WUST

学习总结

Mode by : 董勇

目录
CONTENT

01
论文

02
视频

03
实验

PART 01

题目：An Efficient Hardware Accelerator for Structured Sparse Convolutional Neural Networks on FPGAs

[THIS MANUSCIRPT IS FOR IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRA TION (VLSI) SYSTEMS]

01 Paper

CONV layer

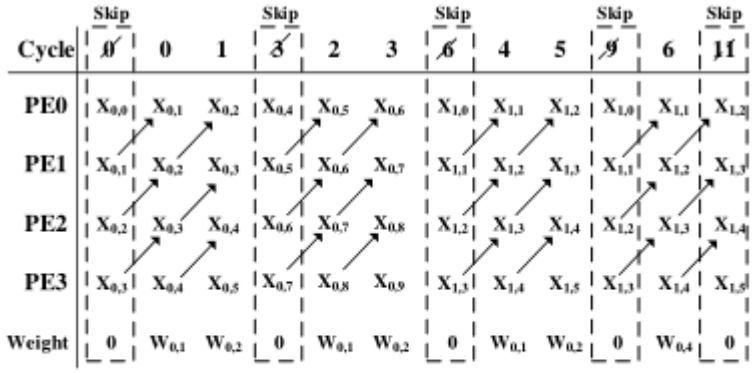


Fig. 3. Sparse wise dataflow for CONV layers. When weights equal to zero, the cycles of processing MACs will be skipped by controlling the upper bound of corresponding loop illustrated in Algorithm 2

FC layer

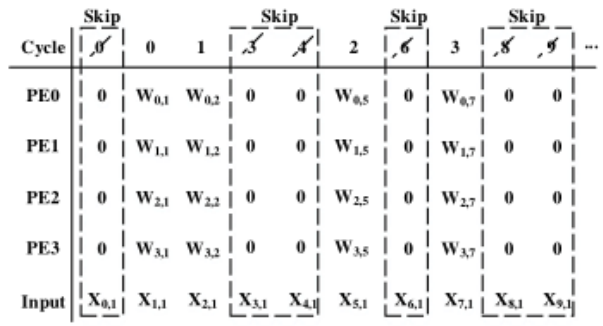


Fig. 4. Execution pattern of FC layers. Similar to CONV layers, the cycles of processing MACs will be skipped when weights equal to zero.

01 Paper

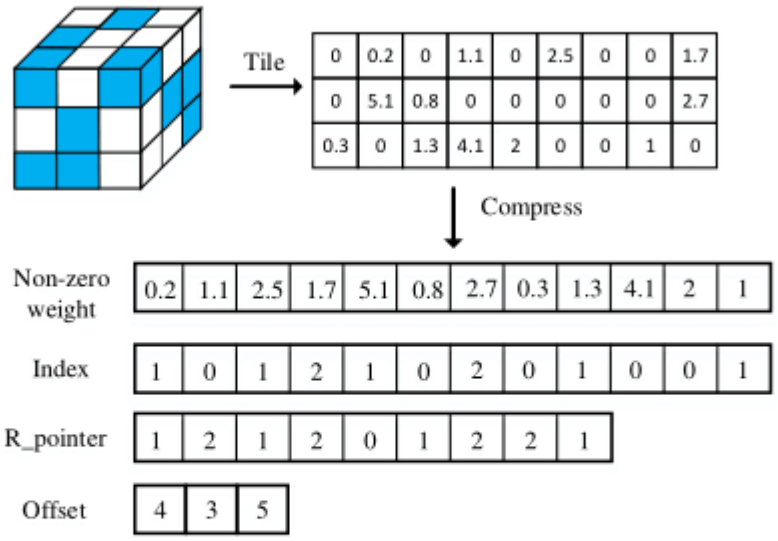


Fig. 7. Index representation of weights in CONV layers. The Index represents the number of pruned weights between two nonzero weights. The r_pointer represents the number of remaining weights in each row. The Offset shows the remaining weights in each channel of one kernel.

01 Paper

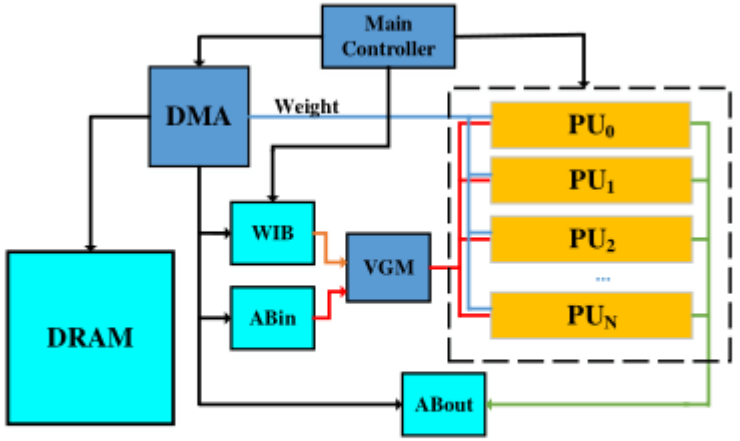


Fig. 6. Accelerator Architecture. The DRAM is implemented with Double Data Rate Random Access Memory in Processing System on Xilinx FPGA.

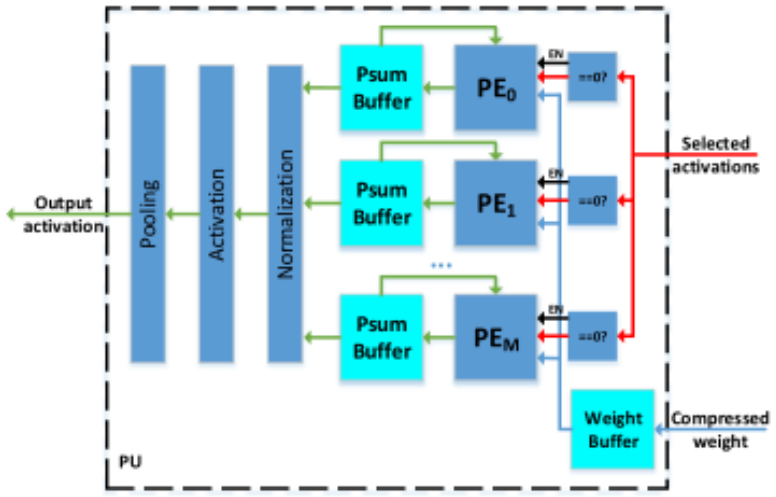


Fig. 10. The architecture of the PU. The PU processes all operations in CNNs. It contains M homogeneous PEs, and the number of PEs can be configured according to the CNN model and FPGA platform.

01 Paper

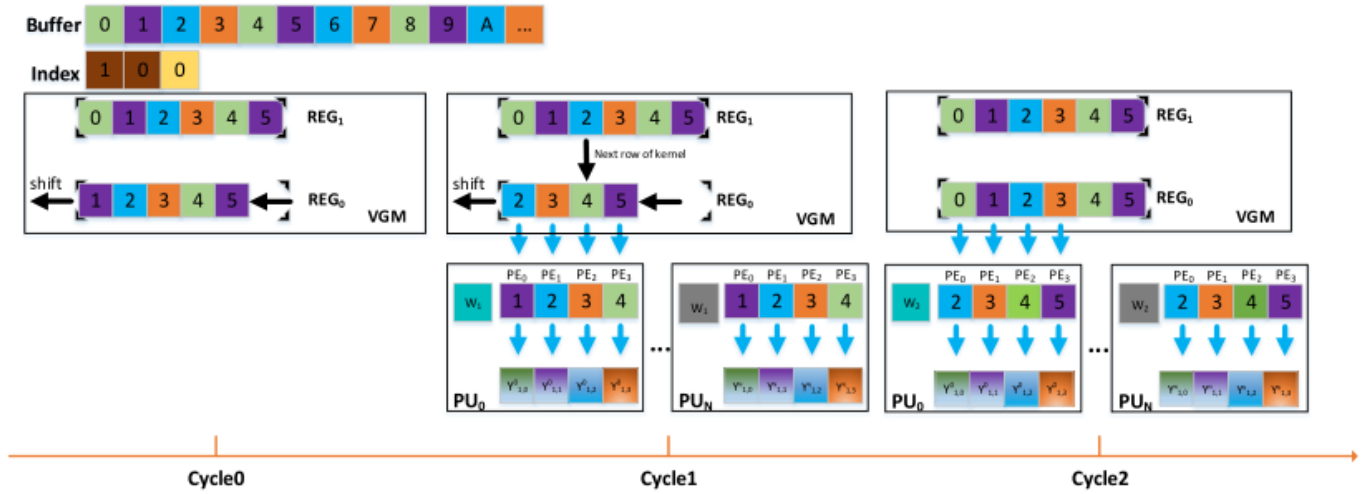


Fig. 9. Vector Generator Module. The VGM buffers and selects input activations to reuse them and to address the weight sparsity. It is shared by all the PUs.

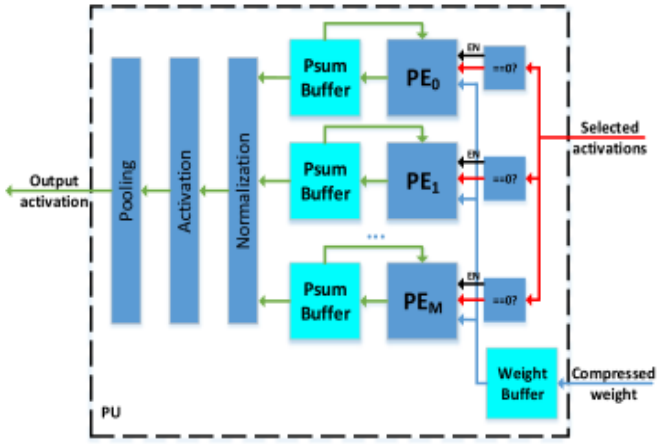


Fig. 10. The architecture of the PU. The PU processes all operations in CNNs. It contains M homogeneous PEs, and the number of PEs can be configured according to the CNN model and FPGA platform.

Conclusion:

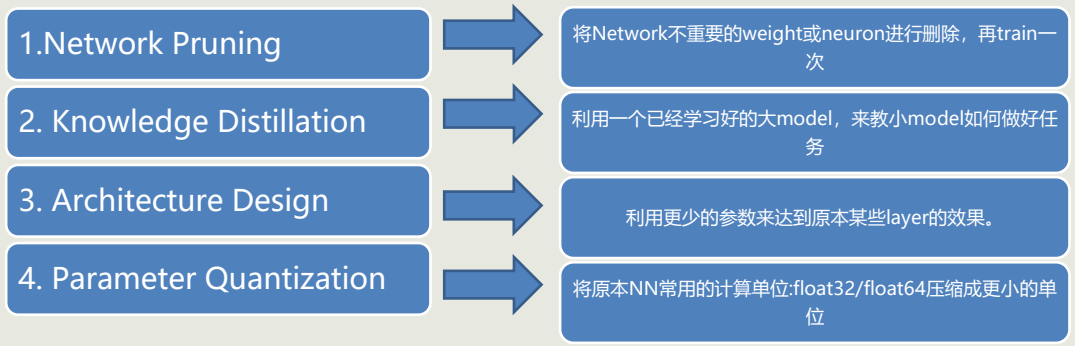
our implementation could achieve 987imag/s and 48imag/s performance for AlexNet and VGG-16 on Xilinx ZCU102, respectively, which provides 1.5× to 6.7× speedup and 2.0× to 6.2× energy-efficiency over pre-vious CNN FPGA accelerators

PART

02

- 网络压缩

02/ Video



问题：1.训练的剪枝是否和推理的剪枝方法相同， train时候的剪枝可以剪去neuron和weight，推理的时候本文只对weight中的0做了处理。
2. 上篇论文中所做的只是在计算时进行了优化，并没有对网络进行实质性的改变。