

实验十八 Android 下使用 Tensorflow 模型

一、 实验目的

- (1) 了解 Tensorflow 模型文件的生成和保存。
- (2) 学习 Android 下使用 C++库的基本步骤。
- (3) 学习 Android 下使用 Tensorflow 所生成的模型。

二、 实验原理

1、Python 开发环境 Anaconda

Anaconda 指的是一个开源的 Python 发行版本，其包含了 conda、Python 等 180 多个科学包及其依赖项。因为包含了大量的科学包，Anaconda 的下载文件比较大，如果只需要某些包，或者需要节省带宽或存储空间，也可以使用 Miniconda。

Anaconda 包括 Conda、Python 以及一大堆安装好的工具包，比如：numpy、pandas 等。其中 Conda 是一个开源的包、环境管理器，可以用于在同一个机器上安装不同版本的软件包及其依赖，并能够在不同的环境之间切换。Miniconda 包括 Conda、Python。

2、Python 下使用 Tensorflow

MNIST 是一个简单的计算机视觉数据集，它包含手写数字的图像集。可以从这个简单的数据集开始练习 Tensorflow 的使用。Tensorflow 编程的一个基本步骤大致为：1、定义数据；2、定义计算图与变量；3、定义会话；4、进行计算。建议阅读[1]学习在 MNIST 上进行神经网络模型搭建的简单 Tensorflow 例程。

3、Android JNI

JNI 的全称就是 Java Native Interface，顾名思义，就是 Java 和 C/C++相互通信的接口。可以通过 JNI 调用系统提供的 API。操作系统，无论是 Linux，Windows 还是 Mac OS，或者一些汇编语言写的底层硬件驱动都是 C/C++ 写的。Java 和 C/C++不同，它不会直接编译成平台机器码，而是编译成虚拟机可以运行的 Java 字节码的.class 文件，通过 JIT 技术即时编译成本地机器码，所以有效率就比不上 C/C++代码，JNI 技术就解决了这一痛点，JNI 可以说是 C 语言和 Java 语言交流的适配器、中间件[2]。

Android 提供了 3 个实用的函数用来加载 JNI 库，分别是 System.loadLibrary(libname)，Runtime.getRuntime().loadLibrary(libname)，以及 Runtime.getRuntime().load(libFilePath)[3]。用 System.loadLibrary(libname) 和 Runtime.getRuntime().loadLibrary(libname)这两个函数加载 so 库，不需要指定 so

库的路径，Android 会默认从系统的共享库目录里面去查找，Android 的共享库目录就是 `vendor/lib` 和 `system/lib`，如果在共享库路径里面找到指定名字的 so 库，就会立即加载这个 so 库，所以我们给 so 库起名的时候要尽量避免和 Android 共享库里面的 so 库同名。如果在共享库目录里面查找不到，就会在 APP 的安装目录里面查找 APP 的私有 so 库，如果查找到，会立即加载这个 so 库。

Android Studio 通过 `CMakeLists.txt` 文件配置需要生成的 so 库。`CMakeLists.txt` 文件中使用 `add_library` 函数用来配置要生成的 so 库的基本信息，比如库的名字，要生成的 so 库是静态的还是共享的，so 库的 C/C++ 源文件列表。使用 `find_library` 函数来从 NDK 目录下面查找特定的 so 库。使用 `target_link_libraries` 函数用来把要生成的 so 库和依赖的其它 so 库进行链接，生成我们需要的 so 库文件。在 app 目录下新建 `CMakeLists.txt` 文件，其实要是在 Android Studio 新建项目时勾选了使用 c++，那么这个文件和一些其他配置已经有了，稍微修改就行。

4、在 Android 上集成 Tensorflow

谷歌开源库 Tensorflow 可被用在安卓系统中实现机器学习。Tensorflow 的核心是用 C++ 编写的；为了在安卓上搭建 Tensorflow，我们需要用 JNI（Java 本地接口）来调用 C++ 函数，比如说 `loadModel`，`getPredictions`，等等。

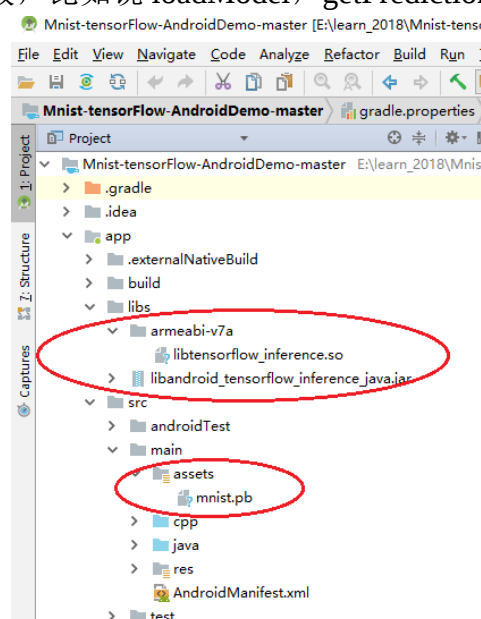


图 1 Android 上运行 Tensorflow 需要的库及模型文件示意

准备好 Tensorflow 支持库和已经在 Tensorflow 下生成的模型文件，如图 1 所示。按照如下步骤在将有关库文件放入 Android 项目[4, 5]。

- (1) 新建一个 Android Project。
- (2) 把刚才的 pb 文件存放到 assets 文件夹下。
- (3) 将 `libandroid_tensorflow_inference_java.jar` 存放到 `/app/libs` 目录下，并且右键“add as Library”。
- (4) 在 `/app/libs` 下新建 `armeabi` 文件夹，并将 `libtensorflow_inference.so` 放进去。

配置 app:gradle 以及 gradle.properties。

在 android 节点下添加 sourceSets，用于制定 jniLibs 的路径

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

在 defaultConfig 节点下添加

```
defaultConfig {
    ndk {
        abiFilters "armeabi"
    }
}
```

在 gradle.properties 中添加下面一行

```
android.useDeprecatedNdk=true
```

5、后续 Android 下对 Tensorflow 的支持

根据 Google 的官方说明[6]，手机上的 Tensorflow 支持从 2019 年开始从 TensorFlow Mobile 向 TensorFlow Lite 迁移。在 Android 下使用 Tensorflow 模型的方法是否可以简化待测试。

Adding TensorFlow to your apps using Android Studio

To add TensorFlow to your own apps on Android, the simplest way is to add the following lines to your Gradle build file:

```
allprojects {
    repositories {
        jcenter()
    }
}

dependencies {
    compile 'org.tensorflow:tensorflow-android:+'
}
```

This automatically downloads the latest stable version of TensorFlow as an AAR and installs it in your project.

图 2 Google 关于 Tensorflow 的声明[6]

三、 示例说明

本实验项目所提供的示例代码参考[5]进行局部修改和增删，主要文件的功能如表 1 所示。示例代码详细说明参见文件中注释信息和原始博文[5]。

表 1 示例代码功能说明

文件名称	功能说明
Mnist-generate-tensorFlow-model\mnist_test.py	基于 MNIST 的 Tensorflow 简单例子
Mnist-generate-tensorFlow-model\mnist-prediction.py	基于 MNIST 训练一个模型
Mnist-generate-tensorFlow-model\mnist-train.py	利用训练好的模型进行手写体识别
Mnist-tensorFlow-AndroidDemo-master 目录	集成了 Tensorflow 支持库的 Android 示例项目

四、 实验要求

1、Python+Tensorflow 环境配置

- (1) 安装 anaconda, 设置好环境, 详细步骤可参考[7]。
- (2) 安装 Tensorflow, 详细步骤可参考[8]。

2、Tensorflow 例程运行与模型生成

- (3) 阅读并运行 Mnist-generate-tensorFlow-model 子目录中的 mnist_test.py。该文件是实现 FC 卷积神经网络识别 mnist 手写体的简单例子, 参考[1]。记录运行结果中识别的正确率。
- (4) 阅读并运行 Mnist-generate-tensorFlow-model 子目录中的另外两个 Python 文件。其中 mnist-prediction.py 利用.py 利用 TensorFlow 生成 MNIST 手写数字识别的模型文件(缺省存放为 model\mnist.pb); mnist-prediction.py 利用已经生成的模型文件对测试图片进行数字识别。

3、将 Tensorflow 下生成的模型集成到 Android 项目中

- (5) 将上一步骤中生成的模型文件拷贝到 Android 项目 Mnist-tensorFlow-AndroidDemo-master 对应的子目录中(Mnist-tensorFlow-AndroidDemo-master\app\src\main\assets\mnist.pb); 编译 Mnist-tensorFlow-AndroidDemo-master 项目。
- (6) 将上一步骤中生成的 APK 文件在自己的 Android 手机上进行测试, 类似图 3 的界面效果。



图 3 APK 运行界面示意

- (7) 阅读 Android 项目 Mnist-tensorFlow-AndroidDemo-master 中的文件:
 - a) Mnist-tensorFlow-AndroidDemo-master\app\build.gradle, 找出其中关于 C++库支持的配置信息。
 - b) Mnist-tensorFlow-AndroidDemo-master\app\ CMakeLists.txt, 找出其中关于 C++库支持的配置信息。

五、 思考题

- (1) 做扩展文献调研，分析 TensorFlow 下生成的模型文件包含哪些信息？
- (2) Android 下 JNI 的作用是什么？
- (3) Armeabi-v7a 指的是什么？

参考文献

- [1] CSDN. python 深度学习库 tensorflow——实现 FC 卷积神经网络识别 mnist 手写体 [Online]. Available: <https://blog.csdn.net/luanpeng825485697/article/details/79100008>
- [2] 简书. Android NDK 开发：JNI 基础篇 [Online]. Available: <https://www.jianshu.com/p/ac00d59993aa>
- [3] CSDN. Android JNI 基础篇 [Online]. Available: <https://blog.csdn.net/kgdwbb/article/details/72810251>
- [4] CSDN. 将 tensorflow 训练好的模型移植到 android 上 [Online]. Available: <https://blog.csdn.net/cxq234843654/article/details/71171293>
- [5] CSDN. 将 tensorflow 训练好的模型移植到 Android (MNIST 手写数字识别) [Online]. Available: <https://blog.csdn.net/guyuealian/article/details/79672257>
- [6] Google. Building TensorFlow on Android [Online]. Available: https://www.tensorflow.org/lite/tfmobile/android_build
- [7] JobBole. Anaconda 使用总结 [Online]. Available: <http://python.jobbole.com/86236/>
- [8] CSDN. 手把手教你如何安装 Tensorflow (Windows 和 Linux 两种版本) [Online]. Available: https://blog.csdn.net/cs_hnu_scw/article/details/79695347