POLYTECHNIQUE MONTRÉAL

Hardware Support Architectures and Implementation Methods for Effective Embedded AI;

Contributions of the NN2GATE/CNN2GATE Project (Ivado funded)

SOME EXPERIMENTS WITH GENERAL FRAMEWORKS TO IMPLEMENT DEEP CONVOLUTIONAL NEURAL NETWORKS

IMPLEMENTING WHITE BOX DEPLOYABLE EMBEDDED AI

YVON SAVARIA, ALIREZA GHAFFARI, MOHAMMED HOSSEIN ASKARI HEMMAT, AND JEAN-PIERRE DAVID. EE DEPT POLYTECHNIQUE MONTREAL

Context and Motivation

AI not so new (1959)

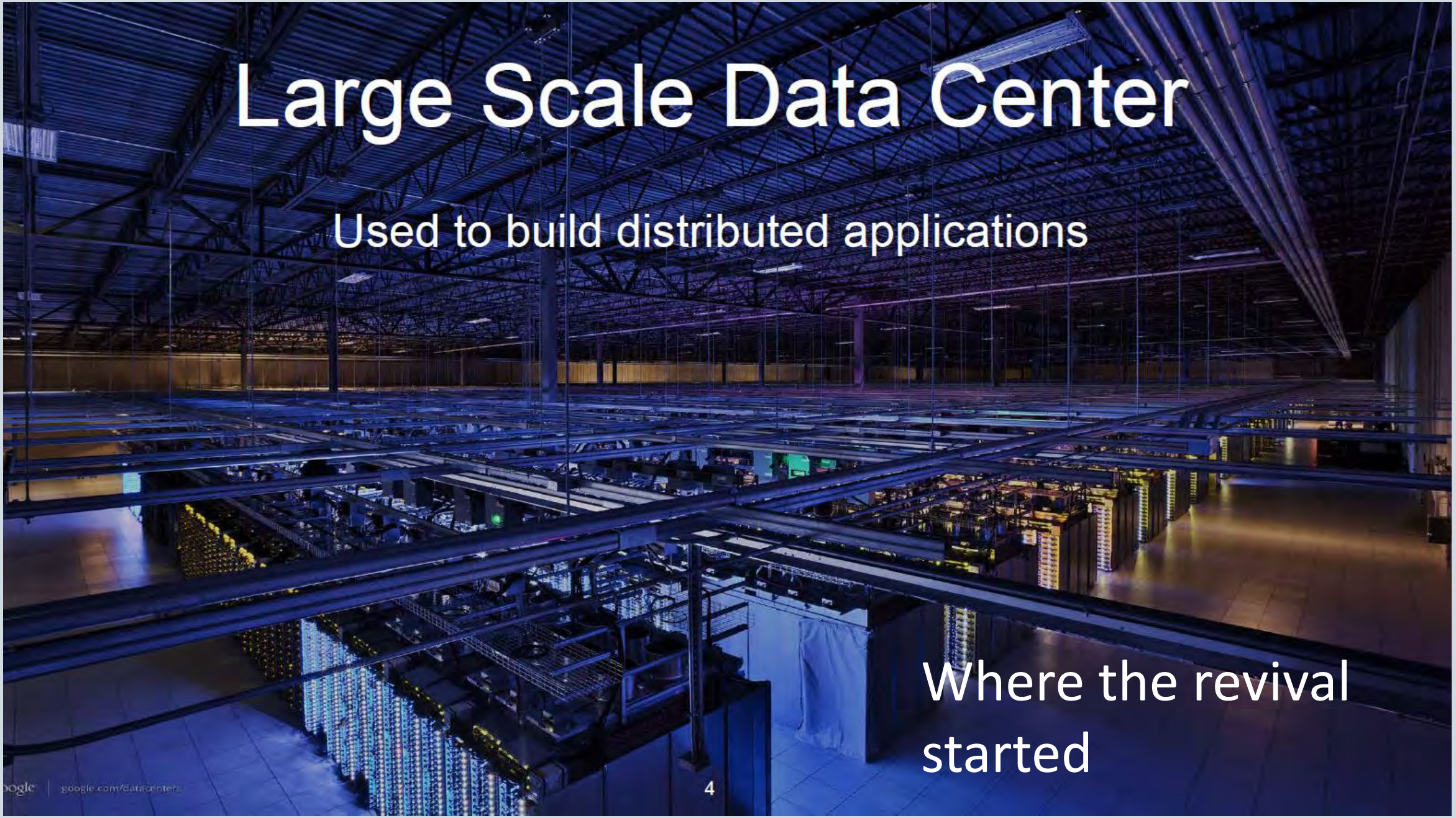
- GPUs enabled effective training of DNNs (2012)
- This sparked a revival of interest and massive investments fueling a revolution

Bubbles blow over unrealistic unfulfilled promises

- However,
  - so much is getting delivered by so many,
  - and a lot more in on the way.
    - I expect a long sustained stream of innovation; like an AI-Moore's law

A revolution is on its way

- If you cannot beat them then join them


Large Scale Data Center
Used to build distributed applications
Where the revival started
google.com/datacenters
4

ASIC Chips for Embedded AI (1)

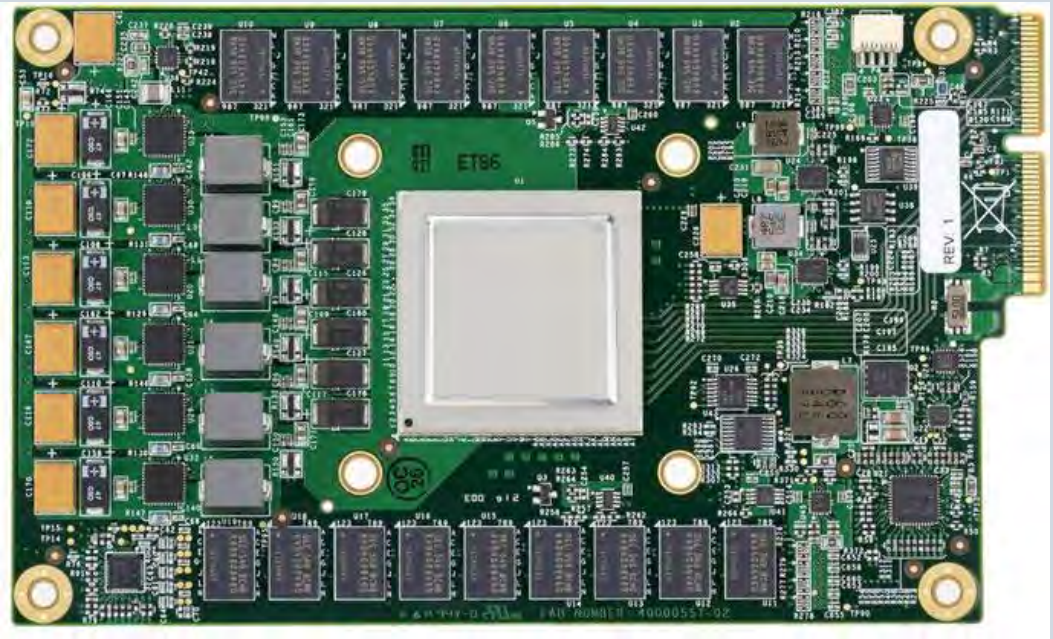
- WHERE IT IS GOING (MITTAL 2018)

'IT IS WIDELY BELIEVED THAT THE ABILITY TO RUN AI ALGORITHMS ON LOW-COST, LOW-POWER PLATFORMS WILL BE CRUCIAL FOR ACHIEVING THE "AI FOR ALL"

- TESLA'S FULL SELF DRIVING CHIP: HTTPS://YOUTUBE/UCP0TTMVQOE?T=4585
  - Throughput:
    - 2300 frames per second
    - 72 TOPS @ 2GHZ for full precision ops
  - Number of Dies: 2
  - Technology: 14nm FinFET CMOS
  - Memory: 32MB SRAM
  - Power: 72W

ASIC Chips for Embedded AI (2)

- GOOGLE'S TENSOR PROCESSING UNIT (TPU):HTTPS://ARXIV.ORG/PDF/1704.04760.PDF
  - Throughput:
    - 92 TOPS @ 700MHZ for 8-bit precision ops
  - Number of Dies: 4
  - Technology: 28nm
  - Memory: 28MiB SRAM
  - Power: 40W



ASIC Chips for Embedded AI (3)

- BITMAIN'S BM1682 CHIP: HTTPS://WWW.SOPHON.AI/PRODUCT/INTRODUCE/BM1682.HTML
  - Throughput:
    - 3 TOPS for full precision ops
    - 360 frames per second
  - Technology: 28nm
  - Memory: 16MB SRAM

ASIC Chips for Embedded AI (4)

- HABANA'S GOYA CHIP: HTTPS://HABANA.AI/PRODUCT/
  - Throughput:
    - Model: Resnet152
    - Batch size: 8
    - 4200 Images per second
    - Power: 100 W

A Survey on Optimized Implementation of Deep Learning Models on the NVIDIA Jetson Platform Sparsh Mittal IIT Hyderabad. E-mail:sparsh@iith.ac.in. Dec.2018

Selected parameters of Jetson, Raspberry Pi and Intel UP (the prices are as of Dec 2018. DP/SP = double/single-precision). For Jetson, peak performance is GPU's performance, whereas for Raspberry, it is CPU's performance. These systems also have other accelerators which are not shown.

	TK1	TX1	TX2	Raspberry Pi 3(B+)	Intel UP
Feature size	28nm	20nm	16nm	40nm	data not found
GPU	192-core Kepler	256-core Maxwell @ 998MHz	256-core Pascal @ 1300MHz	VideoCore IV	Intel HD 400 Graphics, upto 500 MHz
CPU	"4-Plus-1" 2.32GHz ARM quad-core Cortex-A15 CPU with Cortex-A15 battery-saving shadow-core	ARM Cortex-A57 (quad-core) @ 1.73GHz	ARM Cortex-A57 (quad-core) @ 2GHz + NVIDIA Denver2 (dual-core) @ 2GHz	Broadcom BCM2837B0 quad-core ARM Cortex-A53 @ 1.4GHz	Intel Atom x5-Z8350 quad-core CPU, 64-bits @ 1.92GHz
Memory	2GB DDR3L 933MHz EMC x16 using 64-bit data width	4GB 64-bit LPDDR4 @ 1600MHz, 25.6 GB/s	8GB 128-bit LPDDR4 @ 1866Mhz, 59.7 GB/s	1GB LPDDR2 @ 900MHz, 8.5 GB/s	4GB DDR3L @ 1600 MHz
Storage	16 GB eMMC	16 GB eMMC	32GB eMMC	MicroSDHC slot	16/32/64 GB eMMC
Peak performance	>300 SP Gflops [14]	512 SP Gflops [15]	665 SP Gflops [15]	6 DP Gflops [16, 17]	data not found
Power under load	10W	1W to 15W [18]	7.5W to 15W	1.5W to 6.7	6W
Weight	120 gram [19]	85 gram (with thermal transfer plate) [20]	85 gram (with thermal transfer plate) [20]	50 gram [21]	80 gram (with passive heat sink but without package) [22]
Price	Discontinued	$480	$570	$35	$100

And They Get Integrated in Embedded Systems (AI only? The end of software?)

Features

Edge TPU Module (SOM)

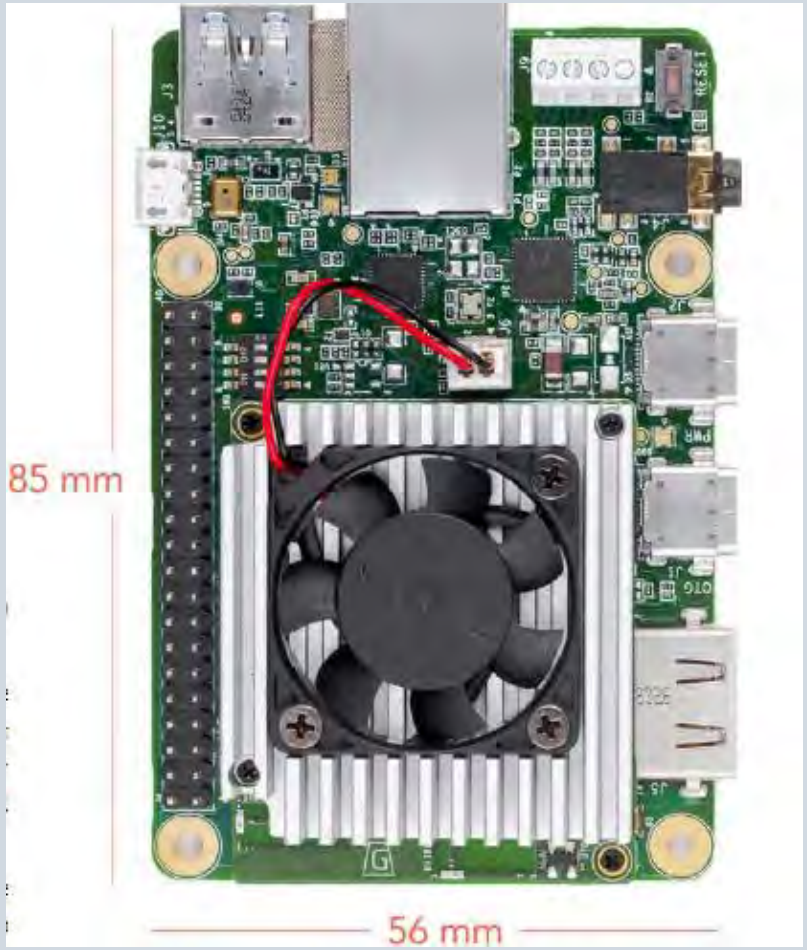
- *NXP i.MX 8M SOC (Quad-core Cortex-A53, plus Cortex-M4F)*
- *Google Edge TPU ML accelerator coprocessor*
- *Cryptographic coprocessor*
- Wi-Fi 2x2 MIMO (802.11b/g/n/ac 2.4/5GHz)
- Bluetooth 4.1
- 8GB eMMC
- 1GB LPDDR4

USB, Audio, Video connections

MicroSD, Gigabit Ethernet port

Linux

Must be powered by 2 - 3A at 5V DC using the USB Type-C power port



Rationale for our DNN2Gate project

ASICs in advanced CMOS cost **100M$US**

- **2-3 years effort per interation**
- Everything in under **VERY TIGHT NDA**
  - **Internal visibility?**
- **More than Moore evolution has 2 (maybe 3) orders of magnitude improvement for us ahead**
- **More effcient architecture can offer another 2 (maybe 3) order of magnitud improvement**
- **What about analog implementation; another story(project)**
- **Taining vs inference!**

Need to optimize (nowhere near the end of the road …)

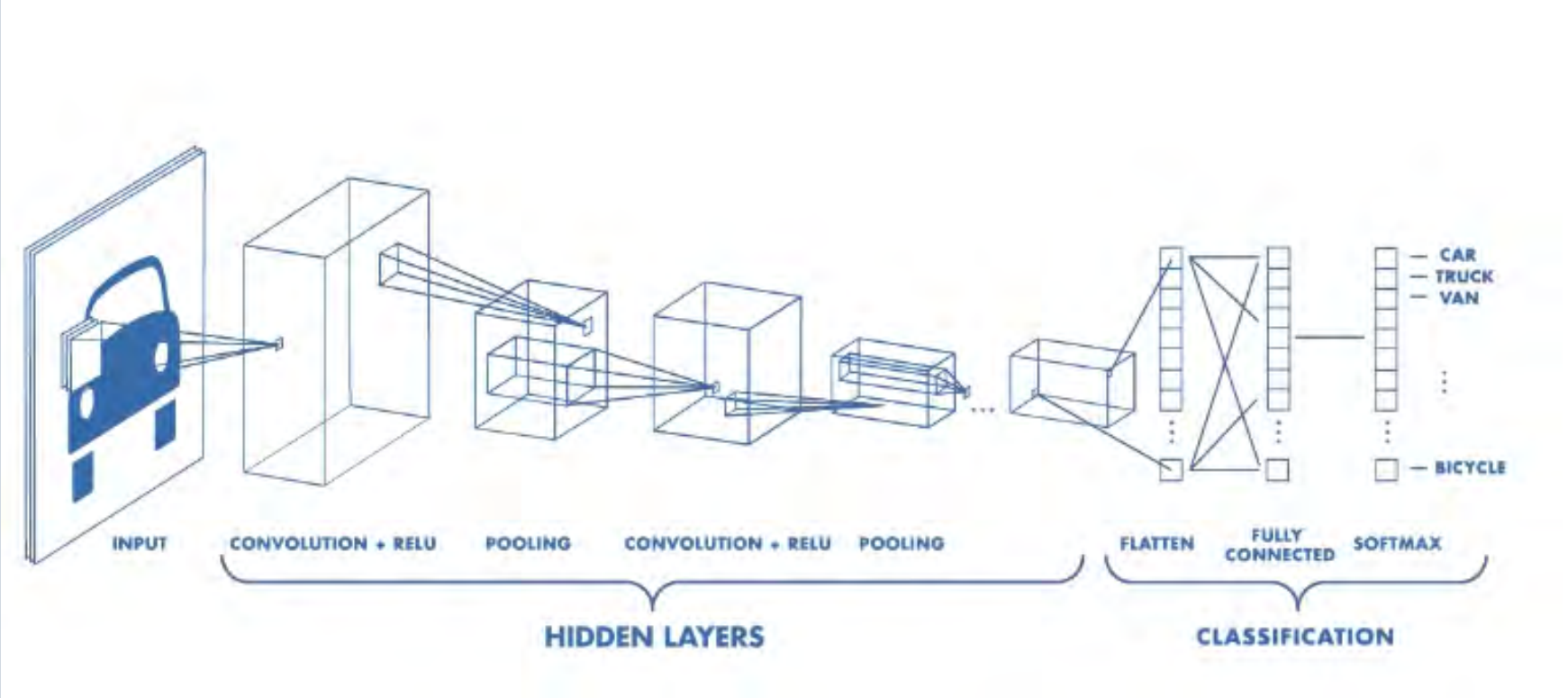
- **Does your brain FLOP?**
- **Why should this be needed??**

Need visibility to explain (certifiifaction)

- **Behavior**
- **Failures (FTA)**

Funding of the DNN2Gate project 60dB below what the major do; started 16 monts ago!

Convolutional Neural Networks (CNN)



Typically 90% of the computational effort!

Convolutional Neural Networks (CNNs)

1. CNNs employed in many applications
   image classification, video analysis and speech recognition.
2. Convolutional kernels are compute intensive part of CNNs.
   highly accelerated by GPUs at the expense of high power consumptions.
3. Because of their higher power consumptions, GPUs are unsuitable for many industrial and mission critical scenarios.
4. Conventional GPUs are hard to use in robotics, drones, self-driving cars and Internet of Things (IoTs) while these fields can highly benefit from deep learning algorithms.
5. Conventional GPUs are typically accessible through some PCI bus on their host computer.
   makes them hard to use them in mission-critical scenarios that need prompt control actions through real-time I/Os.
6. Field Programmable Gate Arrays (FPGAs) can be used in these scenarios to tackle problems introduced by limitations of GPUs without compromising the accuracy of the algorithm.
   Provide many degrees of freedom for optimization and visibility

AI Algorithms On FPGAs

Why they are not so widespread?

1. Their design needs special expertise to develop AI algorithms on FPGAs.
2. Not easy to use and maintain by machine learning researchers
3. FPGA implementation is different from algorithm to algorithm and must be optimized by hand for each case.
4. It might take months to achieve a good FPGA implementation for a simple AI algorithm.
5. The FPGA design environments that exist today are either general purpose or target other types of applications than AI.

   Thus there is room to improve results by designing an environment specific to deep learning.

   Can provide great visibility

AI Algorithms On FPGAs

Why they should be used?

1. FPGAs can be used in industrial use cases to mitigate limitations of GPUs without compromising results accuracy
2. Quantized deep learning algorithms can solve the power consumption issue on FPGAs
   size of implemented circuits shrink with quantization.
3. Having massive connectivity through I/Os is natural in FPGAs.
4. FPGAs are efficient in robotics, drones, self-driving cars and Internet of Things (IoT) in contrast to conventional GPUs.
5. FPGAs can be re-programmed to new architectures suitable for new AI algorithms
   can make them very cost effective in industrial use cases (particularly in low volume applications).
6. FPGA allow open white both implementations

CNNs On FPGAs

Some notable implementations

	fpgaConvNet [1]	DNNWeaver [2]	Caffeine [3]	SnowFlake [4]
FPGA Platform	Zynq XC7045	Zynq XC7020	UltraScale KU060	Zynq XC7045
Frequency	125 MHz	150 MHz	200 MHz	250 MHz
Logic capacity	218.60 kLUTs	53.2 kLUTs	331.68 kLUTs	218.6 kLUTs
Latency (Batch size=1)	8.22 ms	N/A	N/A	9.95 ms
Arithmetic Precision	16 bit fixed point	16 bit fixed point	16 bit fixed point	16 bit fixed point
Performance GOp/s	197.4 (CONV)	20.16 (CONV)	163 (CONV)	120.3 (CONV)

AI Algorithms On FPGAs

Are they going to be accessible by high-end AI developers?

1. Research (architecture, synthesis and optimisation) is progressing rapidly.
2. The next challenge is to improve the results by integrating them in a single tool.
3. Our project (**DNN2Gate/CNN2gate**)

   designing and implementing a High-Level Synthesis (HLS) tool that will generate an **RTL** (Register-Transfer Level) design from the code of an algorithm in **Python**.
4. Is our project a Python to VHDL code-generator? **NO**
5. **CNN2gate** is a Python library that takes a high-end model in Tensor-flow, Keras, Caffe or Pytorch and convert it to RTL.

CNN2gate

1. CNN2gate is mainly based on the capability of currently available FPGAs design tools to use pipelined RTL libraries as OpenCL kernels in high level synthesis.
2. These kernels can help to optimize
   - memory-bandwidth
   - Enable data re-use
   - Enable task mapping capability
   - Includes a library of primitives to which we will add the synthesis directives necessary to achieve better performance
3. CNN2gate uses an open source transport format to exchange deep neural network variables (weights and biases) to hardware.

CNN2gate

pipelined RTL libraries as OpenCL kernels

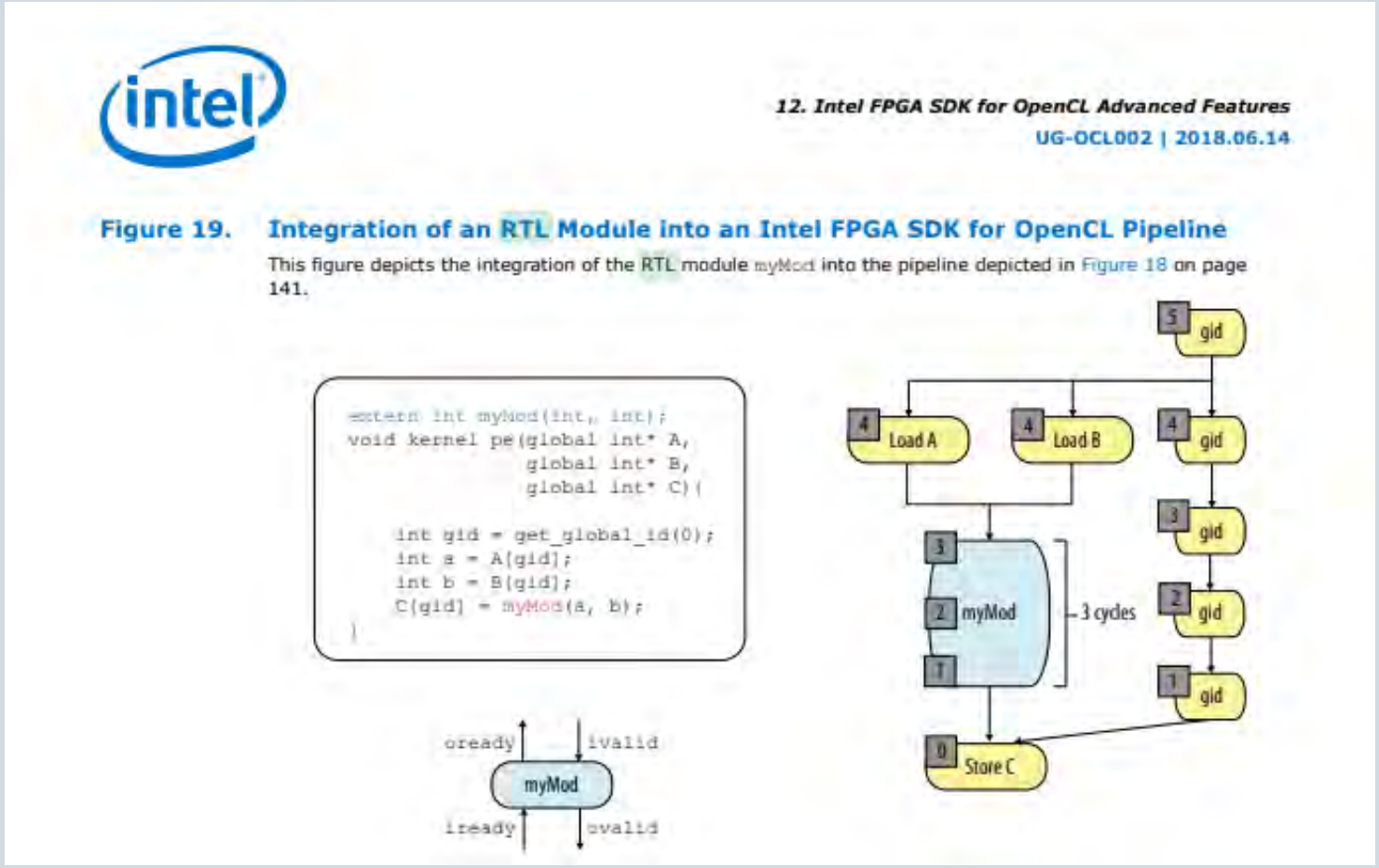


Figure is courtesy of Intel.

CNN2gate
pipelined RTL as **SIMT** compute units

1. Efficient convolution kernels can be designed using pipelined RTLs in Intel or Xilinx FPGAs.
2. Loops can be avoided partially using adder trees and other reduction strategies.

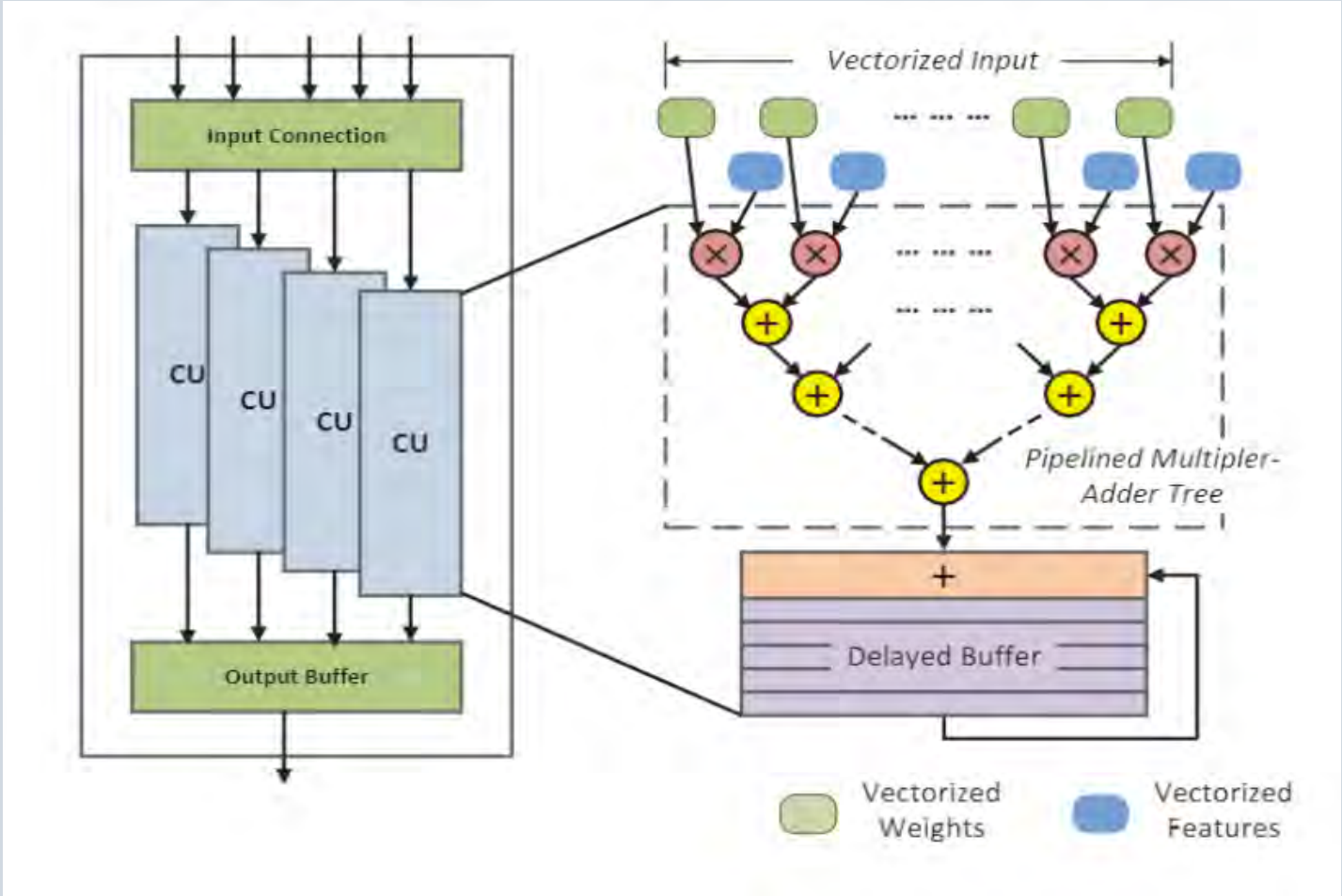
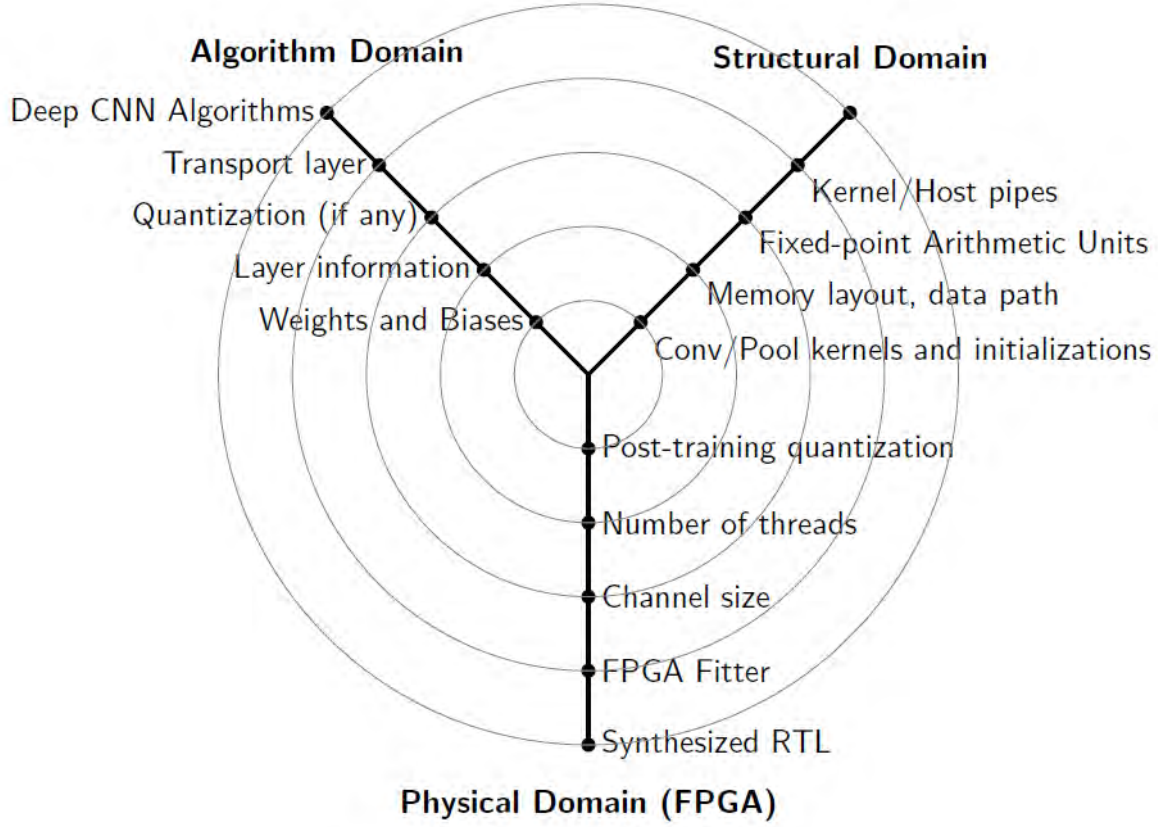


Figure reference: Wang, Dong, Ke Xu, and Diankun Jiang. "PipeCNN: An OpenCL-based open-source FPGA accelerator for convolution neural networks." *2017 International Conference on Field Programmable Technology (ICFPT)*. IEEE, 2017.

CNN2gate

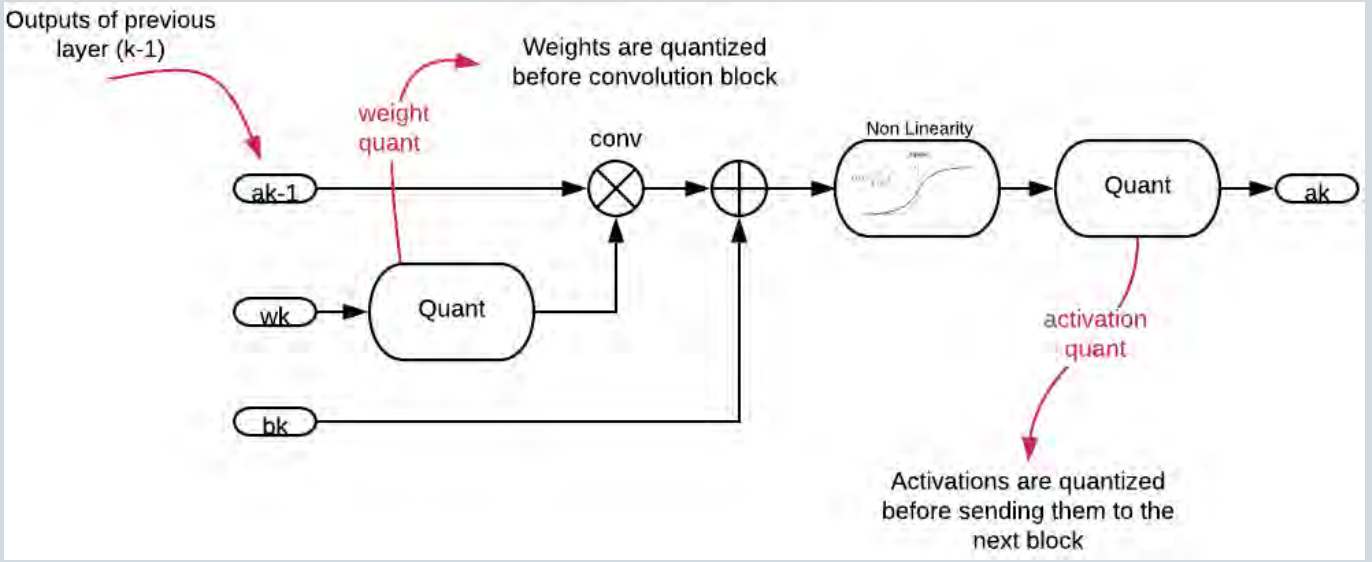
Design methodology



Inspired by the classic Gajski Y-Chart

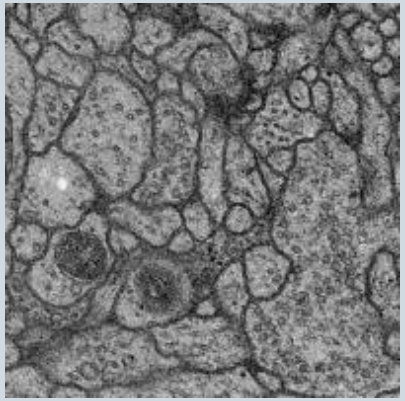
Model Quantization in Deep Neural Networks

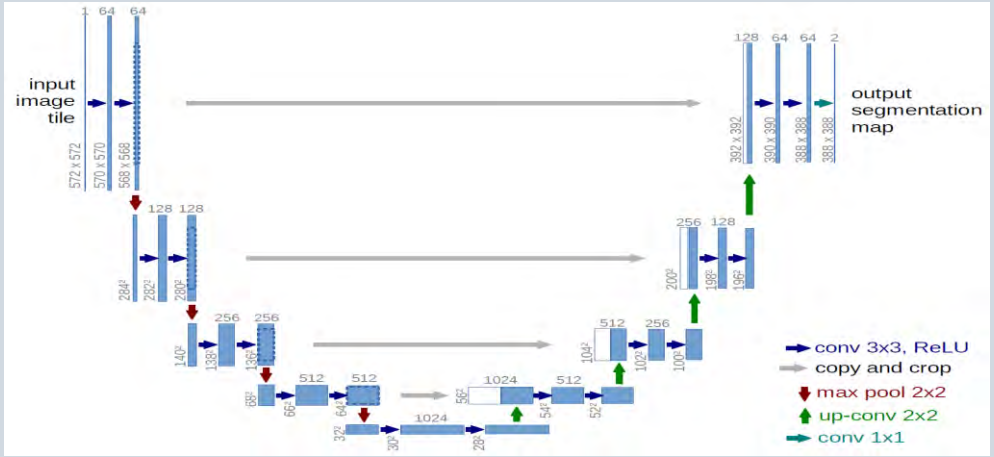
- QUANTIZATION CAN REDUCE MEMORY CONSUMPTION AND THE COMPUTATION TIME OF DEEP NEURAL NETWORKS.
- WEIGHT QUANTIZATION CAN REDUCE MEMORY FOOTPRINT.
- ACTIVATIONS ARE QUANTIZED TO ACCELERATE COMPUTATION AND REDUCE MEMORY FOOTPRINT.
- BIASES OFTEN USED IN FULL PRECISION SINCE HIGH PRECISION ACCUMULATORS ARE ALREADY USED IN CONVOLUTION OUTPUT BLOCKS
  - (no need/benefit to quantize biases).



Model Quantization in Deep Neural Networks

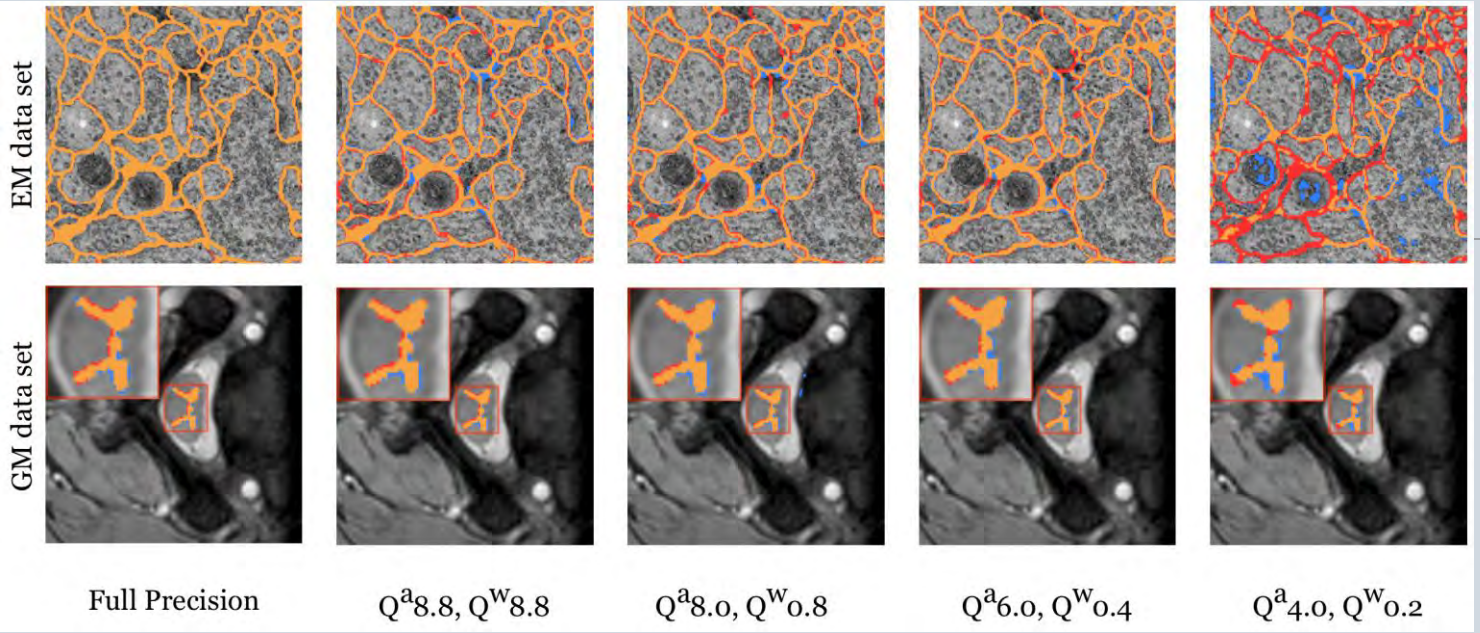
- Model quantization used in many deep learning tasks.
- Model quantization is applied here to a segmentation task in medical imaging.
- Segmentation classifies every pixel in an image.
- The widely known network called U-NET was used .





Model Quantization in Deep Neural Networks

- QUANTIZATION WAS APPLIED TO SPINAL CORD GRAY MATTER (GM) SEGMENTATION AS WELL AS TO THE ISBI CHALLENGE FOR SEGMENTATION OF NEURONAL STRUCTURES IN ELECTRON MICROSCOPIC STACKS (EM).
- OUR FINDING: 4-BIT WEIGHTS AND 6-BIT FOR ACTIVATION IS ENOUGH TO GET EXCELLENT ACCURACY (CLOSE TO FULL PRECISION)



CNN2gate

Some results

Summary of the measured performance and cost on different platforms; its working!

Platform	FPGA type	Resource Capacity	Resource Consumed	Execution time	Frequency
				AlexNet	
DE1-soc	Cyclone-V SEA5	85K LEs 87 DSPs	66K LEs 29 DSPs	420 ms	50 MHz
Nallatech 510	Arria-10 GX1150	1150 K Les 1518 DSPs	997K LEs 122 DSPs	22 ms	100 MHz

How does it compare (ASICs and GPUs)

we are working on it

not so well in a all counts

Take Home Summary

A revolution is on its way in embedded AI

- If you cannot beat them then join them

‘It is widely believed that the ability to run AI algorithms on low-cost, low-power platforms will be crucial for achieving the “AI for all” ‘

- **More than Moore evolution has 2 (maybe 3) orders of magnitude improvement for us ahead**
- **More effcient architecture can offer another 2 (maybe 3) orders of magnitude improvement**
- **What about analog implementation; another story(project)**
- **Training vs inference! Yet another story**

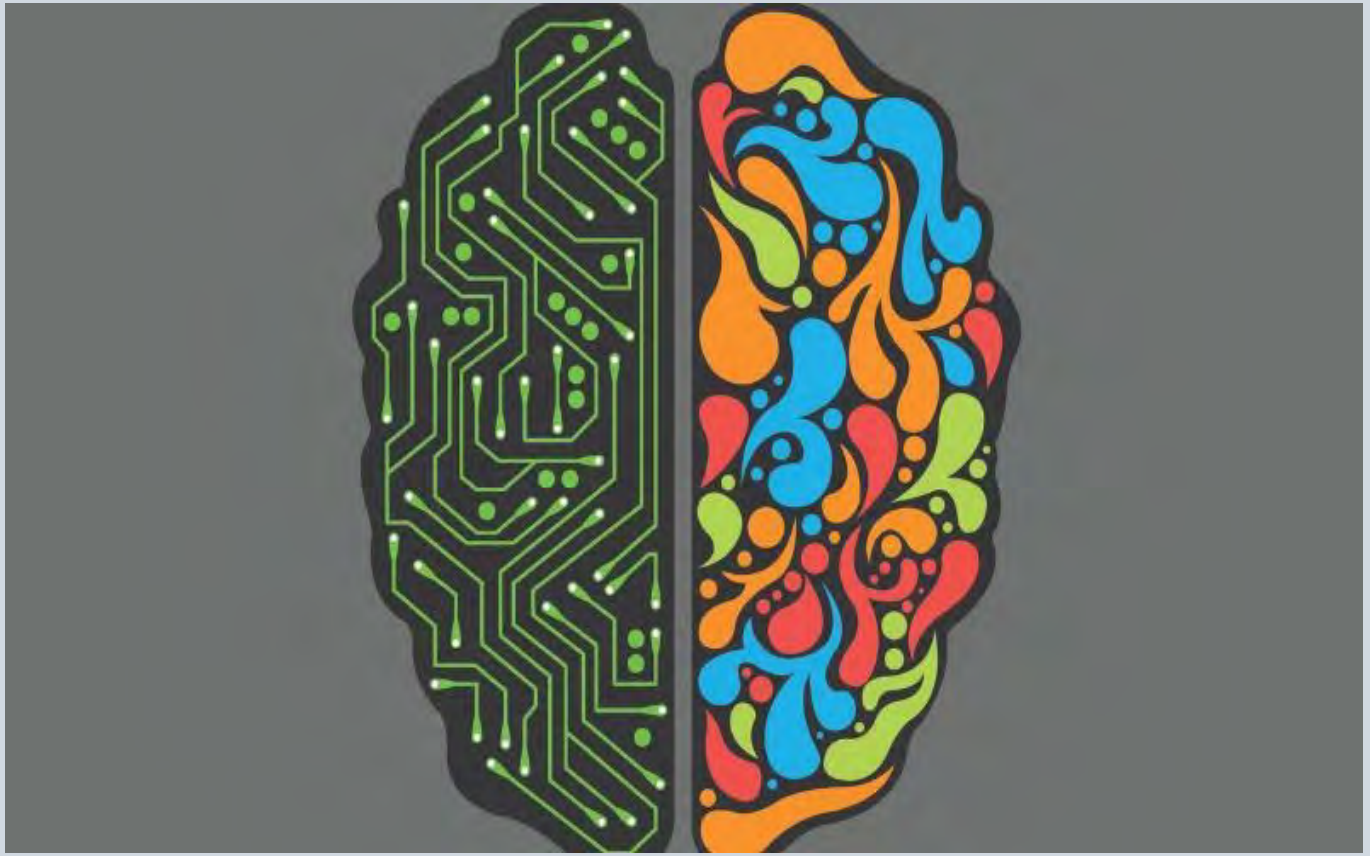
Need to optimize (nowhere near the end of the road …)

- **Does your brain FLOP?**
- **Why should this be needed??**

Need visibility to explain

- **Behavior**
- **Failures (FTA)**
  - **And for aeropace (and other critical applications)** ***An enabler for certification***

Thank for Attention
Questions???



References

[1]. S. Venieris, S.I. and Bouganis, C.S., 2017. fpgaConvNet: A toolflow for mapping diverse convolutional neural networks on embedded FPGAs. arXiv preprint arXiv:1711.08740.

[2]. Sharma, H., Park, J., Amaro, E., Thwaites, B., Kotha, P., Gupta, A., Kim, J.K., Mishra, A. and Esmaeilzadeh, H., 2016. Dnnweaver: From high-level deep network models to fpga acceleration. In the Workshop on Cognitive Architectures.

[3]. Zhang, C., Sun, G., Fang, Z., Zhou, P., Pan, P. and Cong, J., 2018. Caffeine: Towards uniformed representation and acceleration for deep convolutional neural networks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[4]. Gokhale, V., Zaidy, A., Chang, A.X.M. and Culurciello, E., 2017. Snowflake: A model agnostic accelerator for deep convolutional neural networks. arXiv preprint arXiv:1708.02579.

[5] Mittal S. A Survey on optimized implementation of deep learning models on the NVIDIA Jetson platform. Journal of Systems Architecture. 2019 Jan 25.