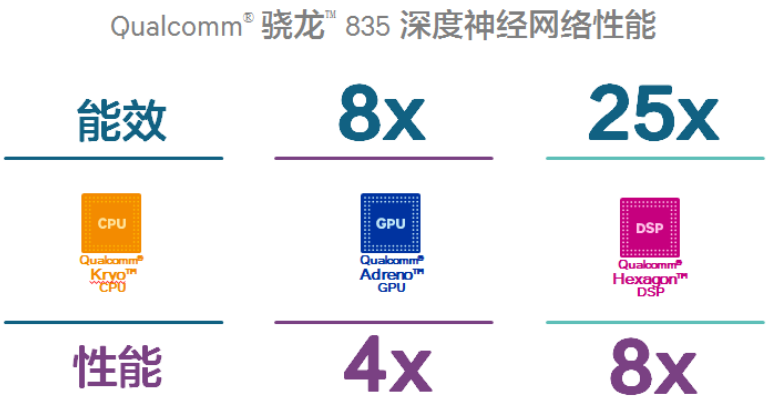
安装使用 Qualcomm Snapdrago Neural Processing Engine (NPE) SDK

NPE SDK 能够帮助开发者做什么事情？

Qualcomm 骁龙神经处理引擎（Neural Processing Engine, NPE）SDK 能够帮助有意创建人工智能（AI）解决方案的开发者，在骁龙移动平台上（不管是 CPU、GPU 还是 DSP）运行通过 Caffe/Caffe2 或 Tensorflow 训练一个或多个神经网络模型，且无需连接到云端，实现边缘计算。

能帮助开发人员在骁龙设备上运行受过训练的神经网络并优化其性能，节约更多时间与精力。NPE SDK 提供了模型转换和执行工具，以及针对核的 API，利用功率和性能配置文件匹配所需的用户体验，优化和节约开发人员的时间和精力。

NPE SDK 支持卷积神经网络、长短期记忆网络（LSTM）和定制层。处理了在骁龙移动平台上运行神经网络所需的大量繁重工作，为开发人员留出更多的时间和资源来专注于 AI 的创造创新应用体验工作方面。



SDK 主要特性有哪些？

- Android 和 Linux 运行环境，供执行神经网络模型
- 支持利用 Qualcomm Hexagon DSP、Qualcomm Adreno GPU 和 Qualcomm Kryo、 CPU（NPE SDK 支持 Qualcomm Snapdragon 820, 835, 625, 626, 650, 652, 653, 660, 630, 636, and 450） 设备

必须有 libOpenCL.so，以支持 Qualcomm Adreno GPU），为应用提供加速

- 支持 Caffe、Caffe2 和 TensorFlow 模型
- 提供控制运行时加载、执行和调度的多个 API
- 用于模型转换的桌面工具
- 用于识别性能瓶颈的性能基准测试
- 示例代码和教程
- HTML 文档

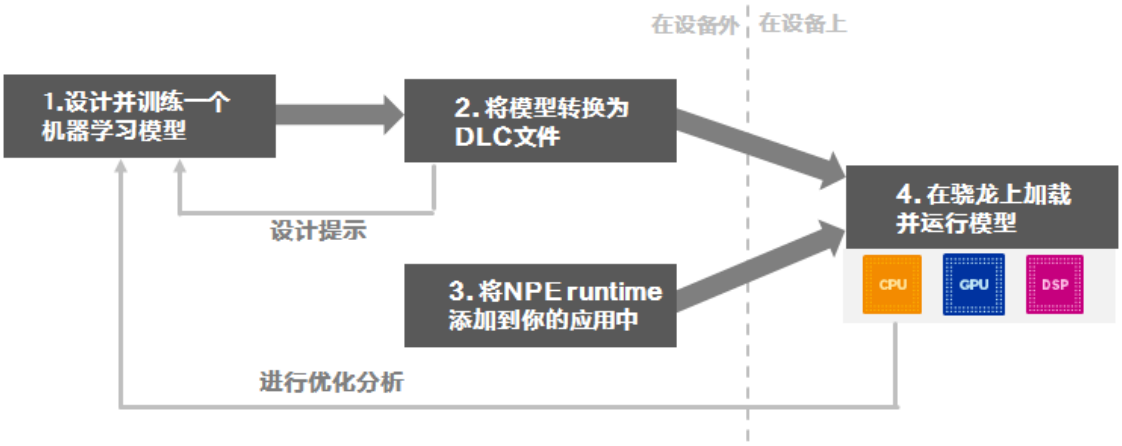
NPE SDK 适合哪些开发者？

使用骁龙 NPE SDK 开发 AI 需要满足以下几个前提，然后才可以开始创建解决方案。

- 你在一个或多个垂直领域需要运行卷积/LSTM 模型，包括移动、汽车、IoT、AI、AR、无人机和机器人等
- 你了解如何设计和训练模型，或者已经有一个预训练的模型文件
- 你选择的框架是 Caffe/Caffe2 或 TensorFlow
- 你可以使用 Android 编写 JAVA 应用，或者基于 Android 或 Linux 系统编写原生应用
- 你有 Ubuntu 14.04 开发环境
- 你有可用于测试应用程序的设备

NPE SDK 使用开发流程

为了让AI开发者更轻松，骁龙NPE SDK没有另行定义网络层库；发布时就支持Caffe/Caffe2和 TensorFlow，开发人员可以选择使用自己熟悉的框架设计和训练网络。开发工作流程如下：



完成模型的设计和训练后，模型文件需要转换成“.dlc”（Deep Learning Container）文件，供骁龙 NPE 运行时使用。转换工具将输出转换信息，包括有关不受支持或非加速层的信息，开发者可以使用这些信息调整初始模型的设计。

搭建 NPE SDK 工作环境

- **系统环境搭建**

建议在专门机器上执行以下操作，以便更好地了解 SDK 依赖项：

1. 安装 Ubuntu 14.04（官网推荐使用）
   如果使用虚拟机安装，可以使用 VirtualBox 工具。虚拟机磁盘空间需要分配大些，建议分配 30G，后续 Android Studio 需要比较大的磁盘空间。
2. 安装最新版 Android Studio，地址：
   https://developer.android.google.cn/studio/index.html
   通过 Android Studio 或独立安装最新版 Android SDK。
3. 安装最新版 Android NDK
   通过 Android Studio SDK Manager 或独立安装。
4. 安装 Caffe，github: https://github.com/BVLC/caffe
   安装说明 ：http://caffe.berkeleyvision.org/installation.html

```
# this will build Caffe (and the pycaffe bindings) from source - see the official instructions for more information
• sudo apt-get install cmake git libatlas-base-dev libboost-all-dev libgflags-dev libgoogle-glog-dev libhdf5-serial-dev libleveldb-dev
```

```
liblmdb-dev libopencv-dev libprotobuf-dev libsnappy-dev protobuf-compiler python-dev python-numpy
•git clone https://github.com/BVLC/caffe.git ~/caffe; cd ~/caffe; git reset --hard d8f79537
• mkdir build; cd build; cmake ..; make all -j4; make install
```

5. 安装 TensorFlow（推荐版本 1.0，github: https://github.com/tensorflow/tensorflow）（可选）
   安装说明： https://www.tensorflow.org/install/

```
# this will download and install TensorFlow in a virtual environment - see the official instructions for more information
• sudo apt-get install python-pip python-dev python-virtualenv
•mkdir ~/tensorflow; virtualenv -- system-site-packages ~/tensorflow; source ~/tensorflow/bin/activate
•pip install --upgrade https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-1.0.0-cp27-none- linux_x86_64.whl
```

- **安装 NPE SDK**

本步骤允许 NPE SDK 通过 python API 与 Caffe 和 Tensorflow 框架进行通信。在 Ubuntu 14.04 上安装 SDK，请执行以下操作：

1. 下载最新的骁龙 NPE SDK。
   地址：https://developer.qualcomm.com/software/snapdragon-neural-processing-engine
   将.zip 文件解压至适当位置（假定在~/snpe-sdk 文件夹中）。
2. 安装缺少的系统包：

```
# install a few more SDK dependencies, then perform a comprehensive check
• sudo apt-get install python-dev python-matplotlib python-numpy python-protobuf python-scipy python-skimage python- sphinx wget zip
• source ~/snpe- sdk/bin/dependencies.sh # verifies that all dependencies are installed
```

```
• source ~/snpe- sdk/bin/check_python_depends.sh # verifies that the python dependencies are installed
```

3. 在当前控制台窗口初始化 Snapdragon NPE SDK 环境
   以后，每个新控制台需重复此操作：

```
# initialize the environment on the current console
• cd ~/snpe-sdk/
• export ANDROID_NDK_ROOT=~/Android/Sdk/ndk-bundle # default location for Android Studio, replace with yours
• source ./bin/envsetup.sh -c ~/caffe
• source ./bin/envsetup.sh -t ~/tensorflow # optional for this guide
```

初始化过程将设置或更新$SNPE_ROOT,?$PATH, $LD_LIBRARY_PATH, $PYTHONPATH, $CAFFE_HOME, $TENSORFLOW_HOME，此外，还在本地复制 Android NDK libgnustl_shared.so 库，更新 Android AAR 存档文件。

下载 ML Models 并转换为.DLC

NPE SDK 没有绑定公开的模型文件，但包含一些脚本，可用于下载一些主流模型，并将其转换为 Deep Learning Container（“DLC”）格式。脚本位于/models 文件夹中，文件夹中还包含 DLC 模型。

1. 下载并转换经预先训练的 Alexnet 示例（Caffe 格式）：

```
•cd $SNPE_ROOT
•python ./models/alexnet/scripts/setup_alexnet.py -a ./temp-assets-cache -d
```

提示：查看执行 DLC 转换的 setup_alexnet.py 脚 本。您可能需要针对 Caffe 模型转换执行相同的操作。

2. 可选：下载并转换经预先训练的“inception_v3”示例（Tensorflow 格式）：

```
•cd $SNPE_ROOT
```

```
•python ./models/inception_v3/scripts/setup_inceptionv3.py -a ./temp-assets-cache - d
```

提示：查看 setup_inceptionv3.py 脚本，此脚 本还对模型进行了量化，大小缩减了 75%（91MB→23MB）。

构建示例 Android APP

示例 Android APP 的源代码演示了如何正确使用 SDK。可以从 ClassifyImageTask.java 开始。示例 Android APP 结合了 Snapdragon NPE 运行环境（/android/snpe-release.aar Android 库提供）和 上述 Caffe Alexnet 示例生成的 DLC 模型。

1. 复制运行环境和模型，为构建 APP 作好准备

```
•cd $SNPE_ROOT/examples/android/image-classifiers
•cp ../../../android/snpe- release.aar ./app/libs # copies the NPE runtime library
•bash ./setup_models.sh # packages the Alexnet example (DLC, labels, imputs) as an Android resource file
```

可选方法 1：从 Android studio 构建 Android APK：

1.启动 Android Studio。

2.打开~/snpe-sdk/examples/android/image- classifiers 文件夹中的项目。

3.如有的话，接受 Android Studio 建议，升级 构建系统组件。

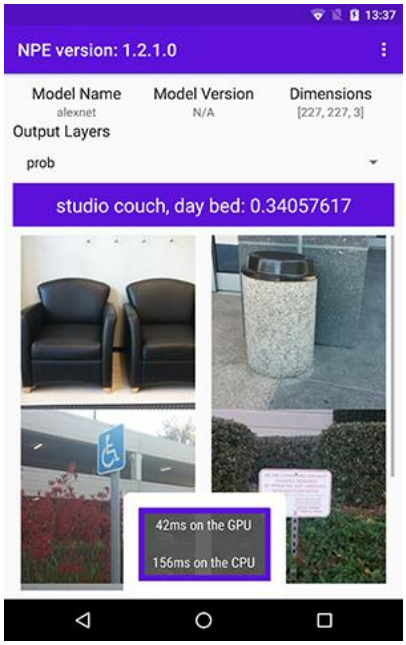
4.按下“运行应用”按钮，构建并运行 APK。

可选方法 2：从命令行构建 Android APK：

```
•sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1
•libbz2-1.0:i386 # Android SDK build dependencies on ubuntu
•./gradlew assembleDebug # build the APK
```

上述命令需要将 ANDROID_HOME 和 JAVA_HOME 设置为系统中的 Android SDK 和 JRE/JDK 所在位置。

执行到这里完成后，示例 APP 已经 build 完成，安装后如下所示：



使用 Snapdragon NPE SDK 制作了第一款示例应用。那么现在，可以开始创建属于自己的 AI 解决方案了！ SDK 随附文档中还有 API 文档、教程和架构详细资料。可以在浏览器中打开/doc/html/index.html 开始学习。

总结

NPE SDK 目前做的还不是非常完美，有些需要定制化的神经网络层可能在原生 NPE 中没有提供。但还好 SDK 提供了用户定义层（UDL）功能，通过回调函数可以自定义算子，并通过重编译 C++代码将自定义文件编译到可执行文件中。如果开发就是使用的 C++，那比较容易实现用户定义层，但如果是运行在 Android 上，开发者需要将上层 java 代码通过 JNI 方式来调用 NPE 原生的 C++编译好的.so 文件，因为用户定义层的代码是不可能预先编译到 NPE 原生.so 文件中的，必须重新开发 NPE 的 JNI。