@IEEE ICASSP2022
Paper#3116

An Efficient Method for Generic DSP Implementation of Dilated Convolution

Harinarayanan.E.V and Sachin Ghanekar
Cadence Design Systems, Pune, India

Introduction

Dilated convolution has an inherent property of capturing wider context in an image and long-term temporal characteristics in an audio signal. We propose a new scheme that allows efficient/generic implementation of 2D Dilated convolution and stride on typical DSPs where the instruction sets are well tuned for standard 1D and 2D filtering and convolution operations.

In this proposal an existing flexible and efficient standard 2D convolution implementation with stride support forms the basic building block to implement Dilated convolution with stride. Thereby, 2D-dilated convolution is equivalently represented as several smaller convolutions with appropriate matrix slicing and re-ordering.

Dilation & Stride

2D Dilation in literature referred to as *atrous* convolution or convolution with holes. Dilation introduces 'zeros' in the kernel matrix of the convolution. An example of pre/post kernel dilation with a factor 2 in height d_h and width d_w shown below,

K_w, K_h

K_{11}	K_{12}	K_{13}
K_{21}	K_{22}	K_{23}
K_{31}	K_{32}	K_{33}

Kernel Width Post Dilation
$K_{w\_D} = K_w + (K_w - 1) * (d_w - 1)$
Kernel Height Post Dilation
$K_{h\_D} = K_h + (K_h - 1) * (d_h - 1)$

K_{w_D}, K_{h_D}

K_{11}	0	K_{12}	0	K_{13}
0	0	0	0	0
K_{21}	0	K_{22}	0	K_{23}
0	0	0	0	0
K_{31}	0	K_{32}	0	K_{33}

Matrices post dilated convolution are generally strided to reduce the dimensions of the output. Stride simply-put is skipping the matrix values by a factor. Stride and Dilation in mathematical form for 2D matrix is as below:

$$Y(x_3, y_3) = \sum_{0 \le x_2 < K_h} \sum_{0 \le y_2 < K_w} I\begin{pmatrix} x_2 * d_h + x_3 * s_h, \\ y_2 * d_w + y_3 * s_w \end{pmatrix} * F(x_2, y_2)$$

Where, I the input matrix, F kernel matrix and s_w/s_h are the stride factors in width and height dimension, $0 \le y_3 < \lfloor (I_w - K_{w\_D})/s_w \rfloor + 1$ and $0 \le x_3 < \lfloor (I_h - K_{h\_D})/s_h \rfloor + 1$

The input matrices are assumed to be appropriately zero padded, if needed, such that input width and input height are always gr. than equal to kernel dimension i.e., $I_w \ge K_{w\_D}$ and $I_h \ge K_{h\_D}$.

A Novel scheme for Dilation and Stride

Dilated convolution in comparison to non-dilation increases the computational complexity by the order of the product of dilation factors in height and width dimension using *atrous* method. We propose a joint solution for dilation and stride. As a special case, the complexity is immune to the dilation factor when stride is unity.

Dilation Scheme

Implementation of Dilated convolution split into three steps: a) Input Slicing b) Standard convolution c) Output stitching. An existing standard convolution forms the central computation block and input-Slicing/output-stitching are the memory alignment processes. Below an illustration of a 7X5 matrix for dilation factor of 2 in height and width.

Slicing

Slicing large input matrix into d_h X d_w sub-matrices by skipping input elements by dilation factor

$$I_{n_1 n_2 d_h d_w}(x, y) = I(n_1 + d_h * x, n_2 + d_w * y)$$

Convolution

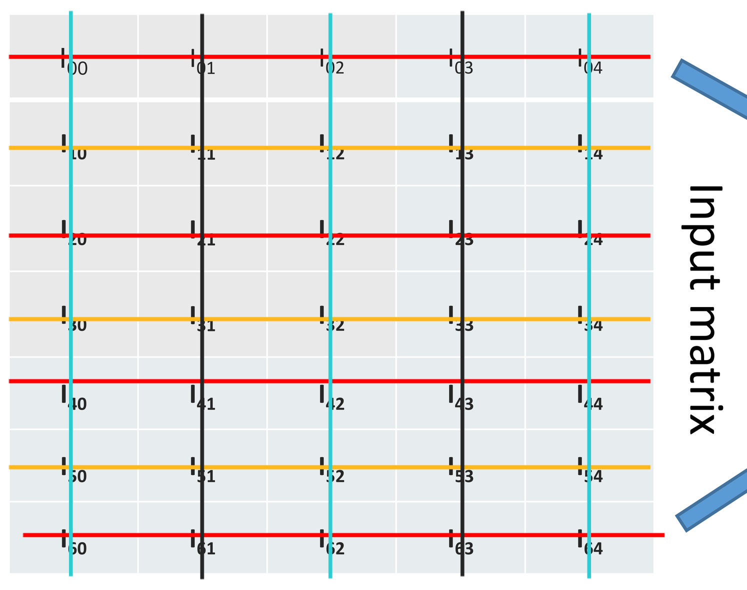
sub-matrices convolved with kernel using std convolution generating output sub-matrices

$$Y_{n_1 n_2 d_h, d_w}(x_{\tilde{n}}, y_{\tilde{n}}) = \sum_{0 \le x_2 < K_h} \sum_{0 \le y_2 < K_w} I_{n_1 n_2 d_h d_w}\begin{pmatrix} x_2 + x_{\tilde{n}}, \\ y_2 + y_{\tilde{n}} \end{pmatrix} * F(x_2, y_2)$$

Stitching

sub-matrix convolved with kernel using std convolution generating output sub-matrices

$$Y(n_1 + x_{\tilde{n}} * d_h, n_2 + y_{\tilde{n}} * d_w) = Y_{n_1 n_2 d_h, d_w}(x_{\tilde{n}}, y_{\tilde{n}})$$



sub-matrix_{00} (Red,Cyan)	sub-matrix_{01} (Red,Black)
sub-matrix_{10} (Yellow,Cyan)	sub-matrix_{11} (Yellow,Black)

Supermatrix contains d_hX d_w sub-matrices. Total no. of elements in the input matrix same as supermatrix.

Standard convolution

Kernel Matrix

output_{00} (Red,Cyan)	output_{01} (Red,Black)
output_{10} (Yellow,Cyan)	output_{11} (Yellow,Black)

Stride Scheme

The skipped values of convolved output by a pre-defined factor is given by stride. The scheme for stride is proposed in the dilation framework as explained above i.e., slicing/convolution/stitching

Problem Statement

From the 'stitching' step the output values of interest after applying stride can be written as,

$$Y(s_h * x, s_w * y) = Y(n_1 + x_{\tilde{n}} * d_h, n_2 + y_{\tilde{n}} * d_w)$$

Where, s_h and s_w are stride in height and width dimension. In other words, matrix values with integer co-ordinates of $< x, y >$ are values of interest as shown below.

$$< x, y > = < \frac{n_1 + x_{\tilde{n}} * d_h}{s_h}, \frac{n_2 + y_{\tilde{n}} * d_w}{s_w} >$$

Alternate Statement

S1: For a given offset pair $< n_1, n_2 >$ find the minimum value of co-ordinates $< x_{\tilde{n}}\ , \ y_{\tilde{n}} >$ say, $< x_{\tilde{n}\_min}, y_{\tilde{n}\_min} >$

S2: From the initial values $< x_{\tilde{n}\_min}, y_{\tilde{n}\_min} >$ for a given $< n_1, n_2 >$ find successive values of $< x_{\tilde{n}}, \ y_{\tilde{n}} >$

Solution

Postulate: Assume S1 is true i.e., for a given $< n_1, \ n_2 >$ there exist a $< x_{\tilde{n}\_min}, \ y_{\tilde{n}\_min} >$. Finding solution for S2 i.e., to find the successive values of $< x_{\tilde{n}}, y_{\tilde{n}} >$ after the initial value $< x_{\tilde{n}\_min}, \ y_{\tilde{n}\_min} >$

Inference: Let the next successive value of $< x_{\tilde{n}}, \ \ y_{\tilde{n}} >$ after $< x_{\tilde{n}\_min}, y_{\tilde{n}\_min} >$ be $< x_{\tilde{n}\_min} + \Delta_x, \ y_{\tilde{n}\_min} + \Delta_y >$. Inserting these values.

$$< x, y > = < \frac{n_1 + (x_{\tilde{n}\_min} * d_h)}{s_h} + \frac{\Delta_x * d_h}{s_h}, \frac{n_2 + (y_{\tilde{n}\_min} * d_w)}{s_w} + \frac{\Delta_y * d_w}{s_w} >$$

The minimum value of Δ_x & Δ_y to contribute an integer value $< x, y >$ is

$$\Delta_x, \Delta_y = \frac{s_h}{GCD(s_h, d_h)}, \frac{s_w}{GCD(s_w, d_w)}$$

Therefore $<\Delta_x, \Delta_y>$ is the periodic pattern for chosen value of dilation and stride. Extending this periodic property, it is sufficient to check the first $<\Delta_x, \Delta_y>$ elements to test the validity of statement S1.

Highlights:

- A sub-matrix $< n_1, \ n_2 >$ not satisfying S1 => no participation in convolution
- Δ_x & Δ_y are the modified stride values for sub-matrix convolution
- Upon re-ordering $\frac{d_h}{GCD(s_h, d_h)}, \frac{d_w}{GCD(s_w, d_w)}$ is the output stride

Re-ordering stride output

$$Y\left(\frac{h_{offset}(n_1) + i_1 * R_h}{s_h}, \frac{w_{offset}(n_2) + i_2 * R_w}{s_w}\right) = Y_{n_1 n_2 d_h d_w s_h s_w}(i_1, i_2)$$

$$n_1 = 0, 1, 2, \ldots, d_h - 1$$
$$n_2 = 0, 1, 2, \ldots, d_w - 1$$
$$h_{offset}(n_1) = n_1 + x_{\tilde{n}\_min}(n_1) * d_h$$
$$w_{offset}(n_2) = n_2 + y_{\tilde{n}\_min}(n_2) * d_w$$
$$R_h = \frac{s_h}{GCD(d_h, s_h)} * d_h \ ; R_w = \frac{s_w}{GCD(d_w, s_w)} * d_w$$

Result

The proposed method of decomposition is compared against atrous method. The method implemented on Cadence's Tensilica HiFi5 processor with NN extension simulator assuming zero memory wait states. Input/kernel data format (N,H,W,C). An improvement of 30X (cycles) can be observed for dilation factor of 16. Scratch memory reduction for the proposed method observed. The implementation is available on Cadence's NN HiFi5 GitHub link .

Dilation Factor	Decomposition Method		Zero Insertion (ZI) Method	
(Stride=2)	Cycles ($X10^6$)	Scratch Memory (KB)	Cycles ($X10^6$)	Scratch Memory (KB)
2	1.94	6.14	4.89	20.20
4	2.01	3.14	10.01	36.33
8	2.14	1.64	26.54	68.58
16	2.32	0.89	69.70	133.08

Computational gain for different Dilation and Stride values published in paper