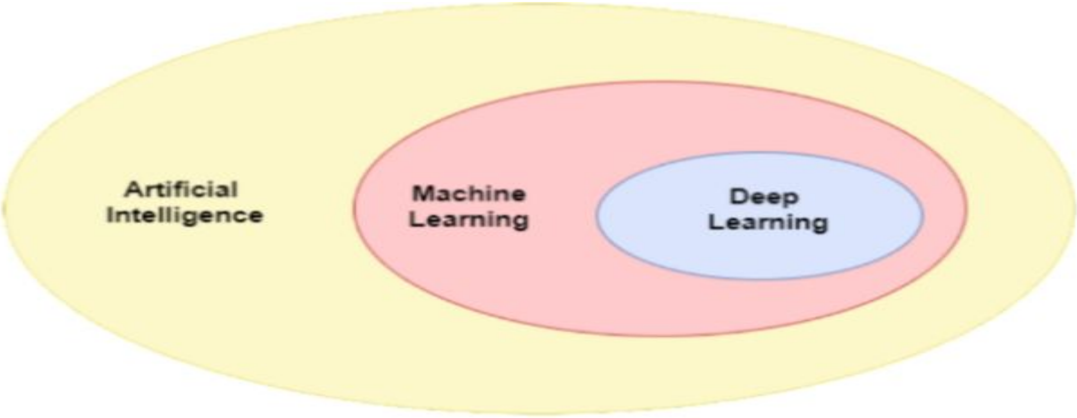
Hardware Architectures and Design for AI

Binod Kumar
Electrical Engineering Department
IIT Jodhpur

Talk at IIT Tirupati (6th April 2022)

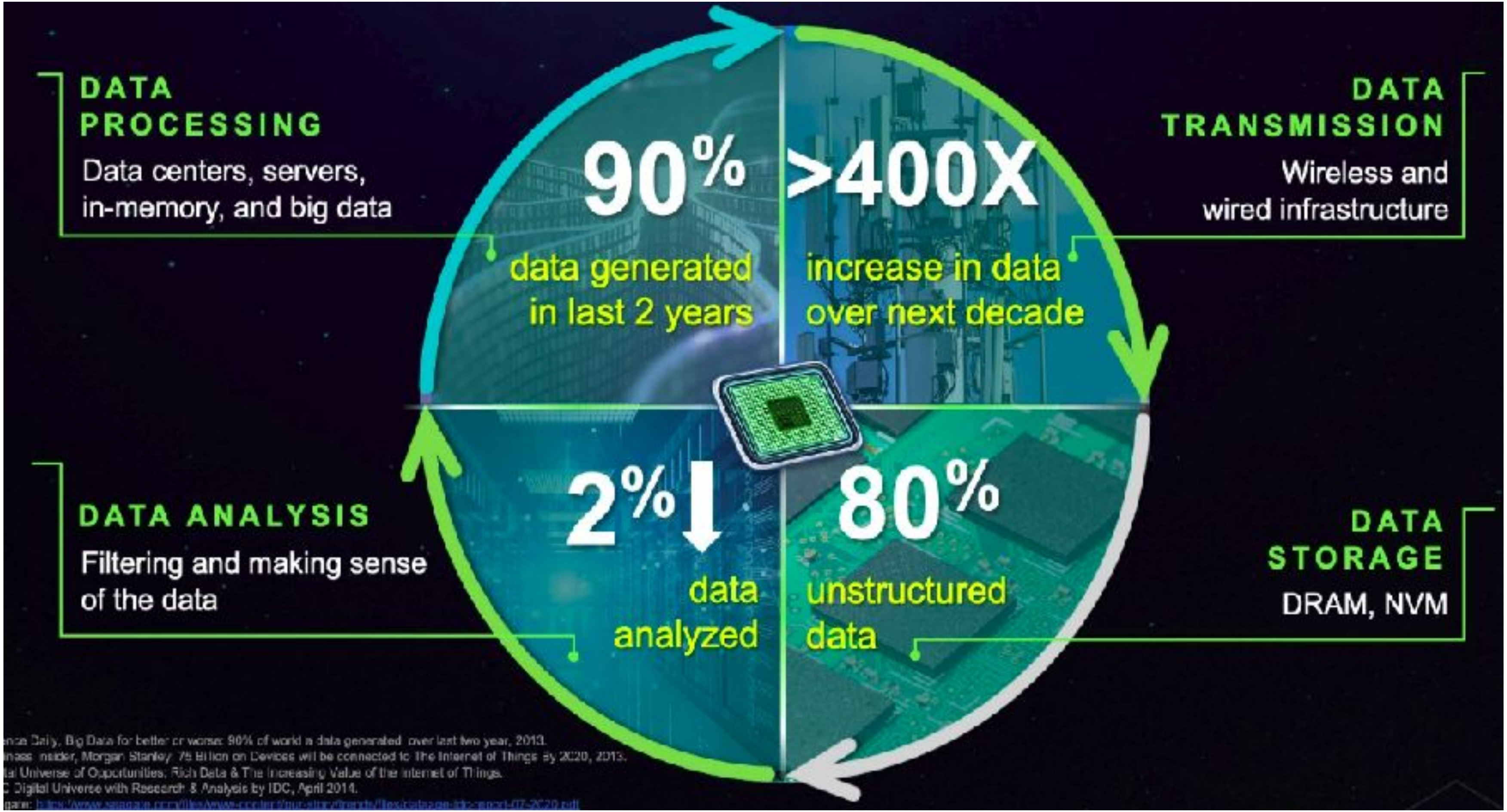
The Technology Evolution



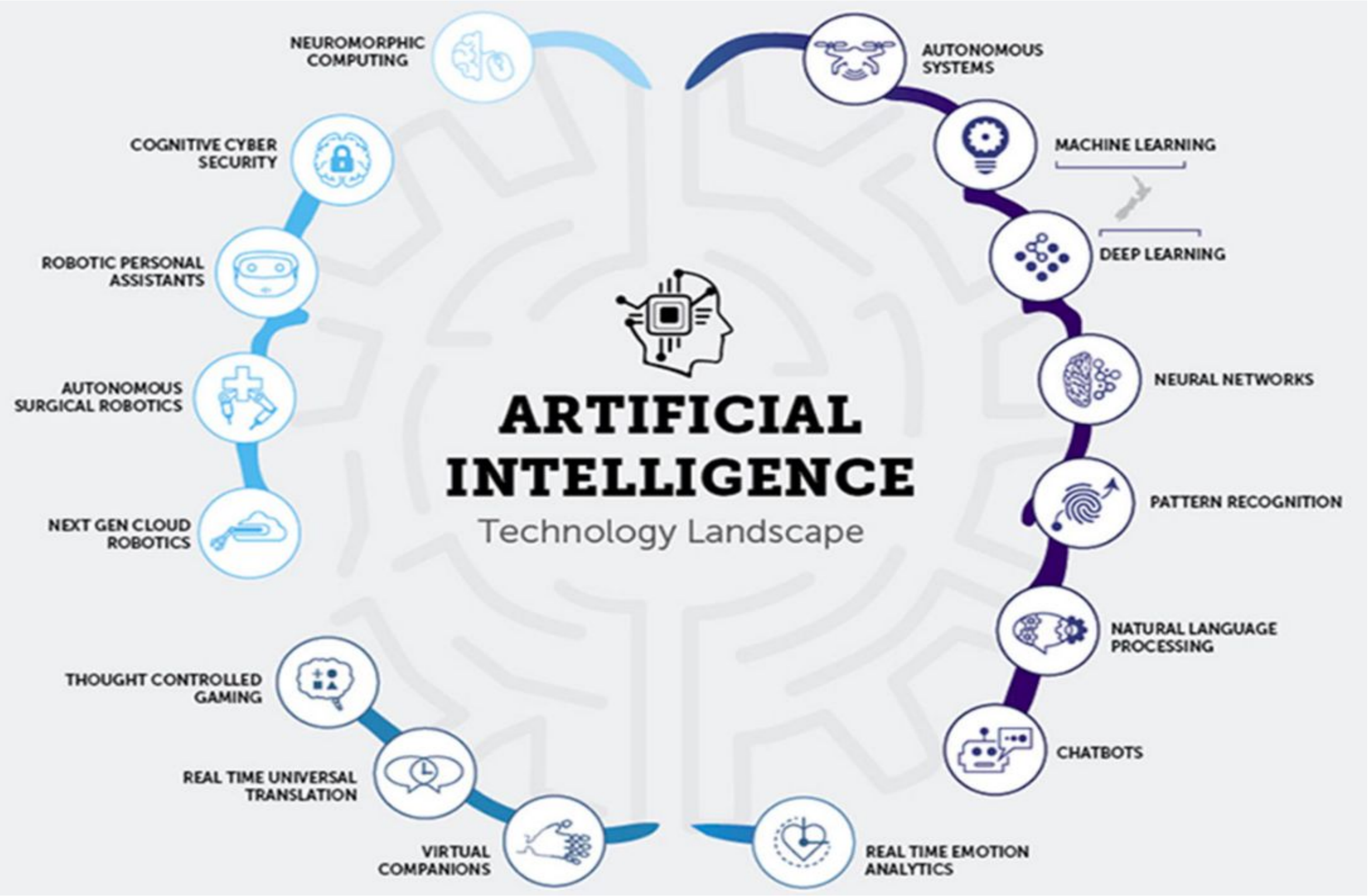
AI: Artificial Intelligence

DL: Deep Learning

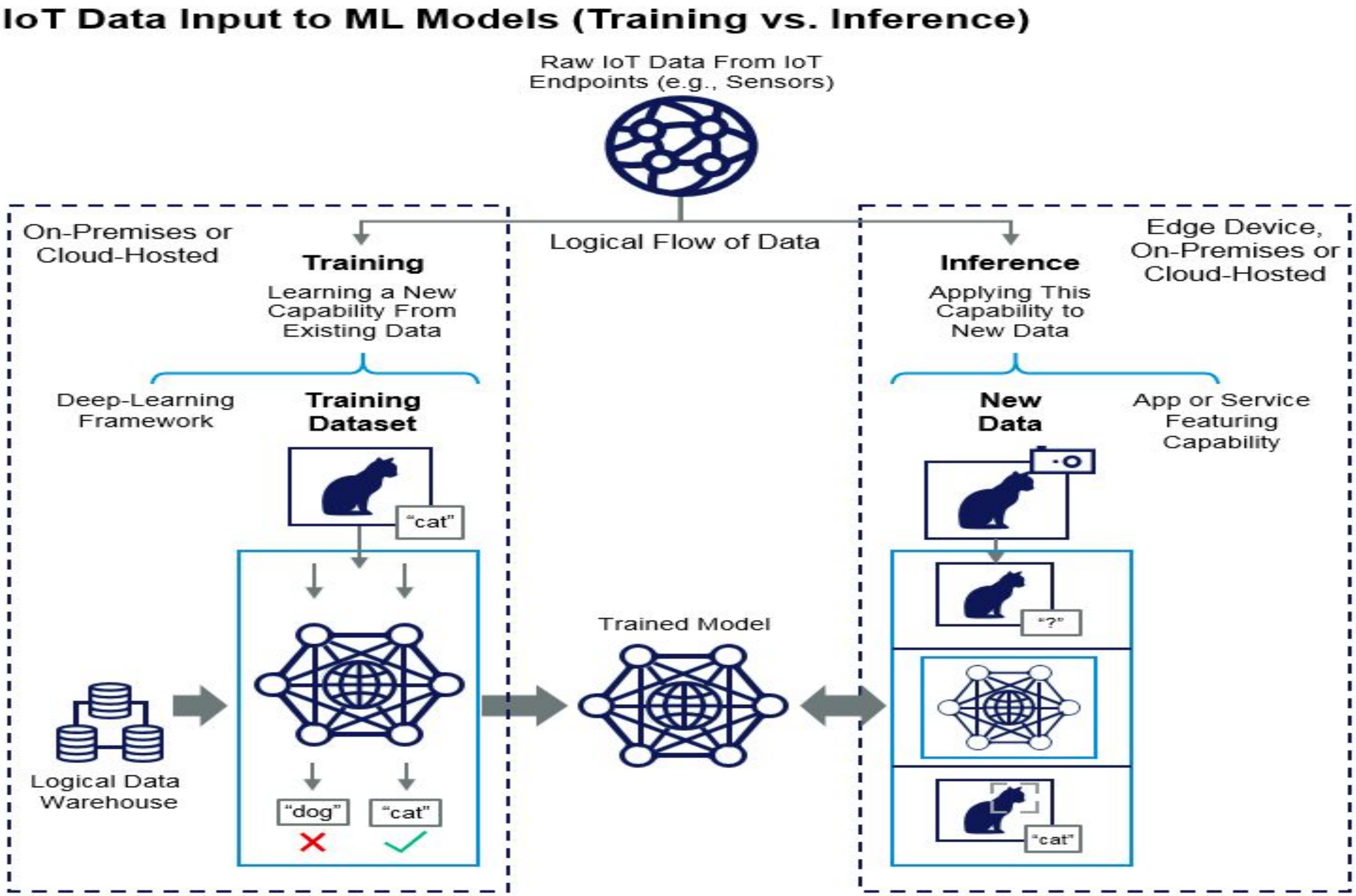
ML: Machine Learning



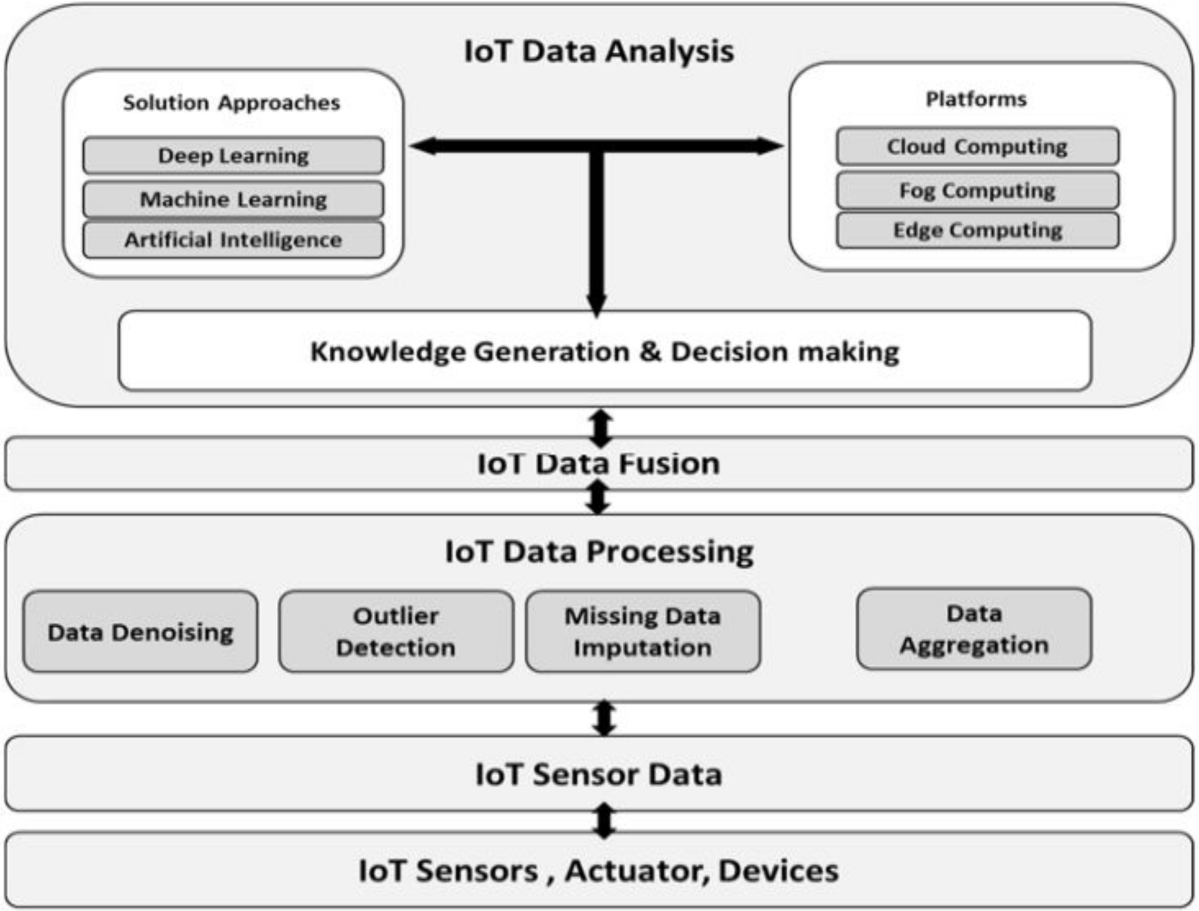
Source: CadenceLive 2021

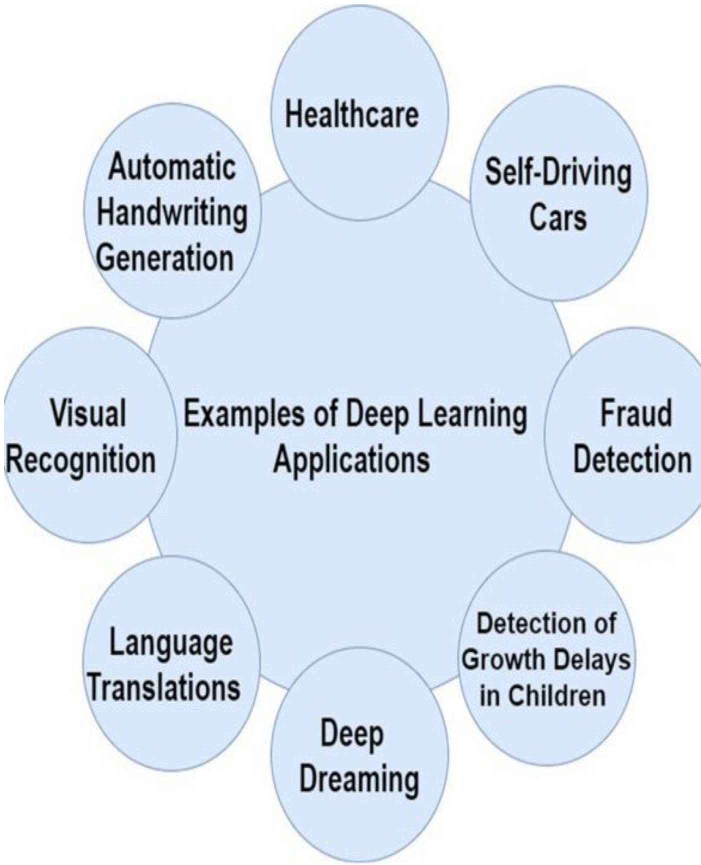

NEUROMORPHIC COMPUTING
COGNITIVE CYBER SECURITY
ROBOTIC PERSONAL ASSISTANTS
AUTONOMOUS SURGICAL ROBOTICS
NEXT GEN CLOUD ROBOTICS
THOUGHT CONTROLLED GAMING
REAL TIME UNIVERSAL TRANSLATION
VIRTUAL COMPANIONS
ARTIFICIAL INTELLIGENCE
Technology Landscape
AUTONOMOUS SYSTEMS
MACHINE LEARNING
DEEP LEARNING
NEURAL NETWORKS
PATTERN RECOGNITION
NATURAL LANGUAGE PROCESSING
CHATBOTS
REAL TIME EMOTION ANALYTICS

AI Training & Inference

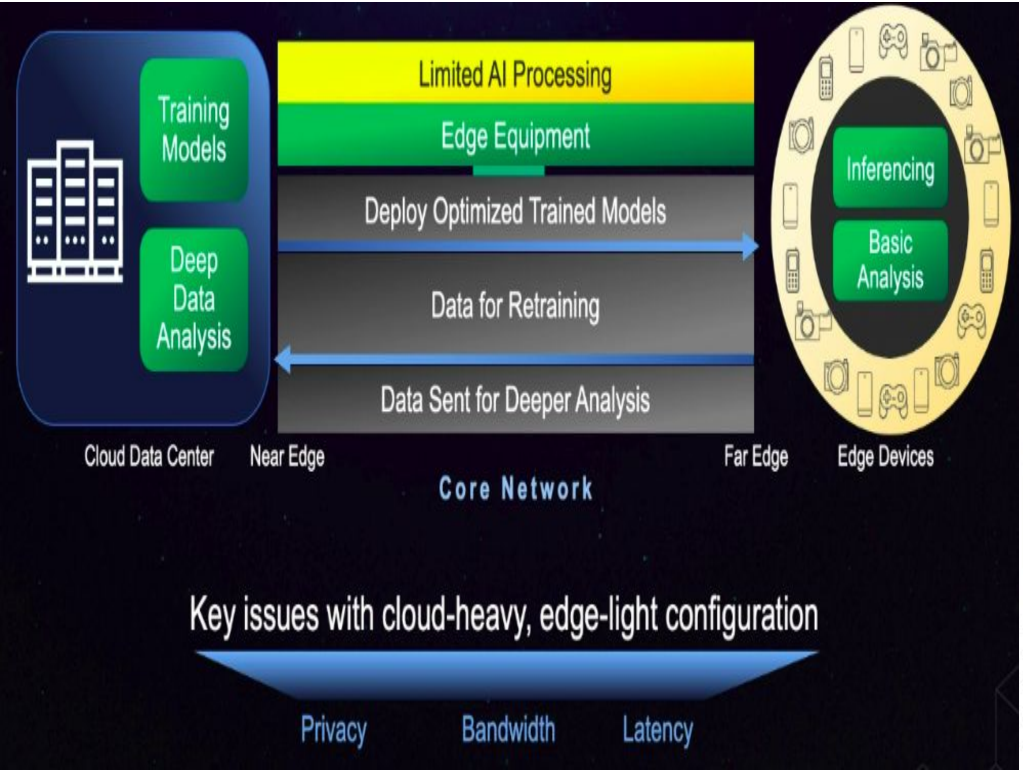


AI use case: IoT Framework

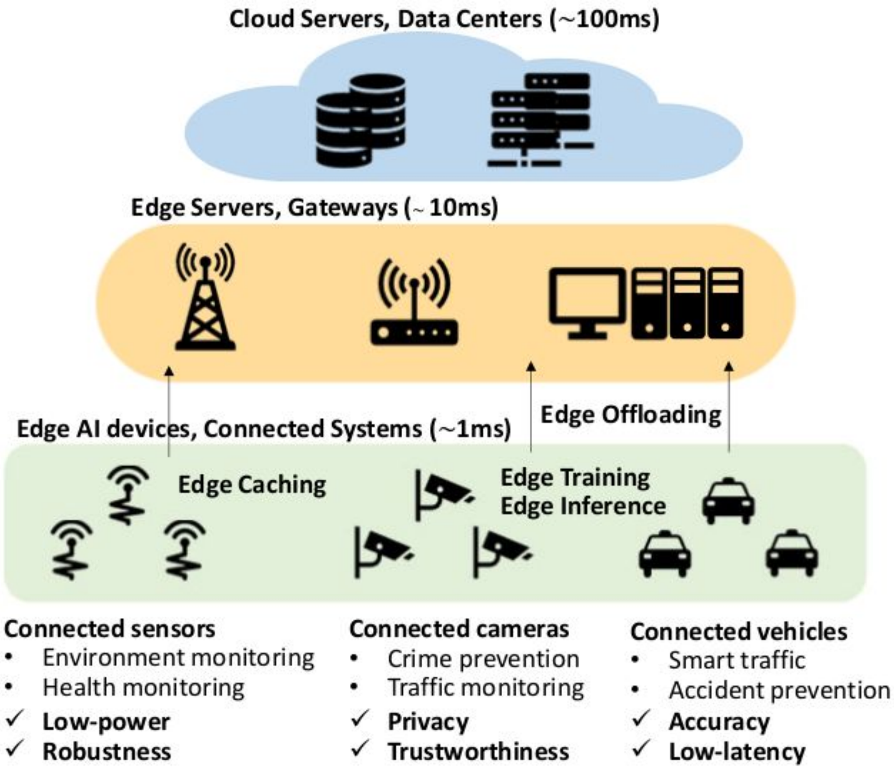




AI Training & Inference at Different Levels



Source: CadenceLive 2021



Adapted from Hao et al., “Enabling Design Methodologies and Future Trends for Edge AI: Specialization and Co-design”, IEEE Design & Test, 2021

Deep Learning Landscape

TensorFlow

Caffe

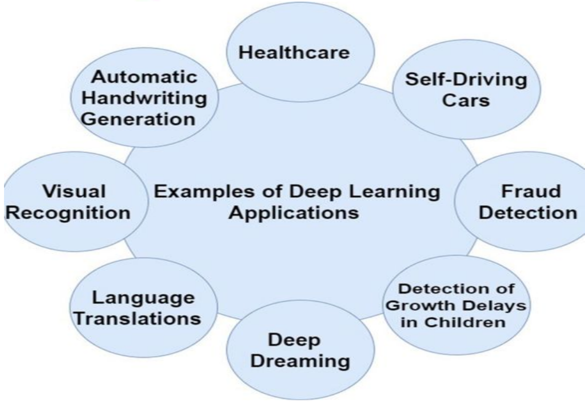
cuDNN

intel Software

MLSL

NVIDIA

TensorRT

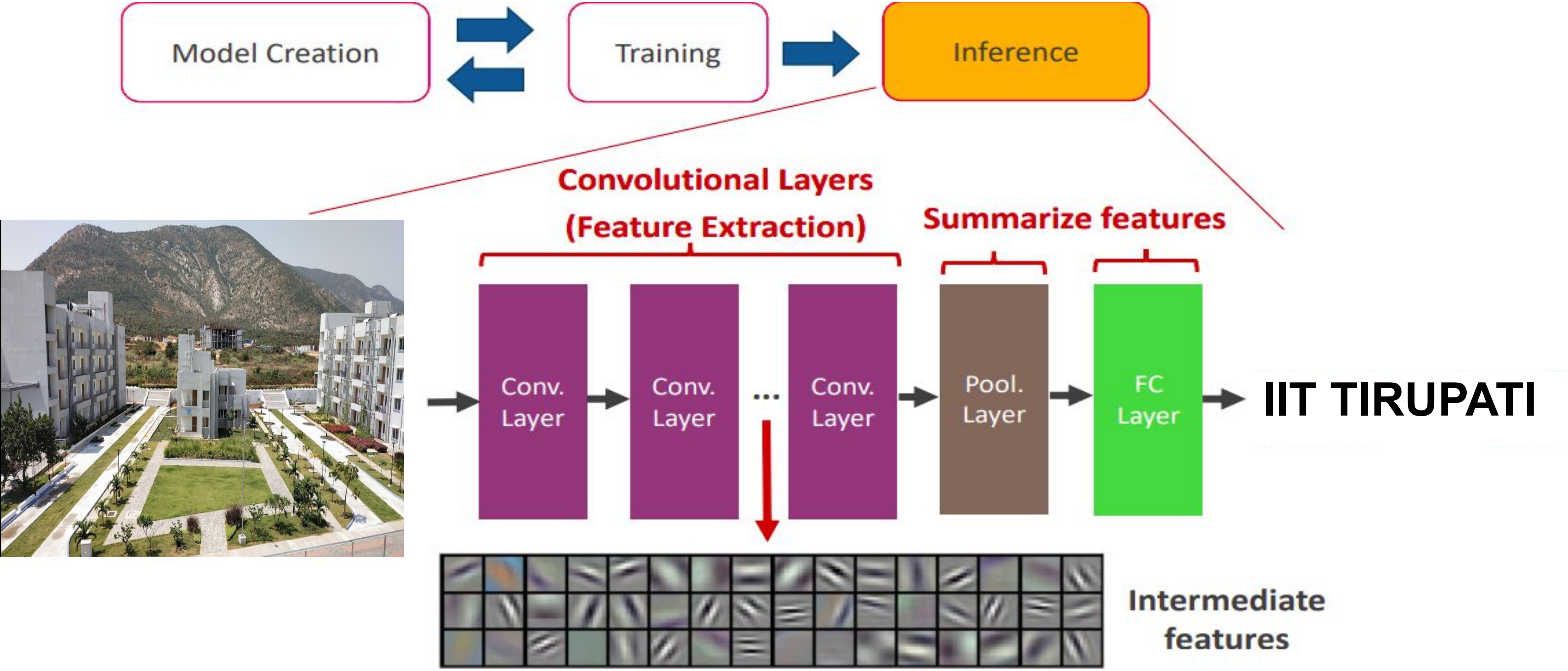


Design Tools

Most of these libraries are open-source and available in Python

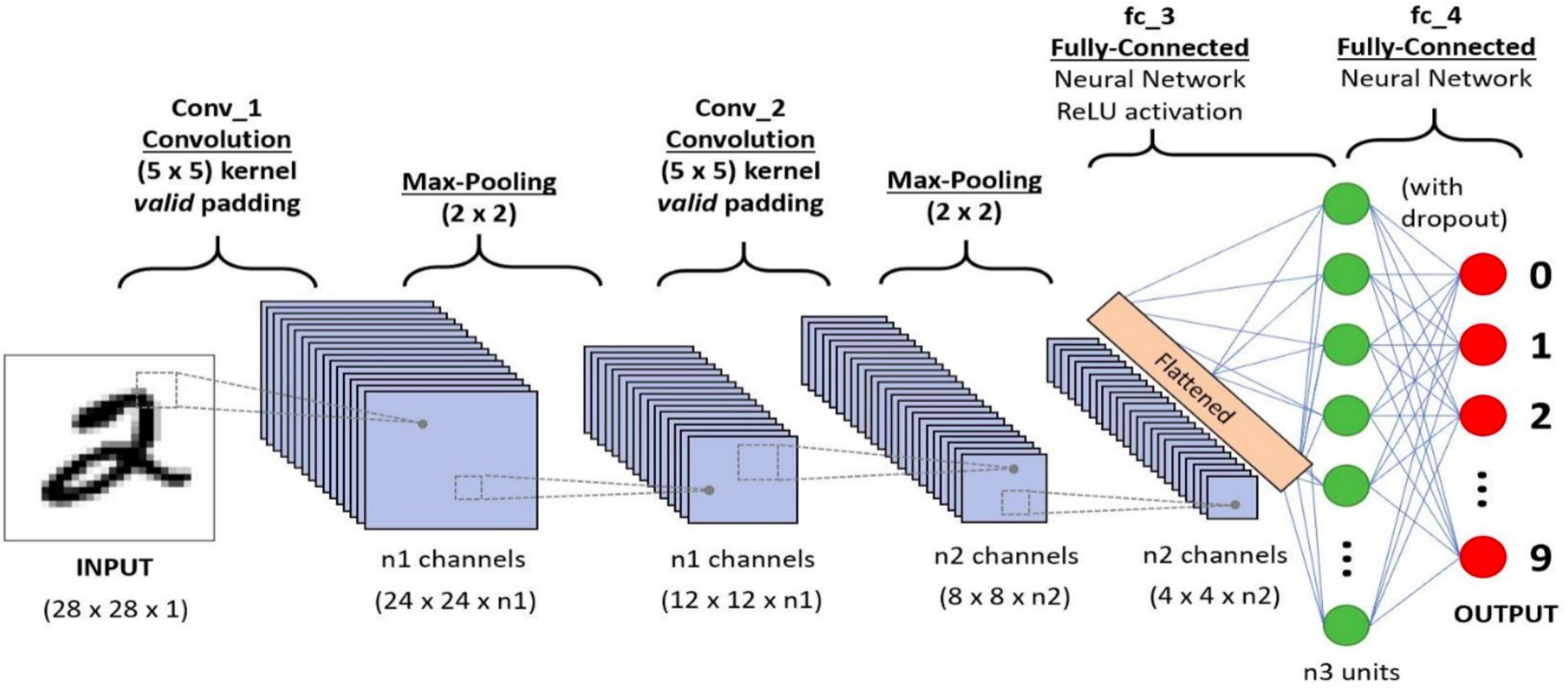
Adapted from Scale-sim Tutorial slides (ASPLOS 2021)

AI Task Example: Image Recognition



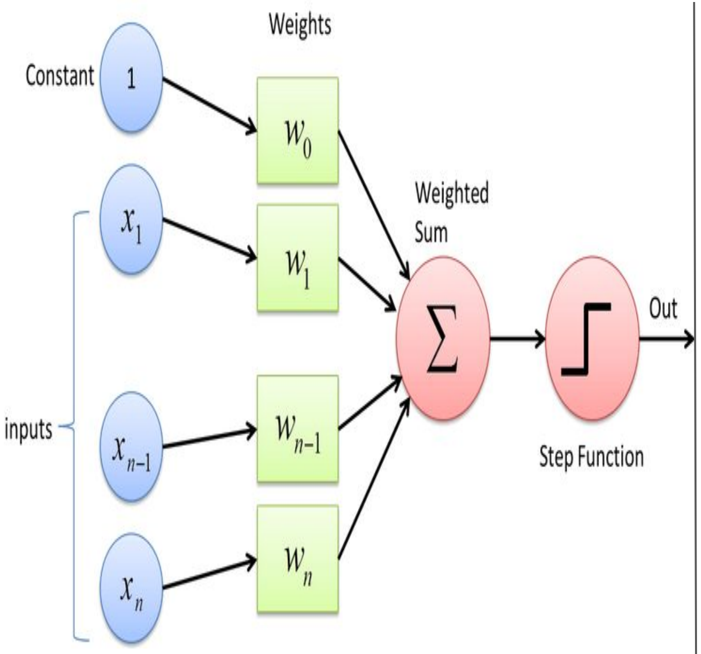
Adapted from Scale-sim Tutorial slides (ASPLOS 2021)

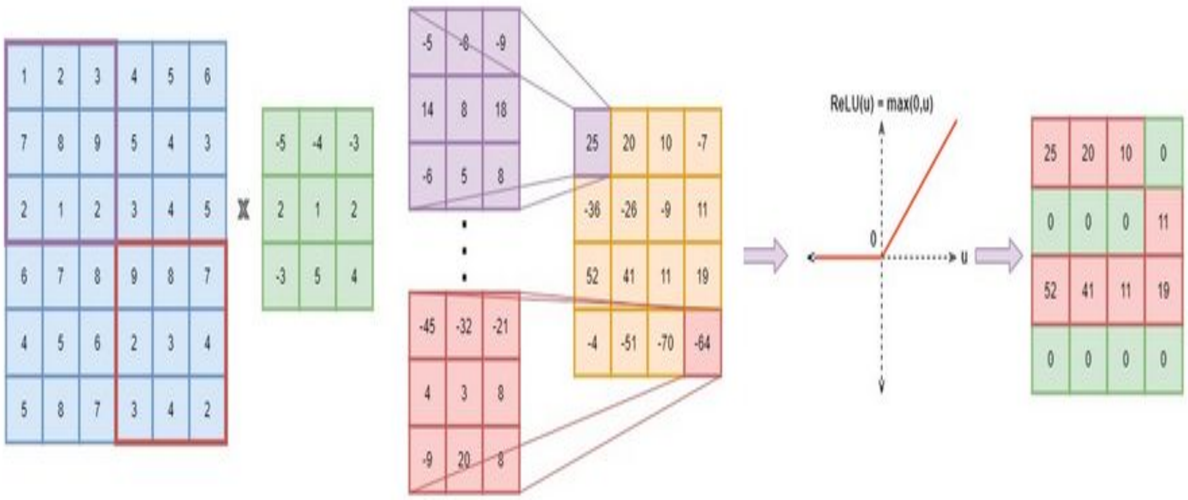
Convolution Neural Network (CNN)



https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

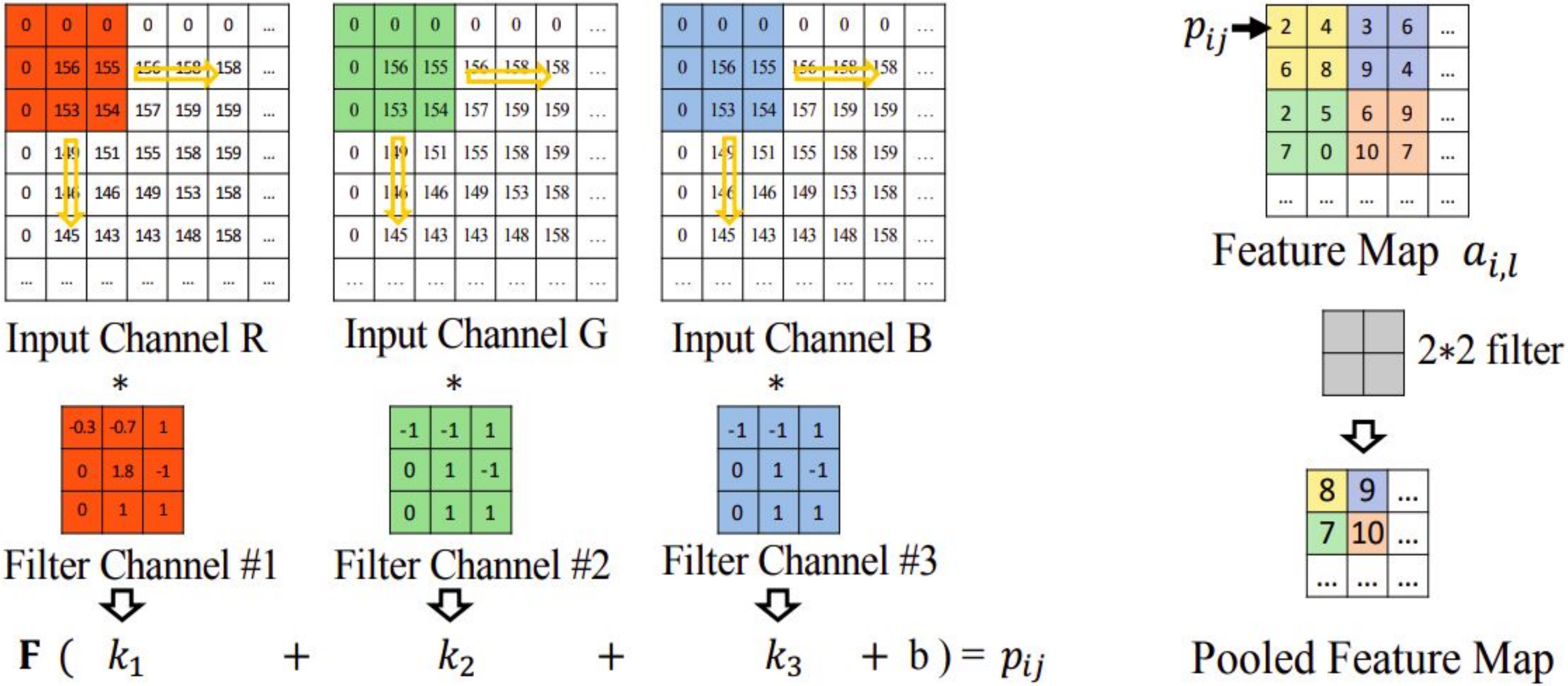
Understanding Convolution Neural Network Opertaions



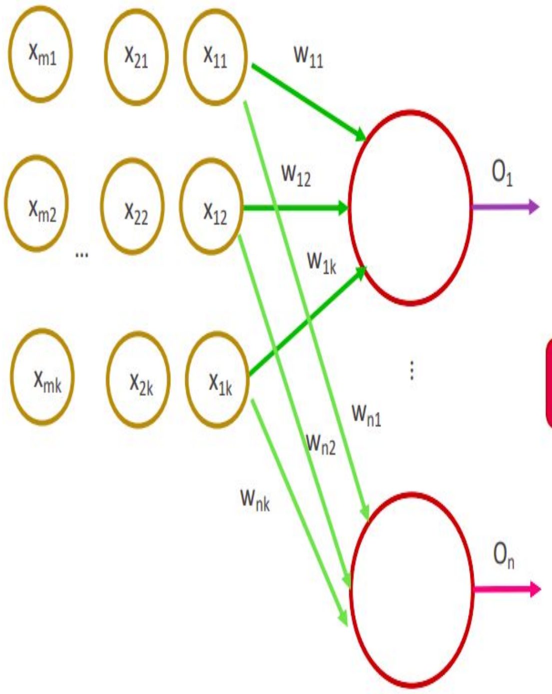


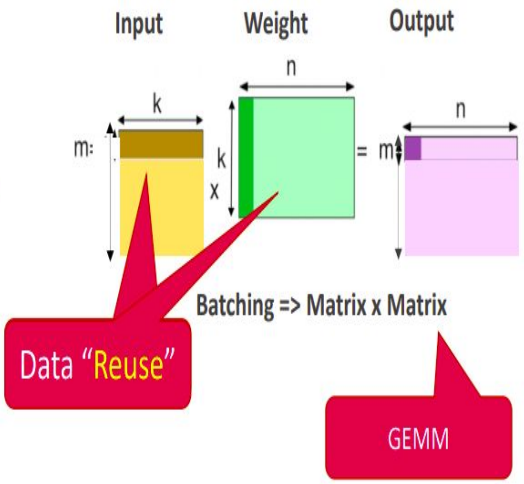
A large number of computations are involved!

Understanding Deep Learning Operations

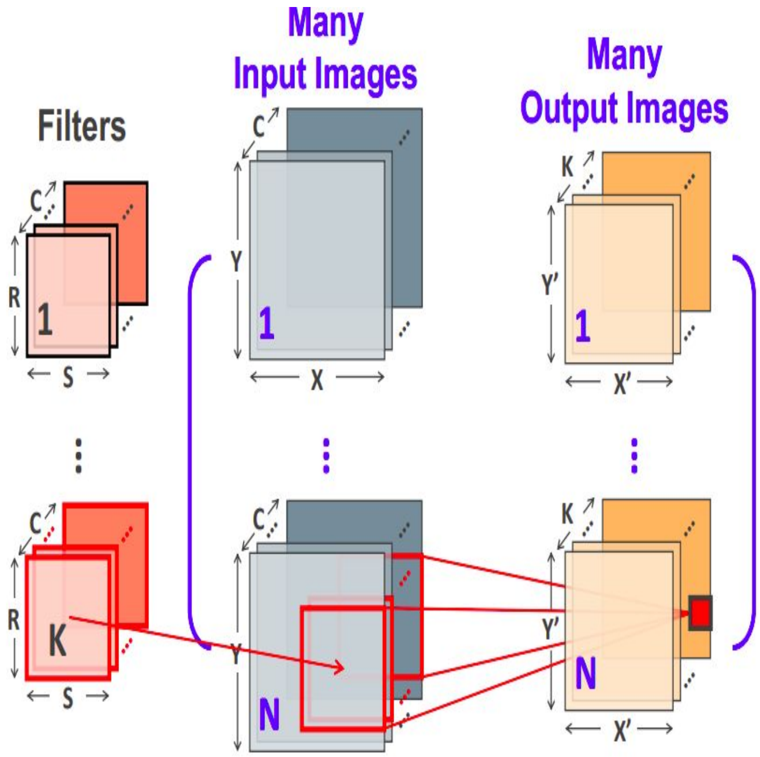


Computations in a CNN: Linear Algebra





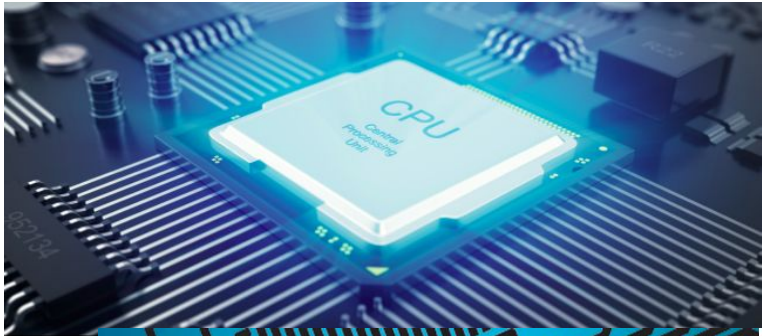
GEMM: GEneral Matrix Multiply

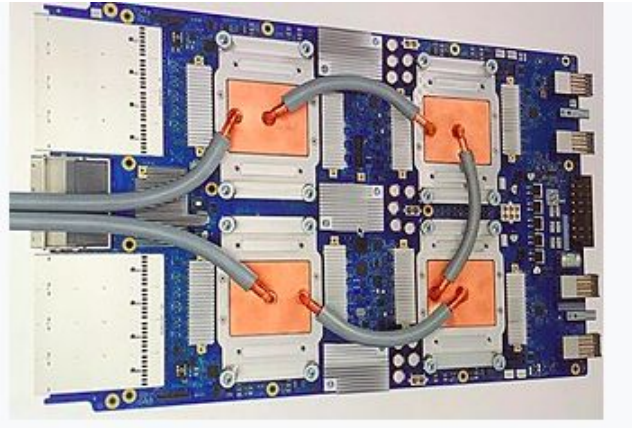


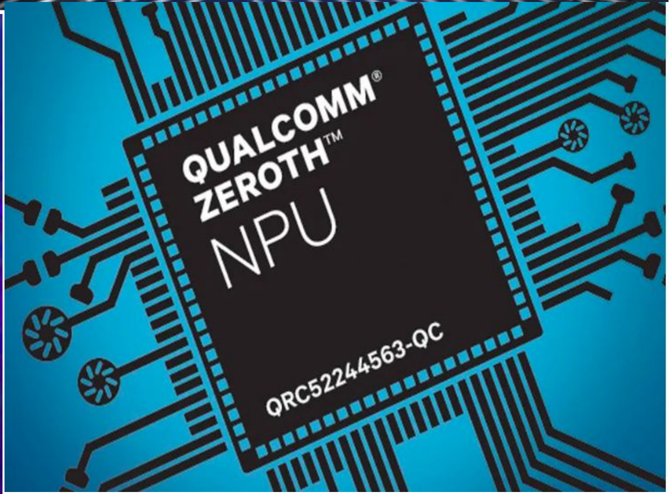
Adapted from Scale-sim Tutorial slides (ASPLOS 2021)

Different Hardware Platforms

1. *CPU (Central Processing Units)*
2. *GPU (Graphics Processing Units)*
3. *TPU (Tensor Processing Units)*
4. *NPU (Neural Processing Units)*
5. *FPGAs*







Comparative Evaluation of Different Devices

	CPU-based	GPU-based	FPGA-based	ASIC-based
Advantages	Good versatility Lowest price Multitasking High programmability	Medium versatility Massive parallelism Moderate programmability	Customized designs Low latency High performance/watt	Extremely low power Highest performance
Limitations	Limited parallelism	Power hungry	Limited on-chip memory Requires design expertise	High development cost Long Time-To-Market Low flexibility
Example Devices	Arm Cortex-M Series Raspberry Pi Series NanoPi Series Sipeed MAIX Series	Nvidia Jetson Series AMD Ryzen Family Arm Mali GPUs	Xilinx Zynq FPGAs Intel Arria 10 FPGAs Lattice iCE40 FPGAs	Google Edge TPU Ascend 310 processor In-memory chips Neuromorphic chips
Development Tools	Arm NN TensorFlow Lite	TensorRT Intel OpenVino	Intel OpenVino Xilinx Edge AI platform	Apache TVM

Adapted from Hao et al., “Enabling Design Methodologies and Future Trends for Edge AI: Specialization and Co-design”, IEEE Design & Test, 2021

Computation Challenges: DL Hardware

AlexNet (ILSVRC 2012 Winner)

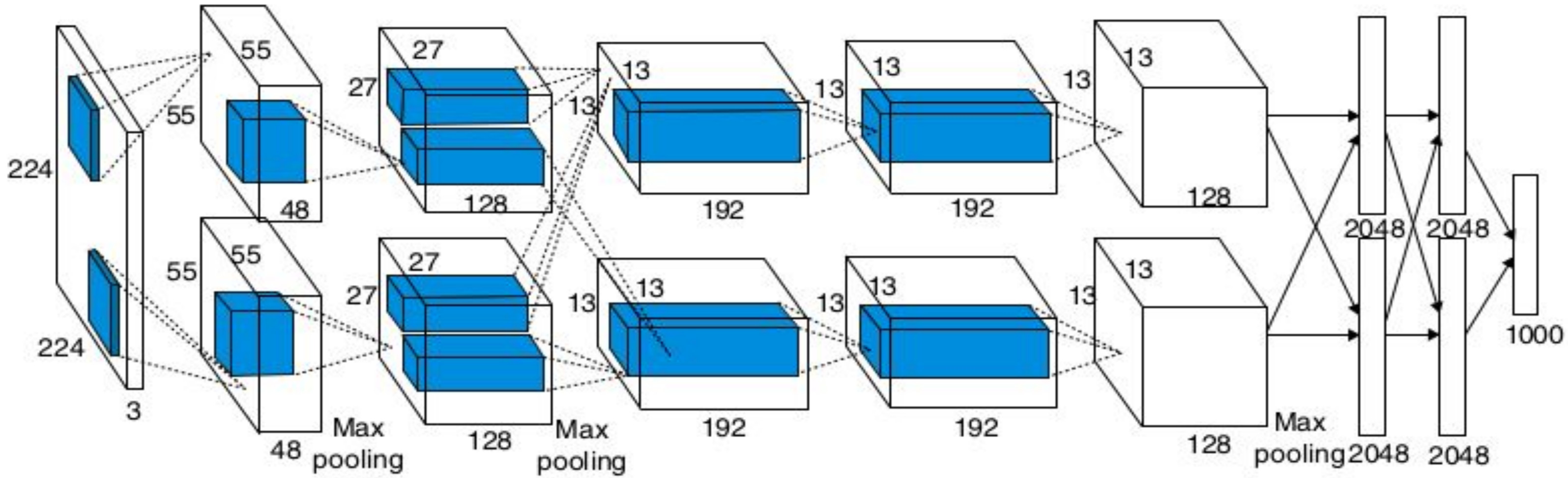
Deep Network

Massive parallel processing

Reconfigurable network

Memory bottleneck

Big input data



Intensive computations

Network pruning

Data sparsity

Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Network, NIPS 2012

Challenges with DNN Computations

- **Millions of Parameters (i.e., weights)**
  - Billions of computations ⇨ **Need lots of parallel computations**

DNN Topology	Number of Weights
AlexNet (2012)	3.98M
VGGnet-16 (2014)	28.25M
GoogleNet (2015)	6.77M
Resnet-50 (2016)	23M
DLRM (2019)	540M
Megatron (2019)	8.3B

This makes CPUs inefficient

PE: Processing Element

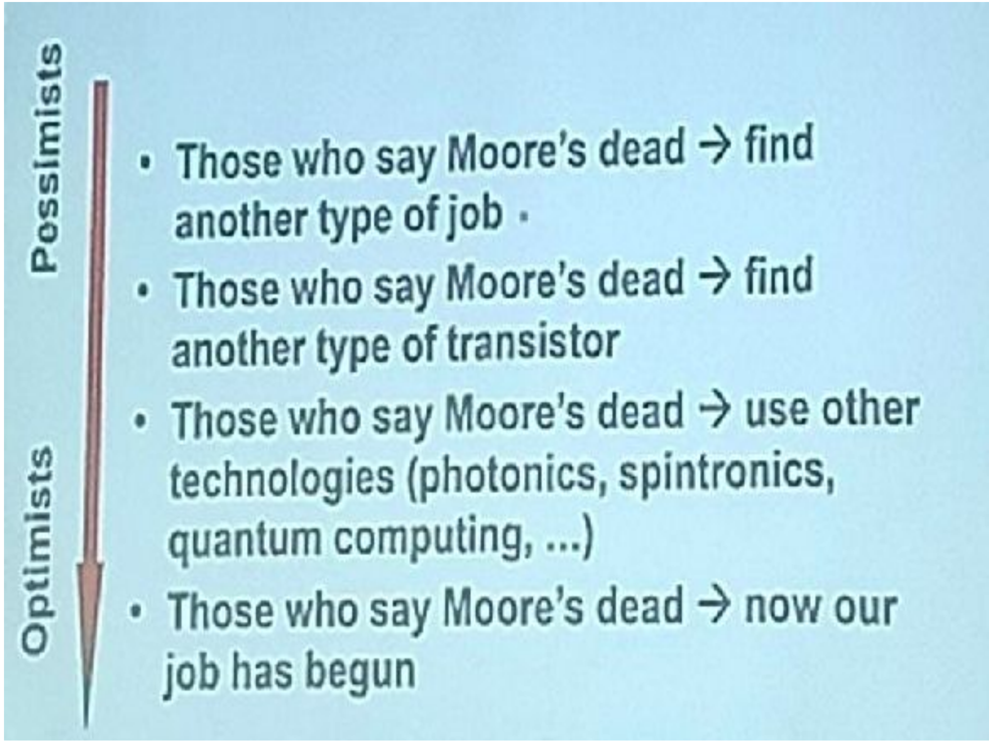
RF: Register File

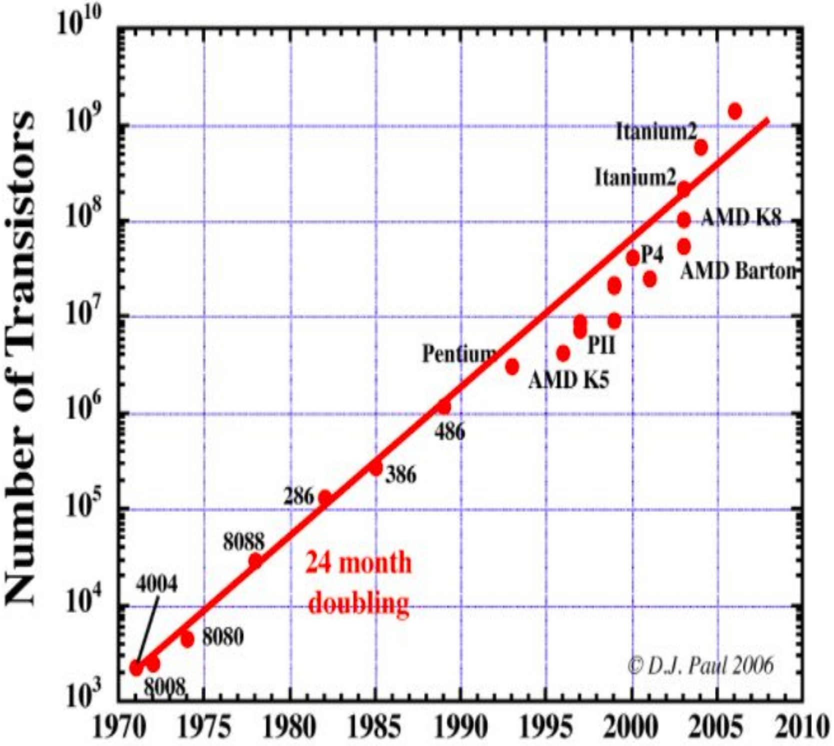
DRAM: Dynamic Random Access Memory

  - Heavy data movement ⇨ **Need to reduce energy**

This makes GPUs inefficient

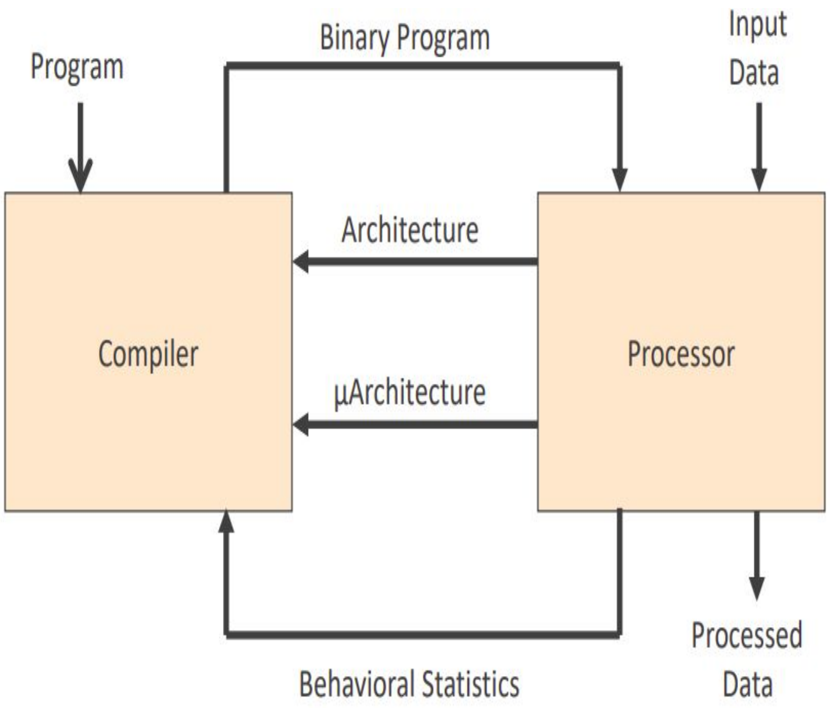
Technology challenges: There are 4 types of people!

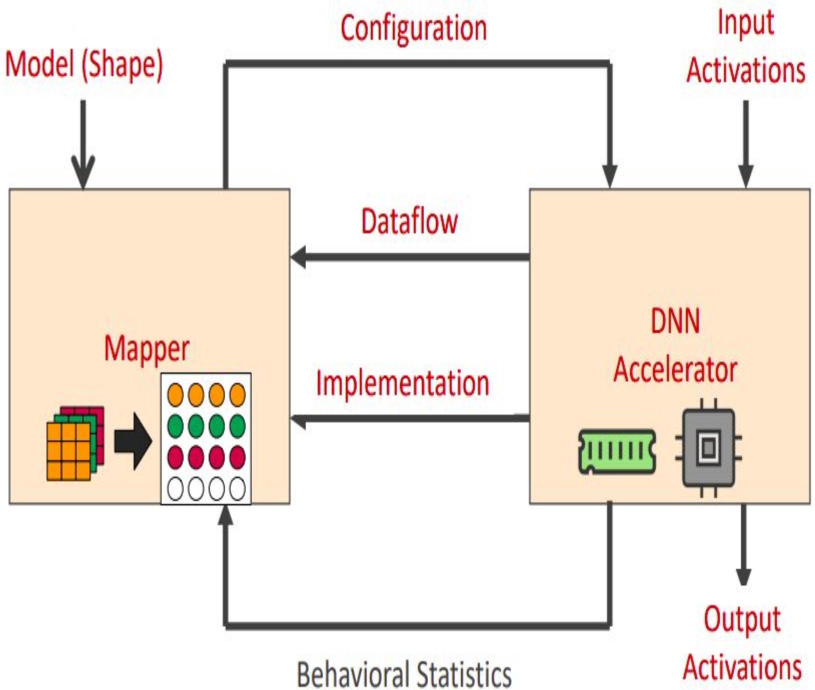




Ref: Prof. B. Razavi's talk

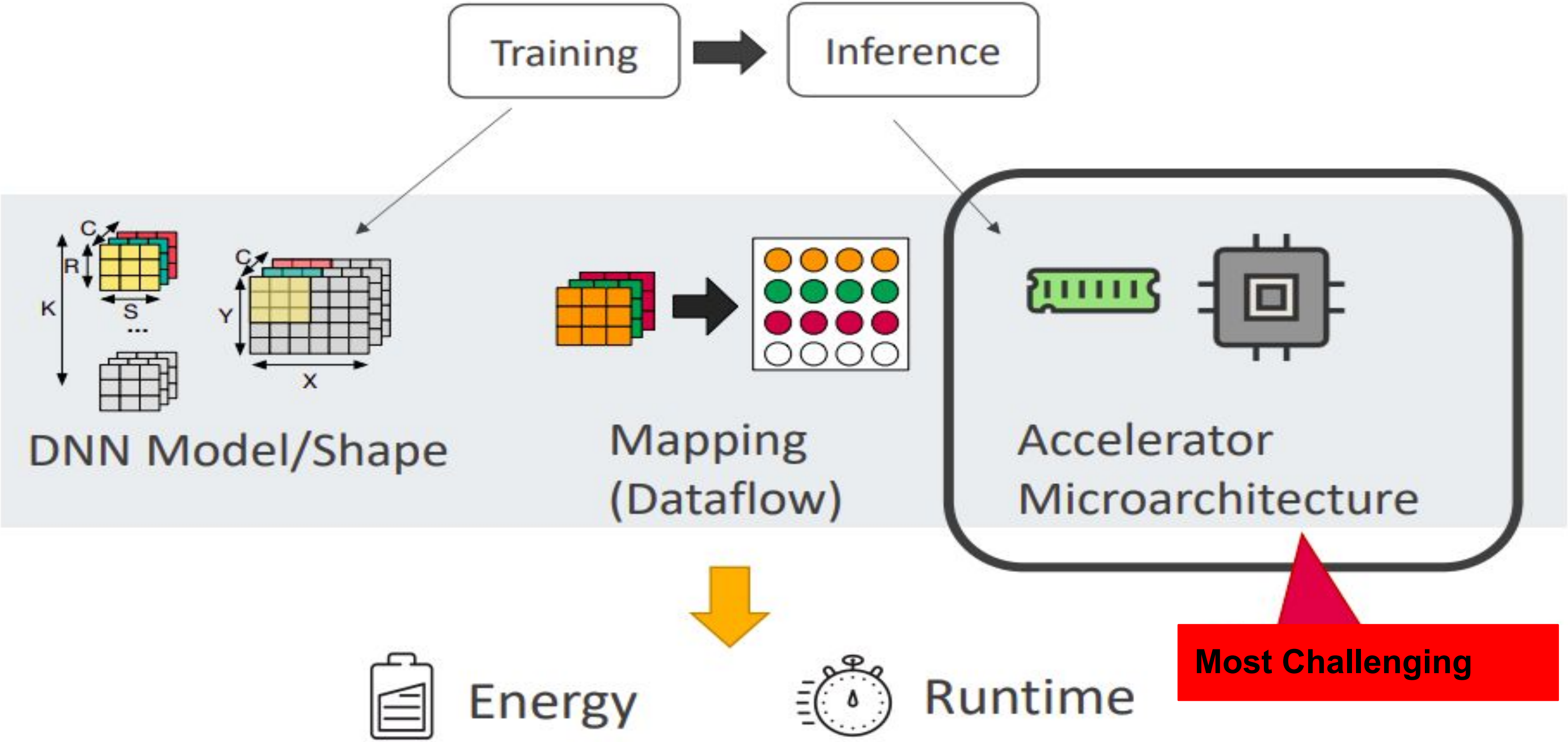
Compute Architectures: CPU v/s Accelerator





Adapted from Eyeriss tutorial slides (ISCA 2019)

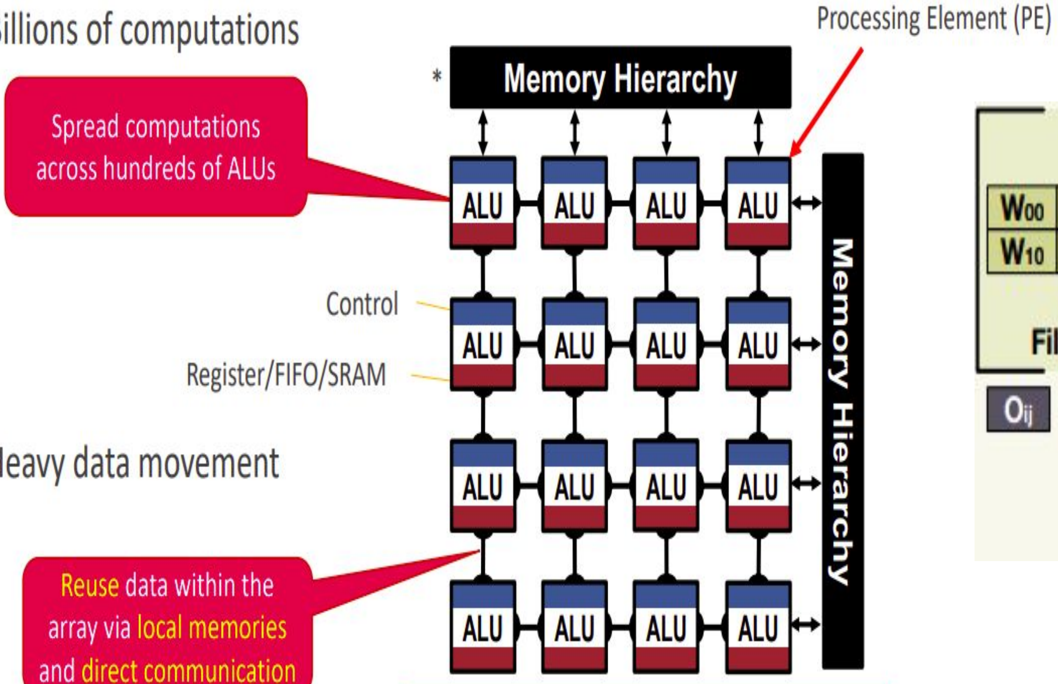
Challenges in AI Hardware Design & Deployment



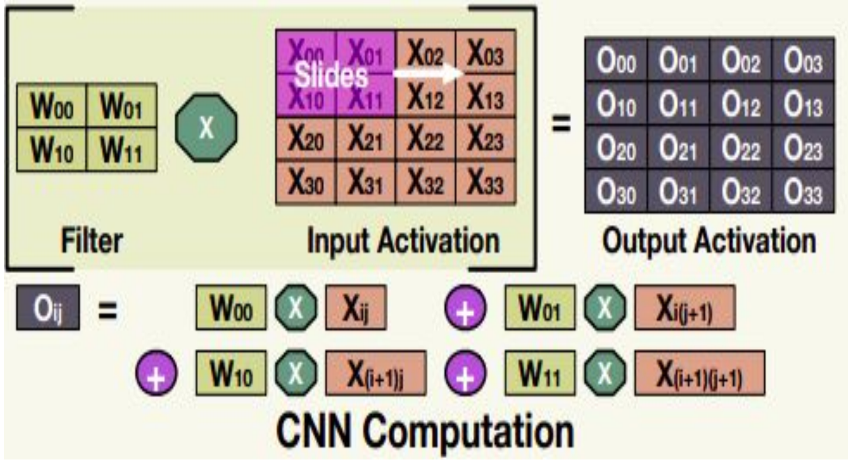
Adapted from Scale-sim Tutorial slides (ASPLOS 2021)

Customised Architectures (Accelerators)

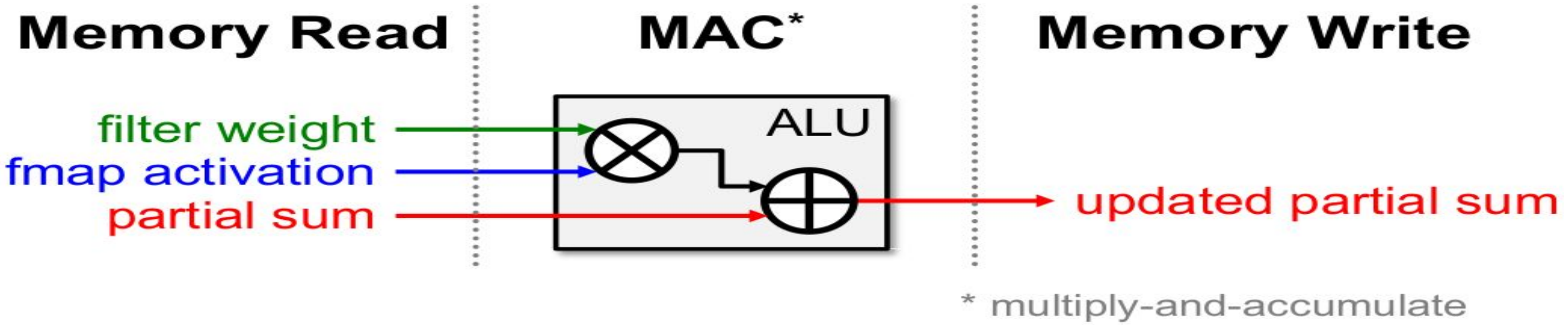
- Millions of Parameters (i.e., weights)
  - Billions of computations
  - Heavy data movement

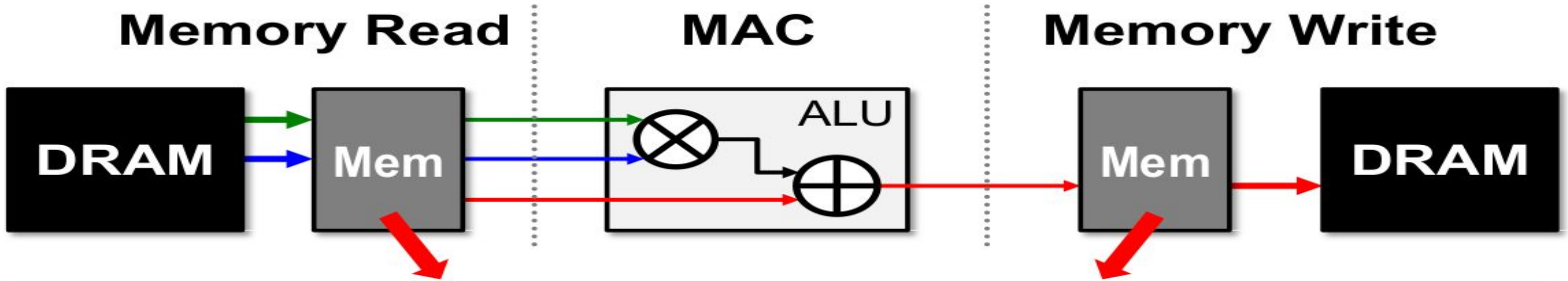


Examples: MIT Eyeriss, Google TPU, ...



Computations in DNN/CNN



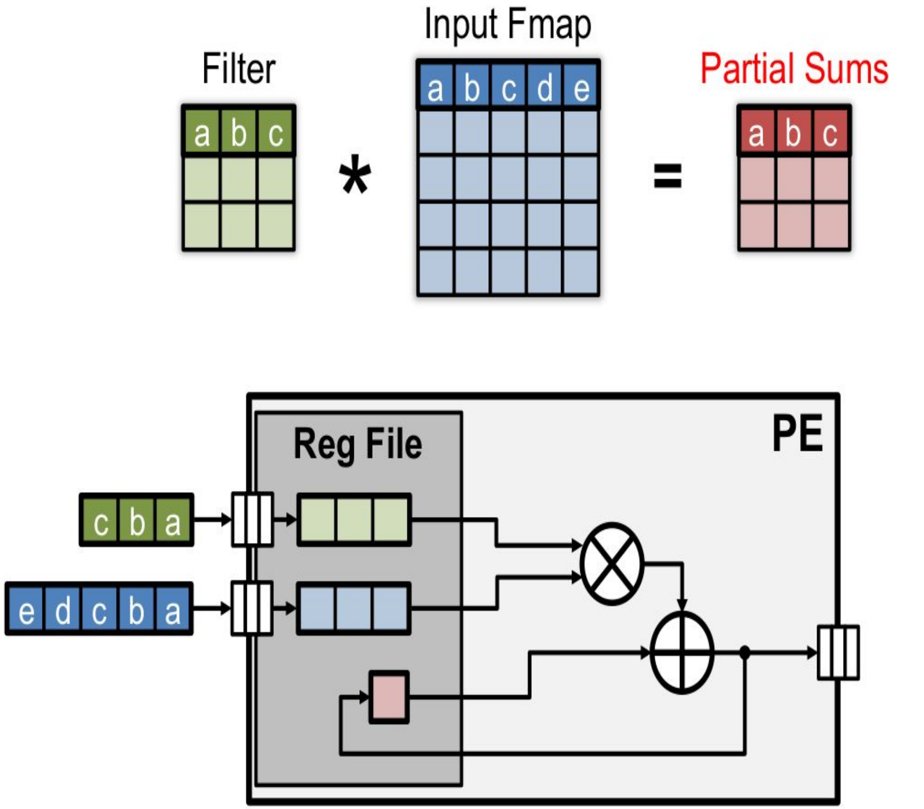


Extra levels of local memory hierarchy

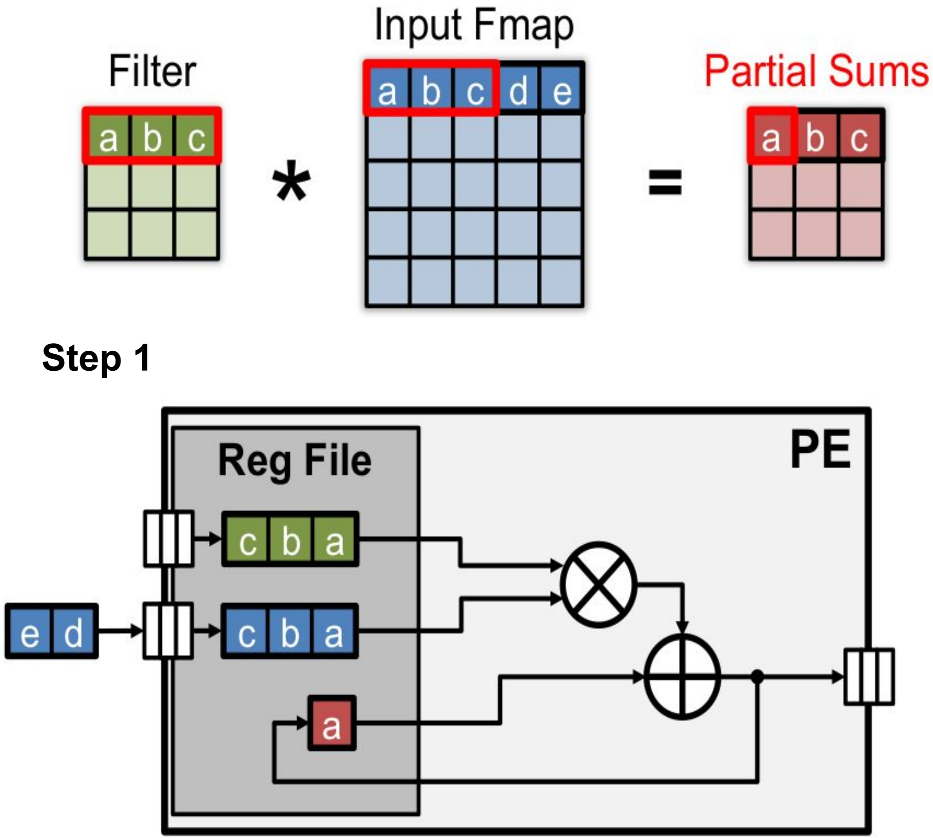
Smaller, but **Faster** and **more Energy-Efficient**

Adapted from Eyeriss tutorial slides (ISCA 2019)

Implementing Convolution

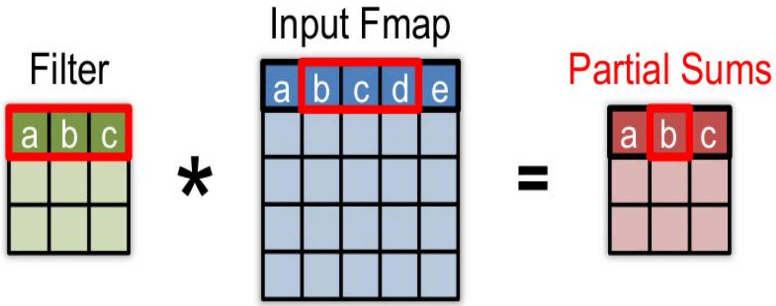


Step 1

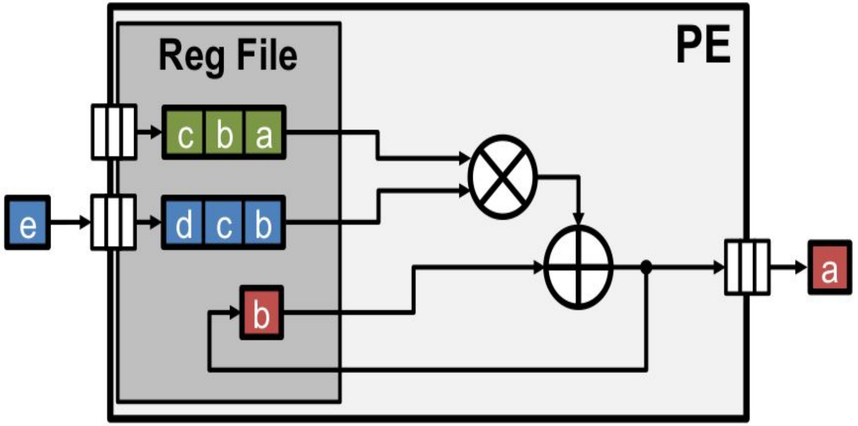


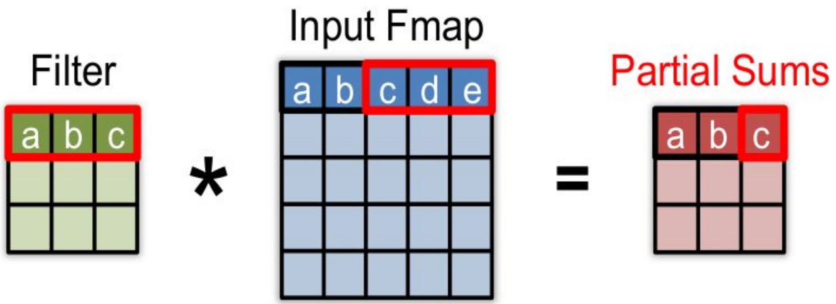
Adapted from Eyeriss tutorial slides (ISCA 2019)

Implementing Convolution

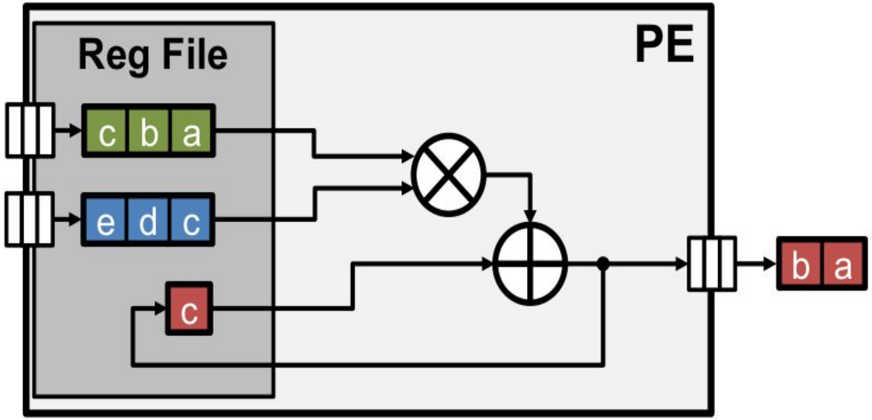


Step 2



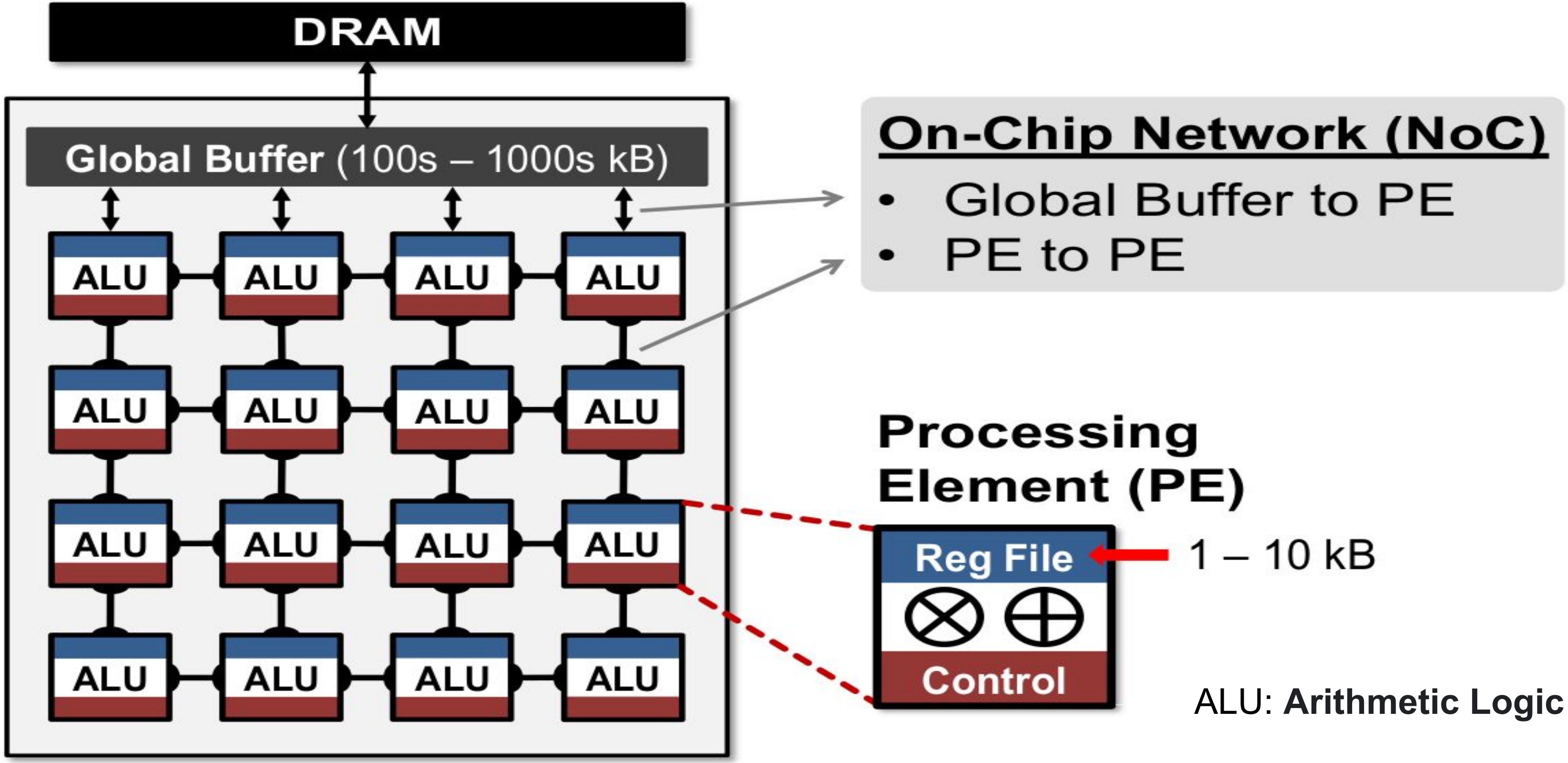


Step 3



Adapted from Eyeriss tutorial slides (ISCA 2019)

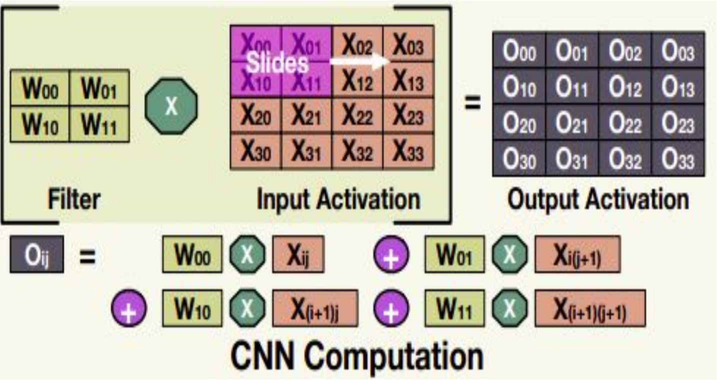
Accelerators for Computations in DNN/CNN

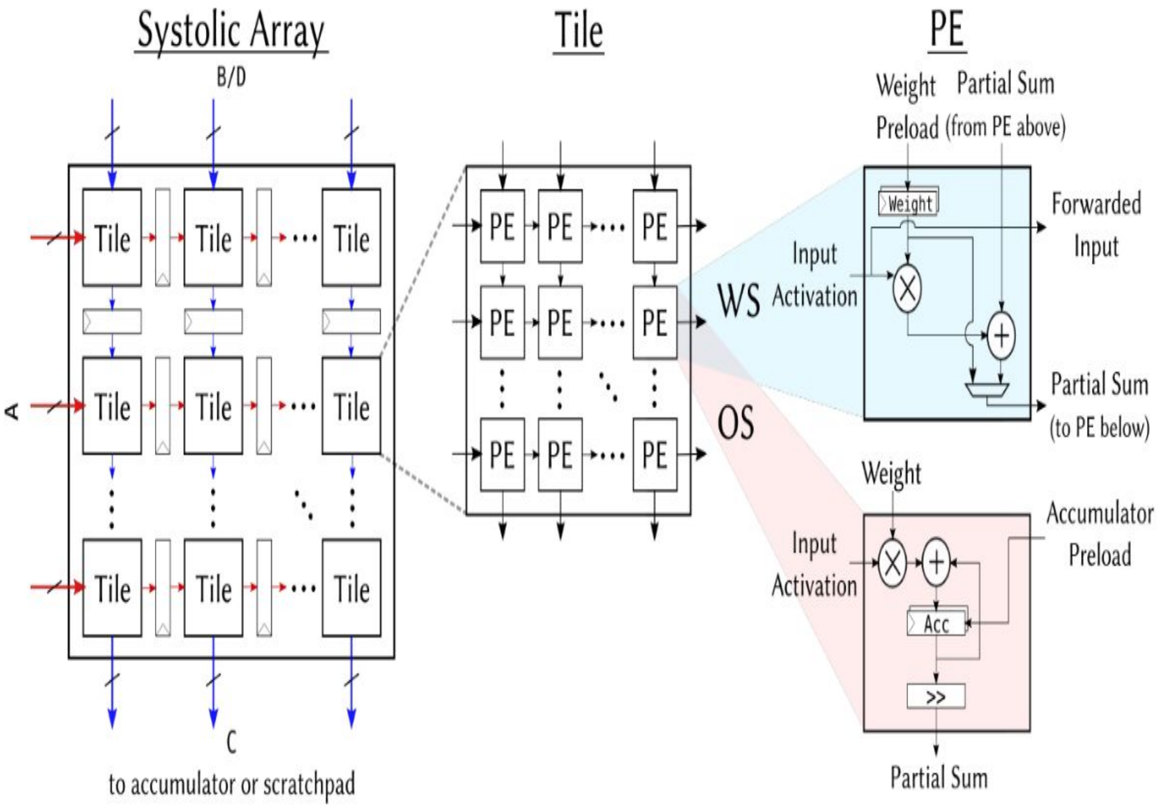


ALU: **Arithmetic Logic Unit**

Adapted from Eyeriss tutorial slides (ISCA 2019)

Different Dataflow Architectures

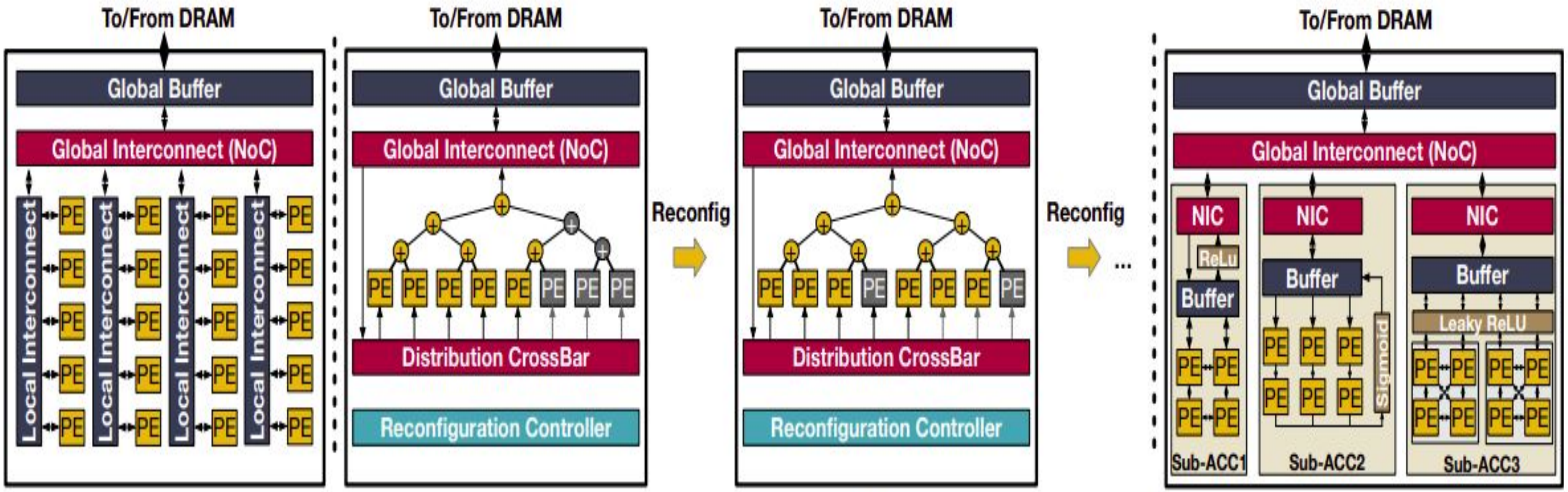




Each dataflow architecture has a set of benefits in terms of throughput and resource utilization.

Ref: USING DATAFLOW TO OPTIMIZE ENERGY EFFICIENCY OF DEEP NEURAL NETWORK ACCELERATORS, IEEE MICRO 2017

Different Dataflow Architectures

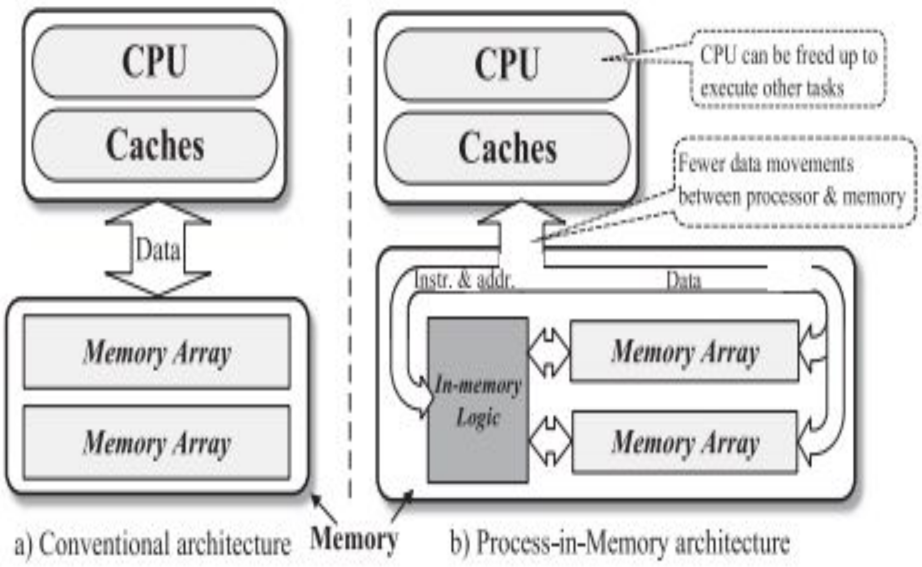


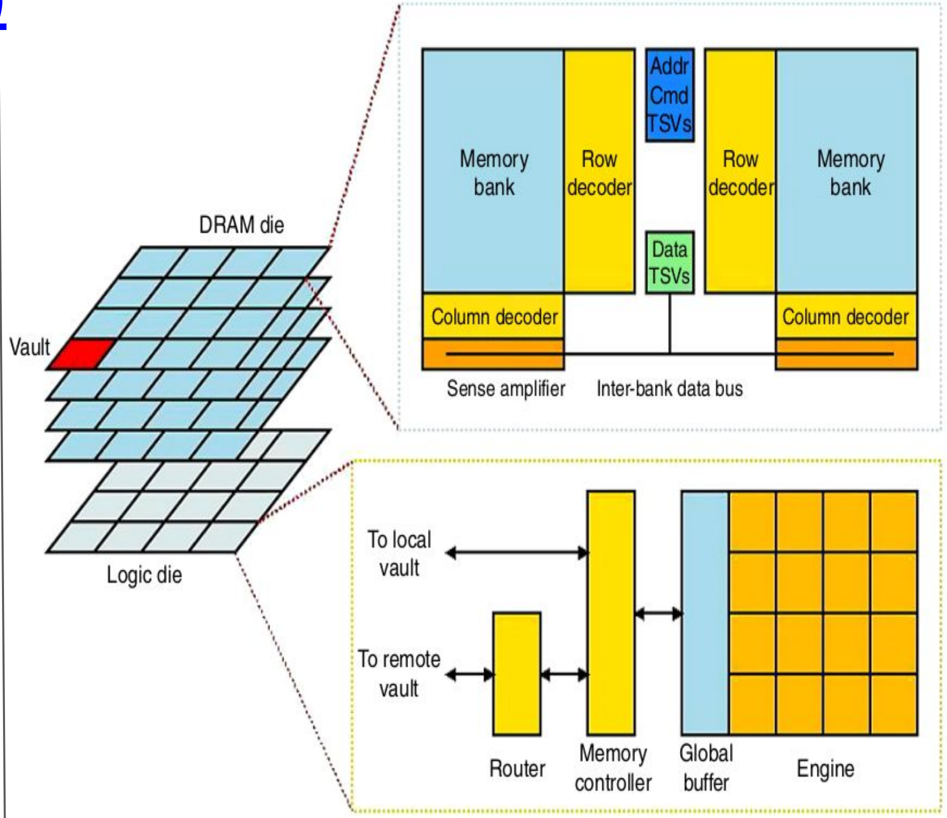
(a) Fixed Dataflow Accelerator (FDA) (b) Reconfigurable Dataflow Accelerator (RDA) (c) Heterogeneous Dataflow Accelerator (HDA)

Ref: Heterogeneous Dataflow Accelerators for Multi-DNN Workloads, HPCA 2021

Processing-In-Memory (PIM)

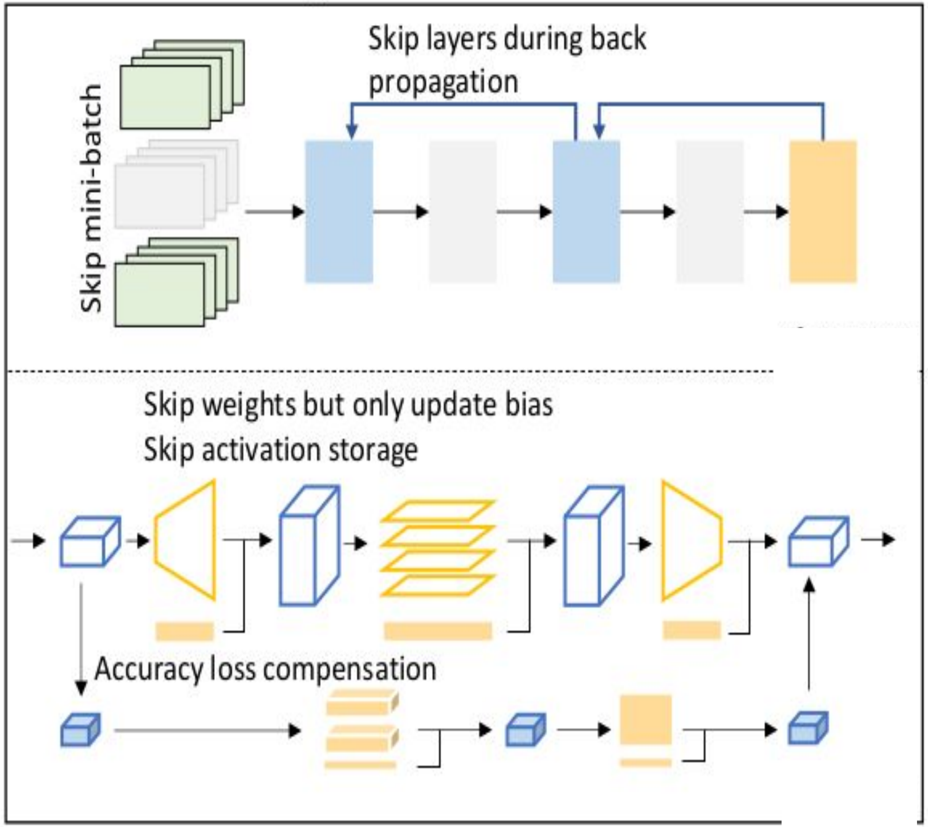
Novel architectural considerations:



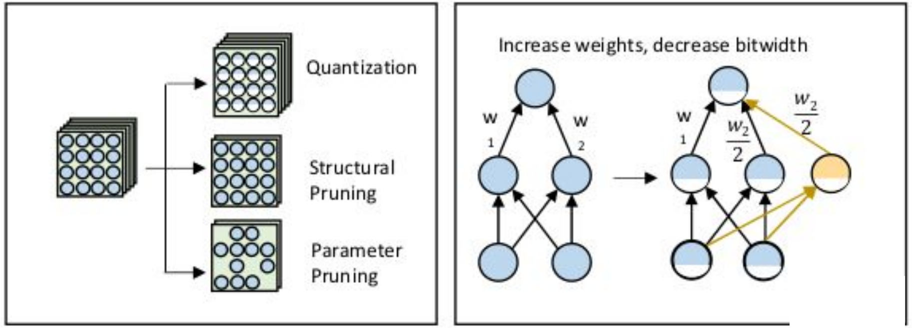


Specialized Techniques

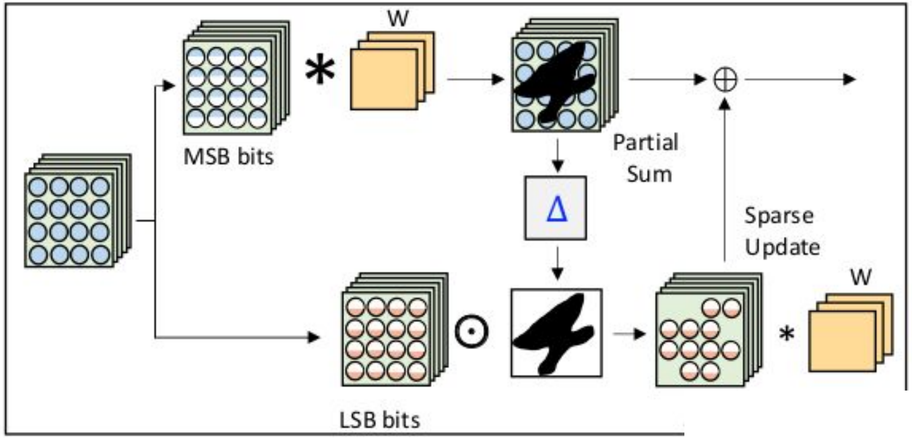
On-device Training



Model Compression

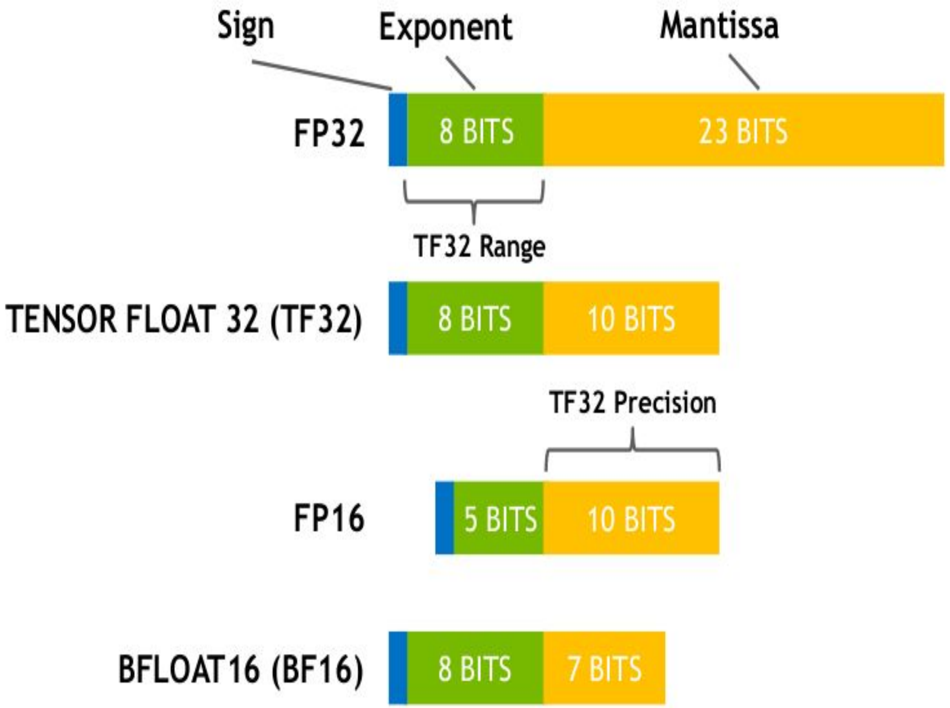


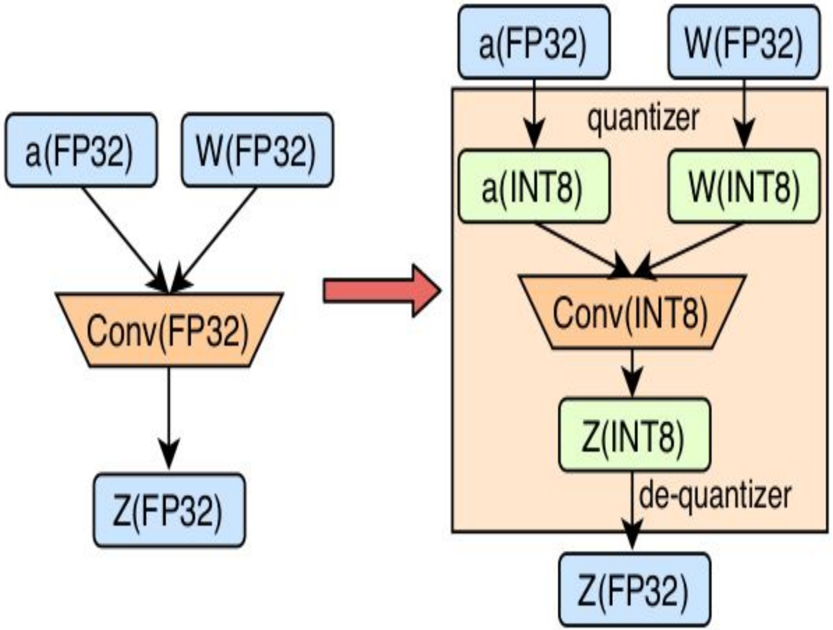
Adaptive Inference



Adapted from Hao et al., "Enabling Design Methodologies and Future Trends for Edge AI: Specialization and Co-design", IEEE Design & Test, 2021

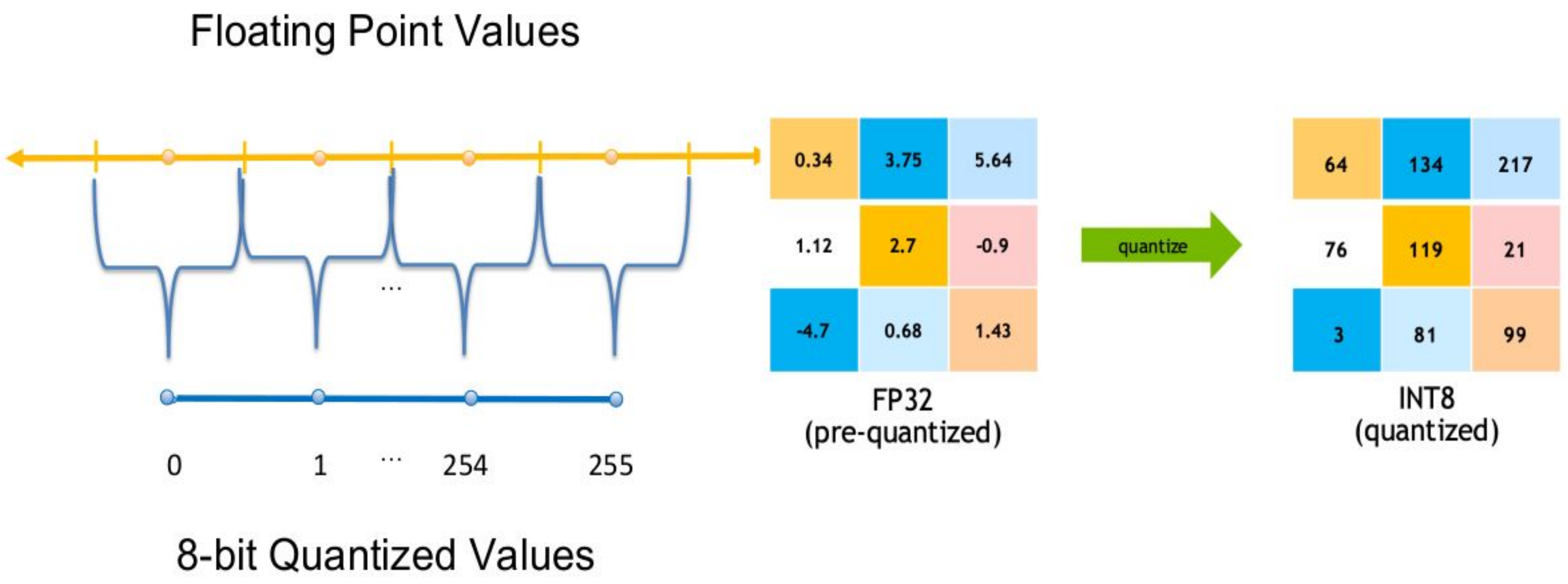
Neural Network Quantization





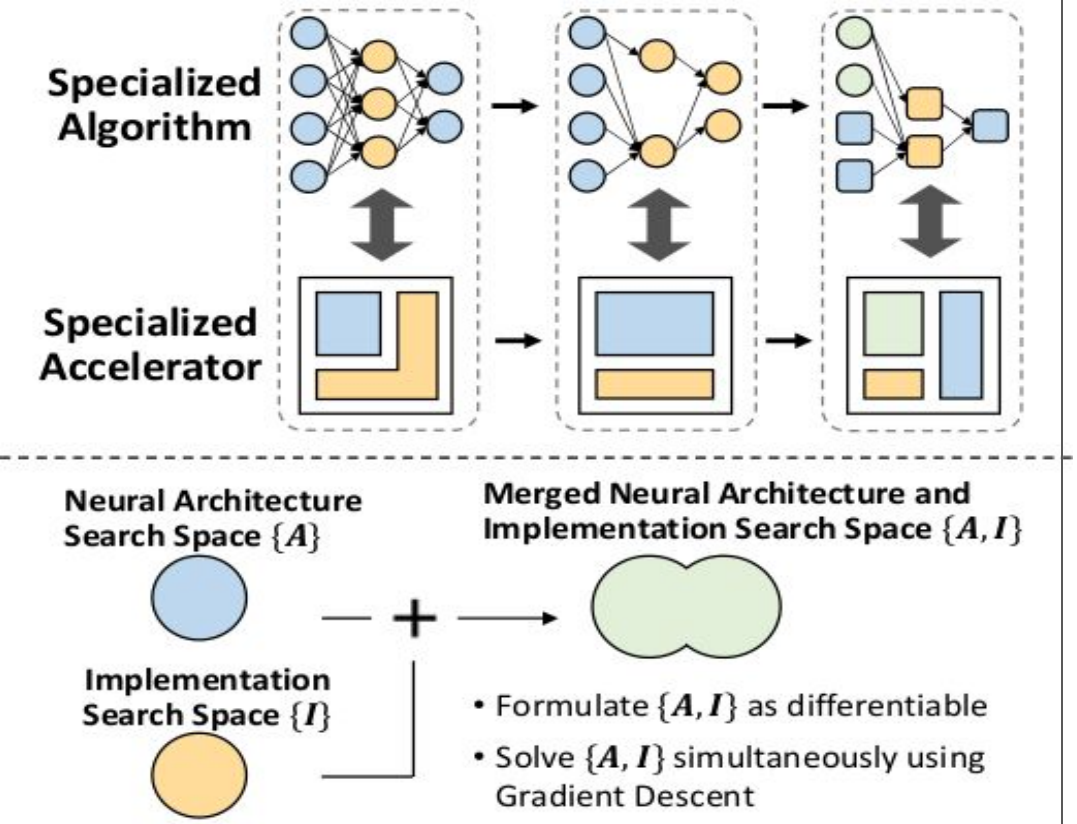
Conventional **With Quantization**

Quantization Example



How can we find the right trade-off for a given latency/power constraint for a given hardware platform?

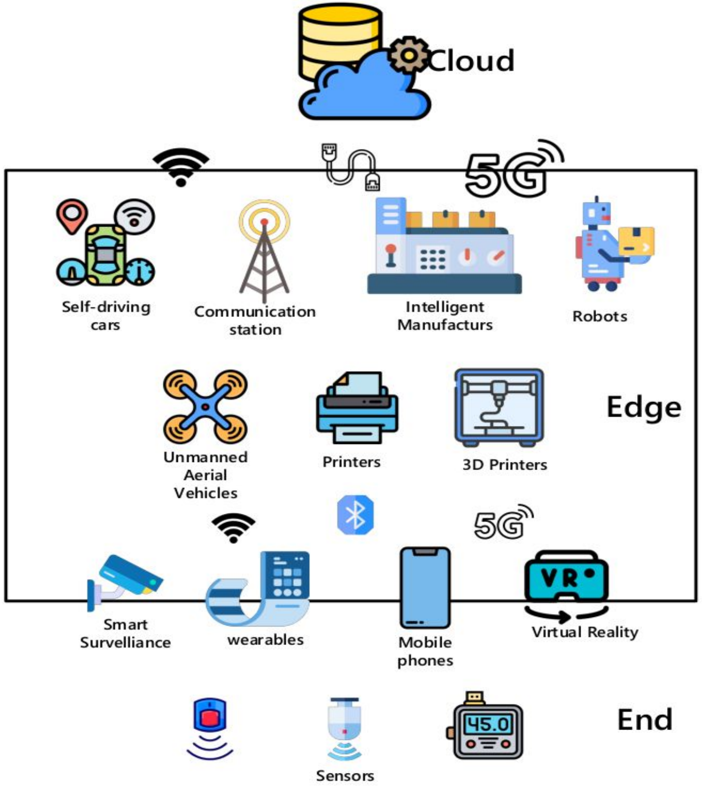
Co-design Methodology

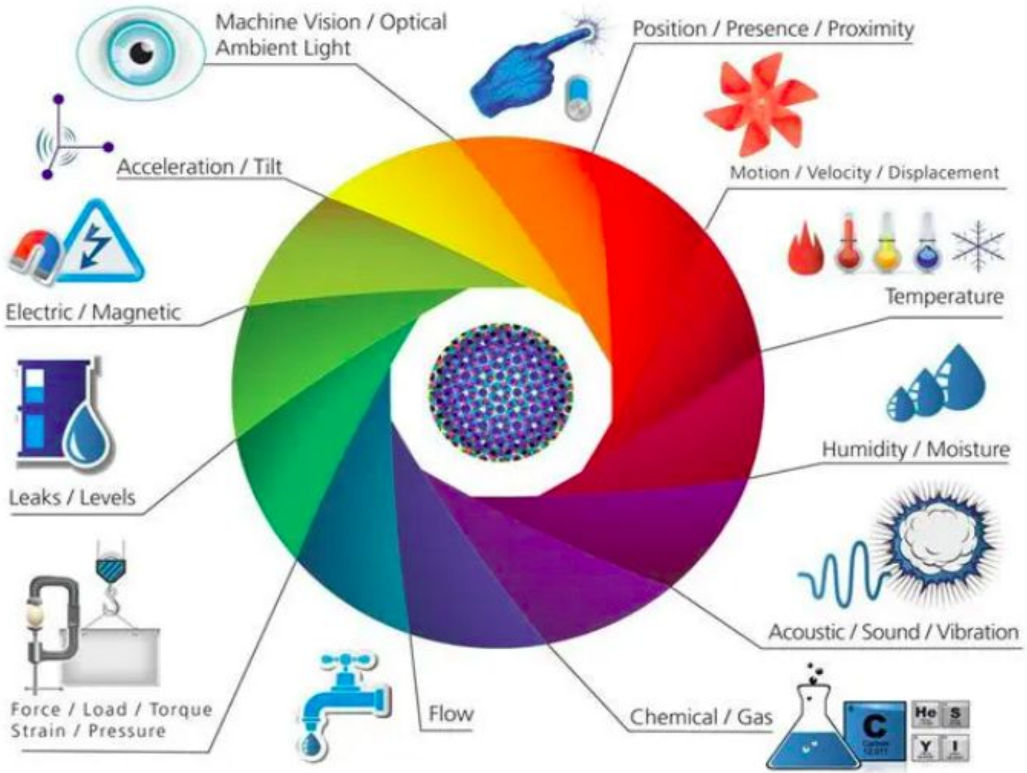


- **Data reuse** is the key to achieving **high energy efficiency**.
- High PE utilization with **adaptive on-chip networks** is the key to achieving **high performance**
- Co-design of **dataflow** and **hardware** is critical for the optimization of **performance**, **energy efficiency** and **flexibility** for DNN accelerators.

Adapted from Hao et al., "Enabling Design Methodologies and Future Trends for Edge AI: Specialization and Co-design", IEEE Design & Test, 2021

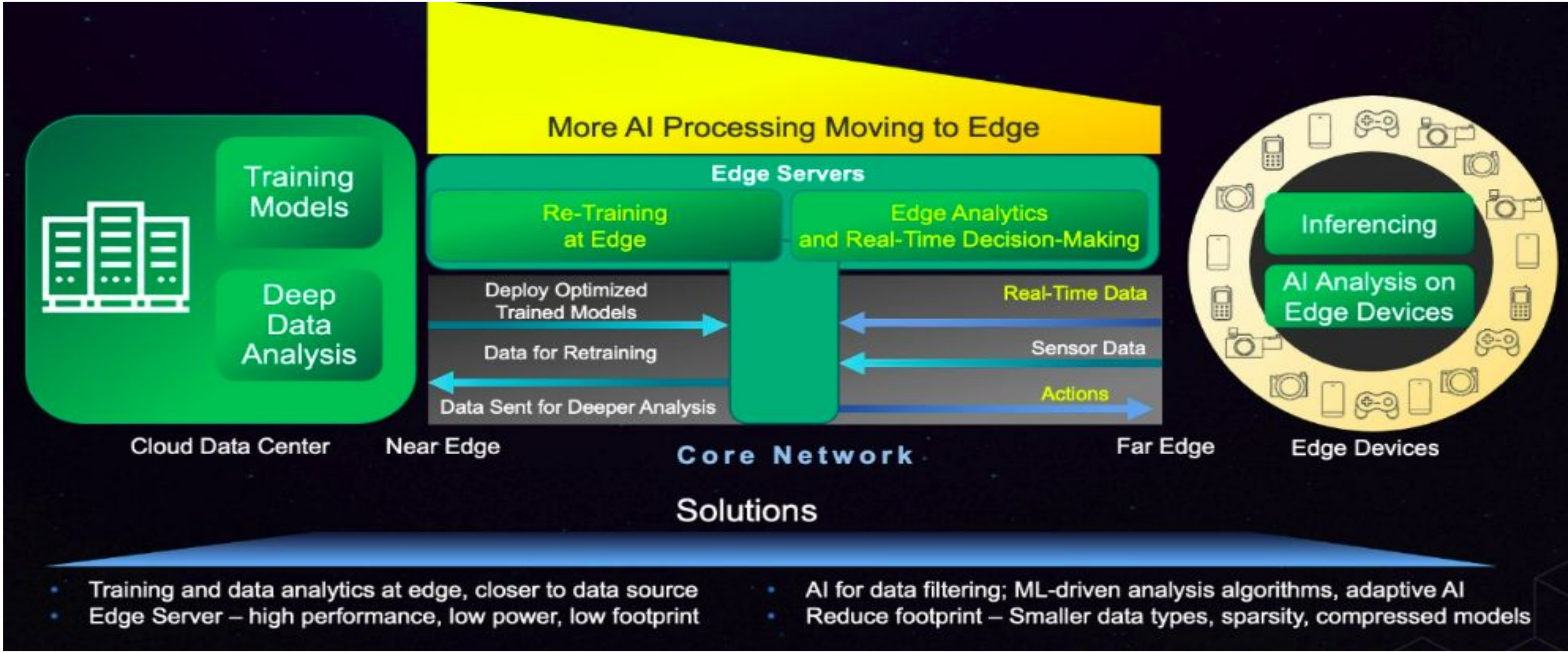
AI Moving Close to Edge





Sensors are everywhere!

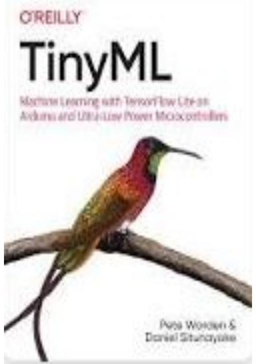
AI Moving Close to Edge



Source: CadenceLive 2021

Power Concerns

https://www.tinyml.org/

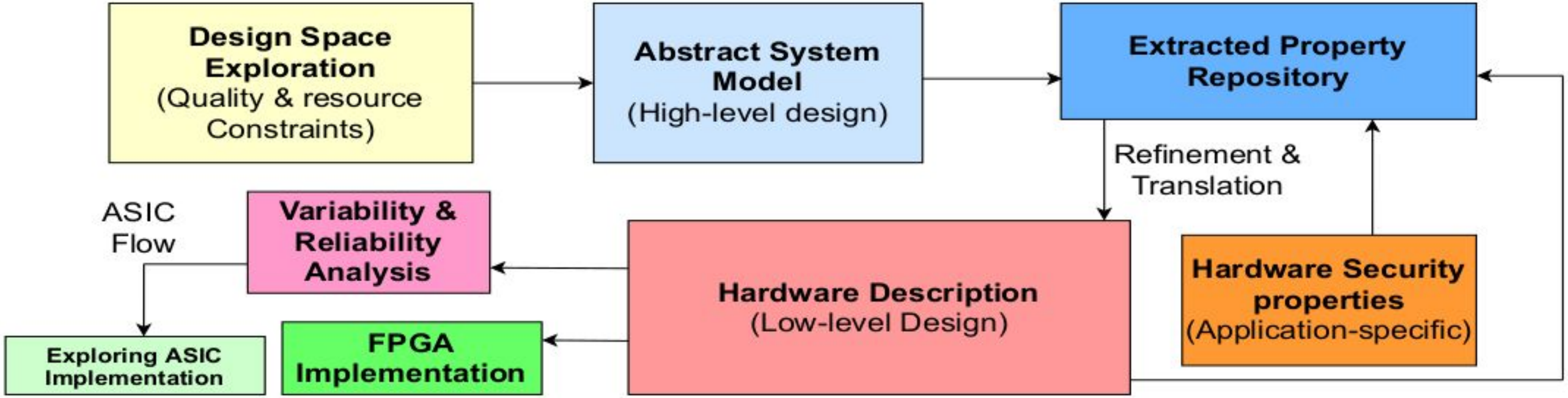


Slide: Courtesy of Xiangyu Yu, and Kurt Keutzer

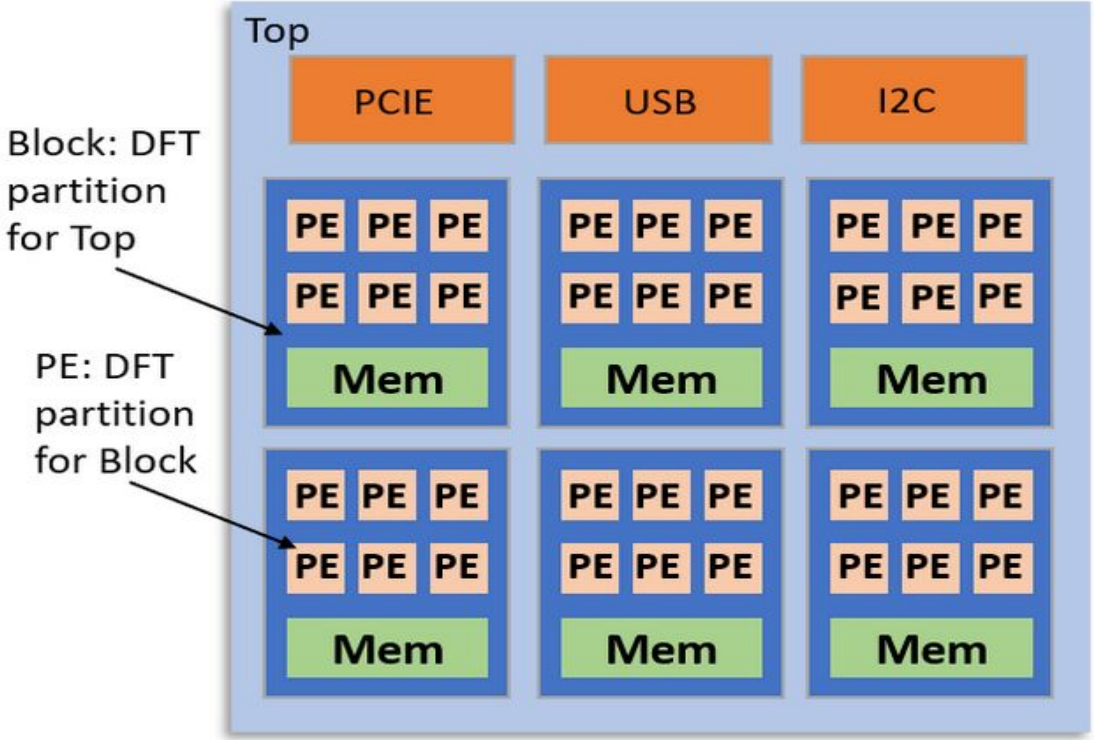
Ref: Amir Gholami’s slides (UCB)

Verification of AI Designs/Architectures

- Property-driven methodology development for hardware designs
- Design size can be relatively small with mix of datapath and control mechanisms
- Verification goals: security and dependability
- Target systems: safety-critical applications (e.g. healthcare)
- **Formal guarantees to ensure correct operation under all conditions**



Test Techniques (DFT) for AI Architectures



1. Incremental test-pattern generation (reuse of test patterns for new/optimized version of the design).

2. Core-level pattern generation and retargeting at top-level.

DFT: Design-for-Testability

https://semiengineering.com/ai-testing-pushing-beyond-dft-architectures/

Some Available Resources (Open-Source)

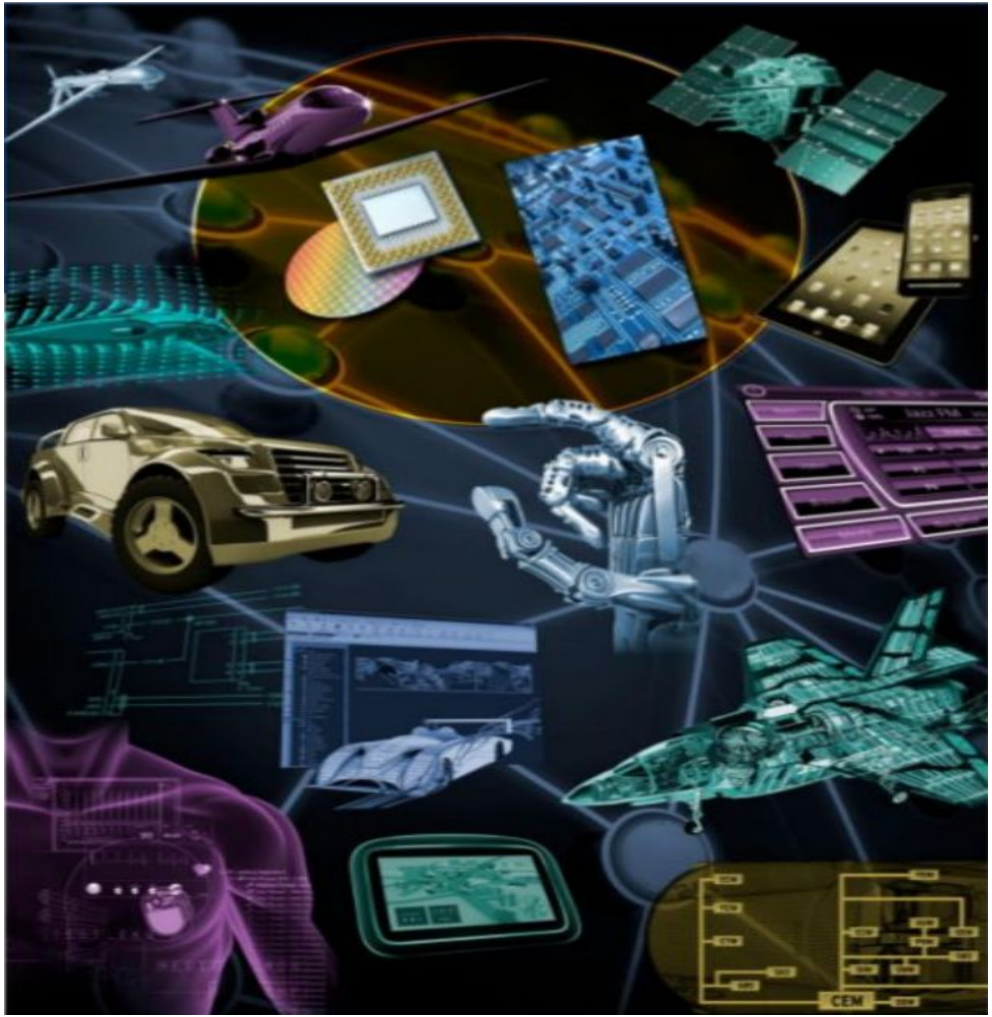
High-level AI Accelerator Modeling	https://github.com/harvard-acc/ALADDIN
DNN Accelerators Design Space Exploration	https://github.com/ARM-software/SCALE-Sim
CPU/Memory Architecture Modeling	https://www.gem5.org/
Accelerator Modelling (Eyeriss Design, MIT)	https://github.com/taoyilee/clacc
Python-based Modeling of Hardware	https://www.myhdl.org/
DRAM Power & Energy Estimation	https://www.es.ele.tue.nl/drampower/
Running ML Algorithms (in Python)	https://scikit-learn.org/stable/
Open-source ASIC flow	https://theopenroadproject.org/
RTL simulation (Verilog/VHDL)	https://www.edaplayground.com/
Online Platform for ML/DL Training/Inference	https://colab.research.google.com/?utm_source=scs-index
Processing-in-Memory simulator	https://github.com/CMU-SAFARI/ramulator-pim

GitHub

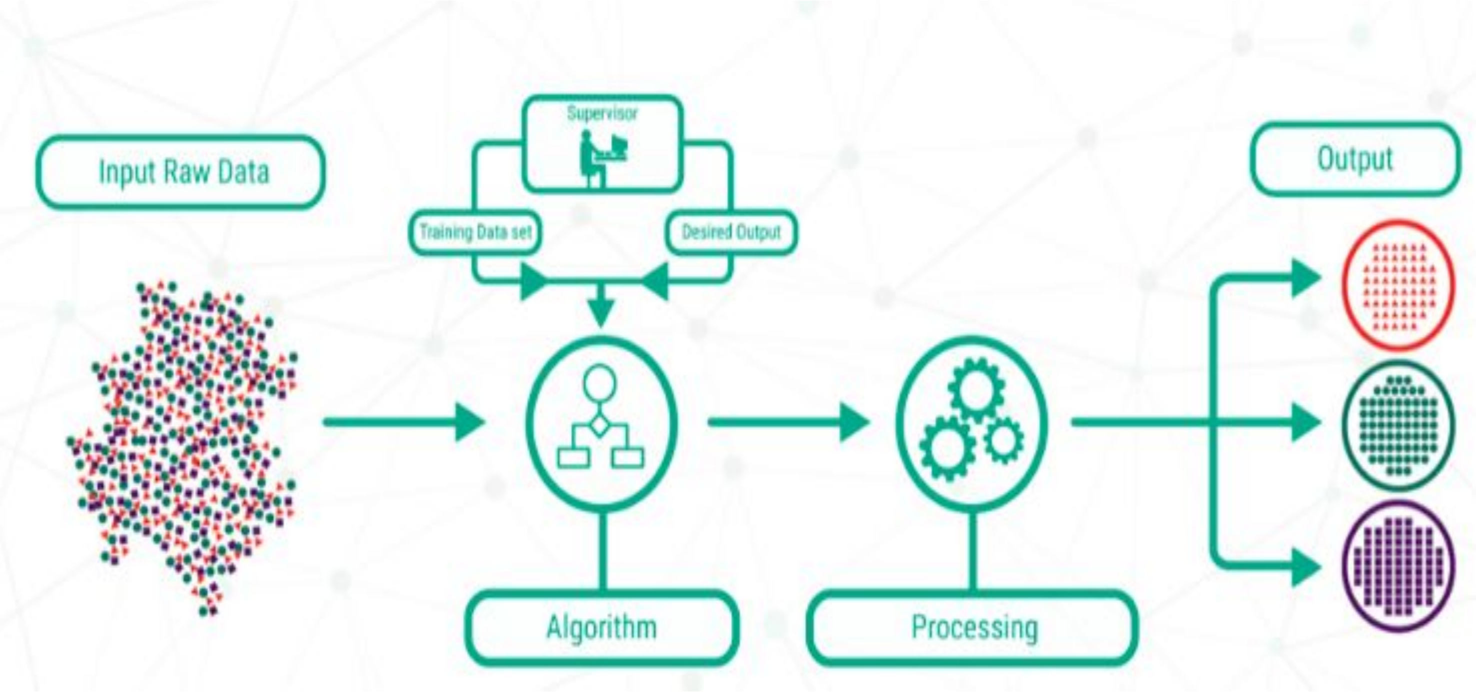
<UBUNTU = "I am because you are.">

THANK YOU

All queries be addressed to binod@iitj.ac.in



Supervised Learning



Unsupervised Learning