猫狗大战

一、定义

项目概览

猫狗大战（Dogs vs Cats）是源自 kaggle 上 2013 年的一个竞赛项目。这个项目主要是为了使用机器学习算法进行图片分类，尤其是深度学习。项目中提供了一个数据集，包括 25000 个已标定的数据和 12500 个未标定数据。通过这 25000 个已标定数据来构建相应的预测算法模型，并使用这个算法模型对 12500 个未标定的数据进行预测，并将预测的结果提交到 kaggle 上，以得到一个综合评分，以评价预测算法模型的优良水平。

近些年，随着深度学习的突破性进展，整个机器学习领域也发展到了新的高度，尤其是卷积神经网络（CNN）在图片识别方面，其识别的准确性已经超过人类的识别水平。卷积神经网络可追溯到上个世纪 80 年代，Fukushima 首次提出 neocognitron 模型，可以看作是卷积神经网络的第一个实现。经过若干年的发展，直到 2006 年深度学习的理论被提出后，卷积神经网络的表征能力得到了关注，并随着数值计算设备的更新开始快速发展。自 2012 年的 AlexNet 开始，卷积神经网络多次成为 ImageNet 大规模视觉识别竞赛（ImageNet Large Scale Visual Recognition Challenge, ILSVRC）的优胜算法，包括 2013 年的 ZFNet，2014 年的 VGGNet 和 GoogleNet，2015 年的 ResNet 等。

问题说明

猫狗大战提供的图片源自真实拍摄，分辨率差异较大。图片中猫和狗的颜色丰富，类型多样，姿态迥异，同时还有复杂的背景，极大的增加了识别的难度。

本项目将使用基于深度学习的卷积神经网络（CNN）来构建预测算法模型，通过 25000 个已标定数据（train data）来训练神经网络，然后使用训练好的神经网络模型来预测未标注的数据（test data），来完成预测准确度排名进入到 kaggle 上猫狗大战 Public Leaderboard 排名前 10%的项目目标。

评价指标

在机器学习领域，通常会对算法模型设定损失函数作为评价指标，以便更好的评价算法模型的优良。常用的分类问题，都会使用交叉熵作为损失函数，猫狗大战属于二分类图像识别问题，需要使用了二分类交叉熵（binary_crossentropy）作为损失函数。具体公式如下：

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

二分类交叉熵，需要在神经网络的最后一层使用 sigmoid 作为激活层来配合使用，以便可以对结果进行很好的分类。

二、分析

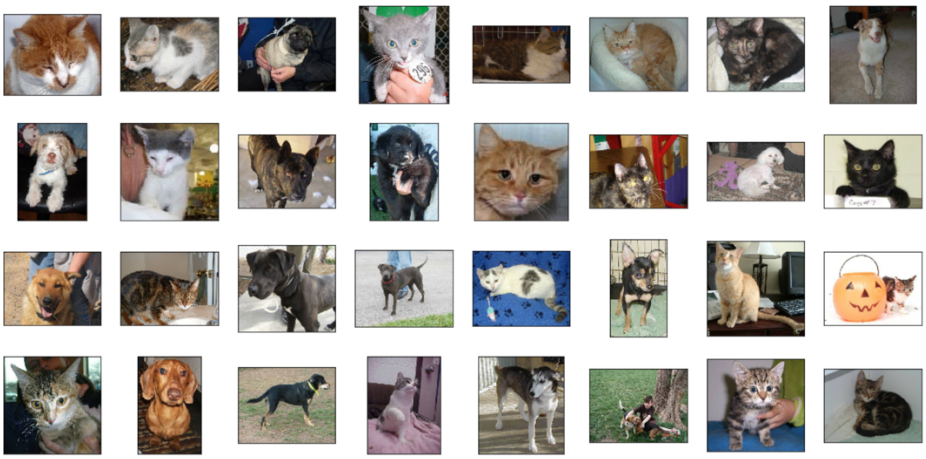
数据研究

为了规范化后面的算法使用，这里需要对下载的数据集进行解压，并满足如下目录结构。或者直接使用 kaggle 的 api 进行下载，也会满足如下目录结构。

```
├── bottleneck_features.ipynb
├── README.md
├── requirements/
│   └── ...
├── sample_submission.csv
├── test.zip
├── train.zip
└── transfer_learning.ipynb
```

数据可视化

为了更好的对数据有直观的了解，将内存中的图片可视化出来。如下：



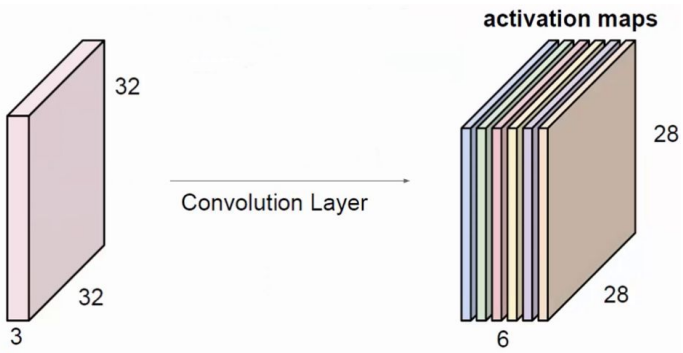
通过上图可以看出，训练所用的数据，尽管有各种复杂的背景，尽管猫和狗的姿势和形态也比较多样性，但整体来看图片都比较清晰，算是比较理想的数据集。

算法与方法

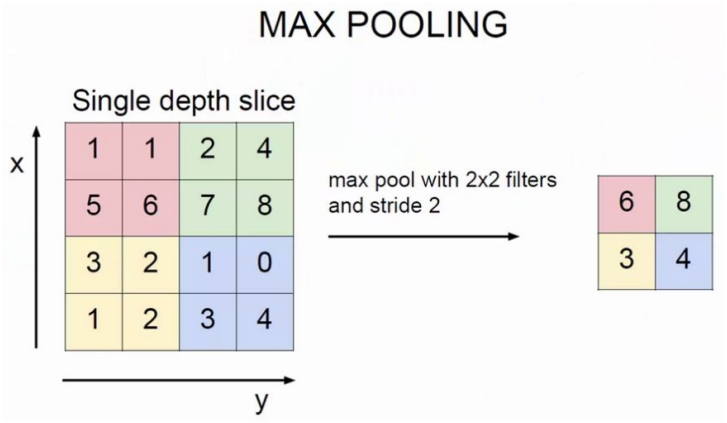
卷积神经网络

卷积神经网络（Convolutional Neural Network，CNN）是一种带有卷积结构的深度神经网络，属于监督学习中的一种。它的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的表现。它包括：卷积层（convolutional layer）和池化层（pooling layer）。

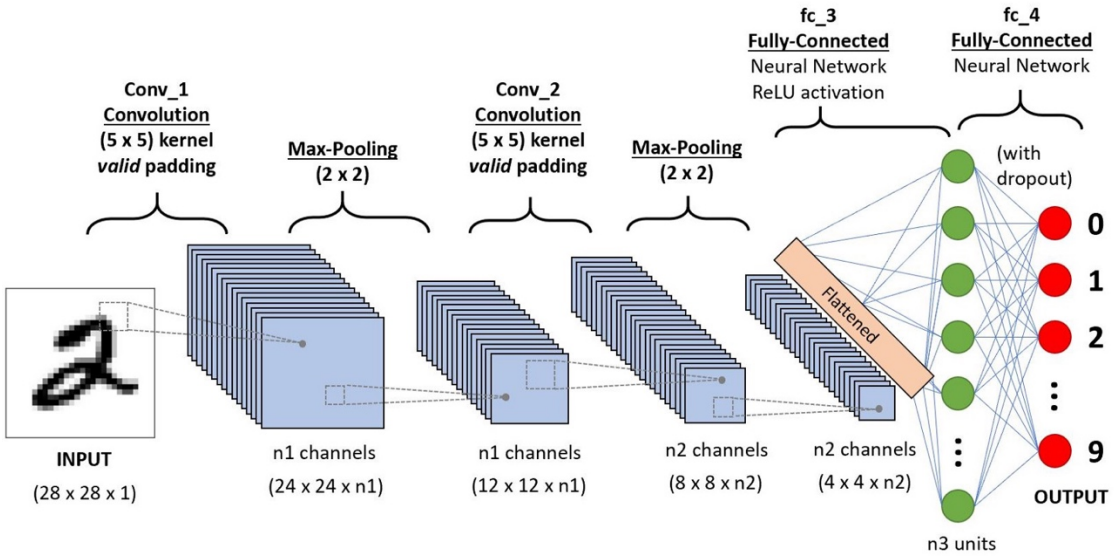
卷积，是在原始的输入上进行特征的提取。特征的提取简言之就上，在原始输入上一个小区域一个小区域进行特征的提取。



池化，就是对特征图进行特征压缩的过程，也称作降采样。选择原来的某个区域的 max 或 mean 代替那个区域，整体就浓缩了。



将卷积层和池化层结合起来，就得到了如下的过程。随着不断的卷积和不断的池化操作，最后得到了一个分辨率很低，但特征很多的图片，然后通过全连接

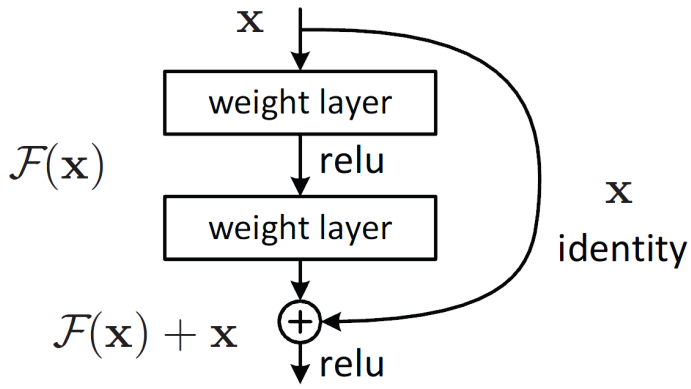


残差网络

ResNet（Residual Neural Network）由微软研究院的 Kaiming He 等四名华人提出，通过使用 ResNet Unit 成功训练出了 152 层的神经网络，并在 ILSVRC2015 比赛中取得冠军，在 top5 上的错误率为 3.75%，同时参数量比 VGGNet 低，效果非常突出。

传统的卷积网络或者全连接网络在信息传递的时候或多或少会存在信息丢失、损耗等问题，同时还会导致梯度消失或者梯度爆炸，导致很深的网络无法训练。ResNet 在一定程度上解决了这个问题，通过直接将输入信息直接输出到下一层，保存了信息的完整性。

ResNet 网络上由若干个 ResidualBlock 和 IdentityBlock 构成的，比较常用的分别是 Resnet50、Resnet101 和 Resnet152。



迁移学习

这里我们为了采用比较简单的 Resnet50 模型来构建我们的神经网络。鉴于整个网络的训练时间比较长，我们使用 ImageNet 的数据集预先训练好的模型，通过迁移学习来训练我们自己的神经网络。

所谓迁移学习，就是运用已有的知识对不同但相关领域问题进行求解的新的一种机器学习方法。在传统的分类学习中，为了保证训练得到的分类模型具有准确性和高可靠性，都必须满足两个基本的条件：

（1） 新的数据集与原始数据集的相似程度；

（2） 新的数据集的大小情况；

使用迁移学习总共有四种情况：

（1） 新的数据集较小，且与原始数据相似度较高；

（2） 新的数据集很小，但与原始数据相似度较低；

（3） 新的数据集很大，且与原始数据相似度较高；

（4） 新的数据集很大，但与原始数据相似度较低；

本项目中使用的 dogs_vs_cats 与 ImageNet 整体上来看，相似度还是比较高的，且 dogs_vs_cats 只提供了 25000 个训练数据（含验证数据）和 12500 个测试数据，样本偏小，比较适合第一种情况。

基准测试

按照 udacity 课程的要求，需要保证训练的模型的准确度在 kaggle 的 Publish Leaderboard 中的排名达到前 10%。由于当前 kaggle 中的总排名人数为 1314，所以要达到的排名需在 1—131 之间。根据排名的分数，测试集的评分需小于 0.06127。

三、方法

数据预处理

从目录结构中可以看出，train 数据集的数据标注上包含在文件名中，因此需要获取数据集的相关信息和标注信息。另外，test 数据集只提供了文件，并没有数据标注，需要将预测的结果提交到 kaggle 上才能获得评分。

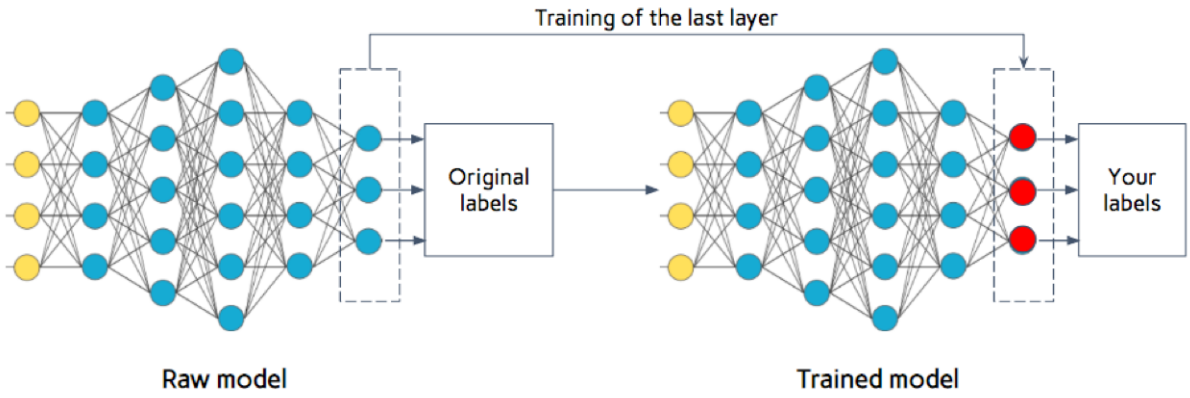
文件	标注（cat=0.0，dog=1.0）
datas/train/cat.6938.jpg	0.0
datas/train/dog.11432.jpg	1.0
datas/train/cat.433.jpg	0.0
datas/train/cat.11305.jpg	0.0

备注：这里需要值得注意的是，kaggle 的 submission 文件中的 id 是按数字排序的（如：0，1，2，3），而加载数据集获取到的文件名通常是按字符串进行排序的（如：0，1，10，100）。

在进行算法建模之前，需要先把用到的图片数据加载到内存中，由于后续的神经网络模型，这里需要将图片分辨率统一为（224，224，3）。为了在训练神经网络模型时对模型进行验证，以避免出现过拟合的现象，需要将 train 数据集进行拆分，分为 train 数据集和 valid 验证集。

算法实施细节

由于我们使用的预训练模型是基于 ImageNet 数据集进行训练的，而项目中的训练集和 ImageNet 的相似度比较高，同时项目中使用的数据集算是相对比较小的，所以需要对全连接层进行重新训练。



首先，构建 Resnet50 模型，使用 ImageNet 的权重初始化网络，并且去掉全连接层。将预先加载的训练集、验证集和测试集数据传入预训练模型，然后将输出的无全连接层的数据特征，以及训练集和验证集的标注信息存储到 bottleneck 文件中。

然后，构建新的全连接层神经网络，将先前提取的训练集和验证集的 bottleneck 特征数据作为输入，进行网络的训练，并将训练的最佳权重保存到 hdf5 文件中。

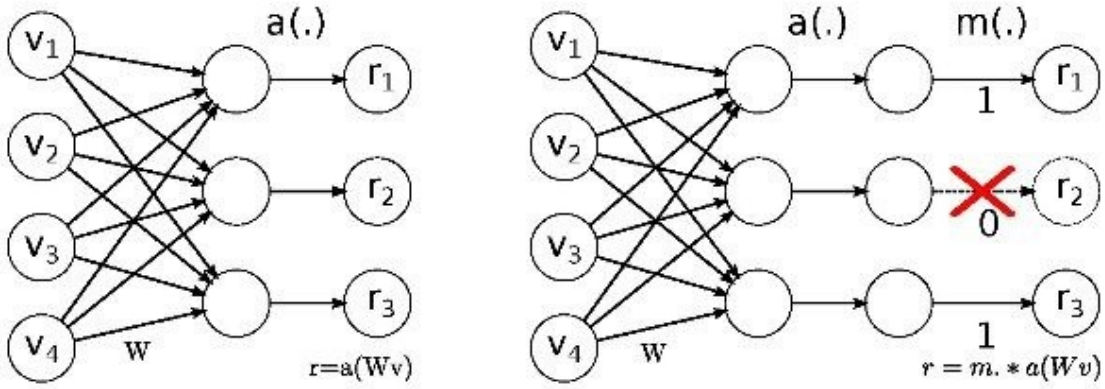
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
global_average_pooling2d_1 ( (None, 2048)              0
_________________________________________________________________
dropout_1 (Dropout)          (None, 2048)              0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 2049
=================================================================
Total params: 2,049.0
Trainable params: 2,049.0
Non-trainable params: 0.0
_________________________________________________________________
```

最后，使用训练好的神经网络，对先前提取的测试集的 bottleneck 特征数据进行预测，并将预测结果保存到 submission.csv 文件中。这里需要注意的是 submission.csv 模版数据中的 id 是和测试集的文件名是匹配的，要保证预测的结果和 id 是匹配的。

模型的改进方法

将模型的 loss 由分类交叉熵（categorical_crossentropy）改为二分交叉熵（binary_crossentropy）。通常，分类交叉熵主要适用于对多分类问题，并使用 softmax 作为输出层的激活函数；交叉熵常用于多标签分类问题，需要使用 sigmoid 作为激活函数。多分类中的每个类别之间是互斥的，所有类别的概率之和为 1，而多标签分类中的每个类型都是独立的，相互之间没有关系。二分类问题属于多标签分类中的一种特殊情况，而多标签分类也是通过 one hot 编码转换为二分类问题多二分类问题进行处理的。

在新的全连接层增加了 Dropout 层。想要提高 CNN 的表达和分类问题，最直接的方法就是使用更深的网络和更多的神经元。但是复杂的网络也意味着更加容易过拟合。Dropout 层会随机在网络的传递过程中丢掉一些神经元节点。这样在多次的训练过程中，会产生不一样的网络计算，可以有效的避免神经网络的过拟合。

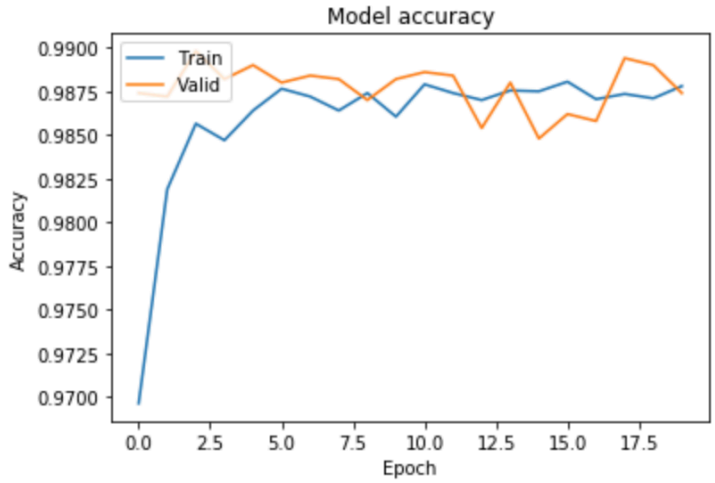


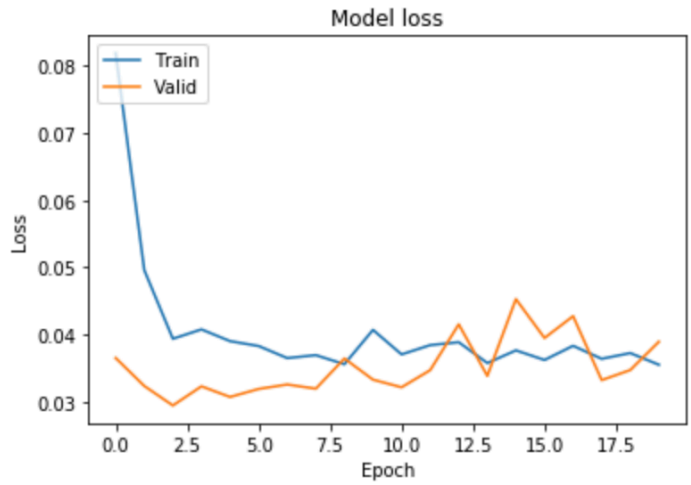
将预测结果 clamp 到[0.005，0.995]之间，主要是由于 kaggle 官方采用的是 logloss，对于小于 0.005 的值，会对最终评分有负面影响。

四、结果

模型评估与验证

通过输出模型训练过程中的 accuracy 和 loss 的曲线，可以看出 train 数据集随着 epoch 的增加，准确度越来越高，接近饱和状态，而 valid 数据集随着 epoch 的增加，一直围绕饱和状态的 train 的准确度波动，说明模型训练前，预训练的模型准确度已经非常高了，并且随着不断训练，并没有出现过拟合的状态；train 数据集的损失度随着 epoch 的增加，出现了大幅的降低，并趋于饱和，而 valid 数据集随着 epoch 的增加，虽有波动，一直稳定在 train 的稳定状态，说明预训练的模型的 loss 非常低了，并且没有随着训练的增加，出现 loss 大幅增加的变化，说明并未出现过拟合。曲线的变化基本上符合预期。





经过多次改进，最终的预测结果评分为 0.05334，小于基准测试标准的评分 0.06127，并通过可视化来查看测试集的预测结果，错误率很低。

Name	Submitted	Wait time	Execution time	Score
submission.csv	2 days ago	0 seconds	0 seconds	0.05334

Complete

五、结论

下图是一组随机的测试集的数据预测，发现图片的角度和背景的多样性，并没有影响预测的准确性。尽管迁移学习的是 ImageNet 的权重模型，但通过重新训练全连接层的神经网络，对于 dogs_vs_cats 的数据集，依旧会有很好的表现。

其实，对于模型的准确度的提升，还是有很多提升空间的，可以尝试使用 Resnet152 来迁移学习，会得到更加准确的预测数据；同时，还可以对数据集中的图片数据做数据增强，以增加数据的多样性，以获得更好的模型。当然，目前也存在一些做法，将多个模型进行融合，以便获得更好的准确度。

参考文献

[1] Sumit Saha, A Comprehensive Guide to Convolutional Neural Networks.

[2] CS231n Convolutional Neural Networks for Visual Recognition.

[3] Jason Yosinski, Jeff Clune, Yoshua Benjio, and Hod Lipson, How transferable are features in deep neural networks.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition.

[5] Jason Brownlee, How to Check-Point Deep Learning Models in Keras.

[6] Aaditya Prakash, One by One [1 x 1] Convolution - counter-intuitively useful.

[7] Francois Chollet, Building powerful image classification models using very little data.