OPERATOR SPECIFICATION

XILINX DEEPHI ARCHITECTURE TEAM

Contents

1 Introduction

This document serves as the design specification of OpLib. Chapter 2 addresses the common fields of all operators, while Chapter 3 defines the unique fields of each operator.

1.1 Supported Model Types

The model types currently supported by XModel are *Caffe* and *TensorFlow (r1.9 or above)*.

1.2 Data Types

In plan, XXVM supports eight data types defined by IEEE 754: int8, int16, int32, int64, float16, float32, float64.

2 Common Fields

In this section, the common fields of *op* element in XModel are detailed. All these fields MUST be given when an op element is defined.

2.1 op_name

op_name	
Field Type	string
Description	The name of operator.

2.2 op_type

op_type	
Field Type	string
Description	The type of operator.

3 Operators

(todo: add more description.)

3.1 AvgPool

- **Description**

  Performs the average pooling on the input.

- **Input**

  A 4-D tensor of shape [batch, channels, height, width].

- **Output**

  A tensor with same type as input.

3.1.1 kernel_size

kernel_size	
Field Type	List of 2 integers of shape [kernel_h, kernel_w], default = [0, 0]
Description	The size of the sliding window for each dimension of input tensor

3.1.2 strides

strides	
Field Type	List of 2 integers of shape [stride_h, stride_w], default = [1, 1]
Description	The stride of the sliding window for each dimension of the input tensor.

3.1.3 padding

padding	
Field Type	List of 4 integers of shape [pad_top, pad_bottom, pad_left, pad_right], default = [0, 0, 0, 0]
Description	The number of pixels to add to each side of the input.

3.1.4 pad_mode

pad_mode	
Field Type	string, default = “PADDING”
Description	The mode of using padding. The value MUST be one of [“PADDING”, “SAME”, “VALID”]. ● If pad_mode == “PADDING”, directly use padding field; ● otherwise, create padding data according to correspondent padding algorithms. “VALID” means “no padding”, while “SAME” results in padding the input such that the output has the same length as the original input. Zero padding is introduced to make the shapes match as needed, equally on every side of the input map.

3.1.5 ceil_mode

ceil_mode	
Field Type	Enum, default = FLOOR
Description	The round mode in output size computation, which should be one of {FLOOR, CEIL, ROUND}.

3.1.6 is_global

is_global	
Field Type	bool, default = False
Description	This property is used to specify whether the pooling is global or not. If is_global == True, the pooling is a global average pooling; otherwise, False. Notice: for a global pooling, the following rule MUST be follow: ● strides == [1, 1] ● padding == [0, 0, 0, 0]

3.2 Concat

- **Description**

  Concatenates tensors along one dimension.

- **Input**

  List of *N* tensors to concatenate. Their ranks and types must match, and their sizes must match in all dimensions except axis.

- **Output**

  A tensor with the concatenation of values stacked along the axis dimension. This tensor's shape matches that of input values except in axis where it has the sum of the sizes.

3.2.1 axis

axis	
Field Type	integer, default = 1
Description	A tensor with the concatenation of values stacked along the axis dimension. This tensor's shape matches that of input values except in axis where it has the sum of the sizes. The value of axis MUST be one of [0, 1], where 0 denotes N axis, while 1 is C axis.

3.3 Conv2d

- **Description**

  Computes a 2-D convolution given 4-D input and filter tensors.

- **Input**

  A 4-D tensor of shape [batch, in_channels, in_height, in_width].

- **Output**

  A 4-D tensor with same type as input and shape [batch, in_channels * kernel_h * kernel_w, out_height, out_width], given a kernel of shape [out_channels, in_channels, kernel_h, kernel_w].

3.3.1 kernel_size

kernel_size	
Field Type	List of 2 integers of shape [kernel_h, kernel_w], default = [0, 0]
Description	The size of the sliding window for each dimension of input tensor.

3.3.2 strides

strides	
Field Type	List of 2 integers of shape [stride_h, stride_w], default = [1, 1]
Description	The stride of the sliding window for each dimension of the input tensor.

3.3.3 padding

padding	
Field Type	List of 4 integers in the order of [pad_top, pad_bottom, pad_left, pad_right]
Description	The number of pixels to add to each side of the input.

3.3.4 pad_mode

pad_mode	
Field Type	string, default = “PADDING”
Description	The mode of using padding. The value MUST be one of [“PADDING”, “SAME”, “VALID”]. ● If pad_mode == “PADDING”, directly use padding field; ● otherwise, create padding data according to correspondent padding algorithms. “VALID” means “no padding”, while “SAME” results in padding the input such that the output has the same length as the original input. Zero padding is introduced to make the shapes match as needed, equally on every side of the input map.

3.3.5 dilation

dilation	
Field Type	List of 2 integers of shape [dilation_h, dilation_w], default = [1, 1]
Description	The dilation rate to use for dilated convolution.

3.3.6 bias_term

bias_term	
Field Type	bool, default = False
Description	Specifies whether apply a set of additive biases to the filter outputs. True, if bias tensor is given; otherwise, False.

3.3.7 group

group	
Field Type	integer, default = 1
Description	If group > 1, the connectivity of each filter is restricted to a subset of the input. In the case of group == in_channels, it is depthwise_conv2d.

3.3.8 ceil_mode

ceil_mode	
Field Type	Enum, default = FLOOR
Description	The round mode in output size computation, which should be one of {FLOOR, CEIL, ROUND}.

3.4 DepthwiseConv2d

- **Description**

  Computes a 2-D convolution given 4-D input and filter tensors.

- **Input**

  A 4-D tensor of shape [batch, in_channels, in_height, in_width].

- **Output**

  A 4-D tensor with the same type as input and shape [batch, in_channels * channel_multiplier, out_height, out_width], given a kernel of shape [channel_multiplier, in_channels, kernel_h, kernel_w].

The fields of depthwise_conv2d are same as conv2d. For details, reference section 3.3.

3.5 Conv2dTranspose

- **Description**

  The transpose of conv2d.

- **Input**

  A 4-D tensor of shape [batch, in_channels, in_height, in_width].

- **Output**

  A 4-D tensor with the same type as input.

3.5.1 kernel_size

kernel_size	
Field Type	List of 2 integers of shape [kernel_h, kernel_w], default = [0, 0]
Description	The size of the sliding window for each dimension of input tensor.

3.5.2 output_shape

output_shape	
Field Type	List of 4 integers of shape [N, C, H, W]
Description	The output shape of the conv2d_transpose op.

3.5.3 strides

strides	
Field Type	List of 2 integers of shape [stride_h, stride_w], default = [1, 1]
Description	The stride of the sliding window for each dimension of the input tensor.

3.5.4 padding

padding

Field Type	List of 4 integers in the order of [pad_top, pad_bottom, pad_left, pad_right]
Description	The number of pixels to add to each side of the input.

3.5.5 pad_mode

pad_mode	
Field Type	string, default = “PADDING”
Description	The mode of using padding. The value MUST be one of [“PADDING”, “SAME”, “VALID”]. ● If pad_mode == “PADDING”, directly use padding field; ● otherwise, create padding data according to correspondent padding algorithms. “VALID” means “no padding”, while “SAME” results in padding the input such that the output has the same length as the original input. Zero padding is introduced to make the shapes match as needed, equally on every side of the input map.

3.5.6 bias_term

bias_term	
Field Type	bool, default = False
Description	Specifies whether apply a set of additive biases to the filter outputs. True, if bias tensor is given; otherwise, False.

3.5.7 ceil_mode

ceil_mode	
Field Type	Enum, default = FLOOR
Description	The round mode in output size computation, which should be one of {FLOOR, CEIL, ROUND}.

3.6 InnerProduct

- **Description**

  Converts a 4-D tensor into a output 4-D tensor with height and width both of 1. InnerProduct op has the same semantics as fully-connected layer in Caffe.

- **Input**

  A 4-D tensor of shape [batch, in_channels, in_height, in_width].

- **Output**

  A 4-D tensor with the same type as input and shape [batch, out_channels, 1, 1].

3.6.1 bias_term

bias_term	
Field Type	bool, default=False
Description	Specifies whether apply a set of additive biases to the filter outputs. True, if bias tensor is given; otherwise, False.

3.7 Eltwise

- **Description**

  Adds all input tensors elementwise.

- **Input**

  List of tensors with same shape and type.

- **Output**

  A tensor with the same shape and type as input.

3.7.1 alpha

alpha	
Field Type	List of integers, of which the size is same as input's.
Description	the coefficients used in the formula: $\sum_i alpha_i \times input_i$.

3.8 MaxPool

- **Description**

  Performs the max pooling on the input.

- **Input**

  A 4-D tensor of shape [batch, channels, height, width].

- **Output**

  A tensor with the same type as input.

3.8.1 kernel_size

kernel_size	
Field Type	List of 2 integers of shape [kernel_h, kernel_w], default = [0, 0]
Description	The size of the sliding window for each dimension of input tensor

3.8.2 strides

strides	
Field Type	List of 2 integers of shape [stride_h, stride_w], default = [1, 1]
Description	The stride of the sliding window for each dimension of the input tensor.

3.8.3 padding

padding	
Field Type	List of 4 integers of shape [pad_top, pad_bottom, pad_left, pad_right], default = [0, 0, 0, 0]
Description	The number of pixels to add to each side of the input.

3.8.4 pad_mode

pad_mode	
Field Type	string, default = “PADDING”

Description	The mode of using padding. The value MUST be one of [“PADDING”, “SAME”, “VALID”]. ● If pad_mode == “PADDING”, directly use padding field; ● otherwise, create padding data according to correspondent padding algorithms. “VALID” means “no padding”, while “SAME” results in padding the input such that the output has the same length as the original input. Zero padding is introduced to make the shapes match as needed, equally on every side of the input map.

3.8.5 ceil_mode

ceil_mode	
Field Type	Enum, default = FLOOR
Description	The round mode in output size computation, which should be one of {FLOOR, CEIL, ROUND}.

3.8.6 is_global

is_global	
Field Type	bool, default = False
Description	This property is used to specify whether the pooling is global or not. If is_global == True, the pooling is a global average pooling; otherwise, False. Notice: for a global pooling, the following rule MUST be follow: ● strides == [1, 1] ● padding == [0, 0, 0, 0]

3.9 Mean

- **Description**

  Computes the mean of elements across dimensions of a tensor.

  The op reduces input tensor along the dimensions given in axis. Unless keep_dims is true, the rank of the tensor is reduced by 1 for each entry in axis. If keep_dims is true, the reduced dimensions are retained with length 1.

- **Input**

A 4-D tensor of shape [batch, channels, height, width].

- **Output**

  The reduced tensor with the same type as input.

3.9.1 axis

axis	
Field Type	List of integers, default=[]
Description	The dimensions to reduce.

3.9.2 keep_dims

keep_dims	
Field Type	bool, default=false
Description	The property is used to specify whether the reduced dimension is kept or not. If keep_dims == True, it retains reduced dimensions with length 1.

3.10 Pad

- **Description**

  Pads a tensor according to the specified padding.

- **Input**

  A tensor of shape [batch, in_channels, in_height, in_width].

- **Output**

  The padded tensor with the same type as input.

3.10.1 mode

mode	
Field Type	string, default = ”CONSTANT”
Description	The padding mode. The value of the property MUST be one of [‘CONSTANT’, ‘REFLECT’, ‘SYMMETRIC’].

3.10.2 pad_with

pad_with	
Field Type	List of integers of shape [D, 2] given an input tensor of D dimensions.
Description	The widths of the padding regions applied to the edges of each axis. For a tensor of shape [N,C,H,W], for example, the value of padding has the shape of [[N_before, N_after], [C_before, C_after], [H_before, H_after], [W_before, W_after]], where X_before / X_after means the number of elements to be padded before / after X axis.

3.10.3 constant_values

constant_values	
Field Type	List of integers of shape [D, 2] given an input tensor of D dimensions.
Description	The value used for padding when the mode is "CONSTANT".

3.11 Relu

- **Description**

  Computes rectified linear: $\max(features, 0)$.

- **Input**

  A 4-D tensor of shape [batch, in_channels, in_height, in_width].

- **Output**

  The tensor with the same shape and type as input.

3.12 LeakyRelu

- **Description**

  Compute the Leaky ReLU activation function. The formula for leaky ReLU is shown as below.

$$leaky\_relu(x) = \begin{cases} alpha \times x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

- **Input**

  A tensor representing preactivation values of type [XINT8, XINT16, XINT32, XINT64, XFLOAT16, XFLOAT32, XFLOAT32].

- **Output**

  The tensor with the same shape and type as input.

3.12.1alpha

alpha	
Field Type	float, default = 0.01
Description	The slope of the activation function at x < 0. It is required that alpha > 0.

3.13 PRelu

- **Description**

  Computes the Parametric ReLU (PReLU) activation function channel wise. The formula for PReLU is shown as below.

$$prelu(x) = \begin{cases} alpha_i \times x_i, & x_i < 0 \\ x_i, & x_i \geq 0 \end{cases}, \text{where } i \text{ is channel index.}$$

- **Input**

  A tensor representing pre-activation values.

- **Output**

  The tensor with the same shape and type as input.

3.14 Relu6

- **Description**

  Compute the ReLU6 activation function. The formula for ReLU6 is shown as below.

$$leaky\_relu(x) = \begin{cases} 0, & x \leq 0 \\ x, & 0 \ < x < 6 \\ 6, & x \geq 6 \end{cases}$$

- **Input**

  A tensor representing pre-activation values.

- **Output**

  The tensor with same shape and type as input.

3.15 Reshape

- **Description**

Reshapes a tensor. It should support reshape a tensor of any dimensions.

- **Input**

  A tensor.

- **Output**

  The reshaped tensor with same type as input.

3.15.1 shape

shape	
Field Type	List of integers
Description	The shape of the output tensor.

3.16 Squeeze

- **Description**

  Removes dimensions of size 1 from the shape of a tensor. It should support input of any dimensions.

- **Input**

  A tensor.

- **Output**

  A tensor containing the same data as input, but having one or more dimensions of size 1 removed.

3.16.1 axis

axis	
Field Type	List of integers, default=[]
Description	The dimensions to be squeezed. The dimension index starts at 0. If aixs == [], it removes all size 1 dimensions.

3.17 DeephiTile

- **Description**

  Constructs a tensor by tiling a given tensor.

- **Input**

  A tensor.

- **Output**

  A tensor, in which $dim_{out}[i] = dim_{in}[i] \times multiples[i]$.

3.17.1 multiples

multiples	
Field Type	List of integers
Description	This property is used to specify how many times input tensor replicates. Length MUST be the same as the number of dimensions in input.

3.18Upsample

- **Description**

  Repeats the rows and columns of the data by scale[0] and scale[1] respectively.

- **Input**

  A 4-D tensor of shape [batch, channels, height, width].

- **Output**

  A 4-D tensor of shape [batch, channels, upsampled_height, upsampled_width]. The data type is same as input.

3.18.1 scale

scale	
Field Type	List of 2 integers, default = [1, 1]
Description	The upsampling factors for rows and columns.

3.18.2 mode

mode	
Field Type	string, default = “BILINEAR”

Description	The upsampling algorithm: one of “BILINEAR” or “NEAREST”.

3.19 DeephiDownsample

- **Description**

  Reduced the rows and columns of the data by scale[0] and scale[1] respectively.

- **Input**

  A 4-D tensor of shape [batch, channels, height, width].

- **Output**

  A 4-D tensor of shape [batch, channels, downsampled_height, downsampled_width]. The data type is same as input.

3.19.1 scale

scale	
Field Type	List of 2 integers, default = [1, 1]
Description	The downsampling factors for rows and columns.

3.19.2mode

mode	
Field Type	string, default = “BILINEAR”
Description	The downsampling algorithm: one of “BILINEAR” or “NEAREST”.

3.20 Reorg

- **Description**

  (Derived from YOLO) Reshapes the output tensor so that H and W dimensions of the tensor matches the other tensor downstream, and these two tensor outputs could be concatenated together.

- **Input**

A 4-D tensor of shape [batch, channels, height, width].

- **Output**

  A 4-D tensor of shape [batch, out_channels, out_height, out_width]. The data type is same as input. The mapping relations between input and output dimensions is defined as below:

  - For forward propagation,

    $$[N, C, H, W] \xRightarrow{Reorg} \left[N, C \times s^2, \frac{H}{s}, \frac{W}{s}\right]$$

    where s is a positive integer. $C \times s^2$, $\frac{H}{s}$ and $\frac{W}{s}$ also have to be positive integer.

  - For backward propagation,

    $$[N, C, H, W] \xRightarrow{Reorg_{reversed}} \left[N, \frac{C}{s^2}, H \times s, W \times s\right]$$

    where s is a positive integer. $\frac{C}{s^2}$, $H \times s$ and $W \times s$ also have to be positive integer.

3.20.1 Reversed

mode	
Field Type	bool, default = False
Description	This property is used to specify the computation runs in either forward or backward propagation. True if the computation runs in forward way; otherwise, False.

3.21 Input

- **Description**

  Serves as a placeholder indicating an input.

- **Input**

  No input.

- **Output**

  A tensor.

3.22 Output

- **Description**

  Serves as a placeholder indicating an output.

- **Input**

  A tensor.

- **Output**

  No output.

3.23 Const

- **Description**

  Serves as a placeholder which correlates to a tensor.

- **Input**

  No input.

- **Output**

  A tensor.