Implementation of Kernel Segregated Transpose Convolution Operation

Vijay Srinivas Tida[†], Sai Venkatesh Chilukoti, Md. Imran Hossen, Liqun Shan, Sonya Hsu and Xiali Hei

Introduction

- The Convolution Layer will compress the output feature map.
- The Transpose Convolution (TC) layer enhances the feature map.
- TC layer in deep learning is considered a combination of upsampling layer and convolution operation.
- The upsampling layer is formed by inserting zeros along both rows and columns after each value.
- **Applications:** Image super-resolution, Generative Adversarial Networks (GANs).

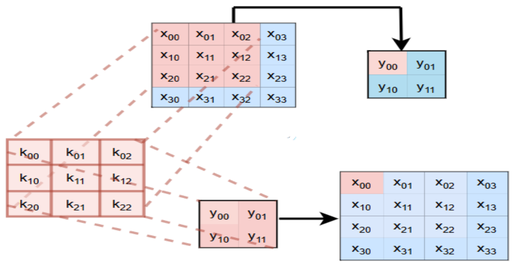

Fig: Convolution vs Transpose Convolutuon

Overview

•These four computation patterns will help to form four sub-kernels.

•Thus, transpose convolution can be treated as four internal convolution operations.

•Red squares represent inactive elements, and green squares represent active elements.

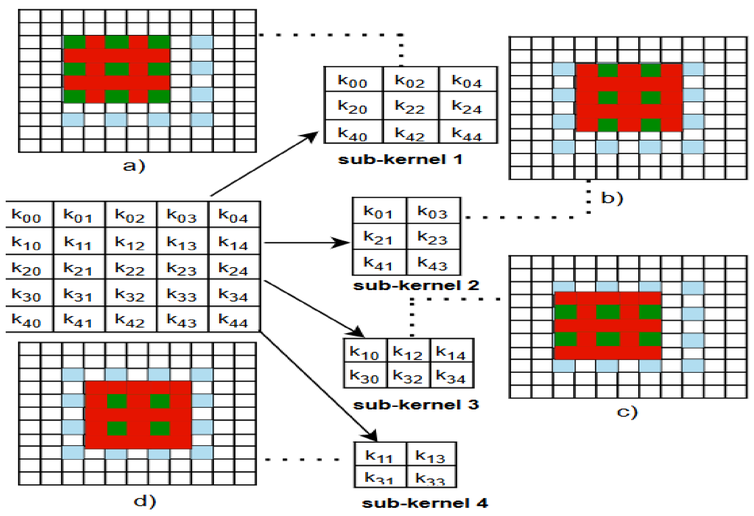

Fig: Computation pattern for Transpose Convolution

Normal vs Segregated TC

- Normal TC operation requires 25 multiplications and 24 additions.
- Proposed Segregated TC operation only requires the utmost 9 multiplications and 8 additions.

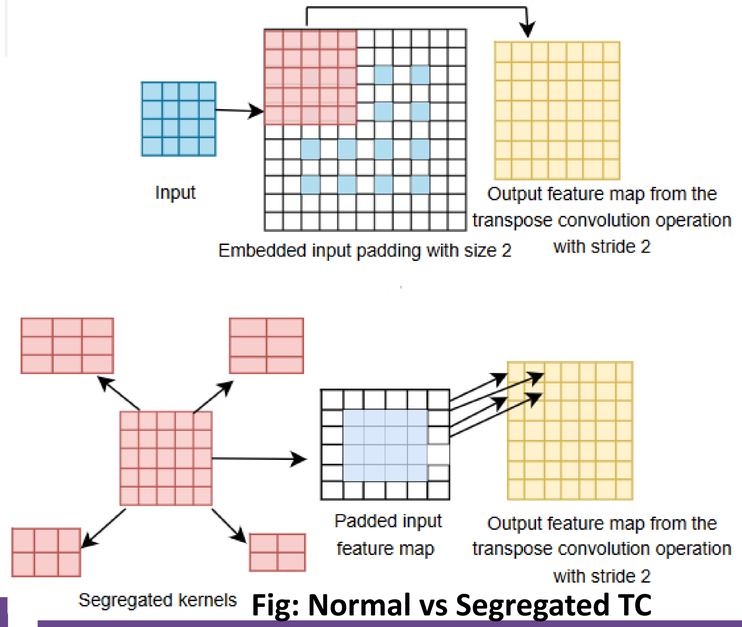

Fig: Normal vs Segregated TC

Results

Table: Power, area and delay using Synopsys DC Compiler

Design	Delay (ns)	Area (cell units)	Power (mW)
45nm technology			
3×3 kernel			
Conventional/ 1 output	1.53	29413.37	19.23
Proposed/ 4 outputs	1.31	29019.63	19.91
4×4 kernel			
Conventional/ 1 output	1.66	54174.12	31.90
Proposed/ 4 outputs	1.35	51217.06	37.57
5×5 kernel			
Conventional/ 1 output	1.77	78509.66	46.48
Proposed/ 4 outputs	1.52	71270.24	56.38
14nm technology			
3×3 kernel			
Conventional/ 1 output	0.49	3105.55	2.93
Proposed/ 4 outputs	0.44	3070.52	2.96
4×4 kernel			
Conventional/ 1 output	0.52	5835.89	5.19
Proposed/ 4 outputs	0.44	5645.68	5.70
5×5 kernel			
Conventional/ 1 output	0.54	8966.66	7.77
Proposed/ 4 outputs	0.49	8549.79	8.31

Discussion and Limitations

- The delay, area and power requirements are noted for 45nm and 14nm technology nodes using Synopsys DC Compiler.
- The implemented design considers the input size of 8 bits and the kernel size of 32 bits.
- The proposed method showed more efficiency in terms of delay, area and power consumption.
- The power consumption for the proposed method is for generating four pixels.
- The average power consumption is very low when compared to the original method for writing a single element in the output feature map.
- The proposed method reduces the computation load up to nearly 3.8× compared to the original implementation.
- The proposed method helps to scale the existing deep learning models having TC to implement on handheld devices.

Limitations

- The proposed method produces four pixels at a time, which results in computing the unwanted elements if the output feature is of odd dimensions.
- Thus, the extra elements needed to be avoided for future computations.

Future Works

- Implement the proposed optimized transpose convolution operation on different Field Programmable Field Arrays (FPGAs) from Intel and AMD companies.
- Also, design the simple neural network which uses transpose convolution layer on FPGAs and analyze the performance.

References

- Tida, V. S., Chilukoti, S. V., Hsu, S. H. Y., & Hei, X. (2023). Kernel-segregated transpose convolution operation. In T. X. Bui (Ed.), 56th hawaii international conference on system sciences, HICSS 2023, maui, hawaii, USA, january 3-6, 2023 (pp. 6934–6943). ScholarSpace.https://hdl.handle.net/10125/103035
- Yazdanbakhsh, A., Samadi, K., Kim, N. S., & Esmaeilzadeh, H. (2018). Ganax: A unified mimd-simd acceleration for generative adversarial networks. 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA), 650–661.