

猫狗大战开题报告

背景

眼睛，作为人类最重要的器官之一，是我们感知周围世界的主要方式；其中大部分的信息都是通过视觉的方式传递给我们的大脑中。而对于机器来说，要想对事件或者物体做出决策判断也不能缺少视觉信息。近年来，随着 imagenet 大量有标签数据集的出现和计算机算力大幅提升，深度学习技术得到迅速的发展。

本项目猫狗大战就是在提供大量有标签的猫狗图像中，分别学习得到猫和狗的不同特征，从而对一张没有标签的未曾见过的图片做出预测，推断图片中是否包含猫或者狗，可以归类为图像中的分类问题。

问题陈述

在本文中，我们的目标是基于 Kaggle dog vs cat 数据集上解决一个二分类问题。具体问题为利用带有标签的猫或者狗的图片，搭建深度神经网络模型并利用这些图片与标签进行训练得到优化的深度神经网络，最终利用优化的深度神经网络对未曾见过的没有标签的图片做出预测，判断出图片中包含猫还是狗从而达到分类的目的。

这里我们利用的数据集是 Kaggle 提供的猫和狗的数据集。最终，我们期望优化过的深度神经网络在测试集上的得分表现可以在 Kaggle 前 10%，具体表现为 Kaggle logloss 比 90% 的人要低。

其中，logloss 计算公式为：

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

输入数据

项目中所用的数据集来自于 Kaggle 竞赛“Dogs vs. Cats Redux: Kernels Edition”，其中训练集一共有 25000 张分别包含猫和狗的图片，图片的命名以“类别+数字”的方式命名，因此我们可以很方便的在读取图片时根据图片的名字得到图片所属的标签（猫或者狗）。另外图片的宽高尺度不一，清晰度不同；我们需要根据输入模型的要求对图片进行预处理，例如不同模型选择预处理方法，例如 Inception V3 的预处理方法是对 RGB 三通道值进行处理，将其缩放至 $[-1, 1]$ 范围内；而 ResNet50 预处理方法是对 BGR 三通道值减 ImageNet 均值。此外，预处理还可以是对训练集中的图片进行旋转、平移或者反转，来实现扩大数据集的目的，以防止过拟合。

解决方案

卷积神经网络作为一个端到端的网络，不再需要像从前一样手动提取图片的特征，因此可以比手动更方便的提取多维度的特征，从不同层次对图片进行描述。随着 imagenet 出现了许多优秀的网络模型，例如 VGG[2]、ResNet[1]、Inception[3]等等，他们对 imagenet 上图片的分类

已经达到了非常高的准确率，证明卷积神经网络用在分类任务是可行的。

本项目利用 keras 快速开发接口，搭建猫狗图像识别的二分类模型，利用迁移学习将 ResNet 在 imagenet 数据集上的训练模型修改为输出为两个分类的模型，fine-tune 卷积神经网络模型，对猫狗数据训练集进行训练。当训练结束后利用优化的卷积神经网络对测试集进行预测，

并将预测结果提交到 kaggle 进行评分。

基准模型与评价指标

我们利用 Logloss 做为损失函数来评估模型的表现，当 Logloss 较小时，模型表现能力强，正确预测猫狗图片的能力强；当 Logloss 较大时，模型表现能力差，正确预测猫狗图片的能力弱。其中，Logloss 计算公式为：

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

其中 n 为测试集中的图片数量；当预测的图片的真实值为狗时 y_i 为 1，真实值为猫时 y_i 为 0； \hat{y}_i 代表的是预测图片为狗的概率。

最终，我们期望优化过的深度神经网络在测试集上的得分表现可以在 Kaggle 前 10%，即 logloss 低于 0.06127。

具体工作流程

1. 对训练集中的数据进行清洗，去掉异常数据，例如背景复杂的图片等。
2. 对清洗过的图片进行预处理，例如对图片进行平移、翻转、旋转一定的角度扩大训练集的数量，另外还需要对图片进行归一化，并且对训练数据进行 shuffle；
3. 将经过步骤 2 处理过的训练集中的数据划分为训练集和验证集，其中验证集占 20% 是随机抽取的图片，剩下的 80% 的数据为训练集。
4. 定义 base model 为 ResNet50，加载在 imagenet 上的预训练值，将最上层的基于 imagenet 1000 个种类的分类器更换为二分类器。
5. 对定义好的模型进行训练得到优化的卷积神经网络：其中可以首先固定 base model 的所有参数不进行训练，只对分类器的参数进行训练；第二步再对网络的参数进行 fine-tuning，得到最适合本项目的分类器[4]。

在所有两步训练中观察 training loss 和 validation loss 的变化关系，防止出现 training loss 不断减少而 validation loss 增大的过拟合现象的发生。如果模型已经停止训练而性能达不到基准阈值，我们可以通过增加 epochs 的数目或者调整 learning rate 来提升模型的性能。

6. 对测试集中的图片进行测试并计算 logloss 的值

参考文献

- [1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015
- [2] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.
- [3] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[C]// Computer Vision and Pattern Recognition. IEEE, 2016:2818-2826.
- [4] Keras Cats Dogs Tutorial. <https://jkjung-avt.github.io/keras-tutorial/>. Published: 2018-04-14