神经网络中由于面对具体问题的不一样可能会在不同的步骤呈现不同的特点。在猫狗问题中，我们面对的主要挑战来自，图片像素尺寸，以及猫狗动作的不一致性非常高。还包含又猫和狗的品种的不一致产生的数据噪音问题。

用深度学习中的术语来描述，则是说图片中提取的特征需要尽可能多，同时又要避免产生过拟合。

在深度学习的整个流程中，特征提取主要是在图片标准化，卷积和池化环境产生。换言之，这几个环节提取的特征数量，对后续的结果输出影响会很大。

## 基准指标

依据本题的考核标准。需要获得kaggle比赛前10%的成绩，在Kaggle的竞赛页面中，截止目前提交成绩人数为1314名。前10%则意味着分数要超过131名，在页面中可以看到第131名的得分为0.06127

基准模型指标得分需要小于0.06127

# 具体方法

## 3.1 迁移学习

### 3.1.1 迁移学习介绍

迁移学习，简单的说，就是能让现有的模型算法稍加调整即可应用于一个新的领域和功能的一项技术。这个概念目前在机器学习中其实比较少见，但其实它的潜力可以相当巨大。杨强教授在CCF-GAIR上的演讲中曾提到一个愿景——利用迁移学习，即使是自身没有条件获得大量训练数据的小公司也可以按照自己的需要应用大公司训练出来的模型，从而普及AI的应用。

本题中从零开始训练一个卷积神经网络，需要进行细致的网格结果设计，并调整大量的参数进行优化。耗费时间和精力非常惊人。因此我们采用迁移学习，通过微调其中参数以得到较好的结果。

### 3.1.2 模型选择

依据之前得出的论断，我们需要的是对特征尽可能多的提取，以保证在后面的全连接层中可以得到较好的分类结果。

而在PyTorch中提供有多种模型，最符合上面筛选条件的是ResNet模型。

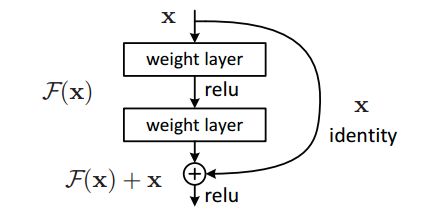
两种主要的迁移学习方法如下：

**微调convnet**：我们使用预训练的网络进行参数初始化而不是参数随机初始化。之后的训练就和平常的训练一样了。

**convnet作为特征提取器**：这里，我们将冻结所有网络层的权重，除了最后一层全连接层。最后一层是一个参数随机初始化的新层，并且只有这一层被训练。

深度残差学习（RESNET）模型，在深度学习中，网络的表达能力随着网络深度的增加而增强，但是同时在RESNET之前，随着网络深度的增加训练的复杂度，网络复杂度和误差都在增大。而RESNET中假设深度网络中某隐含层为 ， 如果可以假设多个非线性层组合可以近似于一个复杂函数，那么同样也可以假设隐含层的残差近似于某个复杂函数。即我们可以将隐含层表示为 .

依据上文假设，我们可以得到全新的残差结构单元。



这个网络具有如下特点：

* + - 1. 网络小，参数数量得到控制
      2. 存在明显的层级
      3. 使用较少的池化层，大量使用下采样，提高传播效率
      4. 没有dropout，使用全局平均池化进行正则化，训练速度有保证

### 3.1.3 数据预处理

由于我们使用的迁移学习，所有数据预处理必须要能符合Resnet101的模型的输入结构。

首先将数据拆分为训练集和验证机.拆分使用的SHELL命令在README2.md文件中.

然后我们在读取数据的时候,对图片进行了均值减去标准差(使用的均值和标准差是来自于Pytorch的 官方教程里面的数值,具体调整还没有仔细研究过),再进行随机切图片,然后resize 成224的大小.同时以0.5的概率翻转图像.

为了方便加载数据,我们将图片分为训练集和验证集,并且,训练集和测试机下都有猫和狗的二级目录.

猫狗训练集分别是11250张,测试机猫和狗分别是1250.

### 3.1.4 模型fine-tune

我们使用预先训练好的模型,这次我们在初期调试的时候,只在最后一层增加一个fc层.让最终的结果输出未两个类别.

## 3.2 实现

### 3.2.1 模型训练

在之前我们对图片进行了预处理中,我们已经做了简单的数据增强行为.主要是随机翻转,同时还有数据标准化.

我所使用的模型是 RESNET101 预训练模型.

载入模型之后, 将预训练模型中的 requires\_grad 设置未False.

因为RESNET101 标准输出的2048个类别,增加一个了一个全连接层.将预训练模型的RESNET101 输出的 2048个类别输出为两个类别.

损失函数使用交叉熵.

优化器使用的Adam 和Momentum,学习率每七个epoch 下降为原始学习率的0.1.

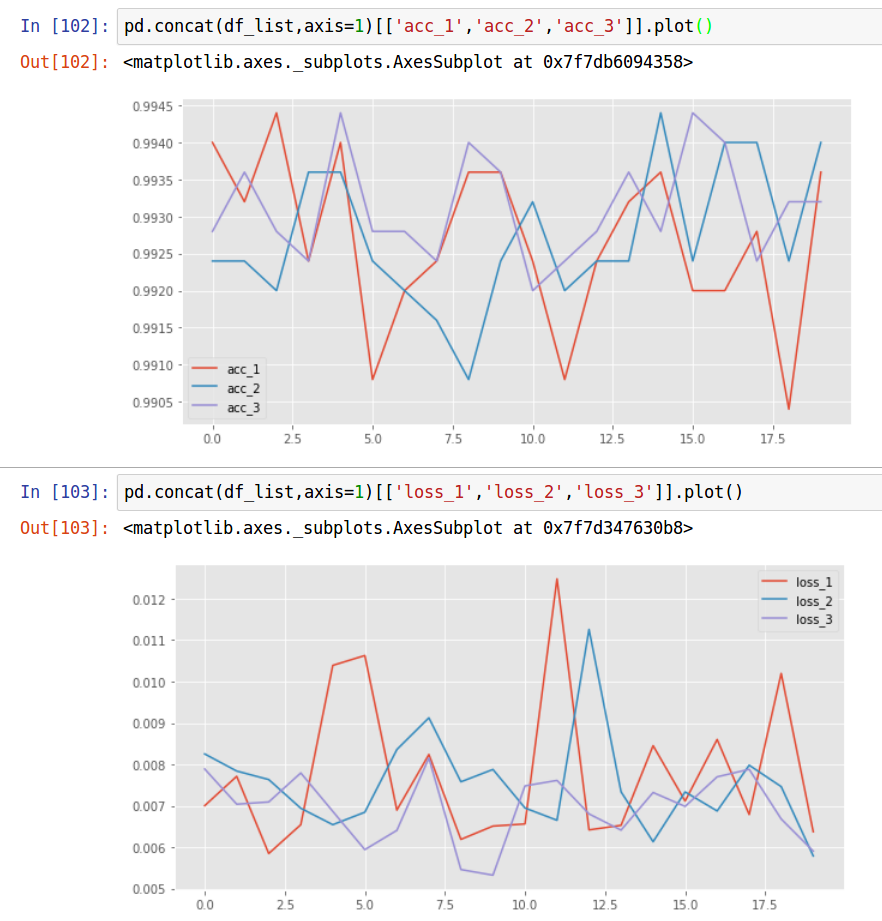
在训练过程中,我们调整了学习率,使用的初始学习率包括,0.1, 0.001, 0.00001, 0.000001 以查看不同学习率下的 得分情况.

结果如下图,图片中的acc\_0 对应0.1,acc\_1 对应0.001 ,acc\_2 对应0.00005

acc\_3 对应0.0000001.loss 也是依次类推.

通过观察图片,在学习率未0.1的时候,图形出现一个较大的波动,在后续的0.001 以及之后的学习率上,结果波动都想类似,依据学习效率上来推算,认为在最少的消耗下可以得到较优的结果.初步确定在使用Adam 优化器时,使用0,001最为学习率.





在图片中由于acc\_0 的波动范围太大,无法准确观察到合适的结果,在去除acc\_0之后,绘制的图形更容易的看出哪个epoch下获得最好的结果.

下图是使用momentum优化器的:



# 4. 结果

## 4.1 模型评估与验证

程序在批量读取测试集中的图片,并根据模型的进行相应的预处理,在初步使用resnet101.

最终几个网站的得分为:

resent101 得分为:0.04898

resnet101的得分,目前处于kaggle 猫狗项目的52 和 53名之间.已经满足目前的得分范围.

# 5.结论

猫狗大战是一个充满挑战的竞赛,使用有限的数据样本,发掘尽可能多的特征.CNN是一个非常灵活的模型,围绕分类以及特征表示,有非常多的方法可以运用,

在整个迁移学习的过程中,尝试过很多中不同的方法,比如结构简单训练速度也很快的VGG模型,但是具体到实际问题的时候,发现存在很多问题.在之前的开发过程中,大致认为是特征提取数量不够,因为原始数据中对图片特征的干扰非常之多.

文献参考:

[1] Deep Residual Learning for Image Recognition

[Kaiming He](https://arxiv.org/find/cs/1/au:+He_K/0/1/0/all/0/1), [Xiangyu Zhang](https://arxiv.org/find/cs/1/au:+Zhang_X/0/1/0/all/0/1), [Shaoqing Ren](https://arxiv.org/find/cs/1/au:+Ren_S/0/1/0/all/0/1), [Jian Sun](https://arxiv.org/find/cs/1/au:+Sun_J/0/1/0/all/0/1)

[2] Very Deep Convolutional Networks for Large-Scale Image Recognition

[Karen Simonyan](https://arxiv.org/find/cs/1/au:+Simonyan_K/0/1/0/all/0/1), [Andrew Zisserman](https://arxiv.org/find/cs/1/au:+Zisserman_A/0/1/0/all/0/1)

[3] THE INTEGRATION OF GEOGRAPHIC DATA WITH REMOTELY SENSED IMAGERY TO IMPROVE CLASSIFICATION IN AN URBAN AREA