

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)

	当前版本:	v2.0.0-beta
文件状态:	作 者:	NPU团队
[]正在修改	编 辑:	刘雯君
[√] 正式发布	审核:	熊伟
	完成日期:	2024-03-22

瑞芯微电子股份有限公司
Rockchips Semiconductor Co., Ltd
(版本所有,翻版必究)



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	 新增 RK3562 exSoftmaxMask OP 支持项 新增 where OP 支持项 新增 MatMul GPU OP 支持项 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
V2.0.0-beta	NPU 团队	2024-03-22	1. 新增 RK3576 平台算子支持规格 2. 新 增 RK3588/RK3562/RK3576 平 台 的 exSDPAttention、exMatMul 的硬件支持规格 3. 新增 Floor、Mod、And、GatherElements 以及 exMeanVarianceNormalization CPU 算子支持 4. 新增 RK2118 平台算子支持规格	



目 录

第一章	RK3566/3568 NPU OP 支持列表	4
第二章	RK3588 NPU OP 支持列表	36
第三章	RV1103/1106 NPU OP 支持列表	72
第四章	RK3562 NPU OP 支持列表	105
第五章	RK3576 NPU OP 支持列表	140
第六章	RK2118 NPU OP 支持列表	. 175
第七章	CPU OP 支持列表	210
第八章	GPU OP 支持列表	216
第九章	模型输入输出规格说明	218



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
				batch/ 输入的 batch			
		intO	input_tensor	channel/ 输入的 channel 支持 ONNX 规范的四维 tensor 的所有			
Add/Bias	支持	int8 float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制	广播操作,详见: <u>注释(1)</u>	per-layer/ per-channel
				width/ 输入的 width			
		int8 float16	channe 输入的 input_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)	
Sub	支持			channel/ 输入的 channel			
Sub	义付		[batch,channel,height, width]:tensor	height/ 输入的 height	ניוו צוין ב		
				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
				batch/ 输入的 batch			
M 1/C 1	±-4±	int8	input_tensor	channel/ 输入的 channel 支持 ONNX 规范的四维 tensor 的所有 广播操作,详见: <u>注释(1)</u> 输入的 height	支持 ONNX 规范的四维 tensor 的所有	per-layer/	
Mul/Scale	支持	float16	[batch,channel,height, width]:tensor		元 限制	广播操作,详见: 注释(1)	per-channel
				width/ 输入的 width			
		float16		batch/ 输入的 batch		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	
			input tensor	channel/ 输入的 channel			
Div	部分支持		[batch,channel,height, width]:tensor	输入的 height 做 broadcasting;	3.OP((N,C,H,W),scalar),即以单个标量		
				width/ 输入的 width		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: <u>注释(1)</u>	



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/ 输入的 batch		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;	per-layer/ per-channel
Max	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制		
IVIAX	XII	float16	width]:tensor	height/ 输入的 height	做 b 3.OI 标量 说明	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。	
				width/ 输入的 width		说明: A 或 B 都可以作为广播方。例 子见: <u>注释(1)</u>	
		int8 float16	input_tensor [batch,channel,height,	batch/ 输入的 batch	 輸入的 batch channel/ 輸入的 channel 在 tensor 的厂播操作,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; 输入的 height 3.OP(A(N,C,H,W),B(scalar)),即以单个 	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;	per-layer/ per-channel
Min	支持			channel/ 输入的 channel			
Milli	ZN		width]:tensor	height/ 输入的 height			
				width/ 输入的 width		说明: A或B都可以作为广播方。例子见: <u>注释(1)</u>	



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/ 输入的 batch	1		
Global AveragePool 支持	-1-14	int8	input_tensor	channel/ 输入的 channel	[1,8192]	1,343] RKNN-Toolkit2 支持范围) 1,7] Complier 支持	per-layer
	文持	支持 float16	# # # # # # # # # #	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)		
				batch/ 输入的 batch	1		
	++4+	int8	input_tensor	channel/ 输入的 channel	[1,8192]		
GlobalMaxPool 支	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		per-layer
				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
				batch/ 输入的 batch	1		
			input_tensor	channel/ 输入的 channel			
			[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]		
				width/ 输入的 width			
			auto_pad:string	auto_pad/ pad 的方式	仅支持 NOTSET		
		int8 float16	ceil_mode:int64	ceil_mode/ 使用 ceil 或 floor 的方式计算输 出的 shape	不支持		per-layer
			count_include_pad:int64	count_include_pad/ 是否包含 pad 数值进行计算	1		
AveragePool	支持		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		
				kernel_w/ width 方向的 kernel 大小	- [1,/]; 兵已田 CPU 支持。 		
				pads_left/ left 方向的 pads 大小			
			pads[pads_top,pads_left,	pads_right/ right 方向的 pads 大小	[0.7]		
			pads_bottom, pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,7]		
				pads_bottom/ bottom 方向的 pads 大小			
			strides[strides_h,	stride_h/ height 方向的 strides 大小	[1 0]		
			strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,8]		



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/ 输入的 batch	1		
			input_tensor	channel/ 输入的 channel			
			[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]		
				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_h/ height 方向的 dilations 大小			
			dilations_w]:int64[]	dilations_w/ widtht 方向的 dilations 大小	1		
MaxPool	支持	int8 float16	kernel shape [kernel h,	kernel_h/ height 方向的 kernel 大小	无限制, NPU 支持-[1,7]; 其它由 CPU 支		per-layer
			kernel_w]:int64[]	kernel_w/ width 方向的 kernel 大小	- [1,/]; 兵匕田 CPU 文 持。		
				pads_left/ left 方向的 pads 大小			
			pads[pads_top,pads_left,	pads_right/ right 方向的 pads 大小	F0 57		
			pads_bottom, pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,7]		
				pads_bottom/ bottom 方向的 pads 大小			
		storage_order: int64	storage_order/优先储存方式	0			
			strides[strides h,	stride_h/ height 方向的 strides 大小	[1 0]	-	
		strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,8]			



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	
			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			
			momentum:double	momentum/ 训练时的滑动平均参数	无限制			
Batch	 支持	int8		batch/ 输入的 batch	1		per-layer/	
Normalization float16	float16	input_tensor [batch,channel,height,	channel/ 输入的 channel			per-channel		
		width]:tensor	height/ 输入的 height	无限制				
			width/ 输入的 width					
			batch/ 输入的 batch	支持多 batch				
			input_tensor	channel/ 输入的 channel				
		邓分支持 float16	[batch,channel,height, width]:tensor	height/ 输入的 height	- - - 无限制			
				width/ 输入的 width				
				layernorm_weight [channel,height,	channel/ 输入的 channel	等于 input_channel	1	
			width]:tensor(const)	height/ 输入的 height	等于 input_height	per-layer		
Layer Normalization	部分支持		layernorm_bias [channel,height, width]:tensor(const)	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
				batch/ 输入的 batch			
CI. /D THE	-+- ++-	int8	input_tensor	channel/ 输入的 channel	T 170 Ad		
Clip/ReLU6	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer
				width/ 输入的 width			
		int8 float16		batch/ 输入的 batch			
			input_tensor	channel/ 输入的 channel	T m del		
Elu	支持		[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		
				width/ 输入的 width			
				batch/ 输入的 batch			
	-t-14	int8	input_tensor	channel/ 输入的 channel	77744		
Gelu 支持	支持	float 16 [batch, channel	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		
				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
				batch/ 输入的 batch			
Relu	支持	int8	input_tensor	channel/ 输入的 channel	无限制		
Keiu X17	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	—— 人 []		per-layer	
			width/ 输入的 width				
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch			
T 1 D 1	++++	int8 float16		channel/ 输入的 channel	工作出		per-layer
LeakyRelu	支持			height/ 输入的 height	无限制		
				width/ 输入的 width			
				batch/ 输入的 batch			
				channel/ 输入的 channel	工作出		
PRelu	支持	持 Int8 [batch	input_tensor [batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer/ per-channel
				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						
				batch/ 输入的 batch	1								
			input_tensor [batch,channel,height, width]:tensor	sequence/ 输入的 sequence	无限制,建议4对齐								
	部分支持 GRU 扩 展以及变 体命名为		widenjitensor	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 layout R 以及变 本命名为 exGRU 算 子,参数 项中指明 (extern) 的项为 exGRU 独	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]								
GRU	exGRU 算 子,参数		float16	float16	参数 float16 ¹ 指明 xtern)	之,参数 float16 页中指明 (extern)	子,参数 项中指明 (extern)	子,参数 项中指明 (extern)	batch_size:int64 (extern)	batch_size/ 指定 GRU 输入的 batchsize	1		per-layer
	(extern)								sequence_size :int64 (extern)	sequence_size/ 指定 GRU 输入的 seqsize	无限制,建议4对齐		
	exGRU 独有的参数		hidden_size:int64 (extern)	hidden_size/ GRU 单元中的 hiddensize	无限制,建议8对齐								
	项。		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									
			input tongon	batch/ 输入的 batch	SSS											
			input_tensor [batch,channel,height,	sequence/输入的 sequence	无限制,建议4对齐											
		U以及变 G 名为 ELSTM I子,参 int8 I项中指 float16 (extern)	width]:tensor	input_size/ 输入的 input_size	无限制,建议8对齐											
						direction:stringdirection/ 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											
	展以及变			以及变	· 是以及变	展以及变	hidden_size:int64 (extern)	hidden_size/ LSTM 单元中的 hiddensize	无限制,建议8对齐							
	exLSTM		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		man lavan/									
LSTM	数项中指 明											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		per-layer/ per-channel
	(extern) 的项为			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。	i									
	exLSTM 独有的参		has_projection:int64 (extern)	has_projection/ projection 变种	1(T) or 0(F) 目前限定 0,即尚不支持											
数项。 	数项。	数项。	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)e: 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
				batch/ 输入的 batch	channel 方向 concat		
			input_tensor	channel/ 输入的 channel	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		
Concat	部分支持	int8 float16	[batch,channel,height, width]:tensor	height/ 输入的 height	据: 8 对齐, 16bit 数据: 4 对齐 其他方向 concat 无限		per-layer
				width/ 输入的 width	制。		
			axis:int64	aixs/ 拼接的维度	无限制		
				batch/ 输入的 batch			
Mish	士性	int8	input_tensor	channel/ 输入的 channel			
Misn	支持	float16 [batch,channel,] width]:tensor	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		
				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
			input_tensor	batch/ 输入的 batch	1		
		int8		hannel/ 输入的 channel	- 无限制		
Pad 支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	7.100 中山			
			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		
				batch/ 输入的 batch			
			input_tensor	channel/ 输入的 channel			
	尚不支持 目	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		
	前由 CPU 实现	float16		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
				batch/ 输入的 batch			
			input_tensor	channel/ 输入的 channel	工四本 1		
ReduceSum 尚不支持 目前由 CPU 实现	 尚不支持 目前	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制 		per-layer/
	float16		width/ 输入的 width			per-channel	
		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		
				batch/ 输入的 batch	支持多 batch		
	部分支持		ingut topoon	channel/ 输入的 channel	[1 0102]		
	目前 NPU 仅支	_	input_tensor [batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]		per-layer
Resize 持超数 线其分	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,	int8 float16	int8	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		
	其余不支持部 分的会 Fallback		mode:string	mode/resize 采用的模式	仅支持 nearest、linear		
	到 CPU 上实现。		scales:int64[]	scales/尺寸放大倍数	仅支持 1-8 整数倍		
		roi:int64[]					



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/ 输入的 batch	无限制		
		int8	input_tensor	channel/ 输入的 channel	[1 0102]		
		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]		
				width/ 输入的 width	[1,8176]		
			Shape (batch_o,channel_o, height_o,width_o):tensor	batch_o/ 输出的 batch_o	无限制		
Reshape	部分支持			channel_o/ 输出的 channel	- [1,8192]		
				height_o/ 输出的 height	[1,8192]		
		int64		width_o/ 输出的 width	[1,8176]		
			8 _ 7 _ 7	[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)	一个文持		



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
				batch/ 输入的 batch			
			input_tensor	channel/ 输入的 channel	无限制		
			[batch,channel,height, width]:tensor	height/ 输入的 height	/LPR 市J		
Reverse Sequence		int8 float16		width/ 输入的 width			
			batch_axis:int64	batch_axis/ 指定是否为 batch 维度	1	0	
			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		
				batch/ 输入的 batch			
Sigmaid	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	无限制		
Sigmoid	文 打	float16	width]:tensor	height/ 输入的 height	→ Nr hr tha		
				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
				batch/ 输入的 batch			
H40''4	支持	int8	input_tensor	channel/ 输入的 channel	- 无限制		
HardSigmoid	Tratusigniolu X ₁ ₁	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	一人吃吃到		per-layer
			width/ 输入的 width	-			
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch			
	+++	int8		channel/ 输入的 channel	- 无限制		
Swish	支持	float16		height/ 输入的 height			
				width/ 输入的 width			
				batch/ 输入的 batch			
H 10 '1	+++	int8	input_tensor	channel/ 输入的 channel	工作品		
HardSwish 3	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		per-layer
				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
				batch/ 输入的 batch			
Saftalua	士柱	int8	input_tensor	channel/ 输入的 channel	- 无限制		a on loven
Softplus	支持 支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	人口 NSC 中门		per-layer
				width/ 输入的 width			
				batch/ 输入的 batch			
			input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel			
Softmax	尚不支持,目前 由 CPU 实现			height/ 输入的 height	无限制		per-layer
				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
				batch/ 输入的 batch			
			input_tensor [batch,channel,height,	channel/ 输入的 channel	无限制		
			width]:tensor	height/ 输入的 height	プレド X (中)		
			width/ 输入的 width				
Slice	部分支持	int8 float16	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。		per-layer
			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		
				batch/ 输入的 batch			
			input_tensor	channel/ 输入的 channel			
			[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		
Split 部分支	部分支持	int8 float16		width/ 输入的 width			per-layer
		Hoatio	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
				batch/ 输入的 batch			
T. 1	++++	int8	input_tensor	channel/ 输入的 channel	丁. 7日 4-1		
Tanh	Tanh 支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer
			width/ 输入的 width				
			batch/ 输入的 batch	[1,1024]			
			input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel	[1 0102]		
				height/ 输入的 height	[1,8192]		
_	علما والمحالم	int8		width/ 输入的 width	[1,8176]		
Transpose 部分支	部分支持	float16	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
				batch/ 输入的 batch			
			input_tensor	channel/ 输入的 channel	channel/		
			[batch,channel,height, width]:tensor	height/ 输入的 height			
				width/ 输入的 width	当 <u>dilation_kernel_h</u> >1 时, width <16383 此外,对首层输入 width 存在限 制,详见 <u>模型输入说明</u>		
				num_output/ 输出的 channel	- 无限制		
			kernel_shape [num_output,	num_input/ 输入的 channel	无 pix 街 j		
			num_input,kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小	- [1,31]		
		:+0		kernel_w/ width 方向的 kernel 大小			non loven/
Convolution	支持	strides[strides_h, height 方向的 strides_w]:int64[] stride_w/ width 方向的 pads_left/		height 方向的 strides 大小	[1,7]		per-layer/ per-channel
				stride_w/ width 方向的 strides 大小	[1,7]		
			left 方向的 pads 大小				
			pads[pads_top, pads left, pads bottom,	pads_right/ right 方向的 pads 大小	[0,15]		
			pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,13]		
				pads_bottom/ bottom 方向的 pads 大小			
			group:int64	group/ group 的大小	无限制		
			dilations[dilations_h,	dilations_h/ height 方向的 dilations 大小	[1, 32]		
			dilations_w]:int64[]	dilations_w/ widtht 方向的 dilations 大小	[.,,22]		



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
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height	无限制		
				width/ 输入的 width	当 <u>dilation_kernel_h</u> > 1 时, width < 16383 此外,对首层输入 width 存 在限制,详见 <u>模型输入说明</u>		
			num_output/ 输出的 channel kernel_shape [num_output, num_input,kernel_h, kernel_w]:int64[] num_output/ 输入的 channel kernel_h/ height 方向的 kernel 大小 [1,8]	输出的 channel num_input/	一 无限制		
Depthwise Convolution	支持	int8 float16		[1,8]		per-layer/ per-channel	
Convolution		Hoatro	strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	[1,7]		per-chaimer
			pads[pads_top, pads_left, pads_bottom,	pads_left/ left 方向的 pads 大小 pads_right/ right 方向的 pads 大小	[0,15]		
			pads_right]:int64[]	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 dilations_w/ widtht 方向的 dilations 大小	[1, 32]		



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
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/	无限制		
				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_output/ 输出的 channel num_input/ 输入的 channel	- 无限制		
			num_input,kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小	[1,31]		
ConvTranspose/ Deconvolution	支持	int8 float16	strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	{1,2,4,8}		per-layer/ per-channel
			pads[pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小 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	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		
			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 dilations_w/ widtht 方向的 dilations 大小	[1, 32]		



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
			input_tensor_1 [M, K]:tensor	M,K,N/			
			input_tensor_2 [K,N]:tensor	输入数据的形状			
Comm	不支持 ,	int8	alpha:double	alpha/ 矩阵 A*B 乘法的 scale	- - - - - - - -		per-layer/
Gemm	由 CPU 实 现	CPU 头 float16	beta:double	beta/ 输入 C 矩阵的 scale	一 不支持−−		per-channel
			transA:int64	transA/ A 矩阵是否转置			
			transB:int64	transB/ B 矩阵是否转置			
			input_tensor_1	batch/ 输入的 batch			
MatMul	不支持 , 由 CPU 实	int8	[batch, K, C]:tensor	K/ 输入的 K	- - 不支持		per-layer/
iviativitii	现	float16	input_tensor_2 [batch, C, H]:tensor	C/ 输入的 C	1 ~10		per-channel
				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
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch			
		int8		channel/ 输入的 channel	— 无限制 —		
		float16		height/ 输入的 height			
Expand	支持			width/ 输入的 width			
Lapand	ZN	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
			x_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			
	int8 float1 int64 Where 支持 bool	float 16 y 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 -		
where		bool	mask_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
	int8 float16	shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o channel_o/ 输出的 channel height_o/ 输出的 height width_o/ 输出的 width	- 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
		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			
exGlu	支持			height/ 输入的 height			
exGlu	大 14			width/ 输入的 width			
		int64	axis:int64	axis/ 切分的维度	axis ==1		



		1112500755001110 01 2,1177540			110 10 10 10 10		
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						
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	同 Convolution					
Convolution + Softplus	支持						
Convolution + HardSigmoid	支持						
Convolution + HardSwish	支持						
Convolution + Elu	支持						
Convolution + Swish	支持						
Convolution + Mish	支持						



				K5500/5500 NI U UI XN 911X		州心城屯1从	BIIIKA
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						
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	同 ConvTranspose					
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						



				255 000 55 00 1 1 2 0 1 963 1 9 1 9 1	0/3500 NI C OI X11711X		
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						
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	同 Depthwise Conv	volution				
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	司 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
				batch/ 输入的 batch			per-layer/ per-channel	
		int8	input tensor	channel/ 输入的 channel	无限制	支持ONNX规范的四维 tensor 的所有广播操作,详见: <u>注释(1)</u>		
Add/Bias	Add/Bias 支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height				已支持
				width/ 输入的 width				
				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;	per-layer/ per-channel	
Sub	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel				以
Suo	文刊	float16	width]:tensor	height/ 输入的 height		维度做 broadcasting; 3.OP(A(N,C,H,W),B(scalar)),即以 单个标量做 broadcasting。		尚不支持
				width/ 输入的 width		说明: A 或 B 都可以作为广播方。 例子见: <u>注释(1)</u>		



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/ 输入的 batch				
		int8	input_tensor	channel/ 输入的 channel		支持 ONNX 规范的四维 tensor 的所	per-layer/	
Mul/Scale 支持	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制	有广播操作,详见: <u>注释(1)</u>	per-channel	尚不支持
				width/ 输入的 width				尚不支持
				batch/ 输入的 batch		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:		
			input tensor	channel/ 输入的 channel		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		
Div	部分支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制	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		
				width/ 输入的 width		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: <u>注释(1)</u>		



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/ 输入的 batch		支持两个 tensor 的广播操作,以 ONNX 默认排列 NCHW 做说明,	per-layer/		
May	本 株	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	T. 17Fl /k.il	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;		尚不支持	
IVIAX	Max 支持	float16	width]:tensor	ht, height/ 输入的 height width/ 输入的 width	height/	→ 无限制 —	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。	per-channel	问小义行
						说明: A 或 B 都可以作为广播方。 例子见: <u>注释(1)</u>			
				batch/ 输入的 batch		支持两个 tensor 的广播操作,以 ONNX 默认排列 NCHW 做说明,支			
Min	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	1 一 无限制 2 2 4 3	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	per-layer/	尚不支持	
Min 支	XN	float16	width]:tensor	height/ 输入的 height		2.OP(A(N,C,H,W),B(C,I,I)), 即 C 维度做 broadcasting; 3.OP(A(N,C,H,W),B(scalar)),即以 单个标量做 broadcasting。	per-channel	间尔文的	
				width/ 输入的 width		说明: A 或 B 都可以作为广播方。 例子见: <u>注释(1)</u>	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
				batch/ 输入的 batch	1			
Global AveragePool 支持	±-t±	int8	input_tensor	channel/ 输入的 channel	[1,8192]		1	业 工 + 上
	又付	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		per-layer	同个文付
				width/ 输入的 width	[1,7] (Complier 支持 范围)			
				batch/ 输入的 batch	1			
	+-+±	int8	input_tensor	channel/ 输入的 channel	[1,8192]			W.T.++±
GlobalMaxPool	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		per-layer	多核协同
				width/ 输入的 width	[1,7] (Complier 支持 范围)			

瑞芯微电子股份有限公司



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/ 输入的 batch				
			input_tensor	channel/ 输入的 channel				
			[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]			
				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			
AveragePool	支持	int8 float16	kernel shape [kernel h,	kernel_h/ height 方向的 kernel 大小	无限制,NPU 支持。		per-layer	尚不支持
			kernel_w]:int64[]	kernel_w/ width 方向的 kernel 大小	持[1,7]; 其它由 CPU 支持。			
				pads_left/ left 方向的 pads 大小				
			pads[pads_top,pads_left, pads_bottom,	pads_right/ right 方向的 pads 大小	[0.7]			
			pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,7]			
				pads_bottom/ bottom 方向的 pads 大小				
			strides[strides_h,	stride_h/ height 方向的 strides 大小	[1 0]			
			strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,8]			



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/ 输入的 batch	1			
			input_tensor	channel/ 输入的 channel				
			[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]			
				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_h/ height 方向的 dilations 大小				
			dilations_w]:int64[]	dilations_w/ widtht 方向的 dilations 大小	1			
MaxPool	支持	int8 float16	kernel shape [kernel h,	kernel_h/ height 方向的 kernel 大小	无限制,NPU 支持		per-layer	尚不支持
			kernel_w]:int64[]	kernel_w/ width 方向的 kernel 大小	[1,7]; 其它由 CPU 支持。			
				pads_left/ left 方向的 pads 大小				
			pads[pads_top,pads_left,	pads_right/ right 方向的 pads 大小	50.53			
			pads_bottom, pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,7]			
				pads_bottom/ bottom 方向的 pads 大小				
	storage_order: int64 storage_order/优先储存方式	storage_order/优先储存方式	0					
			strides[strides h,	stride_h/ height 方向的 strides 大小	[1.0]			
			strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,8]			



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
			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 le-5			
			momentum:double	momentum/ 训练时的滑动平均参数	无限制			多核协同
Batch Normalization	支持	int8 float16		batch/ 输入的 batch	1		1 "	尚不支持
1,0111011		110 401 0	input_tensor [batch,channel,height,	channel/ 输入的 channel			per enamer	尚不支持
		width]:tensor	height/ 输入的 height	无限制				
				width/ 输入的 width				
				batch/ 输入的 batch	支持多 batch			
			input_tensor	channel/ 输入的 channel			per-channel	
			[batch,channel,height, width]:tensor	height/ 输入的 height	无限制			尚不支持
				width/ 输入的 width				
			layernorm_weight [channel,height,	channel/ 输入的 channel	等于 input_channel			er/ annel 尚不支持
			width]:tensor(const)	height/ 输入的 height	等于 input_height			
Layer Normalization	部分 支持	float16	layernorm_bias [channel,height, width]:tensor(const)	width/ 输入的 width	等于 input_width		per-layer	尚不支持
			normalized_shape:int 64[]	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:in t64	elementwise_affine/ 是否具有可学习数	0 或 l (默认为 0)。 当 为 l 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			



			防止除剂	去溢出的偏移参数				
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
				batch/ 输入的 batch				多核协同 已支持 尚不支持
Cli.,/D.LLI	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	无限制			
Clip/ReLU6	又行	float16	width]:tensor	height/ 输入的 height	一 人的民 市门		per-layer	口文14
				width/ 输入的 width				
				batch/ 输入的 batch				己支持
Elu	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	无限制			
Elu	文刊	float16	width]:tensor	height/ 输入的 height	נייו אין בו			问小文符
				width/ 输入的 width				
				batch/ 输入的 batch				
Gelu	支持	int8	input_tensor	channel/ 输入的 channel	无限制			 - - -
Geiu	大 打	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	ניקו איץ טי			阿尔文河
				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
				batch/ 输入的 batch				多核协同 已支持 已支持
Relu	支持	int8	input_tensor	channel/ 输入的 channel	无限制		mon loven	日本株
Keiu	文4	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	→ → → → → → → → → → → → → → → → → → →		per-layer	口文科
				width/ 输入的 width			per-layer per-layer per-layer/ per-channel	
				batch/ 输入的 batch				
I 1 D 1	+-++	int8	input_tensor	channel/ 输入的 channel	丁//日本·J			3 + 4 +
LeakyRelu	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer	C文持
				width/ 输入的 width				
				batch/ 输入的 batch				
				channel/ 输入的 channel	了70 4·1			
PRelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer/ per-channel	已支持
			widenj.tensor	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				
			input tensor	batch/ 输入的 batch	1							
			[batch,channel,heigh t,	sequence/ 输入的 sequence	无限制,建议4对齐			量化方式 多核协同 per-layer 尚不支持				
			width]:tensor	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							
	部分支持 GRU 扩 展以及变		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]							
GRU	体命名为 exGRU 算 子,参数	float16	batch_size:int64 (extern)				nor layer	出不古柱				
GKU	项中指明 (extern)	sequence_size sequence_size/ 指定 GRU 输入的 seqsize hidden_size:int64 (extern) hidden_size/ GRU 单元中的 hiddensize **Table Sequence_size	float16	float16	float16	float16			无限制,建议4对齐		per-rayer	内小文行
	的项为 exGRU 独											_
	有的参数 项。											
			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。	s, Ž						



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	
			input 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。				
			[batch,channel,heigh	sequence/ 输入的 sequence	无限制,建议4对齐		per-		
			t,width]:tensor	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				
	部分支持		batch_size:int64 (extern)	batch_size/ 指定LSTM输入的 batchsize	大于1时仅支持4的倍数				
	LSTM 扩		sequence_size :int64 (extern)	sequence_size/ 指定 LSTM 输入的 seqsize	无限制,建议4对齐				
	展以及变体命名为		hidden_size:int64 (extern)	hidden_size/ LSTM 单元中的 hiddensize	无限制,建议8对齐				
	exLSTM 算子,参	int8	int8	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		I -	尚不支
LSTM	数项中指明	float16	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		per		
	(extern) 的项为		has_dropout:int64 (extern)	has_dropout/ caffe 框架下的 indicator 功能的选择	I(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。				
	exLSTM 独有的参 数项。		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	l.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)						



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/ 输入的 batch channel 方向 concat		输入的 batch channel 方向 concat		
			input_tensor [batch,channel,height,	channel/ 输入的 channel	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:			
Concat	部分支持	int8 float16	width]:tensor	height/ 输入的 height	安州介。州介里: 8bit 数据: 8 对齐, 16bit 数据: 4 对齐 其他方向 concat 无		per-layer	已支持
				width/ 输入的 width	限制。			
			axis:int64	aixs/ 拼接的维度	无限制			
				batch/ 输入的 batch				
Mish	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel				当不支持
IVIISII	メガ	float16	width]:tensor	height/ 输入的 height	- 无限制			尚不支持
				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
				batch/ 输入的 batch	1			
		int8	input_tensor [batch,channel,height,	hannel/ 输入的 channel	- 无限制			
		float16	width]:tensor	height/ 输入的 height	人口以中!			
Pad 支持	支持			width/ 输入的 width	[1,8176]			尚不支持
	ZN.	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			多核协同 尚不支持 一 おおける おおりままままままままままままままままままままままままままままままままままま
		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			尚不支持
				batch/ 输入的 batch				
			input_tensor	channel/ 输入的 channel	- 无限制			
DadwaMaan	尚不支持 目前由 CPU 实	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	7 / 12 12 12 13 13 13 13 13			坐 不去挂
	· 即田 CPU 实 · 现	float16		width/ 输入的 width				
		axes:int64[]	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			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
			input_tensor	batch/ 输入的 batch channel/ 输入的 channel				
D - las - Com	尚不支持 目	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	一 无限制		per-layer/	业 无士特
ReduceSum	前由 CPU 实现	float16		width/ 输入的 width			per-channel	尚不支持
			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			
				batch/ 输入的 batch	支持多 batch			
	部分支持		input tensor	channel/ 输入的 channel	- [1,8192]			
	目前 NPU 仅支 持宽高方向不		[batch,channel,height, width]:tensor	height/ 输入的 height				
Resize	超过 8 倍的整倍数的最近邻和线性插值缩放,其余不支	int8 float16		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		per-layer	尚不支持
	持部分的会 Fallback 到		mode:string	mode/resize 采用的模式	仅支持 nearest、linear			
	CPU上实现。		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])	局		



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/ 输入的 batch	无限制			
		int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	[1,8192]			
		float16	width]:tensor	height/ 输入的 height	[1,8192]			
				width/ 输入的 width	[1,8176]			
				batch_o/ 输出的 batch_o	无限制			
Reshape	部分支持			channel_o/ 输出的 channel	[1 9102]			尚不支持
				height_o/ 输出的 height	[1,8192]			
		int64	Shape (batch_o,channel_o,	width_o/ 输出的 width	[1,8176]			
			height_o,width_o):tensor	[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)	(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			
				batch/ 输入的 batch							
			input_tensor	channel/ 输入的 channel	· 无限制						
			[batch,channel,height, width]:tensor	height/ 输入的 height	ן ניקו אוין בו			多核协同 尚不支持			
Reverse Sequence	尚不支持 目前由 CPU 实现	int8 float16		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						
							batch/ 输入的 batch				
Sigmoid	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制			尚不支持			
Signiolu	XII	float16	width]:tensor	height/ 输入的 height	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \			Heli (1, X-1)			
				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
				batch/ 输入的 batch				
HardSigmoid	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	- - 无限制		per-layer	出不支持
HardSigmoid	文 持	float16	width]:tensor	height/ 输入的 height	一		per-layer	问小义符
				width/ 输入的 width				
				batch/ 输入的 batch				
Swish	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	一 无限制			出不支持
SWISH	又1寸	float16	width]:tensor	height/ 输入的 height	プロドス 中リ			尚不支持
				width/ 输入的 width				
				batch/ 输入的 batch				
HardSwish	支持	int8	input_tensor	channel/ 输入的 channel	一 无限制		man layran	兴 不去持
Haluswish	又1寸	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	<u> </u>		per-layer	
				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
				batch/ 输入的 batch				多核协同 尚不支持
Softplus 支持	古体	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	- - 无限制		and larves	坐 不去持
	文 哲	float16	width]:tensor	height/ 输入的 height	プレドに 中リ		per-layer	问个文符
				width/ 输入的 width				
				batch/ 输入的 batch	无限制			
			input_tensor [batch,channel,height,	channel/ 输入的 channel	硬件支持[1,8192]			
Softmax	支持	int8 float16	width]:tensor	height/ 输入的 height	axis=1, 无限制 axis=3/-1,width[1, - 8192], height 无限制		per-layer	尚不支持
				width/ 输入的 width	且受限于 tranpose 的 规格限制			
		axis:int64	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
				batch/ 输入的 batch				
			input_tensor	channel/ 输入的 channel	T UH Asi			
			[batch,channel,heigh t,width]:tensor	height/ 输入的 height	- 无限制			多核协同 尚不支持
	Slice 部分支持			width/ 输入的 width				
Slice		int8 float16	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。		per-layer	尚不支持
			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			
				batch/ 输入的 batch				
			input_tensor [batch,channel,heigh	channel/ 输入的 channel				
			t, width]:tensor	height/ 输入的 height	- 无限制			
Split	部分支持	int8 float16		width/ 输入的 width			per-layer	尚不支持
			axis:int64	axis/ 切分的维度				尚不支持
			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。			



					588 NFU OF 文持列农			
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/ 输入的 batch				
T1		int8	input_tensor [batch,channel,height	channel/ 输入的 channel				多核协同 尚不支持
Tanh	2,14	float16	width]:tensor	height/ 输入的 height	— 人口PR 申]		per-layer	
			width/ 输入的 width					
				batch/ 输入的 batch	[1,1024]			
			input_tensor [batch,channel,height	channel/ 输入的 channel	[1 9102]			
			width]:tensor	height/ 输入的 height	[1,8192]			
_	<i>→</i> 17 /\ → → ↓	int8		width/ 输入的 width	[1,8176]			尚不支持
Transpose	部分支持	float16	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。			阿小文持



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
			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 当 <u>dilation_kernel_h</u> > 1 时, width < 16383 此外,对首层输入 width 存在限 			
			kernel_shape [num_output, num_input,kernel_h,	num_output/ 输出的 channel num_input/ 输入的 channel	制,详见 <u>模型输入说明</u>			
Convolution		int8	kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小 stride h/	- [1,31]		per-layer/	己支持
Convolution	支持	float16	strides[strides_h, strides_w]:int64[]	height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	- [1,7]		per-channel	
			pads[pads_top,pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小 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	- [0,15]			
			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 dilations_w/ widtht 方向的 dilations 大小	[1, 32]			



Rockchi ^{瑞芯微电}	P			RK3588 NPU OP 支持列表	瑞芯微电子股份	分有限公司		
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
			input_tensor [batch,channel,height,	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height	无限制			
			width]:tensor	width/ 输入的 width	当 <u>dilation_kernel_h</u> > 1 时, width < 16383 此外,对首层输入 width 存在限制,详见 <u>模型输</u> 入说明			
			kernel_shape [num_output,	num_output/ 输出的 channel num_input/ 输入的 channel	- 无限制			
Depthwise Convolution	支持	int8 float16	num_input,kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小	- [1,8]		per-layer/ per-channel	已支持
			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	- [1,7]			
			pads[pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小 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	[0,15]			
			dilations[dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小 dilations_w/ widtht 方向的 dilations 大小	[1, 32]			



Rockchip ^{瑞芯微电子}				RK3588 NPU OP 支持列表			瑞芯微电子股份	有限公司
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
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/		1<		
				输入的 width	此外,对首层输入 width 存在限制,详 见 <u>模型输入说明</u>			
				num_output/ 输出的 channel num_input/	- 无限制			
			kernel_shape [num_output, num_input,kernel_h,	输入的 channel kernel h/				
			kernel_w]:int64[]	height 方向的 kernel 大小 kernel w/	[1,31]			多核协同
ConvTranspose/		int8		width 方向的 kernel 大小 stride h/			per-layer/	 尚不支持
Deconvolution		float16	strides[strides_h, strides_w]:int64[]	height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	{1,2,4,8}		per-channel	
				pads_left/ left 方向的 pads 大小	支持 0-15 设置 pad 时注意:			
			pads[pads_top, pads_left, pads_bottom,	pads_right/ right 方向的 pads 大小 pads top/	不支持 kernel_h * dilations_h - dilations_h - pads_top < 0 不支持 kernel_w * dilations_w - dilations_w -			尚不支持
			pads_right]:int64[]	top 方向的 pads 大小 pads bottom/	pads_left < 0 不支持 stride_h *(height - 1) - pads_top + 1 < output_h 不 支 持 stride_w *(width - 1) -			
				bottom 方向的 pads 大小 group/	pads_left + 1 < output_w 1 当且仅当 num_input=num_output			
			group:int64	group 的大小 dilations h/	时,支持 num_output			
			dilations[dilations_h, dilations_w]:int64[]	height 方向的 dilations 大小 dilations_w/ widtht 方向的 dilations 大小	[1, 32]			



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
			input_tensor_1 [M, K]:tensor	M,K,N/	转为 Matmul 实现,约束同 Matmul			
			input_tensor_2 [K,N]:tensor	输入数据的形状	我內 Maumui 头塊,约米问 Maumui			
Commi	尚不支持 , 目前由 CPU	int8	alpha:double	alpha/ 矩阵 A*B 乘法的 scale	- 无限制		per-layer/	业 不士柱
Semm 目前田 CPU 实现	float16	beta:double	beta/ 输入 C 矩阵的 scale	7 元限制		per-channel	尚不支持	
			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	7 仪 即念 tensor 文 行 转 直			
	部分支持		input_tensor_1	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			
	目前该支持 仅 针 对 双 feature 输入	int8	[batch, K, C]:tensor	K/ 输入的 K				
大 有 有	未来将支持	float16	input tensor 2	C/ 输入的 C	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;		per-layer/ per-channel	尚不支持
	输 入 为 feature+ constan		input_tensor_2 [batch, C, H]:tensor H/ 输入的 H		Teature[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			

瑞芯微电子股份有限公司



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
			input_tensor_1	batch/ 输入的 batch	双 feature 时: batch 无限制 channl、K 支持[8,8192],对齐要求为			
	部分支持		[batch,channel,K, N]:tensor	batch/ 输入的 batch	8bit 数据: 16 对齐, 16bit 数据: 8 对 齐 N 支持[32,19384], 对齐要求: 32 对齐			
MatMul	目前该支持 仅 针 对 双 feature 输入	int8	input_tensor_2	K/ 输入的 K	K*N <=65532 K*M <=65532 M*N <=65532		per-layer/	尚不支持
(4d)	未来将支持输 入 为	float16	[batch,channel,N, M]:tensor	N/ 输入的 M	feature+constant 时: 若 input_tensor_1 为 feature,则转为 batch*channel 个 feature[K,N,1,1] +		per-channel	
	feature+ constant			M/ 输入的 M	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			



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/ 输入的 batch				
		int8	input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel				
		float16		height/ 输入的 height	- 无限制			
Evnand	支持			width/ 输入的 width				尚不支持
Expand	文14	shape(batch_o, channel_o,height_o, width_o):tensor		batch_o/ 输出的 batch_o	- 无限制			问小文母
			shape(batch_o,	channel_o/ 输出的 channel				
			width_o):tensor	height_o/ 输出的 height	JURKIPI			
				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	多核协同
		int8 float16 int64	x_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 ——			
		int8 float16 int64 y_tensor [batch, channel, height, width]:tensor mask_tensor[batch, channel, height, width]:tensor mask_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 ——			
Where	文持		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	
		int8 float16 shape(batch_o, channel_o,height_ width_o):tensor		batch_o/ 输出的 batch_o 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	多核协同
				batch/ 输入的 batch	无限制			
			input_tensor_1[batch, channel,height,width]:tenso	channel/ 输入的 channel	硬件支持[1,8192]			
			r	height/ 输入的 height	axis=1, 无限制 axis=3/-1,width[1, 8192], height			
				width/ 输入的 width	无限制 且受限于 transpose 规格限制			
exSoftmaxMask	and and and interest	l l		batch/ 输入的 batch	1		per-layer	尚不支持
CASOITHIAAIVIASK	maxMask 支持 float16		input_tensor_2[batch, channel,height,width]:tenso	channel/ 输入的 channel	硬件支持[1,8192]		pei-iayei	间小文灯
			r	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	多核协同
				batch/ 输入的 batch				
		int8	x_tensor [batch, channel, height, width]:tensor	channel/ 输入的 channel c*h*w 满足如下限制 8bit 数据: 8 对齐, 16bit 数				
exGlu	exGlu 支持	支持		height/ 输入的 height	据: 4对齐	X	per-layer	尚不支持
CAGIU	XN			width/ 输入的 width				向小义行
			axis:int64	axis/ 切分的维度	axis ==1			
			input_tensor_1[batch,	batch/ 输入的 batch				
exMatMul	exMatMul 支持	int8	K,1, N]:tensor	K/ 输入的 K	- K支持[1,8192]		per-layer	尚不支持
CAIVIAUVIUI	文刊	float16		M/ 输入的 M	K X 1 [1,0172]		per-layer	
			1, M]:tensor	N/ 输入的 N				



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/ 输入的 batch				
		float16	query_tensor [batch,channel,height,	channel/ 输入的 channel	Channel < 8192			
		Hoatro	width]:tensor	height/ 输入的 height	Channel < 8192			
	ev SDPAtten			width/ 输入的 width				
exSDPAtten		float16 [b	key_tensor	batch/ 输入的 batch				W T + ++
tion	又行			channel/ 输入的 channel				尚不支持
			[batch,channel,height, width]:tensor	height/ 输入的 height				
	float16			width/ 输入的 width				
		Slacet 16	value_tensor	batch/ 输入的 batch	W. 44. < 0.102			
		Hoatio	[batch,channel,height, width]:tensor	channel/ 输入的 channel	- Width < 8192			

			1415500111 C OI X117717C		1111011111111	及仍们代益马	
			height/ 输入的 height				
			width/ 输入的 width				
			batch/ 输入的 batch				
	float16	mask_tensor	channel/ 输入的 channel				
	noat16	[batch,channel,height, width]:tensor	height/ 输入的 height				
			width/ 输入的 width				
	float32	scale					



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							已支持
Convolution + Clip	支持							已支持
Convolution + PRelu/LeakyRelu	支持							己支持
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	同 Convolution						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							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	同 ConvTranspose						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							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	同 Depthwise Conv	volution					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
				batch/ 输入的 batch			
		int8	input_tensor	channel/ 输入的 channel	- 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/
Add/Bias	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height			per-channel
				width/ 输入的 width			
	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;	per-layer/
Sub				channel/ 输入的 channel			
Sub				height/ 输入的 height		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	per-channel
				width/ 输入的 width		见: 注释 (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
				batch/ 输入的 batch			
Mul/Scale 支		int8	input_tensor	channel/ 输入的 channel		支持 ONNX 规范的四维 tensor 的所有广播操作,详见: 注释(1)	per-layer/
	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-channel
				width/ 输入的 width			
		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	
				channel/ 输入的 channel			
Div	部分支持			height/ 输入的 height		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	per-layer/ per-channel
				width/ 输入的 width		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)	



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/ 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	
Max 支持	士性	int8	input_tensor	channel/ 输入的 channel	[1 9102]	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。	per-layer/
	大村	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]		per-channel
				width/ 输入的 width	[1,8176]	说明: A 或 B 都可以作为广播方。例 子见: <u>注释(1)</u>	
		int8	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	j per lever/
Min	支持			channel/ 输入的 channel		的 channel	
Min	大田			height/ 输入的 height	[1,8192]	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。	per-channel
				width/ 输入的 width			例



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/ 输入的 batch	1		
Global	支持	:0	input_tensor	channel/ 输入的 channel	[1,8192]		
AveragePool	又付	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		per-layer
				width/ 输入的 width	[1,7] (Complier 支持 范围)		
				batch/ 输入的 batch	1		per-layer
ClahalMayDaal	本 柱	:40	input_tensor	channel/ 输入的 channel	[1,8192]		
GlobalMaxPool	支持		[batch,channel,height, width]:tensor	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
				batch/ 输入的 batch	1		
			input_tensor	channel/ 输入的 channel			
			[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]		
				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		per-layer
		int8	count_include_pad:int64	count_include_pad/ 是否包含 pad 数值进行计算	1		
AveragePool	支持		kernel_shape [kernel_h,	kernel_h/ height 方向的 kernel 大小	无限制,NPU 支持		
			kernel_w]:int64[]	kernel_w/ width 方向的 kernel 大小	[1,]; 其它由 CPU 支 持。		
				pads_left/ left 方向的 pads 大小			
			pads[pads_top,pads_left,	pads_right/ right 方向的 pads 大小	[0.7]		
			pads_bottom, pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,7]		
				pads_bottom/ bottom 方向的 pads 大小			
			strides[strides_h,	stride_h/ height 方向的 strides 大小	[1 0]		
			strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,8]		



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/ 输入的 batch	1		
			input_tensor	channel/ 输入的 channel			
			[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]		
				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_h/ height 方向的 dilations 大小			
			dilations_w]:int64[] dilations_w/ widtht 方向的 dilations 大小		- 1		
MaxPool	支持	int8	kernel_shape [kernel_h,	kernel_h/ height 方向的 kernel 大小	无限制, NPU 支持	_	per-layer
			kernel_w]:int64[]	kernel_w/ width 方向的 kernel 大小	[1,7]; 其它由 CPU 支 持。		
				pads_left/ left 方向的 pads 大小			
			pads[pads_top,pads_left,	pads_right/ right 方向的 pads 大小	FO 73		
			pads_bottom, pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,7]		
				pads_bottom/ bottom 方向的 pads 大小			
			storage_order: int64	storage_order/优先储存方式	0		
			strides[strides h,	stride_h/ height 方向的 strides 大小	r1 01		
			strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,8]		



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
			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		
			momentum:double	momentum/ 训练时的滑动平均参数	无限制		per-layer/
Batch	 支持	int8		batch/ 输入的 batch	1		
Normalization	Normalization	float16	input_tensor [batch,channel,height,	channel/ 输入的 channel			per-channel
		width]:tensor	height/ 输入的 height	无限制			
			width/ 输入的 width				
				batch/ 输入的 batch	支持多 batch		
			input_tensor	channel/ 输入的 channel			
			[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		
				width/ 输入的 width			
			layernorm_weight [channel,height,	channel/输入的 channel	等于 input_channel		
			width]:tensor(const)	height/ 输入的 height	等于 input_height		
Layer Normalization	尚不支持	float16	layernorm_bias [channel,height, width]:tensor(const)	width/ 输入的 width	等于 input_width		per-layer
			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		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
				batch/ 输入的 batch	1		
Cl. /D TTIC	++±	int8	input_tensor	channel/ 输入的 channel			
Clip/ReLU6 支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer	
			width/ 输入的 width				
				batch/ 输入的 batch			
El	++±	int8	input_tensor	channel/ 输入的 channel	T://H /til		
Elu	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		
				width/ 输入的 width			
				batch/ 输入的 batch			
Gelu 支持	++±	int8	input_tensor	channel/ 输入的 channel	T://H /til		
	又持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		
				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
				batch/ 输入的 batch	1		
Relu	支持	int8	input_tensor	channel/ 输入的 channel			
Reiu		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer
				width/ 输入的 width			
	Relu 支持			batch/ 输入的 batch	1		
I 1 D 1		int8 float16	input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel	- 无限制 -		per-layer
LeakyRelu				height/ 输入的 height			
				width/ 输入的 width			
				batch/ 输入的 batch	1		
				channel/ 输入的 channel	[1 0102]		
PRelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]		per-layer/ per-channel
				width/ 输入的 width	[1,8176]		
				slope/PRelu 系数	仅支持单个标量或 C 维度系数	C	



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/ 输入的 batch	1		
			input_tensor [batch,channel,height, width]:tensor	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		
	尚不支持 GRU 扩 展以及变	RU 扩 以及变 命名为 GRU 算 ,参数 float16 中指明 extern) 项 为 GRU 独	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]		
GRU	exGRU 算		batch_size:int64 (extern)	batch_size/ 指定 GRU 输入的 batchsize	1		per-layer
GRO	项中指明 (extern)		sequence_size :int64 (extern)	sequence_size/ 指定 GRU 输入的 seqsize	无限制,建议4对齐		per-rayer
	的 项 为 exGRU 独		RU 独	hidden_size:int64 (extern)	hidden_size/ GRU 单元中的 hiddensize	- 1 1-1/H #1 73+1/1 0 74-7	
	有的参数项。		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							
			input 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。									
			[batch,channel,height, width]:tensor	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						
	郊八士柱		batch_size:int64 (extern)	batch_size/ 指定 LSTM 输入的 batchsize	大于1时仅支持4的倍数									
	部分支持 LSTM 扩		sequence_size :int64 (extern)	sequence_size/ 指定 LSTM 输入的 seqsize	无限制,建议4对齐									
	展以及变体命名为		hidden_size:int64 (extern)	hidden_size/ LSTM 单元中的 hiddensize	无限制,建议8对齐									
	exLSTM 算子,参	exLSTM 算子,参 数项中指 明 (extern) 的项为 exLSTM 触有的参	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		per-layer/							
LSTM	数项中指明		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		per-channel							
	(extern) 的项为		(extern)	(extern)		(extern)	的项为	的项为	的项为	has_dropout:int64 (extern)	has_dropout/ caffe 框架下的 indicator 功能的选择	I(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。		
	exLSTM 独有的参 数项。		has_projection:int64 (extern)	has_projection/ projection 变种	1(T) or 0(F) 目前限定 0, 即尚不支持									
	数≥ 火。		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
				batch/ 输入的 batch	channel 方向 concat		
			input_tensor [batch,channel,height,	channel/ 输入的 channel	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		
Concat	部分支持	int8 float16	width]:tensor	height/ 输入的 height	57°		per-layer
				width/ 输入的 width	制。		
			axis:int64	aixs/ 拼接的维度	无限制		
		持 lnt8 [batch,ch		batch/ 输入的 batch			
W. I	++4		input_tensor	channel/ 输入的 channel	丁. 17月 休山		
Mish	支持		[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		
				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
				batch/ 输入的 batch	1		
		int8	input_tensor	hannel/ 输入的 channel	7 17 14 1		
Pad 支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制			
			width/ 输入的 width	[1,8176]			
	2.19	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 l,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		
				batch/ 输入的 batch			
			input_tensor	channel/ 输入的 channel	- 无限制		
ReduceMean 尚不支持 目由 CPU 实现	尚不支持 目前	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	7 无限制		
	由 CPU 实现	float16		width/ 输入的 width			
		axes:int6-	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		数据类型	输入	 输入参数		广播支持	量化方式	
	×14 19 00		- Tuny V		> 1 / 10 / 20 1 H	, 1m ~ 10	至10.7.7	
			input tensor	batch/ 输入的 batch				
ReduceSum 尚不支持 目前由 CPU 实现				channel/ 输入的 channel				
	出不支持 目前	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer/	
	float16		width/ 输入的 width			per-channel		
			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			
					batch/ 输入的 batch	支持多 batch		
	部分支持			channel/ 输入的 channel	[1 0102]			
	目前 NPU 仅支 持宽高方向不		input_tensor [batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]			
Resize 超过数的线性放弃	超过8倍的整倍数的最近邻和线性插值缩放,其余不支持	int8 float16		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		per-layer	
	部分的会 Fallback到CPU	」 会	mode:string	mode/resize 采用的模式	仅支持 nearest、linear			
	上实现。		scales:int64[]	scales/尺寸放大倍数	仅支持 1-8 整数倍			
	1271700		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
				batch/ 输入的 batch			
		int8	input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel	约束规格: 1. height * width * type bytes <=		
		float16	(input_tensor 的维度为4 维时看作 nchw)	height/ 输入的 height	130816 2. input_tensor 非四维时, shape 无限制		
Reshape	部分支持	width/ 输入的 width					
Resnape	时为 文44		Shape (batch_o,channel_o, height_o,width_o):tensor	batch_o/ 输出的 batch_o	计算量: alignment=16/type bytes;		
		int64		channel_o/ 输出的 channel	约束规格: 1. height_o * width_o * type_bytes <=		
			(输出 shape 指定维度为 4 维时看作 nchw)	height_o/ 输出的 height	65535; 2. Align(height_o * width_o, alignment) <= 8192;		
				width_o/ 输出的 width	3. 输出 shape 非四维时,shape 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
				batch/ 输入的 batch			
			input_tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制		
			width]:tensor	height/ 输入的 height	ניקי אין אין		
Reverse Sequence	尚不支持	int8 float16		width/ 输入的 width			
			batch_axis:int64	batch_axis/ 指定是否为 batch 维度	0		
			time_axis:int64	time_axis/ 指定是否为 time 维度			
			sequence_lens:int64[]	sequence_lens/ 指定序列翻转的数量	仅支持 channel 数		
				batch/ 输入的 batch			
Sigmoid	支持	int8 float16	input_tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制		
Signioid	XII		width]:tensor	height/ 输入的 height	ן יין אַאָּקטע		
				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
				batch/ 输入的 batch			
H40''4	支持	int8	input_tensor	channel/ 输入的 channel	工阳和		
HardSigmoid	Transigniord X17	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer
			width/ 输入的 width				
				batch/ 输入的 batch			
	+++	int8	input_tensor	channel/ 输入的 channel	T. 175 Au		
Swish	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		
				width/ 输入的 width			
				batch/ 输入的 batch			
H 10 11	+-++:	支持 Into Into	input_tensor	channel/ 输入的 channel	T. VEL ALI		
HardSwish 3	文持		[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer
				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
				batch/ 输入的 batch			
	支持	int8	input_tensor	channel/ 输入的 channel	- - 无限制		n on loven
Softplus	X4	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	一人吃咖		per-layer
				width/ 输入的 width			
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel	硬件支持[1,8192]		
Softmax	支持	int8 float16		height/ 输入的 height	axis=1, 无限制 axis=3/-1,width[1, - 8192], height 无限制		per-layer
				width/ 输入的 width	且受限于 tranpose 的规格限制		
			axis:int64	axis/ 做 softmax 的轴	1,3, 即 channel 和 width 方向	idth	



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/ 输入的 batch				
			input_tensor	channel/ 输入的 channel				
			[batch,channel,height, width]:tensor	height/ 输入的 height				
			width/ 输入的 width					
Slice	部分支持	int8 float16	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。	;	per-layer	
			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			
				batch/ 输入的 batch				
				input_tensor	channel/ 输入的 channel			
			[batch,channel,height, width]:tensor	height/ 输入的 height	无限制			
Split	部分支持	int8 float16		width/ 输入的 width	一 儿限制		per-layer	
		1104110	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
				batch/ 输入的 batch			
T. 1		int8	input_tensor	channel/ 输入的 channel	工作品		1
Tanh	floring (Signature)	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		per-layer
			width/ 输入的 width				
				batch/ 输入的 batch	无限制		
		input_tensor [batch,channel,h width]:tensor		channel/ 输入的 channel	[1 0102]		
						height/ 输入的 height	[1,8192]
				width/ 输入的 width	[1,8176]		
Transpose	部分支持	int8 float16	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
				batch/ 输入的 batch channel/ 输入的 channel	无限制		
			input_tensor [batch,channel,height, width]:tensor	制入的 channel height/ 輸入的 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			
			num_output/ 输出的 channel num input/	- 无限制			
		kernel_shape [num_output, num_input,kernel_h,	mum_input/ 输入的 channel kernel h/				
			kernel_w]:int64[]	height 方向的 kernel 大小 kernel w/	[1,31]		
Convolution	支持	int8	strides[strides_h, strides_w]:int64[]	width 方向的 kernel 大小 stride_h/ height 方向的 strides 大小 stride_w/	[1,7]		per-layer/ per-channel
				width 方向的 strides 大小 pads_left/ left 方向的 pads 大小			
			pads[pads_top, pads_left, pads_bottom,	pads_right/ right 方向的 pads 大小 pads_top/	[0,15]		
		pads_right]:int64[]	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 dilations_w/ widtht 方向的 dilations 大小	[1, 32]		



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
		input_tensor [batch,channel,height, width]:tensor	[batch,channel,height,	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height	无限制 当 dilation kernel h > 1 时,		
				width/ 输入的 width	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/ 输出的 channel num_input/ 输入的 channel	- 无限制		
Depthwise	支持	int8	[num_output, num_input,kernel_h, kernel_w]:int64[]	M Channel kernel_h/ height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小	[1,8]		per-layer/
Convolution			strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	- [1,7]		per-channel
		pads_right]:int64[]	pads_left, pads_bottom,	pads_left/ left 方向的 pads 大小 pads_right/ right 方向的 pads 大小 pads_top/ top 方向的 pads 大小 pads_bottom/	[0,15]		
			dilations[dilations_h, dilations_w]:int64[]	bottom 方向的 pads 大小 dilations_h/ height 方向的 dilations 大小 dilations_w/ widtht 方向的 dilations 大小	[1, 32]		



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
			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 当 <u>dilation kernel h</u> > 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_output/ 输出的 channel num_input/ 输入的 channel	- 无限制		
		int8 stride stride pads[pads_ pads_ pads_ dilation	num_input,kernel_h,kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小	- [1,31]		
ConvTranspose/ Deconvolution	支持		strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	[1,8]		per-layer/ per-channel
			pads[pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小 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	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		
			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 dilations_w/ widtht 方向的 dilations 大小	- [1, 32]		



_	t . t.t. t.t. y	No. 111 No. 111	44.5	KV1103/1100 IVI U OI 文項/		加心版电 7 放[
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
			input_tensor_1 [M, K]:tensor	M,K,N/			
		input_tensor_2 [K,N]:tensor	输入数据的形状	转为 Matmul 实现,约束同 Matmul			
Comm	尚不支持 目 前 由 CPU 实现	:40	alpha:double	alpha/ 矩阵 A*B 乘法的 scale	无限制		per-layer/
Gemm		PU 实现 —	beta:double	beta/ 输入 C 矩阵的 scale	人内区即		per-channel
			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	────────────────────────────────────		
			input_tensor_1	batch/ 输入的 batch	双 feature 时: batch、H 无限制		
MatMul	尚不支持 MatMul 目前由		[batch, K, C]:tensor	K/ 输入的 K	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		per-layer/
CPU 实现	int8	input_tensor_2	C/ 输入的 C	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		per-channel	
			[batch, C, H]:tensor	H/ 输入的 H	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		



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/ 输入的 batch			
		int8	input_tensor [batch,channel,height,	channel/ 输入的 channel			
		float16	width]:tensor	height/ 输入的 height	- 无限制		
Expand	支持			width/ 输入的 width			
LAPANG	ZN		shape(batch_o,	batch_o/ 输出的 batch_o	和nnel 无限制		
		int64		channel_o/ 输出的 channel			
		III.04	channel_o,height_o, width_o):tensor	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
		int8		batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width			
	Where 支持			batch/ 输入的 batch channel/	无限制		
Where		mask_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	- 无限制		
	shape(batch_o, channel_o,height_o, width_o):tensor		batch_o/ 输出的 batch_o channel_o/ 输出的 channel height_o/ 输出的 height width_o/ 输出的 width	无限制	- 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
				batch/ 输入的 batch	无限制		
			input_tensor_1[batch, channel,height,width]:tenso	channel/ 输入的 channel	硬件支持[1,8192]		
			r	height/ 输入的 height	axis=1,无限制 - axis=3/-1,width[1, 8192], height 无限制		
				width/ 输入的 width	且受限于 transpose 规格限制	_	per-layer
exSoftmaxMask	部分		3	batch/ 输入的 batch	1		
exsorumaxiviask	支持	float16	input_tensor_2[batch, channel,height,width]:tenso	channel/ 输入的 channel	硬件支持[1,8192]		
		axis:int64	<u> </u>	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
		int8 float16		batch/ 输入的 batch			
			x_tensor [batch, channel,	channel/ 输入的 channel	c*h*w 满足如下限制		
exGlu	支持		height, width]:tensor	height/ 输入的 height	8bit 数据: 8 对齐, 16bit 数据: 4 对齐		per-layer
exGlu				width/ 输入的 width			per-rayer
		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						
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	同 Convolution					
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						
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	同 ConvTranspose					
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						



				KV1105/1100 IVI O OI XII 911X		机心脉电 1 ル	
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						
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	同 Depthwise Con	volution				
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	J 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
				batch/ 输入的 batch			
		int8	input tensor	channel/ 输入的 channel		支持 ONNX 规范的四维 tensor 的所有	1/
Add/Bias	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制	广播操作,详见: 注释(1)	per-layer/ per-channel
				width/ 输入的 width			
				batch/ 输入的 batch		支持两个 tensor 的广播操作,以 ONNX	
Cul	支持	channel/ 输入的 channel	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;	per-layer/			
Sub	又付	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制	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	per-channel
				width/ 输入的 width		子见: <u>注释(1)</u>	



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/ 输入的 batch			
Mul/Scale		int8	input_tensor	channel/ 输入的 channel		支持 ONNX 规范的四维 tensor 的所有	per-layer/
	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-channel
				width/ 输入的 width			
				batch/ 输入的 batch		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:	
			input tensor	channel/ 输入的 channel		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	
Div	部分支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	做 broadcasting; 4.OP(A(N,C,H,W),B(H,W)),即 HW 维度做 broadcasting,目前仅支持 FP1。 类型。	3.OP((N,C,H,W),scalar),即以单个标量	per-layer/ per-channel
				width/ 输入的 width		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	



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/ 输入的 batch		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	
Max	暂不支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制	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;	per-layer/
IVIAX	自小义行	float16	width]:tensor	height/ 输入的 height	プレドス 中 り	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。	per-channel
						说明: A 或 B 都可以作为广播方。例 子见: <u>注释(1)</u>	
				batch/ 输入的 batch		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	
Min	暂不支持	int8 input tensor	input_tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制		per-layer/
IVIIII		float16 [batch, channer, height, width]:tensor		height/ 输入的 height	Jupa in	做 broadcasting; 3.OP(A(N,C,H,W),B(scalar)),即以单个 标量做 broadcasting。	per-channel
				width/ 输入的 width		说明: A 或 B 都可以作为广播方。例 子见: <u>注释(1)</u>	



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/ 输入的 batch	1		
Global	才 楼	0	input_tensor	channel/ 输入的 channel	[1,8192]		
AveragePool	支持	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		per-layer
				width/ 输入的 width	[1,7] (Complier 支持 范围)		
				batch/ 输入的 batch	1		
ClabalManDani	才 快	int8	input_tensor	channel/ 输入的 channel	[1,8192]		
GlobalMaxPool	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		per-layer
				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
				batch/ 输入的 batch	1		
			input_tensor	channel/ 输入的 channel			
			[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]		
			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		
		int8	count_include_pad:int64	count_include_pad/ 是否包含 pad 数值进行计算	1		
AveragePool	支持		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		per-layer
				kernel_w/ width 方向的 kernel 大小] [1,/]; 共占田 CPU 支持。 		
				pads_left/ left 方向的 pads 大小			
			pads[pads_top,pads_left,	pads_right/ right 方向的 pads 大小	[0.7]		
			pads_bottom, pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,7]		
				pads_bottom/ bottom 方向的 pads 大小			
		strides[strides_h,	stride_h/ height 方向的 strides 大小	[1 0]			
			strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,8]		



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/ 输入的 batch	1		
			input_tensor [batch,channel,height,	channel/ 输入的 channel			
			width]:tensor	height/ 输入的 height	[1,8192]		
				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_h/ height 方向的 dilations 大小			
			dilations_w]:int64[]	dilations_w/ widtht 方向的 dilations 大小	1		
MaxPool	支持	int8 float16	kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小	无限制, NPU 支持		per-layer
				kernel_w/ width 方向的 kernel 大小	[1,7]; 其它由 CPU 支 持。		
				pads_left/ left 方向的 pads 大小			
			pads[pads_top,pads_left,	pads_right/ right 方向的 pads 大小	[O 7]		
		F	pads_bottom, pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,7]		
			pads_bottom/ bottom 方向的 pads 大小				
			storage_order: int64	storage_order/优先储存方式	0		
		strides[strides_h,	stride_h/ height 方向的 strides 大小	[1,8]			
			strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,0]		



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
			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		
			momentum:double	momentum/ 训练时的滑动平均参数	无限制		
Batch	支持	int8		batch/ 输入的 batch	1		per-layer/
Normalization	20,7	float16	input_tensor [batch,channel,height,	channel/输入的 channel			per-channel
			width]:tensor	height/ 输入的 height	无限制		
				width/ 输入的 width			
				batch/ 输入的 batch	支持多 batch		
			input_tensor [batch,channel,height,	channel/ 输入的 channel			
			width]:tensor	height/ 输入的 height	无限制		
				width/ 输入的 width			
			layernorm_weight [channel,height,	channel/ 输入的 channel	等于 input_channel		
			width]:tensor(const) layernorm bias	height/输入的 height	等于 input_height		
Layer	支持	float16	[channel,height, width]:tensor(const)	width/ 输入的 width	等于 input_width		per-layer
Normalization			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
				batch/ 输入的 batch			
Cii /D III/	eLU6 支持 int8 float	int8	input_tensor	channel/ 输入的 channel	T 70 4d		
Clip/ReLU6		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer
				width/ 输入的 width			
				batch/ 输入的 batch			
El	支持	int8	input_tensor	channel/ 输入的 channel	丁. 17日 休山		
Elu	又行	float16	float16 [batch,channel,height, width]:tensor	height/ 输入的 height	无限制		
				width/ 输入的 width			
				batch/ 输入的 batch			
	+-++	int8	input_tensor	channel/ 输入的 channel	T.179.44		
Gelu	支持 float16	•		height/ 输入的 height	无限制		
				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
				batch/ 输入的 batch	1		
Dalu		int8	input_tensor	channel/ 输入的 channel			per-layer
Kelu		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer
				width/ 输入的 width			
				batch/ 输入的 batch	1		
I 1 D 1	+++	int8	input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel			
LeakyRelu	支持	float16		height/ 输入的 height	无限制		per-layer
				width/ 输入的 width			
				batch/ 输入的 batch	1		
				channel/ 输入的 channel			
PRelu	支持 int8 float10	int8 float16	input_tensor [batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer/ per-channel
		widthj.tensor	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
				batch/ 输入的 batch	1		
			input_tensor [batch,channel,height, width]:tensor	sequence/ 输入的 sequence	限制 4 对齐		
			Widonjitenser	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		
	部分支持 GRU 扩 展以及变 体命名为		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]		
GRU	exGRU 算 子,参数	float16	batch_size:int64 (extern)	batch_size/ 指定 GRU 输入的 batchsize	1		per-layer
	项中指明 (extern) 的 项 为		sequence_size :int64 (extern)	sequence_size/ 指定 GRU 输入的 seqsize	限制 4 对齐		
	exGRU 独 有的参数		hidden_size:int64 (extern)	hidden_size/ GRU 单元中的 hiddensize	限制 8 对齐		
	项。		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。		

瑞芯微电子股份有限公司



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
			input 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。		
			[batch,channel,height, width]:tensor	sequence/输入的 sequence	无限制,建议4对齐		
			widthj.tensor	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		
	部分支持		batch_size:int64 (extern)	batch_size/ 指定 LSTM 输入的 batchsize	大于1时仅支持4的倍数		
	部分又持 LSTM 扩		sequence_size :int64 (extern)	sequence_size/ 指定 LSTM 输入的 seqsize	无限制,建议4对齐		
	展以及变体命名为		hidden_size:int64 (extern)	hidden_size/ LSTM 单元中的 hiddensize	无限制,建议8对齐		
	exLSTM 算子,参	int8	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		per-layer/
LSTM	数项中指明	float16	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		per-channel
	(extern) 的项为		has_dropout:int64 (extern)	has_dropout/ caffe 框架下的 indicator 功能的选择	I(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。		
	exLSTM 独有的参 数项。		has_projection:int64 (extern)	has_projection/ projection 变种	1(T) or 0(F) 目前限定 0,即尚不支持		
	数 /贝。		input_layout:string (extern)	input_layout/指定与对应输入 shape含义一致的 layout	I.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
				batch/ 输入的 batch	channel 方向 concat		
	部分支持		input_tensor	channel/ 输入的 channel	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		
Concat		int8 float16	[batch,channel,height, width]:tensor	height/ 输入的 height	据: 8 对齐 据: 4 对齐 其他方向 concat 无限		per-layer
				width/ 输入的 width	制。		
			axis:int64	aixs/ 拼接的维度	无限制		
				batch/ 输入的 batch			
Mish	支持	int8	input_tensor	channel/ 输入的 channel	无限制		
IVIISII		[batch,channel,height, width]:tensor	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			batch/ 输入的 batch	1				
		int8 float16	input_tensor	hannel/ 输入的 channel	- 无限制				
			[batch,channel,height, width]:tensor	height/ 输入的 height	1 人取 中川				
Pad				width/ 输入的 width	[1,8176]				
Tuu		文 拉	7 10	2.19	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				
				batch/ 输入的 batch					
			input_tensor	channel/ 输入的 channel	T. 70 4.1				
D - 4 M	尚不支持 目前由 CPU 实现	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制				
ReduceMean		float16		width/ 输入的 width					
			axes:int64[]	axes/ 指定 reduce 的轴	单轴:无限制, 多轴:{2,3}				
		keepdims		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
				batch/ 输入的 batch			
	ReduceSum 尚不支持 目前 由 CPU 实现		input_tensor	channel/ 输入的 channel			
		int8	[batch,channel,height, width]:tensor	height/ 输入的 height	─ 无限制		per-layer/
ReduceSum				width/ 输入的 width			per-channel
			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		
		分支持		batch/ 输入的 batch	支持多 batch		
	部分支持		input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel	[1 0102]		
	目前 NPU 仅支			height/ 输入的 height	[1,8192]		
Resize	0170	持宽高方向不 超过 8 倍的整倍 int8 数 的 最 近 邻 和 float16 线性插值缩放, 其 余 不 支 持 部 分的会 Fallback 到 CPU 上实现。	可不 整倍 int8 B和 float16	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		per-layer
			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
				batch/ 输入的 batch			
		int8	input_tensor	channel/ 输入的 channel	约束规格: 1. height * width * type bytes <=		
		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	8192*8192*16; 2.input_tensor 非四维时,shape 无限制		
Reshape	支持			width/ 输入的 width			
Kesnape	X14	Shap int64 (bate	Shape	batch_o/ 输出的 batch_o	计算量: alignment=16/type bytes;		
				channel_o/ 输出的 channel	约束规格: 1.height_o * width_o * type_bytes <=		
			(batch_o,channel_o, height_o,width_o):tensor	height_o/ 输出的 height	INT32_MAX; 2.Align(height_o * width_o, alignment) <= 8192*8192;		
				width_o/ 输出的 width	3.输出 shape 非四维时,shape 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	
				batch/ 输入的 batch				
			input_tensor	channel/ 输入的 channel	 			
			[batch,channel,height, width]:tensor	height/ 输入的 height				
Reverse Sequence		int8 float16		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			
		支持 mio			batch/ 输入的 batch			
Sigmaid	a: :1 +++		input_tensor [batch,channel,height,	channel/ 输入的 channel	无限制			
Sigmoid	文 打		width]:tensor	height/ 输入的 height	ייח אין טע			
				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
				batch/ 输入的 batch			
1140::4	HardSigmoid 支持	int8	input_tensor	channel/ 输入的 channel	工阳和		
HardSigmoid		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer
			width/ 输入的 width				
		int8 input_tensor [batch,channel width]:tensor		batch/ 输入的 batch			
	+++			channel/ 输入的 channel	丁.17日 休山		
Swish	文持			height/ 输入的 height	无限制		
				width/ 输入的 width			
				batch/ 输入的 batch			
H. 10 . 1		int8	input_tensor	channel/ 输入的 channel	工程和		
HardSwish 支持	文持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer
				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
				batch/ 输入的 batch			
S. frahus	士 ·	int8	input_tensor	channel/ 输入的 channel	- 无限制		n on loven
Solipius	支持 支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	7. 万字 中,		per-layer
				width/ 输入的 width			
			input_tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel	硬件支持[1,8192]		
Softmax	支持	支持 int8 float16 [batch,channel,height, width]:tensor axis:int64	height/ 输入的 height	axis=1,无限制 axis=3/-1,width[1, - 8192], height 无限制		per-layer	
				width/ 输入的 width	且受限于 tranpose 的规格限制		
			axis:int64	axis/ 做 softmax 的轴	1,3,即 channel 和 width 方向		



			RKS302 N O OT XIII/III					
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/ 输入的 batch				
			input_tensor	channel/ 输入的 channel	T 114 4 d			
					height/ 输入的 height	无限制		
Slice 部分支持	int8	int8	width/ 输入的 width					
	float16	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。		per-layer		
		ends:int64[] ends/ channel 方向 Slice 时,channel rends:int64[] bn / channel rends / Sbit 数据: 8 对齐,16bit 数:	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			
				batch/ 输入的 batch				
				input_tensor	channel/ 输入的 channel			
			[batch,channel,height, width]:tensor	height/ 输入的 height	无限制			
Split 部分支持	int8 float16		width/ 输入的 width	/LPK中J		per-layer		
	axis:int64	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
				batch/ 输入的 batch			
Tanh	lonh D	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel			per-layer
Tum		float16	width]:tensor	height/ 输入的 height			per layer
			width/ 输入的 width				
		input_tensor [batch,channel,height, width]:tensor int8 float16 perm:int64[]	batch/ 输入的 batch	无限制			
			[batch,channel,height,	channel/ 输入的 channel			
				height/ 输入的 height	[1,8192]		
				width/ 输入的 width	[1,8176]		
Transpose	支持		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
	input_tensor [batch,channel,heigh width]:tensor		batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height	无限制 <u>当 dilation_kernel_h</u> >1 时, width < 16383			
				输入的 width num output/	此外,对首层输入 width 存在限制,详见模型输入说明		
	[1]	kernel_shape [num_output,	输出的 channel num_input/ 输入的 channel				
		num_input,kernel_h,kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小	[1,31]			
Convolution	支持	int8	strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	[1,7]		per-layer/ per-channel
			pads[pads_top,	pads_left/ left 方向的 pads 大小 pads_right/ right 方向的 pads 大小	[0,15]		
	_	pads_left, pads_bottom, pads_right]:int64[]	pads_top/ top 方向的 pads 大小 pads_bottom/ bottom 方向的 pads 大小	[6,10]			
		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 dilations_w/ widtht 方向的 dilations 大小	[1, 32]			



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
		input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height	无限制 当 dilation kernel h > 1 时,			
				width/ 输入的 width	width < 16383 此外,对首层输入 width 存 在限制,详见		
		kernel_w]:int64[]		num_output/ 输出的 channel num_input/ 输入的 channel	一 无限制		
Depthwise Convolution	支持		num input,kernel h,	kernel_h/ height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小	[1,8]		per-layer/ per-channel
Convolution				stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	[1,7]		per-enamer
			pads_left, pads_bottom,	pads_left/ left 方向的 pads 大小 pads_right/ right 方向的 pads 大小 pads_top/ top 方向的 pads 大小	[0,15]		
				pads_bottom/ bottom 方向的 pads 大小 dilations h/			
			dilations[dilations_h, dilations_w]:int64[]	height 方向的 dilations 大小 dilations_w/ widtht 方向的 dilations 大小	[1, 32]		



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	
			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 当 <u>dilation_kernel_h</u> > 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_output/ 输出的 channel num_input/ 输入的 channel	- 无限制				
ConvTranspose/		正持 int8	num_input,kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小	- [1,31]		per-layer/	
Deconvolution	支持		strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	- [1,8]		per-channe	
			pads[pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小 pads_right/ right 方向的 pads 大小 pads_top/ top 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;			
			group:int64	pads_bottom/ bottom 方向的 pads 大小 group/ group 的大小	不支持 stride_w *(width - 1) - pads_left + 1 < output_w 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 dilations_w/ widtht 方向的 dilations 大小	- [1, 32]			



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
			input_tensor_1 [M, K]:tensor	M,K,N/	转为 Matmul 实现,约束同 Matmul		
			input_tensor_2 [K,N]:tensor	输入数据的形状	表为 Matmut 实现,约来问 Matmut		
Gomm	Gemm 尚不支持 目前由 CPU 实现	int8	alpha:double	alpha/ 矩阵 A*B 乘法的 scale	- 无限制		per-layer/
Geniin		beta:double transA:int64	beta:double	beta/ 输入 C 矩阵的 scale	יה אין אך אך אך אך אך אך אך אין אין אר		per-channel
			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	人们心 (1800) 人门八直		
	部分支持	input_tensor_1[b		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		
	目前该支持仅		channel, K, N]:tensor	channel/ 输入的 channel	对齐, 16bit 数据: 8 对齐 C 支持[32,19384], 对齐要求为 32 对齐		
未来将支持输入 为	int8		K/ 输入的 K	feature+constant 时: 若 input_tensor_1 为 feature,则转为 batch		per-layer/ per-channel	
	为 channel, N, N		N/ 输入的 M	个 feature[K,C,1,1] + weight[H,C,1,1] 的 conv; 若 input_tensor_2 为 feature,则转为 batch			
	feature+constant			M/ 输入的 M	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		



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/ 输入的 batch			
		int8	input_tensor [batch,channel,height,	channel/ 输入的 channel			
		float16 loa wi	width]:tensor	height/ 输入的 height	- 无限制		
Expand	支持			width/ 输入的 width			per-layer
Expand	XN		shape(batch_o,	batch_o/ 输出的 batch_o	- 无限制		pr. my.
				channel_o/ 输出的 channel			
	int64 channe width_	channel_o,height_o, width_o):tensor	height_o/ 输出的 height	- Zupa ipi			
				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 channel/ 输入的 channel height/ 输入的 height width/ 输入的 width batch/ 输入的 batch	无限制			
	int8 float16 int64	y_tensor [batch, channel, height, width]:tensor	channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	无限制		per-layer	
	又行	bool channe	mask_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	无限制	per layer
		int8 float16	shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o 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
				batch/ 输入的 batch	无限制		
			input_tensor_1[batch, channel,height,width]:tenso	channel/ 输入的 channel	硬件支持[1,8192]		
			r	height/ 输入的 height	axis=1, 无限制		
				width/ 输入的 width	axis=3/-1,width[1, 8192], height 无限制 且受限于 transpose 规格限制		
exSoftmaxMask	部分	int8 float16	input_tensor_2[batch, channel,height,width]:tensor	batch/ 输入的 batch	1		man layyan
exSolullaxiviask	支持			channel/ 输入的 channel	硬件支持[1,8192]		per-layer
				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	輸入	输入参数	约束规格	广播支持	量化方式
				batch/ 输入的 batch			
		int8	x_tensor [batch, channel,	channel/ 输入的 channel	c*h*w 满足如下限制	c*h*w 满足如下限制8bit 数据: 4 对齐	
exGlu	支持	float16	height, width]:tensor	height/ 输入的 height	8bit 数据: 8 对齐, 16bit 数据: 4 对齐		mon layon
exGiu	大村			width/ 输入的 width			per-layer
		int64	axis:int64	axis/ 切分的维度	axis ==1		
		持 int8 float16	input_tensor_1[batch,	batch/ 输入的 batch			
ov Mot Mul	古扶		K,1, N]:tensor	K/ 输入的 K	K 支持[1,8192]		man layan
exMatMul 支持	文村		input tensor 25batch K	M/ 输入的 M	■ K 又打[1,0192]		per-layer
			input_tensor_2[batch, K, -1, M]:tensor	N/ 输入的 N			



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/ 输入的 batch			
		float16	query_tensor	channel/ 输入的 channel	- Channel < 8192		
		noatro	[batch,channel,height, width]:tensor	height/ 输入的 height	Chamier < 8192		
				width/ 输入的 width			
exSDPAtten	支持	寺 float16	key_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch			
tion	大 14			channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
		float16 [batch	value_tensor [batch,channel,height,	batch/ 输入的 batch	- Width < 8192		
			width]:tensor	channel/ 输入的 channel	- widin < 8192		

Rockchip ^{瑞芯微电子}

			height/ 输入的 height		
			width/ 输入的 width		
		mask_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch		
	float16		channel/ 输入的 channel		
			height/ 输入的 height		
			width/ 输入的 width		
	float32	scale			



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						
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	同 Convolution					
Convolution + Softplus	支持						
Convolution + HardSigmoid	支持						
Convolution + HardSwish	支持						
Convolution + Elu	支持						
Convolution + Swish	支持						
Convolution + Mish	支持						

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
ConvTranspose + Relu	尚不支持						
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	同 ConvTranspose					
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						



	L. Di. Edusies	stat time site me.it		PA > 6 MB			HA N
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						
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	同 Depthwise Con	volution				
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	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 RK3576 NPU OP 支持列表

瑞芯微电子股份有限公司



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/ 输入的 batch				
Add/Bias 支持			input_tensor	channel/ 输入的 channel				
	int8 float16	[batch,channel,h eight, width]:tensor	height/ 输入的 height	无限制	支持 ONNX 规范的四维 tensor 的 所有广播操作,详见: <u>注释(1)</u>	per-layer/ per-channel	尚不支持	
				width/ 输入的 width				
			input_tensor [batch,channel,h eight, 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,	per-layer/ per-channel	尚不支持
	-+-4-	int8		channel/ 输入的 channel		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)		
Sub 支持	又行	float16		height/ 输入的 height				
				width/ 输入的 width				

瑞芯微电子股份有限公司



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/ 输入的 batch				
Mul/Scale 支持	-t-14-	int8	input_tensor	channel/ 输入的 channel	T 170 day	支持 ONNX 规范的四维 tensor 的所有	per-layer/ per-channel	尚不支持
	文持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	一 无限制	广播操作,详见: <u>注释(1)</u>		
				width/ 输入的 width				
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch		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				
Div	部分支持	float16		height/ 输入的 height	无限制			
				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
			input_tensor [batch,channel,height , width]:tensor	batch/ 输入的 batch		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:		尚不支持
Mov	暂不支持	int8		channel/ 输入的 channel	- 无限制	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;	per-layer/ per-channel	
Max	百个义抗	float16		height/ 输入的 height		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)		
				width/ 输入的 width				
		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
Min	暂不支持			channel/ 输入的 channel				
Nilli	H I Z I			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
				batch/ 输入的 batch	1			
Global	士 杜	:40	input_tensor	channel/ 输入的 channel	[1,8192]		1	业 不士柱
AveragePool	支持	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		per-layer	同小文行
				width/ 输入的 width	[1,7] (Complier 支持 范围)			尚不支持
				batch/ 输入的 batch	1			
ClabalMayDaal		int8	input_tensor	channel/ 输入的 channel	[1,8192]		and laves	光 不去性
GlobalMaxPool 支持	文 村	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		per-layer	问 个又行
				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
				batch/ 输入的 batch	1			
			input_tensor	channel/ 输入的 channel				
			[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]			
				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			
AveragePool	支持	int8	kernel shape [kernel h,	kernel_h/ height 方向的 kernel 大小	无限制,NPU 支持		per-layer	尚不支持
			kernel_w]:int64[]	kernel_w/ width 方向的 kernel 大小	[1,7]; 其它由 CPU 支持。		per-layer	
				pads_left/ left 方向的 pads 大小				
			pads[pads_top,pads_left,	pads_right/ right 方向的 pads 大小	[0.7]			
			pads_bottom, pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,7]			
				pads_bottom/ bottom 方向的 pads 大小				
			strides[strides_h,	stride_h/ height 方向的 strides 大小	[1 0]	1/1		
			strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,8]			



				KK3370 NI O OI 文语列表				
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
				batch/ 输入的 batch	1			
			input_tensor	channel/ 输入的 channel				
			[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]			
				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_h/ height 方向的 dilations 大小					
			dilations_w]:int64[]	dilations_w/ widtht 方向的 dilations 大小	1			
MaxPool	支持	int8 float16	kernel shape [kernel h,	kernel_h/ height 方向的 kernel 大小	无限制, NPU 支持		per-layer	尚不支持
			kernel_w]:int64[]	kernel_w/ width 方向的 kernel 大小	[1,7]; 其它由 CPU 支 持。			
				pads_left/ left 方向的 pads 大小				
			pads[pads_top,pads_left,	pads_right/ right 方向的 pads 大小	F0.73			
			pads_bottom, pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,7]			
				pads_bottom/ bottom 方向的 pads 大小				
			storage_order: int64	storage_order/优先储存方式	0			
			strides[strides h,	stride_h/ height 方向的 strides 大小	[1.0]			
			strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,8]			



				KK3370 NI U OI X1	****																
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													
			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 le-5																
			momentum:double	momentum/ 训练时的滑动平均参数	无限制																
Batch Normalizati	 支持	int8		batch/ 输入的 batch	1		per-layer/	多核协同 尚不支持 尚不支持													
on	211	float16	input_tensor [batch,channel,height,	channel/ 输入的 channel			per-channel														
			width]:tensor	height/ 输入的 height	无限制																
				width/ 输入的 width																	
				batch/ 输入的 batch	支持多 batch																
				input_tensor [batch,channel,height,	channel/ 输入的 channel																
			width]:tensor	height/ 输入的 height	无限制																
				width/ 输入的 width																	
			layernorm_weight [channel,height,	channel/ 输入的 channel	等于 input_channel																
			width]:tensor(const) layernorm bias	height/ 输入的 height	等于 input_height																
Layer	+++	G 416	[channel,height, width]:tensor(const)	width/ 输入的 width	等于 input_width		1	业子士士													
on	Normalizati 支持 1	float16	float16	float16	float16	float16	float16	float16	float16	float16	float16	float16	float16	float16	float16	normalized_shape:int6 4[]	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。		per-layer	阿个义持
			elementwise_affine:int 64	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)。																
	1	1	1	1/1/10 110111101112 1/C3-X1 1/11 1/11 1/11 1/11 1/11 1/11 1/11 1	_ · · ································	I	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
				batch/ 输入的 batch				
	-+	int8	input_tensor [batch,channel,heig	channel/ 输入的 channel				W.T.+.+
Clip/ReLU6	支持	float16	ht, width]:tensor	height/ 输入的 height	- 无限制		per-layer	多核协同 尚不支持 尚不支持
				width/ 输入的 width				
				batch/ 输入的 batch				
EL	±44	int8	input_tensor [batch,channel,heig	channel/ 输入的 channel	工程出			业 不士柱
Elu	支持	float16	ht, width]:tensor	height/ 输入的 height	- 无限制		per-layer	向小又 持
				width/ 输入的 width				
				batch/ 输入的 batch				
0.1	in in	int8	input_tensor [batch,channel,heig	channel/ 输入的 channel	- 			W T + ++
Gelu	支持	float16	ht, width]:tensor	height/ 输入的 height	- 无限制		per-layer	尚不支持
				width/ 输入的 width				

瑞芯微电子股份有限公司



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/ 输入的 batch	1			
Relu	支持	int8	input_tensor [batch,channel,hei	channel/ 输入的 channel			per-layer	多核协同 尚不支持 尚不支持
Ketu		float16	ght, width]:tensor	height/ 输入的 height	无限制		per-rayer	间小文诗
				width/ 输入的 width				
				batch/ 输入的 batch	1			
	1 1 D 1 +++	int8	input_tensor [batch,channel,hei	channel/ 输入的 channel				尚不支持
LeakyRelu	支持	float16	ght, width]:tensor	height/ 输入的 height	- 		per-layer	
				width/ 输入的 width				
				batch/ 输入的 batch	1			
			input tensor	channel/ 输入的 channel				
PRelu	支持	int8 float16	[batch,channel,hei ght,	height/ 输入的 height	无限制		per-layer/ per-channel	尚不支持
			width]:tensor	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	多核协同
				batch/ 输入的 batch	1			
			input_tensor [batch,channel,height, width]:tensor	sequence/ 输入的 sequence	限制 4 对齐			
			widthj.tensor	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]			
GRU	暂不支持	float16	batch_size:int64 (extern)	batch_size/ 指定 GRU 输入的 batchsize	1		per-layer	尚不支持
			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 input_layout/指定与对应输(extern)	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	
			input 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。			
			[batch,channel,height,	sequence/ 输入的 sequence	无限制,建议4对齐			
			width]:tensor	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			
	÷π // → ++		batch_size:int64 (extern)	batch_size/ 指定 LSTM 输入的 batchsize	大于 1 时仅支持 4 的倍数			
	部分支持 LSTM 扩		sequence_size :int64 (extern)	sequence_size sequence_size/ 于四组 建设在对文				
	展以及变		hidden_size:int64 (extern)	hidden_size/ LSTM 单元中的 hiddensize	无限制,建议8对齐			
	体命名为 exLSTM 算子,参	int8	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		per-layer/	
LSTM	数项中指 明	float16	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		per-channe	尚不支持
	(extern) 的项为		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。			
	exLSTM 独有的参		has_projection:int64 (extern)	has_projection/ projection 变 种	1(T) or 0(F) 目前限定 0,即尚不支持			
数功	数项。 input_layout:string	2.(sn)c: 指定 layout 对应的输入 shape 为[seqs*batches,						
			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
				batch/ 输入的 batch	channel 方向 concat			
Concat 部分支			input_tensor [batch,channel,height,	channel/ 输入的 channel	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			
	部分支持	int8 float16	width]:tensor	height/ 输入的 height	37°		per-layer	当 修协问 尚不支持
				width/ 输入的 width	制。			
			axis:int64	aixs/ 拼接的维度	无限制			
				batch/ 输入的 batch				
Mich	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制			当不支持
Mish	X14	float16	width]:tensor	height/ 输入的 height				间小文社
				width/ 输入的 width				

Rockchip ^{瑞芯微电子}

				KK3570 NI O OI XN 713X			和心厥电 1 放	
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/ 输入的 batch	1			
		int8	input_tensor	hannel/ 输入的 channel	T. 17H 4h 1			
		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制			
	支持			width/ 输入的 width	[1,8176]			
Pad	支持	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			尚不支持
		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			
				batch/ 输入的 batch				
			input_tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制			
ReduceMean	尚不支持 目	int8	width]:tensor	height/ 输入的 height	ניקו אין טר			尚不支持
前由 CPU 到	前由 CPU 实现	float16		width/ 输入的 width				间小文和
			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
				batch/ 输入的 batch channel/				
	光 不去性 日於	0	input_tensor [batch,channel,height, width]:tensor	输入的 channel height/ 输入的 height	- 无限制		1	
ReduceSum	尚不支持 目前由 CPU 实现	int8 float16		width/ 输入的 width			per-layer/ per-channel	尚不支持
		axes:	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			
				batch/ 输入的 batch	支持多 batch		per-channel	
	部分支持		input tensor	channel/ 输入的 channel	[1,8192]			
	目前 NPU 仅支		[batch,channel,height, width]:tensor	height/ 输入的 height				尚不支持
Resize	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,	int8 float16	int8		per-layer	尚不支持		
	其余不支持部 分的会 Fallback		mode:string	mode/resize 采用的模式	仅支持 nearest、linear			
	到 CPU 上实现。		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			
				batch/ 输入的 batch				多核协同 尚不支持			
		int8	input_tensor	channel/ 输入的 channel	约束规格: 2. height * width * type_bytes <= 8192*8192*16;						
		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	2.input_tensor 非四维时, shape 无限制						
Daghana	 支持			width/ 输入的 width			mon loven	当不支持			
Reshape	大 44			batch_o/ 输出的 batch_o	计 算 量 : alignment=16/type bytes; 约束规		per-layer	问小文付			
		int64	Shape (batch_o,channel_o,	channel_o/ 输出的 channel	格: 1.height_o * width_o * type_bytes <= INT32 MAX;						
		height_o,width_o):ter	height_o,widi	height_o,width_o):ten	height_o,width_o):ten		height_o/ 输出的 height	2.Align(height_o * width_o, alignment) <= 8192*8192;			
				width_o/ 输出的 width	3.输出 shape 非四维时, shape 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	
				batch/ 输入的 batch					
			input_tensor	channel/ 输入的 channel	无限制				
			[batch,channel,height, width]:tensor	height/ 输入的 height					
Reverse Sequence				width/ 输入的 width			per-layer	尚不支持	
			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				
				batch/ 输入的 batch					
Ciomo i d	支持	int8	input_tensor	channel/ 输入的 channel	无限制		man layyan	当不支持	
Sigmoid	文 村	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	一 人 PR 市引		per-layer	尚不支持	
				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
				batch/ 输入的 batch				
H40''4	士体	int8	input_tensor	channel/ 输入的 channel	 _ 无限制			出不士柱
HardSigmoid	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	一儿限制		per-layer	尚不支持
				width/ 输入的 width				
				batch/ 输入的 batch				
	To the	int8 float16	input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel	一 丁四4山	per-layer	per-layer	尚不支持
Swish	支持			height/ 输入的 height	一 无限制			同 个 文持
				width/ 输入的 width				
				batch/ 输入的 batch				
H 10 11		int8	input_tensor	channel/ 输入的 channel	丁四44	per		N/T+++
HardSwish	文符 	文持 float16 [t	[batch,channel,height, width]:tensor	height/ 输入的 height	一 无限制		per-layer	尚不支持
				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
				batch/ 输入的 batch				
Softplus	支持	int8	input_tensor [batch,channel,	channel/ 输入的 channel			per-layer	尚不支持
Softpius 2,11	float16	height, width]:tensor	height/ 输入的 height	一人吃中山		per-layer	向小文行	
				width/ 输入的 width				
			input_tensor [batch,channel, height, width]:tensor	batch/ 输入的 batch	无限制			
				channel/ 输入的 channel	硬件支持[1,8192]			
Softmax	支持	int8 float16		height/ 输入的 height	axis=1, 无限制 axis=3/-1,width[1, 8192],		per-layer	尚不支持
				width/ 输入的 width	height 无限制 且受限于 tranpose 的规格 限制			
			axis:int64	axis/ 做 softmax 的轴	1,3,即 channel 和 width 方			



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/ 输入的 batch				
			input_tensor [batch,channel,hei	channel/ 输入的 channel	T. 194 Au		量化方式 per-layer	
			ght, width]:tensor	height/ 输入的 height	· 无限制			
		int8		width/ 输入的 width				
Slice	Slice 部分支持 float16		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: 16 对齐,16bit 数据: 8 对齐。其他方向无限制。		per-layer	尚不支持
			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: 16 对齐,16bit 数据: 8 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			
				batch/ 输入的 batch				
			input_tensor [batch,channel,hei	channel/ 输入的 channel				
			ght, width]:tensor	height/ 输入的 height	· 无限制			
Split	部分支持	int8 float16		width/ 输入的 width	无限帅		per-layer	尚不支持
	Spite HP/J Z.19 f	Hoatio	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: 16 对齐,16bit 数据: 8 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
				batch/ 输入的 batch				
Tanh	支持	int8 [ba ght	input_tensor [batch,channel,hei	channel/ 输入的 channel	- - - 无限制		per-layer	尚不支持
Tailli			ght, width]:tensor	height/ 输入的 height	פון אין ארן אין		per-layer	向小叉持
			width/ 输入的 width					
				batch/ 输入的 batch	无限制			
		input_tensor [batch,channel,hei	channel/ 输入的 channel					
			ght, width]:tensor	height/ 输入的 height	[1,8192]			
				width/ 输入的 width	[1,8176]			
Transpose	支持	int8 float16	perm:int64[]	axis order/ 转置的轴顺序	限制与说明如下: 目前支持所有 perm 类型 5. 假设 in_shape[n1,c1,h1,w1],out_shape[n2,c2,h2,w2],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。 6. 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。 (4)N 维度参与 Transpose 变换时,N 的取值范围为[1,8192],否则 N 无限制 (5) perm=[1,2,3,0]时,h1*w1*n 满足 8bit 数据:16 对齐,16bit 数据:8 对齐。 (6) perm=[3,2,1,0]时,h1*w1 满足 8bit 数据:16 对齐,16bit 数据:8 对齐。			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	
			input_tensor [batch,channel,height, width]:tensor	h,channel,height, height/ height/ height/ height					
				width/ 输入的 width	当 <u>dilation kernel h</u> > 1 时, width < 16383 此外,对首层输入 width 存在限制,详见 <u>模型输入说明</u>		per-layer/ per-channel		
			kernel_shape [num_output, num_input,kernel_h, kernel_w]:int64[] nt8 loat16 strides[strides_h, strides_w]:int64[]	num_output/ 输出的 channel num_input/	一 无限制				
		int8 float16		输入的 channel kernel_h/ height 方向的 kernel 大小	— [1,31]				
Convolution	支持			kernel_w/ width 方向的 kernel 大小 stride_h/	[1,7]			支持	
				height 方向的 strides 大小 stride_w/ width 方向的 strides 大小					
				pads[pads top, pads left,	pads_left/ left 方向的 pads 大小 pads_right/				
			pads_bottom, pads_right]:int64[]	right 方向的 pads 大小 pads_top/ top 方向的 pads 大小	[0,15]				
				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]				



				1:1 .: /				
				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
		kernen num kernen num kernen stride stride stride stride pads pads pads pads pads pads	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 当 <u>dilation_kernel_h</u> > 1 时, width < 16383 此外,对首层输入 width 存在限制,详见 			
				num_output/ 输出的 channel num_input/ 输入的 channel	一 无限制		per-layer/ per-channel	
Depthwise Convolution	支持			kernel_h/ height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小	[1,8]			支持
				stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	[1,7]			
			pads[pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小 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	[0,15]			
			dilations[dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小	[1, 32]			

瑞芯微电子股份有限公司

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
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height	无限制			
			r	width/ 输入的 width	当 <u>dilation_kernel_h</u> > 1 时, width < 16383 此外,对首层输入 width 存在限制,详见 <u>模型输入说明</u>			
			num_output/ 输出的 channel num_input/	- 无限制				
ConvTranspose/ Deconvolution	支持	num_input,kernel_h, kernel_w]:int64[]	num_input,kernel_h,	输入的 channel kernel_h/ height 方向的 kernel 大小 kernel_w/	[1,31]		per-layer/ per-channel	支持
				width 方向的 kernel 大小 stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	- [1,8]			
			pads_left/ left 方向的 pads 大小 pads_right/ right 方向的 pads 大小 pads_top/ top 方向的 pads 大小 pads bottom/	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				
			group:int64	pads_bottom/ bottom 方向的 pads 大小 group/ group 的大小	↑支持 stride_w *(width - 1) - pads_left +1 < output_w 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 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
			input_tensor_1 [M, K]:tensor	M,K,N/	转为 exMatmul 实现,约束			
	尚不支持 目前由 CPU 实现	前由 CPU int8	input_tensor_2 [K,N]:tensor	输入数据的形状	同 exMatmul			
Comm			alpha:double	alpha/ 矩阵 A*B 乘法的 scale	- 无限制		per-layer/ per-channel	尚不支持
Genin			beta:double	beta/ 输入 C 矩阵的 scale	プレドド 中リ			四个文行
			transA:int64	transA/ A 矩阵是否转置	一 仅静态 tensor 支持转置			
			transB:int64	transB/ B矩阵是否转置	区附恋 tellsof 文持拉直			
			input_tensor_1[batch, K,1, N]:tensor	batch/ 输入的 batch				
exMatMul 支持	int8	NJ.tensor	K/ 输入的 K	_ K 支持[1,8192]		ner-lover	尚不支持	
	ZN	float16		M/ 输入的 M	- K XN[1,0172]		per-layer	间介文的
			M]:tensor	N/ 输入的 N				



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
			query_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch				
	exSDPAtten tion 支持	float16		channel/ 输入的 channel	— Channel < 8192			
		持 float16		height/ 输入的 height	Channel < 8192			
exSDPAtten				width/ 输入的 width				- 尚不支持
tion			key_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch				间小文诗
				channel/ 输入的 channel				
	f			height/ 输入的 height				
				width/ 输入的 width				

			MC5570 M O O1 文刊列表			110 100 100 100 100	KN FINA 5
			batch/ 输入的 batch				
	float16	value_tensor [batch,channel,height,	channel/ 输入的 channel	· Width < 8192			
	HoatTo	width]:tensor	height/ 输入的 height				
			width/ 输入的 width				
		mask_tensor	batch/ 输入的 batch				
	float16		channel/ 输入的 channel				
	HoatTo	[batch,channel,height, width]:tensor	height/ 输入的 height				
			width/ 输入的 width				
	float32	scale					



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/ 输入的 batch				
		int8	input_tensor	channel/ 输入的 channel	- 无限制		per-layer	
		float16	[batch,channel,height, width]:tensor	height/ 输入的 height				
г 1	+4.			width/ 输入的 width				尚不支持
Expand	支持			batch_o/ 输出的 batch_o				
			shape(batch_o,	channel_o/ 输出的 channel				
	int64	channel_o,height_o, width_o):tensor	height_o/ 输出的 height	- 无限制				
				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	多核协同	
		int8 float16 int64	x_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 —				
Where	支持	int8 float16 int64	y_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	
		bool	mask_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 —				
		int8 float16	shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o channel_o/ 输出的 channel height_o/ 输出的 height	无限制				

	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	多核协同
				batch/ 输入的 batch	无限制			
			input_tensor_1[batch	channel/ 输入的 channel	硬件支持[1,8192]			
			channel,height,width]:tensor	height/ 输入的 height	axis=1, 无限制 axis=3/-1,width[1, 8192], height 无			
				width/ 输入的 width	限制 且受限于 transpose 规格限制			
exSoftmaxMask	暂不支持, 返回 CPU 执行		input_tensor_2[batch, channel,height,width]:tensor	batch/ 输入的 batch	1		per-layer	尚不支持
				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_val	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	多核协同
				batch/ 输入的 batch				
int®	int8	x tensor [batch, channel,	channel/ 输入的 channel	c*h*w 满足如下限制				
GI.	++±	float16	height, width]:tensor	height/ 输入的 height	8bit 数据: 16 对齐, 16bit 数据: 8 对齐			以 <u>不</u> 士柱
exGlu	支持			width/ 输入的 width			per-layer	尚不支持
		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							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	同 Convolution						尚不支持
Convolution + Tanh	支持	PJ Convolution						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							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	同 ConvTranspose	;					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				1			已支持
Depthwise Convolution + Clip	尚不支持							己支持
Depthwise Convolution + PRelu/LeakyRelu	尚不支持							已支持
Depthwise Convolution + Add	尚不支持							己支持
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	一 同 Depthwise Con	volution					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:

(2) 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。



第六章 RK2118 NPU OP 支持列表



				MC2110141 C O1 XIII/IA		为10.100 10 10 10 10 10 10 10 10 10 10 10 10		
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/ 输入的 batch				
Add/Bias 支持			input tensor	channel/ 输入的 channel			. ,	
	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制	支持 ONNX 规范的四维 tensor 的所有广播操作,详见: <u>注释(1)</u>	per-layer/ per-channel		
				width/ 输入的 width				
				batch/ 输入的 batch		支持两个 tensor 的广播操作,以ONNX		
Sub		input_tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制	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;	man lawar/		
Sub	支持	float16	width]:tensor	height/ 输入的 height	プレド ド 中リ	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	per-channel	
				width/ 输入的 width		祝明: A 或 B 都可以作为) 播方。例 子见: <u>注释(1)</u>		



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	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: <u>注释(1)</u>	per-layer/ per-channel
				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
				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
Max	暂不支持	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(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: <u>注释(1)</u>	per-layer/ per-channel
				channel/ 输入的 channel			
				height/ 输入的 height			
				width/ 输入的 width			
Min	暂不支持	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(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)	
				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
Global AveragePool	支持	float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	1) 1m~11	±10,7,2(
				channel/ 输入的 channel	[1,8192]		per-layer
				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	暂不支持	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)		



				KKZII8 NI O OI X 19714X			一
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/ 输入的 batch	1		
			input_tensor [batch,channel,height,	channel/ 输入的 channel			
			width]:tensor	height/ 输入的 height	[1,8192]		
			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		
AveragePool	支持	float16	float16 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		per-layer
				kernel_w/ width 方向的 kernel 大小	支持。		
				pads_left/ left 方向的 pads 大小			
			pads[pads_top,pads_left, pads_bottom,	pads_right/ right 方向的 pads 大小	[0,7]		
			pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,/]		
				pads_bottom/ bottom 方向的 pads 大小			
			st	stride_h/ height 方向的 strides 大小	[1,8]		
			strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,0]		



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
				batch/ 输入的 batch	1		
			input_tensor	channel/ 输入的 channel			
			[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]		
				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			per-layer
				dilations_w/ widtht 方向的 dilations 大小	1		
MaxPool	支持	float16	float16 kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height 方向的 kernel 大小	无限制, NPU 支持		
				kernel_w/ width 方向的 kernel 大小	- [1,7]; 其它由 CPU 支 持。		
				pads_left/ left 方向的 pads 大小			
			pads[pads_top,pads_left,	pads_right/ right 方向的 pads 大小	[0.7]		
			pads_bottom, pads_right]:int64[]	pads_top/ top 方向的 pads 大小	[0,7]		
			pads_bottom/ bottom 方向的 pads 大小				
			storage_order: int64	storage_order/优先储存方式	0		
		strides[strides h,	stride_h/ height 方向的 strides 大小				
			strides_w]:int64[]	stride_w/ width 方向的 strides 大小	[1,8]		



				KKZ118 NI U OI X 1971A		柳心脉电力	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		
			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				
			momentum:double	momentum/ 训练时的滑动平均参数	无限制				
Batch	 支持	float16		batch/ 输入的 batch	1		per-layer/		
Normalization (19)		input_tensor [batch,channel,height,	channel/ 输入的 channel			per-channel			
		width]:tensor	height/ 输入的 height	无限制					
			width/ 输入的 width						
				batch/ 输入的 batch	支持多 batch				
		input_tensor [batch,channel,height, width]:tensor batch,channel,height, width]:tensor channel/输入的 channel height/输入的 height width/输入的 width layernorm_weight [channel,height, width]:tensor(const) height/输入的 channel height/输入的 height width/输入的 height width/输入的 width width]:tensor(const) height/输入的 width height/输入的 width/输入的 width width]:tensor(const) normalized_shape /参与每一批归一化的 Feature elementwise_affine/是否具有可学习数		输入的 channel					
			width]:tensor layernorm_weight			无限制			
						channel/输入的 channel	等于 input_channel		
			width]:tensor(const)	height/ 输入的 height	等于 input_height				
Layer	暂不支持		[channel,height,	width/ 输入的 width	等于 input_width		per-layer		
Normalization			T-ZIN HOALTO	C14 Hoat10	1 2,10	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。		
				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/ 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
				batch/ 输入的 batch			
CI. /D TITC	+-++	0 16	input_tensor	channel/ 输入的 channel	T.178 4d		
Clip/ReLU6	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		per-layer
				width/ 输入的 width			
		克持 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch			
E.	+-++			channel/ 输入的 channel	- 无限制		
Elu	文持			height/ 输入的 height			
				width/ 输入的 width			
				batch/ 输入的 batch			
	+-4+	G. A.	input_tensor	channel/ 输入的 channel	T 772 4-1		
Gelu	支持	float16 [batch,channel,hewidth]:tensor	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		
				width/ 输入的 width			



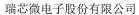
Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
				batch/ 输入的 batch	1		
	支持	float16	input_tensor	channel/ 输入的 channel			
Relu	X14	Поатто	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer
				width/ 输入的 width			
				batch/ 输入的 batch	1		
1 1 1 1	支持	float16	input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel	- 无限制		per-layer
LeakyRelu				height/ 输入的 height			
				width/ 输入的 width			
				batch/ 输入的 batch	1		
				channel/ 输入的 channel			
PRelu	支持	float16 input_tensor [batch,channel,heigh width]:tensor	[batch,channel,height,	height/ 输入的 height	无限制		per-layer/ per-channel
				width/ 输入的 width			
				slope/ PRelu 系数	仅支持单个标量或 C 维度系数	戈 C	



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/ 输入的 batch	1		
			input_tensor [batch,channel,height, width]:tensor	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]				
GRU	暂不支持	float16	batch_size:int64 (extern)	batch_size/ 指定 GRU 输入的 batchsize	1		per-layer
			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 input_size] 2.(sn)c: 指定 layout 对应的输入 shape 为[seqs*bat input_size,1,1] 要求填写指定的 layout, 同时要求填写该 op 实际的 batch_size、sequence_size、hidden_size。 1.sbnc: 指定 layout 对应的输出 shape [seqs,directions,batches, hidden_size] 2.(sn)c: 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,directions*input_size,1,1] 要求填写指定的 layout, 同时要求填写该 op 实际	2.(sn.): 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				
			1 = 0		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		



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						
			input 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。								
			[batch,channel,height, width]:tensor	sequence/输入的 sequence	无限制,建议4对齐								
			widing.tensor	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								
	如八士柱		batch_size:int64 (extern)	batch_size/ 指定 LSTM 输入的 batchsize	大于1时仅支持4的倍数								
	部分支持 LSTM 扩		sequence_size :int64 (extern)	sequence_size/ 指定 LSTM 输入的 seqsize	无限制,建议4对齐								
	展以及变体命名为		hidden_size:int64 (extern)	hidden_size/ LSTM 单元中的 hiddensize	无限制,建议8对齐								
	exLSTM 算子,参	M 参 float16	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		per-layer/						
LSTM	数项中指明		float16	float16	float16	float16	float16	float16	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		per-channel
	(extern) 的项为		has_dropout:int64 (extern)	has_dropout/ caffe 框架下的 indicator 功能的选择	I(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。								
	exLSTM 独有的参 数项。		has_projection:int64 (extern)	has_projection/ projection 变种	1(T) or 0(F) 目前限定 0,即尚不支持								
	蚁火 。	input_layout:string input_layout/才shape 含义一到output_layout:string output_layout/	input_layout/指定与对应输入 shape含义一致的 layout	I.snc: 指定 layout 对应的输入 shape 为[seqs, batches, input_size] 2.(sn)e: 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/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
				batch/ 输入的 batch	channel 方向 concat		
			input_tensor [batch,channel,height,	channel/ 输入的 channel	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		
Concat	部分支持	float16	width]:tensor	height/ 输入的 height	据: 16 对齐, 16bit 数据: 8 对齐 其他方向 concat 无限		per-layer
		axis		width/ 输入的 width	制。		
			axis:int64	aixs/ 拼接的维度	无限制		
				batch/ 输入的 batch			
Mish	支持	float16	input_tensor	channel/ 输入的 channel	- 无限制		
IVIISII	入げ	float16 [batch,channel,height, width]:tensor		height/ 输入的 height) July Ind		
				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	
				batch/ 输入的 batch	1			
		float16	input_tensor	hannel/ 输入的 channel	工阳如			
		Hoatro	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制			
Pad	支持			width/ 输入的 width	[1,8176]			
Pad 文持	2.19	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			
					batch/ 输入的 batch			
			input_tensor	channel/ 输入的 channel	- 无限制			
Paduaal/laan	尚不支持 目	float16	[batch,channel,height, width]:tensor	height/ 输入的 height				
	前由 CPU 实现	поано		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			

RK2118 NPU OP 支持列表



Rockchip 瑞芯微电子	1			RK2118 NPU OP 支持列表		瑞芯微	电子股份有限公司
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/ 输入的 batch			
			input_tensor	channel/ 输入的 channel	无限制		
ReduceSum	尚不支持 目前	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	一 人 PR 市 J		per-layer/
ReduceSum	由 CPU 实现	Поасто		width/ 输入的 width			per-channel
			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		
				batch/ 输入的 batch	支持多 batch		
	部分支持			channel/ 输入的 channel	[1 8102]		
	目前 NPU 仅支		input_tensor [batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]		
Resize	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,	語的整倍 近 邻 和 直缩放, 支 持 部		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		per-layer
	其余不支持部 分的会 Fallback		mode:string	mode/resize 采用的模式	仅支持 nearest、linear		
	到 CPU 上实现。		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
				batch/ 输入的 batch			
		Start C	input_tensor	channel/ 输入的 channel	约束规格: 3. height * width * type_bytes <=		
		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	8192*8192*16; 2.input_tensor 非四维时,shape 无限制		
Danhara	支持			width/ 输入的 width			
Reshape	文村		Shape	batch_o/ 输出的 batch_o	计算量: alignment=16/type bytes;		
		int64		channel_o/ 输出的 channel	约束规格: 1.height_o * width_o * type_bytes <=		
			(batch_o,channel_o, height_o,width_o):tensor	height_o/ 输出的 height	INT32_MAX; 2.Align(height_o * width_o, alignment) <= 8192*8192;		
				width_o/ 输出的 width	3.输出 shape 非四维时, shape 无限制		



Rockchip _{瑞芯微电子}	_	_		RK2118 NPU OP 支持列表	瑞芯微电子股份有限公司		
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/ 输入的 batch			
			input_tensor	channel/ 输入的 channel	- - - - 无限制		
			[batch,channel,height, width]:tensor	height/ 输入的 height			
Reverse Sequence	尚不支持 目前 由 CPU 实现	float16		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		
				batch/ 输入的 batch			
Sigmoid	支持	float16	input_tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制		
Signioid	又1寸	Hoat16	width]:tensor	height/ 输入的 height) OLK IN		
				width/ 输入的 width			



Rockchip 瑞芯微电子		1		RK2118 NPU OP 支持列表		瑞芯微口	电子股份有限公司
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
		float16	input_tensor	batch/ 输入的 batch			
HandSiana aid	支持			channel/ 输入的 channel	- 无限制		non loven
HardSigmoid	大村		[batch,channel,height, width]:tensor	height/ 输入的 height] /LPR市1		per-layer
				width/ 输入的 width			
		支持 float16		batch/ 输入的 batch			
Swish	古柱		input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel	- 无限制		
SWISH	大村			height/ 输入的 height	人PR 市1		
				width/ 输入的 width			
				batch/ 输入的 batch			
HardSwish	支持	float16	input_tensor	channel/ 输入的 channel	- 无限制		per-layer
Haluswish	大 14	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	プロPK 即3		per-layer
				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
				batch/ 输入的 batch			
Softplus 支持	士 ·	float16	input_tensor	channel/ 输入的 channel	- 无限制		and laves
	Hoat16	[batch,channel,height, width]:tensor	height/ 输入的 height	7 人口 阿克 市门		per-layer	
				width/ 输入的 width			
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	无限制		
				channel/ 输入的 channel	硬件支持[1,8192]		
Softmax 暂	暂不支持			height/ 输入的 height	axis=1,无限制 axis=3/-1,width[1, - 8192], height 无限制		per-layer
				width/ 输入的 width	且受限于 tranpose 的规格限制		
			axis:int64	axis/ 做 softmax 的轴	1,3,即 channel 和 width 方向		



Operator		数据类型		 输入参数		广播支持	量化方式	
Орегатог	ZN IAU	双加 天主	11119 / \	batch/ 输入的 batch	=3.X.M.fr	/ 油文10	里化刀八	
			input_tensor	channel/ 输入的 channel				
				[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		
Slice 部分支持	0 16		width/ 输入的 width			,		
	float16	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: 16 对齐,16bit 数据: 8 对齐。其他方向无 限制。	per-	per-layer		
			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: 16 对齐,16bit 数据: 8 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			
				batch/ 输入的 batch				
			input_tensor	channel/ 输入的 channel				
			[batch,channel,height, width]:tensor	height/ 输入的 height	无限制			
Split	部分支持	float16		width/ 输入的 width)Cpx (P)		per-layer	
			axis:int64	axis/ 切分的维度				
		num_outputs:in	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: 16 对齐, 16bit 数据: 8 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
			input_tensor t16 [batch,channel,height,	batch/ 输入的 batch			
Tanh	支持	float16		channel/ 输入的 channel	无限制		per-layer
Tum Zjy Hourio	Hourto	width]:tensor	height/ 输入的 height	- Juna in J		per layer	
			width/ 输入的 width				
				batch/ 输入的 batch	无限制		
		[batch	input_tensor [batch,channel,height,	channel/ 输入的 channel	51.01003		
			width]:tensor	height/ 输入的 height	[1,8192]		
				width/ 输入的 width	[1,8176]		
Transpose	支持	float16	perm:int64[]	axis order/ 转置的轴顺序	限制与说明如下: 7. 假设 in_shape[n1,c1,h1,w1],out_shape[n2,c2,h2,w2] 8. 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。 9. 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。 10. 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
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/	无限制 当 dilation_kernel_h > 1 时, width < 16383 此外,对首层输入 width 存在限		
		kernel_shape [num_output,	输入的 width num_output/ 输出的 channel num_input/ 输入的 channel	此外,对自层输入 Width 存在限制,详见模型输入说明 - 无限制			
Convolution	支持	num_input,kernel_h, kernel_w]:int64[] strides[strides_h, strides_w]:int64[] pads[pads_top, pads_left, pads_bottom, pads_right]:int64[] group:int64 dilations[dilations_h, dilations_w]:int64[]	num_input,kernel_h,	kernel_h/ height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小 stride h/	- [1,31]	l	per-layer/
Convolution	又行			height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	- [1,7]		per-channel
			pads_left/ left 方向的 pads 大小 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	- [0,15]			
			group:int64	group/ group 的大小	无限制		
				dilations_h/ height 方向的 dilations 大小 dilations_w/ widtht 方向的 dilations 大小	- [1, 32]		



Rockchip 瑞芯微电子				RK2118 NPU OP 支持列表		瑞芯	微电子股份有限公司
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
		input_tensor [batch,channel,height, width]:tensor	[batch,channel,height,	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/	无限制 当 dilation_kernel_h > 1 时, width < 16383		
				输入的 width	此外,对首层输入 width 存在限制,详见		
				num_output/ 输出的 channel	无限制		
		kernel_shape [num_output,	num_input/ 输入的 channel) LPK (PI)			
		float16	num_input,kernel_h, kernel_w]:int64[] strides[strides_h, strides_w]:int64[]	kernel_h/ height 方向的 kernel 大小	[1,8]		
Depthwise Convolution	支持			kernel_w/ width 方向的 kernel 大小	[1,0]		per-layer/ per-channel
				stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	[1,7]	_	
				pads_left/ left 方向的 pads 大小			
			pads[pads_top, pads_left, pads_bottom,	pads_right/ right 方向的 pads 大小 pads top/	[0,15]		
			pads_right]:int64[]	top 方向的 pads 大小 pads_bottom/			
				bottom 方向的 pads 大小 dilations h/		_	
		dilations[dilations_h, dilations_w]:int64[]	height 方向的 dilations 大小 dilations_w/ widtht 方向的 dilations 大小	[1, 32]			



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
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height	无限制 <u>当 dilation_kernel_h</u> > 1 时, width < 16383		
			kernel shape	输入的 width num_output/ 输出的 channel num_input/	此外,对首层输入 width 存在限制,详见模型输入说明 - 无限制		
]	[num_output, num_input,kernel_h, kernel_w]:int64[]	输入的 channel kernel_h/ height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小	- [1,31]		
ConvTranspose/ Deconvolution	支持	float16	strides[strides_h, strides_w]:int64[]	stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小	[1,8]		per-layer/ per-channel
			pads[pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left 方向的 pads 大小 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	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		
			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 dilations_w/ widtht 方向的 dilations 大小	- [1, 32]		



				RRZ118 NFU OF 文句列a		111112 198 12 17	以切有限公司		
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		
					input_tensor_1 [M, K]:tensor	M,K,N/	转为 exMatmul 实现,约束同 exMatmul		
		input_tensor_2 [K,N]:tensor	输入数据的形状	我內 eximatmul 实现,约果问 eximatmul					
Comm	Gemm 尚不支持 目前由 CPU 实 现	alpha:double	alpha/ 矩阵 A*B 乘法的 scale			per-layer/			
Gemm			beta:double	beta/ 输入 C 矩阵的 scale	元 [於即]		per-channel		
				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	仪静态 tensor 文持拉直				
			input_tensor_1[batch,	batch/ 输入的 batch					
avMatMul	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		K,1, NJ.tensor	K/ 输入的 K	—— K 支持[1,8192]		n an Ioyyan		
exMatMul 暂不支持	f Hoat16		M/ 输入的 M	■ K 又玩[1,0172]		per-layer			
			1, M]:tensor	N/ 输入的 N					



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/ 输入的 batch			
	exSDPAtten	float16	query_tensor [batch,channel,height,	channel/ 输入的 channel	- Channel < 8192		
		HoatTo	width]:tensor	height/ 输入的 height	Chamier \ 8192		
				width/ 输入的 width			
exSDPAtten		float16 [batch	key_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch			
tion	百个义罚			channel/ 输入的 channel			
	float			height/ 输入的 height			
				width/ 输入的 width			
		value_tensor		batch/ 输入的 batch	- Width < 8192		
		Hourit	[batch,channel,height, width]:tensor	channel/ 输入的 channel	- Width < 8192		

-			. , , , , , , , , , , , , , , , , , , ,	N = 101 B 1 101	
			height/ 输入的 height		
			width/ 输入的 width		
			batch/ 输入的 batch		
	5	mask_tensor	channel/ 输入的 channel		
	float16	[batch,channel,height, width]:tensor	height/ 输入的 height		
			width/ 输入的 width		
	float32	scale			



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/ 输入的 batch			
		float16	input_tensor [batch,channel,height,	channel/ 输入的 channel			
	float16	width]:tensor	height/ 输入的 height	- 无限制			
Expand	支持	shape(batch_o, channel_o,height_o, width_o):tensor		width/ 输入的 width			per-layer
Lapand	ZN			batch_o/ 输出的 batch_o	- 无限制		per rayer
			shape(batch_o,	channel_o/ 输出的 channel			
			width_o):tensor	height_o/ 输出的 height	- Zupa ipi		
				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
		int8 float16 int64	x_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/			
Where 暂不支持	斯本支持 height, width height, width height, width height, width height, width	float16 y_tensor [batch, channel, beight width]tensor	输入的 width batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width				
		mask_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	
		int8 float16 shape(batch_o, channel_o,height_o, width_o):tensor		batch_o/ 输出的 batch_o channel_o/ 输出的 channel height_o/ 输出的 height width_o/ 输出的 width	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
				batch/ 输入的 batch	无限制		
			input_tensor_1[batch,	channel/ 输入的 channel	硬件支持[1,8192]		
			channel,height,width]:tenso	height/ 输入的 height	axis=1,无限制 - axis=3/-1,width[1, 8192], height 无限制		
暂不支持,			width/ 输入的 width	且受限于 transpose 规格限制			
	暂不支持, 返回 CPU	I	input_tensor_2[batch, channel,height,width]:tensor	batch/ 输入的 batch	1	_	per-layer
exSoftmaxMask	执行			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
			x_tensor [batch, channel, height, width]:tensor	batch/ 输入的 batch			
				channel/ 输入的 channel	c*h*w 满足如下限制 8bit 数据: 16 对齐, 16bit 数据: 8 对		per-layer
exGlu	支持			height/ 输入的 height	齐		
exGlu	又1寸			width/ 输入的 width			
		int64	axis:int64	axis/ 切分的维度	axis ==1		



				THE THOU OF AND THE			
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						
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	同 Convolution					
Convolution + Softplus	支持						
Convolution + HardSigmoid	支持	-					
Convolution + HardSwish	支持						
Convolution + Elu	支持						
Convolution + Swish	支持						
Convolution + Mish	支持						



				Idaziioide of Zijiyik		11 - 121 - 1	WW III WA II
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						
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	同 ConvTranspose					
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						



				KK2110 NI O OI XN 917K		10 - 100 - 100	又历日限公司
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		,				
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	同 Depthwise Conv	volution				
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	同 Mul					
Convolution + add + Relu	支持	同 Convolution						

注释:

(3) 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。



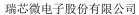
第七章 CPU OP 支持列表



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	
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	



Floor	向下取整函数	无限制	
Mod	取模函数	无限制	
exMeanVarianceNormalization	均方差归一化函数	无限制	
And	与函数	无限制	
GatherElements	元素收集函数	无限制	
Log	Log函数	无限制	



第八章 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》)



第九章 模型输入输出规格说明



1. 模型输入说明

-11- 11.	供刑子曰	输入	* P '11 El to	, 1,	输入宽(width)对齐要求 单位:元素个数		输入宽(width)大小限制	
平台			首层设置输入数据类型	mean/scale/quant 后端实现设备	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 <u>注释 9</u>)	当输入通道 (channel) 非 1,3,4
			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) / 4096) Kernel_h * dilation_kernel_w) / 128)	无限制
	int8		int8					
			float16	- CPU				
			其他类型 (* <u>注释 8</u>)	Cro				
RK3566 /3568			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) / 4096) ConvTranspose/Deconvolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 128)	
	float16		int8					无限制
	Hoatro		float16					
			其他类型 (* <u>注释 8</u>)					
	无限制	非4维	非限制	CPU	1	1	无限制	无限制



- - .		<i>t</i> △)	光 巴	1 1 /	输入宽(width)对齐要求 单位:元素个数		输入宽(width)大小限制	
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	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 <u>注释 9</u>)	当输入通道 (channel) 非 1,3,4
			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	无限制
	int8		int8				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 *	
		4 维度 pat16	float16	- CPU			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)	
			其他类型 (* <u>注释 8</u>)	Ci U				
RK3588			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_w) / 128), 3)	
	float16		int8					
	noutro		float16			1		
			其他类型 (* <u>注释 8</u>)					
	无限制	非4维	非限制	CPU	1	1	无限制	无限制



- - .	推到	<i>t</i> Δ)	光 巴		输入宽(width)对齐要求 单位:元素个数		输入宽(width)大小限制	
平台	芯片 模型首层 输入 平台 精度类型 维度		首层设置输入数据类型	mean/scale/quant 后端实现设备	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 <u>注释 9</u>)	当输入通道 (channel) 非 1,3,4
			uint8	NPU		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	无限制
	int8		int8		- 16		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:	
		4 维度 -	float16	- CPU			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)	
			其他类型 (* <u>注释 8</u>)	CFU				
RK3562			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
	float16		int8					
Hoati	noutro		float16			1		
			其他类型 (* <u>注释 8</u>)					
	无限制	非4维	非限制	CPU	1	1	无限制	无限制



-11- 11.	は到去日	<i>t</i> △)	* = 1. = 1.	mean/scale/quant	输入宽(width)对齐要求 单位:元素个数		输入宽 (width) 大小限制		
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	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 <u>注释 9</u>)	当输入通道 (channel) 非 1,3,4	
			uint8	NPU			各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式: 3. width * dilation kernel h <= 2048 * N	无限制	
	int8		int8		— 16	1	4. 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:		
			float16	· CPU			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)		
		4 维度	其他类型 (* <u>注释 8</u>)	Cro					
RK3576			uint8	CPU	8	1	各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式: 2. 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	
	float16		int8						
Tioatro	noutro		float16					JUPKIF1	
			其他类型 (* <u>注释 8</u>)						
	无限制	非4维	非限制	CPU	1	1	无限制	无限制	



芯片	模型首层	输入	首层设置输	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	
平台	精度类型	维度	入数据类型		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
		4 维度			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	
RK2118	float16	非4维	float16	NPU	1	1	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
RV1103/	102/	uint8 4维度 int8					各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式: 3. width * dilation_kernel_h <= 2048 * N 4. 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:	T 100 day
1106	int8		int8	NPU	16	1	1	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

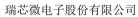
芯片平台	模型输出精度类型 (*注释 2)	输出维度	设置输出 Layout	Channel 对齐要求	H*W 对齐要求
			NCHW	无	无
	:49		NHWC	8 对齐(<u>*注释 1</u>)	无
	int8	- 4 维度	NC1HWC2	最后一层卷积类算子,16 对齐,最后一层非卷积类算子 8 对齐	H*W 要 4 对齐
			UNDEFINE	无	无
RK3566/3568			NCHW	无	无
	fl. v1C		NHWC	4 对齐(<u>*注释 1</u>)	
	float16		NC1HWC2	最后一层卷积类算子,8对齐,最后一层非卷积类算子4对齐	H*W 要 4 对齐
			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
			NCHW	无	无
	int8		NHWC	16 对齐(<u>*注释 1</u>)	
	ints	- 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
			UNDEFINE	无	无
RK3588			NCHW	无	无
	Size 16		NHWC	8 对齐(<u>*注释 1</u>)	无
	float16		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
			NCHW	无	无
	int8		NHWC	无	
	ints	- 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
			UNDEFINE	无	无
RK3562			NCHW	无	无
	Size 16		NHWC	无	无
	float16		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
			NCHW	无	无
	int8		NHWC	无	
	into		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
		4 维度	UNDEFINE	无	无
RK3576			NCHW	无	无
	float16		NHWC	无	无
	Hoatio		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



				(英至相) V 相 国	和心脉也1放闪月散五引
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
		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
RK2118	float16	7 维汉	NHWC	无	无
		非4维	UNDEFINE	无	无
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
		4 维度	NHWC	无	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。