**机器学习纳米学位毕业论文**

**猫狗大战**

**刘珅珅**

**2019年3月29号**

1 问题的定义

* 1. 项目概述

本项目来源于kaggle竞赛dogs-vs-cats比赛[1]，是通过机器学习的方式来识别出图片是猫还是狗，这是一个二元分类问题。选择这个项目作为毕业项目的原因是本人对计算机图像处理感兴趣，并且项目本身会用到深度学习和卷积神经网络的知识，这也是目前人工智能的热点。

本项目的数据集为kaggle提供的数据集，训练集为25000张图片，其中12500张为猫类图片，12500张为狗类图片。测试集为12500张猫狗图片。

* 1. 问题陈述

该项目是一个监督学习的二分类问题，训练集通过图片名称已经打好了标签。考虑到数据集较大以及Udacity课程中深度神经网络有关迁移学习的介绍，可以将问题分为两部分：特征提取和模型分类。特征提取是提取实际图片的边缘特征、轮廓特征以及更高级的能区分猫狗的特征等。模型分类是利用之前提取的特征，输入到一个模型中进行分类训练和预测，并根据训练过程中的参数如准确率、损失大小等来判断是否满足要求来调整模型参数，最终得到符合要求的分类结果。

* 1. 评价指标

在kaggle竞赛中，该项目采用的评价指标为LogLoss，即对数损失。本项目采用与之相同的评价指标。

对于一个二分类问题，LogLoss的计算公式如下所示：



其中为测试集样本数，代表图片的真实标签，如果是狗，否则。表示模型预测图片为狗的概率。

从LogLoss的计算公式可以看出，模型不仅需要预测正确，而且预测的概率要尽可能的大，即对预测结果非常肯定，否则LogLoss的值就会较大。毕业项目要求模型的LogLoss必须在kaggle pubic learnboard中排入前10%，即LogLoss<0.06127.

2 分析

2.1 数据可视化

训练集中有25000张图片，猫和狗各占一半为12500张，通过名称来区分类别。

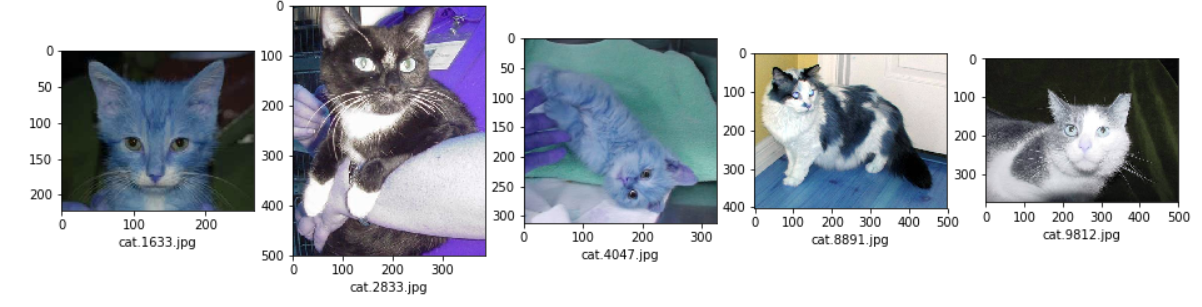


图2-1 训练集中猫类示例图

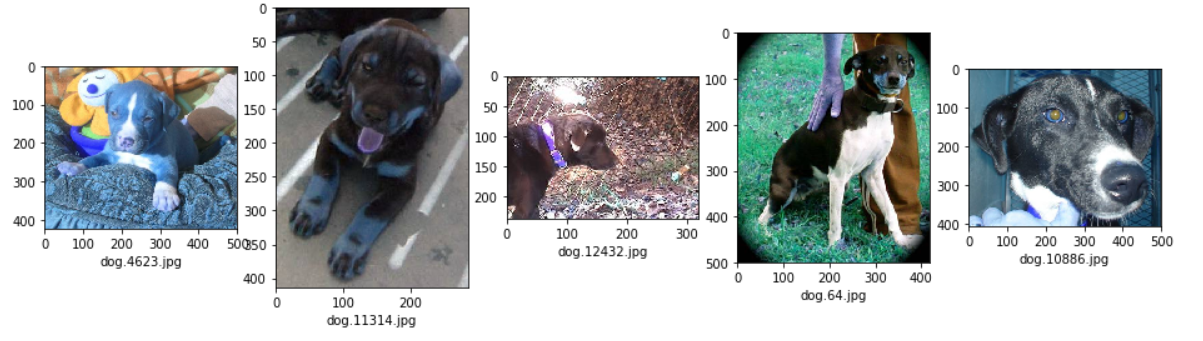


图2-2 训练集中狗类示例图

测试集总共有12500张图片，没有分类。

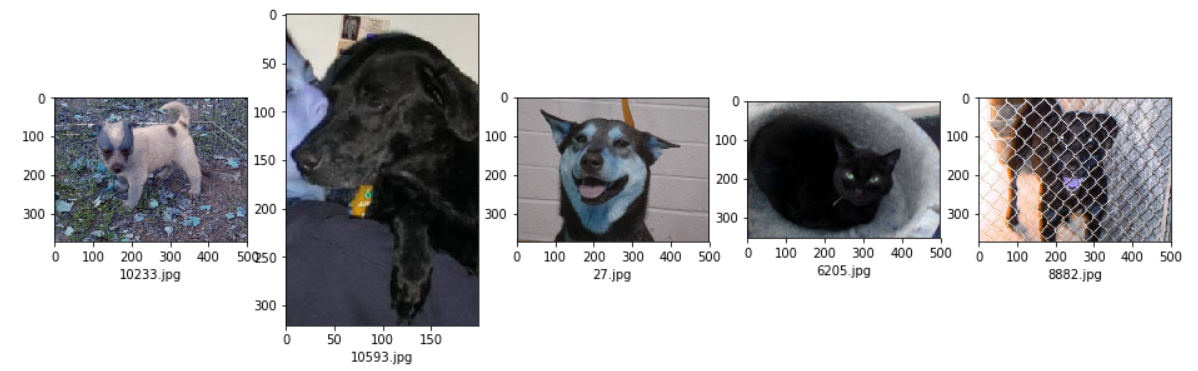


图2-3 测试集示例图

从上图可以看出，大部分图片中猫和狗的特征都非常明显，质量比较高，也有图片如dog.12432.jpg，其中狗所在图片区域较小，而且为侧视图，可能较难分辨。

2.2 算法和技术

在本项目中，考虑到主要是对图片进行建模处理，决定采用卷积神经网络(Convolutional Neural Network, CNN)来进行算法建模。

CNN的研究开始于20世纪90年代。1998年，Lecun等人提出了LeNet-5的卷积网络结构，在字符识别应用中取得良好的效果。随着硬件技术尤其是GPU的快速发展，一些深层的CNN网络相继被提出，2012年由Krizhevsky等人提出的AlexNet模型在ImageNet的图像分类竞赛中获得了冠军，之后牛津大学提出了VGG结构，谷歌提出了GooLeNet结构，微软提出了ResNet结构，[Christian Szegedy](https://arxiv.org/search/cs?searchtype=author&query=Szegedy%2C+C)等人提出了Inception V3结构，Francois Chollet提出了Xception结构。

CNN通常包含以下3层：

1. 卷积层（Convolutional Layer）
2. 池化层（Pooling Layer）

2.2.1 卷积层(特征提取层)

传统神经网络如果输入图片是1000x1000，则需要10^6个输入单元，如果隐藏层也有10^6个结点，则权重系数就会有10^12个，根本无法计算。

1. 局部感知

卷积层采用局部感知的方法来解决这一问题，每个隐藏层的结点都只与部分输入结点相连。

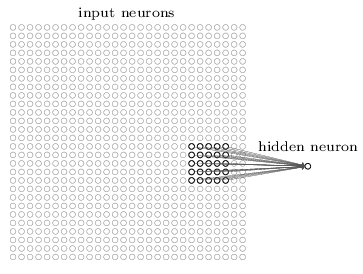


图2-4 卷积层示意图

具体的方式是采用一个filter或kernel对像素矩阵进行卷积来实现。我们移动这样一个filter使其扫描输入矩阵，每次移动1步(stride)都有一个不同的隐藏层结点与之对应。

第1个：

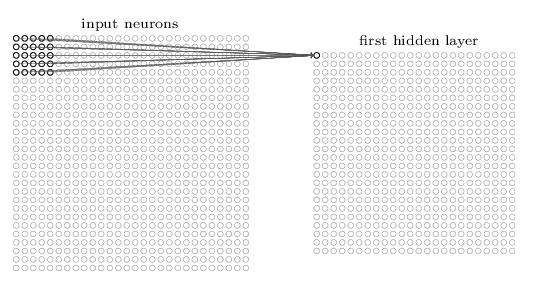


图2-5 卷积窗口(filter)移动示意图

第2个：

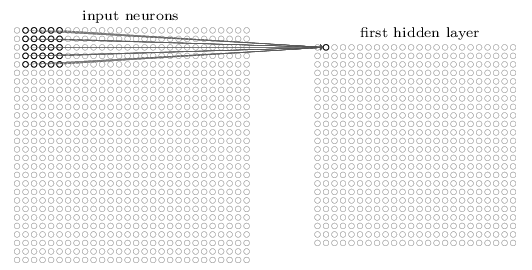


图2-6 卷积窗口(filter)移动示意图

以此类推，可以形成第1个隐藏层，如上图所示，输入为28x28，filter为5x5，得到的隐藏层为24x24(24=28-5+1)。也可以设置每次移动多步。

1. 权值共享

经过局部感知后，隐藏层有24\*24个结点，每个结点对应5\*5个权重系数。



其中代表输入层的矩阵，代表filter中的参数，表示激活函数，一般是Sigmoid或ReLU等，公式表示隐藏层第j行第k列的结点的输入。

虽然比全连接有所降低，但仍然很多，所谓权值共享就是认为隐藏层24\*24个对应的5\*5的权重系数和偏移值都是相同的。这样隐藏层的参数就变成了5\*5.这样认为的原因是：我们认为隐藏层从图像某一部分提取出来的特征也适用于另一部分，这也意味着一个隐藏层只能提取出一种特征。

一般CNN都会有多个卷积核(filter)。例，如果使用3个filter:

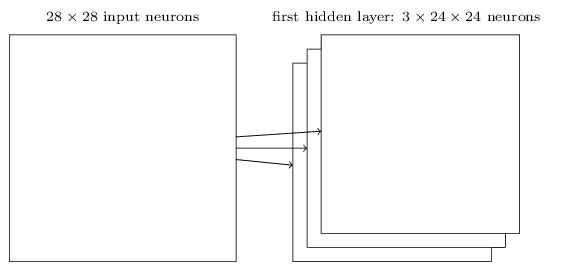


图2-7 多个卷积核

2.2.2 池化层

在卷积层后添加一个池化层，用来简化卷积层的输出，也使用一个filter，但不进行卷积操作，而是通过最大值，最小值、平均值等来进行。

例，2\*2的max filter，选4个参数中最大的一个，从而压缩了卷积层的输出。

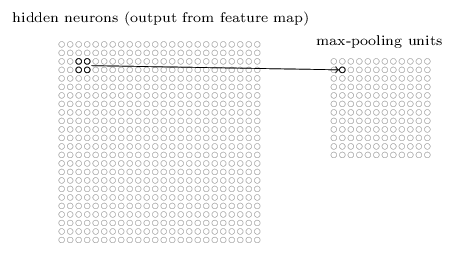


图2-8 最大池化层示意图

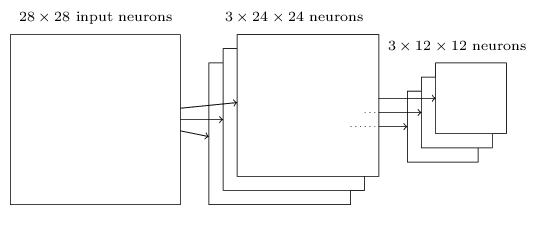


图2-9 卷积层+最大池化层示意图

经过池化层后，会大大减少我们学到的特征。

2.2.3 技术实现

本项目采用基于Tensorflow的开源机器学习库Keras来实现模型的构建、训练和预测。数据集本身包含较多的图片，从头开始构建一个卷积神经网络并进行参数优化非常复杂耗时，在ImageNet中，包含通过猫狗图片进行训练并将参数优化的预训练CNN模型，所以可以采用迁移学习(Tranfer Learning)的方式来进行模型构建。

迁移学习就是将已经预训练CNN模型的参数迁移到新的模型上已提高模型的训练速度和效果。在本项目中使用的训练集与之前ImageNet训练模型的数据集有很高的相似度，所以可以用预训练模型的卷积层参数来进行特征提取，才添加自己的全连接输出层进行分类识别。

在Keras中有5种著名的CNN模型，分别为VGG16，VGG19，ResNet50，InceptionV3，Xception。本项目将分别使用这5种模型进行特征提取，然后对比它们的训练效果，选取最优的模型。

2.3 基准指标

本项目是二分类问题，随机情况下图片是猫还是狗的概率为0.5，这种情况下LogLoss=0.6931，因此我们构建的模型的LogLoss必须小于0.6913.

3 方法

3.1 数据预处理

1) kaggle中提供的训练集图片名称以cat和dog开头，统一放在train文件夹下，所以先将其分类放在cat和dog文件夹。另外需要分出一部分数据作为验证集用于交叉验证，本项目将训练集数据的20%作为验证集，存放在valid文件夹的cat和dog文件夹下，这样训练集图片为20000张，猫狗图片各1000张，验证集图片为5000张，猫狗个2500张。

2) 本项目采用预训练模型VGG16，VGG19，ResNet50，InceptionV3，Xception进行特征提取，这几种模型对图片的预处理要求有所不同。根据keras.applications的源代码可以看出，VGG16，VGG19，ResNet50的预处理方式是相同的，需要图片每个通道都减去ImageNet的mean值[103.939, 116.779, 123.68]，并将图片从RGB转换为BGR；InceptionV3，Xception的预处理方式是相同的，将图片数据缩放到[-1, 1]之间。

3) 本项目使用keras中的ImageDataGenerator类通过实时数据增强批量将图片转换为张量图像数据，其中VGG16，VGG19，ResNet50默认的图像尺寸为(224, 224)，InceptionV3，Xception默认的图像尺寸为(299, 299)。

3.2 特征提取

分别创建上述5种预训练模型用于特征提取，创建时需要将模型的include\_top=False，这样就不包含模型最后的全连接层，只通过卷积层来提取特征。在使用小批量的数据集（训练集1600张，验证集400张）进行测试时，发现输出的特征数据文件非常大，超过了1G，因此统一将模型的pooling=’avg’，即在卷积层的最后添加一个GlobalAveragePooling2D层，用于减少特征量，防止过拟合。

3.3 模型分类构建与训练

特征提取完成后，使用特征数据作为输入数据，仿照预训练模型构建3个全连接层的模型进行训练，模型信息如下图所示。

model = Sequential()

model.add(Dense(2048, input\_shape=(2048,), activation='relu'))

model.add(Dropout(.5))

model.add(Dense(2048, activation='relu'))

model.add(Dropout(.5))

model.add(Dense(1, activation='sigmoid'))

model.compile(Adam(lr=0.001), loss='binary\_crossentropy', metrics=['accuracy'])

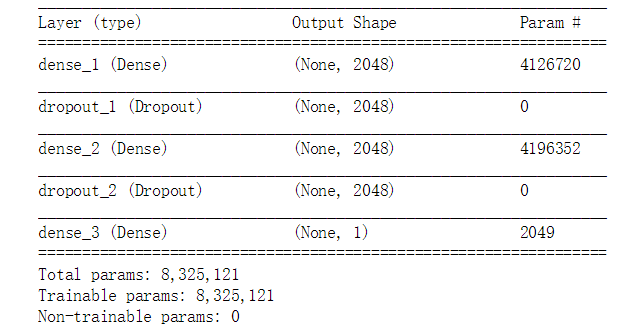


图3-1 分类模型信息

针对5种特征提取模型，分类模型架构几乎完全一致，只有输入参数input\_shape不同，VGG16和VGG19的input\_shape为512，ResNet50，InceptionV3和Xception的input\_shape为2048.

卷积层输出的特征向量维度为2048，分类模型添加两层全连接层结点数也设置为2048，激活函数使用ReLU函数，并添加Dropout层用于防止过拟合。由于是二分类问题，输出层结点为1，激活函数使用Sigmoid函数，优化器使用Adam函数，学习率设置为0.001，损失函数为二分类的交叉验证函数，训练期间评估指标设置为准确率。

3.4 遇到的问题及模型完善

1) 预处理函数的修改。之前根据Udacity课程讲解及参考网上资料，自己写预处理函数，在ImageDataGenerator函数中添加了图片的平移、旋转和随机缩放等步骤，在小批量的数据集（训练集1600张，测试集400张）发现效果并不理想，模型的训练损失和验证损失都比较高，模型不收敛，后来之间使用VGG16等5种模型自带的预处理函数，发现效果比较理想，推测是由于数据集与ImageNet有较大的相似性，使用自带的预处理函数，效果反而更好。

2) 优化器的选择。最初优化器选择的是RMSprop，在小批量数据集训练时，模型很难收敛，增加训练次数训练损失和验证损失几乎没有改进，后尝试了SGD和Adam，最终发现Adam并将学习率设置为0.001的情况下，效果比较理想。

4 结果

4.1 模型评价与验证

使用VGG16，VGG19，ResNet50，InceptionV3和Xception5种模型进行特征提取，然后在自建分类模型上训练得到的结果如下：

1. VGG16的训练准确率超过0.95，验证准确率超过0.92。
2. VGG19的训练准确率超过0.98，验证准确率超过0.93.
3. ResNet50的训练准确率超过0.98，验证准确率超过0.96.
4. InceptionV3的训练准确率和验证准确率均超过0.99.
5. Xception的训练准确率和验证准确率均超过0.99.

4.2 合理性分析

将5种模型训练的最优参数保存下来然后载入到分类模型中，对测试数据集进行预测并将结果上传到kaggle，得到的结果如下图所示。

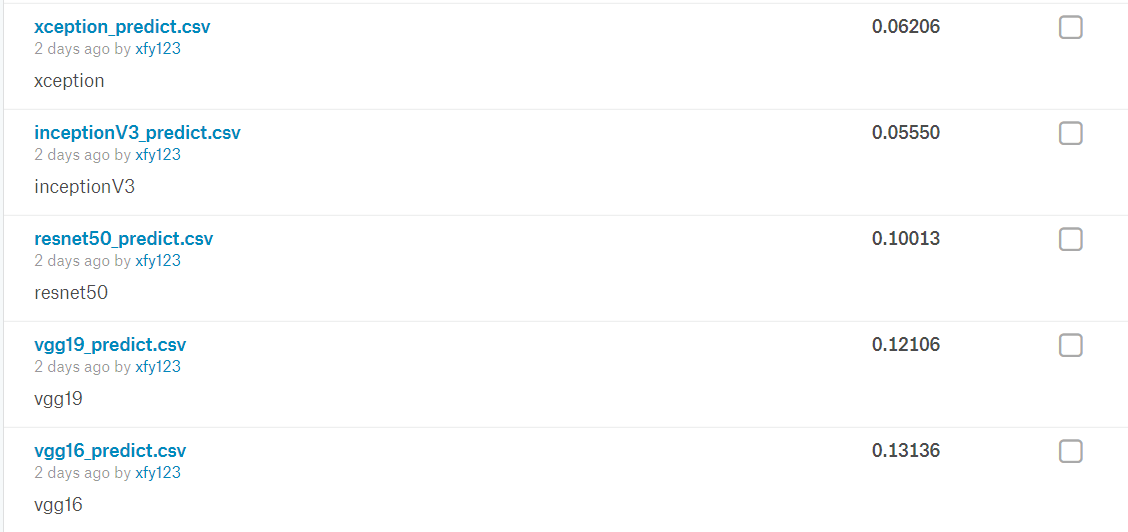


图4-1 kaggle评分结果图

由上图可以看出，InceptionV3模型的LogLoss<0.6127，符合项目设置的要求。

5 项目结论

5.1 结果可视化

InceptionV3模型的准确率

5.2 对项目的思考

本次项目，从数据预处理到模型选择，再到参数的调整都完整的经历了一遍，我有以下收获：

1) 从头在阿里云上搭建了整个训练环境，中间经历了安装Tensorflow失败，Keras版本不匹配的错误，代码也爆出了各种与环境配置相关的错误，最终都成功解决。

2) 在迁移学习中，对于不同的数据集，要多选择几种模型进行比较操作，这样可以选择出最优的模型，没有那个模型可以适合所有类型的数据集。

3) 对优化器的选择一定要多尝试几种不同的情况，之前根据Udacity中项目的经验选择的RMSprop由于结果不收敛，一度让我怀疑迁移学习是否能成功。

参考文献

[1] <https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition>