开放原子开源基金会
OPENATOM FOUNDATION

OpenHarmony

技术筑生态 智联赢未来

第二届开放原子开源基金会OpenHarmony技术大会

2023.11.04 I 中国·北京

主办单位：OpenHarmony项目群技术指导委员会（TSC）
合作单位：华为、润开鸿、九联开鸿、软通动力、深开鸿
合作媒体：电子发烧友、51CTO、SegmentFault 思否、黄大年茶思屋科技网站、稀土掘金

算力NPU在OpenHarmony上的适配及应用赋能

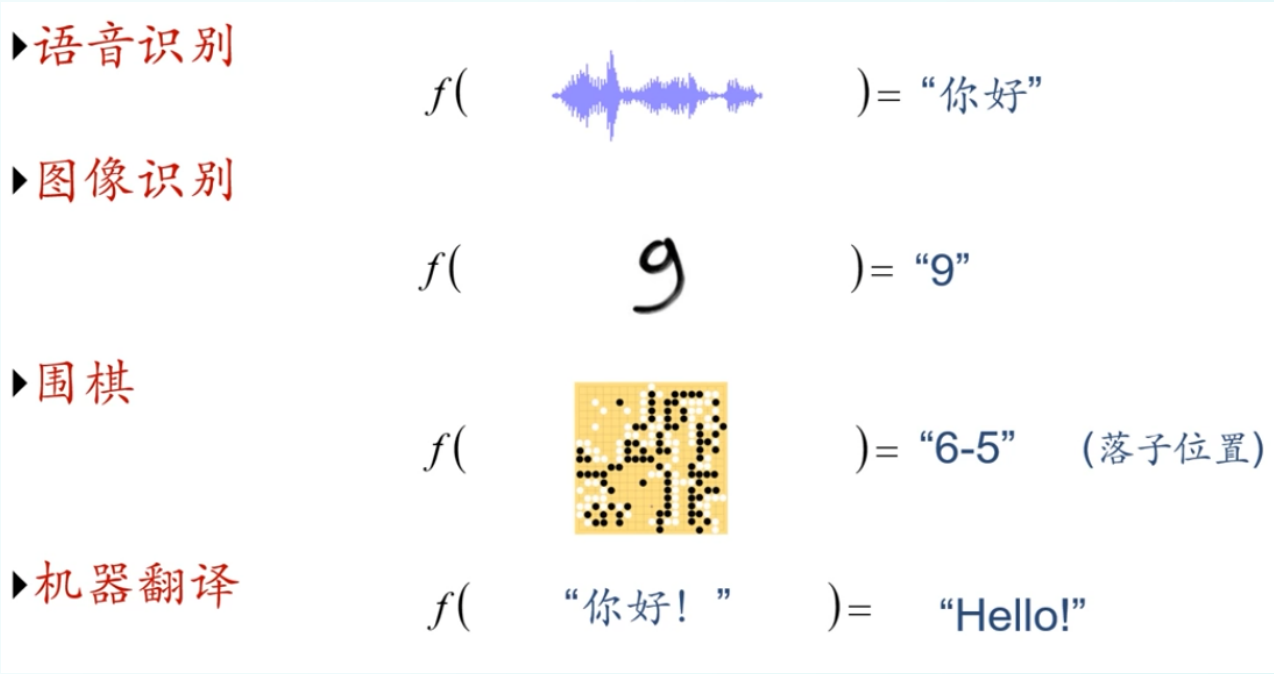
九联科技　　戴浚文

目录

1.神经网络模型的概念

1.1模型的本质

神经网络的本质就是通过线性函数和非线性函数来拟合特征与目标之间的真实函数关系

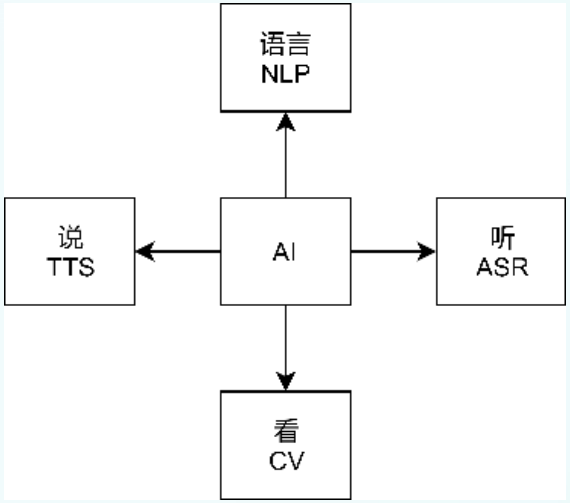


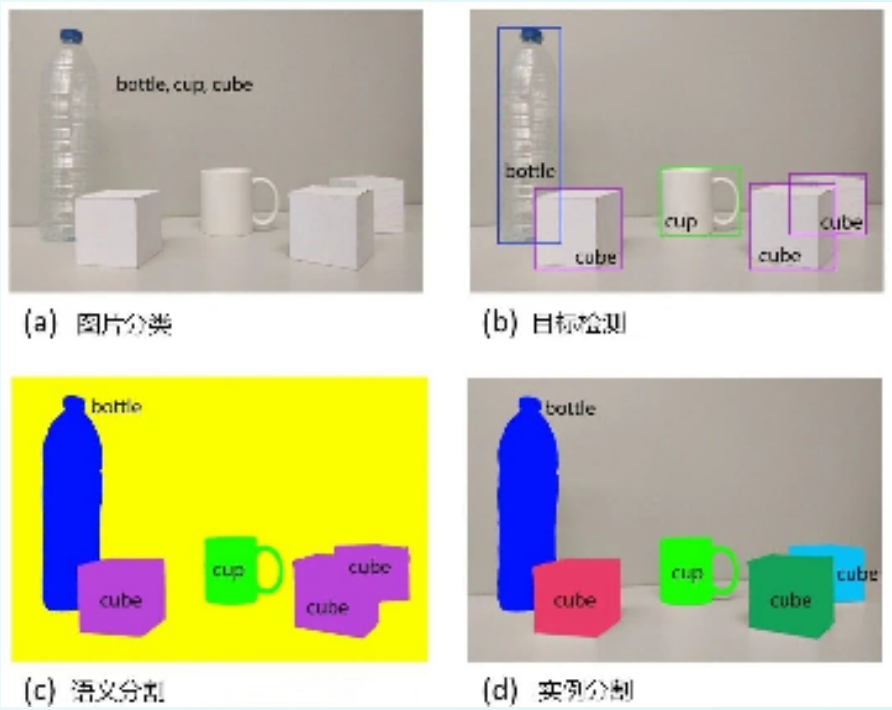
1.神经网络模型的概念

1.2模型的网络类型

网络模型涉及计算机视觉（CV）、语音识别（ASR）、语音合成（TTS）、自然语言处理（NLP）等

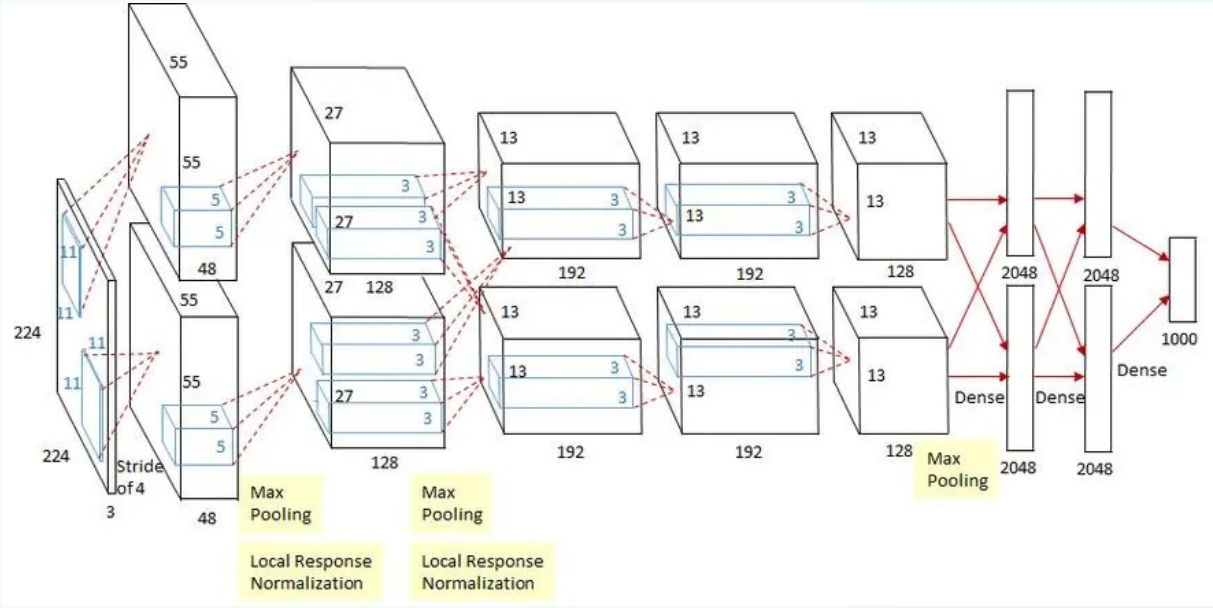
以计算机视觉(CV)为例，可细分为分类、检测、分割等





1.神经网络模型的概念

1.2模型的网络类型

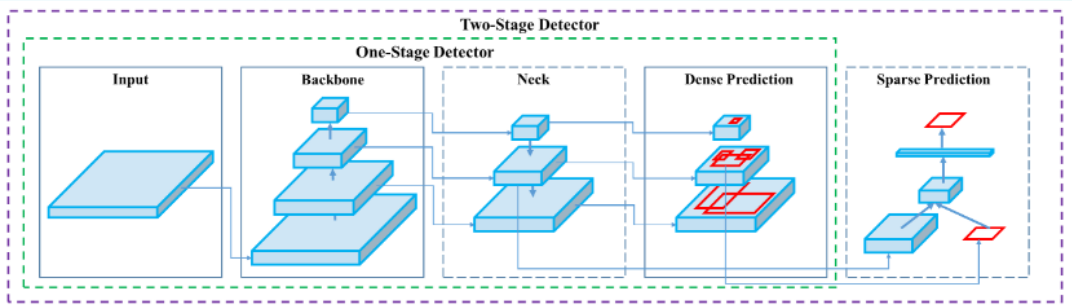


图像分类

1.神经网络模型的概念

1.2模型的网络类型

目标检测



Input: { Image, Patches, Image Pyramid, … }

Backbone: { VGG16 [68], ResNet-50 [26], ResNeXt-101 [86], Darknet53 [63], … }

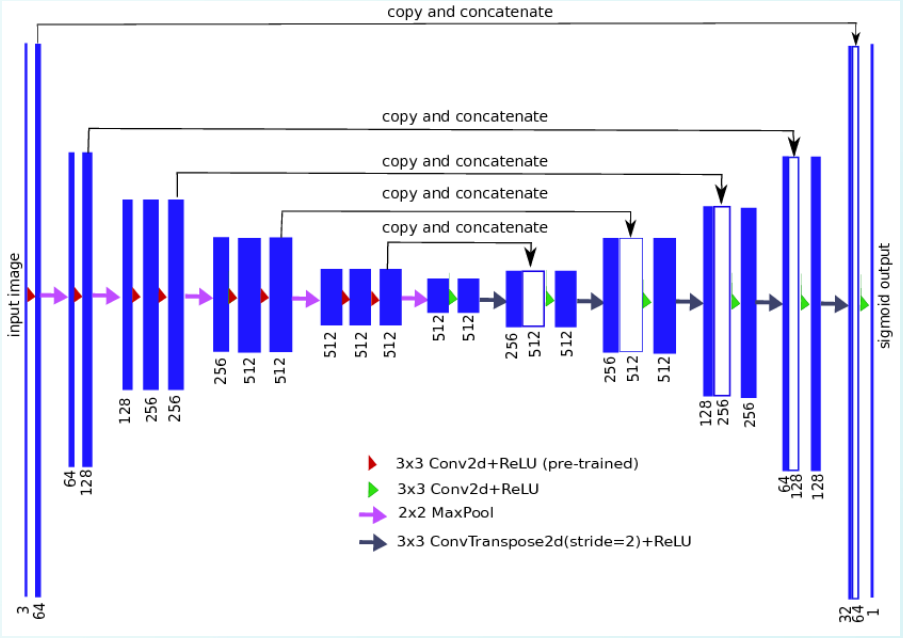
Neck: { FPN [44], PANet [49], Bi-FPN [77], … }

Head:

Dense Prediction: { RPN [64], YOLO [61, 62, 63], SSD [50], RetinaNet [45], FCOS [78], … }

Sparse Prediction: { Faster R-CNN [64], R-FCN [9], … }

语义分割



1.神经网络模型的概念

1.3框架和模型的生成

常见的框架有：

1. Tensorflow
2. Keras
3. PyTorch
4. ONNX
5. Caffe
6. Mxnet
7. Paddle Paddle
8. Mindspore

训练模型流程

2.模型的转换及在板端的应用

2.1模型的量化和转换

模型量化就是将浮点存储（运算）转换为整型存储（运算）的一种模型压缩技术。

量化可以带来：
1.更少的存储开销和带宽需求。
2.更快的计算速度。
3.更低的能耗与占用面积。
4.某些硬件加速器如 DSP/NPU 只支持 int8

Models	Data Shape	Batch Size	Metric	C5.24xlarge Base	C5.24xlarge Fusion	C5.24xlarge Fusion +Quantization	Speedup	FP32 Acc	INT8 Acc
ResNet50 V1	3x224x224	1	Latency(ms)	9.18	6.05	2.41	3.81	76.48	76.10
		64	Throughput(img/sec)	347.77	610.04	2,232.67	6.42		
ResNet101 V1	3x224x224	1	Latency(ms)	17.28	12.08	4.92	3.51	77.30	77.02
		64	Throughput(img/sec)	201.88	316.52	1,210.37	6.00		
MobileNet 1.0	3x224x224	1	Latency(ms)	2.96	2.02	1.01	2.92	72.14	71.97
		64	Throughput(img/sec)	1,070.08	2,222.70	5,778.97	5.40		
Inception V3	3x299x299	1	Latency(ms)	13.24	10.05	6.07	2.18	77.86	77.95
		64	Throughput(img/sec)	304.72	423.26	1,344.95	4.41		
SSD-VGG16	3x300x300	1	Latency(ms)	25.89	25.25	8.24	3.14	83.58 mAP	83.33 mAP
		224	Throughput(img/sec)	78.01	84.09	317.04	4.06		

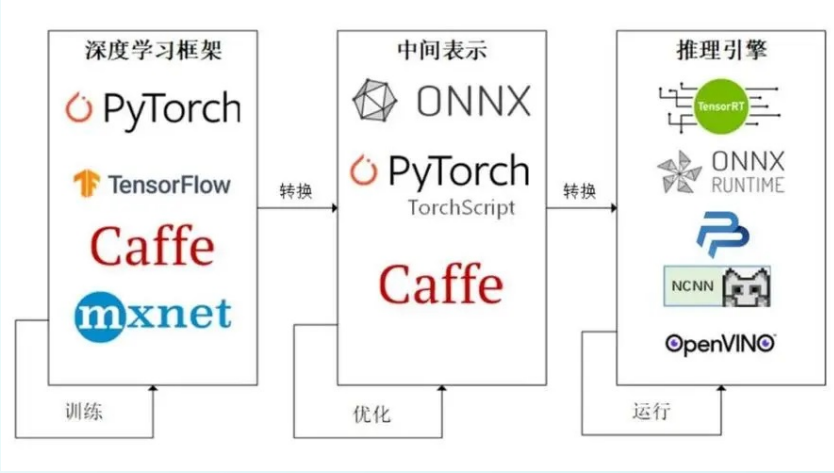
2.模型的转换及在板端的应用

2.2模型的量化和转换

读取载入A框架生成的模型文件，读取并识别模型网络中的张量数据的类型/格式、运算单元的类型和参数、计算图的结构和命名规范，以及它们之间的其他关联信息。

识别得到的A模型结构和模型参数信息翻译成B框架支持的代码格式。比如B框架指Pytorch时，relu激活层(运算单元)这一信息可翻译为torch.nn.ReLu()。
在B框架下保存模型，即可得到B框架支持的模型文件。

训练生成模型文件，转换可优化模型结构和在推理引擎上运行

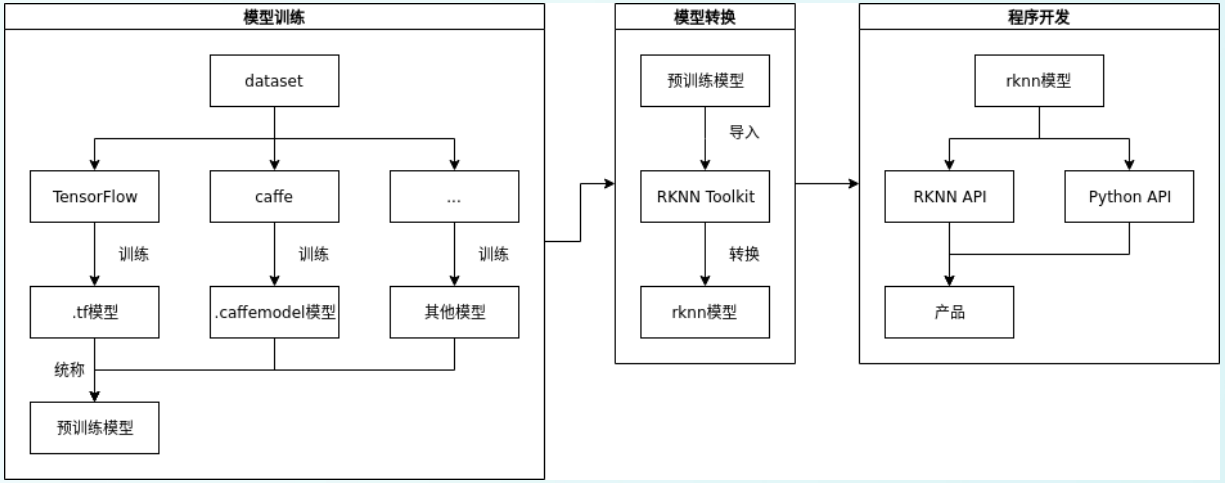


2.模型的转换及在板端的应用

2.2NPU的应用开发流程

NPU开发流程包括:
1.转换得到的NPU厂商所需的预训练模型
2.用NPU厂商的工具量化并转换成能在NPU推理接口运行的模型
3.在PC端可用工具模拟最终硬件模型的推理结果，也可将硬件模型放到板端基于NN API用NPU推理运行

瑞芯微NPU开发流程

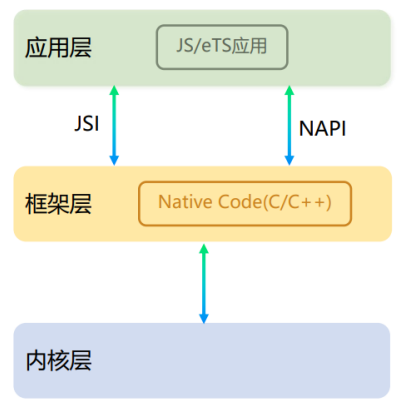


2.模型的转换及在板端的应用

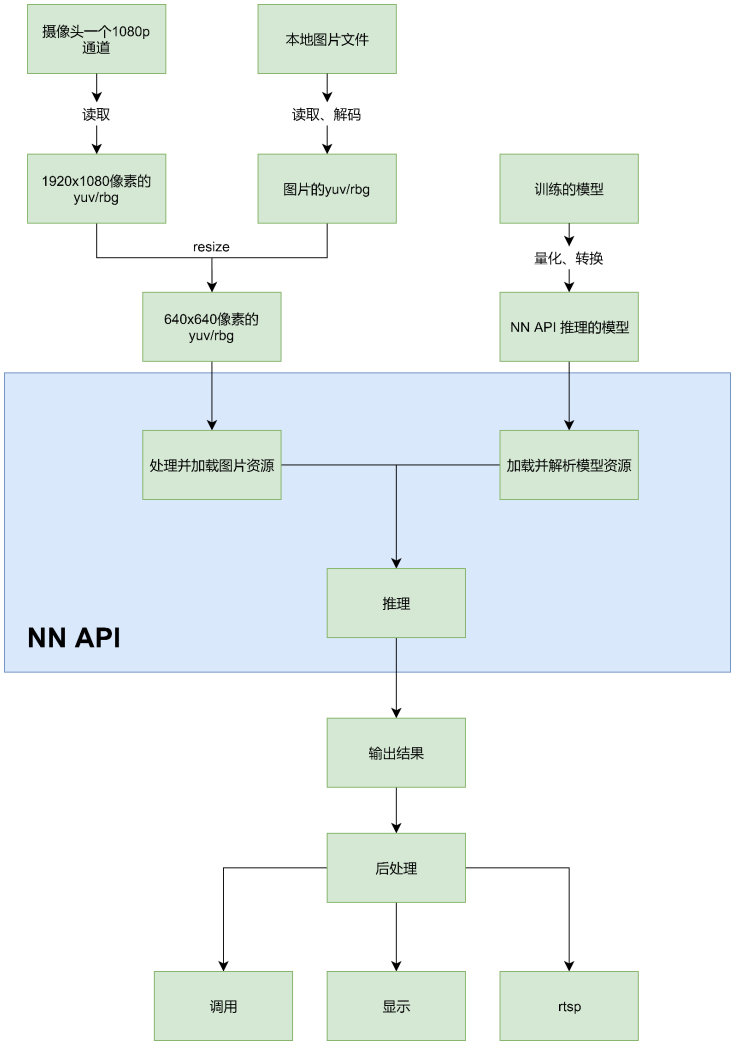
2.3NPU的应用开发流程

步骤：
1.定义图像资源（输入）和模型，对其进行处理。
2.NN API对输入资源处理和解析模型，NPU硬件加速推理。
3.对模型输出进行后处理，并应用后处理的结果。

OpenHarmony可通过NAPI框架将C/C++函数封装转换成JS/eTS函数，在app应用中调用



NAPI框架

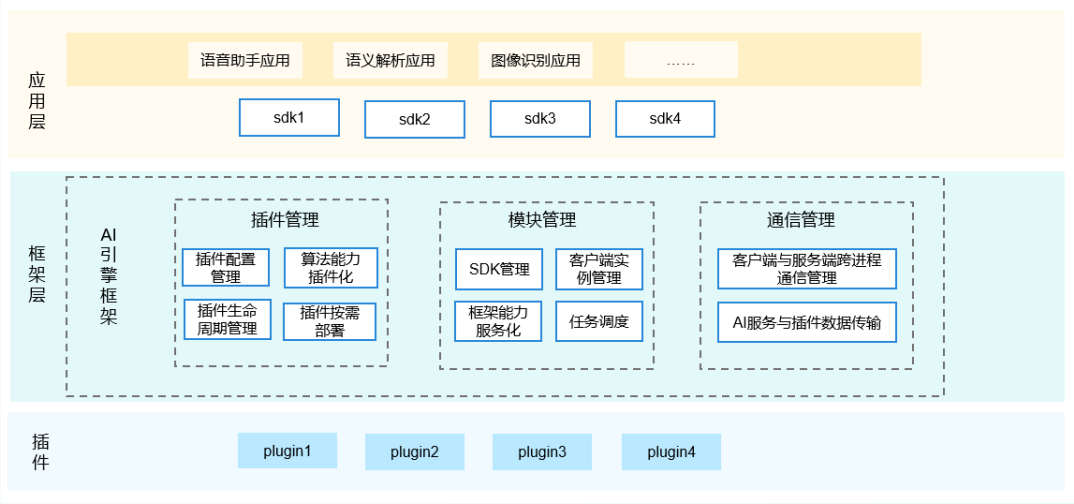


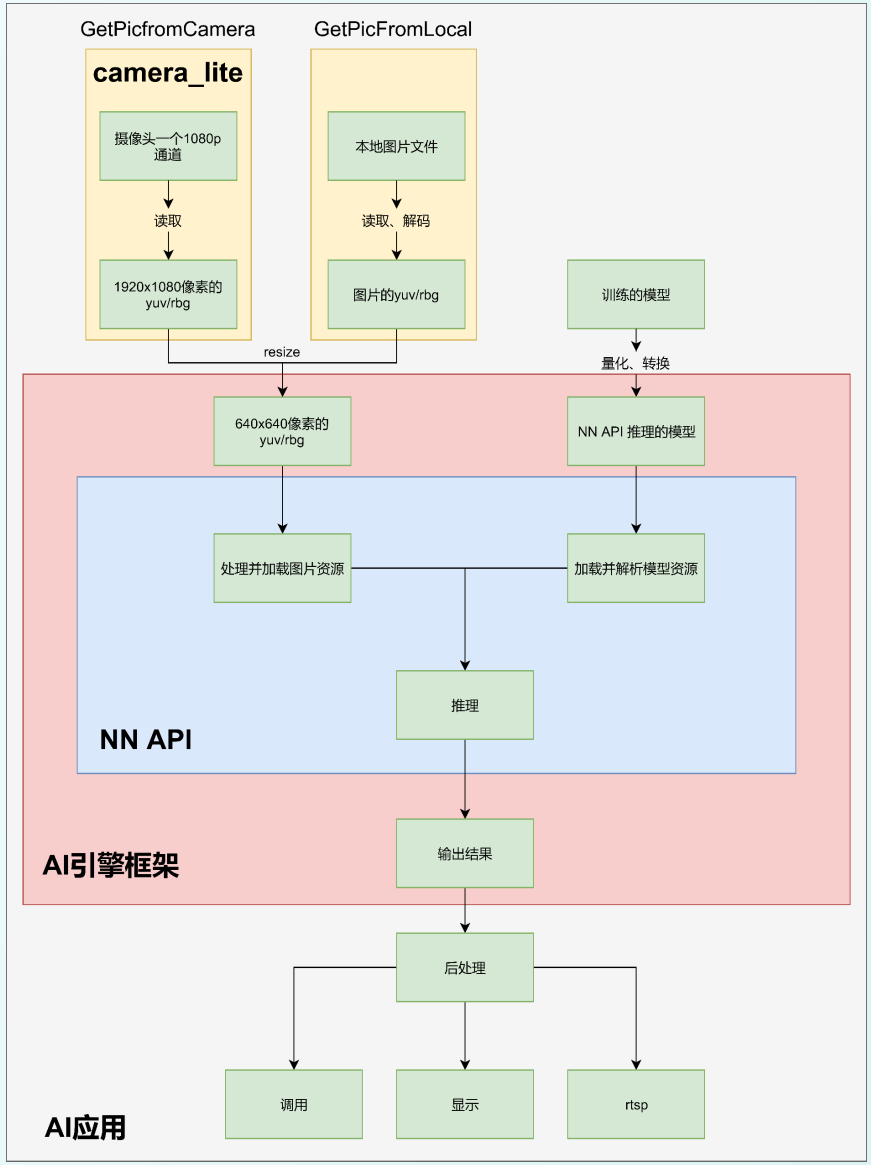
应用开发流程

3.NPU在OpenHarmony上的适配

3.1适配AI引擎框架

AI业务子系统是OpenHarmony提供原生的分布式AI能力的子系统。本次开源范围是提供了统一的AI引擎框架，实现算法能力快速插件化集成。框架中主要包含插件管理、模块管理和通信管理等模块，对AI算法能力进行生命周期管理和按需部署。后续，会逐步定义统一的AI能力接口，便于AI能力的分布式调用。同时，提供适配不同推理框架层级的统一推理接口。





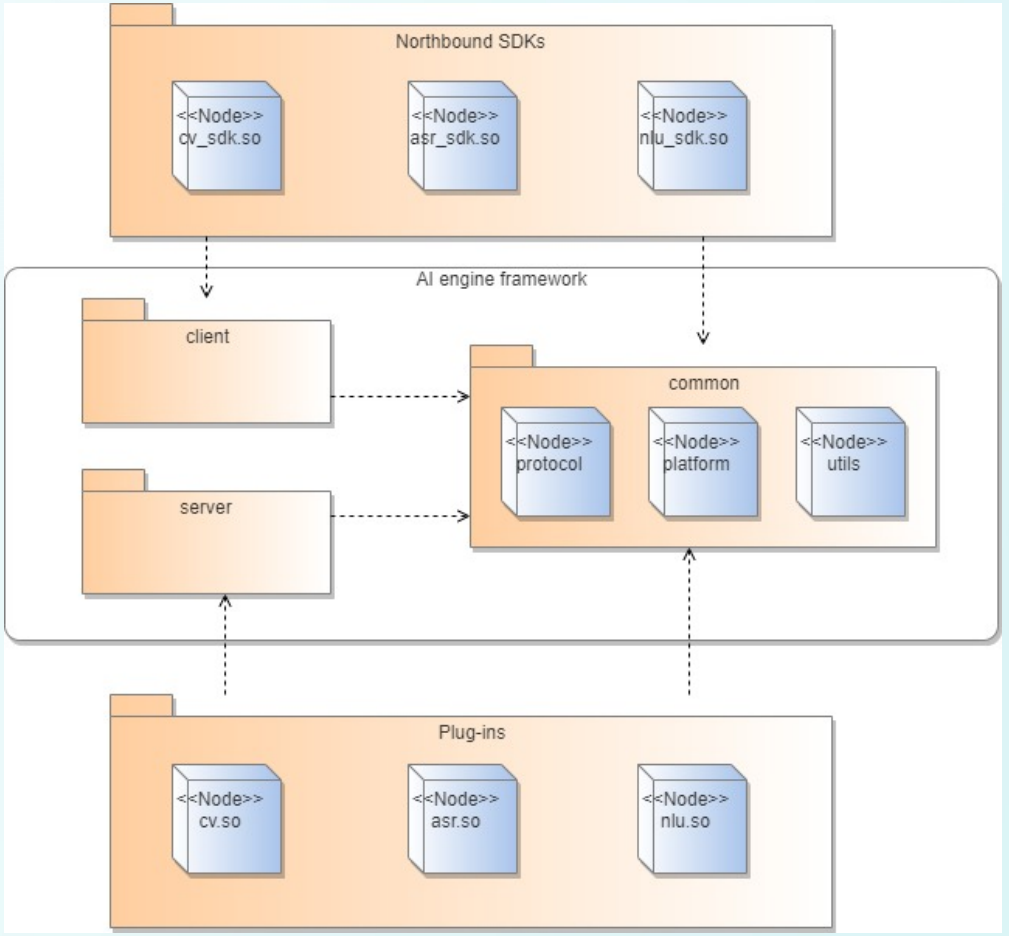
3.NPU在OpenHarmony上的适配

3.1适配AI引擎框架

AI引擎框架包含client、server和common三个主要模块，其中client提供server端连接管理功能，OpenHarmony SDK在算法对外接口中需封装调用client提供的公共接口；server提供插件加载以及任务管理等功能，各Plugin实现由server提供的插件接口，完成插件接入；common提供与平台相关的操作方法、引擎协议以及相关工具类，供其他各模块调用。

client端定义的接口	插件中定义的接口	功能
AieClientPrepare	Prepare	提供推理算法插件初始化功能，例如：加载唤醒词识别模型，将固定位置（/sdcard/wenwen_inst.wk）模型加载至内存。
AieClientSyncProcess	SyncProcess	提供同步执行推理算法的能力，例如：实现同步执行音频推理算法，判断音频中是否存在唤醒词。
AieClientAsyncProcess	AsyncProcess	提供异步执行推理算法的能力，当前唤醒词识别场景不涉及，但开发者可根据具体场景自行实现。
AieClientSetOption	SetOption	提供手动设置算法相关配置项，如置信度阈值、时延等超参数的能力。当前唤醒词识别场景未涉及，开发者可视具体场景自行实现。
AieClientGetOption	GetOption	提供获取算法相关配置项，以唤醒词识别为例：获取唤醒词模型中输入输出的规模，输入规模即为唤醒词识别模型要求输入的MFCC特征的维度（固定值：4000），输出规模即为结果的置信度得分维度（固定值：2）。
AieClientRelease	Release	提供卸载算法模型功能，以唤醒词识别为例：实现卸载相关模型，并清理特征处理器中的动态内存。

client端接口与插件中的接口对应关系及其实现功能



各模块之间的代码依赖关系

3.NPU在OpenHarmony上的适配

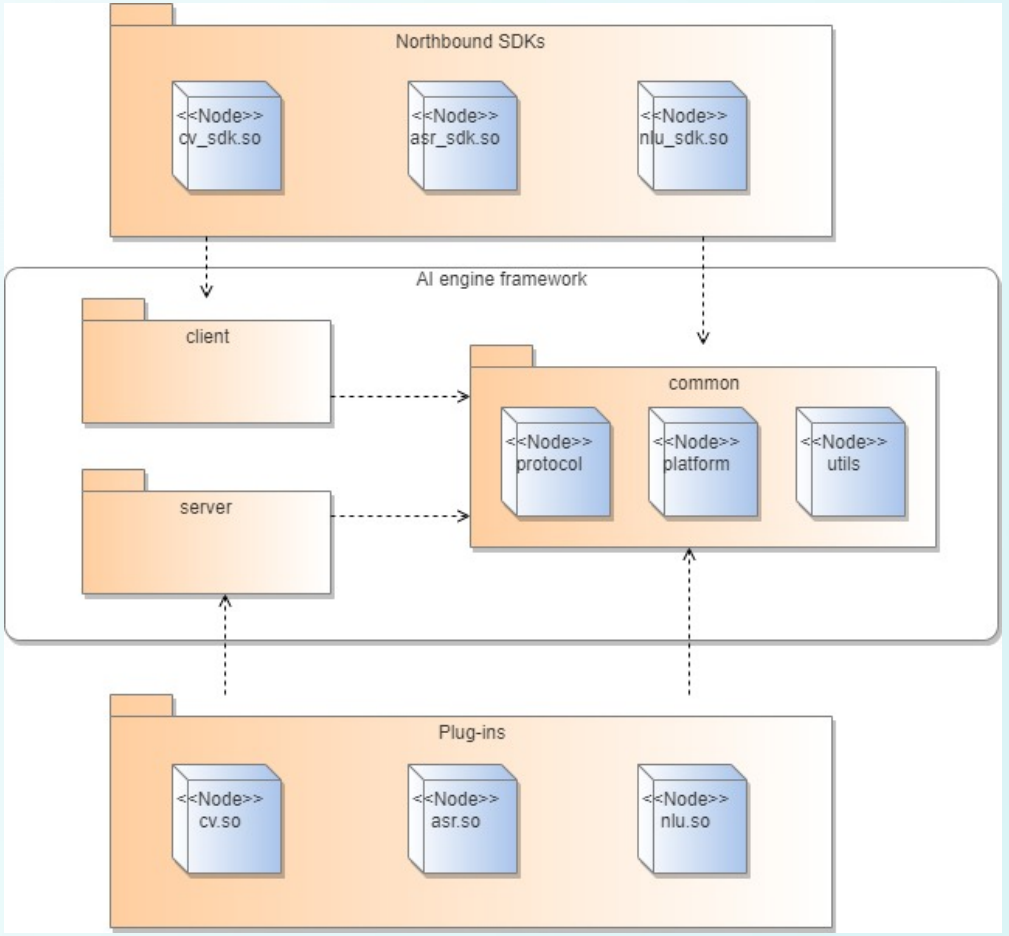
3.1适配AI引擎框架

开发SDK：SDK头文件的功能实现是基于对SDK的调用映射到对客户端的调用。

开发插件：AI引擎框架规定了一套算法插件接入规范，各插件需实现规定接口以实现获取插件版本信息、算法推理类型、同步执行算法、异步执行算法、加载算法插件、卸载算法插件、设置算法配置信息、获取指定算法配置信息等功能。

client端定义的接口	插件中定义的接口	功能
AieClientPrepare	Prepare	提供推理算法插件初始化功能，例如：加载唤醒词识别模型，将固定位置（/sdcard/wenwen_inst.wk）模型加载至内存。
AieClientSyncProcess	SyncProcess	提供同步执行推理算法的能力，例如：实现同步执行音频推理算法，判断音频中是否存在唤醒词。
AieClientAsyncProcess	AsyncProcess	提供异步执行推理算法的能力，当前唤醒词识别场景不涉及，但开发者可根据具体场景自行实现。
AieClientSetOption	SetOption	提供手动设置算法相关配置项，如置信度阈值、时延等超参数的能力。当前唤醒词识别场景未涉及，开发者可视具体场景自行实现。
AieClientGetOption	GetOption	提供获取算法相关配置项，以唤醒词识别为例：获取唤醒词模型中输入输出的规模，输入规模即为唤醒词识别模型要求输入的MFCC特征的维度（固定值：4000），输出规模即为结果的置信度得分维度（固定值：2）。
AieClientRelease	Release	提供卸载算法模型功能，以唤醒词识别为例：实现卸载相关模型，并清理特征处理器中的动态内存。

client端接口与插件中的接口对应关系及其实现功能



各模块之间的代码依赖关系

3.NPU在OpenHarmony上的适配

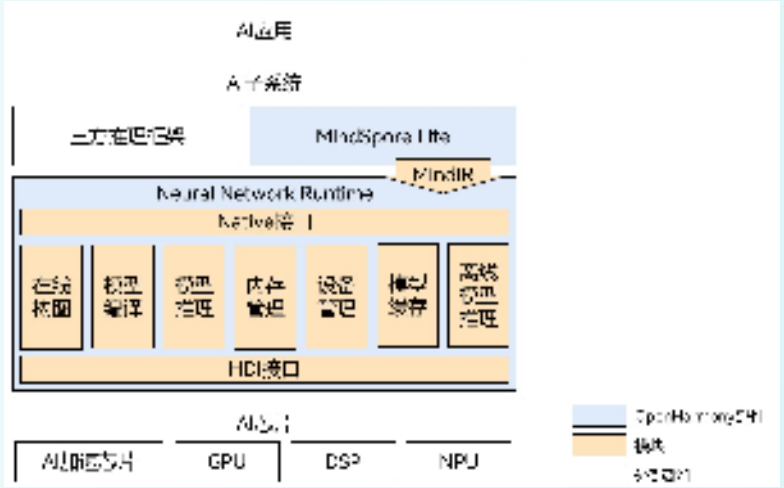
3.2适配NNRt推理计算

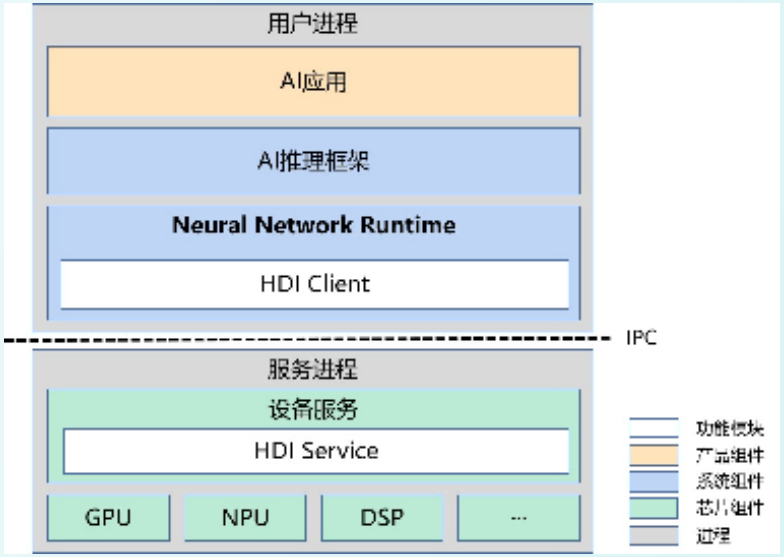
NNRt（Neural Network Runtime，神经网络运行时）是面向AI领域的跨芯片推理计算运行时，作为中间桥梁连通上层AI推理框架和底层加速芯片，实现AI模型的跨芯片推理计算。

NNRt开放了设备接口，芯片厂商通过设备接口将专有加速芯片接入NNRt，从而实现与OpenHarmony社区生态的对接。

整个架构主要分为三层，AI应用在应用层，AI推理框架和NNRt在系统层，设备服务在芯片层。AI应用如果要使用AI专用加速芯片进行模型推理，需要经过AI推理框架和NNRt才能调用到底层AI专用加速芯片，NNRt就是负责适配底层各种AI专用加速芯片的中间层。NNRt开放了标准统一的HDI设备接口，众多AI专用加速芯片都可以通过HDI接口接入OpenHarmony。此外NNRt也开放了标准统一的接口对接上层各种AI推理框架。

Neural Network Runtime架构图





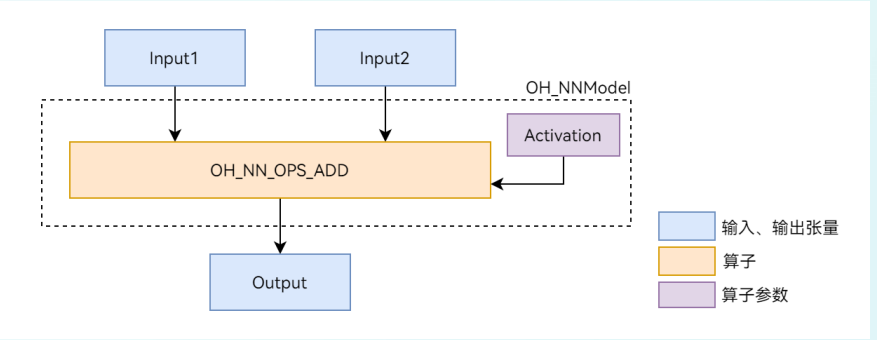
3.NPU在OpenHarmony上的适配

3.2适配NNRt推理计算

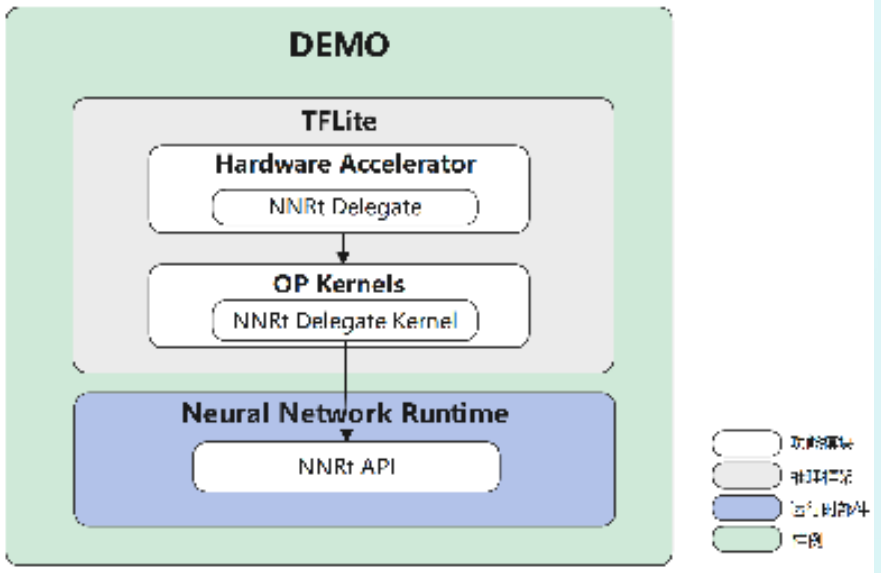
Neural Network Runtime作为AI推理引擎和加速芯片的桥梁，为AI推理引擎提供精简的Native接口，满足推理引擎通过加速芯片执行端到端推理的需求。

以Add单算子模型为例，介绍Neural Network Runtime的开发流程。Add算子包含两个输入、一个参数和一个输出，其中的activation参数用于指定Add算子中激活函数的类型。

Add单算子网络示意图



模块示意图



3.NPU在OpenHarmony上的适配

3.2适配NNRt推理计算

Neural Network Runtime的开发流程主要包含模型构造、模型编译和推理执行三个阶段。

构造模型：

使用Neural Network Runtime接口，构造模型。

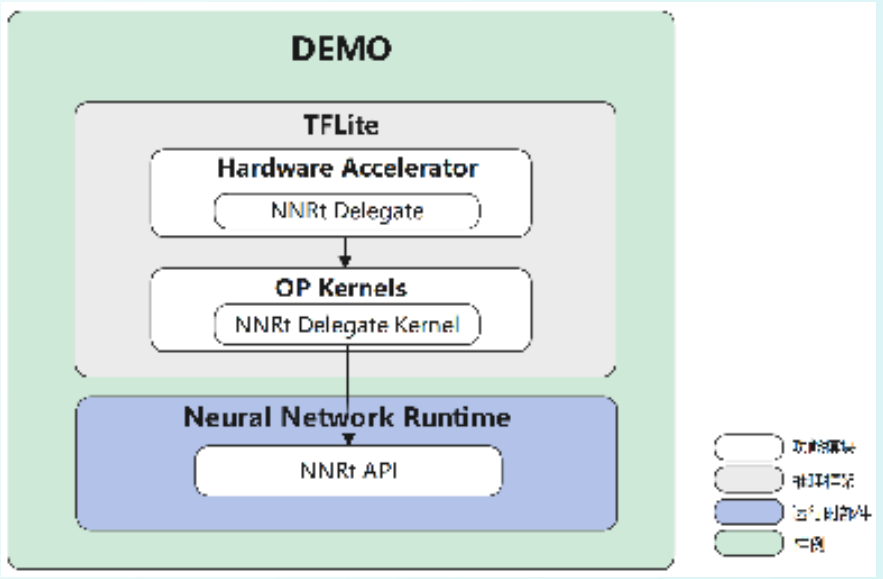
编译模型：

Neural Network Runtime使用抽象的模型表达描述AI模型的拓扑结构，在加速芯片上执行前，需要通过Neural Network Runtime提供的编译模块，将抽象的模型表达下发至芯片驱动层，转换成可以直接推理计算的格式。

推理执行：

通过执行模块提供的接口，将推理计算所需要的输入数据传递给执行器，触发执行器完成一次推理计算，获取模型的推理计算结果。

模块示意图



Thank you.