猫狗大战

一．问题的定义

项目概述

猫狗大战是 kaggle 的一个经典的比赛，最初举办是在 2013 年。猫狗大战要求参赛者分辨猫或者狗的图片，并一度成为 Kaggle 上最受欢迎的比赛。随着机器学习算法的发展，尤其 是深度学习的大规模应用，kaggle 在 2016 年又重新开启这个项目。

问题陈述

猫狗大战要求参赛者训练一个机器学习的模型，用来判别图片是猫还是狗，并给出此图片是狗的概率。比赛给出了一个包含 25000 张图片的训练集以及一个包含 12500 张图片的测试集。

这是一个二分问题，要求参赛者将图片做分类并给出图片属于此类别的概率。借助于深度学习，我们可以实现一个端到端的预测模型，将图片输入到模型中，然后模型输出图片为狗概率。

评价指标

关于模型评估标准，我们采用与 Kaggle 最终评判标准相同的方式。对验证集数据做预测，并计算与真实值之间的 binary_crossentropy，计算公式如下：

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

其中 n 是测试集图片的数量，$\hat{y}$是算法预测这张图片是狗的概率(0~1)，y 是图片标签，表示图片所属种类（0 表示猫，1 表示狗）。LogLoss 越小，表示模型预测结果越好。

二．分析

数据探索与可视化分析

数据来源

猫狗大战的数据集来自于 Asirra（Animal Species Image Recognition for Restricting Access），这是一个图灵测试的项目，要求被测试者分辨出图片是猫还是狗。在深度学习还未蓬勃发展之前，分辨猫狗对于计算机来说是一项困难的工作但是对于人来说可以很轻易的完成，故可以分辨出测试者是否是人，所以 Asirra 是一个非常成功的图灵测试项目。

Asirra 的数据主要来自于 Petfinder.com 这个世界最大的流浪宠物认领网站。他们为微软亚洲研究院提供了三百万张被人工标注的猫或者狗的图片。本项目所用的数据集为这个数据的子集，由 Kaggle 提供。

数据初探

Kaggle 为猫狗大战提供了两个数据集：训练集和测试集。其中训练集包括 25000 张图片，测试集包括 12500 张图片。

训练集包含了 12500 张猫的图片以及 12500 张狗的图片，通过图片命名加以区分，猫的图片命名为 cat.0.jpg ~ cat.12499.jpg，狗的图片命名为 dog.0.jpg ~ dog.12499.jpg。我们从训练集中抽取 5 张猫和狗的图片如 Figure 1、Figure 2 所示。很明显，人类可以分辨出这十张图片中的猫和狗，所以我们可以期待通过通过深度学习的方法来辨别图片。

Figure 1 **训练集-猫示例图片**

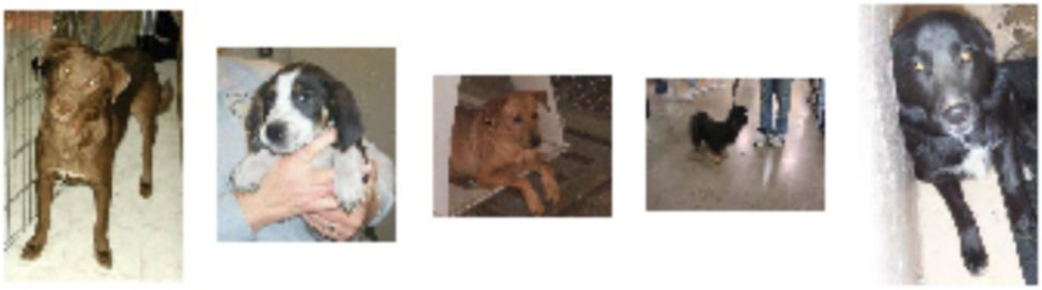


Figure 2 **训练集-狗示例图片**

机器学习模型做图片分类时一般要求其输入固定尺寸，但从上述图片中可以看到训练集中的图片尺寸大小是不同的，需要将图片进行缩放。但在此之前，我们对训练集数据尺寸大

小做统计，结果如 Figure 3 所示。除了有两张图片尺寸较大，其余数图片集中在(200,200)~(500,500)范围内。这位之后将图片 resize 到统一尺寸提供了合理性支持。

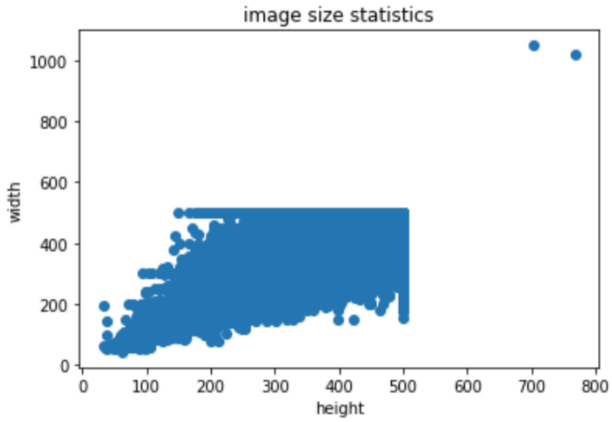


Figure 3 **测试集-图片尺寸**

再次浏览训练集，发现测试集中有些图片非猫非狗，或者猫狗并非图片主要元素，如 Figure 4 所示。其中 cat.6.jpg,cat.2337.jpg,dog.8671.jpg 这三张图片中猫狗只占了很小的篇幅，cat.3261.jpg,dog.9571.jpg,dog.10237.jpg 三张图片非猫非狗。这说明在训练集中存在脏数据，在使用训练集之前我们应该做数据处理。

Figure 4 **异常值数据**

算法和技术

为了实现一个准确的二分类模型，我打算在项目中首先使用在 imagene 冠军模型进行数据清理；然后基于迁移学习，使用 ResNet、Inception、Xception 三个模型对数据进行了特征提取；对提取的特征做特征融合，搭建全连接层进行预测；最后对结果进行 clip 后提交。本章依次介绍上述四种算法。

ImageNet 比赛与冠军模型

ImageNet Large Scale Visual Recognition Challenge(ILSVRC)从大尺度方面评估算法物体检测和图像分类能力。其中一个崇高的目标是允许研究员利用相当昂贵的数据标会成果通过

判别大量不同的物理来比较当前物体检测方面学术的进展。另一个动机是测量计算机视觉在大规模图像索引检索和注释方面的进展。

从 2012 年以来，每年 ILSVRC 都举办一次，并根据结果评出当年的冠军模型，冠军如 Table 1 所示。

Table 1 ImageNet **比赛结果**

年份	2012 冠军	2014 年亚军	2014 冠军	2015 冠军
冠军模型	AlexNet[6]	VGG[5]	InceptionNet[2]	ResNet[3]
Top-5 error	15.3%	7.3%	6.7%	3.57%

resnet，InceptionV3 和 Xception。

Resnet 提出了一个残差学习框架，更容易去最优化，并且可以从相当大的深度中获取精度。此外深度残差网络在具有深度的同时还很好的控制了参数的数量。InceptionV3 则是通过计算同一输入映射上的多个不同变换，拓宽了模型的宽度，更将大的卷积重构成连续的小卷积，降低了训练时的计算复杂度。Xception 提出了一个完全基于深度可分卷积层的卷积神经网络结构。

Dropout

Dropout 是指在深度学习网络的训练过程中，对于神经网络单元，按照一定的概率将其暂时从网络中丢弃。Dropout 是 CNN 中防止过拟合提高结果的一个大杀器。每一次将网络中部分隐藏层神经元删除，就相当于训练了一个由输入和输出以及剩下的神经元组成的网络，多次进行神经元删除就相当于训练了多个网络，最后形成的网络实质是由若干个子网络组成的新的网络。同时在预测结果的时候也是一种投票机制，就是这些子网络中大多数的网络认为是这个结果，最后网络的输出就是这个结果。

Adam 优化算法

Adam 优化算法是随机梯度下降算法的扩展式。随机梯度下降保持单一的学习率更新所有权重，学习率在训练过程中并不会改变。而 Adam 通过计算梯度的一阶矩估计和二阶矩估计而为不同的参数设计独立的自适应性学习率。是深度学习中一种常用的优化算法。

迁移学习

迁移学习(Transfer learning) 顾名思义就是就是把已学训练好的模型参数迁移到新的模型来帮助新模型训练。考虑到大部分数据或任务是存在相关性的，所以通过迁移学习我们可以将已经学到的模型参数（也可理解为模型学到的知识）通过某种方式来分享给新模型从而加快并优化模型的学习效率不用像大多数网络那样从零学习（starting from scratch，tabula rasa）。

ImageNet 的数据量大且复杂，通过其训练出的模型具有很强的泛化能力，而且 ImageNet 中本身就包含了猫/狗的类别，其中狗的分类有 118 种，猫的分类有 7 中。使用 ImageNet 冠军模型以及其权值作为 fine-tuning 的 pre-trained model 或者使用冠军模型做特征提取可以取得很好的效果。[7]

Clip

当模型预测结果与实际结果相反（例如，模型将一张猫的图片预测为 1 概率的狗），若直将预测结果接带入公式计算 Logloss 会出现 Logloss 无穷大的情况，此时 Logloss 失去了

表征模型能力好坏的能力。所以在计算 Logloss 算法的具体实现中，往往会先将模型预约结果限定在一个不包含 0 的范围内，然后再带入 Logloss 公式进行计算。猫狗大战里 Kaggle 计算 Logloss 的具体代码我们不得而知，但我们可以参照 sklearn.metrics.logloss 的代码

```
1666        # Clipping
1667        y_pred = np.clip(y_pred, eps, 1 - eps)
1668
```

Figure 5 numpy.clip **源码**

其中 eps 默认值为 1e-15。

为了提高提高比赛成绩，当模型预测准确率很高时，可以通过将预测结果手动限定在更小的区域内来减小 Logloss。Table 2 分别计算了不同范围下的 Logloss（sklearn.metrics.log_loss 要求类别大于 2，所以选取了两个样本）。可以得出对正确的预测结果进行 clip 带来的 Log_loss 的提高远远小于对错误的结果进行 clip 带来的 Log_loss 的减小，所以正确使用 clip 方法可以减小最终的 Log_loss.。

Table 2 clip **与** log_loss

真实值	[0,1]	[0,1]	[0,1]	[0,1]
预测值	[0,1]	[0.005,0.995]	[1,1]	[0.995,0.995]
Logloss	9.992e-16	0.005	17.27	2.652

基准模型

根据 Udacity 课程要求，猫狗大战项目的最低要求是 Kaggle Public Leaderboard 的前 10%，即测试集的 Logloss 要小于 0.06。

三．方法

数据预处理

如第二章所示，训练集中存在脏数据，需要对训练集做数据清理，去除掉脏数据从而提高模型预测准确性。数据清洗包括异常值辨别以及异常值的处理。

1. 异常值辨别

训练集为图片非数值，所以我们无法通过四分卫，均值，KNN 等方法辨别出其中的异常值，这些图片均为非猫非狗的复杂图片，我们可以采用 ImageNet 的预训练模型行进行异常值的辨别。ImageNet 的模型对 1000 个物品进行分类，其中包含了 118 个狗的细分类以及 7 个猫的细分类。https://blog.csdn.net/zhangjunbob/article/details/53258524 给出了 ImageNet 数据集的分类（猫/狗所属类别请参照附录代码）。

ImageNet 的模型有一个 Top-N 的参数，它表示模型输出图片概率最大的 N 个分类，而 ImageNet 也有两个评价指标 Top-1 和 Top-5 准确率， Figure 6 展示了历年来 ImageNet 优秀模型的准确率

Documentation for individual models

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.715	0.901	138,357,544	23
VGG19	549 MB	0.727	0.910	143,667,240	26
ResNet50	99 MB	0.759	0.929	25,636,712	168
InceptionV3	92 MB	0.788	0.944	23,851,784	159
InceptionResNetV2	215 MB	0.804	0.953	55,873,736	572
MobileNet	17 MB	0.665	0.871	4,253,864	88
DenseNet121	33 MB	0.745	0.918	8,062,504	121
DenseNet169	57 MB	0.759	0.928	14,307,880	169
DenseNet201	80 MB	0.770	0.933	20,242,984	201

The top-1 and top-5 accuracy refers to the model's performance on the ImageNet validation dataset.

Figure 6 ImageNet **冠军模型**

我们使用 Top-5 准确率最高的 Xception[1]模型对训练集进行预测，将 Top-5 分类中没有对应猫/狗的视为异常值。对整个训练集进行检测，总共得到 606 个异常值，随机选取五张图片如 Figure 7 所示。

Figure 7 Xception_Top5_**异常值**

发现用 Top-5 的预测值来做异常检测的得到的结果能够辨别出异常值图片，但是同时也包含了许多包含猫/狗的图片，效果并不是非常理想。接下来我们尝试了不同的 top-N 值，得到结果如 Figure 8 所示

Figure 8 Xception_Top-N_**异常值**

最终我们选取了使用 Top-50 中是否包含图片对应的猫/狗种类来辨别图片异常值。

接下来我们使用了 InceptionV3 和 ResNet 模型采用上述标准对训练集进行异常值检测并与 Xception 所得结果进行对比，发现三种模型所检测到的异常值并不相同，InceptionV3 和 ResNet 新检测出的异常值如 Figure 9 所示。

Figure 9 不同模型下的新异常值

最终我们将 Xception、InceptionV3、ResNet 所检测出的异常值取交集，作为最终最终检测出的异常值。

2. 异常值处理

采用上述方法最终得到 195 个异常值数据，占测试集总数据的 0.78%，我们采用直接将异常值数据剔除的方法处理异常值数据。

执行过程

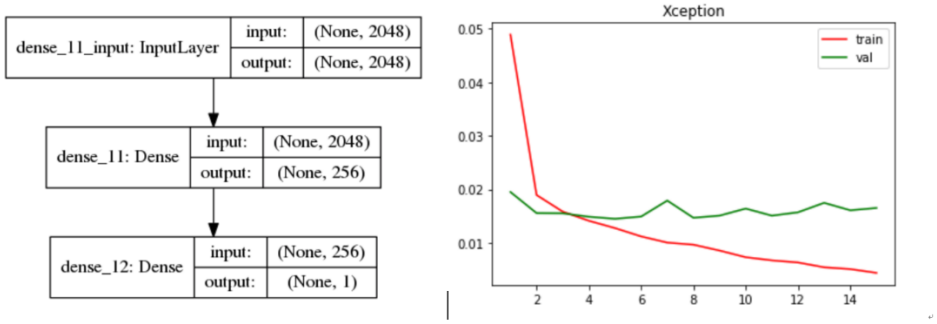
整个模型的训练过程采用了单模型预测和多模型提取特征、特征融合两种方法。

单模型预测

使用 ResNet，InceptionV3，Xception 三个模型提取的特征值，重新设计全连接层，进行训练。报告文中给出 Xception 的训练过程，ResNet 和 InceptionV3 使用同样的训练方法，并给出了最终结果。

Xception 提取特征值后全连接层构建如下：

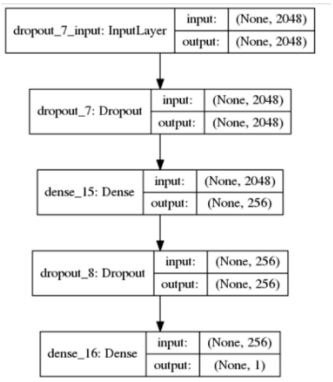
首先随意设计一个全连接模型，模型与训练结果如下所示

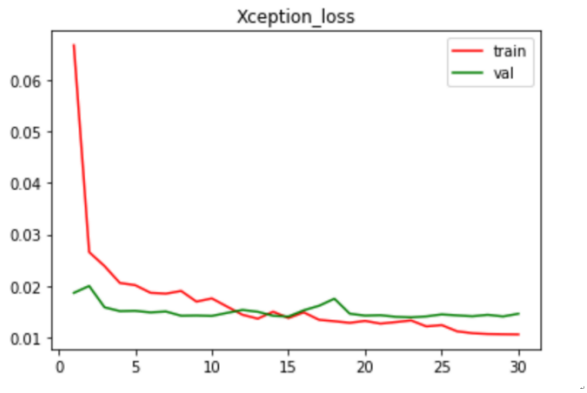


Train_loss	Train_acc	Val_loss	Val_acc
0.004	0.9991	0.0165	0.9944

总的来说，这个模型取得了不错的效果，在验证集上得到了 99.44%的准确率。但是，通过观察 train_los 和 val_loss 可以发现，在第三个 epoch 后，模型开始出现过拟合，train_loss 持续下降但是 val_loss 并没有下降。

通过添加过 Dropout 的方法来抑制过拟合，添加两个 dropout 层后模型与训练结果如所示

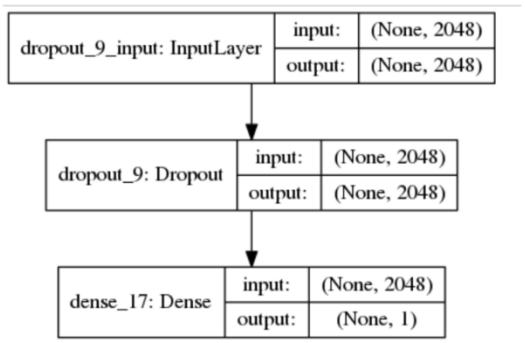


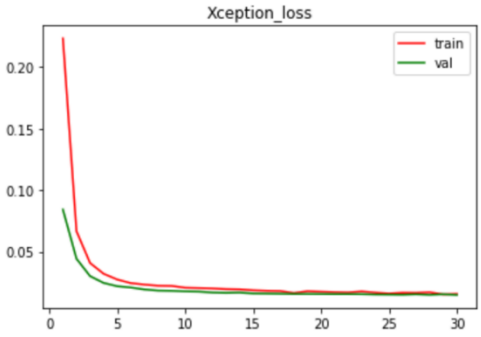


Train_loss	Train_acc	Val_loss	Val_acc
0.0137	0.9954	0.0140	0.9950

添加 dropout 经过 17 个 epoch 模型也出现了过拟合现象，对于过拟合的抑制效果并不是很明显，模型在多个 epoch 后都出现一定的过拟合现象。

接下来通过简化全链接层的减小模型 capacity 来抑制过拟合，将全连接层数减为 1 之后，模型与训练结果如所示





Train_loss	Train_acc	Val_loss	Val_acc
0.0155	0.9951	0.0147	0.9956

虽然验证集上的准确率和 loss 与添加两层 dropout 相同，但是此模型并没有过拟合，模型也更加简单。

最终 Xception 单模型预测采取采用带有权值的 Xception 模型做特征提取，然后采用上述第三种全连接层的模型做预测，最终在验证集上的 loss 为 0.0147。

对于 ResNet，InceptionV3 提取的特征值采用同样的全连接结构进行模型训练和预测，训练结果如 Figure 10、Figure 11 所示

Train_loss	Train_acc	Val_loss	Val_acc
0.0552	0.9777	0.0719	0.9702

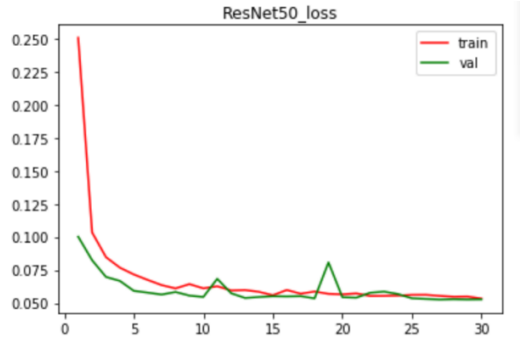


Figure 10 ResNet **训练结果**

Train_loss	Train_acc	Val_loss	Val_acc
0.0158	0.9951	0.0176	0.9940

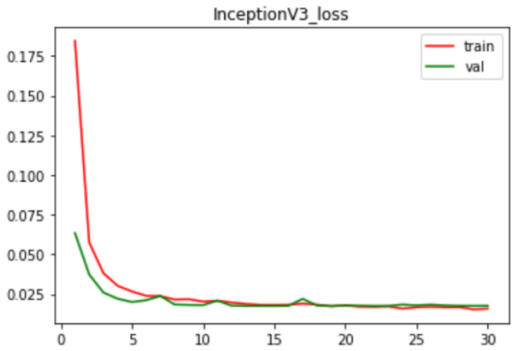


Figure 11 InceptionV3 **训练结果**

单模型的预测能力 Xception 最强，将 Xception 预测结果提交 kaggle，得到 Loss0.04077，这个结果已经超过了我们的基准模型，达到了预期的目标。

多模型融合

为了探索如何达到更好的效果，进一步减小 loss，接下来使用特征融合的方法将上述 Imagenet 的模型提取的特征融合，再做预测。

我们测试了 ResNet50 + InceptionV3 、 InceptionV3 + Xception 和 ResNet50 + InceptionV3 + Xception 三种模型，训练结果及提交 Kaggle 成绩如 Table 3 所示。

Table 3 **模型融合结果**

	Train_loss	Train_acc	Val_loss	Val_acc	Kaggle_loss
ResNet50 + Xception	0.0151	0.9953	0.0145	0.9956	0.3899
Inception + Xcetpion	0.0135	0.9960	0.0127	0.9964	0.3857
All three	0.0127	0.9958	0.0126	0.9964	0.3729

最总我们采用将 ResNet50、InceptionV 和 Xception 三个模型提取的特征融合，然后构建全连接层的模型最终最终我们所采用的模型。

完善

Clip 带来的提升

在算法与技术章节介绍了通过将最终结果限制在一定区域来减小 Log_loss 的方法，本节就来探索一下 Clip 方法是否能够有效减小 Log_loss 以及如何确定一个合适的参数。

由于我们无法得知测试集的 label，所以无法判断被 clip 掉的数据对最终 Log_loss 的影响。在本项目中，我们采用 gridsearch 的方法，选取若干 Clip 阈值选取结果最好的一组作为本项目最终采用的 Clip 阈值。我们实验了[0,1]、[0.002,0.998]、[0.005,.0995]、[0.01,0.99]四种不同的阈值，结果如 Table 4 所示。([0,1]表明我们不进行额外的 clip，与 Kaggle 计算 Log_loss 时如何进行 clip 无关)。最终我们将 pred 结果限定在 0.005-0.995 之间。

Table 4 clip **效果**

Clip 阈值	Kaggle 成绩
0-1	0.0633
0.002-0.998	0.03860
0.005-0.995	0.03729
0.01-0.99	0.3892

数据预处理带来的提升

在数据探索章节，我们发现训练数据中存在脏数据，并提供了使用 imagnet 模型辨别异常值的方法。剔除异常值确实会对模型辨别猫狗能力带来提升，但是由于我们无法对测试集做同样的操作，模型最终在测试集上的表现如何还需要实践来检验。Table 5 给出了使用原始数据以及使用清洗后的数据训练出的模型取得的最终成绩，可见将异常值剔除之后，最终的预测结果提高了。

Table 5 **数据清理效果**

是否进行数据清理	Kaggle 成绩
否	0.03841
是	0.03729

四．结果

本项目最终采用了使用 imagenet 预训练模型提取特征，特征融合，构建全连接层训练的方法，经过比较最终选取了 Figure 12 所示模型为最终模型。

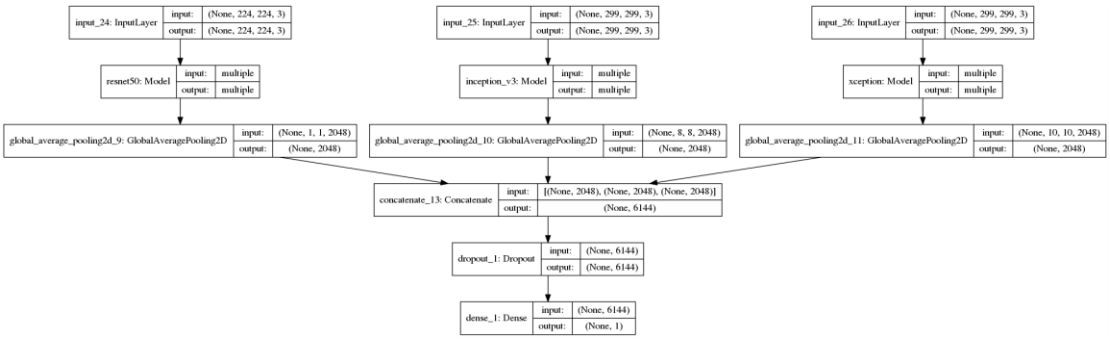


Figure 12 **最终模型描述**

并使用了对训练集进行数据清理以及对最终结果使用 clip 的方法来提高成绩，最终结果提交 Kaggle 获得 Log_loss 为 0.3754。

最终模型的效果已经远远超过了基准 Log_loss 小于 0.06 的要求，在 kaggle public leader board 排名 12。

五. 项目结论

结果可视化

相较于 Kaggle 评分，这里给出更直观的结果可视化的方式，随机选取了 15 张图片，并给出其分类结果以及概率。（为了便于显示将概率最大值限定到 0.995）

Figure 13 **结果可视化展示**

对项目的思考

在项目中我们采用迁移学习的方法，首先对数据进行异常值判断，并剔除异常值完成数据清理，接着使用多个 imagenet 冠军模型对图片进行特征提取，然后使用特征融合的方法

将不同模型提取的特征进行融合，最后搭建全连接层进行预测。以及针对 Log_loss 计算的特点使用了 Clip 方法减小 Log_loss。

最终在验证集上取得了 99.5%的准确率，测试集的 Log_loss 为 0.3754。对猫狗的分类取得了卓越的效果，是一项了不起的工作。

需要做出的改进

我们可以通过数据增强的方法来增加训练数据（对数据进行缩放、翻转、平移等变化），提高模型泛化能力；另外我们可以更换特征提取的模型，使用更强大的模型如 DenseNet 和 SENet 等。我相信通过上述的方法可以进一步提高模型准确程度。

然而我们不能够忽视的是，为了提高准确率，我们构建了一个非常非常复杂的模型，整个模型参数有 70,000,000 个，在一台装有 Ubuntu16.04，Intel Core i-8500，24GB 内存，Nvidia GTX1060 的电脑上对 100 张 3.5s 的图片做预测耗时为。在移动端或者无 GPU 机器处理的时间会更长，所以并不具有很强的推广性。如何简化模型，是本项目需要做出的改进，也是要应用到实际项目中需要进一步做出的努力。

参考文献

[1]. Chollet F. Xception: Deep learning with depthwise separable convolutions[J]. arXiv preprint, 2016.

[2]. Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2818-2826.

[3]. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[4]. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. Cvpr, 2015.

[5]. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[6]. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[7]. 为什么要迁移学习？ https://www.zhihu.com/question/41979241