

基于 Android 和微信小程序端的猫狗图像分类

项目背景

深度学习已在机器学习中的各领域广泛应用，其中图像领域使用最多的是卷积神经网络（CNN）。卷积神经网络从 2012 年以来，历经 AlexNet、VGG、Inception、ResNet 等一系列发展，已经能很好的解决图像分类问题。深度学习常用的框架有 MXNET、Caffe、Tensorflow、Torch、Theano 等，目前最流行的是 google 推出的 Tensorflow。

伴随着移动智能终端的兴起，业界正致力于将深度学习部署在移动终端上。微信是世界上广泛应用的一款移动通信 app，在中国几乎覆盖了所有的智能手机。而倡导“用完即走的”的“微信小程序”在 17 年初推出后，一年时间日活已达到 1.7 亿。

问题描述

训练一个可以处理二分类问题（是猫还是狗）的神经网络，并将模型和参数导出，部署在 Android 应用程序和微信小程序上。在 Android 应用程序上，对摄像头传回的图片在移动端进行实时识别，将结果实时的展示出来。在微信小程序上，拍摄或使用相册中的图片上传到后台，在服务端进行识别，将结果返回给小程序展示。

输入数据

输入数据来自 kaggle (<https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition>)。训练集共 25000 张图片，标注了是猫还是狗，两类图片数量各占一半。其中 15000 张图选为训练数据，10000 图选为验证数据。测试集共 12500 张图片，无标注。

解决办法

框架选用 Tensorflow，模型选用 Resnet。由于输入数据的文件名是按猫和狗排列，造成训练模型读入数据时先读入的都是猫的图片，因此需要对数据进行处理。对图片的文件名进行修改，加上随机数字前缀，从而打乱顺序。

训练过程中，对 `learning_rate` 等参数进行调节，提高训练精度。训练结束后，若训练精度较高，再使用验证集进行验证。如果精度仍较高，则对验证集也进行训练，最后用测试集进行测试。把结果上传至 `kaggle`，排名进入 `public leaderboard` 前 10% 则成功，否则继续调参或考虑更优秀的模型。

把训练好的模型和参数导出为 `pb` 文件，基于 `google` 提供的 `Tensorflow Android Camare demo` (<https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/android>) 进行修改，得到可以识别猫狗的 `Android app`。

开发一个简单的微信小程序，将图片传到后台并展示识别结果。在 `centos` 上使用 `python` 的 `django` 框架搭建服务端，对传来的图像进行猫狗识别。

基准模型

使用包含两个卷积层的简单神经网络做为基准模型。

评估指标

验证集精确度，`kaggle public leaderboard` 排名

(<https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/leaderboard>)

设计大纲

1. 下载训练和测试数据，把训练数据分为训练集和验证集
2. 编写重命名批量处理程序，为训练集和验证集中的图片文件名随机添加数字前缀
3. 编写输入处理程序，将图片转为 `Tensorflow` 类型
4. 编写验证程序
5. 编写基准模型
6. 训练和验证基准模型
7. 编写 `Resnet` 模型
8. 训练和验证 `Resnet` 模型，对模型进行调参，或者对输入进行数据增强
9. 提交和修改程序，直至 `leaderboard` 排名进入前 10%
10. 下载并编译通过 `Tensorflow Android Camare demo`

11. 导出 Resnet 模型为 pb 文件，修改 demo 对其进行调用，能成功区别猫狗
12. 编写微信小程序
13. 在 centos 上搭建 https 验证和 django 框架
14. 基于 django 编写服务端程序，成功使用微信小程序区别猫狗