Neural Network

Layer and Operation Support Guide

Revision: 02 (ref. v1.17-20200623)

Release Date: 2020-07-18

Confidential!
nick@khadas.com
2021-04-12 20:25:41

Amlogic provides this Vivante engineering document for distribution only to its licensee under NDA.

Vivante [**AppNote.VIP.Neural.Network.Layer.and.Operation.Support**-v1.17-20200623.pdf]

This Vivante engineering document contains copyright confidential material disclosed with permission of Vivante Corporation. Vivante has authorized re-distribution of this material restricted to those NDA partners and licensees of *Amlogic* who are engineering *Amlogic products* which includes the discussed Vivante IP.

Application Note:

VIP Neural Network Layer and Operation Support

Confidential
nick@khadas.com
2021-04-12 20:25:41

Document Version 1.17
23 June 2020

Legal Notices

COPYRIGHT INFORMATION

This document contains proprietary information of Vivante Corporation. Vivante Corporation reserves the right to make changes to any products herein at any time without notice. Vivante Corporation does not assume any responsibility or liability arising out of the application or use of any product described herein, except as expressly agreed to in writing by Vivante Corporation; nor does the purchase or use of a product from Vivante Corporation convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual property rights of Vivante Corporation or third parties.

DISCLOSURE/RE-DISTRIBUTION LIMITATIONS

This document contains proprietary information of Vivante Corporation. The information contained in this Vivante Confidential document may be directly re-distributed or may be adapted for re-distribution by Vivante customer SoC vendors who have licensed the related core IP, with the understanding that re-distribution is limited to those customers of the SoC vendor who have active and valid NDA or licenses applicable for the SoC which contains the related Vivante core IP. Otherwise, the information contained herein is not to be used by or disclosed to the third parties without the express written permission of an officer of Vivante Corporation.
(Vivante Distribution **LEVEL B: CONFIDENTIAL – DISTRIBUTION RESTRICTED**).

Upon request Vivante provides this document in Word format for adaptive inclusion in documentation intended for your customers. Vivante requests a pre-release copy of the adaptation for review and approval. In the adapted document please include a reference to the copyright owner, for example: "This chapter contains copyright confidential and proprietary material disclosed with permission of Vivante Corporation. Vivante has authorized re-distribution of this material restricted to those NDA partners and licensees of *[the SoC vendor]* who are engineering *[SoC vendor's product]* which includes the discussed Vivante IP."

Confidential!
nick@khadas.com
2021-04-12 20:25:41

TRADEMARK ACKNOWLEDGMENT

Vivante Corporation and the Vivante Corporation logo design are the trademarks or the registered trademarks of Vivante Corporation. All other brand and product names may be trademarks of their respective companies.

For our current distributors, sales offices, design resource centers, and product information, visit our web page located at http://www.vivantecorp.com and http://www.verisilicon.com.

For technical support, please email vivante-support@verisilicon.com.

Vivante Corporation Confidential, Copyright © 2020 by Vivante Corporation, All rights reserved.

Table of Contents

List of Tables

Confidential!
nick@khadas.com
2021-04-12 20:25:41

1 Overview

This document provides a summary of the neural network layers and operation supported by the VIP ACUITY Tool Kit and compatible Driver Software stack.

2 Deep Learning Frameworks - ACUITY Operation Mapping

The following neural-network operations and corresponding supported API functions are listed in the following table:

- Caffe: some non-standard Caffe layers may be supported according to release schedule.
- TensorFlow API r1.13: all activation ops apply component-wise, and produce a tensor of the same shape as the input tensor.
- TensorFlow (TF) Lite Ops (Schema_v3)
- Onnx (v1.6)
- Darknet: generated from https://pjreddie.com/darknet/
- Keras: generated by Tensorflow 1.13.x.

2.1 Caffe to ACUITY Mapping

Table 1. Caffe-ACUITY Operations Mapping

(Caffe in gray rows; below is ACUITY equivalent in white row)

absval	axpy	batchnorm/bn	convolution	concat
abs	a_times_b_plus_c	batchnormalize	convolution	concat
convolutiondepthwise	dropout	depthwiseconvolution	deconvolution	elu
convolution	dropout	convolution	deconvolution	elu
eltwise	flatten	innerproduct	lrn	l2normalizescale
eltwise	flatten	fullconnect	localresponsenormalization	l2normalizescale
leakyrelu	lstm	normalize	poolwithargmax	premute
leakyrellu	lstm	l2normalize	poolwithargmax	permute
priorbox	prelu	proposal	pooling	roipooling
priorbox	prelu	proposal	pooling	roipooling
reorg	relu	reshape	reverse	swish
reorg	relu	reshape	reverse	swish
slice	scale	shufflechannel	softmax	sigmoid
split	multiply	shuffle	softmax	sigmoid
tanh				
tanh				

Confidential!
nick@kradas.com
2021-04-12 20:25:41

2.2 TensorFlow to ACUITY Mapping

Table 2. TensorFlow - ACUITY Operations Mapping

(TensorFlow in gray rows; below is ACUITY equivalent in white row)

tf.abs	tf.add/tf.nn.bias_add	tf.add_n	tf.argmin	tf.argmax
abs	add	addn/add	argmin	argmax
tf.batch_to_space_nd	tf.nn.batch_normalization	tf.nn.fused_batchnorm	tf.clip_by_value	tf.concat
batch2space	batchnormalize/ instancenormalize/layernormalize/batchnorm_single	batchnormalize	clipbyvalue	concat
tf.nn.conv1d	tf.nn.conv2d/tf.nn.depthwise_conv2d	tf.nn.conv3d	tf.image.crop_and_resize	tf.nn.conv2d_transposed
conv1d	convolution	conv3d	cropsandresize	deconvolution
tf.depth_to_space	tf.equal	tf.exp	tf.nn.elu	tf.nn.embedding_lookup
depth2space	equal	exp	elu	embedding_lookup
tf.maximum	tf.floor	tf.matmul	tf.floordiv	tf.gather_nd
eltwise(MAX)	floor	fullconnect	floor_div	gathernd
tf.gather/ tf.nn.embedding_lookup	tf.nn.rnn_cell_GRUCell tf.nn.dynamic_rnn	tf.nn.rnn_cell_GRUCell	tf.greater	tf.greater_equal
gather	gru	gru_cell	greater	greater_equal
tf.image.resize_bilinear/ tf.image.resize_nearest_neighbor	tf.contrib.layers.instance_norm / tf.nn.fused_batch_norm	tf.nn.local_response_normalization	tf.nn.l2_normalize	tf.nn.rnn_cell_LSTMCell tf.nn_dynamic_rnn
image_resize	instancenormalize	localresponsenormalization_tf	l2normalize	lstm
tf.rnn_cell.LSTMCell	tf.less	tf.less_equal	tf.logical_or	tf.logical_add
lstm_unit	less	less_equal	logical_or	logical_and
tf.nn.leaky_relu	tf.multiply	tf.nn.moments	tf.minimum	tf.matmul/ tf.batch_matmul
leakyrelu	multiply	moments	minimum	matmul
tf.not_equal	tf.negative	tf.pad	tf.transpose	tf.nn.avg_pool/tf.nn.max_pool/ tf.reduce_mean
not_equal	neg	pad	permute	pooling
tf.nn.max_pool_with_argmax	tf.pow	tf.reduce_mean	tf.reduce_sum	tf.reverse/tf.reverse_sequence
poolwithargmax	pow	reducemean	reducesum	reverse
tf.nn.relu	tf.nn.relu6	tf.rsqrt	tf.realdiv	tf.reshape/ tf.expand_dims/ tf.squeeze
relu	relun	rsqrt	real_div	reshape
tf.strided_slice	tf.sqrt	tf.square	tf.subtract	tf.scatter_nd
stridedslice	sqrt	square	subtract	scatternd
tf.stack	tf.nn.sigmoid	tf.signal.frame	tf.slice	tf.nn.softmax
stack	sigmoid	signalframe	slice	softmax
tf.space_to_batch_nd	tf.space_to_depth	tf.split	tf.nn.swish	tf.tile
space2batch	space2depth	split	swish	tile
tf.nn.tanh	tf.unstack	tf.where/tf.select		
tanh	unstack	where		

Confidential!
nick@khadas.com
2021-04-12 20:25:41

2.3 TFLite to ACUITY Mapping

Table 3. TFLite - ACUITY Operations Mapping

(TPLite in gray rows; below is ACUITY equivalent in white row)

ADD	ADD_N	AVERAGE_POOL_2D/MAX_POOL_2D	ABS	BATCH_TO_SPACE_ND
add	addn	pooling	abs	batch2space
CAST	CONV_2D/ DEPTHWISE_CONV_2D	CONCATENATION	DEQUANTIZE	DIV
cast	convolution	concat	dequantize	divide
EMBEDDING_LOOKUP	EXP	EQUAL	EXPAND_DIMS	FLOOR_DIV
embedding_lookup	exp	equal	expanddims	floor_div
FLOOR	FULLY_CONNECTED	GATHER_ND	GATHER	GREATER
floor	fullconnet/ fullconnet_op	gathernd	gather	greater
GREATER_EQUAL	HARD_SWISH	LOGICAL_OR	LOCAL_RESPONSE_NORMALIZATION	LOGISTIC
greater_equal	hard_swish	logical_or	localresponsenormalization	sigmoid
LSTM	L2_NORMALIZATION	L2_POOL_2D	LESS_EQUAL	LOGICAL_AND
lstmunit	l2normalize	l2pooling	less_equal	logical_and
LOG_SOFTMAX	LESS	LEAKY_RELU	MAXIMUM	MUL
log_softmax	less	leakyrelu	maximum	multiply
MEAN	NOT_EQUAL	NEG	POW	PACK
reducemean	not_equal	neg	pow	stack
PAD	PRELU	REDUCE_MIN	RSQRT	REDUCE_MAX
pad	prelu	reducemin	rsqrt	reducemax
RELU	RELU1/RELU_N1_TO_1/ RELU6	RESIZE_BILINEAR/RESIZE_NEAREST_NEIGHBOR	SQUEEZE/RESHAPE	SPLIT
relu	relun	image_resize	reshape	split
SOFTMAX	SVDF	SQUARE	WHERE	SUB
softmax	svdf	square	where	subtract
SLICE	SPACE_TO_BATCH_ND	STRIDED_SLICE	SPACE_TO_DEPTH	TRANSPOSE_CONV
slice	space2batch	stridedslice	space2depth	deconvolution
TRANSPOSE	TILE	TANH	UNPACK	UNIDIRECTIONAL_SEQUEENCE_LSTM
permute	tile	tanh	unstack	lstm

Confidential!
nick@khadas.com
2021-04-12 20:25:41

2.4 ONNX to ACUITY Mapping

Table 4. ONNX - ACUITY Operations Mapping

(ONNX in gray rows; below is ACUITY equivalent in white row)

ArgMin	ArgMax	Add	Abs	And
argmin	argmax	add	abs	logical_and
BatchNormalization	Clip	Cast	Concat	ConvTranspose
batchnormalize	clipbyvalue	cast	concat	deconvolution
Conv	Div	Dropout	DepthToSpace	Equal
conv1d/group_conv1d/depthwise_conv1d/convolution/conv2d_op/depthwise_conv2d_op	divide	dropout	depth2space	equal
Exp	Elu	Floor	InstanceNormalization	Gemm
exp	elu	floor	instancenormalize	matmul/ fullconnect
Gather	Greater	GatherND	Logsoftmax	LRN
gather	greater	gathernd	log_softmax	localresponsenormalization
Log	LeakyRelu	Less	MatMul	Max
log	leakyrelu	less	matmul/ fullconnect	eltwise(MAX)
MaxPool/AveragePool/GlobalAveragePool/GlobalMaxPool	Mul	Neg	Or	Prelu
pooling	multiply	neg	logical_or	prelu
Pad	POW	Relu	Reshape/Squeeze/Unsqueeze/Flatten	ReduceSum
pad	pow	relu	reshape	reducesum
ReduceMean	ReverseSequence	ReduceMax	ReduceMin	Sum
reducemean	reverse_sequence	reducemax	reducemin	eltwise(SUM)
SpaceToDepth	Sqrt	Split	Slice	Squeeze
space2depth	sqrt	split	slice	squeeze
Softmax	Sub	Sigmoid	Tile	Transpose
softmax	subtract	sigmoid	tile	permute
Tanh	Upsample	Where		
tanh	image_resize	where		

Confidential!
nick@khadas.com
2021-04-12 20:25:41

2.5 Darknet to ACUITY Mapping

Table 5. Darknet-ACUITY Operations Mapping

(Darknet in gray rows; below is ACUITY equivalent in white row)

avgpool	batch_normalize	connected	convolutional	depthwise_convolutional
pooling	batchnormalize	fullconnect	convolution	convolution
leaky	logistic	maxpool	region	reorg
leakyrelu	logistic	pooling	region	reorg
relu	route	softmax	shortcut	scale_channels
relu	concat	softmax	add/split+add/pad+add	multiply
swish	upsample	yolo		
swish	upsampling	yolo		

2.6 Keras to ACUITY Mapping

Table 6. Keras-ACUITY Operations Mapping

(Keras in gray rows; below is ACUITY equivalent in white row)

Dense	Flatten/Reshape	LSTM/SimpleRNN	Embedding	BatchNormalization/ BatchNormalizationV1
fullconnect	reshape	lstm_keras	embedding_lookup	batchnormalize
Conv2D	Activation(sigmoid)	Activation(softmax)	Activation(tanh)	Activation(relu)
convolution	sigmoid	softmax	tanh	relu
ZeroPadding2D	MaxPooling2D/ AveragePooling2D/ GlobalAveragePooling2D/ GlobalMaxPooling2D	RELU	Softmax	LeakyRelu
pad	pooling	relu_keras	softmax	leakyrelu
PRelu	ThresholdedReLU	Conv1D	Conv2DTranspose	DepthwiseConv2D
prelu	relun	conv1d	deconvolution	depthwise_convolution
SeparableConv2D	UpSampling2D	Dropout	Subtract	Multiply
depthwise_convolution	image_resize	dropout	subtract	multiply
Concatenate	Cropping2D	RNN	Add	GRU
concat	slice	keras_rnn_lstm	add	gru

Confidential!
nick@khadas.com
2021-04-12 20:25:41

3 ACUITY to OVXLIB Operation Mapping

The following table lists ACUITY operations and their corresponding OVXLIB APIs.

Table 7. ACUITY - OVXLIB Operation Mapping

(ACUITY in white rows; below is OVXLIB equivalent in gray rows)

elu	floor	leakyrelu	prelu	relu
ELU	FLOOR	LEAKY_RELU	PRELU	RELU
relun	relu_keras	swish	hard_swish	rsqrt
RELUN/CLIP	RELU_KERAS	SWISH	HSWISH	RSQRT
sigmoid	softmax	softrelu	sqrt	tanh
SIGMOID	SOFTMAX	SOFTRELU	SQRT	TANH
convolutionrelu	convolutionrelupool	fullconnectrelu	abs	add
CONV_RELU	CONV_RELU_POOL	FCL_RELU	ABS	ADD
addn	clipbyvalue	divide	real_div	equal
ADDN	CLIP	DIVIDE	DIVIDE	RELATIONAL_OPS_EQUAL
exp	log	floor_div	greater	greater_equal
EXP	LOG	FLOORDIV	RELATIONAL_OPS_NOT_GREAT	RELATIONAL_OPS_NOT_GREAT_EQUAL
less	less_equal	logical_and	logical_or	minimum
RELATIONAL_OPS_LESS	RELATIONAL_OPS_LESS_EQUAL	LOGICAL_AND	LOGICAL_OR	MINIMUM
maximum	multiply	neg	not_equal	pow
MAXIMUM	MULTIPLY	NEG	RELATIONAL_OPS_NOT_EQUAL	POW
square	subtract	where	max	a_times_b_plus_c
SQUARE	SUBTRACT	WHERE	ELTWISEMAX	A_TIMES_B_PLUS_C
upsampling	downsample	resizebilinear_image	resizenearest_image	image_resize
RESIZE	RESIZE	RESIZE	RESIZE	RESIZE
upsample	fullconnect	matmul	batchnormalize	instancenormlize
UPSAMPLE	FCL	MATRIXMUL	BATCH_NORM	INSTANCE_NORM
l2normalize	layernormalize	localresponsenormalization	localresponsenormalization_tf	l2normalizescale
L2_NORMALIZE	LAYER_NORM	LRN	LRN2	L2NORMALIZESCALE
fullconnectaxis	fullconnectreluaxis	batch2space	concat	crop
FCL2	FCL2	BATCH2SPACE	CONCAT	CROP
depth2space	gather	gathernd	pad	permute
DEPTH2SPACE	GATHER	GATHER_ND	PAD	PERMUTE
reducemean	reducesum	reducemax	reducemin	reorg
REDUCE_MEAN	REDUCE_SUM	REDUCE_MAX	REDUCE_MIN	REORG
reverse	slice	space2batch	space2depth	split
REVERSE	SLICE	SPACE2BATCH	SPACE2DEPTH	SPLIT
squeeze	reshape	stack	stridedslice	stack_concat
RESHAPE	RESHAPE	STACK	STRIDED_SLICE	TENSORSTACKCONCAT
unstack	gru	gru_cell	lstm	lstm
UNSTACK	GRU_OVXLIB	GRUCELL_OVXLIB	LSTM	LSTM_OVXLIB
lstmunit	convolution	conv2d_op	depthwise_conv2d_op	depthwise_convolution
LSTMUNIT_OVXLIB	CONV2D	CONV2D	CONV2D	CONV2D
deconvolution	depthwise_conv1d	group_conv1d	conv1d	log_softmax
DECONVOLUTION	DEPTHWISE_CONV1D	CONV1D	CONV1D	LOG_SOFTMAX
pooling	l2pooling	poolwithargmax	argmax	argmin
POOL	POOL	POOLWITHARGMAX	ARGMAX	ARGMIN

Confidential!
nick@khadas.com
2021-04-12 20:25:41

dtype_converter	dequantize	quantize	cast	proposal
DATACONVERT	DATACONVERT	DATACONVERT	DATACONVERT	PROPOSAL
roipooling	shuffle	variable	dropout	signalframe
ROI_POOL	SHUFFLECHANNEL	VARIABLE	DROPOUT	SIGNAL_FRAME
svdf_ext	concatshift	spatialtransformer	embedding_lookup	tile
SVDF	CONCATSHIFT	SPATIAL_TRANSFORMER	EMBEDDING_LOOKUP	TILE
moments	batchnorm_single	fullconnect_op	mish	hard_sigmoid
MOMENTS	BATCHNORM_SINGLE	FCL2	MISH	HARD_SIGMOID

Confidential!
nick@khadas.com
2021-04-12 20:25:41

4 VIP Neural Network Layer and Operation Support Overview

The neural-network layers and operations are supported by the following VIP ACUITY Tool Kit and compatible Driver Software stack, where the ACUITY Tool Kit can convert networks containing these layers and generate corresponding Driver Software API calls for inferencing on the VIP hardware.

- ACUITY Tool Kit 5.14.0 or later.
- OVXLIB 1.1.24 or later, High level wrapper library with additional NN APIs for OpenVX, `vx_kernel()` provides for customized kernel implementations.
- Vivante GALVIP Unified Driver 6.4.x version 6.4.3 or later
- Support level as of software releases versions contemporary with this App Note.

Compatible VIP hardware is required where the computing engine provides NN and PPU support. Some operations are hardware independent in the Vivante implementation.

- NN – **N**eural-**N**etwork Engine
- PPU – **P**arallel **P**rocessing **U**nit
- TP – **T**ensor **P**rocessor
- CPU – **C**entral **P**rocessor Unit

This support list is subject to change with routine software updates. Unless specified otherwise, all listed layers are supported for V7x, v80x and V81x VIP and NPU hardware.

Table 8. VIP Neural Network Engine Operation Support

OVXLIB Operation Name	NN	PPU	TP
ELU		✔	
FLOOR		✔	
LEAKY_RELU	✔(8.2)		✔
PRELU	✔(8.2)	✔	✔
RELU	✔(8.x)		✔
RELUN			✔
RELU_KERAS		✔	✔
SWISH			✔
HSWISH			✔
RSQRT		✔	
SIGMOID			✔
SOFTMAX		✔	
SOFTRELU		✔	
SQRT		✔	
TANH			✔
CONV_RELU	✔		
CONV_RELU_POOL	✔		
FCL_RELU	✔		
ABS			✔
ADD	✔		

Confidential!
nick@khadas.com
2021-04-12 20:25:41

OVXLIB Operation Name	NN	PPU	TP
ADDN	✔		
CLIP		✔	
DIVIDE		✔	
RELATIONAL_OPS_EQUAL		✔	
EXP		✔	
LOG		✔	
FLOORDIV		✔	
RELATIONAL_OPS_NOT_GREAT		✔	
RELATIONAL_OPS_NOT_GREAT_EQUAL		✔	
RELATIONAL_OPS_LESS		✔	
RELATIONAL_OPS_LESS_EQUAL		✔	
LOGICAL_AND		✔	
LOGICAL_OR		✔	
MINIMUM		✔	
MAXIMUM		✔	
MULTIPLY		✔	
NEG		✔	
RELATIONAL_OPS_NOT_EQUAL		✔	
POW		✔	
SQURAE		✔	
SUBTRACT		✔	
WHERE		✔	
ELTWISEMAX		✔	
A_TIMES_B_PLUS_C		✔	
RESIZE		✔	
UPSAMPLE		✔	
PRE_PROCESS		✔	
FCL	✔		
MATRIXMUL		✔	
BATCH_NORM		✔	
INSTANCE_NORM		✔	
L2_NORMALIZE		✔	
LAYER_NORM		✔	
LRN			✔
LRN2			✔
L2NORMALIZESCALE		✔	
FCL2			✔
BATCH2SPACE			✔
CONCAT			✔

Confidential!
nick@khadas.com
2021-04-12 20:25:41

OVXLIB Operation Name	NN	PPU	TP
CROP			✔
DEPTH2SPACE			✔
GATHER		✔	
GATHER_ND		✔	
PAD			✔
PERMUTE			✔
REDUCE_MEAN		✔	
REDUCE_SUM		✔	
REDUCE_MAX		✔	
REDUCE_MIN		✔	
REORG			✔
REVERSE			✔
SLICE			✔
SPACE2BATCH			✔
SPACE2DEPTH			✔
SPLIT			
(SQUEEZE)RESHAPE			✔
STACK			✔
STRIDED_SLICE			✔
TENSORSTACKCONCAT		✔	
UNSTACK			✔
GRU_OVXLIB	✔	✔	✔
GRUCELL_OVXLIB	✔	✔	✔
LSTM	✔	✔	✔
LSTM_OVXLIB	✔	✔	✔
LSTMUNIT_OVXLIB		✔	✔
CONV2D	✔		
DECONVOLUTION	✔		
DEPTHWISE_CONV1D		✔	
(group_conv1d) CONV1D	✔		
LOG_SOFTMAX		✔	
(l2_pool) POOL		✔	
POOLWITHARGMAX		✔	
ARGMAX		✔	
ARGMIN		✔	
DATACONVERT			✔
PROPOSAL		✔+CPU	
ROI_POOL			✔
SHUFFLECHANNEL		✔	

confidential!
nick@khadas.com
2021-04-12 20:25:41

OVXLIB Operation Name	NN	PPU	TP
VARIABLE			✔
DROPOUT		✔	
SIGNAL_FRAME		✔	
SVDF			✔
CONCATSHIFT		✔	
SPATIAL_TRANSFORMER		✔	
EMBEDDING_LOOKUP		✔	
TILE		✔	
MOMENTS		✔	
BATCHNORM_SINGLE		✔	
HARD_SIGMOID		✔	
MISH		✔	

Confidential!
nick@khadas.com
2021-04-12 20:25:41

Document Revision History

Version	Date	Compatible HW rev	Notes
1.17	2020-06-23	Vivante SW releases from June 2020	Update Section 2 tables. Table 7: Add operations: fullconnect_op, mish, hard_sigmoid. Table 8: update swish, h_swish support engine, add hard_sigmoid, mish.
1.16	2020-06-12	Vivante SW releases from May 2020	Update Section 4 OVXLIB bullet text.
1.15	2020-05-26	Vivante SW releases from May 2020	Update Section 2. Table 7, 8: Add operations: moments, batchnorm_single
1.14	2020-04-28	Vivante SW releases from April 2020	Add Section 2. Add Section 3. Table 8: Add and enrich ACUITY ops / OVXLIB columns
1.13	2020-03-23	Vivante SW releases from Mar 2020	Tables 1 and 2: Add operations: hard_swish, depthwise_conv1d, group_conv1d and log_softmax.
1.12	2020-03-16	Vivante SW releases from Dec 2019	Legal Notices: distribution level changed from C to B. Update NN support levels for relu, prelu and leaky_relu.
1.11	2020-02-25	Vivante SW release of Dec 2019	Legal Notices: distribution level changed from D to C.
1.10	2019-12-24	Vivante SW release of Dec 2019	Section 1: Update compatible sw versions. Table 1 and Table 2: Add Ops: swish, log, maximum. Table 2: add ONNX column.
1.09	2019-12-03	Vivante SW release of Nov 2019	Section 1: Update compatible sw versions. Table 1: Update Operation Support Engine: gather, add_n, matrix_mul Table 2: Update gather openvx_api
1.08	2019-11-25	Vivante SW release of Nov 2019	Table 1: Update Operation Support Engine. Changed: add, equal, prelu, floor_div, logical_and, where, stack, unstack, argmin. Removed CPU column
1.07	2019-10-28	Vivante SW release of late Oct 2019	Section 1: Update compatible sw versions. Sections 2 and 3, Tables 1 and 2: Added ops: gru_cell, gru, reverse_squeeze and argmin. Miscellaneous refinements, including alphabetic sorting
1.06	2019-07-26	Vivante SW as of v6.4.0.RC2 release.	Table1 and Table 2: Added ops: equal, minimum, where, real_div, add_n, exp, square, negative, abs, clip_by_value, unstack, gather
1.05	2019-05-22	Vivante SW as of v6.3.3.4 release.	Section 2: Added version numbers Section 3: revised Tensorflow API link, added Caffe layer link Table 1: added CPU column for each layer Table1 and Table 2: Added ops: logical_and, resize_nearestneighbor, expand_dims, reduce_sum, stack
1.04	2019-01-08	Vivante SW	Changed from Memo to AppNote.

Confidential!
nick@khadas.com
2021-04-12 20:25:41

		as of v6.3.1 release.	Table 1: added column for TP support. Updated names and support. Added Table 2. Neural Network API Reference Names. Changed page orientation to landscape for Table 2.
1.03	2018-10-04	Vivante SW as of v6.3.1 release.	Table 1 changes: - Changed category Activations to Activation. - Element Wise category: changed floor from PPU to NN, removed max, added mean, removed scale. - Normalization category: changed batch_normalize from NO-OP to PPU, removed I2_normalize_scale. - RNN category: change lstm from PPU to NN. - Changed category Tensor Reshape to Reshape. - Reshape category: changed concat, flatten, reshape, split and squeeze from NO-OP to NN, changed depth_to_space and space_to_depth from PPU to NN, moved reverse from Other category and changed from PPU to NN, added shuffle_channel. - Others category: removed dropout, embedded_lookup, hashtable_lookup, lsh_projection and svdf; change roi_pool from NN to PPU. Removed Notes (1)~(3) at the bottom of Table 1.
1.02	2018-09-20	Vivante SW as of v6.3.1 release.	Updated Table 1. Minor refinements.
1.01	2018-07-03	Vivante SW as of v6.3.1 release.	Updated Table 1.
1.00	2018-03-14	Vivante SW as of v6.3.1 release.	Initial

Confidential!
nick@khadas.com
2021-04-12 20:25:41