

X86仿真示例运行说明

0 前言

本文主要讲解如何基于PC端模拟器仿真推理，下面以最新发布的XJ3的v1.12.7版本包里00-quickstart示例

(`ddk/samples/ai_toolchain/horizon_runtime_sample/code/00_quick_start`) 作一个简要说明。主要包括示例包结构说明、环境准备、编译、运行、结果展示五个部分。

1 示例包结构说明

x86仿真示例是在 `horizon_runtime_sample` 目录下，该目录的目录结构如下所示：

```
Python
1 |—— code # 示例源码
2 |   |—— 00_quick_start # 快速入门示例,用mobilenetv1读取单张图片推理的示例
3 |   |—— 01_api_tutorial # dnn API使用示例代码
4 |   |—— 02_advanced_samples # 特殊功能示例
5 |   |—— 03_misc # 杂项示例
6 |   |—— build_xj3.sh # xj3 ARM端编译脚本
7 |   |—— build_x86.sh # PC端编译脚本
8 |   |—— CMakeLists.txt
9 |   |—— CMakeLists_x86.txt
10 |   |—— deps # 编译依赖库
11 |—— xj3
12 |   |—— data # 预置数据文件
13 |   |—— model # 模型文件
14 |   |—— script # ARM端示例运行脚本
15 |   |—— script_x86 # PC端示例运行脚本
16 |—— README.md
```

当前x86仿真示例只提供了基于00_quick_start示例的一个使用方式，主要包括两部分：

`code/00_quick_start` 和 `script_x86`。

2 环境准备

由于是PC端推理，不需要开发板环境准备工作，仅包含**开发机部署**部分。对于开发机的环境部署，地平线支持使用Docker和手工安装两种方式，您只需要选择其中一种即可。具体两种方式的使用可参考《*Horizon AI Toolchain User Guide*》手册中的《环境部署》这一小节。



地平线强烈建议您使用Docker方式，避免在安装过程中出现问题。

3 编译

编译主要操作步骤可参照如下几步：

1. 确保当前PC端下的GCC版本号是否符合编译要求；
2. 执行 `horizon_runtime_sample/code` 目录下的 `build_x86.sh` 脚本即可一键编译PC端环境下的可执行程序，可执行程序 and 对应依赖库生成并分别存放在 `xj3/script_x86/x86` 目录下的 `bin` 和 `lib` 子目录；
3. 正确完成编译后，`xj3/script_x86/x86` 的目录结构为：

Bash

```
1 |—— 00_quick_start
2 |   |—— README.md
3 |   |—— run_mobilenetV1.sh          # 示例快速运行脚本
4 |—— README.md
5 |—— x86
6 |   |—— bin
7 |       |—— run_mobileNetV1_224x224  # 编译产生的可执行文件
8 |       |—— lib                      # 编译执行依赖相关库
9 |           |—— libdnn.so
10 |           |—— libhbdk_sim_x86.so
11 |           |—— libopencv_world.so.3.4
```

4 运行

正确完成编译后，进入 `xj3/script_x86/00_quick_start` 目录下，执行 `run_mobilenetV1.sh` 脚本即可运行。

Bash

```
1 sh run_mobilenetV1.sh  # 运行mobilenetV1示例推理脚本
```

5 结果展示

运行成功结果如下图所示：

```
[root@aafaldfala22 00_quick_start]# sh run_mobilenetV1.sh
../x86/bin/run_mobileNetV1_224x224 --model_file=../model/runtime/mobilenetv1/mobilenetv1_224x224_nv12.bin --image_file=../
./data/cls_images/zebra_cls.jpg --top_k=5
I0000 00:00:00.000000 200 vlog_is_on.cc:197] RAW: Set VLOG level for "*" to 3
core[0] open!
core[1] open!
```

```
[HBRT] set log level as 0. version = 3.13.31
[DNN] Runtime version = 1.9.3 (3.13.31 HBRT)
I0614 10:55:19.099920 200 run_mobileNetV1_224x224.cc:157] read image to tensor as nv12 success
WARNING: cnn_core_fc_avl_cap is ignored in pseudo firmware
WARNING: bpu_mem_cache_flush is ignored in pseudo firmware
I0614 10:55:19.935587 200 run_mobileNetV1_224x224.cc:192] TOP 0 result id: 340
I0614 10:55:19.935652 200 run_mobileNetV1_224x224.cc:192] TOP 1 result id: 292
I0614 10:55:19.935668 200 run_mobileNetV1_224x224.cc:192] TOP 2 result id: 282
I0614 10:55:19.935683 200 run_mobileNetV1_224x224.cc:192] TOP 3 result id: 83
I0614 10:55:19.935707 200 run_mobileNetV1_224x224.cc:192] TOP 4 result id: 290
```

至此，在x86端成功完成模型推理和结果输出的全过程。