SMM-Conv: Scalar Matrix Multiplication with Zero Packing for Accelerated Convolution

Amir Ofir and Gil Ben-Artzi

ARIEL UNIVERSITY

CVPR JUNE 19-24 2022 NEW ORLEANS • LOUISIANA

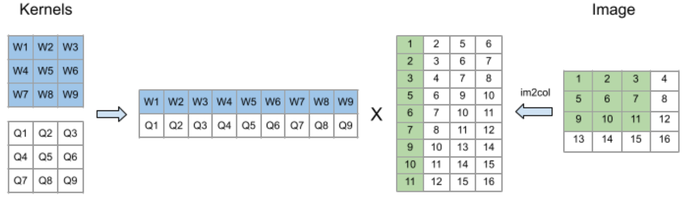
Motivation

- Acceleration of convolutions on CPU-based architectures.
- Reduction of memory overhead.

Contribution

- Formulate convolution as a linear combination of shifted sub-metrices.
- Scalar-matrix multiplication is more efficient than matrix-matrix multiplication.
- Parallel version as well.

Background



Commonly used computation method of general matrix multiplication (GEMM) is optimized for CPU execution while having two major disadvantages:

- Memory overhead and inefficient memory access, caused by packing overlapping image blocks
- Inefficient execution, due to irregular dimensions GEMM doesn't perform as well on convolutional matrices as on classical high performance computing applications.

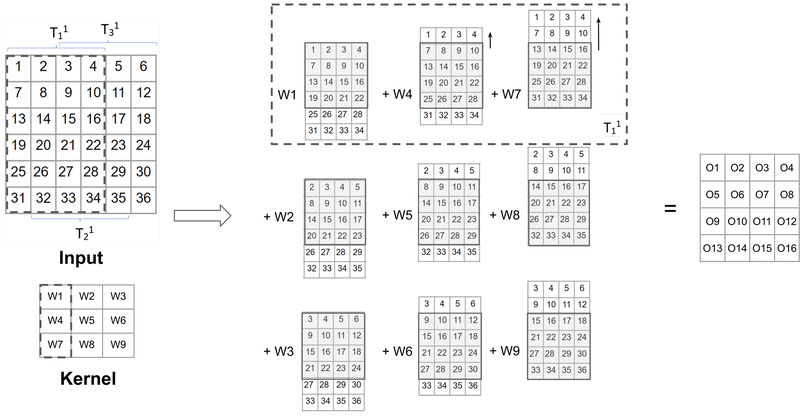
Linear Combination of Sub-Metrices

$$\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix} * \begin{pmatrix}10 & 20 & 30\\ 40 & 50 & 60\\ 70 & 80 & 90\end{pmatrix} = 1 * \begin{pmatrix}10 & 20\\ 40 & 50\end{pmatrix} + 2 * \begin{pmatrix}20 & 30\\ 50 & 60\end{pmatrix} + 3 * \begin{pmatrix}40 & 50\\ 70 & 80\end{pmatrix} + 4 * \begin{pmatrix}50 & 60\\ 80 & 90\end{pmatrix}$$

Implemented as scalar-matrix multiplication

Method

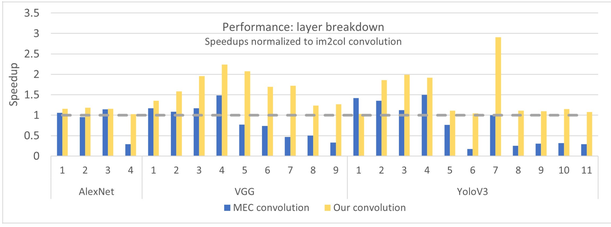
The 2D output of convolution of an input tensor I of size $h \times w$ with kernel of size $k_h \times k_w$ can be considered as summation of $k_h * k_w$ shifted versions of the input tensor I, with corresponding sub-matrices of size $h' \times w'$ multiplied by corresponding coefficient. Therefore, we consecutively extract the sub-matrices $T_j^c, j \in [k_w]$ which consist of all the rows of the I and w' columns, $I[c, 1{:}\,h, j{:}\,j + w' - 1]$ and multiply each sub-matrix of size $h' \times w'$ with the corresponding kernel weight and sum.



Performance

Execution times (in seconds)

Network	Im2col	MEC	Ours	Speedup
AlexNet	0.4608	0.2008	**0.1348**	3.4183
VGG	2.3670	2.8562	**1.3535**	2.1102
YoloV3	0.4478	0.5779	**0.2889**	2.0003



Scalability