

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:	v1.6.0
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:	2023-11-30

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



## 目 录

第一章	RK3566/3568 NPU OP 支持列表	1
第二章	RK3588 NPU OP 支持列表	33
第三章	RV1103/1106 NPU OP 支持列表	67
第四章	RK3562 NPU OP 支持列表	100
第五章	CPU OP 支持列表	133
第六章	GPU OP 支持列表	138
第七章	模型输入输出规格说明	140



# 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			
		int8	input tensor	channel/ 输入的 channel	支持 ONNX 规范的四维 tensor 的所有		
Add/Bias	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制	文持 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			
		int8 float16	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)	per-layer/ per-channel
Sub	支持			channel/ 输入的 channel			
Sub	文符		[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
				batch/ 输入的 batch	无限制		per-layer/ per-channel
Mul/Scale 支持	±-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: 注释(1)	
	<b>文</b> 村	float16	T T	height/ 输入的 height			
				width/ 输入的 width			
		float16	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((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			
Div	部分支持		[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
				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/
Mari	<b>本</b> 株	int8	input tensor	channel/ 输入的 channel			
Max 支持	<b>文</b> 村	float16	[batch,channel,height, width]:tensor	无限制	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>	
		int8 float16	2	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	per-layer/ per-channel
Min	支持			channel/ 输入的 channel			
Willi	X14		width]:tensor	height/ 输入的 height	) LPK IPG	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>	



Operator	支持情况	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	-1-1-1	int8	input_tensor	channel/ 输入的 channel [1,8192]			
AveragePool	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)		
		int8 float16		batch/ 输入的 batch	1		per-layer
ClahalMayDaal	支持		input tensor	channel/ 输入的 channel	[1,8192]		
GlobalMaxPool	文村		[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		
		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		
				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			
			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	1		
			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 float16	dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height 方向的 dilations 大小		per-l	
				dilations_w/ widtht 方向的 dilations 大小	- 1		
MaxPool	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 支 持。		
				kernel_w/ width 方向的 kernel 大小			
				pads_left/ left 方向的 pads 大小			
			pads[pads_top,pads_left,	pads_right/ right方向的 pads 大小	50.73		
			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	F1 01	1	
			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 Normalization 支持 int8 float16			batch/ 输入的 batch	1		per-layer/		
	float16	input_tensor	channel/ 输入的 channel			per-channel		
		[batch,channel,height, width]:tensor	height/ 输入的 height	无限制				
			width/ 输入的 width					
			batch/ 输入的 batch	支持多 batch				
		3分支持 float16	input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel				
				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		
			14114	height/ 输入的 height	等于 input_height			
Layer Normalization	部分支持		float16 layernorm_bias [channel,height, width]:tensor(const)  normalized_shape:int64[]	[channel,height,	width/ 输入的 width	等于 input_width		per-layer
				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			
Clim/D at 114	支持	int8	input_tensor	channel/ 输入的 channel	一 无限制		m on loven
Clip/ReLU6	文付	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	一		per-layer
				width/ 输入的 width			
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch			
E1	+-+±	int8 float16		channel/ 输入的 channel	- 无限制		
Elu	支持			height/ 输入的 height			
				width/ 输入的 width			
				batch/ 输入的 batch			
	+-+±.	int8	input tensor	channel/ 输入的 channel	T:17F 4:1		
Gelu 支	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	一 无限制		
				width/ 输入的 width			



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			
Dala	Relu 支持	int8	input_tensor	channel/ 输入的 channel	- 无限制		non lavor
Keiu	<b>大</b> 村	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	<b>一</b>		per-layer
				width/ 输入的 width			
		int8 float16		batch/ 输入的 batch			
			input tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel	<b>一</b> 77日4-1		
LeakyRelu	支持			height/ 输入的 height	<del>一</del> 无限制		per-layer
				width/ 输入的 width			
				batch/ 输入的 batch			
				channel/ 输入的 channel	工7月4月		
PRelu 支持		input_tensor [batch,channel,height, width]:tensor	height/ 输入的 height	<del>一</del> 无限制		per-layer/ per-channel	
			Widelijaensor	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																										
				input_size/ 输入的 input_size	无限制,建议8对齐																										
		J 扩 人及变 了名为 RU 算 参数 float16 指明 ttern) i为	扩 受 名为 之 多数 float16 float16	GRU 扩展以及变体命名为exGRU 算子,参数 float16 (extern)的项为exGRU 独	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				RU 扩展以及变体	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	子,参数					batch_size:int64 (extern)	batch_size/ 指定 GRU 输入的 batchsize	1		per-layer																					
	(extern)					extern) 项为 GRU 独	(extern) 的项为 exGRU 独	(extern) 的项为 exGRU 独	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																														
	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 [seqs,directions,batches, hidden_size]	s, Ž																									



Operator	支持情况	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	SSS						
			[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						
		本命名为 exLSTM 章子,参 int8 数项中指 float16	STM 扩 屡以及变 本命名为 xLSTM 淳子,参 int8 数项中指 float16		direction:string	direction/ 指定 LSTM 的运算方向	forward: 指定 LSTM 的运算方向为前向 reverse: 指定 LSTM 的运算方向为反向 bidirectional: 指定 LSTM 的运算方向为双向				
				batch_size:int64 (extern)	batch_size/ 指定 LSTM 输入的 batchsize	大于1时仅支持4的倍数					
				TM 扩 以及变 命名为 LSTM 子,参 int8	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						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		1 /		
LSTM	型					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	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	LSTM 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。		input_layout:string (extern)								
				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		
Concat 部分支持		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			
	int8 float16	width]:tensor	height/ 输入的 height	齐。对齐量: 8bit 数 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		float 16	input_tensor	channel/ 输入的 channel	- 无限制		
IVIIOII X JT	支持		[batch,channel,height, width]:tensor	height/ 输入的 height	נייו איין / ר		
				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		
	ad 支持	int8	input_tensor	hannel/ 输入的 channel	- 无限制		
		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	7.5 195 中月		
Pad				width/ 输入的 width	[1,8176]		
Tau		int64 pads:tensor	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 [batch,channel,height,	channel/ 输入的 channel	· 无限制		
PaduaaMaan	ZaducaMann	int8	width]:tensor	height/ 输入的 height	人口 的文 即 1		
Keducelviean			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
			input_tensor	batch/ 输入的 batch channel/ 输入的 channel			
ReduceSum 尚不支持 目前	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	<ul><li>─ 无限制</li><li>─</li></ul>		per-layer/	
	由 CPU 实现	float16		width/ 输入的 width	V (1		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		
		前 NPU 仅支 竞 高 方 向 不 过 8 倍的整倍 int8 t 的 最 近 邻 和 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	支持多 batch		
	部分支持			channel/ 输入的 channel	[1,8192]		
	目前 NPU 仅支			height/ 输入的 height			per-layer
数的最近邻和 线性插值缩放, 其余不支持部 分的会 Fallback	音 int8 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		
		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	无限制		
		int8	input_tensor	channel/ 输入的 channel	[1 9102]		
		float16	[batch,channel,height, 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	int64		channel_o/ 输出的 channel	- [1,8192]		
				height_o/ 输出的 height	[1,0192]		
			Shape (batch o,channel o, height_o,width_o):tensor	width_o/ 输出的 width	[1,8176]		
				[n,c,h1,w1]- >[n,c,h2,w2]/ (h1*w1=h2*w2)	支持		
				[1,c,h,w]- >[c1,hw1,1,1]/ (c1=c/a, h*w=hw1/a, a 为整数)	- 不支持		
				[n,c,1,1]->[1,n1,h,w]/ (c=h*w/a, n1=n/a, a 为整数)	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 [batch,channel,height,	channel/ 输入的 channel	- 无限制		
			width]:tensor	height/ 输入的 height	יים אאן טי		
Reverse Sequence	尚不支持 目前 由 CPU 实现			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 float10	into [bar	input_tensor	channel/ 输入的 channel	・ 无限制		
Sigmoid			[batch,channel,height, width]:tensor	height/ 输入的 height	ניח אא 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
		int8		batch/ 输入的 batch			
II1C::1	士性		input_tensor	channel/ 输入的 channel	工阳幽		
HardSigmoid	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			
Swish	支持		input_tensor	channel/ 输入的 channel	- 无限制		
SWISH	文14		[batch,channel,height, width]:tensor	height/ 输入的 height			
				width/ 输入的 width			
				batch/ 输入的 batch			
HandSwich	z持 int8 float16	input tensor	channel/ 输入的 channel	<b>子阳</b> 割		per-layer	
Haluswish			[batch,channel,height, width]:tensor	height/ 输入的 height	一 无限制		per-tayer
				width/ 输入的 width			



		Jul 1 - 11 - 11	RR5300/3306 NFU OF 文行列収			畑心城出 J 放切有限公司		
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	
				batch/ 输入的 batch				
支持	int8	input_tensor	channel/ 输入的 channel			per-layer		
	float16 [batch,channel,height width]:tensor	width]:tensor	height/ 输入的 height	- 无限制		per-layer		
			width/ 输入的 width					
			input tensor	batch/ 输入的 batch				
				channel/ 输入的 channel				
Softmax 尚不支持,目前 由 CPU 实现	,目前 last16	[batch,channel,height, width]:tensor	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				
			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	
		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			per-layer		
	Hoatro	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	Tanh 支持 int8 float16	int8	input tensor	channel/ 输入的 channel	T: 1/El /k-l		
lanh		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	[1,0100]		
				height/ 输入的 height	[1,8192]		
	) m ()	int8		width/ 输入的 width	[1,8176]		
Transpose	部分支持	int8 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	
			input tensor	batch/ 输入的 batch channel/ 输入的 channel	无限制			
			[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>			
			kernel shape	num_output/ 输出的 channel num input/	- 无限制			
			[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]			
Convolution	支持	int8 float16	strides[strides_h, strides_w]:int64[]	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	[1,7]		per-layer/ per-channel	
			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 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/ 输入的 batch channel/ 输入的 channel height/ 输入的 height width/ 输入的 width	<ul> <li>无限制</li> <li>当 dilation kernel h &gt; 1 时, width &lt; 16383</li> <li>此外,对首层输入 width 存</li> </ul>		
			kernel_shape [num_output,	m/h) width  num output/ 输出的 channel  num input/ 输入的 channel	在限制,详见 <u>模型输入说明</u> - 无限制		
Depthwise Convolution	7 VT	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 stride h/	- [1,8]		per-layer/ per-channel
			strides[strides_h, strides_w]:int64[]	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]		



Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
	支持		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	<ul> <li></li></ul>		
			kernel_shape [num_output,	num_output/ 输出的 channel num_input/ 输入的 channel	- 无限制		
ConvTranspose/		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	[1,31]		per-layer/
Deconvolution			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-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	- 不支持 -		per-layer/
Bemm 由 CPU 实现		CPU 头   float16	beta:double	beta/ 输入 C 矩阵的 scale			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/
现 现		float16	input tensor 2	C/ 输入的 C			per-channel
			[batch, C, H]:tensor	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
				batch/ 输入的 batch			
	int8	input_tensor	channel/ 输入的 channel				
		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		
Expand	支持			width/ 输入的 width			
Dapane	~13		shape(batch_o, channel_o,height_o, width_o):tensor	batch_o/ 输出的 batch_o	- - 无限制		
		int64		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
		int8 float16 int64 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			
Where	+++		y_tensor [batch, channel, height, width]:tensor	channel/ 输入的 channel height/ 输入的 height width/ 输入的 width			per-layer
wnere	又付		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		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 width_o/ 输出的 width	<b></b> 无限制		

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
			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
		int8		channel/ 输入的 channel			
ov.Ch.	支持	float16		height/ 输入的 height			
exGlu	又付			width/ 输入的 width			
		int64	axis:int64	axis/ 切分的维度	axis ==1		



				5500/5500 NI O OI XN 711X		机心版电1版仍有限公司	
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
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						



		KK3500/5500 M U U C X X X X X X			和心脉电 1 放历有限公司		
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
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。

瑞芯微电子股份有限公司



## 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				
		int8	input tensor	channel/ 输入的 channel		支持ONNX 规范的四维 tensor 的所	per-layer/ per-channel	
Add/Bias	Add/Bias 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>		已支持
				width/ 输入的 width				
				batch/ 输入的 batch		支持两个 tensor 的广播操作,以	man lavan/	
Cl.	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel		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;		尚不支持
Sub	<b>文</b> 44	float16	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。	per-channel	问小文持
				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				
		int8	input_tensor	channel/ 输入的 channel		支持ONNX 规范的四维 tensor 的所	per-layer/	
Mul/Scale	Mul/Scale 支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制	有广播操作,详见: 注释(1)	per-channel	尚不支持
				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		
Div	部分支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height 3.OP((N,C,H,W),scalar),即以单个标量做 broadcasting;	height/ height 3.OP((N,C,H,W),scalar),即以单个 per-channel 标量做 broadcasting;	元限制 3.OP((N,C,H,W),scalar),即以单个 per-ch		尚不支持
				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	输入的 hatch	ICO	支持两个 tensor 的广播操作,以 ONNX 默认排列 NCHW 做说明,		<b>多核协同</b> 尚不支持
Mov	<b>支</b> 株	int8	input tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制	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	per-layer/	出不去扶
Max 支持	<b>大</b> 14	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。	per-channel	尚不支持
				width/ 输入的 width		说明: <b>A</b> 或 <b>B</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	- 无限制	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 3	×14	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。	per-channel	
				width/ 输入的 width		说明: <b>A</b> 或 <b>B</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	1				
Global	<b>去快</b>	int8	input tensor	channel/ 输入的 channel	[1,8192]		a sa lavon	光工士柱	
AveragePool	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		per-layer	per-layer 尚不支	尚不支持
				width/ 输入的 width	[1,7] (Complier 支持 范围)				
				batch/ 输入的 batch	1				
ClahalMayDaal	支持	int8	input tensor [batch,channel,height,	channel/ 输入的 channel	[1,8192]		mon loven	出不去持	
GlobalMaxPool	<b>大</b> 14	float16	width]:tensor	height/ 输入的 height	[1,343] (RKNN-Toolkit2 支持范围)		per-layer	尚不支持	
				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 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		per-layer	尚不支持
			kernel_w]:int64[]	kernel_w/ width 方向的 kernel 大小	CPU 支持。		per-layer	
				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				
		st	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	[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	F1 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	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 1e-5			
			momentum:double	momentum/ 训练时的滑动平均参数	无限制			<b>多核协同</b> 尚不支持
Batch Normalization	支持	int8 float16		batch/ 输入的 batch	1		per-layer/ per-channel	
1,611114111411		110 401 0	input_tensor [batch,channel,height,	channel/ 输入的 channel			Por commerci	
			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, height/						
			[batch,channel,height, width]:tensor	height/ 输入的 height	无限制			annel
				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	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			



			防止除治	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				<b>多核协同</b> 已支持 尚不支持
Cli.,/D.LLIC	支持	int8	input_tensor	channel/ 输入的 channel	无限制			
Clip/ReLU6	又付	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	一 儿胶制		per-layer	
				width/ 输入的 width				
				batch/ 输入的 batch				
Elu	支持	int8	input tensor [batch,channel,height,	channel/ 输入的 channel	无限制			尚不支持
Elu	又打	float16	width]:tensor	height/ 输入的 height	<b>□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ </b>			间小文村
				width/ 输入的 width				
				batch/ 输入的 batch				
Gelu	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	无限制			出不支持
Geiu	文刊	float16	width]:tensor	height/ 输入的 height				1时7, 文刊
				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				
Relu	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	无限制		man layran	口支铁
Keiu		float16	width]:tensor	height/ 输入的 height	√L PK (P1)		per-layer	多核       己支持       己支持
				width/ 输入的 width				
				batch/ 输入的 batch				
I 1 D 1	-+-++	int8	input_tensor	channel/ 输入的 channel	丁7月4月			
LeakyRelu	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer	己支持
				width/ 输入的 width				
				batch/ 输入的 batch				
				channel/ 输入的 channel	丁7月4月			
PRelu	支持	int8 float16	input_tensor [batch,channel,height, width]:tensor	height/ 输入的 height	无限制		per-layer/ per-channel	己支持
			radii ji teribor	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															
			width]:tensor	input_size/ 输入的 input_size	无限制,建议8对齐															
l			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]		per-layer													
GRU	体命名为 exGRU 算 子,参数	float16	batch size:int64 (extern)	batch size/ 指定 GRU 输入的 batchsize	1			尚不支持												
GKU	项中指明 (extern)	seque sint64 i为 RU 独 J参数 linear reset: input (ext	float16	_	-	: : :	sequence_size :int64 (extern)	sequence_size/ 指定 GRU 输入的 seqsize	无限制,建议4对齐		per-rayer	间小文14								
	的项为 exGRU 独						1	]	1			-	-		(extern)	hidden_size/ GRU 单元中的 hiddensize	无限制,建议8对齐			
1	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。	s,														
			output_layout:string (extern)	output_layout/指定与对应输 出 shape 含义一致的 layout	1.sbnc: 指定 layout 对应的输出 shape 为 [seqs,directions,batches, hidden_size] 2.(sn)c: 指定 layout 对应的输出 shape 为[seqs*batches, directions*input_size,1,1] 要求填写指定的 layout,同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。directions>1 时仅支持 batches=1。															



Operator	支持情况	输入类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同	
			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				
			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	l la			
	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	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-	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- layer/ per channel	per	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 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				
	34.70		input_layout:string (extern)	input_layout/指定与对应输入 shape 含义一致的 layout	1.snc: 指定 layout 对应的输入 shape 为[seqs, batches, input_size] 2.(sn)c: 指定 layout 对应的输入 shape 为[seqs*batches, input_size,1,1] 要求填写指定的 layout,同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。				
			output_layout:string (extern)		output_layout/指定与对应输 出 shape 含义一致的 layout	1.sbnc: 指定 layout 对应的输出 shape 为[seqs,directions,batches, hidden_size] 2.(sn)c: 指定 layout 对应的输出 shape 为[seqs*batches, directions*input_size,1,1] 要求填写指定的 layout,同时要求填写该 op 实际对应的 batch_size、sequence_size、hidden_size。directions>1 时仅支持 batches=1。			



Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
				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:			
Concat	部分支持	int8 float16	width]:tensor	height/ 输入的 height	安州が。州が重:   8bit 数据: 8 对齐,   16bit 数据: 4 对齐   其他方向 concat 无		per-layer	己支持
				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	- 无限制			尚不支持
Mish	XII	float16	width]:tensor	height/ 输入的 height	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\			己支持
				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	hannel/ 输入的 channel	无限制		j	
		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	1 / L PR 市川			
Pad 支	支持			width/ 输入的 width	[1,8176]			尚不支持
T au	XN	float   pads:tensor   float   float		阿尔文的				
		float	constant_value:tensor		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 目前由 CPU 实	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	人口 PR 中山			尚不支持
ReduceMean	, 现 】	float16		width/ 输入的 width				向小文符
			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
			input_tensor	batch/ 输入的 batch channel/ 输入的 channel				
ReduceSum	尚不支持 目	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		per-layer/ per-channel 尚不了	尚不支持
	前由 CPU 实现	float16		width/ 输入的 width				间介文的
			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			
	CPU 上实现。		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	无限制			
		int8	input tensor	channel/ 输入的 channel	[1,8192]			
		float16	[batch,channel,height, 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	[1,8192]			尚不支持
				height_o/ 输出的 height	[1,0192]			
		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)				

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 [batch,channel,height,	channel/ 输入的 channel	无限制			
			width]:tensor	height/ 输入的 height	<b>)</b> / L P P R 市 J			
Reverse Sequence	尚不支持 目前由 CPU 实现	int8 float16		width/ 输入的 width				尚不支持
			batch_axis:int64	batch_axis/ 指定是否为 batch 维度	1		أ	
			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
Signiola	又打	float16	width]:tensor	height/ 输入的 height	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\			时小,文社
				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	channel/ 输入的 channel	无限制		mon layon	尚不支持
nardSigilioid	<b>大</b> 4	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	<b>プレドド 中</b> 北		per-layer	间小义17
				width/ 输入的 width			量化方式 per-layer per-layer	
				batch/ 输入的 batch				
Swish	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	- 无限制			尚不支持
SWISH	<b>大</b> 4	float16	width]:tensor	height/ 输入的 height	<b>プレドド 中</b> 引			间小义17
				width/ 输入的 width				
				batch/ 输入的 batch				
HardSwish	支持	int8	input tensor	channel/ 输入的 channel	无限制		mon layon	尚不支持
naruswish	人村	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	プロソス 中リ		per-layer	四小义村
				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,height,	channel/ 输入的 channel	· 无限制		per-layer	尚不支持
Solipius	<b>文</b> 村	float16	width]:tensor	height/ 输入的 height	】 / L PR 市 J			问个文行
				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		per-layer 尚不	
			axis:int64	axis/ 做 softmax 的轴	1,3 , 即 channel 和 width 方向	和		



Operator	支持情况	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			per-layer 尚不支	
			[batch,channel,heigh t,width]:tensor	height/ 输入的 height	- 无限制			
				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	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				
			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。			



					S NFU OF 文持列表			X 切有 PK 公 印
Operator	支持情况	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	input tensor [batch,channel,height	channel/ 输入的 channel	- 无限制		man layan	<b>多核协同</b> 尚不支持
rann	文4	float16	width]:tensor	height/ 输入的 height	人。以		per-layer	问小又行
				width/ 输入的 width				
				batch/ 输入的 batch	[1,1024]			
			input_tensor [batch,channel,height	channel/ 输入的 channel	[1,8192]			
			, width]:tensor	height/ 输入的 height	[1,0192]			
T	<del>2</del> π /\ <del>-  -  -  -</del>	int8		width/ 输入的 width	[1,8176]			ルナナナ
Transpose	部分支持	float16	perm:int64[]	axis order/ 转置的轴顺序	(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。			<b>问</b> 个文 <del>行</del>



支持情况	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/					
		. , ,		无限制				
				_				
		,		当 <u>dilation_kernel_h</u> > 1 时,				
				width < 16383 此外 对首目输入 width 左在限				
				制,详见 <u>模型输入说明</u>				
			num_output/			man layan/		
				无限制				
		kernel shape [num output,						
		num_input,kernel_h,			-			
		kernel_w]:int64[]	_					
				[1,31]				
			width 方向的 kernel 大小					
支持			stride_h/				已支持	
	110at16	strides[strides_h,		[1 7]		per-channel		
		strides_w]:int64[]		[1,/]				
					ation_kernel_h > 1 时, 16383			
		nads[nads_ton_nads_left						
				[0,15]				
			top 方向的 pads 大小					
			batch/ 输入的 batch channel/ 输入的 channel height/ 输入的 height  width/ 输入的 width  width/ 输入的 width  num_output/ 输出的 channel num input/ 输入的 channel kernel_h/ height 方向的 kernel 大小 stride_h/ height 方向的 strides 大小 stride_w/ width 方向的 strides 大小 pads_left/ left 方向的 pads 大小 pads_right/ right 方向的 pads 大小 pads_right/ right 方向的 pads 大小 pads_left/ left 方向的 pads 大小 pads_right/ right 方向的 pads 大小 pads_right/ right 方向的 pads 大小 pads top/  mt64[]					
			輸入的 batch         channel/height         无限制           height/ 輸入的 height         当 dilation kernel h > 1 时, width 16383 业外, 对首是输入 width 存在限制, 证见模型输入说明           num_output/ 输出的 channel num_input/ 输入的 channel kernel_h/height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小 stride_h/height 方向的 strides 大小 stride_w/ width 万向的 strides 大小 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 group/ group 的大小 dilations_m/         [1,7]           4[]         top 方向的 pads 大小 pads bottom/ bottom 方向的 pads 大小 group/ group 的大小 dilations_m/         [0,15]					
		group:int64		无限制				
					-			
		dilations[dilations h	height 方向的 dilations 大小					
				[1, 32]				
		文材情况 型	input_tensor [batch,channel,height, width]:tensor  kernel_shape [num_output, num_input,kernel_h, kernel_w]:int64[]	支持   地域   地域   地域   地域   地域   地域   地域   地	支持	A	A	



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	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		per-layer/per-channel	
			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]			己支持
			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]			



								THRA H
Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式	多核协同
			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 j.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}			尚不支持
		pads pads	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	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	投入   Maumul		per-layer/ per-channel	
Gemm	尚不支持 , 目前由 CPU	int8	alpha:double	alpha/ 矩阵 A*B 乘法的 scale	- 无限制			尚不支持
Gemm	实现	1 1100116	· 人PR 和J		per-channel	问个义行		
			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				
	部分支持			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	齐要求: 32 对齐 K*C <=65532 C*H <=65532 K*H <=65532		ner-laver/	
MatMul	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;			尚不支持
	输 人 为 feature+ constan	ıre+	[batch, C, H]:tensor	H/ 输入的 H	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			



Operator	支持情况	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
		该支持 · 对 双 e 输入 int8 float16 将支持 入 为 e+	input_tensor_1 [batch,channel,K, N]:tensor	batch/ 输入的 batch	双 feature 时: batch 无限制 channl、K 支持[8,8192],对齐要求为			
	部分支持			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		per-layer/	尚不支持
MatMul	目前该支持 仅针对双 feature输入 未来将支持输入 feature+ constant		input_tensor_2 [batch,channel,N, M]:tensor	K/ 输入的 K	K*N <=65532 K*M <=65532 M*N <=65532			
(4d)				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	14.1 ×14
				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
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch			
		int8		channel/ 输入的 channel	一 无限制		
		float16		height/ 输入的 height			
Expand	支持			width/ 输入的 width			
Expund	ZN			batch_o/ 输出的 batch_o			
		int64	shape(batch_o, channel_o,height_o, width_o):tensor	channel_o/ 输出的 channel	- 无限制		
				height_o/ 输出的 height			
				width_o/ 输出的 width			



Sala vez vila vez Y			1	KK3388 NPU OF 文句列	<i>χ</i>			
Operator	支持情况	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/ 输入的 width	无限制			
		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	- 无限制		
Where	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			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,height,width]:tenso	channel/ 输入的 channel	硬件支持[1,8192]		
				height/ 输入的 height	axis=1, 无限制 - axis=3/-1,width[1, 8192], height 无限制		
				width/ 输入的 width	且受限于 transpose 规格限制		per-layer
exSoftmaxMask			input_tensor_2[batch, channel,height,width]:tensor	batch/ 输入的 batch	1		
exsorumaxiviask	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_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			
		int8		channel/ 输入的 channel	c*h*w 满足如下限制		
avCh.	支持	float16		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			
		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							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



			KK3388 NPU OP X	2147440		- 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	多核协同
Depthwise Convolution + Relu	尚不支持						,	己支持
Depthwise Convolution + Clip	尚不支持							己支持
Depthwise Convolution + PRelu/LeakyRelu	尚不支持							己支持
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 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	同 Mul							
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
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch			per-layer/ per-channel
Add/Bias				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)	
	支持	int8 float16		height/ 输入的 height	无限制		
				width/ 输入的 width			
	支持	int8	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(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-layer/ per-channel
Sub				channel/ 输入的 channel	- 无限制		
Sub				height/ 输入的 height			
				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 支持	+++	int8	input_tensor	channel/ 输入的 channel	- 无限制	支持 ONNX 规范的四维 tensor 的所有	per-layer/
	又付	float16	[batch,channel,height, width]:tensor	height/ 输入的 height		广播操作,详见: 注释(1)	per-channel
				width/ 输入的 width			
		float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	- 无限制		
				channel/ 输入的 channel			
Div	部分支持			height/ 输入的 height			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: <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	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	+++	int8	input tensor	channel/ 输入的 channel	[1,0102]	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/
	又付	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>	
				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	
Min	支持	int8	input_tensor [batch,channel,height,	channel/ 输入的 channel	51.01007	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/
Min	大団	float16	width]:tensor	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	[1,8176]	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	++4	. 40	input tensor	channel/ 输入的 channel	[1,8192]		1
AveragePool	又行		[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		
ClabalManDaal	士性	:40	input tensor	channel/ 输入的 channel	[1,8192]		1
GlobalMaxPool	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)	<b>学</b>	



Operator	支持情况	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	, l	
			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_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,]; 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[]				
MaxPool	支持	int8	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	[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]		



Operator	支持情况	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				
			batch,channel,height,	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	所支持 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: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	1		
CL: /D -LII/	支持	int8	input_tensor	channel/ 输入的 channel			
Clip/ReLU6	文4	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 float16	input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel	- 无限制		
Elu	支持			height/ 输入的 height			
				width/ 输入的 width			
				batch/ 输入的 batch			
	+++	int8	input tensor	channel/ 输入的 channel	T: 17H Ad		
Gelu	支持	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 [batch,channel,height,	channel/ 输入的 channel			per-layer
Keiu	文刊	float16	width]:tensor	height/ 输入的 height	无限制		per-rayer
				width/ 输入的 width			
			batch/ 输入的 batch	1			
L calm D also	支持	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	· into	input_tensor [batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]	<b>艾 C</b>	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		



Operator	支持情况	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	layout  GRU 扩展以及变体命名为 batch_siz 子,参数 float16  gextern int64 (extern) hidden_s (extern)	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
GKO	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		GRU 独	hidden_size:int64 (extern)	hidden_size/ GRU 单元中的 hiddensize	无限制,建议8对齐	
	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			
	如八士壮		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	以及变 令名为 STM 子,参 页中指 int8	hidden_size:int64 (extern)	hidden_size/ LSTM 单元中的 hiddensize	无限制,建议8对齐					
	exLSTM 算子,参		xLSTM 江子,参 江项中指 ((extern)	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			ででは、 int8 int8 int8 int8 int8 int8 int8 int8	· · · · · · · · · · · · · · · · · · ·	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					页为	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					
双·贝。	· 女文之八。		input_layout:string 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 sha input_size,1,1] 要求填写指定的 layout,同时要求均	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
				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	37。 M 37 単: 80 ht 数   据: 8 对齐, 16bit 数   据: 4 对齐   其他方向 concat 无限		per-layer
		width/ 输入的 width  axis:int64  axis:int64  #i 。	制。				
			axis:int64		无限制		
			batch/ 输入的 batch				
Minh	<b>本株</b>	int8	input_tensor	channel/ 输入的 channel	- 无限制		
Mish	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	<b>プレトK 中</b> 7		
				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			
		float16	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		
Pad	支持			width/ 输入的 width	[1,8176]		
rau	义打	int64	pads:tensor	[n_begin,c_begin,h_begin,w_begin,n_end,c_end,h_end,w_end]/输入各轴上前后插入的 pad 大小	目前仅支持: n_begin,c_begin,n_end,c_end 为 1,h_begin,w_begin,h_end,w_end 无限制		
		float	constant_value:tensor	constant_value/ 填充入 pad 的值	无限制		
		string	mode:string	mode/pad 模式	仅支持 constant		
				batch/ 输入的 batch			
			input_tensor	channel/ 输入的 channel	T.199 Au		
D I M	尚不支持 目前	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		
ReduceMean	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	T 152 Add		
D 1 6	尚不支持 目前	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	─ 无限制		per-layer/
ReduceSum	ReduceSum 由 CPU 实现	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	-	
		<del>}</del> 支持	input_tensor [batch,channel,height, width]:tensor	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			height/ 输入的 height	[1,8192]		
Resize	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		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		
		re	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	110万文村	int64		batch_o/ 输出的 batch_o	计算量: alignment=16/type bytes;		
			Shape (batch_o,channel_o, height_o,width_o):tensor	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	יין אין		
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		
				batch/ 输入的 batch			
Sigmoid	支持	支持 int8 float16	input_tensor	channel/ 输入的 channel	- 无限制		
Signification X15	\\ \N		[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
				batch/ 输入的 batch			
HandSiamaid	支持	int8	input_tensor	channel/ 输入的 channel	无限制		mon layran
HardSigmoid	float16	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	一		per-layer
				width/ 输入的 width			
		int8	batch/ 输入的 batch				
Swish	支持			channel/ 输入的 channel	无限制		
SWISH	又1寸			height/ 输入的 height	<b>□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</b>		
				width/ 输入的 width			
				batch/ 输入的 batch			
HandCreich	支持	int8	input tensor	channel/ 输入的 channel	无限制		mon layran
HardSwish	义付	float16 [batch,channel width]:tensor	[batch,channel,height, width]:tensor	height/ 输入的 height	<b>元</b> 岭南山		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			
	古体	int8	input_tensor	channel/ 输入的 channel	工阻却		a on loven
Softplus	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		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	ridth	



Operator	支持情况	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, width]:tensor	channel/ 输入的 channel	m- (b)		
				height/ 输入的 height			
Slice 部分支持	0		width/ 输入的 width				
	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 [batch,channel,height,	channel/ 输入的 channel			
			width]:tensor	height/ 输入的 height			
Split	部分支持	int8 float16		width/ 输入的 width	7L PK (P)		per-layer
			axis:int64	axis/ 切分的维度			
		num_outputs:int64 split: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							
т. 1	支持	int8	input tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel	T. VE Ad		1				
Tanh	14iiii Xiv	float16		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	channel/ 输入的 channel	[1,0102]			
			[batch,channel,height, width]:tensor	height/ 输入的 height	[1,8192]						
								width/ 输入的 width	[1,8176]		
Trongnogo   当(分文文	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<512; 16bit 数据时,h1*w1<8176,w1*c1<512; 16bit 数据时,h1*w1<8176,w1*c1<512; 16bit 数据时,h1*w1<8176。							



Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
Convolution 支持			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			
			kernel_shape [num_output,	num_output/ 输出的 channel num input/ 输入的 channel	制,详见模型输入说明 - 无限制		
	支持	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 stride h/	[1,31]		per-layer/
	λ.,		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]		



Operator	支持情况	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>			
		kernel_shape [num_output,	num output/ 输出的 channel num input/ 输入的 channel				
Depthwise Convolution	支持	strides[st strides_w	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/
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]		For Samuel
			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	[0,15]		
			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/ 输入的 height width/ 输入的 width	<ul> <li></li></ul>		
		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	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[]  group:int64  dilations[dilations h, dilations_w]: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/ 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	tput 时,	
				dilations_h/ height 方向的 dilations 大小 dilations_w/ widtht 方向的 dilations 大小	[1, 32]		

Rockchip <sub>端芯微电子</sub>				RV1103/1106 NPU OP 支持歹	·  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
			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		
	尚不支持目前由	int8	alpha:double	alpha/ 矩阵 A*B 乘法的 scale			per-layer/
Geniin	CPU 实现	into	beta:double	beta/ 输入 C 矩阵的 scale			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	KIT心 telisor 文刊农直		
			input_tensor_1	batch/ 输入的 batch	双 feature 时: batch、H 无限制		
MatMul	尚不支持	尚不支持 目 前 由 int8 CPU 实现	[batch, K, C]:tensor 尚不支持	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/
			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				
	f	float16	width]:tensor	height/ 输入的 height	- 无限制		
Expand	支持			width/ 输入的 width			
Dapana	<b>7</b> 19		shape(batch_o,	batch_o/ 输出的 batch_o			
				channel_o/ 输出的 channel	- 无限制		
	11104		height_o/ 输出的 height	Zurkiri			
			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 int64 int8 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		
			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	- 无限制	
Where	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		per-layer
		int8	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
			input_tensor_1[batch,	batch/ 输入的 batch	无限制		
				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		
				width/ 输入的 width	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		per-layer
exSolullaxiviask	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_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
				batch/ 输入的 batch			
		int8	x_tensor [batch, channel,	channel/ 输入的 channel	c*h*w 满足如下限制		per-layer
exGlu	支持	float16	height, width]:tensor	height/ 输入的 height	8bit 数据: 8 对齐, 16bit 数据: 4 对齐		
exGiu				width/ 输入的 width			
		int64	axis:int64	axis/ 切分的维度	axis ==1		



				K, 1103/1100 11 C O1 2,11/1/4		111101111111111111111111111111111111111	
Operator	支持情况	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						



				KV1103/1100 NTO OT XN 911X			23 13 17 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
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 NPU OF 文持列表				
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
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			
		:40	input tensor	channel/ 输入的 channel			
Add/Bias	支持	float16  [batch,channel,height, width]:tensor  [batch,channel,height, height/输入的 he	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: 注释(1)	per-layer/ per-channel	
				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
Sub	支持			channel/ 输入的 channel			
Sub	<b>文</b> 村			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
				batch/ 输入的 batch		支持 ONNX 规范的四维 tensor 的所有	per-layer/
	-1-44	int8	input_tensor	channel/ 输入的 channel			
Mul/Scale	支持	float16	botob obonnol borott	广播操作,详见: 注释(1)	per-channel		
		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			
Div	部分支持			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
				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
	暂不支持	int8	input tensor	channel/ 输入的 channel			
Max	首小文行	float16	[batch,channel,height, width]:tensor	height/ 输入的 height			
				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: <u>注释(1)</u>	per-layer/ per-channel
Min	暂不支持			channel/ 输入的 channel			
Min	百个文的			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				batch/ 输入的 batch	1		
	-1-44		input tensor	channel/ 输入的 channel	[1,8192]		per-layer
	支持	int8	[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)		
				batch/ 输入的 batch	1		per-layer
ClahalMayDaal	<b>士</b> 柱	int8 float16	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		
		int8	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		
				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			
			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[]				
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 [1,7]; 其它由 CPU 支		per-layer
				kernel_w/ width 方向的 kernel 大小	─ [1,/]; 共巳田 CPU メ  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]		



Operator	支持情况	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	→ <del>/ -</del>	int8		batch/ 输入的 batch	1		per-layer/					
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 [batch,channel,height, width]:tensor  layernorm_weight [channel,height, width]:tensor(const)		channel/ 输入的 channel								
				height/ 输入的 height	无限制							
				width/ 输入的 width								
			[channel,height,	channel/输入的 channel	等于 input_channel							
				height/ 输入的 height	等于 input_height							
Layer	支持 float16	支持 float16	· 挂 float16	float16	float16	支持 float16	[channel,	[channel,height, width]:tensor(const)	width/ 输入的 width	等于 input_width		per-layer
Normalization		normalized_shape		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/ 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			
Clin/Dal II/	古林	int8	input_tensor	channel/ 输入的 channel	- - 无限制		non lavor
Clip/RelU6	Clip/ReLU6 支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height			per-layer
				width/ 输入的 width			
			input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch			
Ele	支持	int8 float16		channel/ 输入的 channel	- - 无限制		
Elu				height/ 输入的 height	一元限制		
				width/ 输入的 width			
				batch/ 输入的 batch			
Cala	<del>+</del> +	int8	input tensor	channel/ 输入的 channel	工作出		
Gelu	支持	て行   floot16   [□	[batch,channel,height, width]:tensor	height/ 输入的 height	一 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 [batch,channel,height,	channel/ 输入的 channel			per-layer
Keiu	文刊	float16	width]:tensor	height/ 输入的 height	无限制		per-rayer
				width/ 输入的 width			
			batch/ 输入的 batch	1			
L calm Dala		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			
PRelu	支持	克持   Into   [batch,ch	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	限制 4 对齐							
			width jieonsor	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: 輸入 shape 为:     [seq_length, batch_size, input_size]     输出 shape 为:     [seq_length, num_directions, batch_size, hidden_size] 1: 輸入 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	RU 算 参数 float16 中指明 ktern ) 项 为 RU 独	float16	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								ern)	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_layou (extern)	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							
		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						
	如八十井		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	以及变 命名为 LSTM 子,参 int8 项中指 float16	hidden size:int64 (extern)	hidden size/ LSTM 单元中的 hiddensize	无限制,建议8对齐						
	exLSTM 算子,参		LSTM 子,参 int8 项中指 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						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		has projection:int64 (extern)	has_projection/ projection 变种	1(T) or 0(F) 目前限定 0,即尚不支持						
· 女子只。	<b>双</b> -贝。	extern output_la	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	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		
Concat	部分支持	int8 float16	width]:tensor	输入的 height 据: 4对齐	据: 8 对齐, 16bit 数		per-layer
		width/ 输入的 width  axis:int64  aixs/ 拼接的维度  无限制	制。				
			axis:int64		无限制		
		int8 float16 input_tensor [batch,channel,heig width]:tensor		batch/ 输入的 batch			
Mish	古柱		input_tensor	channel/ 输入的 channel	- 无限制		
IVIISII	<b>文</b> 特			height/ 输入的 height	) June (Pri		
				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 [batch,channel,height,	hannel/ 输入的 channel	无限制		
		float16	width]:tensor	height/ 输入的 height	<b>万山</b>		
Pad	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			
		float	constant_value:tensor	constant_value/ 填充入 pad 的值	无限制		
			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		
DadwaaMaan	尚不支持 目	int8	[batch,channel,height, width]:tensor	height/ 输入的 height	/LPR中		
ReduceMean 前由	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		



0 4		*************************************		S302 NPU OF 文持列农	かまを		」 放切有限公司
Operator	支持情况	数据类型	输入	输入参数	<b>  约束规格</b> 	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		
ReduceSum	尚不支持 目前	int8	width]:tensor	height/ 输入的 height	- J-LPK (PJ		per-layer/
ReduceSum	由 CPU 实现	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		
		前 NPU 仅支 宽高方向不 过 8 倍的整倍 int8 的最近邻和 float16	input_tensor [batch,channel,height, width]:tensor	batch/ 输入的 batch	支持多 batch	per-la	
	部分支持			channel/ 输入的 channel	[1,8192]		
	目前 NPU 仅支			height/ 输入的 height			
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,		-	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
分的会	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			
		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		
Daghana	支持			width/ 输入的 width			
Reshape	<b>大</b> 14	int64	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 [batch,channel,height,	channel/ 输入的 channel	无限制		
			width]:tensor	height/ 输入的 height			
Reverse Sequence				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	channel/ 输入的 channel	- 无限制		
Signioid	<b> </b>	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	ן ייין אא טיי		
				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			
HandSiamaid	古体	int8	input_tensor	channel/ 输入的 channel	- - 无限制		man lavan
HardSigmoid	支持	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	一人吹响		per-layer
				width/ 输入的 width			
				batch/ 输入的 batch			
Carriela	古体	持 int8 float16	input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel	工四生		
Swish	支持			height/ 输入的 height	一 无限制		
				width/ 输入的 width			
				batch/ 输入的 batch			
H 10 11	+-++	int8	input tensor	channel/ 输入的 channel	工作化		1
HardSwish	支持	支持 float16 [batch,channel,height, width]:tensor	[batch,channel,height, width]:tensor	height/ 输入的 height	一 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			
C.A.L. ±±	支持	int8	input_tensor	channel/ 输入的 channel	工阳生		a sa lavos
Softplus	又行	float16	[batch,channel,height, width]:tensor	height/ 输入的 height	- 无限制		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	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		



Operator	支持情况	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 100 Ad			
			[batch,channel,height, width]:tensor	height/ 输入的 height	无限制			
ints	int8	nt8	width/ 输入的 width					
Slice		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:int64[]	axes/ 选取切分的轴	支持任意 0~3 轴, 支持同时多轴选择	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	分支持 int8 float16		width/ 输入的 width			per-layer	
			axis:int64	axis/ 切分的维度				
			num_o	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	Tanh 支持	int8	input tensor [batch,channel,height,	channel/ 输入的 channel	无限制		per-layer
		float16	width]:tensor	height/ 输入的 height			per any or
				width/ 输入的 width			
				batch/ 输入的 batch	无限制		
			input_tensor [batch,channel,height, width]:tensor	channel/ 输入的 channel			
				height/ 输入的 height	[1,8192]		
				width/ 输入的 width	[1,8176]		
Transpose 支持	支持	int8 float16	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/ 输入的 batch channel/ 输入的 channel	无限制		
			[batch,channel,height, width]:tensor	height/ 输入的 height			
			width/ 输入的 width	当 dilation_kernel_h > 1 时, width < 16383 此外,对首层输入 width 存在限制,详见模型输入说明			
		kernel shape	num_output/ 输出的 channel num input/				
		[num_output, num_input,kernel_h,	输入的 channel kernel h/				
		持 int8	kernel_w]:int64[]	height 方向的 kernel 大小 kernel_w/	[1,31]		
Convolution	支持		strides[strides_h, strides_w]:int64[]	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	[1,7]		per-layer/ per-channel
			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 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,	[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				
Depthwise Convolution	支持	支持 int8 strides[st strides_w] pads[pad pads_left pads_right]	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	
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-enamer	
			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]			



Operator	支持情况	数据类型	输入	输入参数	约束规格	广播支持	量化方式
ConvTranspose/ Deconvolution 支持			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	五限制  当 <u>dilation kernel h</u> > 1 时, width < 16383 此外,对首层输入 width 存在限制,详		
			kernel_shape [num_output, num_input,kernel_h, kernel_w]:int64[]	num_output/ 输出的 channel num input/ 输入的 channel kernel_h/	- 无限制		
	支持	int8	strides[strides_h, strides_w]:int64[]	height 方向的 kernel 大小 kernel_w/ width 方向的 kernel 大小 stride_h/ height 方向的 strides 大小 stride_w/	· [1,31] · [1,8]	-	per-layer/ per-channel
		pads[pads top, pads_left, pads_bottom, pads_right]:int64[]	width 方向的 strides 大小 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		数据类型	输入	· 输入参数		广播支持	量化方式
		<i></i>	input_tensor_1 [M, K]:tensor  input_tensor_2 [K,N]:tensor	M,K,N/ 输入数据的形状	转为 Matmul 实现,约束同 Matmul	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
Gemm 尚不支持 目前 由 CPU 实现	int8	alpha:double	alpha/ 矩阵 A*B 乘法的 scale	<b></b> 无限制		per-layer/	
		beta:double	beta/ 输入 C 矩阵的 scale			per-channer	
			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				
	部分支持		input_tensor_1[batch,	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		
MatMul (4d) 输入 未来将3 入	针对双 feature 输入	int8		K/ 输入的 K	feature+constant 时: 若 input_tensor_1 为 feature,则转为 batch		per-layer/ per-channel
	/ ·	input_tensor_2[batch, channel, N, M]:tensor		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	无限制		
	float16	width]:tensor	height/ 输入的 height	プロドル   プロドル			
Expand	<b>支</b> 挂	支持 int64		width/ 输入的 width			per-layer
Expand	XM		shape(batch_o,	batch_o/ 输出的 batch_o			1 ,
				channel_o/ 输出的 channel	一 无限制		
			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 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			
		int8 float16 int64 bool int8 float16	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			
Where	文持		int8 float16  mask_tensor[batch, channel,height, width]:tensor  shape(batch_o, channel_o,height_o, width_o):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
				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		
				width/ 输入的 width	且受限于 transpose 规格限制		per-layer
exSoftmaxMask	部分	int8	float16	batch/ 输入的 batch	1		
exSolullaxiviask	支持	float16		channel/ 输入的 channel	硬件支持[1,8192]		per-rayer
				height/ 输入的 height	1		
				width/ 输入的 width	axis=1: [1] axis=3/-1,[1, 8192]		
				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
				batch/ 输入的 batch			
		int8	x_tensor [batch, channel,	channel/ 输入的 channel	c*h*w 满足如下限制		
exGlu	支持	float16	height, width]:tensor	height/ 输入的 height	8bit 数据: 8 对齐, 16bit 数据: 4 对齐		nor lover
exGlu	又付			width/ 输入的 width			per-layer
		int64	axis:int64	axis/ 切分的维度	axis ==1		



				THE STOP IN COLUMN TO		1000000	
Operator	支持情况	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						



				KK3302 IVI O OI XIII 9148		机心灰电子及	D3 13174 3
Operator	支持情况	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						



				KK3302 NI O OI XN 713X		- 州心灰电 1 及[	
Operator	支持情况	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					
Convolution + add + Relu	支持	同 Convolution					

#### 注释:

### (1) 广播说明:

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

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

### 广播支持举例:

- 1. OP(A(N,C,H,W), B(N,C,H,W)): OP(A(1,16,32,8), B(1,16,32,8)) = C(1,16,32,8)
- 2. OP(A(N,C,H,W), B(C,1,1)): OP(A(1,16,32.8), B(16)) = C(1,16,32.8)
- 3. OP(A(N,C,H,W), B(scalar)): OP(A(1,16,32,8), B(1)) = C(1,16,32,8)
- 4. OP(A(N,C,H,W), B(H,W)): OP(A(1,16,32,8), B(32x8)) = C(1,16,32,8)

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



# 第五章 CPU OP 支持列表



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



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



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



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



## 第六章 GPU OP 支持列表



Operator	描述	规格约束	说明
MatMul	多维矩阵相乘	无限制(文持四维X四维、四维X二维订异) 	只支持float16,需设置GPU优先(参考 《Rockchip_RKNPU_User_Guide_RKNN_S DK》)



## 第七章 模型输入输出规格说明



### 1. 模型输入说明

- <del>  -</del>  -  -	構刑光目	<i>t</i> A )	<b>学</b> 早.九	/ 1 /	输入宽(wid 单位:元素个	th)对齐要求 〉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(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) ConvTranspose/Deconvolution: N = 8 - CEIL( (dilation_kernel_h * dilation_kernel_w) / 128)	
	int8		int8					无限制
into			float16	CPU				
		4 维度	其他类型 (* <u>注释 8</u> )					
RK3566 /3568			uint8	· CPU	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			
	float16		int8		4	1	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:	无限制
Hoatre	1104110		float16		·		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_h * dilation_kernel_w) / 128)	<b>万□ ÞR</b> 申リ
			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



- <del>++-</del> 11.	<b>推到</b>	<i>t</i> A.)	<b>光</b> 日 江 閏 <i>松</i>	1 1 /	输入宽(wid 单位:元素个	th)对齐要求 〉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: 1. width * dilation_kernel_h <= 2048 * N	
	int8   int8   16   1	1	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:	于四集山				
			float16	CPU		di Do * N	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 )	
		4 维度	其他类型 (* <u>注释 8</u> )					
RK3588			uint8	- СРИ			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	
	float16		int8		8	1	2. width <= 8192 其中 N 必须为 1 到 7 的整数,超出范围的卷积不受支持,各卷积类型 N 的计算方式如下: Convolution: N = 12 - MAX( CEIL( (dilation_kernel_h * dilation_kernel_w) / 128), 3) Depthwise Convolution: N = 12 - MAX( CEIL( (dilation_kernel_h * dilation_kernel_w) / 2048), 3) ConvTranspose/Deconvolution: N = 12 - MAX( CEIL( (dilation_kernel_h * dilation_kernel_w) / 128), 3)  各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式:	
Hoativ			float16				dilation_kernel_w) / 128 ), 3 ) Depthwise Convolution: N = 12 - MAX( CEIL( (dilation_kernel_h * dilation_kernel_w) / 2048 ), 3 ) ConvTranspose/Deconvolution:	751,000
			其他类型 (* <u>注释 8</u> )					
	无限制	非4维	非限制	СРИ	1	1	无限制	无限制



<del>-11-</del> 11.	体型光目	<i>t</i> A.)	<b>光</b> 日 江 昭 <b>松</b>	/ 1 /	输入宽(wid 单位:元素个	th)对齐要求 〉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	_		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:  1. width * dilation_kernel_h <= 2048 * N  2. width <= 4096 其中 N 必须为 1 到 7 的整数,超出范围的卷积不受支持,各卷积类型 N 的计算方式如下: Convolution: N = 8 - MAX( CEIL( (dilation_kernel_h * dilation_kernel_w) / 128 ), 2 ) Depthwise Convolution: N = 8 - CEIL( (dilation_kernel_h * dilation_kernel_w) / 4096 ) ConvTranspose/Deconvolution: N = 8 - MAX( CEIL( (dilation_kernel_h * dilation_kernel_w) / 128 ), 2 )	无限制
	int8		int8		- 16	1		
			float16	CPU				
		4 维度	其他类型 (* <u>注释 8</u> )	CPU				
RK3562			uint8	- CPU			各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式:	
	float16		int8		8	1	当输入通道(channel)为 1,3,4(声明见 <u>注释 9</u> ) (channe 非 1,3,4  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 )	
Tioa	Hourto		float16			1	dilation_kernel_w) / 128 ), 2 ) Depthwise Convolution: N = 8 - CEIL( (dilation_kernel_h * dilation_kernel_w) / 4096 )	JUPK IP 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



	模型首层精度类型	输入 维度	首层设置输入数据类型	mean/scale/quant 后端实现设备	输入宽(width)对齐要求 单位:元素个数		输入宽(width) 大小限制	
					当输入通道 (channel) 为 1,3,4	当输入通道 (channel) 非 1,3,4	当输入通道(channel)为 1,3,4(声明见注释 9)	当输入通道 (channel) 非 1,3,4
RV1103/ 1106	int8	4 维度	uint8		16	1	各卷积类型的 width/kernel_h/kernel_w 需要满足以下两式:  1. width * dilation_kernel_h <= 2048 * N  2. width <= 4096 其中 N 必须为 1 到 7 的整数,超出范围的卷积不受支持,各卷积类型 N 的计算方式如下: Convolution: N = 8 - MAX( CEIL( (dilation_kernel_h * dilation_kernel_w) / 128 ), 2 ) Depthwise Convolution: N = 8 - CEIL( (dilation_kernel_h * dilation_kernel_w) / 4096 ) ConvTranspose/Deconvolution: N = 8 - MAX( CEIL( (dilation_kernel_h * dilation_kernel_w) / 128 ), 2 )	
			int8	NPU				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
	int8	- 4 维度	NCHW	无	无
			NHWC	8 对齐( <u>*注释 1</u> )	无
			NC1HWC2	最后一层卷积类算子, 16 对齐, 最后一层非卷积类算子 8 对齐	H*W 要 4 对齐
			UNDEFINE	无	无
RK3566/3568	float16		NCHW	无	无
			NHWC	4 对齐( <u>*注释 1</u> )	
			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
	int8	- 4 维度	NCHW	无	无
			NHWC	16 对齐( <u>*注释 1</u> )	
			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	float16		NCHW	无	无
			NHWC	8 对齐( <u>*注释 1</u> )	无
			NC1HWC2	最后一层卷积类算子, 16 对齐, 最后一层非卷积类算子 8 对齐	H*W 要 4 对齐
			UNDEFINE	无	无
	无限制	非 4 维	UNDEFINE	无	无



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



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

### 注释:

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