

TensorFlow集成Android工程的框架



SpikeKing (/u/e2b4dd6d3eb4) [+ 关注](#)

2017.09.21 14:50* 字数 392 阅读 972 评论 2 喜欢 14

(/u/e2b4dd6d3eb4)

欢迎Follow我的GitHub (<https://link.jianshu.com?t=https://github.com/SpikeKing>)
，关注我的简书 (<https://www.jianshu.com/users/e2b4dd6d3eb4/>)

在Android工程中，集成TensorFlow模型。运行TensorFlow的默认Android工程，请参考 (<https://www.jianshu.com/p/dff6ad105c8e>)。

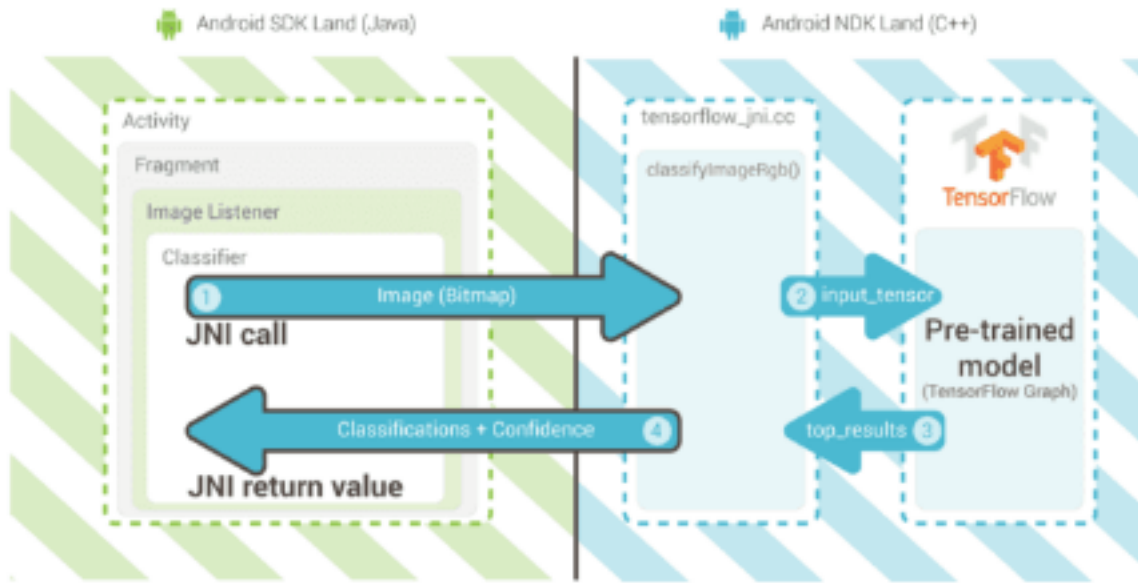
Android源码：<https://github.com/SpikeKing/TFAndroid/tree/master> (<https://link.jianshu.com?t=https://github.com/SpikeKing/TFAndroid/tree/master>)

库及模型的大小

```
libtensorflow_inference.so  10.2 M  
libandroid_tensorflow_inference_java.jar  27 KB  
optimized_tfdroid.pb  291 B
```

如果将so转换为jar库，参考 (<https://www.jianshu.com/p/585be9d8db53>)，则TF的so由10.2M缩小至4.1M。





TF Android

TensorFlow

TF模型源码：

https://github.com/SpikeKing/MachineLearningTutorial/blob/master/tests/android_test.py
(https://link.jianshu.com?t=https://github.com/SpikeKing/MachineLearningTutorial/blob/master/tests/android_test.py)

创建TensorFlow模型，简单的 $y = WX + b$ ，存储图信息 `write_graph`，存储参数信息 `saver.save`。输入数据placeholder是 `x`，输出数据是 `o`。



```
import tensorflow as tf

I = tf.placeholder(tf.float32, shape=[None, 3], name='I') # input
W = tf.Variable(tf.zeros(shape=[3, 2]), dtype=tf.float32, name='W') # weights
b = tf.Variable(tf.zeros(shape=[2]), dtype=tf.float32, name='b') # biases
O = tf.nn.relu(tf.matmul(I, W) + b, name='O') # activation / output

saver = tf.train.Saver()
init_op = tf.global_variables_initializer()

with tf.Session() as sess:
    sess.run(init_op)

    tf.train.write_graph(sess.graph_def, './data/android/', 'tfdroid.pbtxt') # 存储

    # 训练数据, 本例直接赋值
    sess.run(tf.assign(W, [[1, 2], [4, 5], [7, 8]]))
    sess.run(tf.assign(b, [1, 1]))

    # 存储checkpoint文件, 即参数信息
    saver.save(sess, './data/android/tfdroid.ckpt')
```

创建Freeze的图, 将图结构与参数组合在一起, 生成模型, 参考 (<https://link.jianshu.com?t=http://blog.csdn.net/czq7511/article/details/72452985>)。



```
def gnr_freeze_graph(input_graph, input_saver, input_binary, input_checkpoint,
                    output_node_names, output_graph, clear_devices):
    """
    将输入图与参数结合在一起

    :param input_graph: 输入图
    :param input_saver: Saver解析器
    :param input_binary: 输入图的格式, false是文本, true是二进制
    :param input_checkpoint: checkpoint, 检查点文件

    :param output_node_names: 输出节点名称
    :param output_graph: 保存输出文件
    :param clear_devices: 清除训练设备
    :return: NULL
    """
    restore_op_name = "save/restore_all"
    filename_tensor_name = "save/Const:0"

    freeze_graph.freeze_graph(
        input_graph=input_graph, # 输入图
        input_saver=input_saver, # Saver解析器
        input_binary=input_binary, # 输入图的格式, false是文本, true是二进制
        input_checkpoint=input_checkpoint, # checkpoint, 检查点文件
        output_node_names=output_node_names, # 输出节点名称
        restore_op_name=restore_op_name, # 从模型恢复节点的名字
        filename_tensor_name=filename_tensor_name, # tensor名称
        output_graph=output_graph, # 保存输出文件
        clear_devices=clear_devices, # 清除训练设备
        initializer_nodes="") # 初始化节点
```

优化模型，剪切节点，模型只保留输入输出的参数。



```
def gnr_optimize_graph(graph_path, optimized_graph_path):
    """
    优化图
    :param graph_path: 原始图
    :param optimized_graph_path: 优化的图
    :return: NULL
    """
    input_graph_def = tf.GraphDef() # 读取原始图
    with tf.gfile.Open(graph_path, "r") as f:
        data = f.read()
        input_graph_def.ParseFromString(data)

    # 设置输入输出节点, 剪切分支, 大约节省1/4
    output_graph_def = optimize_for_inference_lib.optimize_for_inference(
        input_graph_def,
        ["I"], # an array of the input node(s)
        ["O"], # an array of output nodes
        tf.float32.as_datatype_enum)

    # 存储优化的图
    f = tf.gfile.FastGFile(optimized_graph_path, "w")
    f.write(output_graph_def.SerializeToString())
```

执行函数, 生成模型, frozen_tfdroid.pb 和 optimized_tfdroid.pb。

```
if __name__ == "__main__":
    input_graph_path = MODEL_FOLDER + MODEL_NAME + '.pbtxt' # 输入图
    checkpoint_path = MODEL_FOLDER + MODEL_NAME + '.ckpt' # 输入参数
    output_path = MODEL_FOLDER + 'frozen_' + MODEL_NAME + '.pb' # Freeze模型

    gnr_freeze_graph(input_graph=input_graph_path, input_saver="",
                     input_binary=False, input_checkpoint=checkpoint_path,
                     output_node_names="O", output_graph=output_path, clear_devices=

    optimized_output_graph = MODEL_FOLDER + 'optimized_' + MODEL_NAME + '.pb'

    gnr_optimize_graph(output_path, optimized_output_graph)
```

Android

编译Android的库, 参考 (<https://link.jianshu.com?t=https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/android>), 或者, 直接在Nightly中下载, 参考 (<https://link.jianshu.com?t=https://ci.tensorflow.org/view/Nightly/job/nightly-android/44/artifact/>) , archive.zip, 大约158M。

创建Android工程，添加 app/libs/ 中添加库文件。

```
armeabi-v7a/libtensorflow_inference.so  
libandroid_tensorflow_inference_java.jar
```

在build.gradle中，添加

```
android {  
    sourceSets {  
        main {  
            jniLibs.srcDirs = ['libs']  
        }  
    }  
}
```

在app/src/main/assets中，添加模型 optimized_tfdroid.pb 文件。

在MainActivity中，添加so库。

```
static {  
    System.loadLibrary("tensorflow_inference");  
}
```

模型文件在assets中，TF的核心接口类TensorFlowInferenceInterface。

```
private static final String MODEL_FILE = "file:///android_asset/optimized_tfdroid.pb"  
  
private TensorFlowInferenceInterface mInferenceInterface;
```

初始模型文件

```
mInferenceInterface = new TensorFlowInferenceInterface();  
mInferenceInterface.initializeTensorFlow(getAssets(), MODEL_FILE);
```

模型Feed数据，输入点名称是 INPUT_NODE ，输入结构 INPUT_SIZE ，输入数据inputFloats
。



```
float[] inputFloats = {num1, num2, num3};  
mInferenceInterface.fillNodeFloat(INPUT_NODE, INPUT_SIZE, inputFloats);
```

模型执行文件，输出点名称是 OUTPUT_NODE ，即"O"

```
mInferenceInterface.runInference(new String[]{OUTPUT_NODE});
```

输出数据结构

```
float[] resu = {0, 0};  
mInferenceInterface.readNodeFloat(OUTPUT_NODE, resu);
```

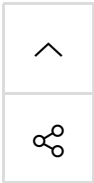
最后，在layout中创建GUI布局。

效果



Demo

TensorFlow集成至春雨医生

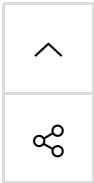


CY-TF

That's all! Enjoy it!

文章都看到这了, 那就捧个"钱场"呗!

赞赏支持





SpikeKing (/u/e2b4dd6d3eb4)

写了 80746 字，被 6875 人关注，获得了 3972 个喜欢
(/u/e2b4dd6d3eb4)

+ 关注

资深数据挖掘工程师资深Android研发工程师春雨医生中国科学院自动化所北京交通大学欢迎Follow我的Git...

喜欢 | 14



更多分享

(http://cwb.assets.jianshu.io
/notes
/images
/17000057



下载简书 App ▶
随时随地发现和创作内容



b.jpg)

(/apps/download?utm_source=nbc)

被以下专题收入，发现更多相似内容



数据挖掘 机器... (/c/df72002ab06e?utm_source=desktop&
utm_medium=notes-included-collection)



程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-
collection)



android (/c/82e1ab06ccab?utm_source=desktop&utm_medium=notes-
included-collection)



android... (/c/9699216a22f7?utm_source=desktop&utm_medium=notes-
included-collection)



机器学习 (/c/1d0f3beef6b1?utm_source=desktop&utm_medium=notes-
included-collection)



Android - 收藏集 (/p/dad51f6c9c4d?utm_campaign=maleskine&utm_c...

用两张图告诉你，为什么你的 App 会卡顿？ - Android - 掘金 Cover 有什么料？从这篇文章中你能获得这些料：知道setContentView()之后发生了什么？... Android 获取 View 宽高的常用正确方式，避免为零 - 掘金相信有很多...



passiontim (/u/e946d18f163c?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

Android - 收藏集 - 掘金 (/p/5ad013eb5364?utm_campaign=maleskine&...

用两张图告诉你，为什么你的 App 会卡顿？ - Android - 掘金Cover 有什么料？从这篇文章中你能获得这些料：知道setContentView()之后发生了什么？... Android 获取 View 宽高的常用正确方式，避免为零 - 掘金相信有很多朋友...

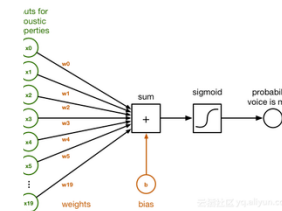


掘金官方 (/u/5fc9b6410f4f?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/b370ac791613?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

IOS平台TensorFlow实践 (/p/b370ac791613?utm_ca...

1 天前 作者简介：MATTHIJS HOLLEMANS 荷兰人，独立开发者，专注于底层编码，GPU优化和算法研究。目前研究方向为IOS上的深度学习及其在APP上的应用。推特地址：<https://twitter.com/mhollemans> 邮件地址：mailto:ma...

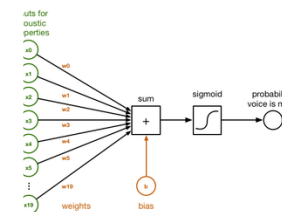


阿里云栖社区 (/u/12532d36e4da?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/abc4e084bdbd?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

深度学习指南：在iOS平台上使用TensorFlow (/p/abc...

在利用深度学习网络进行预测性分析之前，我们首先需要对其加以训练。目前市面上存在着大量能够用于神经网络训练的工具，但TensorFlow无疑是其中极为重要的首选方案之一。这就是Tensor的全部含义。在卷积神经网络等深度学习方案当中，大家会需要与四维张量打交道。但本示例中提...



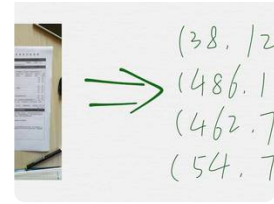
BURIBURI_ZAEMON (/u/00d1ed2b53ae?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)



(/p/5ae69f175379?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

[转载]基于 TensorFlow 和 OpenCV 实现文档检测功能...

文章来源：github 作者：fengjian 前言 本文不是神经网络或机器学习的入门教学，而是通过一个真实的产品案例，展示了在手机客户端上运行一个神经网络的关键技术点 在卷积神经网络适用的领域里，已经出现了一些很经典的图像分类网络，比如 VGG16/VGG19，Incep...



dopami (/u/66640ecf0a46?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/15487dd30743?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

人间真的失格了吗 (/p/15487dd30743?utm_campaign=...

读过太宰治的《人间失格》，读完之后，我的第一感觉是，小说主人公叶藏太过夸张，甚至有些鄙夷主人公的荒谬和颓废。可是后来，我渐渐同情起叶藏。他在身为人类的同时却又极度畏惧人类，同样生而为人我很抱歉。叶藏眼里的人类都是虚伪的，包括他自己，在深刻认识到人类的虚伪的同时，他却又...



逆着星光 (/u/d9d677315776?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

曾国藩的读书观 (/p/1fdf8742fd2e?utm_campaign=maleskine&utm_cont...

曾国藩把学习作为提高个人修养的基本要求，将“敬德修业”当做自己进步的基础。在他看来，要想取得事业的成功，就必须积极进取，没有百折不挠、坚韧不拔的精神难以成就大事。一方面，曾国藩继承了先秦孔子、孟子、墨子的观点，认为读书是为了提高品德情操，增长知识才干，使自己成为“贤人”...



friend6522 (/u/f0a21ac7cb22?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

Dozer 使用小结 (/p/bf8f0e8aee23?utm_campaign=maleskine&utm_cont...

这篇文章是本人在阅读Dozer官方文档（5.5.1版本，官网已经一年多没更新了）的过程中，整理下来我认为比较基础的应用场景。本文中提到的例子应该能覆盖JavaBean映射的大部分场景，希望对你有帮助。概述 Dozer是什么？Dozer是一个JavaBean映射工具库...



静默虚空 (/u/e336bc023b9d?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

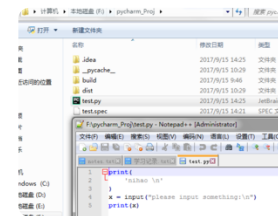


utm_medium=seo_notes&utm_source=recommendation)

(/p/a53b0083f7bf?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

python代码执行过程，pyinstaller使用 (/p/a53b0083f...

原帖：http://www.cnblogs.com/kym/archive/2012/05/14/2498728.html 上面的链接主要介绍了python程序的执行过程。python作为脚本语言，我就想如何把python程序更简单的融合到项目中，以下拿windows平台...



iFavorite (/u/5783673564e1?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

那一曲恋歌，吟唱多年 (/p/ad6e0d7d4114?utm_campaign=maleskine&ut...

那一曲恋歌，吟唱多年 您听过鲤鱼跳跃的声音吗？这是清晨的易湖湖畔，请你侧耳倾听。多少人一生没有听过这种声音，住在湖畔，这个声音会日日回荡在您的睡梦里。天天鲤鱼跃龙门，这就是我魂牵梦绕的她。您见过梅花鹿眸子的美丽吗？这是黄昏的梦蝶钢塑一角，请您静心观赏，多少人少有机会与生灵...

躲开让我来 (/u/3b43be7d88e4?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

