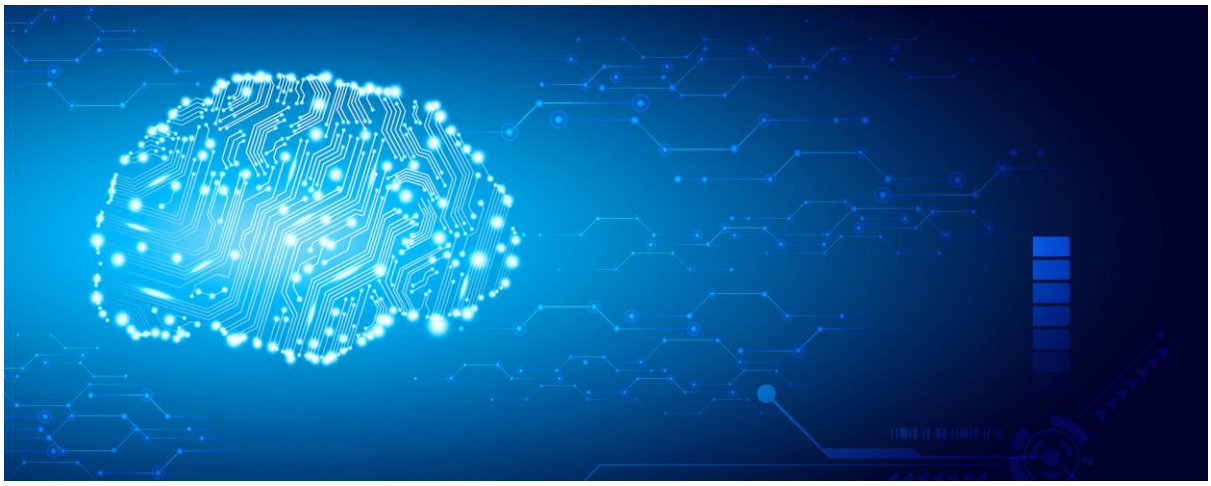

华为HiAI DDK使用手册

文档版本 v100.150.10

发布日期 2018-03-09

华为技术有限公司

商标声明

 和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

HiAI 申请方式

1.发送申请邮件到到邮箱：developer@huawei.com

2.邮件名称：HUAWEI HiAI+公司名称+产品名称

3.邮件正文：合作公司+联系人+联系方式+联系邮箱

4.我们将在收到邮件的 5 个工作内邮件给您反馈结果，请您注意查收。

官网：

http://developer.huawei.com/consumer/cn/devunion/ui/server/HiAI.html

目 录

华为 HiAI DDK 使用手册

1 概述

HiAI是面向移动终端的AI计算平台。其中HiAI API是移动计算平台中的人工智能计算库，该计算库面向人工智能应用程序开发人员，让开发者便捷高效地编写在移动设备上运行的人工智能应用程序。

HiAI API将作为统一的二进制文件发布。这组API主要作用是通过HiAI异构计算平台来加速神经网络的计算，当前仅支持在Kirin SoC上运行。

HiAI API集成在使用Kirin SOC芯片的android系统上，开发者可以在集成环境中运行神经网络模型，调用HiAI API进行加速计算。HiAI DDK(Device Development Kit)为对第三方开发者开放的HiAI资源包。

2 集成说明

2.1 功能介绍

HiAI DDK支持的基本功能：

- AI Model Manager

模型管理接口，提供模型管理功能，包括模型加载、模型计算、模型卸载等接口。

2.2 执行模式

HiAI DDK计算库执行方式：HiAI DDK使用经过编译优化的离线模型进行神经网络计算，实现更少的内存占用、更高的运算性能。用户通过caffe/tensorflow框架训练好的模型可以通过工具转换成离线模型。

2.2.1 离线模型生成

HiAI DDK提供模型转换工具支持将caffe和tensorflow深度学习模型转换为NPU格式的模型，转换工具的使用说明请参考《华为HiAI DDK集成手册》中第5章节“模型转换“。

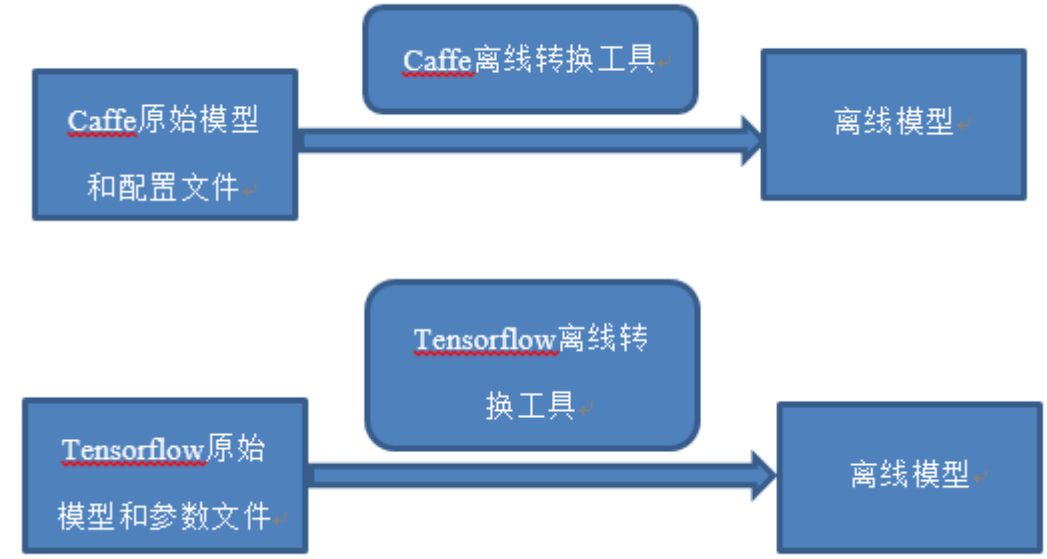

图 1

2.2.1 离线模型计算

在用户APK使用DDK的时候，需要调用模型管家的加载离线模型、运行离线模型、卸载离线模型的接口，完成用户输入数据在HIAI异构加速系统上的计算。详细参考《华为HIAI DDK集成手册》中第6.2章节”接口集成“。

2.4 支持算子

参考华为HiAI DDK算子规格说明文档。

2.5 限制条件

考虑到手机芯片上，ROM和RAM空间有限，需要对模型大小和运行内存有限制：

- 建议模型大小不超过100M
- 建议峰值内存大小不超过200M

2.6 支持接口

2.6.1 获取 DDK 版本号

实现HIAI_GetVersion获取DDK版本号的接口，函数内部维护一个静态的版本号，调用安卓系统接口__system_property_get去获取系统属性ro.config.hiaiversion中的值，即DDK版本号。

功能描述	获取系统中的 DDK 版本号
接口原型	char* HIAI_GetVersion();
参数说明	无
返回值	执行成功，返回相应的 DDK 版本号。 版本号名称，以<major>.<middle>.<minor>.<point>的形式描述版本。 <major>：产品形态，1XX：手机形态，2XX：边缘计算形态，3XX：Cloud 形态 <middle>：XXX 三位数表示，同一产品形态的 V 版本，如手机形态下，HiAIV100，HiAIV200 等 <minor>：增量 C 版本，新增大特性，XXX 三位数表示 < point >：B 版本或者补丁版本，XXX 三位数表示，最后一位非 0，表示补丁版本。 例如 kirin970 系统返回的版本号是 100.150.010.010 如果返回 000.000.000.000，表示此版本不支持 NPU 加速； 执行失败，返回相应错误值。

表格 1 HIAI_GetVersion 接口说明

2.6.2 创建模型管家

2.6.2.1 同步接口

创建模型管理对象实例后才可以使用模型接口，当前支持一个进程创建三个模型管

理实例，同步接口原型：

功能描述	创建模型管理实例接口
接口原型	HIAI_ModelManager* HIAI_ModelManager_create(void);
参数说明	无
返回值	模型管理对象实例指针，该指针需要用户调用 HIAI_ModelManager_destroy 主动销毁。

表格 2 HIAI_ModelManager同步接口说明

2.6.2.2 异步接口

功能描述	创建模型管理引擎的动态接口类
接口原型	HIAI_ModelManager* HIAI_ModelManager_create(HIAI_ModelManagerListener* listener);
参数说明	HIAI_ModelManagerListener：异步回调函数指针的结构体，包含了模型加载完成、模型运行完成、模型卸载完成、超时、错误处理、死亡监听的回调函数指针。 其中：HIAI_ModelManagerListener 的定义如下： typedef struct HIAI_ModelManagerListener_struct { void (*onLoadDone)(void* userdata, int taskStamp); void (*onRunDone)(void* userdata, int taskStamp); void (*onUnloadDone)(void* userdata, int taskStamp); void (*onTimeout)(void* userdata, int taskStamp); void (*onError)(void* userdata, int taskStamp, int errCode); void (*onServiceDied)(void* userdata); void* userdata; } HIAI_ModelManagerListener; void (*onLoadDone)(void* userdata, int taskStamp)：模型加载成功完成回调函数指针 void (*onRunDone)(void* userdata, int taskStamp)：模型运行完成回调函数指针 void (*onUnloadDone)(void* userdata, int taskStamp)：模型卸载完成回调函数指针 void (*onTimeout)(void* userdata, int taskStamp)：超时回调函数指针 void (*onError)(void* userdata, int taskStamp, int errCode)：错误处理回调函数指针 void (*onServiceDied)(void* userdata)：死亡监听回调函数指针 void* userdata; 上下文用户数据
返回值	模型管理引擎的对象接口。

表格 3 HIAI_ModelManager异步接口说明

2.6.3 模型加载

模型从应用层加载时，一是从APP下的assets目录下去做内存加载,这里模型是由

APP自己管理，读取模型的动作由APP来完成;二是可以从sdcard下做文件加载。第一步需要创建HIAI_ModelBuffer，第二步调用模型加载的接口实现模型加载，最后需要实现HIAI_ModelBuffer的销毁。

2.6.3.1 创建HIAI_ModelBuffer

功能描述	从路径中加载模型，创建 HIAI_ModelBuffer(用于从应用层 sdcard 中加载模型)
接口原型	HIAI_ModelBuffer* HIAI_ModelBuffer_create_from_file(const char* name, const char* path, HIAI_DevPerf perf)
参数说明	name：要加载的模型名称 path：要加载的模型所在路径 perf：NPU 频率，对应高中低三挡
返回值	返回 HIAI_ModelBuffer

表格 4 从文件中创建HIAI_ModelBuffer接口说明

功能描述	读取模型数据后加载模型，创建 HIAI_ModelBuffer(用于从应用层 assets 中加载模型)
接口原型	HIAI_ModelBuffer* HIAI_ModelBuffer_create_from_buffer(const char* name, void* modelBuf, int size, HIAI_DevPerf perf);
参数说明	name：要加载的模型名称 modelBuf：模型数据地址 size：模型数据长度 perf：NPU 频率，对应高中低三挡
返回值	返回 HIAI_ModelBuffer

表格 5 从模型数据地址创建HIAI_ModelBuffer接口说明

2.6.3.2 模型加载

功能描述	加载模型接口
接口原型	int HIAI_ModelManager_loadFromModelBuffers(HIAI_ModelManager* manager, HIAI_ModelBuffer* bufferArray[], int nBuffers);
参数说明	manager：模型管理引擎对象接口。 bufferArray[]：HIAI_ModelBuffer，单模型和多模型均可。 nBuffers：加载模型的个数。
返回值	执行成功，返回 0；执行失败，返回相应错误值。

表格 6 HIAI_ModelManager_loadFromModelBuffers 接口说明

2.6.3.3 销毁HIAI_ModelBuffer

功能描述	销毁 HIAI_ModelBuffer
接口原型	void HIAI_ModelBuffer_destroy(HIAI_ModelBuffer* b);
参数说明	b：需要销毁的 HIAI_ModelBuffer
返回值	无。

表格 7 HIAI_ModelBuffer_destroy 接口说明

2.6.4 获取模型输入输出

模型加载成功后，可以从读取指定模型名的输入shape和输出shape，读取完成后，将输入输出shape信息保存在内存中，最后释放。

2.6.4.1 获取模型输入输出的形状信息

功能描述	获取模型输入输出的形状信息
接口原型	HIAI_ModelTensorInfo* HIAI_ModelManager_getModelTensorInfo(HIAI_ModelManager* manager, const char* modelName);
参数说明	manager：模型管家实例 modelName：模型名称
返回值	执行成功，返回模型输入输入信息指针；执行失败，返回 NULL。

表格 8 HIAI_ModelManager_getModelTensorInfo 接口说明

2.6.4.2 释放模型输入输出形状信息内存

功能描述	释放模型输入输出形状信息内存
接口原型	void HIAI_ModelManager_releaseModelTensorInfo(HIAI_ModelTensorInfo* modelTensor);
参数说明	modelTensor：模型输入输入信息指针
返回值	无

表格 9 HIAI_ModelManager_getModelTensorInfo 接口说明

2.6.5 模型运行一

模型运行时，提供了特征数据的输入接口，模型运行接口，以及模型运行后，数据获取等接口。

2.6.5.1 创建HIAI_TensorBuffer

功能描述	创建 HIAI_TensorBuffer
接口原型	HIAI_TensorBuffer* HIAI_TensorBuffer_create(int n, int c, int h, int w);
参数说明	模型输入或者输出的 nchw n：tensor 的 batch c：tensor 的 channel h：tensor 的 height w：tensor 的 width
返回值	返回 HIAI_TensorBuffer

表格 10 HIAI_TensorBuffer_create 接口说明

2.6.5.2 获取模型输入或者输出的数据地址

功能描述	获取模型输入或者输出数据地址
接口原型	void* HIAI_TensorBuffer_getRawBuffer(HIAI_TensorBuffer* b);
参数说明	b：模型输入或输出的 HIAI_TensorBuffer
返回值	返回模型输入或者输出的数据地址

表格 11 HIAI_TensorBuffer_getRawBuffer 接口说明

2.6.5.3 获取模型输入或者输出的数据长度

功能描述	获取模型输入或者输出数据长度
接口原型	int HIAI_TensorBuffer_getBufferSize(HIAI_TensorBuffer* b);
参数说明	b：模型输入或输出的 HIAI_TensorBuffer
返回值	返回模型输入或者输出数据的长度

表格 12 HIAI_TensorBuffer_getBufferSize 接口说明

2.6.5.4 运行模型

功能描述	模型运行接口
接口原型	int HIAI_ModelManager_runModel(HIAI_ModelManager* manager, HIAI_TensorBuffer* input[], int nInput, HIAI_TensorBuffer* output[], int nOutput, int ulTimeout, const char* modelName);

参数说明	manager：模型管理引擎对象接口。 input[]：模型输入，支持多输入。 nInput：模型输入的个数。 output[]：模型输出，支持多输出。 nOutput：模型输出的个数。 ulTimeout：超时时间，在同步调用时不生效。 modelName：模型名称。
返回值	执行成功，返回 0；执行失败，返回相应错误值。

表格 13 HIAI_ModelManager_runModel 接口说明

2.6.5.5 销毁HIAI_TensorBuffer

功能描述	销毁 HIAI_TensorBuffer
接口原型	void HIAI_TensorBuffer_destroy(HIAI_TensorBuffer* b);
参数说明	b：需要销毁的 HIAI_TensorBuffer
返回值	无。

表格 14 HIAI_TensorBuffer_destroy 接口说明

2.6.6 模型运行二

除了2.6.4章节中的HIAI_ModelManager_runModel外，模型运行还可以使用本章节的接口。本章节的接口只提供了同步方式的接口，未提供异步方式的接口。

2.6.6.1 设置输入和输出

功能描述	设置模型的输入和输出
接口原型	int HIAI_ModelManager_setInputsAndOutputs(HIAI_ModelManager* manager, const char* modelname, HIAI_TensorBuffer* input[], int nInput, HIAI_TensorBuffer* output[], int nOutput);
参数说明	manager：模型管理引擎对象接口。 modelName：模型名称。 input[]：模型输入，支持多输入。

	nInput：模型输入的个数。 output[]：模型输出，支持多输出。 nOutput：模型输出的个数。
返回值	执行成功，返回 0；执行失败，返回相应错误值。

表格 15 HIAI_ModelManager_setInputsAndOutputs 接口说明

2.6.6.2 启动计算

功能描述	启动计算接口
接口原型	int HIAI_ModelManager_startCompute(HIAI_ModelManager* manager, const char* modelname);
参数说明	manager：模型管理引擎对象接口。 modelName：模型名称。
返回值	执行成功，返回 0；执行失败，返回相应错误值。

表格 16 HIAI_ModelManager_startCompute 接口说明

2.6.7 模型卸载

用户所有数据处理完成后，调用此接口停止计算。

功能描述	卸载模型
接口原型	int HIAI_ModelManager_unloadModel(HIAI_ModelManager* manager);
参数说明	manager：模型管理引擎对象接口。
返回值	执行成功，返回 0；执行失败，返回相应错误值。

表格 17 HIAI_ModelManager_unloadModel 接口说明

2.6.8 销毁模型管家

用户在卸载完模型之后可以去销毁模型管家。

功能描述	销毁模型管家
接口原型	void HIAI_ModelManager_destroy(HIAI_ModelManager* manager);
参数说明	manager：模型管理引擎对象接口。
返回值	无。

表格 18 HIAI_ModelManager_destroy 接口说明

3 如何集成

参考《华为HiAI DDK集成手册》

4 附录

4.1错误码定义

错误码	错误类型	错误码原型	错误码触发条件
1	模型名称长度错误	MODEL_NAME_LEN_ERROR	模型长度1-128
2	模型文件路径路径为空	MODEL_DIR_ERROR	模型文件路径为空
3	模型文件解密秘钥长度错误	MODEL_SECRET_KEY_ERROR	长度不为0或者64
4	模型参数文件解密秘钥长度错误	MODEL_PARA_SECRET_KEY_ERROR	长度不为0或者64
5	模型框架类型(tensorflow/caffe)选择错误	FRAMEWORK_TYPE_ERROR	框架选择不为tensorflow或者caffe或者KALDI
6	在线或离线模型类型选择错误	MODEL_TYPE_ERROR	模型类型选择不为在线和离线
7	IPU频率设置不正确	IPU_FREQUENCY_ERROR	频率设置不为LOW、Normal、Hign
8	加载的模型数量错误	MODEL_NUM_ERROR	模型数量为0或者超过20

9	模型大小错误	MODEL_SIZE_ERROR	模型大小为0
10	超时时间设置错误	TIMEOUT_ERROR	设置超时时间超过60000ms
11	输入数据形状错误	INPUT_DATA_SHAPE_ERROR	n*c*h*w为0
12	输出数据形状错误	OUTPUT_DATA_SHAPE_ERROR	n*c*h*w为0
13	输入数据的数量错误	INPUT_DATA_NUM_ERROR	输入数据数量为0或者超过20
14	输出数据的数量错误	OUTPUT_DATA_NUM_ERROR	输出数据数量为0或者超过20
15	创建的模型管家实例超过最大值	MODEL_MANAGER_TOO_MANY_ERROR	单个进程创建了三个以上client
18	模型名称重复错误	MODEL_NAME_DUPLICATE_ERROR	多个模型中文件名重复
19	hiaiserver连接失败	HIAI_SERVER_CONNECT_ERROR	hiaiserver服务未启动
20	hiaiserver连接中断	HIAI_SERVER_CONNECT_IRPT	hiaiserver连接后中断
500	模型尺寸不匹配	MODEL_TENSOR_SHAPE_NO_MATCH	输入输出的nchw和模型的nchw不匹配
999	接口过期	EXPIRATION_FUCNTION	调用编译模型接口
1000	内部错误	INTERNAL_ERROR	内部错误