# EadNet: Efficient Architecture for Decomposed Convolutional Neural Networks

Fangxuan Sun, Jun Lin and Zhongfeng Wang
Department of Electronic Science and Engineering, Nanjing University, P.R. China

## Motivation

**Compression results of CPD-CNNs [1]**

- compress the weights by 7x
- reduce the computational complexity by 3.5x

**Drawbacks of current accelerators on CPD-CNNs**

- unnecessary on-chip memory
- extra data access

## Main Contributions

- different data reuse schemes are analyzed for CPD-CNNs
- a computation flow which can efficiently cache activations and reduce the size of on-chip memory is proposed
- an optimized hardware architecture called EadNet is proposed

## Comparisons

| | Layer | Area ($mm^2$) | SRAM (KB) | Frequency (MHz) | MACs Num. | Power (mW) | $T_{typ}$ (TOP/s) | BW (GB/s) | Efficiency (layers/s/MAC) |
|---|---|---|---|---|---|---|---|---|---|
| [2] | CONV2 | 12.25 | - | 200 | 168 | 288 | 0.067 | - | 0.57 |
| [3] | CONV2 | 16 | - | 200 | 512 | - | 0.205 | - | 0.81 |
| EadNet | CONV2 | 6.11 | 150 | 600 | 256 | 286 | 3.049 | 2.76 | 3.98 |
| [2] | CONV3 | 12.25 | - | 200 | 168 | 266 | 0.067 | - | 1.01 |
| [3] | CONV3 | 16 | - | 200 | 512 | - | 0.205 | - | 1.13 |
| EadNet | CONV3 | 9.05 | 212 | 700 | 256 | 507 | 3.19 | 2.62 | 10.28 |

**Table 1:** Results and Comparisons

## References

1. Astrid M, Lee S I. CP-decomposition with Tensor Power Method for Convolutional Neural Networks Compression[C]. *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on. IEEE*, 2017: 115-118.
2. Chen Y H, Krishna T, Emer J S, et al. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks[J]. *IEEE Journal of Solid-State Circuits*, 2017, 52(1): 127-138.
3. Tu F, Yin S, Ouyang P, et al. Deep convolutional neural network architecture with reconfigurable computation patterns[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017, 25(8): 2220-2233.

## Architecture

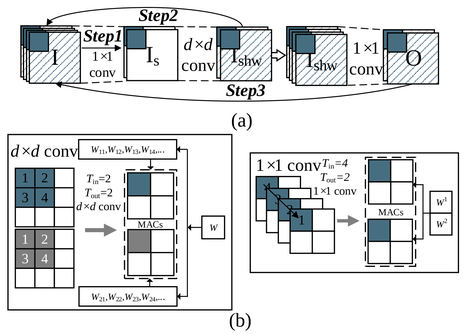


**Figure 1:** (a) The computation flow of EadNet. (b) Computation patterns

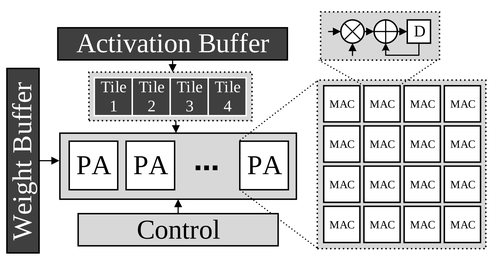


**Figure 2:** Overview of architecture

**Computation flow**

- **Step 1:** $N_{in}$ maps are sent to PAs in serial to get $N_P$ tiled feature maps of $I_s$
- **Step 2:** $N_P$ tiled $I_s$ are directly used for the computation of $I_{shw}$ rather than being buffered. $N_P$ tiled maps of $I_{shw}$ will be obtained and sent to on-chip buffers.
- **Step 3:** After all $R$ tiled maps of $I_{shw}$ are gotten, $N_{out}$ tiled output feature maps will be calculated and sent to off-chip DRAM sequentially.

**Architecture**

- The Weight Buffer unit is an SRAM which stores all convolution weights
- The Activation Buffer unit stores temporary activations
- Each Tiled Registers (TR) unit stores a $T_H \times T_W$ tiled feature map. $T_{in}$ TR units are employed
- PA is used for all computations in convolutional layers

## Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants No. 61774082 and 61604068; the Fundamental Research Funds for the Central Universities under Grant No. 021014380065