密级状态：绝密(　)　秘密(　)　内部(　)　公开(√)

RKNN Compiler Support Operator List

（技术部，图形计算平台中心）

文件状态： [　] 正在修改 [√] 正式发布	当前版本：	V1.5.0
	作　者：	NPU团队
	编　辑：	刘雯君
	审　核：	熊伟
	完成日期：	2023-05-22

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多核协同运行支持情况 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 3. 更新输入大分辨率规格支持	熊伟

目 录

第一章 RK3566/3568 NPU Operator List

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Div	部分支持	float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制	支持两个tensor的广播操作，以ONNX默认排列NCHW做说明，支持以下广播方式： 1、OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的tensor进行操作 2、OP(A(N,C,H,W),B(C,1,1))，即C维度做broadcasting 3、OP((N,C,H,W),scalar)，即以单个标量做broadcasting 4、OP(A(N,C,H,W),B(H,W))，即HW维度做broadcasting，目前仅支持 FP16类型说明：A或B都可以作为广播方。 例子见注释（1）	per-layer/ per-channel
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Max	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制	支持两个tensor的广播操作，以ONNX默认排列NCHW做说明，支持以下广播方式： 1、OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的tensor进行操作 2.OP(A(N,C,H,W),B(C,1,1))，即C维度做broadcasting 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做broadcasting 说明：A或B都可以作为广播方。	per-layer/ per-channel
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		
Min	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制	支持两个tensor的广播操作，以ONNX默认排列NCHW做说明，支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的tensor进行操作 2.OP(A(N,C,H,W),B(C,1,1))，即C维度做broadcasting 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做broadcasting 说明：A或B都可以作为广播方。	per-layer/ per-channel
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Global AveragePool	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height	[1,343]（toolkit2支持范围）		
				width/ 输入的width			
GlobalMaxPool	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height	[1,343]（toolkit2支持范围）		
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
AveragePool	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height	[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		
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height方向的kernel大小	无限制，NPU支持[1,7]；其它由CPU支持。		
				kernel_w/ width方向的kernel大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,7]		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,8]		
				stride_w/ width方向的strides大小			

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

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Clip/ReLU6	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Elu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Gelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Relu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
LeakyRelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
PRelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
				slope/ PReLU系数	仅支持单个标量或C维度系数		

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

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

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Pad	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		
				channel/ 输入的channel	无限制		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		
		int64	pads:tensor	[n_begin,c_begin,h_begin,w_begin,n_end,c_end,h_end,w_end]/ 输入各轴上前后插入的pad大小	目前仅支持n_begin,c_begin,n_end,c_end为1 h_begin,w_begin,h_end,w_end无限制		
		float	constant_value:tensor	constant_value/ 填充入pad的值	无限制		
		string	mode:string	mode/pad模式	仅支持constant		
ReduceMean	尚不支持 目前由CPU实现	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width			
			axes:int64[]	axes/ 指定reduce的轴	单轴:无限制，多轴:{2,3}		
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0		

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
ReduceSum	尚不支持 目前由CPU实现	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width			
			axes:int64[]	axes/ 指定reduce的轴	单轴:无限制，多轴:{2,3}		
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0		
Resize	部分支持 目前NPU仅支持宽高方向不超过8倍的整倍数的最邻近插值缩放，其余不支持部分的会Fallback到CPU上实现。	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	支持多batch		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	1.[1,8176] 2.设放大倍数为s（s为正整数），width*s*(s-1)<=8192		
			mode:string	mode/resize采用的模式	仅支持nearest		
			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	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Reshape	部分支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		
		int64	shape (batch_o, channel_o, height_o, width_o):tensor	batch_o/ 输出的batch_o	无限制		
				channel_o/ 输出的channel	[1,8192]		
				height_o/ 输出的height			
				width_o/ 输出的width	[1,8176]		
				[n,c,h1,w1]->[n,c,h2,w2]/ (h1*w1=h2*w2)	支持		
				[1,c,h,w]->[c1,hw1,1,1]/ (c1=c/a, h*w=hw1/a,a为整数)	不支持		
				[n,c,1,1]->[1,n1,h,w]/ (c=h*w/a, n1=n/a, a为整数)			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Reverse Sequence	尚不支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		
			batch_axis:int64	batch_axis/ 指定是否为batch维度	1		
			time_axis:int64	time_axis/ 指定是否为time维度	0		
			sequence_lens:int64[]	sequence_lens/ 指定序列翻转的数量	仅支持channel数		
Sigmoid	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
HardSigmoid	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Swish	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
HardSwish	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Softplus	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Softmax	尚不支持，目前由CPU实现	float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel	[1,8192] 建议4对齐		
				height/ 输入的height	1		
				width/ 输入的width			
			axis:int64	axis/ 做softmax的轴	1,即channel方向		

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Slice	部分支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
			starts:int64[]	start/ 切分的起始位置	channel方向Slice时，channel_start要对齐。 对齐量： 8bit数据：8对齐，16bit数据：4对齐。 其他方向无限制。		
			ends:int64[]	ends/ 切分的终止位置	channel方向Slice时，channel_end要对齐。 对齐量： 8bit数据：8对齐，16bit数据：4对齐。 其他方向无限制。		
			axes:int64[]	axes/ 选取切分的轴	支持任意0~3轴， 支持同时多轴选择		
			steps:int64[]	steps/ 选取切分对应轴的步长	1		
Split	部分支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
			axis:int64	axis/ 切分的维度			
			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	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Tanh	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Transpose	部分支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	[1,1024]		
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		
			perm:int64[]	axis order/ 转置的轴顺序	仅支持 （1）perm=[3,1,2,0],in_shape=[n,c,1,1],且n,c要求8bit数据：8对齐，16bit数据：4对齐。 （2）perm=[3,1,2,0],in_shape=[1,c,1,w],且w,c要求8bit数据：8对齐，16bit数据：4对齐。 （3）perm=[2,1,0,3],in_shape=[n,c,1,1],且n,c要求8bit数据：8对齐，16bit数据：4对齐。 （4）perm=[2,1,0,3],in_shape=[1,c,h,1],且h,c要求8bit数据：8对齐，16bit数据：4对齐。		

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Convolution	支持	int8 float16	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	无限制		
				height/ 输入的height	无限制		
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明		
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制		
				num_input/ 输入的channel			
				kernel_h/ height方向的kernel大小	[1,31]		
				kernel_w/ width方向的kernel大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,7]		
				stride_w/ width方向的strides大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,15]		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			group:int64	group/ group的大小	无限制		
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]		
				dilations_w/ widtht方向的dilations大小			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Depthwise Convolution	支持	int8 float16	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	无限制		
				height/ 输入的height	无限制		
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明		
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制		
				num_input/ 输入的channel	无限制		
				kernel_h/ height方向的kernel大小	[1,8]		
				kernel_w/ width方向的kernel大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,7]		
				stride_w/ width方向的strides大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,15]		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]		
				dilations_w/ widtht方向的dilations大小			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
ConvTranspose/ Deconvolution	支持	int8 float16	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	无限制		
				height/ 输入的height	无限制		
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明		
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制		
				num_input/ 输入的channel	无限制		
				kernel_h/ height方向的kernel大小	[1,31]		
				kernel_w/ width方向的kernel大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	{2,4,8}		
				stride_w/ width方向的strides大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	支持0-15 设置pad时注意: 不支持 kernel_h * dilations_h - dilations_h - pads_top < 0 不支持 kernel_w * dilations_w - dilations_w - pads_left < 0 不支持 stride_h *(height - 1) - pads_top + 1 < output_h 不支持 stride_w *(width - 1) - pads_left + 1 < output_w		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			group:int64	group/ group的大小	1 当且仅当num_input=num_output时，支持num_output		
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]		
				dilations_w/ widtht方向的dilations大小			

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Expand	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
		int64	shape (batch_o, channel_o, height_o, width_o):tensor	batch_o/ 输出的batch_o	无限制		
				channel_o/ 输出的channel			
				height_o/ 输出的height			
				width_o/ 输出的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Convolution + Relu	支持	同Convolution					
Convolution + Clip	支持						
Convolution + PRelu/LeakyRelu	支持						
Convolution + Add	支持						
Convolution + Mul	尚不支持						
Convolution + Sigmoid	尚不支持						
Convolution + Tanh	尚不支持						
Convolution + Softplus	尚不支持						
Convolution + HardSigmoid	尚不支持						
Convolution + HardSwish	尚不支持						
Convolution + Elu	支持						
Convolution + Swish	尚不支持						
Convolution + Mish	尚不支持						

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
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						

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Depthwise Convolution + Relu	支持	同Depthwise Convolution					
Depthwise Convolution + Clip	支持						
Depthwise Convolution + PRelu/LeakyRelu	支持						
Depthwise Convolution + Add	尚不支持						
Depthwise Convolution + Mul	尚不支持						
Depthwise Convolution + Sigmoid	尚不支持						
Depthwise Convolution + Tanh	尚不支持						
Depthwise Convolution + Softplus	尚不支持						
Depthwise Convolution + HardSigmoid	尚不支持						
Depthwise Convolution + HardSwish	尚不支持						
Depthwise Convolution + Elu	支持						
Depthwise Convolution + Swish	尚不支持						
Depthwise Convolution + Mish	尚不支持						

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
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)
设计建议：当除数是常量时，建议转换成除数倒数的乘法。乘法在运算效率显著大于除法。

(2) 约束规格中，[a,b]表示支持a-b；{a,b,c}表示支持a,b,c。

第二章 RK3588 NPU Operator List

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Div	部分支持	float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制	支持两个tensor的广播操作，以ONNX默认排列NCHW做说明，支持以下广播方式： 1、OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的tensor进行操作 2、OP(A(N,C,H,W),B(C,1,1))，即C维度做broadcasting 3、OP((N,C,H,W),scalar)，即以单个标量做broadcasting 4、OP(A(N,C,H,W),B(H,W))，即HW维度做broadcasting，目前仅支持FP16类型 说明：A或B都可以作为广播方。 例子见注释（1）	per-layer/ per-channel	尚不支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				
Max	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制	支持两个tensor的广播操作，以ONNX默认排列NCHW做说明，支持以下广播方式： 1、OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的tensor进行操作 2.OP(A(N,C,H,W),B(C,1,1))，即C维度做broadcasting 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做broadcasting 说明：A或B都可以作为广播方。	per-layer/ per-channel	尚不支持
				channel/ 输入的channel	[1,8192]			
				height/ 输入的height				
				width/ 输入的width	[1,8176]			
Min	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制	支持两个tensor的广播操作，以ONNX默认排列NCHW做说明，支持以下广播方式： 1.OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的tensor进行操作 2.OP(A(N,C,H,W),B(C,1,1))，即C维度做broadcasting 3.OP(A(N,C,H,W),B(scalar))，即以单个标量做broadcasting 说明：A或B都可以作为广播方。	per-layer/ per-channel	尚不支持
				channel/ 输入的channel	[1,8192]			
				height/ 输入的height				
				width/ 输入的width	[1,8176]			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Global AveragePool	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer	尚不支持
				channel/ 输入的channel	[1,8192]			
				height/ 输入的height	[1,343]（toolkit2支持范围）			
				width/ 输入的width				
GlobalMaxPool	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer	尚不支持
				channel/ 输入的channel	[1,8192]			
				height/ 输入的height	[1,343]（toolkit2支持范围）			
				width/ 输入的width				

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
AveragePool	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer	尚不支持
				channel/ 输入的channel	[1,8192]			
				height/ 输入的height	[1,8192]			
				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			
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height方向的kernel大小	无限制，NPU支持[1,7]；其它由CPU支持。			
				kernel_w/ width方向的kernel大小				
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,7]			
				pads_right/ right方向的pads大小				
				pads_top/ top方向的pads大小				
				pads_bottom/ bottom方向的pads大小				
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,8]			
				stride_w/ width方向的strides大小				

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

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Clip/ReLU6	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer	已支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				
Elu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制			尚不支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				
Gelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制			尚不支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Relu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer	已支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				
LeakyRelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer	已支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				
PRelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel	已支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				
				slope/ PReLU系数	仅支持单个标量或C维度系数			

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

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

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Pad	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1			尚不支持
				channel/ 输入的channel	无限制			
				height/ 输入的height				
				width/ 输入的width	[1,8176]			
		int64	pads:tensor	[n_begin,c_begin,h_begin,w_begin,n_end,c_end,h_end,w_end]/ 输入各轴上前后插入的pad大小	目前仅支持n_begin,c_begin,n_end,c_end为1 h_begin,w_begin,h_end,w_end无限制			
		float	constant_value:tensor	constant_value/ 填充入pad的值	无限制			
		string	mode:string	mode/pad模式				
ReduceMean	尚不支持 目前由CPU实现	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制			尚不支持
				channel/ 输入的channel	[1,8192]			
				height/ 输入的height				
				width/ 输入的width				
			axes:int64[]	axes/ 指定reduce的轴	单轴:无限制，多轴:{2、3}			
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
ReduceSum	尚不支持 目前由CPU实现	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel	尚不支持
				channel/ 输入的channel	[1,8192]			
				height/ 输入的height				
				width/ 输入的width				
			axes:int64[]	axes/ 指定reduce的轴	单轴:无限制，多轴:{2、3}			
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0			
Resize	部分支持 目前NPU仅支持宽高方向不超过8倍的整倍数的最邻近插值缩放，其余不支持部分的会Fallback到CPU上实现。	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	支持多batch		per-layer	尚不支持
				channel/ 输入的channel	[1,8192]			
				height/ 输入的height				
				width/ 输入的width	1.[1,8176] 2.设放大倍数为s（s为正整数），width*s*(s-1)<=8192			
			mode:string	mode/resize采用的模式	仅支持nearest			
			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	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Reshape	部分支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制			尚不支持
				channel/ 输入的channel	[1,8192]			
				height/ 输入的height				
				width/ 输入的width	[1,8176]			
		int64	shape (batch_o, channel_o, height_o, width_o):tensor	batch_o/ 输出的batch_o	无限制			
				channel_o/ 输出的channel	[1,8192]			
				height_o/ 输出的height				
				width_o/ 输出的width	[1,8176]			
				[n,c,h1,w1]->[n,c,h2,w2]/ (h1*w1=h2*w2)	支持			
				[1,c,h,w]->[c1,hw1,1,1]/ (c1=c/a, h*w=hw1/a, a为整数)	当c,c1,hw,hw1均（i8 16对齐，fp16 8对齐）时支持			
				[n,c,1,1]->[1,n1,h,w]/ (c=h*w/a, n1=n/a, a为整数)	当n,c,n1,hw均（i8 16对齐，fp16 8对齐）时支持			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Reverse Sequence	尚不支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制			尚不支持
				channel/ 输入的channel	[1,8192]			
				height/ 输入的height				
				width/ 输入的width	[1,8176]			
			batch_axis:int64	batch_axis/ 指定是否为batch维度	1			
			time_axis:int64	time_axis/ 指定是否为time维度	0			
			sequence_lens:int64[]	sequence_lens/ 指定序列翻转的数量	仅支持channel数			
Sigmoid	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制			尚不支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
HardSigmoid	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer	尚不支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				
Swish	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制			尚不支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				
HardSwish	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer	尚不支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Softplus	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer	尚不支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				
Softmax	尚不支持，目前由CPU实现	float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer	尚不支持
				channel/ 输入的channel	[1,8192] 建议8对齐			
				height/ 输入的height	1			
				width/ 输入的width				
			axis:int64	axis/ 做softmax的轴	1,即channel方向			

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Tanh	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer	尚不支持
				channel/ 输入的channel				
				height/ 输入的height				
				width/ 输入的width				
Transpose	部分支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制			尚不支持
				channel/ 输入的channel	[1,8192]			
				height/ 输入的height				
				width/ 输入的width	[1,8176]			
			perm:int64[]	axis order/ 转置的轴顺序	RK3588支持所有RK3566/3568上支持的transpose操作， 在该基础上支持：n轴不参与转置时允许c、h、w三轴如下四种转置。限制与说明如下： 1、假设in_shape[n1,c1,h1,w1],out_shape[n2,c2,h2,w2] 2、四种转换分别为(1）perm=[0,2,3,1]，NCHW->NHWC。（2）perm=[0,2,1,3]，NCHW->NHCW。（3）perm=[0,3,1,2]，NCHW->NWCH。（4）perm=[0,3,2,1]，NCHW->NWHC。 3、以上四种转置无对齐要求。但在满足对齐要求时效率更高。对齐要求为：第1点中参数的c1、c2均要满足8bit数据：16对齐，16bit数据：8对齐。 4、NPU限制项：（1）perm=[0,2,3,1]时，8bit数据时，h1*w1<8176，w1*c1<512；16bit数据时，h1*w1<8176，w1*c1<1023。（2）perm=[0,3,1,2]时，h1*w1<8176。（3）perm=[0,3,2,1]时，h1*w1<16384,h2*w2<8*8192, w1<1024。			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Convolution	支持	int8 float16	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel	已支持
				channel/ 输入的channel	无限制			
				height/ 输入的height	无限制			
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明			
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制			
				num_input/ 输入的channel				
				kernel_h/ height方向的kernel大小	[1,31]			
				kernel_w/ width方向的kernel大小				
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,7]			
				stride_w/ width方向的strides大小				
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,15]			
				pads_right/ right方向的pads大小				
				pads_top/ top方向的pads大小				
				pads_bottom/ bottom方向的pads大小				
			group:int64	group/ group的大小	无限制			
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]			
				dilations_w/ widtht方向的dilations大小				

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Depthwise Convolution	支持	int8 float16	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel	已支持
				channel/ 输入的channel	无限制			
				height/ 输入的height	无限制			
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明			
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制			
				num_input/ 输入的channel	无限制			
				kernel_h/ height方向的kernel大小	[1,8]			
				kernel_w/ width方向的kernel大小				
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,31]			
				stride_w/ width方向的strides大小				
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,15]			
				pads_right/ right方向的pads大小				
				pads_top/ top方向的pads大小				
				pads_bottom/ bottom方向的pads大小				
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]			
				dilations_w/ widtht方向的dilations大小				

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
ConvTranspose/ Deconvolution	支持	int8 float16	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel	尚不支持
				channel/ 输入的channel	无限制			
				height/ 输入的height	无限制			
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明			
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制			
				num_input/ 输入的channel	无限制			
				kernel_h/ height方向的kernel大小	[1,31]			
				kernel_w/ width方向的kernel大小				
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	{2,4,8}			
				stride_w/ width方向的strides大小				
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	支持0-15 设置pad时注意: 不支持 kernel_h * dilations_h - dilations_h - pads_top < 0 不支持 kernel_w * dilations_w - dilations_w - pads_left < 0 不支持 stride_h *(height - 1) - pads_top + 1 < output_h 不支持 stride_w *(width - 1) - pads_left + 1 < output_w			
				pads_right/ right方向的pads大小				
				pads_top/ top方向的pads大小				
				pads_bottom/ bottom方向的pads大小				
			group:int64	group/ group的大小	支持1 当且仅当num_input=num_output时，支持num_output			
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]			
				dilations_w/ widtht方向的dilations大小				

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Gemm	尚不支持 目前由CPU实现	int8	input_tensor_1 [M, K]:tensor	M,K,N/ 输入数据的形状	转为Matmul实现，约束同Matmul		per-layer/ per-channel	尚不支持
			input_tensor_2 [K,N]:tensor					
			alpha:double	alpha/ 矩阵A*B乘法的scale	无限制			
			beta:double	beta/ 输入C矩阵的scale	无限制			
			transA:int64	transA/ A矩阵是否转置	仅静态tensor支持转置			
			transB:int64	transB/ B矩阵是否转置				
MatMul	部分支持 目前该支持仅针对双feature输入 未来将支持输入为feature+constant	int8 float16	input_tensor_1 [batch, K，C]:tensor	batch/ 输入的batch	双feature时： batch、H无限制 K支持[8,8192]，对齐要求为8bit数据：16对齐，16bit数据：8对齐 C支持[32,19384]，对齐要求：32对齐 K*C <=65532 C*H <=65532 K*H <=65532 feature+constant时： 若input_tensor_1为feature，则转为batch个feature[K,C,1,1] + weight[H,C,1,1]的conv； 若input_tensor_2为feature，则转为batch个feature[1,C,H,1] + weight[K,C,1,1]的conv； C对齐要求：32对齐 其他约束和conv相同		per-layer/ per-channel	尚不支持
				K/ 输入的K				
			input_tensor_2 [batch, C, H]:tensor	C/ 输入的C				
				H/ 输入的H				
MatMul (4d）	部分支持 目前该支持仅针对双feature输入 未来将支持输入为 feature+constant	int8 float16	input_tensor_1 [batch, channel，K, N]:tensor	batch/ 输入的batch	双feature时： batch无限制 channl、K支持[8,8192]，对齐要求为8bit数据：16对齐，16bit数据：8对齐 N支持[32,19384]，对齐要求：32对齐 K*N <=65532 K*M <=65532 M*N <=65532 feature+constant时： 若input_tensor_1为feature，则转为batch*channel个feature[K,N,1,1] + weight[M,N,1,1]的conv； 若input_tensor_2为feature，则转为batch*channel个feature[1,N,M,1] + weight[K,N,1,1]的conv； N对齐要求：32对齐 其他约束和conv相同		per-layer/ per-channel	尚不支持
				channel/ 输入的channel				
			input_tensor_2 [batch, channel, N, M]:tensor	K/ 输入的K				
				N/ 输入的M				
				M/ 输入的M				

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Expand	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
		int64	shape (batch_o, channel_o, height_o, width_o):tensor	batch_o/ 输出的batch_o	无限制		
				channel_o/ 输出的channel			
				height_o/ 输出的height			
				width_o/ 输出的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
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	尚不支持							尚不支持
Convolution + Tanh	尚不支持	同Convolution						尚不支持
Convolution + Softplus	尚不支持							尚不支持
Convolution + HardSigmoid	尚不支持							尚不支持
Convolution + HardSwish	尚不支持							尚不支持
Convolution + Elu	支持							尚不支持
Convolution + Swish	尚不支持							尚不支持
Convolution + Mish	尚不支持							尚不支持

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
ConvTranspose + Relu	尚不支持							尚不支持
ConvTranspose + Clip	尚不支持							尚不支持
ConvTranspose + PRelu/LeakyRelu	尚不支持							尚不支持
ConvTranspose + Add	尚不支持							尚不支持
ConvTranspose + Mul	尚不支持							尚不支持
ConvTranspose + Sigmoid	尚不支持							尚不支持
ConvTranspose + Tanh	尚不支持	同ConvTranspose						尚不支持
ConvTranspose + Softplus	尚不支持							尚不支持
ConvTranspose + HardSigmoid	尚不支持							尚不支持
ConvTranspose + HardSwish	尚不支持							尚不支持
ConvTranspose + Elu	尚不支持							尚不支持
ConvTranspose + Swish	尚不支持							尚不支持
ConvTranspose + Mish	尚不支持							尚不支持

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Depthwise Convolution + Relu	支持	同Depthwise Convolution						已支持
Depthwise Convolution + Clip	支持							已支持
Depthwise Convolution + PRelu/LeakyRelu	支持							已支持
Depthwise Convolution + Add	尚不支持							已支持
Depthwise Convolution + Mul	尚不支持							尚不支持
Depthwise Convolution + Sigmoid	尚不支持							尚不支持
Depthwise Convolution + Tanh	尚不支持							尚不支持
Depthwise Convolution + Softplus	尚不支持							尚不支持
Depthwise Convolution + HardSigmoid	尚不支持							尚不支持
Depthwise Convolution + HardSwish	尚不支持							尚不支持
Depthwise Convolution + Elu	支持							尚不支持
Depthwise Convolution + 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	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式	多核协同运行支持情况
Add+Relu	支持	同Add						
Mul+Relu	支持	同Mul						
Convolution+ Add+Relu	支持	同Convolution						

注释：
(1) 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)
设计建议：当除数是常量时，建议转换成除数倒数的乘法。乘法在运算效率显著大于除法。

(2) 约束规格中，[a,b]表示支持a-b；{a,b,c}表示支持a,b,c。

第三章 RV1103/1106 NPU Operator List

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

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Global AveragePool	支持	int8	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height	[1,343]（toolkit2支持范围）		
				width/ 输入的width			
GlobalMaxPool	支持	int8	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height	[1,343]（toolkit2支持范围）		
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
AveragePool	支持	int8	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height	[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		
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height方向的kernel大小	无限制，NPU支持[1,8]；其它由CPU支持。		
				kernel_w/ width方向的kernel大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,7]		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,8]		
				stride_w/ width方向的strides大小			

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

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Clip/ReLU6	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	无限制		
				height/ 输入的height			
				width/ 输入的width			
Elu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Gelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Relu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	无限制		
				height/ 输入的height			
				width/ 输入的width			
LeakyRelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	无限制		
				height/ 输入的height			
				width/ 输入的width			
PRelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer/ per-channel
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		
				slope/ PReLU系数	仅支持单个标量或C维度系数		

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
GRU	尚不支持 GRU 扩展以及变体命名为exGRU算子，参数项中指明（extern）的项为exGRU独有的参数项。	float16	input_tensor [sequence，batch，input_size] :tensor	batch/ 输入的batch	1		per-layer
				sequence/ 输入的sequence	无限制，建议8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		
			batch_size:int64（extern）	batch_size/ 指定GRU输入的batchsize	1		
			sequence_size :int64（extern）	sequence_size/ 指定GRU输入的seqsize	无限制，建议4对齐		
			hidden_size:int64（extern）	hidden_size/ GRU单元中的hiddensize	无限制，建议8对齐		
			linear_before_ reset:int64	linear_before_ reset/ LBR变种的选择	1(T) or 0(F)		
			input_layout:string（extern）	input_layout/指定与对应输入shape含义一致的layout	1、snc：指定layout对应的输入shape为[seqs, batches, input_size] 2、(sn)c：指定layout对应的输入shape为[seqs*batches, input_size,1,1] 要求填写指定的layout，同时要求填写该op实际对应的batch_size、sequence_size、hidden_size。		
			output_layout:string（extern）	output_layout/指定与对应输出shape含义一致的layout	1、sbnc：指定layout对应的输出shape为[seqs,directions,batches, hidden_size] 2、(sn)c：指定layout对应的输出shape为[seqs*batches, directions*input_size,1,1] 要求填写指定的layout，同时要求填写该op实际对应的batch_size、sequence_size、hidden_size。 directions>1时仅支持batches=1。		

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

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Pad	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		
				channel/ 输入的channel	无限制		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		
		int64	pads:tensor	[n_begin,c_begin,h_begin,w_begin,n_end,c_end,h_end,w_end]/ 输入各轴上前后插入的pad大小	目前仅支持n_begin,c_begin,n_end,c_end为1 h_begin,w_begin,h_end,w_end无限制		
		float	constant_value:tensor	constant_value/ 填充入pad的值	无限制		
		string	mode:string	mode/pad模式	仅支持constant		
ReduceMean	尚不支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width			
			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	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
ReduceSum	尚不支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width			
			axes:int64[]	axes/ 指定reduce的轴	单轴:无限制，多轴:{2,3}		
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0		
Reshape	部分支持	int8 float16	input_tensor [batch, channel, height, width]:tensor (input_tensor的维度为4维时看作nchw)	batch/ 输入的batch	约束规格： （1） height * width * type_bytes <= 130816； （2） input_tensor非四维时，shape无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
		int64	shape (batch_o, channel_o, height_o, width_o):tensor (输出shape指定维度为4维时看作nchw)	batch_o/ 输出的batch_o	计 算 量 ： alignment=16/type_bytes； 约束规格： （1） height_o * width_o * type_bytes <= 65535； （2） Align(height_o * width_o, alignment) <= 8192； （3） 输出shape非四维时，shape无限制		
				channel_o/ 输出的channel			
				height_o/ 输出的height			
				width_o/ 输出的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Resize	部分支持 目前NPU仅支持宽高方向不超过8倍的整倍数的最邻近插值缩放，其余不支持部分的会Fallback到CPU上实现。	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	支持多batch		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	1.[1,8176] 2.设放大倍数为s（s为正整数），width*s*(s-1)<=8192		
			mode:string	mode/resize采用的模式	仅支持nearest		
			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])		
Reverse Sequence	尚不支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		
			batch_axis:int64	batch_axis/ 指定是否为batch维度	1		
			time_axis:int64	time_axis/ 指定是否为time维度	0		
			sequence_lens:int64[]	sequence_lens/ 指定序列翻转的数量	仅支持channel数		
Sigmoid	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
HardSigmoid	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Swish	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
HardSwish	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Softplus	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Softmax	部分支持	float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel	硬件支持[1,4096] 建议8对齐		
				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			
			axis:int64	axis/ 做softmax的轴	1,即channel方向		

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

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Convolution	支持	int8	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	无限制		
				height/ 输入的height	无限制		
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明		
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制		
				num_input/ 输入的channel	无限制		
				kernel_h/ height方向的kernel大小	[1,31]		
				kernel_w/ width方向的kernel大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,7]		
				stride_w/ width方向的strides大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,15]		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			group:int64	group/ group的大小	无限制		
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]		
				dilations_w/ widtht方向的dilations大小			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Depthwise Convolution	支持	int8	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	无限制		
				height/ 输入的height	无限制		
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明		
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制		
				num_input/ 输入的channel	无限制		
				kernel_h/ height方向的kernel大小	[1,8]		
				kernel_w/ width方向的kernel大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,7]		
				stride_w/ width方向的strides大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,15]		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]		
				dilations_w/ widtht方向的dilations大小			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
ConvTranspose/ Deconvolution	支持	int8	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	无限制		
				height/ 输入的height	无限制		
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明		
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制		
				num_input/ 输入的channel	无限制		
				kernel_h/ height方向的kernel大小	[1,31]		
				kernel_w/ width方向的kernel大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[2,8]		
				stride_w/ width方向的strides大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	支持0-15 设置pad时注意: 不支持 kernel_h * dilations_h - dilations_h - pads_top < 0 不支持 kernel_w * dilations_w - dilations_w - pads_left < 0 不支持 stride_h *(height - 1) - pads_top + 1 < output_h 不支持 stride_w *(width - 1) - pads_left + 1 < output_w		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			group:int64	group/ group的大小	1，当且仅当num_input=num_output时，支持num_output		
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]		
				dilations_w/ widtht方向的dilations大小			

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Expand	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
		int64	shape (batch_o, channel_o, height_o, width_o):tensor	batch_o/ 输出的batch_o	无限制		
				channel_o/ 输出的channel			
				height_o/ 输出的height			
				width_o/ 输出的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
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	尚不支持						
Convolution + Tanh	尚不支持	同Convolution					
Convolution + Softplus	尚不支持						
Convolution + HardSigmoid	尚不支持						
Convolution + HardSwish	尚不支持						
Convolution + Elu	支持						
Convolution + Swish	尚不支持						
Convolution + Mish	尚不支持						

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
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						

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Depthwise Convolution + Relu	支持	同Depthwise Convolution					
Depthwise Convolution + Clip	支持						
Depthwise Convolution + PRelu/LeakyRelu	支持						
Depthwise Convolution + Add	支持						
Depthwise Convolution + Mul	尚不支持						
Depthwise Convolution + Sigmoid	尚不支持						
Depthwise Convolution + Tanh	尚不支持						
Depthwise Convolution + Softplus	尚不支持						
Depthwise Convolution + HardSigmoid	尚不支持						
Depthwise Convolution + HardSwish	尚不支持						
Depthwise Convolution + Elu	支持						
Depthwise Convolution + Swish	尚不支持						
Depthwise Convolution + Mish	尚不支持						

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
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)
设计建议：当除数是常量时，建议转换成除数倒数的乘法。乘法在运算效率显著大于除法。

(2) 约束规格中，[a,b]表示支持a-b；{a,b,c}表示支持a,b,c。

第四章 RK3562 NPU Operator List

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Div	部分支持	float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制	支持两个tensor的广播操作，以ONNX默认排列NCHW做说明，支持以下广播方式： 1、OP(A(N,C,H,W),B(N,C,H,W))，即两个维度相同的tensor进行操作 2、OP(A(N,C,H,W),B(C,1,1))，即C维度做broadcasting 3、OP((N,C,H,W),scalar)，即以单个标量做broadcasting 4、OP(A(N,C,H,W),B(H,W))，即HW维度做broadcasting，目前仅支持FP16类型 说明：A或B都可以作为广播方。 例子见注释（1）	per-layer/ per-channel
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Max	暂不支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1	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。	per-layer/ per-channel
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		
Min	暂不支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1	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。	per-layer/ per-channel
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Global AveragePool	支持	int8	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height	[1,343]（toolkit2支持范围）		
				width/ 输入的width			
GlobalMaxPool	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height	[1,343]（toolkit2支持范围）		
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
AveragePool	支持	int8	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height	[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		
			kernel_shape [kernel_h, kernel_w]:int64[]	kernel_h/ height方向的kernel大小	无限制，NPU支持[1,8]；其它由CPU支持。		
				kernel_w/ width方向的kernel大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,7]		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,8]		
				stride_w/ width方向的strides大小			

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

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Clip/ReLU6	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	无限制		
				height/ 输入的height			
				width/ 输入的width			
Elu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Gelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Relu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	无限制		
				height/ 输入的height			
				width/ 输入的width			
LeakyRelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer
				channel/ 输入的channel	无限制		
				height/ 输入的height			
				width/ 输入的width			
PRelu	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		per-layer/ per-channel
				channel/ 输入的channel	无限制		
				height/ 输入的height			
				width/ 输入的width			
				slope/ PReLU系数	仅支持单个标量或C维度系数		

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
GRU	尚不支持 GRU 扩展以及变体命名为exGRU算子，参数项中指明（extern）的项为exGRU独有的参数项。	float16	input_tensor [sequence，batch，input_size] :tensor	batch/ 输入的batch	1		per-layer
				sequence/ 输入的sequence	无限制，建议8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		
			batch_size:int64（extern）	batch_size/ 指定GRU输入的batchsize	1		
			sequence_size :int64（extern）	sequence_size/ 指定GRU输入的seqsize	无限制，建议4对齐		
			hidden_size:int64（extern）	hidden_size/ GRU单元中的hiddensize	无限制，建议8对齐		
			linear_before_ reset:int64	linear_before_ reset/ LBR变种的选择	1(T) or 0(F)		
			input_layout:string（extern）	input_layout/指定与对应输入shape含义一致的layout	1、snc：指定layout对应的输入shape为[seqs, batches, input_size] 2、(sn)c：指定layout对应的输入shape为[seqs*batches, input_size,1,1] 要求填写指定的layout，同时要求填写该op实际对应的batch_size、sequence_size、hidden_size。		
			output_layout:string（extern）	output_layout/指定与对应输出shape含义一致的layout	1、sbnc：指定layout对应的输出shape为[seqs,directions,batches, hidden_size] 2、(sn)c：指定layout对应的输出shape为[seqs*batches, directions*input_size,1,1] 要求填写指定的layout，同时要求填写该op实际对应的batch_size、sequence_size、hidden_size。 directions>1时仅支持batches=1。		

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

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Pad	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	1		
				channel/ 输入的channel	无限制		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		
		int64	pads:tensor	[n_begin,c_begin,h_begin,w_begin,n_end,c_end,h_end,w_end]/ 输入各轴上前后插入的pad大小	目前仅支持n_begin,c_begin,n_end,c_end为1 h_begin,w_begin,h_end,w_end无限制		
		float	constant_value:tensor	constant_value/ 填充入pad的值	无限制		
		string	mode:string	mode/pad模式	仅支持constant		
ReduceMean	尚不支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width			
			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	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
ReduceSum	尚不支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width			
			axes:int64[]	axes/ 指定reduce的轴	单轴:无限制，多轴:{2,3}		
			keepdims:int64[]	keepdims/ 是否需要保持维度不变	0		
Reshape	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor (input_tensor的维度为4维时看作nchw)	batch/ 输入的batch	约束规格： （1） height * width * type_bytes <= 8192*8192*16； （2） input_tensor非四维时，shape无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
		int64	shape (batch_o, channel_o, height_o, width_o):tensor (输出shape指定维度为4维时看作nchw)	batch_o/ 输出的batch_o	计 算 量 ： alignment=16/type_bytes； 约束规格： （1） height_o * width_o * type_bytes <= INT32_MAX； （2） Align(height_o * width_o, alignment) <= 8192*8192； （3） 输出shape非四维时，shape无限制		
				channel_o/ 输出的channel			
				height_o/ 输出的height			
				width_o/ 输出的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Resize	部分支持 目前NPU仅支持宽高方向不超过8倍的整倍数的最邻近插值缩放，其余不支持部分的会Fallback到CPU上实现。	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	支持多batch		per-layer
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	1.[1,8176] 2.设放大倍数为s（s为正整数），width*s*(s-1)<=8192		
			mode:string	mode/resize采用的模式	支持nearest/linear		
			scales:int64[]	scales/尺寸放大倍数	H scale * W scale <= 64		
			roi:int64[]	roi/进行resize的输入范围	仅支持全局([0,0,0,0,1,1,1,1])		
Reverse Sequence	尚不支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel	[1,8192]		
				height/ 输入的height			
				width/ 输入的width	[1,8176]		
			batch_axis:int64	batch_axis/ 指定是否为batch维度	1		
			time_axis:int64	time_axis/ 指定是否为time维度	0		
			sequence_lens:int64[]	sequence_lens/ 指定序列翻转的数量	仅支持channel数		
Sigmoid	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
HardSigmoid	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Swish	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
HardSwish	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Softplus	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
Softmax	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer
				channel/ 输入的channel	硬件支持[1,8192]		
				height/ 输入的height	axis=1, 无限制 axis=3/-1,width[1, 8192], height无限制		
				width/ 输入的width			
			axis:int64	axis/ 做softmax的轴	1,3，即channel和width方向		

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

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

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Convolution	支持	int8 float16	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	无限制		
				height/ 输入的height	无限制		
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明		
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制		
				num_input/ 输入的channel	无限制		
				kernel_h/ height方向的kernel大小	[1,31]		
				kernel_w/ width方向的kernel大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,7]		
				stride_w/ width方向的strides大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,15]		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			group:int64	group/ group的大小	无限制		
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]		
				dilations_w/ widtht方向的dilations大小			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Depthwise Convolution	支持	int8 float16	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	无限制		
				height/ 输入的height	无限制		
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明		
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制		
				num_input/ 输入的channel	无限制		
				kernel_h/ height方向的kernel大小	[1,8]		
				kernel_w/ width方向的kernel大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[1,7]		
				stride_w/ width方向的strides大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	[0,15]		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]		
				dilations_w/ widtht方向的dilations大小			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
ConvTranspose/ Deconvolution	支持	int8 float16	input_shape [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		per-layer/ per-channel
				channel/ 输入的channel	无限制		
				height/ 输入的height	无限制		
				width/ 输入的width	仅对首层输入width存在限制 详见首层输入说明		
			kernel_shape [num_output, num_input, kernel_h, kernel_w]:int64[]	num_output/ 输出的channel	无限制		
				num_input/ 输入的channel	无限制		
				kernel_h/ height方向的kernel大小	[1,31]		
				kernel_w/ width方向的kernel大小			
			strides [strides_h, strides_w]:int64[]	stride_h/ height方向的strides大小	[2,8]		
				stride_w/ width方向的strides大小			
			pads [pads_top, pads_left, pads_bottom, pads_right]:int64[]	pads_left/ left方向的pads大小	支持0-15 设置pad时注意: 不支持 kernel_h * dilations_h - dilations_h - pads_top < 0 不支持 kernel_w * dilations_w - dilations_w - pads_left < 0 不支持 stride_h *(height - 1) - pads_top + 1 < output_h 不支持 stride_w *(width - 1) - pads_left + 1 < output_w		
				pads_right/ right方向的pads大小			
				pads_top/ top方向的pads大小			
				pads_bottom/ bottom方向的pads大小			
			group:int64	group/ group的大小	1，当且仅当num_input=num_output时，支持num_output		
			dilations [dilations_h, dilations_w]:int64[]	dilations_h/ height方向的dilations大小	[1, 32]		
				dilations_w/ widtht方向的dilations大小			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Gemm	尚不支持 目前由CPU实现	int8 float16	input_tensor_1 [M, K]:tensor	M,K,N/ 输入数据的形状	转为Matmul实现，约束同Matmul		per-layer/ per-channel
			input_tensor_2 [K,N]:tensor				
			alpha:double	alpha/ 矩阵A*B乘法的scale	无限制		
			beta:double	beta/ 输入C矩阵的scale	无限制		
			transA:int64	transA/ A矩阵是否转置	仅静态tensor支持转置		
			transB:int64	transB/ B矩阵是否转置			
MatMul（4d）	部分支持 目前该支持仅针对双feature输入 未来将支持输入为feature+constant	int8 float16	input_tensor_1 [batch, channel，K, N]:tensor	batch/ 输入的batch	双feature时： batch无限制 channl、K支持[1 ,8192] feature+constant时： 若input_tensor_1为feature，则转为batch*channel个feature[K,N,1,1] + weight[M,N,1,1]的conv； 若input_tensor_2为feature，则转为batch*channel个feature[1,N,M,1] + weight[K,N,1,1]的conv； N对齐要求：32对齐 其他约束和conv相同		per-layer/ per-channel
				channel/ 输入的channel			
			input_tensor_2 [batch, channel, N, M]:tensor	K/ 输入的K			
				N/ 输入的M			
				M/ 输入的M			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Expand	支持	int8 float16	input_tensor [batch, channel, height, width]:tensor	batch/ 输入的batch	无限制		
				channel/ 输入的channel			
				height/ 输入的height			
				width/ 输入的width			
		int64	shape (batch_o, channel_o, height_o, width_o):tensor	batch_o/ 输出的batch_o	无限制		
				channel_o/ 输出的channel			
				height_o/ 输出的height			
				width_o/ 输出的width			

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Convolution + Relu	支持						
Convolution + Clip	支持						
Convolution + PRelu/LeakyRelu	支持						
Convolution + Add	支持						
Convolution + Mul	支持						
Convolution + Sigmoid	尚不支持						
Convolution + Tanh	尚不支持	同Convolution					
Convolution + Softplus	尚不支持						
Convolution + HardSigmoid	尚不支持						
Convolution + HardSwish	尚不支持						
Convolution + Elu	尚不支持						
Convolution + Swish	尚不支持						
Convolution + Mish	尚不支持						

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Convolution + Relu	支持						
Convolution + Clip	支持						
Convolution + PRelu/LeakyRelu	支持						
Convolution + Add	支持						
Convolution + Mul	支持						
Convolution + Sigmoid	尚不支持						
Convolution + Tanh	尚不支持	同Convolution					
Convolution + Softplus	尚不支持						
Convolution + HardSigmoid	尚不支持						
Convolution + HardSwish	尚不支持						
Convolution + Elu	尚不支持						
Convolution + Swish	尚不支持						
Convolution + Mish	尚不支持						

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
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						

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
Depthwise Convolution + Relu	支持	同Depthwise Convolution					
Depthwise Convolution + Clip	支持						
Depthwise Convolution + PRelu/LeakyRelu	支持						
Depthwise Convolution + Add	尚不支持						
Depthwise Convolution + Mul	尚不支持						
Depthwise Convolution + Sigmoid	尚不支持						
Depthwise Convolution + Tanh	尚不支持						
Depthwise Convolution + Softplus	尚不支持						
Depthwise Convolution + HardSigmoid	尚不支持						
Depthwise Convolution + HardSwish	尚不支持						
Depthwise Convolution + Elu	尚不支持						
Depthwise Convolution + Swish	尚不支持						
Depthwise Convolution + Mish	尚不支持						

operator	支持情况	输入数据类型	输入	设置项/ 输入参数含义	约束规格	广播支持（维度补齐）	量化支持方式
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)
设计建议：当除数是常量时，建议转换成除数倒数的乘法。乘法在运算效率显著大于除法。

(2) 约束规格中，[a,b]表示支持a-b；{a,b,c}表示支持a,b,c。

第五章 CPU Operator List

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	
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	

第六章 模型输入输出说明

1、模型输入说明

芯片平台	模型首层精度类型	输入维度	首层设置输入数据类型	mean/scale/quant后端实现设备	输入宽（width）对齐要求 单位：元素个数		输入宽（width）大小限制	
					当输入通道（channel）为1,3,4	当输入通道（channel）非1,3,4	当输入通道（channel）为1,3,4（声明见注释9）	当输入通道（channel）非1,3,4
RK3566/3568	int8	4维度	uint8	NPU	8	1	各卷积类型的width/kernel_h/kernel_w需要满足以下两式： 1. width * dilation_kernel_h < 1024*N 2. width <= 4096 其中N必须为1到7的整数，超出范围的卷积不受支持，各卷积类型N的计算方式如下： Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 128) Depthwise Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 4096) ConvTranspose/Deconvolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 128)	无限制
			int8					
			float16	CPU				
			其他类型（*注释8）					
	float16		uint8	CPU	4	1	各卷积类型的width/kernel_h/kernel_w需要满足以下两式： 1. width * dilation_kernel_h < 1024*N 2. width <= 4096 其中N必须为1到7的整数，超出范围的卷积不受支持，各卷积类型N的计算方式如下： Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 128) Depthwise Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 4096) ConvTranspose/Deconvolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 128)	无限制
			int8					
			float16					
			其他类型（*注释8）					
	无限制	非4维	无限制	CPU	1	1	无限制	无限制
RK3588	int8	4维度	uint8	NPU	16	1	各卷积类型的width/kernel_h/kernel_w需要满足以下两式： 1. width * dilation_kernel_h <= 2048 * N 2. width <= 8192 其中N必须为1到7的整数，超出范围的卷积不受支持，各卷积类型N的计算方式如下： Convolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 3) Depthwise Convolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 2048), 3) ConvTranspose/Deconvolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 3)	无限制
			int8					
			float16	CPU				
			其他类型（*注释8）					
	float16		uint8	CPU	8	1	各卷积类型的width/kernel_h/kernel_w需要满足以下两式： 1. width * dilation_kernel_h <= 1024 * N 2. width <= 8192 其中N必须为1到7的整数，超出范围的卷积不受支持，各卷积类型N的计算方式如下： Convolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 3) Depthwise Convolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 2048), 3) ConvTranspose/Deconvolution: N = 12 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 3)	无限制
			int8					
			float16					
			其他类型（*注释8）					
	无限制	非4维	无限制	CPU	1	1	无限制	无限制
RV1103/ RV1106	int8	4维	uint8	NPU	16	1	各卷积类型的width/kernel_h/kernel_w需要满足以下两式： 1. width * dilation_kernel_h <= 2048 * N 2. width <= 4096 其中N必须为1到7的整数，超出范围的卷积不受支持，各卷积类型N的计算方式如下： Convolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2) Depthwise Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 4096) ConvTranspose/Deconvolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2)	无限制
			int8					

RK3562	int8	4维度	uint8	NPU	16	1	各卷积类型的width/kernel_h/kernel_w需要满足以下两式： 1. width * dilation_kernel_h <= 2048 * N 2. width <= 4096 其中N必须为1到7的整数，超出范围的卷积不受支持，各卷积类型N的计算方式如下： Convolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2) Depthwise Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 4096) ConvTranspose/Deconvolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2)	无限制
			int8					
			float16	CPU				
			其他类型（*注释8）					
	float16		uint8	CPU	8	1	各卷积类型的width/kernel_h/kernel_w需要满足以下两式： 1. width * dilation_kernel_h <= 2048 * N 2. width <= 4096 其中N必须为1到7的整数，超出范围的卷积不受支持，各卷积类型N的计算方式如下： Convolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2) Depthwise Convolution: N = 8 - CEIL((dilation_kernel_h * dilation_kernel_w) / 4096) ConvTranspose/Deconvolution: N = 8 - MAX(CEIL((dilation_kernel_h * dilation_kernel_w) / 128), 2)	无限制
			int8					
			float16					
			其他类型（*注释8）					
	无限制	非4维	无限制	CPU	1	1	无限制	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_API》

2、模型输出说明

芯片平台	模型输出精度类型（*注释2）	输出维度	设置输出Layout	Channel对齐要求	H*W对齐要求
RK3566/3568	int8	4维度	NCHW	无	无
			NHWC	8对齐（*注释1）	无
			NC1HWC2	最后一层卷积类算子，16对齐，最后一层非卷积类算子8对齐	H*W要4对齐
			UNDEFINE	无	无
	float16		NCHW	无	无
			NHWC	4对齐（*注释1）	
			NC1HWC2	最后一层卷积类算子，8对齐，最后一层非卷积类算子4对齐	H*W要4对齐
			UNDEFINE	无	无
	无限制	非4维	UNDEFINE	无	无
RK3588	int8	4维度	NCHW	无	无
			NHWC	16对齐（*注释1）	
			NC1HWC2	最后一层卷积类算子，32对齐，最后一层非卷积类算子16对齐	H*W要4对齐
			UNDEFINE	无	无
	float16		NCHW	无	无
			NHWC	8对齐（*注释1）	无
			NC1HWC2	最后一层卷积类算子，16对齐，最后一层非卷积类算子8对齐	H*W要4对齐
			UNDEFINE	无	无
	无限制	非4维	UNDEFINE	无	无

RV1103/ RV1106	int8	4维	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	
RK3562	int8	4维度	NCHW	无	无
			NHWC	无	
			NC1HWC2	最后一层卷积类算子，32对齐，最后一层非卷积类算子16对齐	H*W要4对齐
			UNDEFINE	无	无
	float16		NCHW	无	无
			NHWC	无	无
			NC1HWC2	最后一层卷积类算子，16对齐，最后一层非卷积类算子8对齐	H*W要4对齐
			UNDEFINE	无	无
	无限制	非4维	UNDEFINE	无	无

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；