**机器学习（进阶）纳米学位毕业项目**

**基于深度学习的图像分类**

****

**董仁广**

2018.6.28

**目录**

一、问题定义 3

1.1 项目概览 3

1.2. 问题说明 3

1.3. 评价指标 3

二、项目分析 4

2.1 数据研究和可视化 4

2.2 算法与方法 6

2.3 基准测试 9

三、实现方法 9

3.1 数据预处理 9

3.2 实施 10

3.3 改进 11

四、结果 12

4.1 模型评估与验证 12

4.2 结果分析 13

五、结论 14

5.1 自由形态的可视化 14

5.2 思考 15

5.3 改进 15

六、引用文献 16

# 一、问题定义

### 1.1 项目概览

“猫狗大战”项目来源于一个Kaggle比赛[1]。早在2013年，Kaggle就举办过一次猫与狗识别比赛：识别图片中的动物是猫还是狗。但是在那之后，机器学习领域——特别是深度学习与图像识别领域发生了一系列的突破性进展：理论上，诸多更加成熟高效的深度神经网络模型被提出，颠覆了传统的机器学习领域，并被大量应用于诸如：计算机视觉、自然语言处理、搜索引擎、医疗诊断等领域了；实践上，诸如：Tensor Flow、Keras、Caffe等深度学习框架或平台被大量应用于工业生产环境，使得深度学习问题能够更加简单高效地得到解决[2]。因此，现重新将这个课题拿出来，应用更加”现代化”、成熟的理论和工具，来解决猫与狗识别的问题。Kaggle官方也提供了训练数据集和测试数据集：其中训练数据集中有25000张图片，猫和狗各12500张；测试数据集中有12500张图片，猫和狗都有。

### 1.2. 问题说明

本项目要解决的问题是在猫和狗的照片中将它们正确地识别出来。这是一个典型的二分类问题，因此我们的目标是实现一个分类器：向分类器输入猫或狗的图片，分类器会输出照片中的动物是狗的概率（阈值为[0, 1]）有多大：1表示是狗，0表示是猫。因此便可以得出图片中的是猫还是狗。要实现这样的分类器，可以通过卷积神经网络，从已知的训练数据（知识）中学习猫和狗的特征，以识别出未知图片（测试数据集）中的是猫还是狗。

### 1.3. 评价指标

运用训练数据集训练好卷积神经网络模型后，用测试数据集来衡量模型的准确率。准确率，即预测正确的样本数量除以总体测试数据集数量。

另外，Kaggle还提出了一个更加严格的，计算选手的模型好坏的标准：

LogLess表示模型误差，值越小说明模型的误差越小，即模型越优。

# 二、项目分析

### 2.1 数据研究和可视化

Kaggle官方提供了训练数据集和测试数据集，其中训练集中有25000张图片，测试数据集中有12500张图片。除此之外，本项目中还是用了一份来自牛津大学的扩展数据集——*The Oxford-IIIT Pet Dataset*用于扩展训练数据集，该扩展数据中包含了37个品种的宠物（猫和狗）。

Kaggle官方提供的训练数据集包含了25000张图片，猫和狗各12500张。每张图片的文件名称中包含了”dog”、”cat”关键字以及编号，例如：cat.192.jpg、dog.90.jpg。因此可以通过文件名来标记图片中的是狗还是猫，即：标签（label）。训练集作为已知的”知识”，是分类器学习的对象，分类器将从这些标记好的图片中学习属于猫和狗各自的特征。图片中有各种场景下的不同形态的猫或狗，以猫为例：不同品种、颜色、花纹、姿势的猫，全身猫和半身猫，跟人的合影猫，两只合影的猫，不同光线条件下的猫。其中有一些异常图片，例如：cat.724.jpg，虽然训练集将它标志为”cat”，但是里面包含了狗和猫，而且狗占的面积比猫要大很多。所以训练之前需要将这些异常图片去掉。另外，训练数据集中的图片尺寸并非相同，根据不同的模型的需求，需要对图片预先进行裁剪，例如：下文中我们采用了Xception模型，需要将图片统一裁剪为299\*299的RGB图。

测试数据集中包含12500张图片，是按照数值ID来进行命名的。测试数据集用于衡量我们的分类器的准确率。分类器需要预测出测试集中每一张图片是狗的概率有多大，1表示是狗，0表示是猫。

此外，扩展训练数据中共有7349张图片。其中包含了12个品种共2371张猫的图片，和25个品种共4978张狗的图片。同上所述，扩展的训练数据集同样需要进行裁剪等预处理操作。

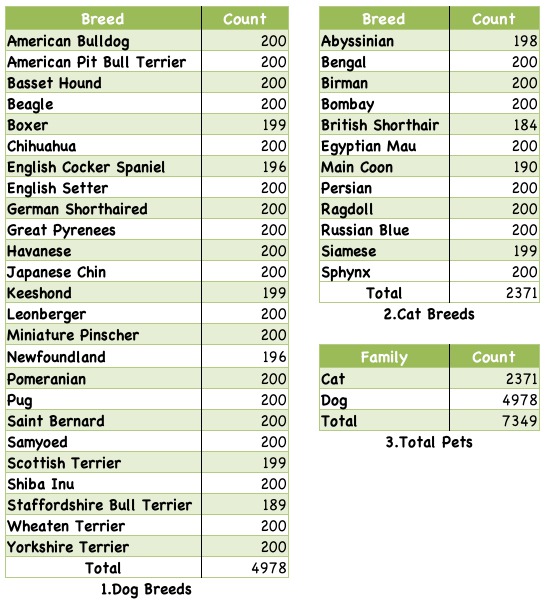


图 2.1 扩展训练数据的分布

综上，训练数据集和测试数据集的分布如图2.2所示。训练数据集中猫和狗的分布如图2.3所示，其中训练数据中猫和狗的数量不一致，猫的数量比狗少了15%，这是由于我们的扩展训练数据中狗的图片数量比猫大，但是由于训练数据基数较大，所以猫和狗的数量不一致对于模型训练结果的影响较小。

图2.2 训练数据和测试数据分布（左），训练数据中猫和狗的分布（右）

### 2.2 算法与方法

机器学习典型的两个应用场景就是分类和回归。根据前面的叙述可知，本项目要解决的是一个分类问题，更具体地说是一个二分类问题。分类问题的解决方法有很多，比如基于传统的基于统计的机器学习方法有：支持向量机、决策树、逻辑回归和高斯朴素贝叶斯等，以及本文我们要使用的深度学习（或叫深层神经网络）方法。之所以使用深度学习来进行本项目，而不采用传统的机器学习方法，是因为深度学习被证明可以更好地解决大数据量下图片分类或识别的问题。

#### 2.2.1 神经网络

神经网络是从人类脑神经元的研究中获得灵感，模拟其神经元的功能和网络结构，来完成认知任务的一类机器学习算法。早在1943年，神经网络就由McCulloc 和Pitts提出，后经由Frank Rosenblatt和Werbos等人的改进。但是受限与当时的数据量和计算条件，神经网络并不被看好，一度遭到冷落。

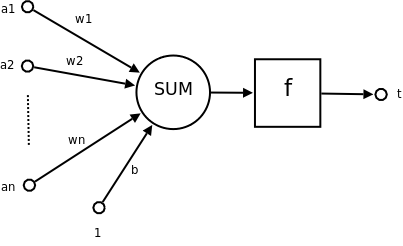
 机器学习中的神经元是以生物的神经元为原型，但是对其进行了抽象、简化。神经网络模型最基本的构成元素有：输入、激活函数和输出。神经网络由若干的神经元互相连接而成，每个神经元从其他神经元处获得输入信息，少部分神经元也从接收器获得信息；神经元处理这些输入信息，一旦被激活，就会继续发送信号至其他相连接的神经元。

图 2.3 神经元模型

上面提到了激活函数，它的作用是将输入数据的加权和进行一个映射，然后讲结果传递给下一层网络或者是输出。如果这个映射函数是非线性的，那么整个神经网络就是非线性的。非线性是神经网络的一大重要的特性，这让神经网络具有了解决一些复杂的非线性问题的能力。

#### 2.2.2 深度学习

理论上说，神经网络模型越复杂、参数越多，意味着它能完成 更复杂的任务。深度学习一般指深层神经网络，也就是加深神经网络的层数。如图2.5所示，相邻的两层之间有边相连，底层的数据输出作为其相邻的顶层的输入，然后一层一层往顶层传递，直至到达输出层。其中位于输入层和输出层之间的叫做隐藏层。

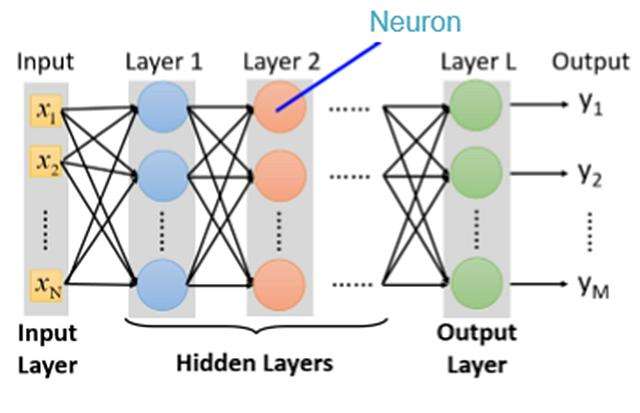


图2.4 一个典型的深度神经网络

除了激活函数外，多层次是神经网络的另一个重要的特性。Marvin Minsky和Seymour Papert曾提出：感知机（可以简单理解为单层神经网络）无法模拟异或运算。而多层网络却可以，因为多层次的结构可以让网络逐层地学习到一些不同层面或者不同抽象程度的内容。

#### 2.2.3 卷积神经网络

卷积神经网络是一种特殊的善于解决图像识别问题的深度神经网络。在图2.4中可以看到，相邻两层之间的每个结点间都有边相连，这叫做全连结网络。全连接网络最大的问题就是参数过多，导致训练效率低下。而卷积神经网络相邻两层之间只有部分结点相连。相比于全连接层，卷积神经网络的优点是可以极大降低参数的个数，并且可以提高泛化能力，因为同一层共享参数。

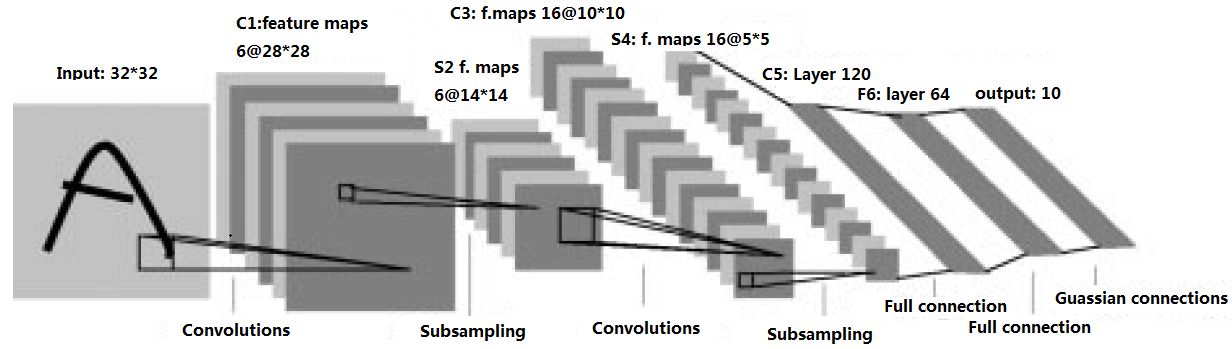


图 2.5 一个典型的卷积神经网络结构

除了结构类似外，卷积神经网络的输入输出以及训练流程与全连结神经网络也基本一致，卷积神经网络的输入层就是图像的原始像素，而输出层中的每一个节点代表了不同类别的可信度。它们唯一的不同就是相邻两层的连接方式。

#### 2.2.4 Xception模型

他山之石，可以攻玉。一般而言，从头开始训练一个复杂的模型是非常耗费经历的，而且由于数据量、枯燥的参数微调等原因，有可能得不到一个理想的结果。所以本项目中进行了迁移学习，也就是将别人训练好的模型直接引入，然后在此基础上进行调整。其中使用了Xception模型。Xception是google继Inception后提出的对Inception v3的另一种改进，主要是采用depthwise separable convolution来替换原来Inception v3中的卷积操作。

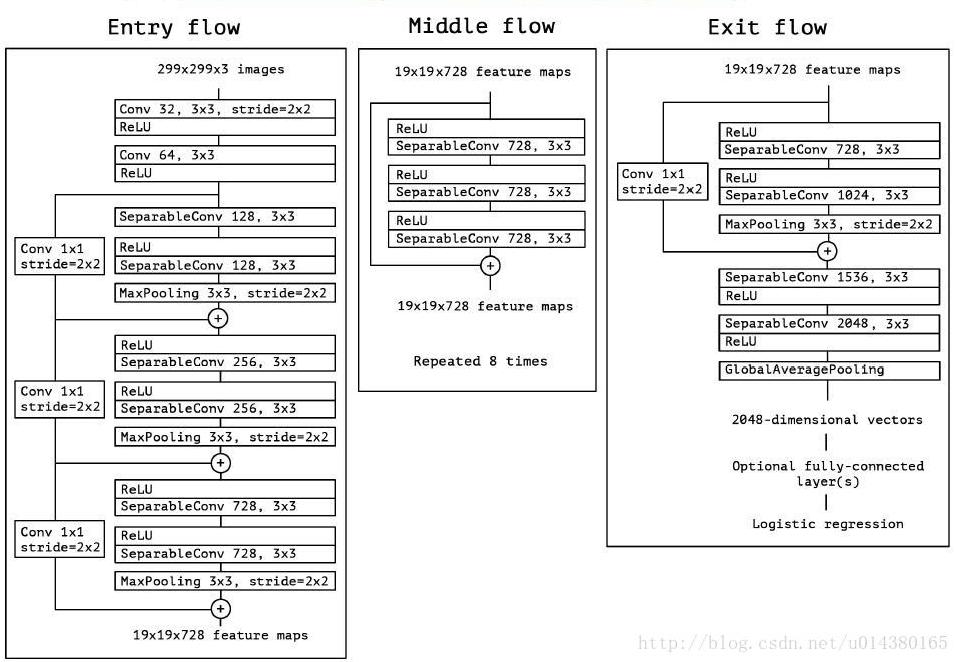


图 2.6 Xception模型的结构图

### 2.3 基准测试

本项目使用了基于Xception模型的迁移学习。Xceptio模型的权重由ImageNet数据集训练而成。在ImageNet上,该模型取得了验证集top1 0.790和top5 0.945的正确率。

在1.3 小节中提到，既然是Kaggle比赛项目，Kaggle官方提供了一套评价标准。对于12500条测试数据，参赛者需要提交一个相应具有12500条结果的csv文件。此csv文件中的每一条记录了测试图片的编号，以及模型预测的该图片是狗的概率——1表示是狗，0表示是猫。然后Kaggle通过下面的公式来计算模型表现的好坏：

LogLess表示模型误差，值越小说明模型的误差越小，即模型越优。其中，n是测试集中的样本总量；是模型预测的图片是狗的概率；log为自然对数。

目前，该比赛已经结束，总有1314个团队参赛，本项目的目标是进入排名的top 10%，也就是至少进入前131名，LogLoss最多在0.06127。

# 三、实现方法

### 3.1 数据预处理

由于本项目使用了基于Xception的迁移学习，所以应该预处理图片使其符合Xception模型的要求。

首先是通过通过OpenVC库读取图片文件，然后进行裁剪，使其成为299\*299尺寸的RGB图，所以最终程序中最终获得的是299\*299\*3的数字矩阵。

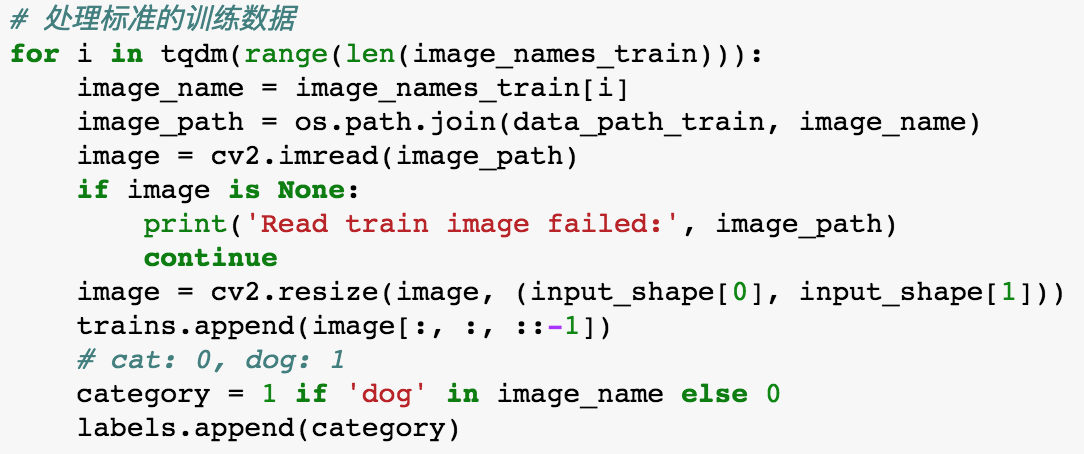


图 3.1 预处理测试数据

由于扩展的训练数据中的图片名中没有”dog”、”cat”关键字，所以只能通过种类来区分该图片属于哪个分类。这其中进行了防御性编程，因为扩展数据集集中除了.jpg图片外，还有.mat文件，以及会出现读取图片失败的情况（处理标准训练数据和测试数据时也是类似），这种情况下需要将读取的数据丢弃。

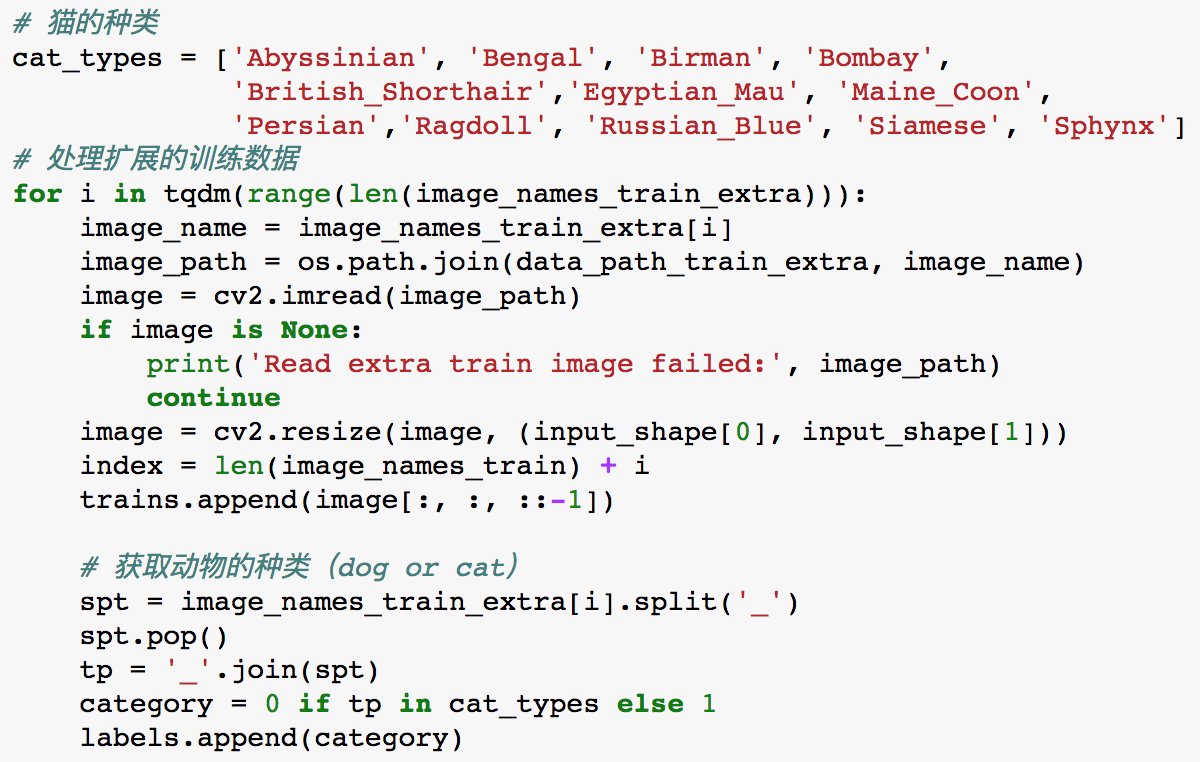


图 3.2 预处理扩展测试数据集

最终将训练数据集（trains）、标签集（labels）、测试集（tests）转化为numy数组后，调用Xception内置的预处理函数——xception.preprocess\_input来进行其他的预处理工作。

TODO：清除异常数据

### 3.2 实施

本项目使用Xception模型，该网络已经在ImageNet数据集上进行了训练，因为ImageNet数据集包含多种“猫”类和多种“狗”类，这个模型已经能够学习与我们这个数据集相关的特征了。我们只利用Xception网络的卷积层部分，把全连接以上的部分去掉。然后在训练集和测试集上跑一遍模型，将得到的训练集和测试集中图片的特征（即“bottleneck feature”，网络在全连接之前的最后一层激活的feature map）记录在两个numpy数组里。同时，对提取出来的bottleneck feature进行全局平均池化，这样做一方面是为了减少后续训练参数，提高训练效率；另一方面也可以防止过拟合。另外，为了提高后续的训练效率，将得到的两组特征保存到本地（因为提取完一遍特征是非常耗时的操作，而且后续不需要重复提取），以方便后续的模型调整。

我们将训练数据集拆分为：训练集和验证集，其中验证集占比0.2，并且随机打乱数据的顺序。

然后，我们基于记录下来的训练集特征构建新模型。这个时候新模型的输入，也就是上面提取的训练数据集的特征，已经不是原始的图片，而是经过Xception模型（除了最后的全连接层）训练过得到的特征向量，该向量的第一个维度是对应于每一张图片样本；第二个维度是经过全局平均池化之后的卷积核输出，对于Xception来说这个维度的长度是2048。

新模型的搭建较为简单，只需要训练一个输入特征长度为2048的全连接层即可。

### 3.3 改进

为了防止过拟合，我们使用了Dropout。由于是二分类问题，我们使用的sigmoid激活函数，损失函数选用binary\_crossentrop。另外，我们选用了adadelta优化器，参数皆使用默认值（默认值来源于论文，已经具有较优的表现，所以我们没有再做调整）。

我们知道，深度学习的流行，很大程度上也得益于大数据技术的发展，和计算机计算能力的提升。一般而言，深度学习的训练数据集越大，模型越可能学习到数据中存在的特征，模型准确率也就越高。因为除了Kaggle官方提供的训练数据外，还额外引入了牛津大学的扩展数据集——*The Oxford-IIIT Pet Dataset*用于训练，该扩展数据中包含了37个品种的宠物（猫和狗）。

# 四、结果

### 4.1 模型评估与验证

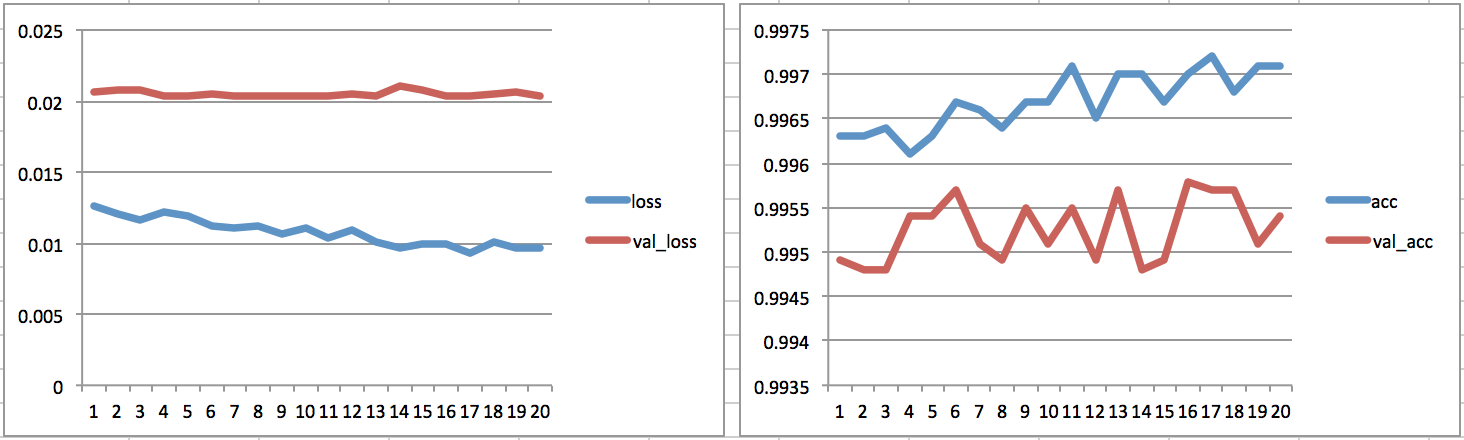
因为使用了迁移学习，基于Xception模型，我们构建的新模型较为简单，参数较少，且诸如adadelta优化器我们使用了默认参数（这也是推荐的做法，因为默认参数来源于论文，被证明是较优的）。由于在前面我们利用预训练好的Xception模型提取了图片的特征，这些特征已经能够较好地体现图片自身的特性，所以仅仅再经过一个全连结层之后得到的效果已经非常好（Kaggle排名21名，已经达到预期）。所以这里可微调的项也比较少。

TODO 比较Dropout不同数据

TODO 比较不同adadelta

TODO bitch\_size

模型搭建好后接着进行训练，我们设置的batch\_size是32，epochs为20。得出的训练集和验证集上的loss随着epochs的变化趋势如图4.1所示。可以看出epoch 4之后验证集上的loss开始趋于平稳，而训练集上的loss依然在呈下降趋势；epoch 6之后验证集上的准确率开始出现震荡，且没有明显提升，而训练集上的准确率整体上还在呈上升趋势。说明随着epoch的增大，模型有过拟合的分险，因此epochs应控制在10以内，例如设为6。



4.1 训练集和验证集上损失函数的值随着epochs变化情况

### 4.2 结果分析

从上面的描述可得：我们的模型识别准确率比较高，达到了99%以上，识别猫和狗的效果不错。虽然我们的模型得出的结果达到了前21名的成绩，但是跟公开排行榜上的第一名还是有一些差距。可能是由哪些原因造成的呢？下图展现了在12500个测试数据上，预测是狗的概率在0.4 ~ 0.6之间的图片（也就是说模型不能准确判断该图片是狗还是猫）。

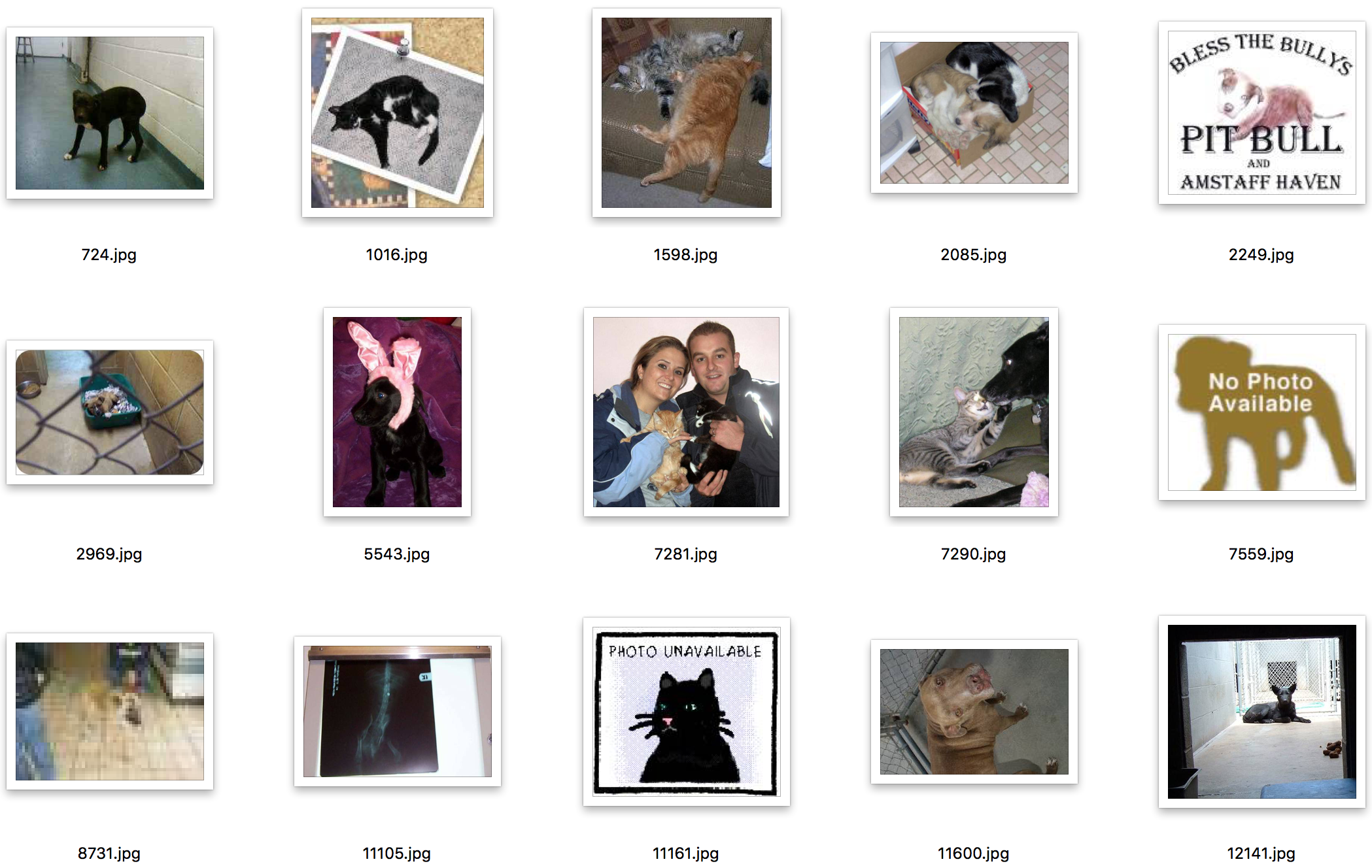


图 4.3 模型无法准确分类的图片示例

经过观察可知，模型无法分类这些图片可能由以下几种原因：

(1) 图片本身质量太差，例如：8731.jpg图片尺寸很小，画面很模糊，人眼也很难识别；其他的还有7559.jpg、11105.jpg等。

(2) 动漫类型的图片，风格跟真实的照片不一样，例如：2249.jpg、11161.jpg等。

(3) 图片中动物所占据的区域较小，而障碍物比较多，例如：7281.jpg、2969.jpg等。

(4) 图片中猫或狗的品种在训练数据中没有覆盖到，例如： 11600.jpg、12141.jpg。

所以模型仍然有提升的空间。

# 五、结论

### 5.1 自由形态的可视化

经过上面模型的预测，将得出的预测结果（.csv文件）上传到Kaggle查看得分，是0.04213。根据得分榜，可以排到第21名，达到了前面提出的进入前10%要求（也就是前131名）。

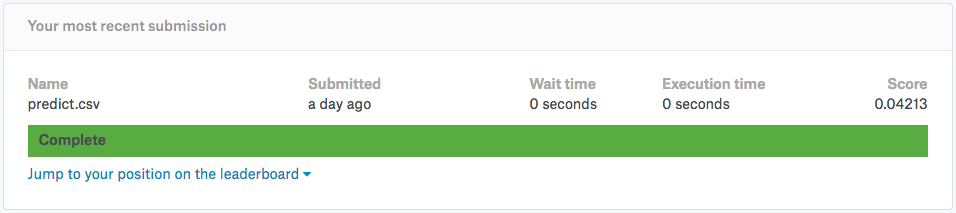


图 4.1 Kaggle得分

对于模型预测的12500张图片，我们分析了每张图片是狗的可能性数据。这个数值越接近1说明图片中是狗的可能越大；越接近0说明是猫的可能性越大。例如：数值在0.9 ~ 1.0之间的便可以认为是狗的可能性是极大的；数值在0.0 ~ 0.1之间的便可以认为是猫的可能性是极大的；而在0.5左右的，例如在0.4 ~ 0.6之间的便可以认为模型无法确定是猫还是狗。我们以0.1为组间距，分析了各个区域的测试数据量，如图4.2所示。

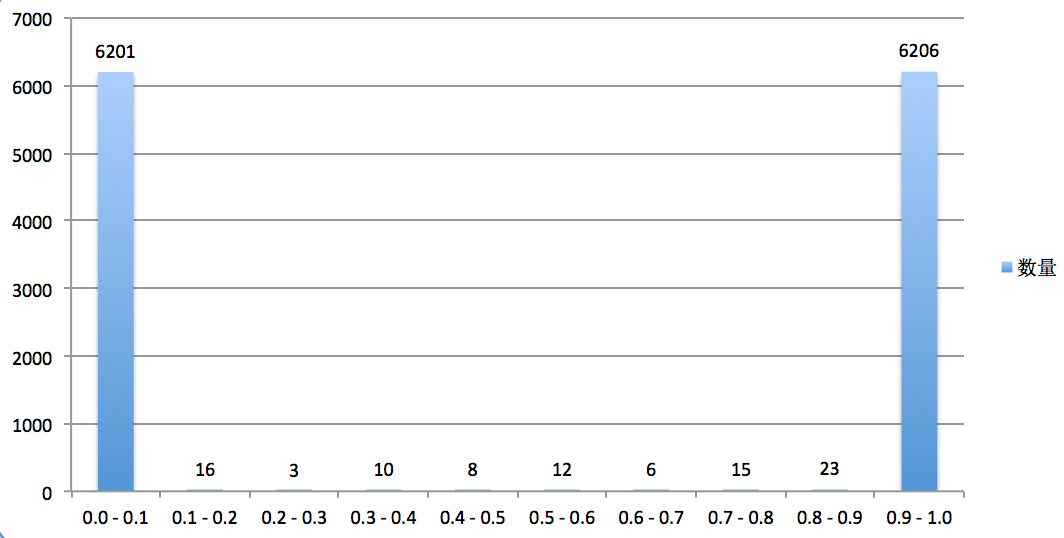


图 4.2 测试数据是狗的可能性分布情况

可以看出，绝大部分数据落在了0.0 ~ 0.1和0.9 ~ 1.0这两个区间，占比99.256%。

### 5.2 思考

解决类似图像分类问题的一个可行的步骤是：先明确问题域；分析已有的数据，大致勾勒出一个解决方案；具体实施中首先需要进行数据预处理，然后构建模型，然后进行训练，参数调优；预测。

卷积神经网络对于图像识别具有很好的效果。本项目使用了预训练的Xception模型进行迁移学习，只保留Xception的卷基层，先提取出训练数据集和测试数据集中的图片特征。为了减少训练的参数个数，以及避免过拟合，利用了全局平均池化方法，所以最终Xception提取出的每张图片的特征是一个长度为2048的numpy数组。然后在此基础上加了一个全连结层。实验效果良好，提交到Kaggle后得分是0.04213，排名21，满足了前面制定的最低要求。

相比于传统的机器学习算法，卷积神经网络在图形识别方面展现了很好的效果。Xception模型是基于卷积神经网络技术，提出了一个复杂且高效的网络结构。并且预训练的Xceptio模型的权重由ImageNet数据集训练而成，已经能够非常好的概括图像的特征。所以我们基于Xception的迁移学习所得出的模型准确率比较高。

牛顿说：”如果说我比别人看得更远些，那是因为我站在了巨人的肩上”。这告诉我们要善于学习、利用前人的研究成果，并应用于解决自己的问题，或者在此基础上在提出新的想法。本项目使用的迁移学习方法就是一个例子，基于预训练好的Xception模型，来解决图像识别问题。一方面这样可以节省很多手动搭建卷积神经网络的步骤，另一方面极大地节省了繁琐的参数调整的时间。所以这种方式可以非常适合快速地将想法付诸实践，或者解决一些较复杂的问题。

### 5.3 改进

可以看到，我们的模型最终预测的得分跟Kaggle上的最好成绩仍有一定差距。4.2节中也提出了一些造成这种结果的原因，一个可行的解决方案是对原图片进行更多的预处理，例如随机地进行裁剪、旋转、翻转、调整饱和度等等预处理手段。

另外，一个实践上的改进是，后续可以开发一个APP、HTML 5或者小程序，接入我们的服务，用户利用手机的摄像头拍摄图像，实时传到我们的后端服务，然后后端会调用我们的模型进行预测，然后将结果返回给用户。

# 六、引用文献

Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

Minsky M, Papert S. Perceptrons[J]. American Journal of Psychology, 1969, 84(3):449–452.

Zeiler M D. ADADELTA: An Adaptive Learning Rate Method[J]. Computer Science, 2012.

http://www.robots.ox.ac.uk/~vgg/data/pets/

Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions[J]. 2016:1800-1807.

https://keras-cn-docs.readthedocs.io/zh\_CN/latest/blog/image\_classification\_using\_very\_little\_data/