# 机器学习基石开题报告

## 一，项目背景

随着科技的发展，人工智能技术在我们生活中的应用越来越多，例如无人驾驶汽车，AlphaGo，智能音箱，手机拍照软件等等，而这些产品中，大部分都要使用计算机视觉技术，例如图像分类就是无人驾驶汽车中重要的一环，因为无人驾驶所用到的图像识别技术就是以图像分类作为基础的。再比如用于识别宠物狗种类的相机就是利用图像分类技术将拍摄的狗的种类快速识别出来。

既然计算机视觉技术在人工智能领域显得如此重要，那么我们现在就来通过一个项目实例来加强对计算机视觉的理解。由于计算机视觉的研究方向众多，比如图像分类，图像定位以及图像识别，考虑到数据集的易得性，决定采用图像分类来作为本次项目的主题。

那么现在就来简要地说说计算机视觉领域的这些具有代表性的模型的发展历程。那首先不得不说的就是1998年LeCun等众神提出的LeNet5[1]了,改论文的发表代表着CNN的正式提出，具有开创性的意义。LeNet5当时主要是用于解决手写数字识别，手写的数字0-9。但是苦于当时的硬件条件，使得该网络难以训练。时间转到2007年，一个对计算机视觉有着深远影响的项目展开，那就是由李飞飞团队发起的标注数据集合ImageNet。截止至目前，该数据集已经超过了1200万张标注图片，为CNN的发展提供了极大的数据支持。并且从2010年开始，ImageNet每年都会举办ILSVRC比赛，来自全世界的开发人员都可以参加，而且获得比赛第一名的人员不是进入了知名企业就是获得了风险投资自己创业。接下来，我们就以此为索引来介绍比较具有深远意义的CNN。如果非要定义一个人工智能热潮的起点的话，那就是2012年吧，在2012年的ImageNet的ILSVRC上，由Geoffrey Hinton、Ilya Sutskever和Alex Krizhevsky提交了AlexNet一举夺得了第一名的成绩。其中AlexNet的几个创新点不得不提：

1，使用Relu激活单元代替了sigmoid单元，加速了训练过程，减少了计算量

2，采用dropout随机地丢弃某些神经元，使得模型不完全依赖于某一个神经元，达到

了正则化的效果。

3，采用了max-pooling的技术。使得模型在处理图像的时候具有平移不变特性。

3，采用GPU训练网络，加快了训练速度。

现在时间转到2014年，就在这一年，产生了两个影响重大的CNN模型。首先来介绍VGG[3]。VGG的比较突出的地方就是率先使用多个比较小的3\*3的卷积核来代替以往的7\*7或者9\*9,卷积核，这样能够显著减少参数的使用。之后我们将要介绍的 GoogleNet 和 ResNet均采用此种方式。其实VGG和AlexNet相比主要的改变还是更深，因为VGG相继推出了16和19层版本，可见从很早开始，人们就意识到越深越强大。我们的VGG在2014年的ImageNet举办的ILSVRC中取得了第二名的成绩。那谁取得了第一名呢？当然是接下来我们要介绍的GoogleNet[4]了。该网络在2014年的ILSVRC中击败了VGG，夺得第一名，我们就来看一看它的神奇之处吧。GoogleNet除了延续了Alexnet的基本单元外，还引入了一种叫做“inception”的block，它的原理其实就是利用不同尺寸的卷积来同时提取不同的特征，最后再将卷积的结果连接起来，使得我们对卷积大小的选择不再那么敏感，既然不知道选多大的合适那就都选吧！其次，为了避免由此造成的计算量增加的问题，GoogleNet引入了1\*1卷积，在每个3\*3或者5\*5卷积的前面加入了一个1\*1的卷积，可以有效降低输入的channel，减小计算量。同时为了避免池化之后的输出结果channel过大，GoogleNet也在池化之后加入了1\*1的卷积来减小输出channel。具体如下图所示。其实inception经过了几代的改进，比如v2加入了BN，减少了隐藏层数据的偏移程度。利用两个3\*3卷积代替1个5\*5卷积减少了参数数量和计算量。同时，其输出层采用池化+softmax，而不是过多地使用全连接层，大大减少了参数数量。

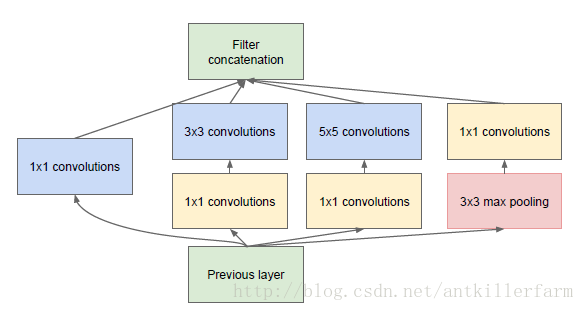
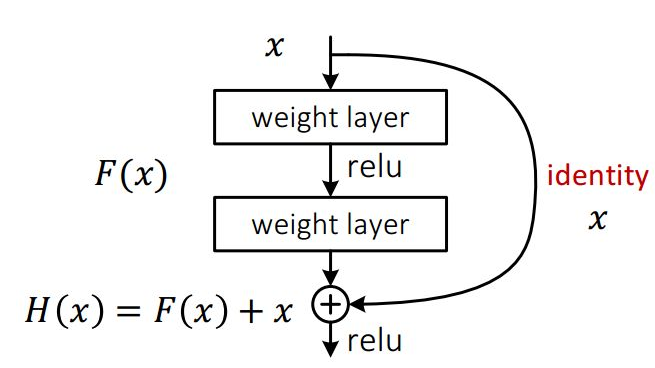


图1. Inception V1 block

现在我们来描述另外一种网络，它由一种兼做残差块儿的结构组成，名叫ResNet[5]，其特点就是深！传说其层数已经突破了一千层！其为什么能够做到如此深层呢？我们都知道，当网络层数变深之后，网络总是能够在特定数据集上表现越来越好。但是随着网络层数过度加深，由于梯度消失等原因，网络会变得难以训练。所以会出现网络越深，表现反而越差的现象。但是ResNet的出现极大地缓解了这一问题。如下图：



上图就是我们所说的构成ResNet的残差块，这种网络结构可以很好地解决由网络深度造成的梯度消失问题。我们知道梯度消失的根源是梯度连乘造成的，在上述结构中，如果我们对x求梯度的话，总会出现一个1+F’(x)，那么该梯度比1大一点儿，所以在连乘的时候不必担心会出现梯度消失。对于为什么这种网络能够如此有效还有另一种解释，我们可以把中间的两层weight layer看成是为了加深网络深度而后加入的，所以如果中间的两层网络如果不能够学习到有用的东西的话，那么中间的权重直接学习成0就可以了，因为网络想要学习这个恒等式还是比较容易的，这样新加入的层不会对原有的层有任何损伤，并且如果中间层如果学习到有用的信息还会对网络整体具有积极作用。也就是说残差块儿能够在不影响网络原有性能的情况下提升性能。对于残差块儿中的卷积的使用，采取了GoogleNet的1\*1卷积来降低计算量，而不是使用连续的3\*3卷积。resNet在2015年的ILSVRC中取得了第一名的成绩。

## 二，问题描述

本次项目的名称是“dog vs cats”，是来自于kaggle(www.kaggle.com)竞赛的一个项目，该比赛的目标（要解决的问题）是：实现一个算法，能够分辨输入的图片内容是猫还是狗。虽然这对人眼来说是很容易的事情，但是对于机器来说，肯定是有一定挑战的。通常来说，我们的算法需要输出该图片分别属于各个分类的概率，并且，通常我们选择概率最高的作为算法最终的分类结果。问题描述就这样简单，那么由于我们并没有开始做项目，所以对项目的很多细节以及会出现的问题都不可能了解得很全面，所以我感觉我们应该首先快速地搭建一个系统，并进行迭代。而不是想出一个完美的模型或者完美的策略之后再进行。

我们需要一个基准的算法或者模型来进行系统的快速搭建。我们的问题是一个分辨猫狗的问题，也就是数据集中不是猫就是狗，那这是一个典型的二分类问题。再者，我们的输入数据是图像数据，而不是结构化数据，考虑到输入数据的特殊性，我们应该更多地考虑使用CNN而不是类似SVM等模型来解决，因为我们可以利用CNN的参数共享以及稀疏链接的特征来很好地训练图像，但是对于SVM等分类器来说，因为其算法没有针对图像来做特定的优化，并且图像数据对于传统的机器学习算法来说特征维度过高，无法取得和CNN一样的好的结果。所以我们选择采用了CNN的算法。预期的方案为使用单独的ResNet-50模型来进行训练，然后单独用于猫狗的预测。我会在项目刚开始的时候使用这个初步的方案，然后利用偏差方差分析或者误差分析来对进行超参数的调节或者更换方案。

## 三，数据集

数据集获得途径：本项目的数据集由kaggle官方提供下载。“dog vs cats”数据集以打包的形式提供，包括了两个打包文件，train.zip和test1.zip，所以kaggle已经帮助我把训练集和测试集分好了。其中train.zip里面包含了25000张jpg格式的图片，猫和狗的图片各有12500张，用于算法的训练。而test1.zip里面包含了12500张jpg格式的图片，用于测试。并且我们需要将测试结果（每张测试图片属于狗的概率）按照CSV文档的格式提交以获得最终的竞赛评分。

数据集：我随机地从训练集中抽取了几张图片，发现这些图片均是把猫/狗作为主体（不清楚其他为抽取的图片是否含有异常值）的图片，而且图片比较清楚，主次分明，非常有利于我们的学习。这些图片均为彩色图片（三通道）。并且我们发现，并没有单独的文件来说明每个样本的label，其实这些信息都在图片的名称里面，比如cat.1.jpg，cat就描述了这张图片属于猫。我们可以通过这个信息来提取所有样本的label。

输入数据：由于我们的样本是图片，所以我们需要用专门的工具（比如OPENCV）来将图片转换为实际的数据，其中数据是一个numpy数组，每一张图片由三维数组表示，我们习惯的数据的每一维的表达方式是Hight,Width,Channel。如果要表示整个数据集，应该再加上一维Batch,Hight,Width,Channel。其中batch表示具体哪一个样本。但是当经过了对数据集的仔细分析后发现，图片的尺寸并不是统一的，而且不同的模型对图片的尺寸要求可能是不同的，比如resnet50要求224\*224\*3，而inceptionv3要求299\*299\*3，所以我们需要一些措施能够将图像输入大小任意改变。在Keras环境下，如果采用了Keras支持的预训练模型，如resnet50和inceptionv3，那么Keras也一定会提供相应模型的预处理函数（也可以自己写），该预处理函数就可以将对输入的图片进行预处理，那么这个预处理函数要放在哪儿呢？首先，可以放入ImageNetgenerator中，这样的话在每次生成一个batch的时候均会对该batch的数据做预处理，然后输入给模型。当然，预处理函数还可以放入另一个地方，就将其作为我们模型的输入层，这样无论多大的图片在输入模型后都会自动得到处理了。

## 四，解决方案描述

对于该项目，我们不会从头开始搭建自己的神经网络，而是采用已经得到认可的模型，并且这些模型在各个大数据集上都有着非常不错的效果。比如ResNet-50或者inceptionV3等网络在ImageNet数据集上有着相当出色的表现，而且ImageNet和我们的项目有着相似之处，比如我们训练的对象都是图片，而且我们做的任务都是分类。只不过ImageNet的分类数目更加庞大，其拥有1000个不同的分类，其中就包括猫狗。所以既然上述算法在ImageNet上能够表现好，那就一定能够在我们的猫狗数据集上表现好。

另一方面，我们也不打算从头开始训练我们选择的网络，因为这些网络已经在比如ImageNet等数据集上使用多GPU训练了数周甚至数月，我们可以借鉴这些已经训练过的权重进行局部微调甚至整体微调。因为对于计算机视觉的任务来说，神经网络学习到的东西很多都是可以通用的。比如在我们的卷积神经网络的前几层，网络学习到的知识总是一些直线，边缘，简单的形状等，然而在网络的顶层，网络将学会更多抽象的东西，比如将器官组成人脸，或者组成整个人[6]。所以我们可以分享模型或者网络从其他数据集学习到的知识,而不是自己从头训练。GitHub里已经为我们提供了权重文件的下载。

进一步地，对于模型的选择，我有三套方案。

1，我会独立训练多个不同的模型，利用验证集来选择表现最好的一个模型。

2，同样独立地训练不同的模型，但是现在采用集成学习的思维，把多个模型的结果（概率）综合（求平均）起来，得到一个结果，看是否能利用模型本身的diversity来获得比较好的集成效果。

3，将多个模型的top层去掉（一般是卷积之后的全连接层），然后将多个模型的卷积输出结果综合起来，输入一个sigmoid函数中，得出分类的结果。这也是一种集成。我们可以仅仅微调sigmoid（输出层）的参数，也可以尝试少量开放每个模型的最后几层来进行微调（我们不会开放太多的层，因为毕竟训练数据集只含有25000个样本，并不是特别多，开放太多可能还达不到之前预训练的效果）

## 五，基准

现在我们现在需要定义一个基准，这个基准用于当前该问题的平均解决水平，同时也作为本项目的最低标准。该项目的最低标准为kaggle中“dog vs cat”的kaggle leaderboard 前 10%，我查看了一下当前的情况，当前提交结果的人数是1314，而前10%大概就是前130名以内。可以看到第130名的score是0.06114，所以我们之后搭建的模型的score应该低于这个score。

## 六，衡量标准

我们已经知道了至少要做到多少的score，现在，考虑如何来衡量算法的优劣（衡量指标）呢？对于本项目而言，kaggle官方规定了本次项目的score的衡量的方法是log loss，公式如下，其中y hat代表预测该样本是狗的概率，y为该样本的标签。该项目没有对算法的预测时间（满足指标）做出要求，所以我们也就不额外添加其他满足指标了。



## 七，项目设计

前面说过，我们的当务之急是快速搭建模型并进行迭代。大概分为下面几个步骤：

1，指定评价标准

我们前面已经将logloss设定为评价标准，有了这个评价标准，我们可以轻松判断模型的性能，以及模型当前的方差偏差问题等。

2，设立验证集，训练集，测试集

我们已经包含了训练集和测试集，是直接从kaggle下载得来。但是模型的训练过程中需要验证集来评价当前模型的训练情况。由于现在训练集有25000个样本，其中猫狗各占一半，我们可以从训练集中抽取5000个样本作为验证集，对于衡量算法的性能来说，足够了，并且这5000个应该是猫狗各占2500个，为了和训练集的分布相同。

3，快速搭建一个初始系统并迭代

我们首先采取的是一个单独的并且经过ImageNet数据集预训练的ResNet-50网络。由于是采用的开源的网络架构，所需要选择的超参数比较少。其次，我们将resnet-50的输出层改为sigmoid以适应我们的项目，然后锁定全连接层以外的所有层，使其不进行梯度更新，仅仅只微调全连接层。

4，进行偏差方差分析

当初始训练完成以后，我们需要对模型进行偏差方差分析。如果当前的模型的偏差比较大（模型训练集logloss比较小），应该首先解决偏差问题，比如尝试训练更长时间，采用更多的优化的optimizer（如momentum，RMSprop，Adam等）。如果当前模型的方差比较大（验证集logloss和训练集logloss相差比较大），应该加入dropout，或者正则化或者data augmentation等操作来减小方差。但是，应该优先解决偏差，然后再考虑方差问题。

5，采用不同的模型和方法

我们可以采用不同的模型和方法（第四节描述的方法），然后重复本节第三步和第四步，最终得出最好的方法。

参考文献：

[1]:Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[2]:A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25, pages 1106–1114, 2012.

[3]:Karen Simonyan, Andrew Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR abs/1409.1556 (2014)

[4]:Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich: Going deeper with convolutions. CVPR 2015: 1-9

[5]:Kaiming, He, Xiangyu, Zhang, Shaoqing, Ren, Jian, Sun. Deep Residual Learning for Image Recognition[Z]. 2015

[6]:Matthew, D, Zeiler, Rob, Fergus. Visualizing and Understanding Convolutional Networks[Z]. 222: 333,2013.