项目背景

我选择Kaggle上的Dogs vs. Cats Redux: Kernels Edition项目作为此次毕业项目。这是一个计算机视觉方向的项目。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，并进一步做图像处理，用计算机处理成为更适合人眼观察或传送给仪器检测的图像。 作为一门科学学科，计算机视觉研究相关的理论和技术，试图创建能够从图像或者多维数据中获取“信息”的人工智能系统。这里所指的信息指香农定义的，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。

计算机视觉的一些经典的问题包括：

识别

- 识别（狭义的）：对一个或多个经过预先定义或学习的物体或物类进行辨识，通常在辨识过程中还要提供他们的二维位置或三维姿态。
- 鉴别：识别辨认单一物体本身。例如：某一人脸的识别，某一指纹的识别。
- 监测：从图像中发现特定的情况内容。例如：医学中对细胞或组织不正常技能的发现，交通监视仪器对过往车辆的发现。监测往往是通过简单的图象处理发现图 像中的特殊区域，为后继更复杂的操作提供起点。

识别的几个具体应用方向：

- 基于内容的图像提取：在巨大的图像集合中寻找包含指定内容的所有图片。被指定的内容可以是多种形式，比如一个红色的大致是圆形的图案，或者一辆自行车。在这里对后一种内容的寻找显然要比前一种更复杂，因为前一种描述的是一个低级直观的视觉特征，而后者则涉及一个抽象概念（也可以说是高级的视觉特征），即‘自行车’，显然的一点就是自行车的外观并不是固定的。
- 姿态评估：对某一物体相对于摄像机的位置或者方向的评估。例如：对机器臂姿态和位置的评估。
- 光学字符识别对图像中的印刷或手写文字进行识别鉴别，通常的输出是将之转化成易于编辑的文档形式。

运动
基于序列图像的对物体运动的监测包含多种类型，诸如：

- 自体运动：监测摄像机的三维刚性运动。
- 图像跟踪：跟踪运动的物体。

场景重建
给定一个场景的二或多幅图像或者一段录像，场景重建寻求为该场景创建一个三维模型。最简单的情况便是生成一组三维空间中的点。更复杂的情况下会创建起完整的三维表面模型。

图像恢复
图像恢复的目标在于移除图像中的噪声，例如仪器噪声，模糊等。

关于计算机视觉的相关研究有很多：

- Ask Your Neurons: A Neural-Based Approach to Answering Questions About Images， 研究人员实现了一种基于循环神经网络来看图回答问题的方法，提出了一种结合了卷积神经网络和长短期记忆的方法。
- Learning to See by Moving，研究人员发现使用相同数量的训练图像，使用运动学习的特征作为监督，比使用类标签学习的特征更有利于监督场景识别，物体识别，视觉测距和关键点匹配任务。
- Local Convolutional Features With Unsupervised Training for Image Retrieval， 介绍一种深卷积体系结构，它可以产生补丁级描述符，作为图像检索中流行的SIFT描述符的替代方法。

正因为计算机视觉领域近期有很大发展，许多研究者提出很多新颖和高效的解决方法，使我也希望能在这一领域开拓发展，为此我选择此领域相关问题作为我的毕业项目。

问题描述

猫狗大战（Dogs vs. Cats Redux: Kernels Edition)这个项目属于计算机视觉中的图像识别问题。我的目标是让计算机分辨出图片中是猫还是狗的图像， 并达到尽可能高的准确率。这对于人类来说是相对容易的事，对于计算机确实一个挑战。这类问题我们称之为CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) or HIP (Human Interactive Proof)。我们常用这类问题组成图片识别验证来阻止垃圾邮件和博客以及一些对网站密码的硬攻击。

因为这是一个二元分类问题（猫还是狗），因此我能想到最简单的方法就是Logistic回归，它是处理二元分类最经典的方法，通过输出0/1判断是猫还是狗。

Logistic模型如下：

$$\hat{y} = \frac{1}{1+e^{-(w^T x + b)}}$$

$y = \sigma(w^{*} x + b)$

这里$\sigma(z) = \frac{1}{1+e^{-z}}$。 x是特征向量，我们可以通过梯度下降法得到w和b。

输入数据

这次的数据源来自Asirra (Animal Species Image Recognition for Restricting Access)数据集，它是被用来让用户验证图片显示的是狗还是猫。这一数据集的独特性来自于它是和全球最大的为流浪宠物寻找住处的网站：Petfinder.com合作的。他们向微软研究院提供了超过3百万张猫和狗的图片，并且由来自全美数千个动物庇护机构的工作人员人工标签过。Kaggle提供的是一部分子数据集。

数据集包括如下数据：

- 训练数据目录包含25000张猫和狗的图片。此目录下的每张图片都打上了标记（作为文件名的一部分）。
- 测试数据目录包含12500张猫和狗的图片，以数字id命名。对于测试集中的每一张图片，我需要预测出这张图片显示的是狗的概率（狗=1， 猫=0）。

解决办法

针对这个问题，我准备使用卷积神经网络处理。

卷积神经网路（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。
卷积神经网路由一个或多个卷积层和顶端的全连通层（对应经典的神经网路）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路能够利用输入数据的二维结构。与其他深度学习结构相比，卷积神经网路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、前馈神经网路，卷积神经网路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。

卷积神经网络通常的结构分为：

卷积层
卷积层（Convolutional layer），卷积神经网络中每层卷积层由若干卷积单元组成，每个卷积单元的参数都是通过反向传播算法最佳化得到的。卷积运算的目的是提取输入的不同特征，第一层卷积层可能只能提取一些低级的特征如边缘、线条和角等层级，更多层的网路能从低级特征中迭代提取更复杂的特征。

线性整流层
线性整流层（Rectified Linear Units layer, ReLU layer）使用线性整流（Rectified Linear Units, ReLU） $f(x) = \max(0, x)$ 作为这一层神经的激励函数（Activation function）。它可以增强判定函数和整个神经网路的非线性特性，而本身并不会改变卷积层。 事实上，其他的一些函数也可以用于增强网路的非线性特性，如双曲正切函数 $f(x) = \tanh(x)$, $f(x) = |\tanh(x)|$ ，或者Sigmoid函数 $f(x) = (1 + e^{-x})^{-1}$。相比其它函数来说，ReLU函数更受青睐，这是因为它可以将神经网路的训练速度提升数倍，而并不会对模型的泛化准确度造成显著影响。

池化层
池化（Pooling）是卷积神经网络中另一个重要的概念，它实际上是一种形式的向下采样。有多种不同形式的非线性池化函数，而其中“最大池化（Max pooling）”是最为常见的。它是将输入的图像划分为若干个矩形区域，对每个子区域输出最大值。直觉上，这种机制能够有效地原因在于，在发现一个特征之后，它的精确位置远不及它和其他特征的相对位置的关系重要。池化层会不断地减小数据的空间大小，因此参数的数量和计算量也会下降，这在一定程度上也控制了过拟合。通常来说，CNN的卷积层之间都会周期性地插入池化层。
池化层通常会分别作用于每个输入的特征并减小其大小。目前最常用形式的池化层是每隔2个元素从图像划分出 2×2的区块，然后对每个区块中的4个数取最大值。这将会减少75%的数据量。
除了最大池化之外，池化层也可以使用其他池化函数，例如“平均池化”甚至“L2-范数池化”等。过去，平均池化的使用曾经较为广泛，但是最近由于最大池化在实践中的表现更好，平均池化已经不太常用。
由于池化层过快地减少了数据的大小，目前文献中的趋势是使用较小的池化滤镜，甚至不再使用池化层。

全连接层
全连接层（Fully connected layer） 这一层每个神经元会和前一层的所有神经元相连。理论上它和传统的多层感知器神经网络中的结构一样。

损失函数层
损失函数层（loss layer）用于决定训练过程如何来“惩罚”网络的预测结果和真实结果之间的差异，它通常是网络的最后一层。各种不同的损失函数适用于不同类型的任务。例如，Softmax交叉熵损失函数常常被用于在K个类别中选出一个，而Sigmoid交叉熵损失函数常常用于多个独立的二分类问题。欧几里得损失函数常常用于结果取值范围为任意实数的问题。

基准模型

斯坦福大学的Philippe Golle提出了一种准确率达到82.7%的分类器：http://xenon.stanford.edu/~pgolle/papers/dogcat.pdf 。此分类器是基于支持向量机（Support Vector Machine)对图像色彩和纹理抽取特征训练而得到的。他声称此分类器解决一个12-image Asirra问

题的成功率达到10.3%。

支持向量机的实现：

$$K(v, v') = exp(-\gamma|v - v'|^2)$$

参数γ是通过5折交叉验证趋近达到最好的测试错误表现。作者发现$\gamma = 10^-3$时抽取色彩特征最好，$\gamma = 10 - 1$时抽取纹理特征最好。作者使用了LIBSVM对于SVM的Java实现。

这是作者的测试结果:

Features	# Images		Classifier accuracy	
	Total	Training set	mean	stdev
$(F_1 \cup F_2 \cup F_3) + G_2$	5,000	4,000	**80.3 %**	1.4
$(F_1 \cup F_2 \cup F_3) + G_2$	10,000	8,000	**82.7 %**	0.5

其中F1, F2, F3是色彩特征，G2是纹理特征。

评估指标

虽然我也可以像上述作者一样用准确率（accuracy）评判分类器的性能，但是准确率无法优化，所以我将通过Log Loss判断结果的优劣：

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

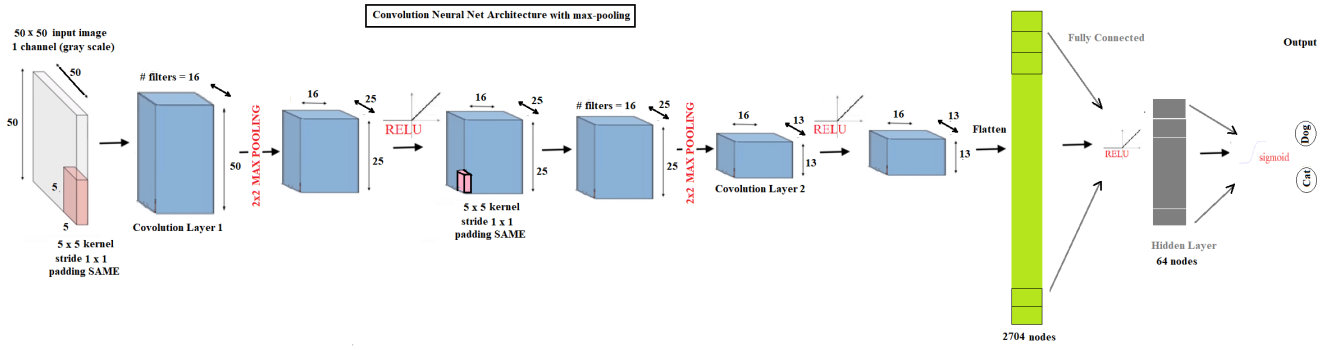
- n是测试集图片的数量
- $\hat{y}_i$是预测图像是狗的概率
- y_i为1如果图像是狗，为0如果是猫
- $log()$是自然对数（以e为底）

Log Loss 越小越好。

设计大纲

我设计的卷积神经网络大致有如下结构：

- 2个卷积层
- 每个卷积层使用5x5的核，1x1的步长，SAME补丁，每层16个卷积滤波器
- 2个最大池化层
- 每个池化层使用2x2的核，2x2的步长（每次过滤前一层75%的神经元），每层16个池化滤波器
- 64个隐层节点



从上面的结构图可以看到，输入图片经过卷积层（通过ReLu非线性层），最大池化层，全连接层直到输出层输出识别为猫或狗的概率。其中卷积+池化可以有多层，更多的卷积+池化能抽取更多特征。同样全连接层也可以有多层，更多的全连接层可以处理更复杂的模型，提高分类的准确率。

引用

1. https://zh.wikipedia.org/wiki/%E8%AE%A1%E7%AE%97%E6%9C%BA%E8%A7%86%E8%A7%89
2. https://www.d2.mpi-inf.mpg.de/sites/default/files/iccv15-neural_qa.pdf
3. http://arxiv.org/pdf/1505.01596.pdf
4. https://hal.inria.fr/hal-01207966/document
5. https://sandipanweb.wordpress.com/2017/08/13/dogs-vs-cats-image-classification-with-deep-learning-using-tensorflow-in-python/
6. http://xenon.stanford.edu/~pgolle/papers/dogcat.pdf