

基于存内计算芯片开发板 机器学习算法开发

---训练手册

目录

实验环境搭建

模块一：软件包下载及环境搭建

动手实验：机器学习简单算法开发实践

模块一：Python 模型训练及导出

模块二：Mapper 知存工具链 mapper 的输入及产物示例

模块三：Project 板端工程示例

模块四：Tools 串口工具及模型转换工具

实验环境搭建

模块一：软件包下载及环境搭建

步骤一：搭建 docker、mappper 环境：

①Linux 下通过 Docker 直接下载，获取指令：

```
docker pull witin/toolchain:v001.000.024
```

以下为 Mapper 两种安装流程

② 本地 mapper 镜像加载流程：

预先在 windows wsl 或 Liunx 系统下安装 docker 环境。

- 加载镜像：**docker load -i 工具链镜像.tar**
- 查看镜像：**docker images**
- 重新进入容器：1)**docker start 工具链镜像名称**；2)**docker attach 工具链镜像名称**
- 退出容器：**exit**
- 将本地文件（或文件夹）上传至容器：**docker cp 本地文件路径 工具链镜像名称:/workspace/witin_mapper(即本地容器名称:/+容器内路径)**
- 将容器文件（或文件夹）下载至本地：**docker cp 工具链镜像名称:/workspace/witin_mapper(即本地容器名称:/+容器内路径) 本地文件路径**

②Window 环境，可以通过 docker desktop 来使用 docker:

1) 下载安装 Docker desktop(win10 或以上) :

[Docker Desktop: The #1 Containerization Tool for Developers | Docker](#)

2) 通常需要更新 WSL，下载链接如下，更新后需要重启生效:

[旧版 WSL 的手动安装步骤 | Microsoft Learn](#)

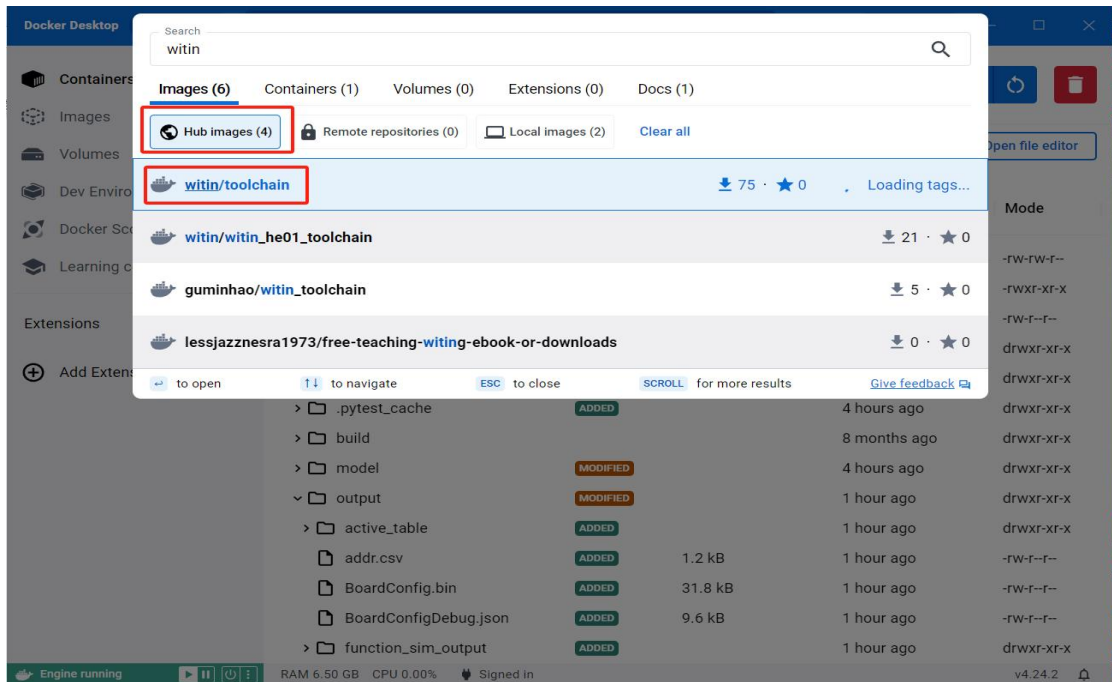
3) Docker desktop 基本使用教程 :

[Docker-desktop\(Docker 桌面版\)——入门篇_dockerdesktop 干嘛用的-CSDN 博客](#)

4) Docker desktop 通常默认安装在 c:\Program File\docker，可以通过软连接的形式修改 Docker 安装路径 :

[如何将 Docker \(Windows 桌面版\)自定义安装目录_自定义 docker 安装路径-CSDN 博客](#)

5) 在 Docker desktop 里，可以通过搜索获得 witin_toolchain，我们需要的是 034 版本 (ps:Hub 反应慢可以开 VPN 获取，或者使用镜像路径，具体操作方式见 3 链接)



5) 测试：

①：管理员模式下打开命令行窗口

docker run -it --name XXXX witin/toolchain:v001.000.024

②：默认进入 workspace 目录下，可以进入 witin_mapper 下执行测

试脚本：

cd witin_mapper

python3

tests/python/frontend/onnx/witin/wtm2101/precision/XXXX.py

```
管理员: 命令提示符 - docker attach 92f68360bd03
root@92f68360bd03:/workspace/witin_mapper# python3 tests/python/frontend/onnx/witin/wtm2101/precision/test_forward_TDNN.py
[07:14:16] [INFO]low_level_mapping/src/session/pass/array_extend_pass.cpp:201: Array Alloc: have set reserved column 128[07:
14:16] [INFO]low_level_mapping/src/memory/array/array_alloc_interface.cpp:43: Array Alloc: have set reserved column 128
[07:14:16] [INFO]low_level_mapping/src/session/pass/array_extend_pass.cpp:201: Array Alloc: have set reserved column 1282023
9 18 7 14 19
create ./output/output_test_forward_TDNN18_Oct_2023_07_14_16_570292/map//array_space.png
[07:14:22] [WARNING]witin_mapper_pro/witin_mapper/src/runtime/graph/npu_graph_runtime.cc:434: Warning: cannot find "in" amon
g input, have input node:
0 : in_kws;
use default input 0
[07:14:22] [WARNING]witin_mapper_pro/witin_mapper/src/runtime/graph/npu_graph_runtime.cc:434: Warning: cannot find "in" amon
g input, have input node:
0 : in_kws;
use default input 0
[07:14:22] [INFO]low_level_mapping/src/session/pass/array_extend_pass.cpp:201: Array Alloc: have set reserved column 128[07:
14:22] [INFO]low_level_mapping/src/session/pass/array_extend_pass.cpp:201: Array Alloc: have set reserved column 128[07:14:2
2] [INFO]low_level_mapping/src/memory/array/array_alloc_interface.cpp:43: Array Alloc: have set reserved column 128
Enter json mode!
/workspace/witin_mapper/witin_sim/BB04P1/filter.config
model version: v001.000.034
model name: model_1_net_in_kws_tdn9
chip_id: ChipId::BB04P1
data: 2023-10-18-7-14-18
net count: 1
npu_simulator cost 850.6357940000ms
func simulation result is equal to cmodel result!
root@92f68360bd03:/workspace/witin_mapper#
```

③：使用 exit 退出，再次进入可按如下操作：

④：通过 docker ps -a 获取容器 id，然后打开进入容器

```
管理员: 命令提示符 - docker attach 92f68360bd03
C:\WINDOWS\system32>docker run -it --name witin_toolchain_34 witin/toolchain:v001.000.034
root@92f68360bd03:/workspace# exit
exit

C:\WINDOWS\system32>docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS   NAMES
92f68360bd03   witin/toolchain:v001.000.034        "/bin/bash"             About a minute ago    Exited (0) 31 s     witin_toolchain_34
econds ago
140bc8ae49a6   witin/toolchain:v001.000.021        "/bin/bash"             4 hours ago         Exited (0) 2 ho     witin_toolchain
urs ago

C:\WINDOWS\system32>docker start 92f68360bd03
92f68360bd03

C:\WINDOWS\system32>docker attach 92f68360bd03
root@92f68360bd03:/workspace#
```

Tips: 也可使用现场提供的搭建好的 mapper 环境, 记得安装 python 和 pytorch

动手实验：机器学习简单算法开发实践

模块一：模型训练及导出

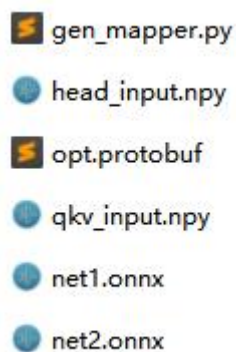
步骤一：进入 python 文件夹，运行 train.py，模型训练完毕后，在 models 文件夹下生成 bestModel.pth，此即我们模型的网络权重。

步骤二：运行 export_onnx.py，在 models 文件夹下生成 net1.onnx 和 net2.onnx

模块二：知存 onnx 格式转芯片所需 mapper 格式

mapper 是知存工具链制定的一套工具，目的在于将模型转换至芯片所需格式。

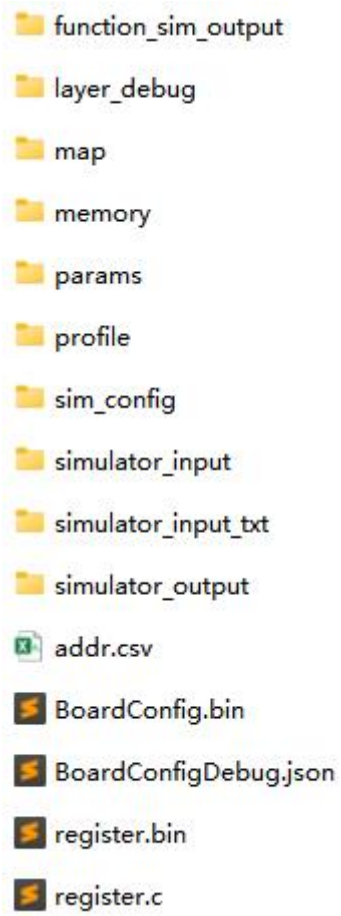
步骤一：进入 mapper/in 文件夹,将上一步训练完毕的 net1.onnx 和 net2.onnx 拷贝过来，此文件夹下即为给知存工具链 mapper 的输入。



步骤二：我们在工具链 mapper 环境下，运行 python gen_mapper.py

mapper 的环境搭建，可咨询知存工作人员，或使用现场提前搭建好 mapper 环境的电脑。

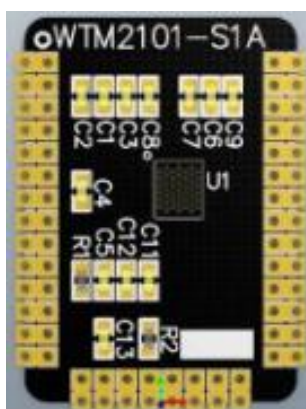
mapper 的生成产物示例如下：



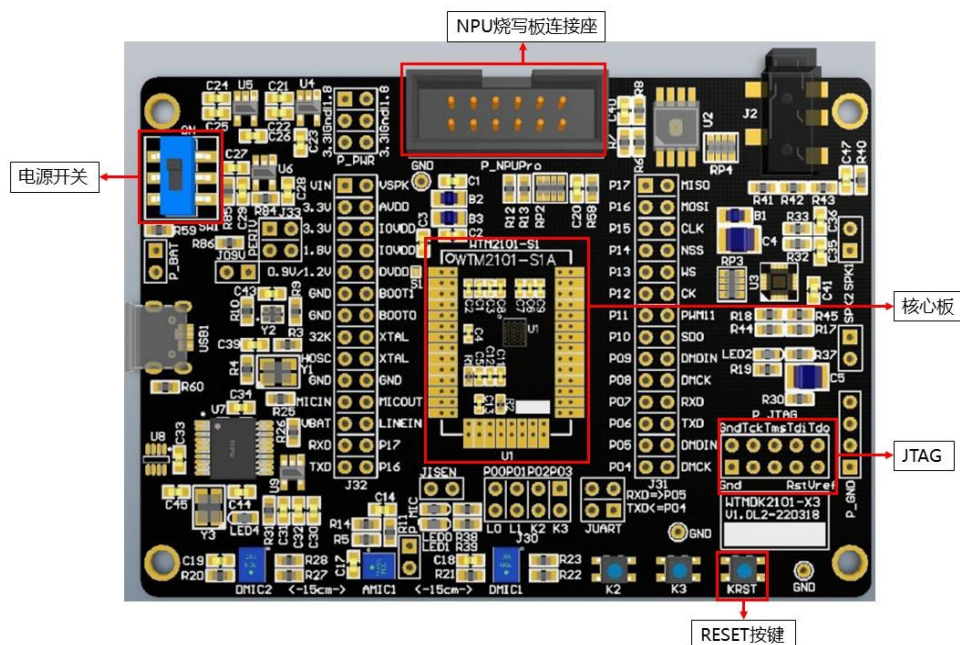
模块三：烧写 NPU 权重

WTMDK2101-X3 是针对 WTM2101 AI SOC 设计的评估板，包含：

- (1) **WTM2101 核心板**，即我们的存算芯片。
- (2) **和 I/O 板**：WTM2101 运行需要的电源、以及应用 I/O 接口等。



核心板示意图



WTMDK2101-X3 I/O 板示意图

步骤一：使用 Project/ WitinProgramTool_WTM2101 下的 WitinProgramTool.exe 进行模型权重烧写。

步骤二：烧写指令：

`.\WitinProgramTool.exe -m init`

`.\WitinProgramTool.exe -m program -i XXXX\map.csv -k 2`

其中 XXXX 为【模块二】中生成的 output\map

示例:

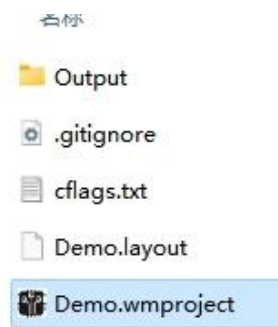
```
.\WitinProgramTool.exe -m init
```

```
.\WitinProgramTool.exe -m program -i C:\workspace\yxp\witmemDemo\mapper\output\map\map.csv -k 2
```

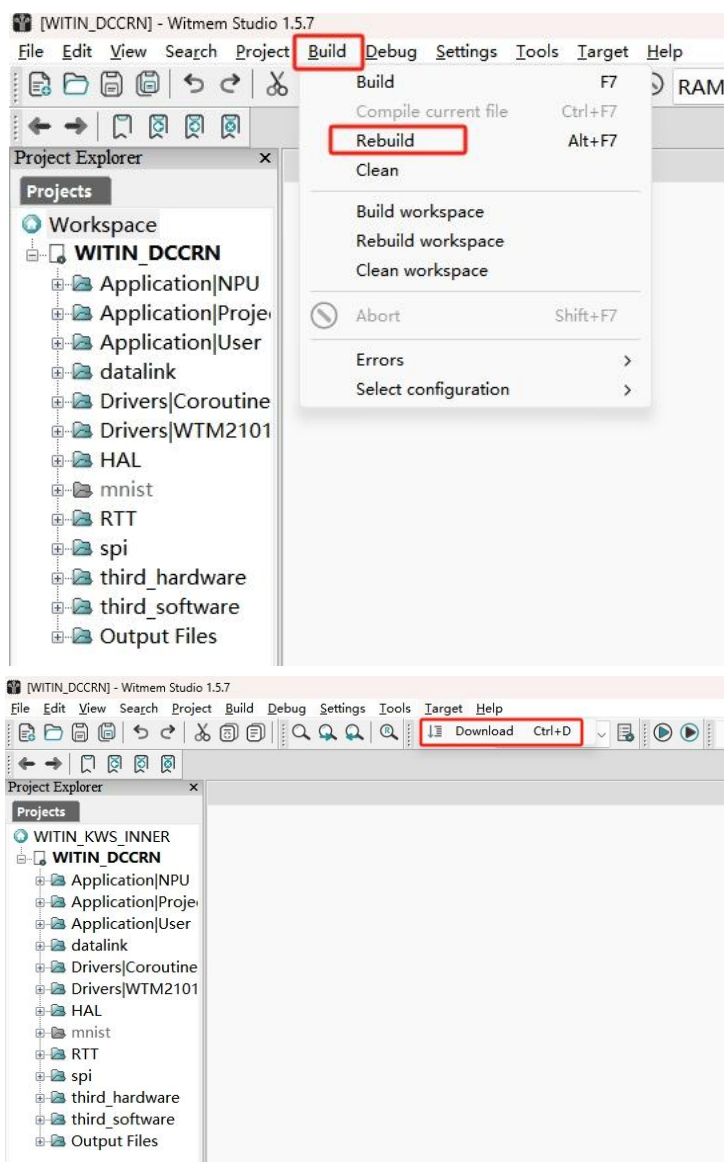
模块四：编译下载板端工程，验证测试

步骤一： 将【模块二】生成的 register.c 放在 Project/np_model/ output_attention 下,

步骤二： 打开工程 project/witinkws_WTM2101_P1/SES-RISCV/Demo.wmproject

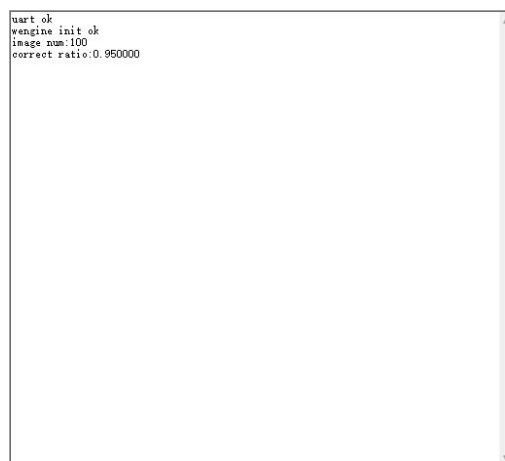
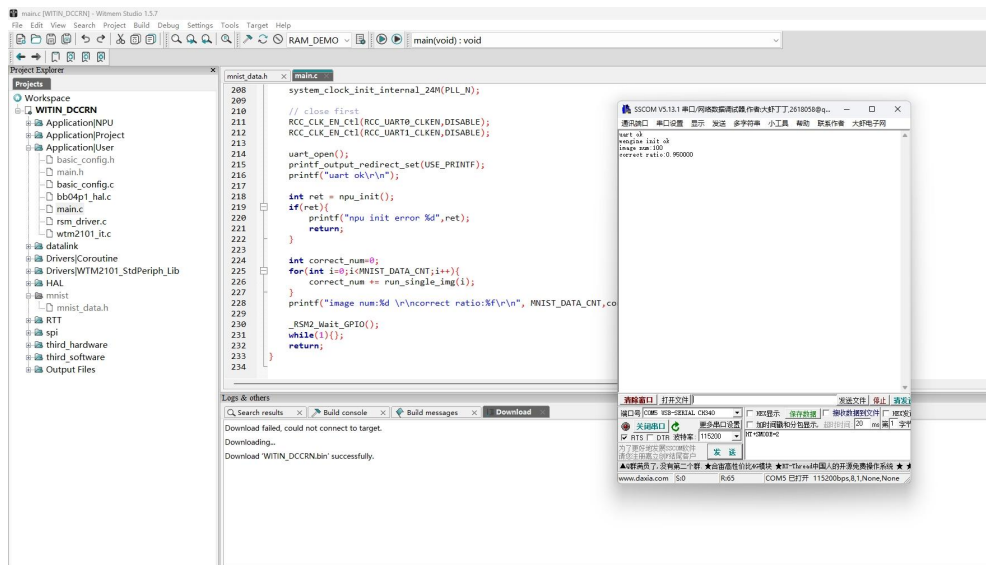


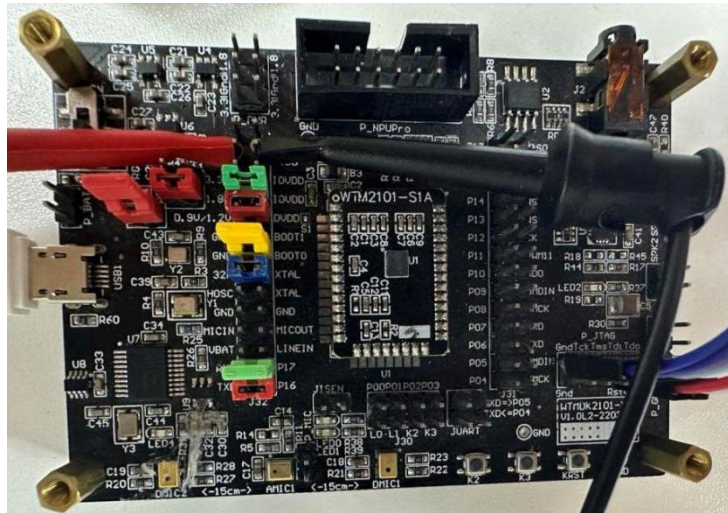
步骤三：下载工程



注：本 demo 所用为 x3 开发板，所用串口 GPIO 为 16,17。若是其他开发板，请根据情况修改串口。

步骤四： 打开 tools 中的串口工具，波特率 115200，查看输出：





至此，我们完成了机器学习算法从训练到部署的全流程，本教程结束。