A Brief Introduction to Resource Constrained Embedded Deep Learning

Mehrdad Yaghoobi

Institute for Digital Communications, University of Edinburgh

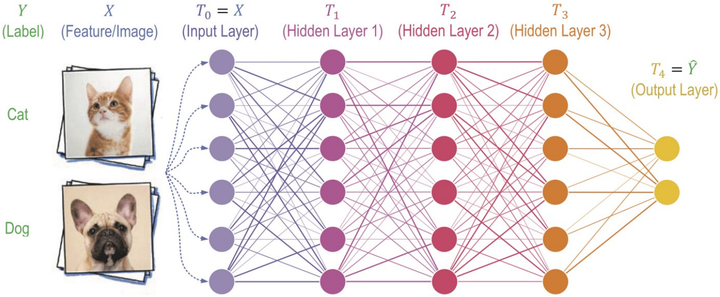
UDRC Summer School, Machine Learning Track, June 2020

Deep Neural Networks Deployment

- Brining the success of deep learning to the ”sensor” side
- Running machine learning tasks on the edge, in real time
- Preserving data privacy and reducing the dependency on the network access

A Recap on Deep Neural Networks Inference and Deployment

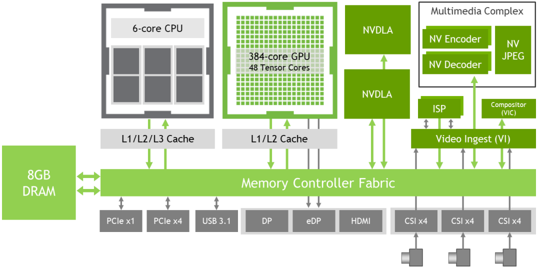
Deep Neural Networks



https://bit.ly/30V55AC

- Consecutive **multiplication**, **additions**, and **non-linear operators**; $\hat{Y} = f_4(f_3(\ldots f_2(\mathbf{W2} * f_1(\mathbf{W1} * X + \mathbf{b1}) + \mathbf{b2})))$.

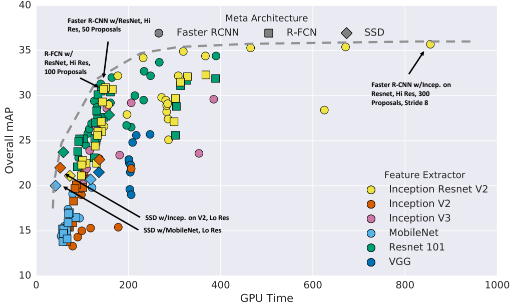
Embedded Deep Neural Networks Deployments



Jetson Xavier NX processor engines, high-speed I/O, and memory fabric

- Challenges to be addressed:
  1. Computational complexity
  2. Memory limitation
  3. Communication bandwidth

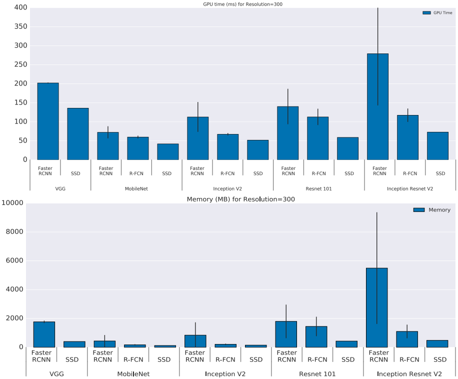
Computational Complexity



arXiv:1611.10012v3

- While DNN inference needs much less computational resources than learning, **real time** implementations on power constrained embedded platforms are still challenging.
- More computation gives more accuracy, *i.e.* mean Average Precision (mAP)

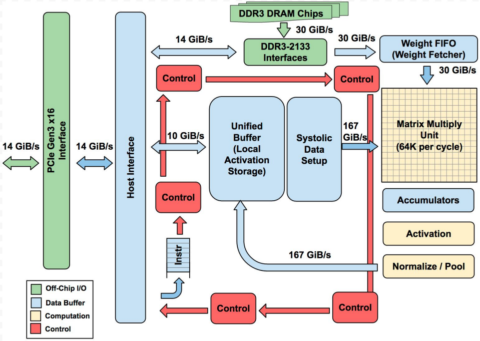
Memory Limitation



arXiv:1611.10012v3

- Higher capacity networks often need more memory
- Latency in memory read-out can be the bottleneck

I/O and Communication Bandwidths and Delays

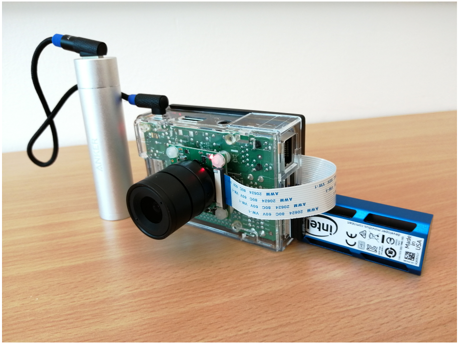


Google (HPC) TPU structure. This bandwidth is not achievable in the embedded TPUs.

- Issues in acquiring full rate signals/images
- Asynchronous sensor measurements
- Time delays in measurements

ASIC/FPGA AI Accelerators

Hardware Platforms and AI Accelerators

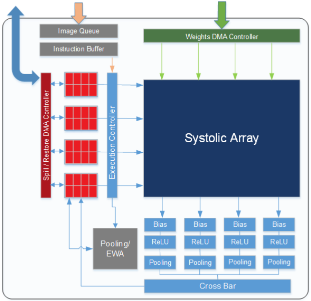


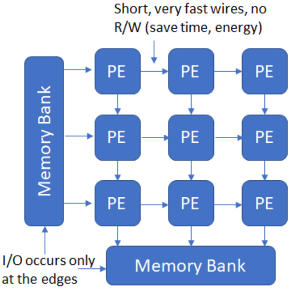
- Low power CPU or Microcontroller, *e.g.* ARM, RISC-V
- I/O interfaces and sensors, *e.g.* camera(s), microphones, depth-sensors
- ASIC or FPGA accelerators, *e.g.* GPU, TPU and RISC many-core processors

AI Accelerators

- Matrix-matrix multiplication accelerator, Multiplier-Accumulator (MAC), Systolic MAC
- Structured fast memory, Multi-level dedicated/shared caches (L1/L2 cache GPU)
- Recursive implementation facilitators

Systolic Multiplier-Accumulator

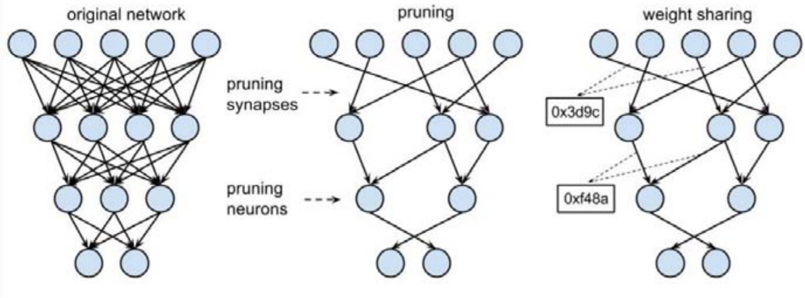




"Exploration and Tradeoffs of Different Kernels in FPGA Deep Learning Applications" by Elliott Delaye, 2018

- The breakthrough in computation is in the systolic MAC or massive parallel processing elements (PE),
- PEs in a systolic MAC are configured for one- (few-) shot tensor multiplications
- PEs are suitable for scaling up array sizes.

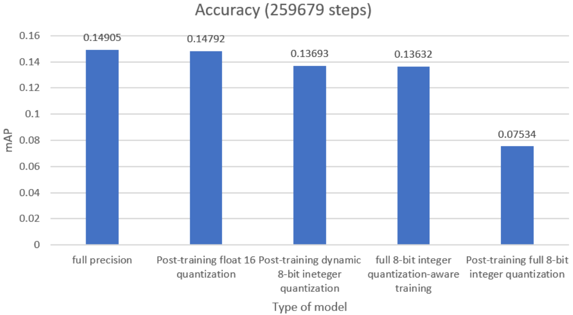
Embedded Deep Models



"Exploration and Tradeoffs of Different Kernels in FPGA Deep Learning Applications" by Elliott Delaye, 2018

- Learned models with highest mAP: over-parametrized with double precision weights
- Model simplification with pruning: cutting ineffective weights and sharing weights
- Model quantization: not losing much using quantized models

Quantised Models



mAP of MobileNetV2 SSD in different settings

- Full precision models: computationally and memory expensive
- Simple *post-quantization*: degrades mAP performance
- Quantized learning and dynamic post-quantization : showing close performance in a comparison with full precision.

Take-Home Messages

- Various benefits in **on-the-device** and **on-the-edge** computing
- **Embedded DNN Deployment Hardware**
  - Powerful **AI accelerators** are available now and they will be highly optimised in the near future
  - Each structure has its pros and cones. They mainly accelerate **parallel MAC** operations
  - Integration of multiple AI units accelerates almost linearly
- **Embedded DNN Deployment Models**
  - **Pruning:** sparse and shared weights
  - **Quantization:** acceleration given a fixed IC footprint and lower power consumption