密级状态：绝密(　　)　　秘密(　　)　　内部(　　)　　公开(√)

RKNN Compiler Support Operator List

（技术部，图形计算平台中心）

文件状态： [　] 正在修改 [√] 正式发布	当前版本：	v1.6.0
	作　　者：	NPU团队
	编　　辑：	刘雯君
	审　　核：	熊伟
	完成日期：	2023-11-30

更新记录

版本	修改人	修改日期	修改说明	核定人
v1.3.0	NPU 团队	2022-03-06	更新 RK3588 OP 列表，增加 CPU OP List	熊伟
v1.3.1	NPU 团队	2022-03-26	增加首层输入说明列表	熊伟
v1.3.2	NPU 团队	2022-04-21	更新 RV1103/1106 OP 支持列表	熊伟
v1.4.0	NPU 团队	2022-09-02	1.新增 RK3588 多核协同 2.更新 LSTM、transpose、softmax 等 OP 支持情况 3.新增 Conv-Add/Add-ReLu/Mul-ReLu Fuse OP 支持情况	熊伟
v1.4.1	NPU 团队	2022-12-05	1.新增 Conv-Add-Relu Fuse OP 支持情况 2.新增输出接口的 tensor 和 layout 说行	熊伟
v1.4.1b20	NPU 团队 /HPC 团队	2023-01-12	1.更新 RK3588 首层输入宽的限制 2.更新 RV1106 Conv-Add-Relu Fuse OP 支持情况 3.更新 RK3588/RV1106 Transpose 限制	熊伟
v1.4.2	NPU 团队	2023-02-13	1.新增 RK3562 OP 支持列表 2.修复部分描述错误	熊伟
v1.5.0	NPU 团队	2023-05-22	1.新增部分 CPU OP 支持项 2.对所有平台新增 add/mul 更多广播支持项 更新输入大分辨率规格支持	熊伟

版本	修改人	修改日期	修改说明	核定人
v1.5.1	NPU 团队	2023-06-06	新增部分 CPU OP 支持项	熊伟
v1.5.2	NPU 团队	2023-08-04	1. 新增 RK3562 exSoftmaxMask OP 支持项 2. 新增 where OP 支持项 3. 新增 MatMul GPU OP 支持项 4. 新增 exGlu OP 支持	熊伟
V1.6.0	NPU 团队	2023-11-30	1. 新增卷积类 OP 输入宽度限制 2. 新增 MatMul CPU/GPU 非对称维度计算支持 3. 新增 RK3588/RV1106 Softmax Op 支持项 4. 新增 RK3588/RV1106 exSoftmaxMask Op 支持项 5. 更新 Conv-Sigmoid/Tanh/Softplus/HardSigmoid HardSwish/Swish/Mish Fuse OP 支持情况 6. 更新 LayerNorm Op 支持情况 7. 新增 RK3562 GRU Op 支持项 8. 更新 Min/Max Op 规格约束 9. 更新部分 ONNX 规范参数 10. 新增 Erf 等 CPU 算子的支持	熊伟

目 录

第一章 RK3566/3568 NPU OP 支持列表

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Add/Bias	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持 ONNX 规范的四维 tensor 的所有广播操作，详见：注释（1）	per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Sub	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明,支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。例子见：注释（1）	per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Mul/Scale	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持 ONNX 规范的四维 tensor 的所有广播操作，详见：注释（1）	per-layer/ per-channel
Div	部分支持	float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明，支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP((N,C,H,W),scalar)，即以单个标量做 broadcasting； 4.OP(A(N,C,H,W),B(H,W))，即 HW 维度做 broadcasting，目前仅支持 FP16 类型。 说明：A 或 B 都可以作为广播方。例子见：注释（1）	per-layer/ per-channel

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Max	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明，支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))， 即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。例子见：注释（1）	per-layer/ per-channel
Min	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明,支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))， 即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。例子见：注释（1）	per-layer/ per-channel

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Global AveragePool	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		
				width/ 输入的 width	[1,7] (Complier 支持范围)		
GlobalMaxPool	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		
				width/ 输入的 width	[1,7] (Complier 支持范围)		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
AveragePool	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width			
			auto_pad:string	auto_pad/ pad 的方式	仅支持 NOTSET		
			ceil_mode:int64	ceil_mode/ 使用 ceil 或 floor 的方式计算输出的 shape	不支持		
			count_include_pad:int64	count_include_pad/ 是否包含 pad 数值进行计算	1		
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小	无限制，NPU 支持 [1,7]；其它由 CPU 支持。		
				kernel_w/ width 方向的 kernel 大小			
			pads[pads_top,pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,7]		
				pads_right/ right 方向的 pads 大小			
				pads_top/ top 方向的 pads 大小			
				pads_bottom/ bottom 方向的 pads 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,8]		
				stride_w/ width 方向的 strides 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
MaxPool	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width			
			auto_pad:string	auto_pad/ pad 的方式	仅支持 NOTSET		
			ceil_mode:int64	ceil_mode/ 使用 ceil 或 floor 的方式计算输出的 shape	不支持		
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	1		
				dilations_w/ widtht 方向的 dilations 大小			
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小	无限制，NPU 支持 [1,7]；其它由 CPU 支持。		
				kernel_w/ width 方向的 kernel 大小			
			pads[pads_top,pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,7]		
				pads_right/ right 方向的 pads 大小			
				pads_top/ top 方向的 pads 大小			
				pads_bottom/ bottom 方向的 pads 大小			
			storage_order：int64	storage_order/优先储存方式	0		
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,8]		
				stride_w/ width 方向的 strides 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Batch Normalization	支持	int8 float16	epsilon:double	epsilon/ 除以标准差时加上防止除 0 的实数	非 0 实数，参考值为 1e-5		per-layer/ per-channel
			momentum:double	momentum/ 训练时的滑动平均参数	无限制		
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
Layer Normalization	部分支持	float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	支持多 batch		per-layer
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
			layernorm_weight [channel,height, width]:tensor(const) layernorm_bias [channel,height, width]:tensor(const)	channel/ 输入的 channel	等于 input_channel		
				height/ 输入的 height	等于 input_height		
				width/ 输入的 width	等于 input_width		
			normalized_shape:int64[]	normalized shape /参与每一批归一化的 Feature 的尺寸	NPU 仅支持，包含除第 0 维（batch 维）以外的其他所有维度，如 input_shape[n,c,h,w]，仅支持 normalized_shape[c,h,w]，如 input_shape[n,c,h]，仅支持 normalized_shape[c,h]，如 input_shape[n,c]，仅支持 normalized_shape[c]，其余情况会转到 CPU 执行。		
			elementwise_affine:int64	elementwise affine/ 是否具有可学习数	0 或 1 （默认为 0）。 当为 1 时拥有 LayerNorm.weight 与 LayerNorm.bias,仅支持 weight/bias 的尺寸：elementwise_shape 与 normalized_shape 一致；当为 0 时 LayerNorm.weight 为全 1 值，LayerNorm.bias 为全 0 值。		
			eps:double	eps/ 防止除法溢出的偏移参数	无限制		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Clip/ReLU6	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Elu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Gelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Relu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
LeakyRelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
PRelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
				slope/ PRelu 系数	仅支持单个标量或 C 维度系数		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
GRU	部分支持 GRU 扩展以及变体命名为 exGRU 算子，参数项中指明（extern）的项为 exGRU 独有的参数项。	float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				sequence/ 输入的 sequence	无限制，建议 4 对齐		
				input_size/ 输入的 input_size	无限制，建议 8 对齐		
			direction:string	direction/ 指定 GRU 的运算方向	forward：指定 GRU 的运算方向为前向 reverse：指定 GRU 的运算方向为反向 bidirectional：指定 GRU 的运算方向为双向		
			layout	输入输出数据的排列方式	0：输入 shape 为： [seq_length, batch_size, input_size] 输出 shape 为： [seq_length, num_directions, batch_size, hidden_size] 1：输入 shape 为: [batch_size, seq_length, input_size] 输出 shape 为: [batch_size, seq_length, num_directions, hidden_size]		
			batch_size:int64（extern）	batch_size/ 指定 GRU 输入的 batchsize	1		
			sequence_size :int64（extern）	sequence_size/ 指定 GRU 输入的 seqsize	无限制，建议 4 对齐		
			hidden size:int64（extern）	hidden size/ GRU 单元中的 hiddensize	无限制，建议 8 对齐		
			linear_before_ reset:int64	linear_before_ reset/ LBR 变种的选择	1(T) or 0(F)		
			input layout:string（extern）	input_layout/指定与对应输入 shape 含义一致的 layout	1.snc：指定 layout 对应的输入 shape 为[seqs, batches, input_size] 2.(sn)c：指定 layout 对应的输入 shape 为[seqs*batches, input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。		
			output layout:string（extern）	output_layout/指定与对应输出 shape 含义一致的 layout	1.sbnc：指定 layout 对应的输出 shape 为[seqs,directions,batches, hidden_size] 2.(sn)c：指定 layout 对应的输出 shape 为[seqs*batches, directions*input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。directions>1 时仅支持 batches=1。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
LSTM	部分支持 LSTM 扩展以及变体命名为 exLSTM 算子，参数项中指明（extern）的项为 exLSTM 独有的参数项。	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	sss		per-layer/ per-channel
				sequence/ 输入的 sequence	无限制，建议 4 对齐		
				input_size/ 输入的 input_size	无限制，建议 8 对齐		
			direction:string	direction/ 指定 LSTM 的运算方向	forward：指定 LSTM 的运算方向为前向 reverse：指定 LSTM 的运算方向为反向 bidirectional：指定 LSTM 的运算方向为双向		
			batch_size:int64（extern）	batch_size/ 指定 LSTM 输入的 batchsize	大于 1 时仅支持 4 的倍数		
			sequence_size :int64（extern）	sequence_size/ 指定 LSTM 输入的 seqsize	无限制，建议 4 对齐		
			hidden size:int64（extern）	hidden size/ LSTM 单元中的 hiddensize	无限制，建议 8 对齐		
			proj_size:int64（extern）	proj_size/ LSTM 单元存在 projection 时的 proj_size	0<=proj_size<=hiddensize 目前限定 0，即尚不支持 projection 功能		
			input_forget:int64	input_forget/ cifg 变种的选择	1(T) or 0(F) 目前限定 0，即尚不支持		
			has dropout:int64（extern）	has_dropout/ caffe 框架下的 indicator 功能的选择	1(T) or 0(F) Caffe 框架下，启用该功能要求输入 indicator，工具端自动配置，无需手动配置。		
			has projection:int64（extern）	has_projection/ projection 变种	1(T) or 0(F) 目前限定 0，即尚不支持		
			input_layout:string（extern）	input_layout/指定与对应输入 shape 含义一致的 layout	1.snc：指定 layout 对应的输入 shape 为[seqs, batches, input_size] 2.(sn)c：指定 layout 对应的输入 shape 为[seqs*batches, input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。		
			output layout:string（extern）	output_layout/指定与对应输出 shape 含义一致的 layout	1.sbnc：指定 layout 对应的输出 shape 为[seqs,directions,batches, hidden_size] 2.(sn)c：指定 layout 对应的输出 shape 为[seqs*batches, directions*input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。directions>1 时仅支持 batches=1。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Concat	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	channel 方向 concat 时，除了最后一个输入外，其他输入的 channel 大小需要对齐。对齐量：8bit 数据：8 对齐，16bit 数据：4 对齐 其他方向 concat 无限制。		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axis:int64	aixs/ 拼接的维度	无限制		
Mish	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Pad	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		
				hannel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width	[1,8176]		
		int64	pads:tensor	[n_begin,c_begin,h_be gin,w_begin,n_end,c_e nd,h_end,w_end]/ 输入各轴上 前后插入的 pad 大小	目前仅支持: n_begin, c_begin, n_end, c_end 为 1, h_begin, w_begin, h_end, w_end 无限制		
		float	constant_value:tensor	constant_value/ 填充入 pad 的值	无限制		
		string	mode:string	mode/pad 模式	仅支持 constant		
ReduceMean	尚不支持 目 前由 CPU 实现	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axes:int64[]	axes/ 指定 reduce 的轴	单轴:无限制，多轴:{2,3}		
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
ReduceSum	尚不支持 目前由 CPU 实现	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axes:int64[]	axes/ 指定 reduce 的轴	单轴：无限制，多轴:{2,3}		
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0		
Resize	部分支持 目前 NPU 仅支持宽高方向不超过 8 倍的整倍数的最近邻和线性插值缩放，其余不支持部分的会 Fallback 到 CPU 上实现。	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	支持多 batch		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width	1.[1,8176] 2.设放大倍数为 s（s 为正整数），width*s*(s-1)<=8192		
			mode:string	mode/resize 采用的模式	仅支持 nearest、linear		
			scales:int64[]	scales/尺寸放大倍数	仅支持 1-8 整数倍		
			roi:int64[]	roi/进行 resize 的输入范围	仅支持全局([0,0,0,0,1,1,1,1])		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Reshape	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width	[1,8176]		
		int64	Shape (batch_o,channel_o, height_o,width_o):tensor	batch_o/ 输出的 batch_o	无限制		
				channel_o/ 输出的 channel	[1,8192]		
				height_o/ 输出的 height			
				width_o/ 输出的 width	[1,8176]		
				[n,c,h1,w1]- >[n,c,h2,w2]/ (h1*w1=h2*w2)	支持		
				[1,c,h,w]- >[c1,hw1,1,1]/ (c1=c/a, h*w=hw1/a, a 为整数)	不支持		
				[n,c,1,1]->[1,n1,h,w]/ (c=h*w/a, n1=n/a, a 为整数)			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Reverse Sequence	尚不支持 目前由 CPU 实现	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			batch_axis:int64	batch_axis/ 指定是否为 batch 维度	1		
			time_axis:int64	time_axis/ 指定是否为 time 维度	0		
			sequence_lens:int64[]	sequence_lens/ 指定序列翻转的数量	仅支持 channel 数		
Sigmoid	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
HardSigmoid	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		per-layer
Swish	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		
HardSwish	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		per-layer

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Softplus	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Softmax	尚不支持，目前由 CPU 实现	float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axis:int64	axis/ 做 softmax 的轴			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Slice	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			starts:int64[]	start/ 切分的起始位置	channel 方向 Slice 时，channel_start 要对齐。对齐量： 8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。		
			ends:int64[]	ends/ 切分的终止位置	channel 方向 Slice 时，channel_end 要对齐。对齐量： 8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。		
			axes:int64[]	axes/ 选取切分的轴	支持任意 0~3 轴， 支持同时多轴选择		
			steps:int64[]	steps/ 选取切分对应轴的步长	1		
Split	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axis:int64	axis/ 切分的维度			
			num_outputs:int64	split 成几个输出			
			split:int64[]	spilt/ 指定切分后维度的长度	channel 方向 Split 时，除了最后一个输出外，其他输出的 channel 需要对齐。 对齐量： 8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Tanh	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Transpose	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	[1,1024]		
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width	[1,8176]		
			perm:int64[]	axis order/ 转置的轴顺序	仅支持 (1) perm=[3,1,2,0],in_shape=[n,c,1,1],且 n,c 要求 8bit 数据：8 对齐，16bit 数据：4 对齐。 (2) perm=[3,1,2,0],in_shape=[1,c,1,w],且 w,c 要求 8bit 数据：8 对齐，16bit 数据：4 对齐。 (3) perm=[2,1,0,3],in_shape=[n,c,1,1],且 n,c 要求 8bit 数据：8 对齐，16bit 数据：4 对齐。 (4) perm=[2,1,0,3],in_shape=[1,c,h,1],且 h,c 要求 8bit 数据：8 对齐，16bit 数据：4 对齐。 (5) perm=[0,3,1,2],in_shape=[n,c,h,w],且 w 要求 8bit 数据：8 对齐，16bit 数据：4 对齐，并且 c*h<8192。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Convolution	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width	当 dilation_kernel_h > 1 时, width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明		
			kernel_shape [num_output, num input,kernel h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制		
				num input/ 输入的 channel			
				kernel_h/ height 方向的 kernel 大小	[1,31]		
				kernel_w/ width 方向的 kernel 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,7]		
				stride_w/ width 方向的 strides 大小			
			pads[pads top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,15]		
				pads_right/ right 方向的 pads 大小			
				pads top/ top 方向的 pads 大小			
				pads bottom/ bottom 方向的 pads 大小			
			group:int64	group/ group 的大小	无限制		
			dilations[dilations h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]		
				dilations_w/ widtht 方向的 dilations 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Depthwise Convolution	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width	当 dilation_kernel_h > 1 时, width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明		
			kernel_shape [num_output, num_input,kernel_h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制		
				num_input/ 输入的 channel			
				kernel_h/ height 方向的 kernel 大小	[1,8]		
				kernel_w/ width 方向的 kernel 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,7]		
				stride_w/ width 方向的 strides 大小			
			pads[pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,15]		
				pads_right/ right 方向的 pads 大小			
				pads_top/ top 方向的 pads 大小			
				pads_bottom/ bottom 方向的 pads 大小			
			dilations[dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]		
				dilations_w/ widtht 方向的 dilations 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
ConvTranspose/ Deconvolution	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width	当 dilation_kernel_h > 1 时，width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明		
			kernel_shape [num_output, num input,kernel h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制		
				num input/ 输入的 channel			
				kernel_h/ height 方向的 kernel 大小	[1,31]		
				kernel_w/ width 方向的 kernel 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	{1,2,4,8}		
				stride_w/ width 方向的 strides 大小			
			pads[pads top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	支持 0-15 设置 pad 时注意: 不支持 kernel_h * dilations_h - dilations_h - pads_top < 0 不支持 kernel_w * dilations_w - dilations_w - pads_left < 0 不支持 stride_h *(height - 1) - pads_top + 1 < output_h 不支持 stride_w *(width - 1) - pads_left + 1 < output_w		
				pads_right/ right 方向的 pads 大小			
				pads top/ top 方向的 pads 大小			
				pads bottom/ bottom 方向的 pads 大小			
			group:int64	group/ group 的大小	1 当且仅当 num_input=num_output 时，支持 num_output		
			dilations[dilations h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]		
				dilations_w/ widtht 方向的 dilations 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Gemm	不支持，由 CPU 实现	int8 float16	input_tensor_1 [M, K]:tensor	M,K,N/ 输入数据的形状	不支持		per-layer/ per-channel
			input_tensor_2 [K,N]:tensor				
			alpha:double	alpha/ 矩阵 A*B 乘法的 scale			
			beta:double	beta/ 输入 C 矩阵的 scale			
			transA:int64	transA/ A 矩阵是否转置			
			transB:int64	transB/ B 矩阵是否转置			
MatMul	不支持，由 CPU 实现	int8 float16	input_tensor_1 [batch, K，C]:tensor	batch/ 输入的 batch	不支持		per-layer/ per-channel
				K/ 输入的 K			
			input_tensor_2 [batch, C, H]:tensor	C/ 输入的 C			
				H/ 输入的 H			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Expand	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int64	shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o	无限制		
				channel_o/ 输出的 channel			
				height_o/ 输出的 height			
				width_o/ 输出的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Where	支持	int8 float16 int64	x_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int8 float16 int64	y_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		bool	mask_tensor[batch, channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int8 float16	shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o	无限制		
				channel_o/ 输出的 channel			
				height_o/ 输出的 height			
				width_o/ 输出的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
exGlu	支持	int8 float16	x_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	c * h * w 满足如下限制 8bit 数据：8 对齐，16bit 数据：4 对齐		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int64	axis:int64	axis/ 切分的维度	axis ==1		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Convolution + Relu	支持	同 Convolution					
Convolution + Clip	支持						
Convolution + PRelu/LeakyRelu	支持						
Convolution + Add	支持						
Convolution + Mul	尚不支持						
Convolution +Sigmoid	支持						
Convolution + Tanh	支持						
Convolution + Softplus	支持						
Convolution + HardSigmoid	支持						
Convolution + HardSwish	支持						
Convolution + Elu	支持						
Convolution + Swish	支持						
Convolution + Mish	支持						

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
ConvTranspose + Relu	尚不支持	同 ConvTranspose					
ConvTranspose + Clip	尚不支持						
ConvTranspose + PRelu/LeakyRelu	尚不支持						
ConvTranspose + Add	尚不支持						
ConvTranspose + Mul	尚不支持						
ConvTranspose + Sigmoid	尚不支持						
ConvTranspose + Tanh	尚不支持						
ConvTranspose + Softplus	尚不支持						
ConvTranspose + HardSigmoid	尚不支持						
ConvTranspose + HardSwish	尚不支持						
ConvTranspose + Elu	尚不支持						
ConvTranspose + Swish	尚不支持						
ConvTranspose + Mish	尚不支持						

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Depthwise Convolution + Relu	尚不支持	同 Depthwise Convolution					
Depthwise Convolution + Clip	尚不支持						
Depthwise Convolution + PRelu/LeakyRelu	尚不支持						
Depthwise Convolution + Add	尚不支持						
Depthwise Convolution + Mul	尚不支持						
Depthwise Convolution + Sigmoid	尚不支持						
Depthwise Convolution + Tanh	尚不支持						
Depthwise Convolution + Softplus	尚不支持						
Depthwise Convolution + HardSigmoid	尚不支持						
Depthwise Convolution + HardSwish	尚不支持						
Depthwise Convolution + Elu	尚不支持						
Depthwise Convolutione + Swish	尚不支持						
Depthwise Convolution + Mish	尚不支持						

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Add+Relu	支持	同 Add					
Mul+Relu	支持	同 Mul					
Convolution + add + Relu	支持	同 Convolution					

注释：

(1) 广播说明：

以 ONNX 默认排列 NCHW 做说明，包含以下广播方式（A 或 B 都可以作为广播方）：

1. OP(A(N,C,H,W), B(N,C,H,W))，即两个维度相同的 tensor 进行操作；
2. OP(A(N,C,H,W), B(C,1,1))，即 C 维度做 broadcasting；
3. OP(A(N,C,H,W), B(scalar))，即以单个标量做 broadcasting；
4. OP(A(N,C,H,W), B(H,W))，即 HW 维度做 broadcasting。

广播支持举例：

1. OP(A(N,C,H,W), B(N,C,H,W)): OP(A(1,16,32,8), B(1,16,32,8)) = C(1,16,32,8)
2. OP(A(N,C,H,W), B(C,1,1)): OP(A(1,16,32 8), B(16)) = C(1,16,32,8)
3. OP(A(N,C,H,W), B(scalar)): OP(A(1,16,32,8), B(1)) = C(1,16,32,8)
4. OP(A(N,C,H,W), B(H,W)): OP(A(1,16,32,8), B(32x8)) = C(1,16,32,8)

约束规格中，[a,b]表示支持 a 到 b 之间的整数；{a,b,c}表示支持 a,b,c。

第二章 RK3588 NPU OP 支持列表

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Add/Bias	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持 ONNX 规范的四维 tensor 的所有广播操作，详见：注释（1）	per-layer/ per-channel	已支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
Sub	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明,支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。 例子见：注释（1）	per-layer/ per-channel	尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Mul/Scale	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持 ONNX 规范的四维 tensor 的所有广播操作，详见：注释（1）	per-layer/ per-channel	尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
Div	部分支持	float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明，支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP((N,C,H,W),scalar)，即以单个标量做 broadcasting； 4.OP(A(N,C,H,W),B(H,W))，即 HW 维度做 broadcasting，目前仅支持 FP16 类型。 说明：A 或 B 都可以作为广播方。 例子见：注释（1）	per-layer/ per-channel	尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Max	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明，支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。 例子见：注释（1）	per-layer/ per-channel	尚不支持
Min	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明,支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。 例子见：注释（1）	per-layer/ per-channel	尚不支持

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Global AveragePool	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer	尚不支持
				channel/ 输入的 channel	[1,8192]			
				height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)			
				width/ 输入的 width	[1,7] (Complier 支持范围)			
GlobalMaxPool	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer	尚不支持
				channel/ 输入的 channel	[1,8192]			
				height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)			
				width/ 输入的 width	[1,7] (Complier 支持范围)			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
AveragePool	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer	尚不支持
				channel/ 输入的 channel	[1,8192]			
				height/ 输入的 height				
				width/ 输入的 width				
			auto_pad:string	auto_pad/ pad 的方式	仅支持 NOTSET			
			ceil_mode:int64	ceil_mode/ 使用 ceil 或 floor 的方式计算输出的 shape	不支持			
			count_include_pad:int64	count_include_pad/ 是否包含 pad 数值进行计算	1			
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小	无限制，NPU 支持[1,7]；其它由 CPU 支持。			
				kernel_w/ width 方向的 kernel 大小				
			pads[pads_top,pads_left, pads bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,7]			
				pads_right/ right 方向的 pads 大小				
				pads_top/ top 方向的 pads 大小				
				pads_bottom/ bottom 方向的 pads 大小				
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,8]			
				stride_w/ width 方向的 strides 大小				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
MaxPool	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer	尚不支持
				channel/ 输入的 channel	[1,8192]			
				height/ 输入的 height				
				width/ 输入的 width				
			auto_pad:string	auto_pad/ pad 的方式	仅支持 NOTSET			
			ceil_mode:int64	ceil_mode/ 使用 ceil 或 floor 的方式计算输出的 shape	不支持			
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	1			
				dilations_w/ widtht 方向的 dilations 大小				
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小	无限制，NPU 支持 [1,7]；其它由 CPU 支持。			
				kernel_w/ width 方向的 kernel 大小				
			pads[pads_top,pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,7]			
				pads_right/ right 方向的 pads 大小				
				pads_top/ top 方向的 pads 大小				
				pads_bottom/ bottom 方向的 pads 大小				
			storage_order：int64	storage_order/优先储存方式	0			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,8]			
				stride_w/ width 方向的 strides 大小				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Batch Normalization	支持	int8 float16	epsilon:double	epsilon/ 除以标准差时加上防止除 0 的实数	非 0 实数，参考值为 1e-5		per-layer/ per-channel	尚不支持
			momentum:double	momentum/ 训练时的滑动平均参数	无限制			
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1			
				channel/ 输入的 channel	无限制			
				height/ 输入的 height				
				width/ 输入的 width				
Layer Normalization	部分 支持	float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	支持多 batch		per-layer	尚不支持
				channel/ 输入的 channel	无限制			
				height/ 输入的 height				
				width/ 输入的 width				
			layernorm_weight [channel,height, width]:tensor(const) layernorm_bias [channel,height, width]:tensor(const)	channel/ 输入的 channel	等于 input_channel			
				height/ 输入的 height	等于 input_height			
				width/ 输入的 width	等于 input_width			
			normalized shape:int64[]	normalized_ shape /参与每一批归一化的 Feature 的尺寸	NPU 仅支持，包含除第 0 维（batch 维）以外的其他所有维度，如 input_shape[n,c,h,w], 仅支持 normalized_shape[c,h,w]，如 input_shape[n,c,h], 仅支持 normalized_shape[c,h]，如 input_shape[n,c], 仅支持 normalized_shape[c]，其余情况会转到 CPU 执行。			
			elementwise_affine:int64	elementwise_affine/ 是否具有可学习数	0 或 1（默认为 0）。 当为 1 时拥有 LayerNorm.weight 与 LayerNorm.bias,仅支持 weight/bias 的尺寸：elementwise_shape 与 normalized_shape 一致； 当为 0 时 LayerNorm.weight 为全 1 值，LayerNorm.bias 为全 0 值。			
			eps:double	eps/	无限制			

				防止除法溢出的偏移参数				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Clip/ReLU6	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		per-layer	已支持
Elu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制			尚不支持
Gelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制			尚不支持

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Relu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer	已支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
LeakyRelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer	已支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
PRelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel	已支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
				slope/ PRelu 系数	仅支持单个标量或 C 维度系数			

Operator	支持情况	输入类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
GRU	部分支持 GRU 扩展以及变体命名为 exGRU 算子，参数项中指明（extern）的项为 exGRU 独有的参数项。	float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer	尚不支持
				sequence/ 输入的 sequence	无限制，建议 4 对齐			
				input_size/ 输入的 input_size	无限制，建议 8 对齐			
			direction:string	direction/ 指定 GRU 的运算方向	forward：指定 GRU 的运算方向为前向 reverse：指定 GRU 的运算方向为反向 bidirectional：指定 GRU 的运算方向为双向			
			layout	输入输出数据的排列方式	0：输入 shape 为： [seq_length, batch_size, input_size] 输出 shape 为： [seq_length, num_directions, batch_size, hidden_size] 1：输入 shape 为: [batch_size, seq_length, input_size] 输出 shape 为: [batch_size, seq_length, num_directions, hidden_size]			
			batch_size:int64（extern）	batch_size/ 指定 GRU 输入的 batchsize	1			
			sequence_size :int64（extern）	sequence_size/ 指定 GRU 输入的 seqsize	无限制，建议 4 对齐			
			hidden_size:int64（extern）	hidden_size/ GRU 单元中的 hiddensize	无限制，建议 8 对齐			
			linear_before_ reset:int64	linear_before_ reset/ LBR 变种的选择	1(T) or 0(F)			
			input_layout:string（extern）	input_layout/指定与对应输入 shape 含义一致的 layout	1.snc：指定 layout 对应的输入 shape 为[seqs, batches, input_size] 2.(sn)c：指定 layout 对应的输入 shape 为[seqs*batches, input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。			
			output_layout:string（extern）	output_layout/指定与对应输出 shape 含义一致的 layout	1.sbnc：指定 layout 对应的输出 shape 为[seqs,directions,batches, hidden_size] 2.(sn)c：指定 layout 对应的输出 shape 为[seqs*batches, directions*input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。directions>1 时仅支持 batches=1。			

Operator	支持情况	输入类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
LSTM	部分支持 LSTM 扩展以及变体命名为 exLSTM 算子，参数项中指明（extern）的项为 exLSTM 独有的参数项。	int8 float16	input tensor [batch,channel,height,width]:tensor	batch/ 输入的 batch	batch>1 时要求 batch=4n,（n 为正整数），建议 n<=4。 注：LSTM 单向：无限制，LSTM 双向：不同时支持多 batch。		per-layer/ per channel	尚不支持
				sequence/ 输入的 sequence	无限制，建议 4 对齐			
				input_size/ 输入的 input_size	无限制，建议 8 对齐			
			direction:string	direction/ 指定 LSTM 的运算方向	forward：指定 LSTM 的运算方向为前向 reverse：指定 LSTM 的运算方向为反向 bidirectional：指定 LSTM 的运算方向为双向			
			batch size:int64（extern）	batch_size/ 指定 LSTM 输入的 batchsize	大于 1 时仅支持 4 的倍数			
			sequence size :int64（extern）	sequence_size/ 指定 LSTM 输入的 seqsize	无限制，建议 4 对齐			
			hidden size:int64（extern）	hidden_size/ LSTM 单元中的 hiddensize	无限制，建议 8 对齐			
			proj_size:int64（extern）	proj_size/ LSTM 单元存在 projection 时的 proj_size	0<=proj_size<=hiddensize 目前限定 0，即尚不支持 projection 功能			
			input_forget:int64	input_forget/ cifg 变种的选择	1(T) or 0(F) 目前限定 0，即尚不支持			
			has dropout:int64（extern）	has_dropout/ caffe 框架下的 indicator 功能的选择	1(T) or 0(F) Caffe 框架下，启用该功能要求输入 indicator，工具端自动配置，无需手动配置。			
			has projection:int64（extern）	has_projection/ projection 变种	1(T) or 0(F) 目前限定 0，即尚不支持			
			input_layout:string（extern）	input_layout/指定与对应输入 shape 含义一致的 layout	1.snc：指定 layout 对应的输入 shape 为[seqs, batches, input_size] 2.(sn)c：指定 layout 对应的输入 shape 为[seqs*batches, input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。			
			output_layout:string（extern）	output_layout/指定与对应输出 shape 含义一致的 layout	1.sbnc：指定 layout 对应的输出 shape 为[seqs,directions,batches, hidden_size] 2.(sn)c：指定 layout 对应的输出 shape 为[seqs*batches, directions*input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。directions>1 时仅支持 batches=1。			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Concat	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	channel 方向 concat 时，除了最后一个输入外，其他输入的 channel 大小需要对齐。对齐量：8bit 数据：8 对齐，16bit 数据：4 对齐 其他方向 concat 无限制。		per-layer	已支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
			axis:int64	aixs/ 拼接的维度	无限制			
Mish	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制			尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Pad	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1			尚不支持
				hannel/ 输入的 channel	无限制			
				height/ 输入的 height				
				width/ 输入的 width	[1,8176]			
		int64	pads:tensor	[n_begin,c_begin,h_be gin,w_begin,n_end,c_e nd,h_end,w_end]/ 输入各轴上前后插入的 pad 大小	目前仅支持: n_begin, c_begin, n_end, c_end 为 1, h_begin, w_begin, h_end, w_end 无限制			
		float	constant_value:tensor	constant_value/ 填充入 pad 的值	无限制			
		string	mode:string	mode/pad 模式	仅支持 constant			
ReduceMean	尚不支持 目前由 CPU 实现	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制			尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
			axes:int64[]	axes/ 指定 reduce 的轴	单轴:无限制，多轴:{2,3}			
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
ReduceSum	尚不支持 目前由CPU实现	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel	尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
			axes:int64[]	axes/ 指定 reduce 的轴	单轴:无限制，多轴:{2,3}			
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0			
Resize	部分支持 目前NPU仅支持宽高方向不超过 8 倍的整倍数的最近邻和线性插值缩放，其余不支持部分的会Fallback 到CPU上实现。	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	支持多 batch		per-layer	尚不支持
				channel/ 输入的 channel	[1,8192]			
				height/ 输入的 height				
				width/ 输入的 width	1.[1,8176] 2.设放大倍数为 s（s 为正整数），width*s*(s-1)<=8192			
			mode:string	mode/resize 采用的模式	仅支持 nearest、linear			
			scales:int64[]	scales/尺寸放大倍数	仅支持 1-8 整数倍			
			roi:int64[]	roi/进行 resize 的输入范围	仅支持全局([0,0,0,0,1,1,1,1])			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Reshape	部分支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制			尚不支持
				channel/ 输入的 channel	[1,8192]			
				height/ 输入的 height				
				width/ 输入的 width	[1,8176]			
		int64	Shape (batch_o,channel_o, height_o,width_o):tensor	batch_o/ 输出的 batch_o	无限制			
				channel_o/ 输出的 channel	[1,8192]			
				height_o/ 输出的 height				
				width_o/ 输出的 width	[1,8176]			
				[n,c,h1,w1]->[n,c,h2,w2]/ (h1*w1=h2*w2)	支持			
				[1,c,h,w]->[c1,hw1,1,1]/ (c1=c/a, h*w=hw1/a, a 为整数)	不支持			
				[n,c,1,1]->[1,n1,h,w]/ (c=h*w/a, n1=n/a, a 为整数)				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Reverse Sequence	尚不支持 目前由 CPU 实现	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制			尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
			batch_axis:int64	batch_axis/ 指定是否为 batch 维度	1			
			time_axis:int64	time_axis/ 指定是否为 time 维度	0			
			sequence_lens:int64[]	sequence_lens/ 指定序列翻转的数量	仅支持 channel 数			
Sigmoid	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制			尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
HardSigmoid	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		per-layer	尚不支持
Swish	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制			尚不支持
HardSwish	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		per-layer	尚不支持

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Softplus	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer	尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
Softmax	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer	尚不支持
				channel/ 输入的 channel	硬件支持[1,8192]			
				height/ 输入的 height	axis=1，无限制 axis=3/-1,width[1, 8192], height 无限制 且受限于 tranpose 的 规格限制			
				width/ 输入的 width				
			axis:int64	axis/ 做 softmax 的轴	1,3，即 channel 和 width 方向			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Slice	部分支持	int8 float16	input_tensor [batch,channel,height,width]:tensor	batch/ 输入的 batch	无限制		per-layer	尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
			starts:int64[]	start/ 切分的起始位置	channel 方向 Slice 时，channel_start 要对齐。对齐量： 8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。			
			ends:int64[]	ends/ 切分的终止位置	channel 方向 Slice 时，channel_end 要对齐。对齐量： 8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。			
			axes:int64[]	axes/ 选取切分的轴	支持任意 0~3 轴， 支持同时多轴选择			
			steps:int64[]	steps/ 选取切分对应轴的步长	1			
Split	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer	尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
			axis:int64	axis/ 切分的维度				
			num_outputs:int64	split 成几个输出				
			split:int64[]	spilt/ 指定切分后维度的长度	channel 方向 Split 时，除了最后一个输出外，其他输出的 channel 需要对齐。 对齐量：8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Tanh	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer	尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width				
Transpose	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	[1,1024]			尚不支持
				channel/ 输入的 channel	[1,8192]			
				height/ 输入的 height				
				width/ 输入的 width	[1,8176]			
			perm:int64[]	axis order/ 转置的轴顺序	仅支持 (1) perm=[3,1,2,0],in_shape=[n,c,1,1],且 n,c 要求 8bit 数据：8 对齐，16bit 数据：4 对齐。 (2) perm=[3,1,2,0],in_shape=[1,c,1,w],且 w,c 要求 8bit 数据：8 对齐，16bit 数据：4 对齐。 (3) perm=[2,1,0,3],in_shape=[n,c,1,1],且 n,c 要求 8bit 数据：8 对齐，16bit 数据：4 对齐。 (4) perm=[2,1,0,3],in_shape=[1,c,h,1],且 h,c 要求 8bit 数据：8 对齐，16bit 数据：4 对齐。 (5) perm=[0,3,1,2],in_shape=[n,c,h,w],且 w 要求 8bit 数据：8 对齐，16bit 数据：4 对齐，并且 c*h<8192。			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Convolution	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel	已支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width	当 dilation_kernel_h > 1 时，width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明			
			kernel_shape [num_output, num_input,kernel_h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制			
				num input/ 输入的 channel				
				kernel_h/ height 方向的 kernel 大小	[1,31]			
				kernel_w/ width 方向的 kernel 大小				
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,7]			
				stride_w/ width 方向的 strides 大小				
			pads[pads_top,pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,15]			
				pads_right/ right 方向的 pads 大小				
				pads top/ top 方向的 pads 大小				
				pads bottom/ bottom 方向的 pads 大小				
			group:int64	group/ group 的大小	无限制			
			dilations[dilations h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]			
				dilations_w/ widtht 方向的 dilations 大小				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Depthwise Convolution	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel	已支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width	当 dilation_kernel_h > 1 时, width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明			
			kernel_shape [num_output, num_input,kernel_h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制			
				num_input/ 输入的 channel				
				kernel_h/ height 方向的 kernel 大小	[1,8]			
				kernel_w/ width 方向的 kernel 大小				
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,7]			
				stride_w/ width 方向的 strides 大小				
			pads[pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,15]			
				pads_right/ right 方向的 pads 大小				
				pads_top/ top 方向的 pads 大小				
				pads_bottom/ bottom 方向的 pads 大小				
			dilations[dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]			
				dilations_w/ widtht 方向的 dilations 大小				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
ConvTranspose/ Deconvolution	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel	尚不支持
				channel/ 输入的 channel				
				height/ 输入的 height				
				width/ 输入的 width	当 dilation_kernel_h > 1 时，width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明			
			kernel_shape [num_output, num_input,kernel_h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制			
				num_input/ 输入的 channel				
				kernel_h/ height 方向的 kernel 大小	[1,31]			
				kernel_w/ width 方向的 kernel 大小				
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	{1,2,4,8}			
				stride_w/ width 方向的 strides 大小				
			pads[pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	支持 0-15 设置 pad 时注意: 不支持 kernel_h * dilations_h - dilations_h - pads_top < 0 不支持 kernel_w * dilations_w - dilations_w - pads_left < 0 不支持 stride_h *(height - 1) - pads_top + 1 < output_h 不支持 stride_w *(width - 1) - pads_left + 1 < output_w			
				pads_right/ right 方向的 pads 大小				
				pads_top/ top 方向的 pads 大小				
				pads_bottom/ bottom 方向的 pads 大小				
			group:int64	group/ group 的大小	1 当且仅当 num_input=num_output 时，支持 num_output			
			dilations[dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]			
				dilations_w/ widtht 方向的 dilations 大小				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Gemm	尚不支持，目前由 CPU 实现	int8 float16	input_tensor_1 [M, K]:tensor	M,K,N/ 输入数据的形状	转为 Matmul 实现，约束同 Matmul		per-layer/ per-channel	尚不支持
			input_tensor_2 [K,N]:tensor					
			alpha:double	alpha/ 矩阵 A*B 乘法的 scale	无限制			
			beta:double	beta/ 输入 C 矩阵的 scale				
			transA:int64	transA/ A 矩阵是否转置	仅静态 tensor 支持转置			
			transB:int64	transB/ B 矩阵是否转置				
MatMul	部分支持 目前该支持仅针对双 feature 输入 未来将支持输入为 feature+ constan	int8 float16	input_tensor_1 [batch, K，C]:tensor	batch/ 输入的 batch	双 feature 时： batch、H 无限制 K 支持[8,8192]，对齐要求为 8bit 数据：16 对齐，16bit 数据：8 对齐 C 支持[32,19384]，对齐要求：32 对齐 K*C <=65532 C*H <=65532 K*H <=65532 feature+constant 时： 若 input_tensor_1 为 feature，则转为 batch 个 feature[K,C,1,1] + weight[H,C,1,1]的 conv； 若 input_tensor_2 为 feature，则转为 batch 个 feature[1,C,H,1] + weight[K,C,1,1]的 conv； C 对齐要求：32 对齐 其他约束和 conv 相同		per-layer/ per-channel	尚不支持
				K/ 输入的 K				
			input_tensor_2 [batch, C, H]:tensor	C/ 输入的 C				
				H/ 输入的 H				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
MatMul（4d）	部分支持 目前该支持仅针对双 feature 输入 未来将支持输入为 feature+ constant	int8 float16	input_tensor_1 [batch,channel,K, N]:tensor	batch/ 输入的 batch	双 feature 时： batch 无限制 channl、K 支持[8,8192]，对齐要求为 8bit 数据：16 对齐，16bit 数据： 8 对齐 N 支持[32,19384]，对齐要求：32 对齐 K*N <=65532 K*M <=65532 M*N <=65532 feature+constant 时： 若 input_tensor_1 为 feature，则转为 batch*channel 个 feature[K,N,1,1] + weight[M,N,1,1]的 conv； 若 input_tensor_2 为 feature，则转为 batch*channel 个 feature[1,N,M,1] + weight[K,N,1,1]的 conv； N 对齐要求：32 对齐其他约束和 conv 相同		per-layer/ per-channel	尚不支持
				batch/ 输入的 batch				
			input_tensor_2 [batch,channel,N, M]:tensor	K/ 输入的 K				
				N/ 输入的 M				
				M/ 输入的 M				

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Expand	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int64	shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o	无限制		
				channel_o/ 输出的 channel			
				height_o/ 输出的 height			
				width_o/ 输出的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Where	支持	int8 float16 int64	x_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int8 float16 int64	y_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		bool	mask_tensor[batch, channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int8 float16	shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o	无限制		
				channel_o/ 输出的 channel			
				height_o/ 输出的 height			
				width_o/ 输出的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
exSoftmaxMask	部分支持	int8 float16	input_tensor_1[batch, channel,height,width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel	硬件支持[1,8192]		
				height/ 输入的 height	axis=1，无限制 axis=3/-1,width[1, 8192], height 无限制 且受限于 transpose 规格限制		
				width/ 输入的 width			
			input_tensor_2[batch, channel,height,width]:tensor	batch/ 输入的 batch	1		
				channel/ 输入的 channel	硬件支持[1,8192]		
				height/ 输入的 height	1		
				width/ 输入的 width	axis=1: [1] axis=3/-1,[1, 8192]		
			axis:int64	axis/ 做 softmax 的轴	1,3，即 channel 和 width 方向		
			mask_value:int64	mask/ 需要 mask 的值	0 或 1		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
exGlu	支持	int8 float16	x_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	c * h * w 满足如下限制 8bit 数据：8 对齐，16bit 数据：4 对齐		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int64	axis:int64	axis/ 切分的维度	axis ==1		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Convolution + Relu	支持	同 Convolution						已支持
Convolution + Clip	支持							已支持
Convolution + PRelu/LeakyRelu	支持							已支持
Convolution + Add	支持							已支持
Convolution + Mul	尚不支持							尚不支持
Convolution +Sigmoid	支持							尚不支持
Convolution + Tanh	支持							尚不支持
Convolution + Softplus	支持							尚不支持
Convolution + HardSigmoid	支持							尚不支持
Convolution + HardSwish	支持							尚不支持
Convolution + Elu	支持							尚不支持
Convolution + Swish	支持							尚不支持
Convolution + Mish	支持							尚不支持

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
ConvTranspose + Relu	尚不支持	同 ConvTranspose						尚不支持
ConvTranspose + Clip	尚不支持							尚不支持
ConvTranspose + PRelu/LeakyRelu	尚不支持							尚不支持
ConvTranspose + Add	尚不支持							尚不支持
ConvTranspose + Mul	尚不支持							尚不支持
ConvTranspose + Sigmoid	尚不支持							尚不支持
ConvTranspose + Tanh	尚不支持							尚不支持
ConvTranspose + Softplus	尚不支持							尚不支持
ConvTranspose + HardSigmoid	尚不支持							尚不支持
ConvTranspose + HardSwish	尚不支持							尚不支持
ConvTranspose + Elu	尚不支持							尚不支持
ConvTranspose + Swish	尚不支持							尚不支持
ConvTranspose + Mish	尚不支持							尚不支持

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Depthwise Convolution + Relu	尚不支持	同 Depthwise Convolution						已支持
Depthwise Convolution + Clip	尚不支持							已支持
Depthwise Convolution + PRelu/LeakyRelu	尚不支持							已支持
Depthwise Convolution + Add	尚不支持							已支持
Depthwise Convolution + Mul	尚不支持							尚不支持
Depthwise Convolution + Sigmoid	尚不支持							尚不支持
Depthwise Convolution + Tanh	尚不支持							尚不支持
Depthwise Convolution + Softplus	尚不支持							尚不支持
Depthwise Convolution + HardSigmoid	尚不支持							尚不支持
Depthwise Convolution + HardSwish	尚不支持							尚不支持
Depthwise Convolution + Elu	尚不支持							尚不支持
Depthwise Convolutione + Swish	尚不支持							尚不支持
Depthwise Convolution + Mish	尚不支持							尚不支持

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
Add+Relu	支持	同 Add						尚不支持
Mul+Relu	支持	同 Mul						尚不支持
Convolution + add + Relu	支持	同 Convolution						尚不支持

注释：

(1) 广播说明：

以 ONNX 默认排列 NCHW 做说明，包含以下广播方式（A 或 B 都可以作为广播方）：

1. OP(A(N,C,H,W), B(N,C,H,W))，即两个维度相同的 tensor 进行操作；
2. OP(A(N,C,H,W), B(C,1,1))，即 C 维度做 broadcasting；
3. OP(A(N,C,H,W), B(scalar))，即以单个标量做 broadcasting；
4. OP(A(N,C,H,W), B(H,W))，即 HW 维度做 broadcasting。

广播支持举例：

1. OP(A(N,C,H,W), B(N,C,H,W)): OP(A(1,16,32,8), B(1,16,32,8)) = C(1,16,32,8)
2. OP(A(N,C,H,W), B(C,1,1)): OP(A(1,16,32 8), B(16)) = C(1,16,32,8)
3. OP(A(N,C,H,W), B(scalar)): OP(A(1,16,32,8), B(1)) = C(1,16,32,8)
4. OP(A(N,C,H,W), B(H,W)): OP(A(1,16,32,8), B(32x8)) = C(1,16,32,8)

约束规格中，[a,b]表示支持 a 到 b 之间的整数；{a,b,c}表示支持 a,b,c。

第三章 RV1103/1106 NPU OP 支持列表

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Add/Bias	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持 ONNX 规范的四维 tensor 的所有广播操作，详见：注释（1）	per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Sub	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明,支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。详见：注释（1）	per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Mul/Scale	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持 ONNX 规范的四维 tensor 的所有广播操作，详见：注释（1）	per-layer/ per-channel
Div	部分支持	float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明，支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP((N,C,H,W),scalar)，即以单个标量做 broadcasting； 4.OP(A(N,C,H,W),B(H,W))，即 HW 维度做 broadcasting，目前仅支持 FP16 类型。 说明：A 或 B 都可以作为广播方。例子见：注释（1）	per-layer/ per-channel

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Max	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明，支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。例子见：注释（1）	per-layer/ per-channel
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width	[1,8176]		
Min	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明,支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。例子见：注释（1）	per-layer/ per-channel
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width	[1,8176]		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Global AveragePool	支持	int8	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		
				width/ 输入的 width	[1,7] (Complier 支持范围)		
GlobalMaxPool	支持	int8	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		
				width/ 输入的 width	[1,7] (Complier 支持范围)		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
AveragePool	支持	int8	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width			
			auto_pad:string	auto_pad/ pad 的方式	仅支持 NOTSET		
			ceil_mode:int64	ceil_mode/ 使用 ceil 或 floor 的方式计算输出的 shape	不支持		
			count_include_pad:int64	count_include_pad/ 是否包含 pad 数值进行计算	1		
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小	无限制，NPU 支持 [1,]；其它由 CPU 支持。		
				kernel_w/ width 方向的 kernel 大小			
			pads[pads_top,pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,7]		
				pads_right/ right 方向的 pads 大小			
				pads_top/ top 方向的 pads 大小			
				pads_bottom/ bottom 方向的 pads 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,8]		
				stride_w/ width 方向的 strides 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
MaxPool	支持	int8	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width			
			auto_pad:string	auto_pad/ pad 的方式	仅支持 NOTSET		
			ceil_mode:int64	ceil_mode/ 使用 ceil 或 floor 的方式计算输出的 shape	不支持		
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	1		
				dilations_w/ widtht 方向的 dilations 大小			
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小	无限制，NPU 支持 [1,7]；其它由 CPU 支持。		
				kernel_w/ width 方向的 kernel 大小			
			pads[pads_top,pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,7]		
				pads_right/ right 方向的 pads 大小			
				pads_top/ top 方向的 pads 大小			
				pads_bottom/ bottom 方向的 pads 大小			
			storage_order：int64	storage_order/优先储存方式	0		
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,8]		
				stride_w/ width 方向的 strides 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Batch Normalization	支持	int8 float16	epsilon:double	epsilon/ 除以标准差时加上防止除 0 的实数	非 0 实数，参考值为 1e-5		per-layer/ per-channel
			momentum:double	momentum/ 训练时的滑动平均参数	无限制		
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
Layer Normalization	尚不支持	float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	支持多 batch		per-layer
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
			layernorm_weight [channel,height, width]:tensor(const) layernorm_bias [channel,height, width]:tensor(const)	channel/ 输入的 channel	等于 input_channel		
				height/ 输入的 height	等于 input_height		
				width/ 输入的 width	等于 input_width		
			normalized_shape:int64[]	normalized_ shape /参与每一批归一化的 Feature 的尺寸	NPU 仅支持，包含除第 0 维（batch 维）以外的其他所有维度，如 input_shape[n,c,h,w]，仅支持 normalized_shape[c,h,w]，如 input_shape[n,c,h]，仅支持 normalized_shape[c,h]，如 input_shape[n,c]，仅支持 normalized_shape[c]，其余情况会转到 CPU 执行。		
			elementwise_affine:int64	elementwise_affine/ 是否具有可学习数	0 或 1 （默认为 0）。 当为 1 时拥有 LayerNorm.weight 与 LayerNorm.bias,仅支持 weight/bias 的尺寸：elementwise_shape 与 normalized_shape 一致；当为 0 时 LayerNorm.weight 为全 1 值，LayerNorm.bias 为全 0 值。		
			eps:double	eps/ 防止除法溢出的偏移参数	无限制		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Clip/ReLU6	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
Elu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Gelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Relu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
LeakyRelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
PRelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer/ per-channel
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width	[1,8176]		
				slope/PRelu 系数	仅支持单个标量或 C 维度系数		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
GRU	尚不支持 GRU 扩展以及变体命名为 exGRU 算子，参数项中指明（extern）的项为 exGRU 独有的参数项。	float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				sequence/ 输入的 sequence	无限制，建议 4 对齐		
				input_size/ 输入的 input_size	无限制，建议 8 对齐		
			direction:string	direction/ 指定 GRU 的运算方向	forward：指定 GRU 的运算方向为前向 reverse：指定 GRU 的运算方向为反向 bidirectional：指定 GRU 的运算方向为双向		
			layout	输入输出数据的排列方式	0：输入 shape 为： [seq_length, batch_size, input_size] 输出 shape 为： [seq_length, num_directions, batch_size, hidden_size] 1：输入 shape 为: [batch_size, seq_length, input_size] 输出 shape 为: [batch_size, seq_length, num_directions, hidden_size]		
			batch_size:int64（extern）	batch size/ 指定 GRU 输入的 batchsize	1		
			sequence_size :int64（extern）	sequence_size/ 指定 GRU 输入的 seqsize	无限制，建议 4 对齐		
			hidden_size:int64（extern）	hidden_size/ GRU 单元中的 hiddensize	无限制，建议 8 对齐		
			linear_before_ reset:int64	linear_before_ reset/ LBR 变种的选择	1(T) or 0(F)		
			input_layout:string（extern）	input_layout/指定与对应输入 shape 含义一致的 layout	1.snc：指定 layout 对应的输入 shape 为[seqs, batches, input_size] 2.(sn)c：指定 layout 对应的输入 shape 为[seqs*batches, input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。		
			output_layout:string（extern）	output_layout/指定与对应输出 shape 含义一致的 layout	1.sbnc：指定 layout 对应的输出 shape 为[seqs,directions,batches, hidden_size] 2.(sn)c：指定 layout 对应的输出 shape 为[seqs*batches, directions*input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。directions>1 时仅支持 batches=1。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
LSTM	部分支持 LSTM 扩展以及变体命名为 exLSTM 算子，参数项中指明（extern）的项为 exLSTM 独有的参数项。	int8	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	batch>1 时要求 batch=4n,（n 为正整数），建议 n<=4。 注：LSTM 单向：无限制，LSTM 双向：不同时支持多 batch。		per-layer/ per-channel
				sequence/ 输入的 sequence	无限制，建议 4 对齐		
				input_size/ 输入的 input_size	无限制，建议 8 对齐		
			direction:string	direction/ 指定 LSTM 的运算方向	forward：指定 LSTM 的运算方向为前向 reverse：指定 LSTM 的运算方向为反向 bidirectional：指定 LSTM 的运算方向为双向		
			batch_size:int64（extern）	batch_size/ 指定 LSTM 输入的 batchsize	大于 1 时仅支持 4 的倍数		
			sequence_size :int64（extern）	sequence_size/ 指定 LSTM 输入的 seqsize	无限制，建议 4 对齐		
			hidden_size:int64（extern）	hidden_size/ LSTM 单元中的 hiddensize	无限制，建议 8 对齐		
			proj_size:int64（extern）	proj_size/ LSTM 单元存在 projection 时的 proj_size	0<=proj_size<=hiddensize 目前限定 0，即尚不支持 projection 功能		
			input_forget:int64	input_forget/ cifg 变种的选择	1(T) or 0(F) 目前限定 0，即尚不支持		
			has_dropout:int64（extern）	has_dropout/ caffe 框架下的 indicator 功能的选择	1(T) or 0(F) Caffe 框架下，启用该功能要求输入 indicator，工具端自动配置，无需手动配置。		
			has_projection:int64（extern）	has_projection/ projection 变种	1(T) or 0(F) 目前限定 0，即尚不支持		
			input_layout:string（extern）	input_layout/指定与对应输入 shape 含义一致的 layout	1.snc：指定 layout 对应的输入 shape 为[seqs, batches, input_size] 2.(sn)c：指定 layout 对应的输入 shape 为[seqs*batches, input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。		
			output_layout:string（extern）	output_layout/指定与对应输出 shape 含义一致的 layout	1.sbnc：指定 layout 对应的输出 shape 为[seqs,directions,batches, hidden_size] 2.(sn)c：指定 layout 对应的输出 shape 为[seqs*batches, directions*input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。directions>1 时仅支持 batches=1。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Concat	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	channel 方向 concat 时，除了最后一个输入外，其他输入的 channel 大小需要对齐。对齐量：8bit 数据：8 对齐，16bit 数据：4 对齐 其他方向 concat 无限制。		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axis:int64	aixs/ 拼接的维度	无限制		
Mish	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Pad	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		
				hannel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width	[1,8176]		
		int64	pads:tensor	[n_begin,c_begin,h_begin,w_begin,n_end,c_end,h_end,w_end]/ 输入各轴上前后插入的 pad 大小	目前仅支持: n_begin,c_begin,n_end,c_end 为 1,h_begin,w_begin,h_end,w_end 无限制		
		float	constant_value:tensor	constant_value/ 填充入 pad 的值	无限制		
		string	mode:string	mode/pad 模式	仅支持 constant		
ReduceMean	尚不支持 目前由 CPU 实现	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axes:int64[]	axes/ 指定 reduce 的轴	单轴:无限制，多轴:{2,3}		
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
ReduceSum	尚不支持 目前由 CPU 实现	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axes:int64[]	axes/ 指定 reduce 的轴	单轴：无限制，多轴:{2,3}		
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0		
Resize	部分支持 目前 NPU 仅支持宽高方向不超过 8 倍的整倍数的最近邻和线性插值缩放，其余不支持部分的会 Fallback 到 CPU 上实现。	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	支持多 batch		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width	1.[1,8176] 2.设放大倍数为 s（s 为正整数），width*s*(s-1)<=8192		
			mode:string	mode/resize 采用的模式	仅支持 nearest、linear		
			scales:int64[]	scales/尺寸放大倍数	仅支持 1-8 整数倍		
			roi:int64[]	roi/进行 resize 的输入范围	仅支持全局([0,0,0,0,1,1,1,1])		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Reshape	部分支持	int8 float16	input tensor [batch,channel,height, width]:tensor (input_tensor的维度为4维时看作 nchw)	batch/ 输入的 batch	约束规格： 1. height * width * type_bytes <= 130816 2. input_tensor 非四维时，shape 无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int64	Shape (batch_o,channel_o, height_o,width_o):tensor (输出 shape 指定维度为4 维时看作 nchw)	batch_o/ 输出的 batch_o	计 算 量 ： alignment=16/type_bytes； 约束规格： 1. height_o * width_o * type_bytes <= 65535； 2. Align(height_o * width_o, alignment) <= 8192； 3. 输出 shape 非四维时，shape 无限制		
				channel_o/ 输出的 channel			
				height_o/ 输出的 height			
				width_o/ 输出的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Reverse Sequence	尚不支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			batch_axis:int64	batch_axis/ 指定是否为 batch 维度	1		
			time_axis:int64	time_axis/ 指定是否为 time 维度	0		
			sequence_lens:int64[]	sequence_lens/ 指定序列翻转的数量	仅支持 channel 数		
Sigmoid	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
HardSigmoid	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		per-layer
Swish	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		
HardSwish	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		per-layer

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Softplus	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Softmax	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel	硬件支持[1,8192]		
				height/ 输入的 height	axis=1，无限制 axis=3/-1,width[1, 8192], height 无限制 且受限于 tranpose 的规格限制		
				width/ 输入的 width			
			axis:int64	axis/ 做 softmax 的轴	1,3，即 channel 和 width 方向		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Slice	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			starts:int64[]	start/ 切分的起始位置	channel 方向 Slice 时，channel_start 要对齐。对齐量： 8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。		
			ends:int64[]	ends/ 切分的终止位置	channel 方向 Slice 时，channel_end 要对齐。对齐量： 8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。		
			axes:int64[]	axes/ 选取切分的轴	支持任意 0~3 轴， 支持同时多轴选择		
			steps:int64[]	steps/ 选取切分对应轴的步长	1		
Split	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axis:int64	axis/ 切分的维度			
			num_outputs:int64	split 成几个输出			
			split:int64[]	spilt/ 指定切分后维度的长度	channel 方向 Split 时，除了最后一个输出外，其他输出的 channel 需要对齐。 对齐量：8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Tanh	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Transpose	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width	[1,8176]		
			perm:int64[]	axis order/ 转置的轴顺序	RV1106、RV1103 支持所有 RK3566/3568 上支持的 transpose 操作， 在该基础上支持：n 轴不参与转置时允许 c、h、w 三轴如下四种转置。 限制与说明如下： 1. 假设 in_shape[n1,c1,h1,w1],out_shape[n2,c2,h2,w2] 2. 四种转换分别为 (1) perm=[0,2,3,1]，NCHW->NHWC。 (2) perm=[0,2,1,3]，NCHW->NHCW。 (3) perm=[0,3,1,2]，NCHW->NWCH。 (4) perm=[0,3,2,1]，NCHW->NWHC。 3. 以上四种转置无对齐要求。但在满足对齐要求时效率更高。对齐要求为：第 1 点中参数的 c1、c2 均要满足 8bit 数据：16 对齐， 16bit 数据：8 对齐。 4. NPU 限制项： (1) perm=[0,2,3,1]时，8bit 数据时， h1*w1<8176，w1*c1<512；16bit 数据 时 ，h1*w1<8176， w1*c1<1023。 (2) perm=[0,3,1,2]时，h1*w1<8176。 (3) perm=[0,3,2,1]时，h1*w1<16384,h2*w2<8*8192, w1<1024。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Convolution	支持	int8	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width	当 dilation_kernel_h > 1 时, width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明		
			kernel_shape [num_output, num input,kernel h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制		
				num input/ 输入的 channel			
				kernel_h/ height 方向的 kernel 大小	[1,31]		
				kernel_w/ width 方向的 kernel 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,7]		
				stride_w/ width 方向的 strides 大小			
			pads[pads top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,15]		
				pads_right/ right 方向的 pads 大小			
				pads top/ top 方向的 pads 大小			
				pads bottom/ bottom 方向的 pads 大小			
			group:int64	group/ group 的大小	无限制		
			dilations[dilations h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]		
				dilations_w/ widtht 方向的 dilations 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Depthwise Convolution	支持	int8	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width	当 dilation_kernel_h > 1 时, width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明		
			kernel_shape [num_output, num_input,kernel_h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制		
				num_input/ 输入的 channel			
				kernel_h/ height 方向的 kernel 大小	[1,8]		
				kernel_w/ width 方向的 kernel 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,7]		
				stride_w/ width 方向的 strides 大小			
			pads[pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,15]		
				pads_right/ right 方向的 pads 大小			
				pads_top/ top 方向的 pads 大小			
				pads_bottom/ bottom 方向的 pads 大小			
			dilations[dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]		
				dilations_w/ widtht 方向的 dilations 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
ConvTranspose/ Deconvolution	支持	int8	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width	当 dilation_kernel_h > 1 时，width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明		
			kernel_shape [num_output, num input,kernel h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制		
				num input/ 输入的 channel			
				kernel_h/ height 方向的 kernel 大小	[1,31]		
				kernel_w/ width 方向的 kernel 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,8]		
				stride_w/ width 方向的 strides 大小			
			pads[pads top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	支持 0-15 设置 pad 时注意: 不支持 kernel_h * dilations_h - dilations_h - pads_top < 0 不支持 kernel_w * dilations_w - dilations_w - pads_left < 0 不支持 stride_h *(height - 1) - pads_top + 1 < output_h 不支持 stride_w *(width - 1) - pads_left + 1 < output_w		
				pads_right/ right 方向的 pads 大小			
				pads top/ top 方向的 pads 大小			
				pads bottom/ bottom 方向的 pads 大小			
			group:int64	group/ group 的大小	1 当且仅当 num_input=num_output 时，支持 num_output		
			dilations[dilations h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]		
				dilations_w/ widtht 方向的 dilations 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Gemm	尚不支持 目前由 CPU 实现	int8	input_tensor_1 [M, K]:tensor	M,K,N/ 输入数据的形状	转为 Matmul 实现，约束同 Matmul		per-layer/ per-channel
			input_tensor_2 [K,N]:tensor				
			alpha:double	alpha/ 矩阵 A*B 乘法的 scale	无限制		
			beta:double	beta/ 输入 C 矩阵的 scale			
			transA:int64	transA/ A 矩阵是否转置	仅静态 tensor 支持转置		
			transB:int64	transB/ B 矩阵是否转置			
MatMul	尚不支持 目前由 CPU 实现	int8	input_tensor_1 [batch, K，C]:tensor	batch/ 输入的 batch	双 feature 时： batch、H 无限制 K 支持[8,8192]，对齐要求为 8bit 数据：16 对齐，16bit 数据：8 对齐 C 支持[32,19384]，对齐要求为 32 对齐 feature+constant 时： 若 input_tensor_1 为 feature，则转为 batch 个 feature[K,C,1,1] + weight[H,C,1,1]的 conv； 若 input_tensor_2 为 feature，则转为 batch 个 feature[1,C,H,1] + weight[K,C,1,1]的 conv； C 对齐要求：32 对齐其他约束和 conv 相同		per-layer/ per-channel
				K/ 输入的 K			
			input_tensor_2 [batch, C, H]:tensor	C/ 输入的 C			
				H/ 输入的 H			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Expand	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int64	shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o	无限制		
				channel_o/ 输出的 channel			
				height_o/ 输出的 height			
				width_o/ 输出的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Where	支持	int8 int64	x_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int8 int64	y_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		bool	mask_tensor[batch, channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int8	shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o	无限制		
				channel_o/ 输出的 channel			
				height_o/ 输出的 height			
				width_o/ 输出的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
exSoftmaxMask	部分 支持	int8 float16	input_tensor_1[batch, channel,height,width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel	硬件支持[1,8192]		
				height/ 输入的 height	axis=1，无限制 axis=3/-1,width[1, 8192], height 无限制 且受限于 transpose 规格限制		
				width/ 输入的 width			
			input_tensor_2[batch, channel,height,width]:tensor	batch/ 输入的 batch	1		
				channel/ 输入的 channel	硬件支持[1,8192]		
				height/ 输入的 height	1		
				width/ 输入的 width	axis=1: [1] axis=3/-1,[1, 8192]		
			axis:int64	axis/ 做 softmax 的轴	1,3，即 channel 和 width 方向		
			mask_value:int64	mask/ 需要 mask 的值	0 或 1		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
exGlu	支持	int8 float16	x_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	c * h * w 满足如下限制 8bit 数据：8 对齐，16bit 数据：4 对齐		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int64	axis:int64	axis/ 切分的维度	axis ==1		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Convolution + Relu	支持	同 Convolution					
Convolution + Clip	支持						
Convolution + PRelu/LeakyRelu	支持						
Convolution + Add	支持						
Convolution + Mul	尚不支持						
Convolution +Sigmoid	支持						
Convolution + Tanh	支持						
Convolution + Softplus	支持						
Convolution + HardSigmoid	支持						
Convolution + HardSwish	支持						
Convolution + Elu	支持						
Convolution + Swish	支持						
Convolution + Mish	支持						

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
ConvTranspose + Relu	尚不支持	同 ConvTranspose					
ConvTranspose + Clip	尚不支持						
ConvTranspose + PRelu/LeakyRelu	尚不支持						
ConvTranspose + Add	尚不支持						
ConvTranspose + Mul	尚不支持						
ConvTranspose + Sigmoid	尚不支持						
ConvTranspose + Tanh	尚不支持						
ConvTranspose + Softplus	尚不支持						
ConvTranspose + HardSigmoid	尚不支持						
ConvTranspose + HardSwish	尚不支持						
ConvTranspose + Elu	尚不支持						
ConvTranspose + Swish	尚不支持						
ConvTranspose + Mish	尚不支持						

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Depthwise Convolution + Relu	尚不支持	同 Depthwise Convolution					
Depthwise Convolution + Clip	尚不支持						
Depthwise Convolution + PRelu/LeakyRelu	尚不支持						
Depthwise Convolution + Add	尚不支持						
Depthwise Convolution + Mul	尚不支持						
Depthwise Convolution + Sigmoid	尚不支持						
Depthwise Convolution + Tanh	尚不支持						
Depthwise Convolution + Softplus	尚不支持						
Depthwise Convolution + HardSigmoid	尚不支持						
Depthwise Convolution + HardSwish	尚不支持						
Depthwise Convolution + Elu	尚不支持						
Depthwise Convolutione + Swish	尚不支持						
Depthwise Convolution + Mish	尚不支持						

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Add+Relu	支持	同 Add					
Mul+Relu	支持	同 Mul					
Convolution + add + Relu	支持	同 Convolution					

注释：

(1) 广播说明：

以 ONNX 默认排列 NCHW 做说明，包含以下广播方式（A 或 B 都可以作为广播方）：

1. OP(A(N,C,H,W), B(N,C,H,W))，即两个维度相同的 tensor 进行操作；
2. OP(A(N,C,H,W), B(C,1,1))，即 C 维度做 broadcasting；
3. OP(A(N,C,H,W), B(scalar))，即以单个标量做 broadcasting；
4. OP(A(N,C,H,W), B(H,W))，即 HW 维度做 broadcasting。

广播支持举例：

1. OP(A(N,C,H,W), B(N,C,H,W)): OP(A(1,16,32,8), B(1,16,32,8)) = C(1,16,32,8)
2. OP(A(N,C,H,W), B(C,1,1)): OP(A(1,16,32 8), B(16)) = C(1,16,32,8)
3. OP(A(N,C,H,W), B(scalar)): OP(A(1,16,32,8), B(1)) = C(1,16,32,8)
4. OP(A(N,C,H,W), B(H,W)): OP(A(1,16,32,8), B(32x8)) = C(1,16,32,8)

约束规格中，[a,b]表示支持 a 到 b 之间的整数；{a,b,c}表示支持 a,b,c。

第四章 RK3562 NPU OP 支持列表

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Add/Bias	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持 ONNX 规范的四维 tensor 的所有广播操作，详见：注释（1）	per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Sub	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明,支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。例子见：注释（1）	per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Mul/Scale	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持 ONNX 规范的四维 tensor 的所有广播操作，详见：注释（1）	per-layer/ per-channel
Div	部分支持	float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明，支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP((N,C,H,W),scalar)，即以单个标量做 broadcasting； 4.OP(A(N,C,H,W),B(H,W))，即 HW 维度做 broadcasting，目前仅支持 FP16 类型。 说明：A 或 B 都可以作为广播方。例子见：注释（1）	per-layer/ per-channel

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Max	暂不支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明，支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。例子见：注释（1）	per-layer/ per-channel
Min	暂不支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制	支持两个 tensor 的广播操作，以 ONNX 默认排列 NCHW 做说明,支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的 tensor 进行操作； 2.OP(A(N,C,H,W),B(C,1,1))，即 C 维度做 broadcasting； 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做 broadcasting。 说明：A 或 B 都可以作为广播方。例子见：注释（1）	per-layer/ per-channel

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Global AveragePool	支持	int8	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		
				width/ 输入的 width	[1,7] (Complier 支持范围)		
GlobalMaxPool	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		
				width/ 输入的 width	[1,7] (Complier 支持范围)		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
AveragePool	支持	int8	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width			
			auto_pad:string	auto_pad/ pad 的方式	仅支持 NOTSET		
			ceil_mode:int64	ceil_mode/ 使用 ceil 或 floor 的方式计算输出的 shape	不支持		
			count_include_pad:int64	count_include_pad/ 是否包含 pad 数值进行计算	1		
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小	无限制，NPU 支持 [1,7]；其它由 CPU 支持。		
				kernel_w/ width 方向的 kernel 大小			
			pads[pads_top,pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,7]		
				pads_right/ right 方向的 pads 大小			
				pads_top/ top 方向的 pads 大小			
				pads_bottom/ bottom 方向的 pads 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,8]		
				stride_w/ width 方向的 strides 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
MaxPool	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width			
			auto_pad:string	auto_pad/ pad 的方式	仅支持 NOTSET		
			ceil_mode:int64	ceil_mode/ 使用 ceil 或 floor 的方式计算输出的 shape	不支持		
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	1		
				dilations_w/ widtht 方向的 dilations 大小			
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小	无限制，NPU 支持 [1,7]；其它由 CPU 支持。		
				kernel_w/ width 方向的 kernel 大小			
			pads[pads_top,pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,7]		
				pads_right/ right 方向的 pads 大小			
				pads_top/ top 方向的 pads 大小			
				pads_bottom/ bottom 方向的 pads 大小			
			storage_order：int64	storage_order/优先储存方式	0		
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,8]		
				stride_w/ width 方向的 strides 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Batch Normalization	支持	int8 float16	epsilon:double	epsilon/ 除以标准差时加上防止除 0 的实数	非 0 实数，参考值为 1e-5		per-layer/ per-channel
			momentum:double	momentum/ 训练时的滑动平均参数	无限制		
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
Layer Normalization	支持	float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	支持多 batch		per-layer
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
			layernorm_weight [channel,height, width]:tensor(const) layernorm_bias [channel,height, width]:tensor(const)	channel/ 输入的 channel	等于 input_channel		
				height/ 输入的 height	等于 input_height		
				width/ 输入的 width	等于 input_width		
			normalized_shape:int64[]	normalized_ shape /参与每一批归一化的 Feature 的尺寸	NPU 仅支持，包含除第 0 维（batch 维）以外的其他所有维度，如 input_shape[n,c,h,w]，仅支持 normalized_shape[c,h,w]，如 input_shape[n,c,h]，仅支持 normalized_shape[c,h]，如 input_shape[n,c]，仅支持 normalized_shape[c]，其余情况会转到 CPU 执行。		
			elementwise_affine:int64	elementwise affine/ 是否具有可学习数	0 或 1（默认为 0）。 当为 1 时拥有 LayerNorm.weight 与 LayerNorm.bias,仅支持 weight/bias 的尺寸：elementwise_shape 与 normalized_shape 一致；当为 0 时 LayerNorm.weight 为全 1 值，LayerNorm.bias 为全 0 值。		
			eps:double	eps/ 防止除法溢出的偏移参数	无限制		
			pre_norm:int64[]	pre_norm/ 预先 normaliz 可选项，防止 LN 溢出	无限制，当为 1 时硬件对输入做预先 normalize 处理：xi'=xi/max(|x|)。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Clip/ReLU6	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Elu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Gelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Relu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
LeakyRelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
PRelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer/ per-channel
				channel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width			
				slope/ PRelu 系数	仅支持单个标量或 C 维度系数		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
GRU	部分支持 GRU 扩展以及变体命名为 exGRU 算子，参数项中指明（extern）的项为 exGRU 独有的参数项。	float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		per-layer
				sequence/ 输入的 sequence	限制 4 对齐		
				input_size/ 输入的 input_size	限制 8 对齐		
			direction:string	direction/ 指定 GRU 的运算方向	forward：指定 GRU 的运算方向为前向 reverse：指定 GRU 的运算方向为反向 bidirectional：指定 GRU 的运算方向为双向		
			layout	输入输出数据的排列方式	0：输入 shape 为： [seq_length, batch_size, input_size] 输出 shape 为： [seq_length, num_directions, batch_size, hidden_size] 1：输入 shape 为: [batch_size, seq_length, input_size] 输出 shape 为: [batch_size, seq_length, num_directions, hidden_size]		
			batch_size:int64（extern）	batch_size/ 指定 GRU 输入的 batchsize	1		
			sequence_size :int64（extern）	sequence_size/ 指定 GRU 输入的 seqsize	限制 4 对齐		
			hidden size:int64 （extern）	hidden size/ GRU 单元中的 hiddensize	限制 8 对齐		
			linear_before_ reset:int64	linear before reset/ LBR 变种的选择	1(T) or 0(F)		
			input_layout:string （extern）	input_layout/指定与对应输入 shape 含义一致的 layout	1.snc：指定 layout 对应的输入 shape 为[seqs, batches, input_size] 2.(sn)c：指定 layout 对应的输入 shape 为[seqs*batches, input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。		
			output_layout:string （extern）	output_layout/指定与对应输出 shape 含义一致的 layout	1.sbnc：指定 layout 对应的输出 shape 为[seqs,directions,batches, hidden_size] 2.(sn)c：指定 layout 对应的输出 shape 为[seqs*batches, directions*input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。directions>1 时仅支持 batches=1。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
LSTM	部分支持 LSTM 扩展以及变体命名为 exLSTM 算子，参数项中指明（extern）的项为 exLSTM 独有的参数项。	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	batch>1 时要求 batch=4n,（n 为正整数），建议 n<=4。 注：LSTM 单向：无限制，LSTM 双向：不同时支持多 batch。		per-layer/ per-channel
				sequence/ 输入的 sequence	无限制，建议 4 对齐		
				input_size/ 输入的 input_size	无限制，建议 8 对齐		
			direction:string	direction/ 指定 LSTM 的运算方向	forward：指定 LSTM 的运算方向为前向 reverse：指定 LSTM 的运算方向为反向 bidirectional：指定 LSTM 的运算方向为双向		
			batch_size:int64（extern）	batch_size/ 指定 LSTM 输入的 batchsize	大于 1 时仅支持 4 的倍数		
			sequence size :int64（extern）	sequence size/ 指定 LSTM 输入的 seqsize	无限制，建议 4 对齐		
			hidden size:int64（extern）	hidden size/ LSTM 单元中的 hiddensize	无限制，建议 8 对齐		
			proj_size:int64（extern）	proj_size/ LSTM 单元存在 projection 时的 proj_size	0<=proj_size<=hiddensize 目前限定 0，即尚不支持 projection 功能		
			input_forget:int64	input_forget/ cifg 变种的选择	1(T) or 0(F) 目前限定 0，即尚不支持		
			has dropout:int64（extern）	has_dropout/ caffe 框架下的 indicator 功能的选择	1(T) or 0(F) Caffe 框架下，启用该功能要求输入 indicator，工具端自动配置，无需手动配置。		
			has projection:int64（extern）	has_projection/ projection 变种	1(T) or 0(F) 目前限定 0，即尚不支持		
			input_layout:string（extern）	input_layout/指定与对应输入 shape 含义一致的 layout	1.snc：指定 layout 对应的输入 shape 为[seqs, batches, input_size] 2.(sn)c：指定 layout 对应的输入 shape 为[seqs*batches, input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。		
			output_layout:string（extern）	output_layout/指定与对应输出 shape 含义一致的 layout	1.sbnc：指定 layout 对应的输出 shape 为[seqs,directions,batches, hidden_size] 2.(sn)c：指定 layout 对应的输出 shape 为[seqs*batches, directions*input_size,1,1] 要求填写指定的 layout，同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。directions>1 时仅支持 batches=1。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Concat	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	channel 方向 concat 时，除了最后一个输入外，其他输入的 channel 大小需要对齐。对齐量：8bit 数据：8 对齐，16bit 数据：4 对齐 其他方向 concat 无限制。		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axis:int64	aixs/ 拼接的维度	无限制		
Mish	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Pad	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1		
				hannel/ 输入的 channel	无限制		
				height/ 输入的 height			
				width/ 输入的 width	[1,8176]		
		int64	pads:tensor	[n_begin,c_begin,h_be gin,w_begin,n_end,c_e nd,h_end,w_end]/ 输入各轴上 前后插入的 pad 大小	目前仅支持: n_begin, c_begin, n_end, c_end 为 1, h_begin, w_begin, h_end, w_end 无限制		
		float	constant_value:tensor	constant_value/ 填充入 pad 的值	无限制		
		string	mode:string	mode/pad 模式	仅支持 constant		
ReduceMean	尚不支持 目 前由 CPU 实现	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axes:int64[]	axes/ 指定 reduce 的轴	单轴:无限制，多轴:{2,3}		
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
ReduceSum	尚不支持 目前由 CPU 实现	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axes:int64[]	axes/ 指定 reduce 的轴	单轴：无限制，多轴:{2,3}		
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0		
Resize	部分支持 目前 NPU 仅支持宽高方向不超过 8 倍的整倍数的最近邻和线性插值缩放，其余不支持部分的会 Fallback 到 CPU 上实现。	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	支持多 batch		per-layer
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width	1.[1,8176] 2.设放大倍数为 s（s 为正整数），width*s*(s-1)<=8192		
			mode:string	mode/resize 采用的模式	仅支持 nearest、linear		
			scales:int64[]	scales/尺寸放大倍数	仅支持 1-8 整数倍		
			roi:int64[]	roi/进行 resize 的输入范围	仅支持全局([0,0,0,0,1,1,1,1])		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Reshape	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	约束规格： 1. height * width * type_bytes <= 8192*8192*16； 2.input_tensor 非四维时，shape 无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int64	Shape (batch_o,channel_o, height_o,width_o):tensor	batch_o/ 输出的 batch_o	计 算 量 ： alignment=16/type_bytes； 约束规格： 1.height_o * width_o * type_bytes <= INT32_MAX； 2.Align(height_o * width_o, alignment) <= 8192*8192； 3.输出 shape 非四维时，shape 无限制		
				channel_o/ 输出的 channel			
				height_o/ 输出的 height			
				width_o/ 输出的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Reverse Sequence	尚不支持 目前由 CPU 实现	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			batch_axis:int64	batch_axis/ 指定是否为 batch 维度	1		
			time_axis:int64	time_axis/ 指定是否为 time 维度	0		
			sequence_lens:int64[]	sequence_lens/ 指定序列翻转的数量	仅支持 channel 数		
Sigmoid	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
HardSigmoid	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		per-layer
Swish	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		
HardSwish	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		per-layer

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Softplus	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Softmax	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel	硬件支持[1,8192]		
				height/ 输入的 height	axis=1，无限制 axis=3/-1,width[1, 8192], height 无限制 且受限于 tranpose 的规格限制		
				width/ 输入的 width			
			axis:int64	axis/ 做 softmax 的轴	1,3，即 channel 和 width 方向		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Slice	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			starts:int64[]	start/ 切分的起始位置	channel 方向 Slice 时，channel_start 要对齐。对齐量：8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。		
			ends:int64[]	ends/ 切分的终止位置	channel 方向 Slice 时，channel_end 要对齐。对齐量：8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。		
			axes:int64[]	axes/ 选取切分的轴	支持任意 0~3 轴， 支持同时多轴选择		
			steps:int64[]	steps/ 选取切分对应轴的步长	1		
Split	部分支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
			axis:int64	axis/ 切分的维度			
			num_outputs:int64	split 成几个输出			
			split:int64[]	spilt/ 指定切分后维度的长度	channel 方向 Split 时，除了最后一个输出外，其他输出的 channel 需要对齐。 对齐量：8bit 数据：8 对齐，16bit 数据：4 对齐。其他方向无限制。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Tanh	支持	int8 float16	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Transpose	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel	[1,8192]		
				height/ 输入的 height			
				width/ 输入的 width	[1,8176]		
			perm:int64[]	axis order/ 转置的轴顺序	限制与说明如下： 1. 假设 in_shape[n1,c1,h1,w1],out_shape[n2,c2,h2,w2] 2. 四种转换分别为： (1) perm=[0,2,3,1]，NCHW->NHWC。 (2) perm=[0,2,1,3]，NCHW->NHCW。 (3) perm=[0,3,1,2]，NCHW->NWCH。 (4) perm=[0,3,2,1]，NCHW->NWHC。 3. 以上四种转置无对齐要求。但在满足对齐要求时效率更高。对齐要求为：第 1 点中参数的 c1、c2 均要满足 8bit 数据：16 对齐， 16bit 数据：8 对齐。 4. NPU 限制项： (1) perm=[0,2,3,1]时，8bit 数据时， h1*w1<2048*2048，w1*c1<2048*512；16bit 数据时，h1*w1<2048*2048，w1*c1<2048*256。 (2) perm=[0,3,1,2]时，h1*w1<2048*2048。 (3) perm=[0,3,2,1]时 h1*w1<2048*2048,h2*w2<2048*2048。		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Convolution	支持	int8	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width	当 dilation_kernel_h > 1 时, width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明		
			kernel_shape [num_output, num input,kernel h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制		
				num input/ 输入的 channel			
				kernel_h/ height 方向的 kernel 大小	[1,31]		
				kernel_w/ width 方向的 kernel 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,7]		
				stride_w/ width 方向的 strides 大小			
			pads[pads top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,15]		
				pads_right/ right 方向的 pads 大小			
				pads top/ top 方向的 pads 大小			
				pads bottom/ bottom 方向的 pads 大小			
			group:int64	group/ group 的大小	无限制		
			dilations[dilations h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]		
				dilations_w/ widtht 方向的 dilations 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Depthwise Convolution	支持	int8	input tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width	当 dilation_kernel_h > 1 时, width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明		
			kernel_shape [num_output, num_input,kernel_h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制		
				num_input/ 输入的 channel			
				kernel_h/ height 方向的 kernel 大小	[1,8]		
				kernel_w/ width 方向的 kernel 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,7]		
				stride_w/ width 方向的 strides 大小			
			pads[pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	[0,15]		
				pads_right/ right 方向的 pads 大小			
				pads_top/ top 方向的 pads 大小			
				pads_bottom/ bottom 方向的 pads 大小			
			dilations[dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]		
				dilations_w/ widtht 方向的 dilations 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
ConvTranspose/ Deconvolution	支持	int8	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width	当 dilation_kernel_h > 1 时，width < 16383 此外,对首层输入 width 存在限制，详见模型输入说明		
			kernel_shape [num_output, num input,kernel h, kernel_w]:int64[]	num_output/ 输出的 channel	无限制		
				num input/ 输入的 channel			
				kernel_h/ height 方向的 kernel 大小	[1,31]		
				kernel_w/ width 方向的 kernel 大小			
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小	[1,8]		
				stride_w/ width 方向的 strides 大小			
			pads[pads top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小	支持 0-15 设置 pad 时注意: 不支持 kernel_h * dilations_h - dilations_h - pads_top < 0 ；不支持 kernel_w * dilations_w - dilations_w - pads_left < 0 ； 不支持 stride_h *(height - 1) - pads_top + 1 < output_h； 不支持 stride_w *(width - 1) - pads_left + 1 < output_w		
				pads_right/ right 方向的 pads 大小			
				pads top/ top 方向的 pads 大小			
				pads bottom/ bottom 方向的 pads 大小			
			group:int64	group/ group 的大小	1 当且仅当 num_input=num_output 时，支持 num_output		
			dilations[dilations h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]		
				dilations_w/ widtht 方向的 dilations 大小			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Gemm	尚不支持 目前由 CPU 实现	int8	input_tensor_1 [M, K]:tensor	M,K,N/ 输入数据的形状	转为 Matmul 实现，约束同 Matmul		per-layer/ per-channel
			input_tensor_2 [K,N]:tensor				
			alpha:double	alpha/ 矩阵 A*B 乘法的 scale	无限制		
			beta:double	beta/ 输入 C 矩阵的 scale			
			transA:int64	transA/ A 矩阵是否转置	仅静态 tensor 支持转置		
			transB:int64	transB/ B 矩阵是否转置			
MatMul (4d)	部分支持 目前该支持仅针对双 feature 输入 未来将支持输入为 feature+constant	int8	input_tensor_1[batch, channel，K, N]:tensor	batch/ 输入的 batch	双 feature 时： batch、H 无限制 K 支持[8,8192]，对齐要求为 8bit 数据：16 对齐，16bit 数据：8 对齐 C 支持[32,19384]，对齐要求为 32 对齐 feature+constant 时： 若 input_tensor_1 为 feature，则转为 batch 个 feature[K,C,1,1] + weight[H,C,1,1] 的 conv； 若 input_tensor_2 为 feature，则转为 batch 个 feature[1,C,H,1] + weight[K,C,1,1] 的 conv； C 对齐要求：32 对齐其他约束和 conv 相同		per-layer/ per-channel
				channel/ 输入的 channel			
			input_tensor_2[batch, channel, N, M]:tensor	K/ 输入的 K			
				N/ 输入的 M			
				M/ 输入的 M			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Expand	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int64	shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o	无限制		
				channel_o/ 输出的 channel			
				height_o/ 输出的 height			
				width_o/ 输出的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Where	支持	int8 float16 int64	x_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int8 float16 int64	y_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		bool	mask_tensor[batch, channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int8 float16	shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o	无限制		
				channel_o/ 输出的 channel			
				height_o/ 输出的 height			
				width_o/ 输出的 width			

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
exSoftmaxMask	部分支持	int8 float16	input_tensor_1[batch, channel,height,width]:tensor	batch/ 输入的 batch	无限制		per-layer
				channel/ 输入的 channel	硬件支持[1,8192]		
				height/ 输入的 height	axis=1，无限制 axis=3/-1,width[1, 8192], height 无限制 且受限于 transpose 规格限制		
				width/ 输入的 width			
			input_tensor_2[batch, channel,height,width]:tensor	batch/ 输入的 batch	1		
				channel/ 输入的 channel	硬件支持[1,8192]		
				height/ 输入的 height	1		
				width/ 输入的 width	axis=1: [1] axis=3/-1,[1, 8192]		
			axis:int64	axis/ 做 softmax 的轴	1,3，即 channel 和 width 方向		
			mask_value:int64	mask/ 需要 mask 的值	0 或 1		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
exGlu	支持	int8 float16	x_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch	c * h * w 满足如下限制 8bit 数据：8 对齐，16bit 数据：4 对齐		per-layer
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		int64	axis:int64	axis/ 切分的维度	axis ==1		

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Convolution + Relu	支持	同 Convolution					
Convolution + Clip	支持						
Convolution + PRelu/LeakyRelu	支持						
Convolution + Add	支持						
Convolution + Mul	尚不支持						
Convolution +Sigmoid	支持						
Convolution + Tanh	支持						
Convolution + Softplus	支持						
Convolution + HardSigmoid	支持						
Convolution + HardSwish	支持						
Convolution + Elu	支持						
Convolution + Swish	支持						
Convolution + Mish	支持						

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
ConvTranspose + Relu	尚不支持	同 ConvTranspose					
ConvTranspose + Clip	尚不支持						
ConvTranspose + PRelu/LeakyRelu	尚不支持						
ConvTranspose + Add	尚不支持						
ConvTranspose + Mul	尚不支持						
ConvTranspose + Sigmoid	尚不支持						
ConvTranspose + Tanh	尚不支持						
ConvTranspose + Softplus	尚不支持						
ConvTranspose + HardSigmoid	尚不支持						
ConvTranspose + HardSwish	尚不支持						
ConvTranspose + Elu	尚不支持						
ConvTranspose + Swish	尚不支持						
ConvTranspose + Mish	尚不支持						

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Depthwise Convolution + Relu	尚不支持	同 Depthwise Convolution					
Depthwise Convolution + Clip	尚不支持						
Depthwise Convolution + PRelu/LeakyRelu	尚不支持						
Depthwise Convolution + Add	尚不支持						
Depthwise Convolution + Mul	尚不支持						
Depthwise Convolution + Sigmoid	尚不支持						
Depthwise Convolution + Tanh	尚不支持						
Depthwise Convolution + Softplus	尚不支持						
Depthwise Convolution + HardSigmoid	尚不支持						
Depthwise Convolution + HardSwish	尚不支持						
Depthwise Convolution + Elu	尚不支持						
Depthwise Convolutione + Swish	尚不支持						
Depthwise Convolution + Mish	尚不支持						

Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Add+Relu	支持	同 Add					
Mul+Relu	支持	同 Mul					
Convolution + add + Relu	支持	同 Convolution					

注释：

(1) 广播说明：

以 ONNX 默认排列 NCHW 做说明，包含以下广播方式（A 或 B 都可以作为广播方）：

1. OP(A(N,C,H,W), B(N,C,H,W))，即两个维度相同的 tensor 进行操作；
2. OP(A(N,C,H,W), B(C,1,1))，即 C 维度做 broadcasting；
3. OP(A(N,C,H,W), B(scalar))，即以单个标量做 broadcasting；
4. OP(A(N,C,H,W), B(H,W))，即 HW 维度做 broadcasting。

广播支持举例：

1. OP(A(N,C,H,W), B(N,C,H,W)): OP(A(1,16,32,8), B(1,16,32,8)) = C(1,16,32,8)
2. OP(A(N,C,H,W), B(C,1,1)): OP(A(1,16,32 8), B(16)) = C(1,16,32,8)
3. OP(A(N,C,H,W), B(scalar)): OP(A(1,16,32,8), B(1)) = C(1,16,32,8)
4. OP(A(N,C,H,W), B(H,W)): OP(A(1,16,32,8), B(32x8)) = C(1,16,32,8)

约束规格中，[a,b]表示支持 a 到 b 之间的整数；{a,b,c}表示支持 a,b,c。

第五章 CPU OP 支持列表

Operator	描述	规格约束	说明
Add	加法操作	无限制	
AveragePool	平均池化	无限制	
ArgMin	取最小值的index	无限制	
ArgMax	取最大值的index	无限制	
BatchNormalization	批量归一化	无限制	
Cast	数据类型转换	SRC 支持：float32/bool/int8/float16/int32/int64 DST支持：float32/int8/int32/float16	
Clip	数据截断激活层	无限制	
Concat	合并操作	axis仅支持{0,1, 2, 3}	
Convolution	卷积操作	无限制	
ConvTranspose/Deconvolution	转置卷积	无限制	
Cos	余弦函数	无限制	
DataConvert	数据类型转换	仅支持 bool/int8/float类型转换	
DepthToSpace	通道方向空间方向转换	无限制	
Div	除法操作	无限制	
Equal	等于	无限制	
Exp	指数函数	无限制	

Operator	描述	规格约束	说明
Flatten	拉平操作	无限制	
Gather	聚集操作	无限制	
Greater	大于	无限制	
GreaterOrEqual	大等于	无限制	
GRU	门控循环单元	无限制	
GRU（extern）	门控循环单元	无限制	ONNX扩展算子
HardSwish (extern)	激活函数	无限制	ONNX扩展算子
InstanceNormalization	单例归一化	无限制	
LayerNorm (extern)	层归一化	无限制	ONNX扩展算子
Less	小于	无限制	
LessOrEqual	小等于	无限制	
LogSoftmax	激活函数	batchsize 仅支持 1	
LpNormalization	Lp归一化	无限制	
LRN (extern)	局部响应归一化	无限制	ONNX扩展算子
MatMul	多维矩阵相乘	无限制（支持四维x四维、四维x三维计算）	
Max	取最大值	无限制	
MaxPool	最大池化	无限制	

Operator	描述	规格约束	说明
MaxRoiPool	区域最大池化	无限制	
MaxUnpool	反向最大池化	无限制	
Mish(extern)	激活函数	无限制	ONNX扩展算子
Min	取最小值	无限制	
Mul	乘法	无限制	
Pad	填充	无限制	
Pow	指数计算	无限制	
Proposal (extern)	区域提议网络	batchsize 仅支持 1	ONNX扩展算子
ReduceMax	沿指定维度计算Max	输出维度不能超过4维	
ReduceMean	沿指定维度计算Mean	输出维度不能超过4维	
ReduceSum	沿指定维度计算Sum	输出维度不能超过4维	
ReduceMin	沿指定维度计算Min	输出维度不能超过4维	
Reorg	数据重排	无限制	
Reshape	数据形状改变	无限制	
Resize	数据宽高方向缩放	支持插值方式 bilinear; nearest2d	
ReverseSequence	序列翻转	无限制	
RMSNorm（extern）	均方根归一化	无限制	ONNX扩展算子

Operator	描述	规格约束	说明
RoiAlign	区域对齐池化	仅支持Avg Pool Mode,batchsize 仅支持 1	
ScatterND	N维索引取数	无限制	
Sin	正弦函数	无限制	
Slice	切片操作	batchsize 仅支持 1	
Softmax	激活函数	batchsize 仅支持 1	与ONNX OPSET 11规范一致
Softmax (extern)	激活函数	batchsize 仅支持 1	ONNX扩展算子，与ONNX OPSET 13规范一致
SpaceToDetph	空间方向向通道方向转换	无限制	
Split	拆分数据	无限制	
Sqrt	求平方根	无限制	
Squeeze	压缩数据维度	无限制	
Sub	减法	无限制	
Tanh	双曲正切函数	无限制	
Tile	扩充拷贝数据	batchsize 仅支持 1,不支持broadcasting	
Transpose	转置计算	无限制	
Upsample	上采样	支持插值方式 bilinear; nearest2d	
Not	按元素取非	无限制	
where	通过mask取数	无限制	
Erf	误差函数	无限制	

第六章 GPU OP 支持列表

Operator	描述	规格约束	说明
MatMul	多维矩阵相乘	无限制（支持四维x四维、四维x三维计算）	只支持float16，需设置GPU优先（参考《Rockchip_RKNPU_User_Guide_RKNN_SDK》）

第七章 模型输入输出规格说明

1. 模型输入说明

芯片平台	模型首层精度类型	输入维度	首层设置输入数据类型	mean/scale/quant 后端实现设备	输入宽（width）对齐要求 单位：元素个数		输入宽（width）大小限制	
					当输入通道（channel）为 1,3,4	当输入通道（channel）非 1,3,4	当输入通道（channel）为 1,3,4（声明见注释 9）	当输入通道（channel）非 1,3,4
RK3566/3568	int8	4 维度	uint8	NPU	8	1	各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式： 1. width * dilation_kernel_h < 1024*N 2. 2. width <= 4096 其中 N 必须为 1 到 7 的整数，超出范围的卷积不受支持，各卷积类型 N 的计算方式如下： Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 128) Depthwise Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 4096) ConvTranspose/Deconvolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 128)	无限制
			int8					
			float16	CPU				
			其他类型（*注释 8）					
	float16		uint8	CPU	4	1	各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式： 1. width * dilation_kernel_h < 1024*N 2. 2. width <= 4096 其中 N 必须为 1 到 7 的整数，超出范围的卷积不受支持，各卷积类型 N 的计算方式如下： Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 128) Depthwise Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 4096) ConvTranspose/Deconvolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 128)	无限制
			int8					
			float16					
			其他类型（*注释 8）					
	无限制	非 4 维	非限制	CPU	1	1	无限制	无限制

芯片平台	模型首层精度类型	输入维度	首层设置输入数据类型	mean/scale/quant 后端实现设备	输入宽（width）对齐要求 单位：元素个数		输入宽（width）大小限制	
					当输入通道（channel）为 1,3,4	当输入通道（channel）非 1,3,4	当输入通道（channel）为 1,3,4（声明见注释 9）	当输入通道（channel）非 1,3,4
RK3588	int8	4 维度	uint8	NPU	16	1	各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式： 1. width * dilation_kernel_h <= 2048 * N 2. width <= 8192 其中 N 必须为 1 到 7 的整数，超出范围的卷积不受支持，各卷积类型 N 的计算方式如下： Convolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 3) Depthwise Convolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 2048), 3) ConvTranspose/Deconvolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 3)	无限制
			int8					
			float16	CPU				
			其他类型（*注释 8）					
	float16		uint8	CPU	8	1	各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式： 1. width * dilation_kernel_h <= 1024 * N 2. width <= 8192 其中 N 必须为 1 到 7 的整数，超出范围的卷积不受支持，各卷积类型 N 的计算方式如下： Convolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 3) Depthwise Convolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 2048), 3) ConvTranspose/Deconvolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 3)	无限制
			int8					
			float16					
			其他类型（*注释 8）					
	无限制	非 4 维	非限制	CPU	1	1	无限制	无限制

芯片平台	模型首层精度类型	输入维度	首层设置输入数据类型	mean/scale/quant 后端实现设备	输入宽（width）对齐要求 单位：元素个数		输入宽（width）大小限制	
					当输入通道（channel）为 1,3,4	当输入通道（channel）非 1,3,4	当输入通道（channel）为 1,3,4（声明见注释 9）	当输入通道（channel）非 1,3,4
RK3562	int8	4 维度	uint8	NPU	16	1	各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式： 1. width * dilation_kernel_h <= 2048 * N 2. width <= 4096 其中 N 必须为 1 到 7 的整数，超出范围的卷积不受支持，各卷积类型 N 的计算方式如下： Convolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2) Depthwise Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 4096) ConvTranspose/Deconvolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2)	无限制
			int8					
			float16	CPU				
			其他类型（*注释 8）					
	float16		uint8	CPU	8	1	各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式： 1. width * dilation_kernel_h <= 2048 * N 2. width <= 4096 其中 N 必须为 1 到 7 的整数，超出范围的卷积不受支持，各卷积类型 N 的计算方式如下： Convolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2) Depthwise Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 4096) ConvTranspose/Deconvolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2)	无限制
			int8					
			float16					
			其他类型（*注释 8）					
	无限制	非 4 维	非限制	CPU	1	1	无限制	无限制

芯片平台	模型首层精度类型	输入维度	首层设置输入数据类型	mean/scale/quant 后端实现设备	输入宽（width）对齐要求 单位：元素个数		输入宽（width）大小限制	
					当输入通道（channel）为 1,3,4	当输入通道（channel）非 1,3,4	当输入通道（channel）为 1,3,4（声明见注释 9）	当输入通道（channel）非 1,3,4
RV1103/1106	int8	4 维度	uint8 int8	NPU	16	1	各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式： 1. width * dilation_kernel_h <= 2048 * N 2. width <= 4096 其中 N 必须为 1 到 7 的整数，超出范围的卷积不受支持，各卷积类型 N 的计算方式如下： Convolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2) Depthwise Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 4096) ConvTranspose/Deconvolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2)	无限制

注释：

1. 该对齐约束仅针对零拷贝 API，普通 API 无此对齐约束
2. 输入宽的对齐要求可从零拷贝 API 中的 w_stride 属性查询到，注意：w_stride 不支持更改
3. 仅对输入宽（width）在不同的通道（channel）条件下有对齐要求，其他无约束
4. 若输入不需要 mean 和 scale，需要将 mean 和 scale 配置为 0 和 1
5. 若通道（channel）> 4，则 mean/scale 将统一使用第一个数值，即 mean[0]和 scale[0]
6. 若首层为浮点类型则没有 quant 操作
7. RV1106/RV1103 不支持 CPU 的 mean/scale/quant 操作
8. 输入对齐要求可能变动
9. 声明：
CEIL(x)将 x 向上取整 (示例：CEIL(0.4)=1)
MAX(x, y)将获取 x、y 中的较大值（示例：MAX(2,3) = 3） dilation_kernel_h = kernel_h * dilations_h - dilations_h + 1 dilation_kernel_w = kernel_w * dilations_w - dilations_w + 1
10. 详细的用法请参考《Rockchip_RKNPU_User_Guide_RKNN_SDK》

2. 模型输出说明

芯片平台	模型输出精度类型（*注释 2）	输出维度	设置输出 Layout	Channel 对齐要求	H*W 对齐要求
RK3566/3568	int8	4 维度	NCHW	无	无
			NHWC	8 对齐（*注释 1）	无
			NC1HWC2	最后一层卷积类算子，16 对齐，最后一层非卷积类算子 8 对齐	H*W 要 4 对齐
			UNDEFINE	无	无
	float16		NCHW	无	无
			NHWC	4 对齐（*注释 1）	
			NC1HWC2	最后一层卷积类算子，8 对齐，最后一层非卷积类算子 4 对齐	H*W 要 4 对齐
			UNDEFINE	无	无
	无限制	非 4 维	UNDEFINE	无	无

芯片平台	模型输出精度类型（*注释 2）	输出维度	设置输出 Layout	Channel 对齐要求	H*W 对齐要求
RK3588	int8	4 维度	NCHW	无	无
			NHWC	16 对齐（*注释 1）	
			NC1HWC2	最后一层卷积类算子，32 对齐，最后一层非卷积类算子 16 对齐	H*W 要 4 对齐
			UNDEFINE	无	无
	float16		NCHW	无	无
			NHWC	8 对齐（*注释 1）	无
			NC1HWC2	最后一层卷积类算子，16 对齐，最后一层非卷积类算子 8 对齐	H*W 要 4 对齐
			UNDEFINE	无	无
	无限制	非 4 维	UNDEFINE	无	无

芯片平台	模型输出精度类型（*注释 2）	输出维度	设置输出 Layout	Channel 对齐要求	H*W 对齐要求
RK3562	int8	4 维度	NCHW	无	无
			NHWC	无	
			NC1HWC2	最后一层卷积类算子，32 对齐，最后一层非卷积类算子 16 对齐	H*W 要 4 对齐
			UNDEFINE	无	无
	float16		NCHW	无	无
			NHWC	无	无
			NC1HWC2	最后一层卷积类算子，16 对齐，最后一层非卷积类算子 8 对齐	H*W 要 4 对齐
			UNDEFINE	无	无
	无限制	非 4 维	UNDEFINE	无	无

芯片平台	模型输出精度类型（*注释 2）	输出维度	设置输出 Layout	Channel 对齐要求	H*W 对齐要求
RV1103/1106	int8	4 维度	NC1HWC2	最后一层卷积类算子，32 对齐，最后一层非卷积类算子 16 对齐	H*W 要 4 对齐
			NHWC	无	

注释：

1. 如果输出 tensor 类型是 NHWC 的，输出转换是 NPU 实现的输出，则有对齐要求，CPU 实现的没有对齐要求。
2. 输出精度类型 int8/float16 表示模型最后一层原始输出的数据类型。
3. NCHW 输出，如果是 NPU 实现采用零拷贝接口则输出内存开辟的 size 以 query 出来的 size 为准。
4. NC1HWC2 输出，输出内存开辟的 size 以 query 出来的 size 为准。