

小白也可以操作的手机TensorFlow教程：Android版和iOS版

TensorFlow通常用于训练海量数据，但新兴的智能手机市场也不可忽略。那些不能等待未来和love Machine Learning的人正在通过制造工具来突破界限，他们的目标是拥有数万亿美元的市场。



本教程使用了一个更稳定的Tensorflow版本，遵循这些步骤实现你的模型，并使用它们对移动设备进行优化。

我们将通过重新训练incepta-v3模型的最终(瓶颈)层来创建一个图像分类器，然后优化智能设备模型。

本教程只包含5-6个步骤：

步骤1：创建带有TensorFlow的模型。

本教程严格专注于智能手机的实现模型，我推荐教程“在CPU上使用自定义图像进行初始化”。所以我们最好在同一页面，并在新目录下用新训练的模型开始操作。

教程地址：<https://towardsdatascience.com/training-inception-with-tensorflow-on-custom-images-using-cpu-8ecd91595f26>

仅供参考：数据模型的训练包含类型的花是郁金香，雏菊，向日葵，蒲公英和玫瑰花。

在这之后，你应该有两个文件，其中一个包含了被选中网络的版本，其最后一层对你的类别进行了重新训练：

```
tf_files/retrained_graph.pb
```

另一个文件包含了文本标签：

```
tf_files/retrained_labels.txt
```

菜单

首页

专栏

问答

登录

```
python -m tensorflow.python.tools.optimize_for_inference \
  --input=tf_files/retrained_graph.pb \
  --output=tf_files/optimized_graph.pb \
  --input_names="input" \
  --output_names="final_result"
```

它将创建一个新的优化模型文件 `tf_files/optimized_graph.pb`。

权衡

为了减少应用程序的预处理,同时减少库的大小, tensorflow只支持在推理过程中常用的操作子集。不支持的操作包括tensorflow / contrib / makefile / tf_op_files.txt。

不支持操作地址:

https://github.com/tensorflow/tensorflow/blob/master/tensorflow/contrib/makefile/tf_op_files.txt

现在,要确保我们刚刚创建的任何图形文件都包括如下可支持操作:

验证

要确保你的新优化图形正在运行,并且optimize_for_inference文件删除了给定输入和输出集不需要的所有节点,但并不改变网络的输出。

利用label_file比较retrained_graph.pb和optimized_graph.pb相同图像的输出。

在retrained_graph.pb上:

```
python -m scripts/label_image \
  --graph=tf_files/retrained_graph.pb\
  --image=tf_files/flower_photos/daisy/3475870145_685a19116d.jpg
```

在optimized_graph.pb上:

```
python -m scripts/label_image \
  --graph=tf_files/optimized_graph.pb \
  --image=tf_files/flower_photos/daisy/3475870145_685a19116d.jpg
```

一个一个地运行这些命令,如果两个输出都是相同的,那就意味着optimized_graph.pb完全创建。

步骤3:量化模型,然后压缩

问题是模型的尺寸仍然很大,而且绝对不适合移动。因为,图形中占据的大部分空间都是由大块浮点数的权值构成的。每一个权值都有一个稍微不同的浮点值,具有非常小的规律性。

但是压缩工作是利用数据的规律性,这就解释了失败的原因。

量化有助于通过对网络的权值进行量化,以按因子减小神经网络的大小。这使得图形中有了更多的repetition,这将对之后的压缩中有很大的帮助。

菜单

首页

专栏

问答

登录

```
python -m scripts/quantize_graph \
  --input=tf_files/optimized_graph.pb \
  --output=tf_files/rounded_graph.pb \
  --output_node_names=final_result \
  --mode=weights_rounded
```

然后压缩模型:

```
gzip -c tf_files/rounded_graph.pb > tf_files/rounded_graph.pb.gz

gzip -l tf_files/rounded_graph.pb.gz
```

这将创建一个rounded_graph.pb文件。

你应该会看到压缩的一些显著改进。

注意:如果在运行quantize_graph时遇到任何错误,请下载该文件并将其粘贴到Tensorflow库的tools/quantization /quantize_graph.py中。

从这里开始,教程分为两个部分:Android版和iOS版。

iOS

步骤4:添加TensorFlow-experimental pod

在pod文件中添加TensorFlow-experimental pod,它安装了一个通用的二进制框架。这是在iOS中运行tensorflow最简单的方法。

步骤5:创建应用程序

创建自己的应用程序或加载你在XCode中创建的应用程序。

在项目根目录下添加一个名为Podfile的文件,其中包含以下内容:

```
target 'YourProjectName'
pod 'TensorFlow-experimental'
```

运行 pod install,并下载安装TensorFlow-experimental pod。

开放YourProjectName.xcworkspace并添加你的代码。

在应用程序的生成设置中,确保将I \$(herited) 添加到其他链接器标志和标题搜索路径部分。

步骤6:运行样本

你将需要Xcode 7.3或稍后运行iOS示例。

在简单、基准和相机方面,有三个示例。你可以复制该代码。

代码地址: <https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/ios#building-the-tensorflow-ios-libraries-from-source>

```
mkdir -p ~/graphs
curl -o ~/graphs/inception5h.zip \
  https://storage.googleapis.com/download.tensorflow.org/models/inception5h.zip \
  && unzip ~/graphs/inception5h.zip -d ~/graphs/inception5h
cp ~/graphs/inception5h/* tensorflow/examples/ios/benchmark/data/
cp ~/graphs/inception5h/* tensorflow/examples/ios/camera/data/
cp ~/graphs/inception5h/* tensorflow/examples/ios/simple/data/
```

更改为一个示例目录, 下载Tensorflow-experimental pod, 并打开Xcode工作区。注意, 安装pod可能需要很长时间, 因为它很大(~450MB)。如果想运行简单的示例, 那么可以运行以下应用程序:

```
cd tensorflow/examples/ios/simple
pod install
open tf_simple_example.xcworkspace #note .xcworkspace, not .xcodeproj
```

在XCode模拟器中运行这个简单的应用程序。你应该会看到一个带有运行模型按钮的单屏幕应用程序。点击它, 你就会看到格Grace Hopper图像。一旦你建立并运行它, 你应该得到一个实时的相机视图, 你可以指向物体获得实时的识别结果。

注意: 我很确定我犯了一些错误, 或者在iOS部分留下了一些东西。如果你有任何疑问, 请浏览下面的官方链接。

网站地址: https://www.tensorflow.org/mobile/ios_build

现在, 到了本文的Android部分。

Android

步骤4: 设置Android Studio和测试运行

有两种方法可以实现Android Studio和Bazel。我会用更多人熟悉的方法。

如果你还没有安装它, 那么就到这里安装它。

安装地址: <https://developer.android.com/studio/index.html>

测试运行

为了检查Android Studio中是否一切正常, 我们要进行一次测试运行。

打开Android Studio和选择 “



菜单

首页 专栏 问答

登录



到tensorflow-for-poets-2/android/tfmobile目录下。

打开Build.gradle文件，同步等级。如果一切正常，单击BUILD>BUILD APK按钮。

现在，应该出现一个带有app.apk文件的文件夹，将其复制到你的手机中并安装它。同时，要记住打开手机的开发者模式。

步骤5: 运行定制的应用程序

默认的应用程序是一个有1000个类别的Imagenet的分类图像应用。

现在，运行定制的应用程序执行以下两个步骤：

将模型文件添加到项目中。

现在，演示应用程序正在查看graph.pb文件。在android/tfmobile/assets中显示的label.txt文件不在你的rounded_graph.pb and retrained_labels.txt上。

现在，用下面的命令替换这些文件，也可以手动执行。

```
cp tf_files/rounded_graph.pb android/tfmobile/assets/graph.pb
cp tf_files/retrained_labels.txt android/tfmobile/assets/labels.txt
```

在ClassifierActivity.java文件中更改“output_name”。

我们的模型的输出节点有一个不同的名称：“final_result”。按照下面的步骤开放 ClassifierActivity.java 和 更新 OUTPUT_NAME变量：

```
private static final String INPUT_NAME = "input";
private static final String OUTPUT_NAME = "final_result";
```

运行一遍，所有的一切就都完成了。

原文发布于微信公众号 - ATYUN订阅号 (atyun_com)
原文发表时间：2018-02-18

菜单

首页 专栏 问答

登录

TensorFlow

Android

iOS

6

分享 举报



ATYUN订阅号
418 篇文章 33 人订阅

订阅专栏

0 条评论

写下你的评论

评论

相关文章

来自专栏 人工智能LeadAI
配置深度学习主机与环境（TensorFlow+1080Ti） | 第一章：硬件选购与主机组装
53 6

来自专栏 人工智能LeadAI
基于Spark /Tensorflow使用CNN处理NLP的尝试
53 6

来自专栏 人工智能LeadAI
TensorFlow从0到1 | 第六篇：解锁梯度下降算法
68 9

来自专栏 pangguoming
Java Annotation 及几个常用开源项目注解原理简析
73 5

来自专栏 人工智能LeadAI
TensorFlow从0到1 | 第十三章： AI驯兽师：神经网络调教综述

菜单

首页 专栏 问答

登录

来自专栏 人工智能LeadAI

如何优雅地用TensorFlow预测时间序列：TFTS库详细教程

79 12

免责声明

社区规范

分享计划

翻译社

联系我们

腾讯开放平台 QQ物联 DNSPod 微信公众平台 企业QQ 腾讯优图 腾讯蓝鲸 腾讯微云 友情链接

中国站



Copyright © 2013-2018 Tencent Cloud. All Rights Reserved. 腾讯云 版权所有 京公网安备 11010802020287 京ICP备11018762号