

AI学习笔记--GPU 与 NNAPI

• 环境搭建

TensorFlow 支持 GPU、CPU、NNAPI 等方式运行，首先安装 TensorFlow 的环境。

```
Git clone https://github.com/tensorflow/tensorflow.git
```

clone 好源代码后，在配置相关的 tensor-flow 环境，需要 install bazel 工具。然后在配置相关的 build 环境：

```
→ tensorflow git:(master) x ./configure
```

可以一路点击 n，也可以依据自己电脑的配置进行环境配置，设置好 sdk 路径，build 的版本号等就可以搭建好 build 的环境了。

```
→ examples git:(master) x bazel build --config opt \
//tensorflow/java:tensorflow \
//tensorflow/java:libtensorflow_jni
```

Install java 版本的 TensorFlow 环境，可以按照下面地址步骤：

```
https://github.com/tensorflow/tensorflow/blob/b9fce5027160a606b87b2bf6cce5f9839058f07e/tensorflow/java/README.md
```

之后，释放到 Maven 库当中：

```
bazel build -c opt //tensorflow/java:pom
mvn install:install-file \
-Dfile=../../bazel-bin/tensorflow/java/libtensorflow.jar \
-DpomFile=../../bazel-bin/tensorflow/java/pom.xml
```

如此步骤，基本上可以完成整个 TensorFlow-Lite 的环境搭建。

• NNAPI 与 GPU API

这里我们打开一个 project，地址在：

/Users/genesis/Workplace/AI_Workplace/tensorflow/tensorflow-lite/java/demo

该例子提供了三种方式调用接口方法实现的时间损耗和执行效率对比。导入工程后，可能需要删除 miniSDK 的配置项目。并且，注意主代码中的相关接口代码（在 ImageClassifier）：

```
/** holds a gpu delegate */
GpuDelegate gpuDelegate = null;
/** holds an nnapi delegate */
NnApiDelegate nnapiDelegate = null;
```

我们看到 git 提交记录可以看到，最新的 code 增加了 GPU 的 Delegate 方法，除了这两个，还有 flex、xnnpack 等实现方式代理。

之后同步代码，build 就可以了，也可以使用 bazel 对项目进行 build。

```
bazel build -c opt  
//tensorflow/lite/java/demo/app/src/main:TfLiteCameraDemo
```

其后，会在bin 目录下产生一个 apk，整个 demo 使用 bazel 进行代码连接。也可以看整个 lite 的代码结构。