
DTU

Network Embedded Systems

Week 11: Embedded AI

Emil Njor
Ph.D. Student
emjn@dtu.dk
www.compute.dtu.dk/~emjn

Guest Lecturer

Ph.D. Student at the Technical University of Denmark (DTU)

Currently a Visiting Researcher at Harvard University

Research on Ressource Constrained Machine Learning

Presentation Outline

Introduction to Machine Learning

Machine Learning Systems

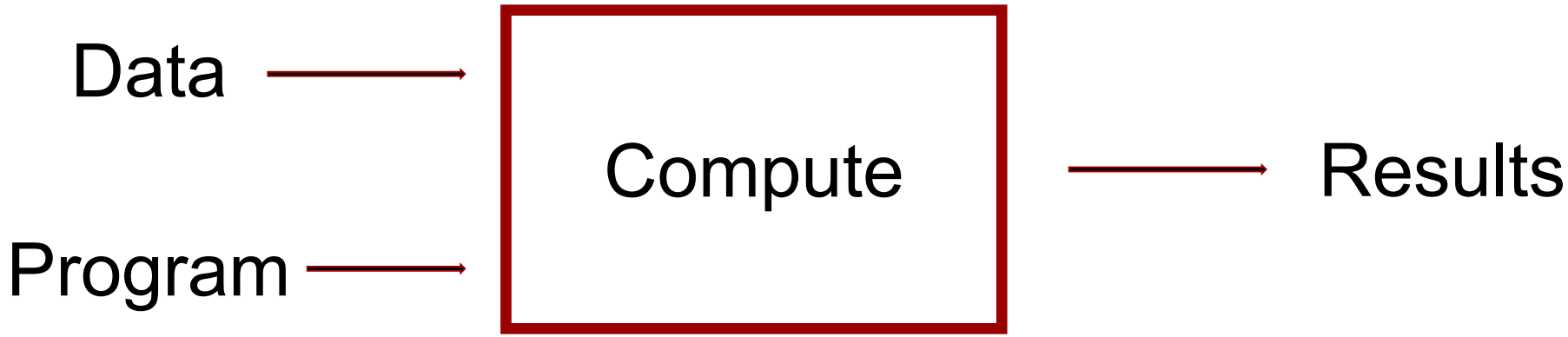
Designing Embedded ML Systems

- Model Optimizations
- Data Optimizations
- Hardware Optimizations

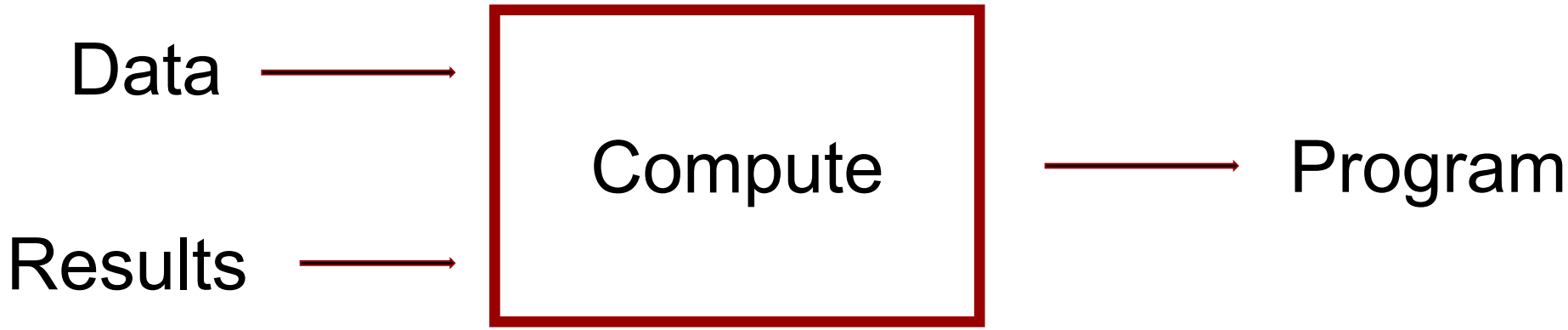
Practical Showcase

Introduction to Machine Learning

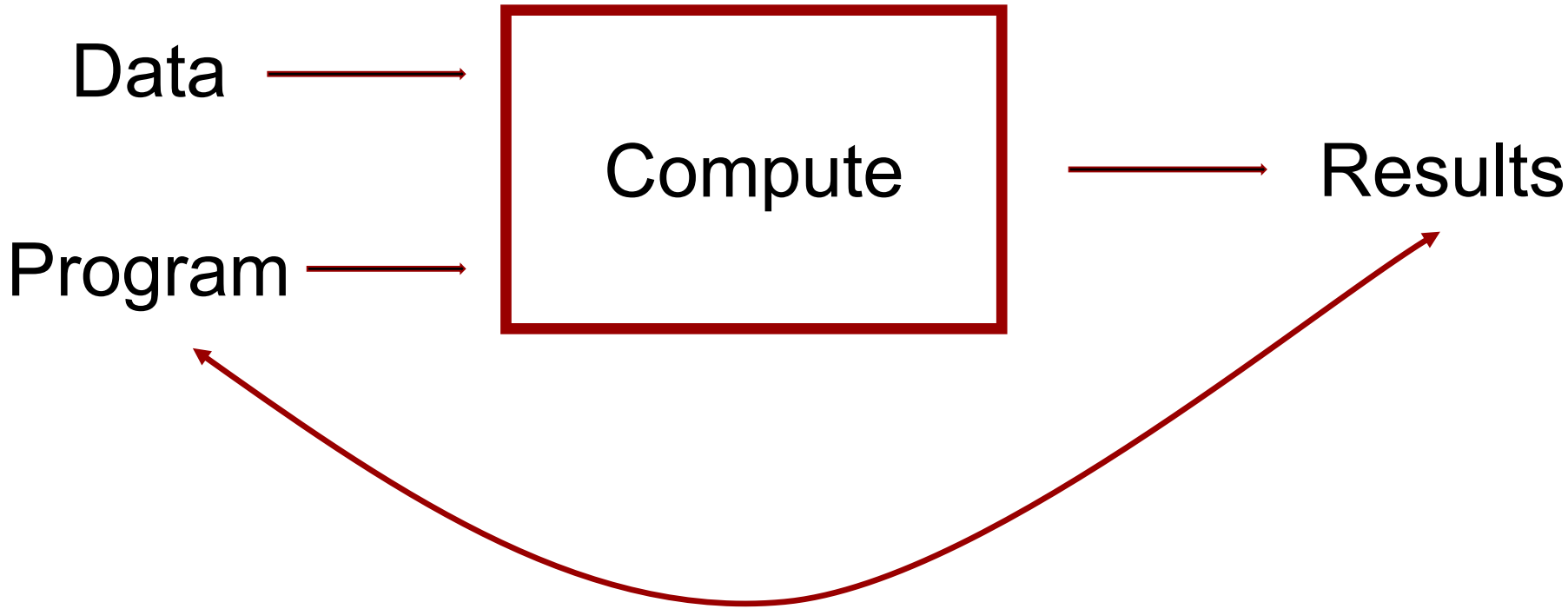
Typical Programming



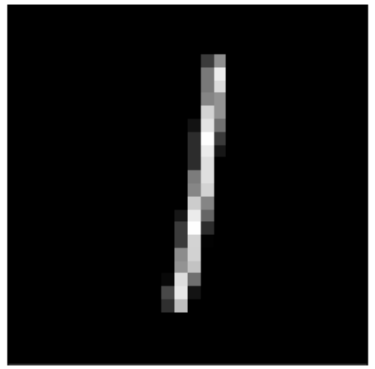
Machine Learning (Learning)



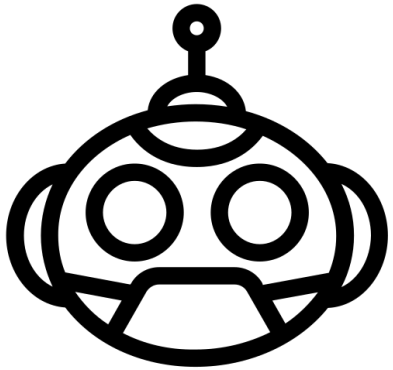
Machine Learning (Use)



Advantages

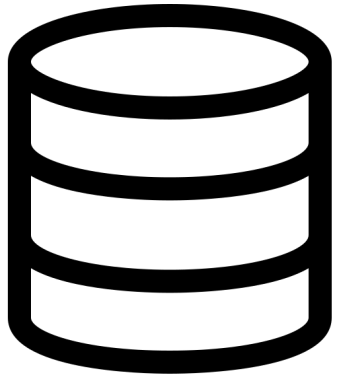


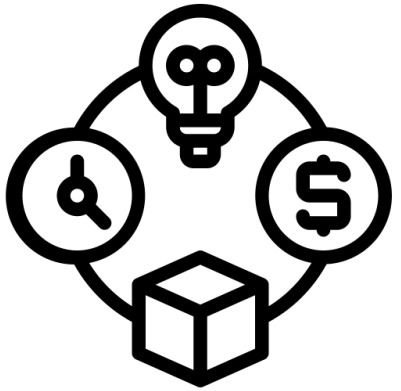
Pattern Recognition



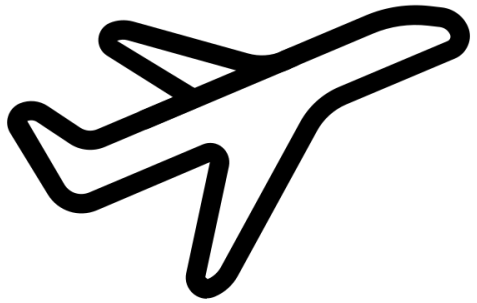
Automation

Disadvantages



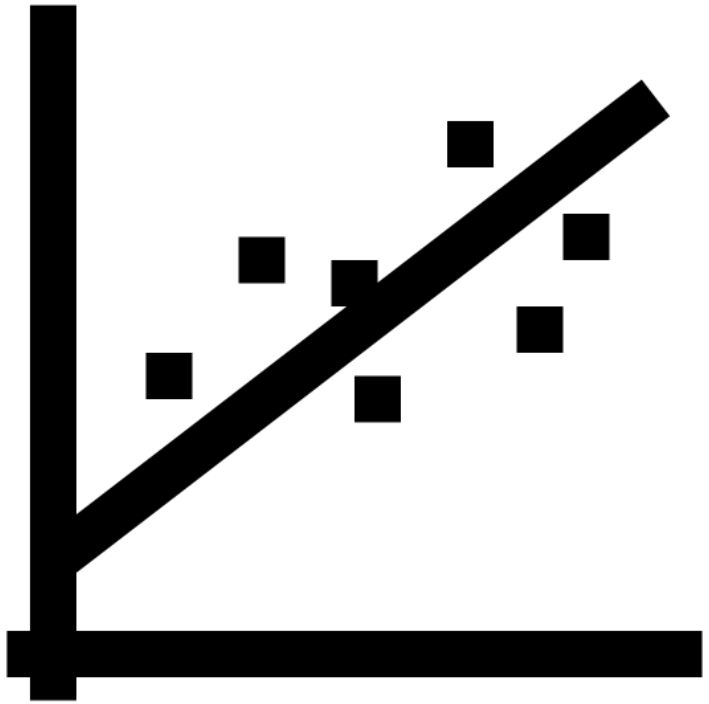


Resources Consumed

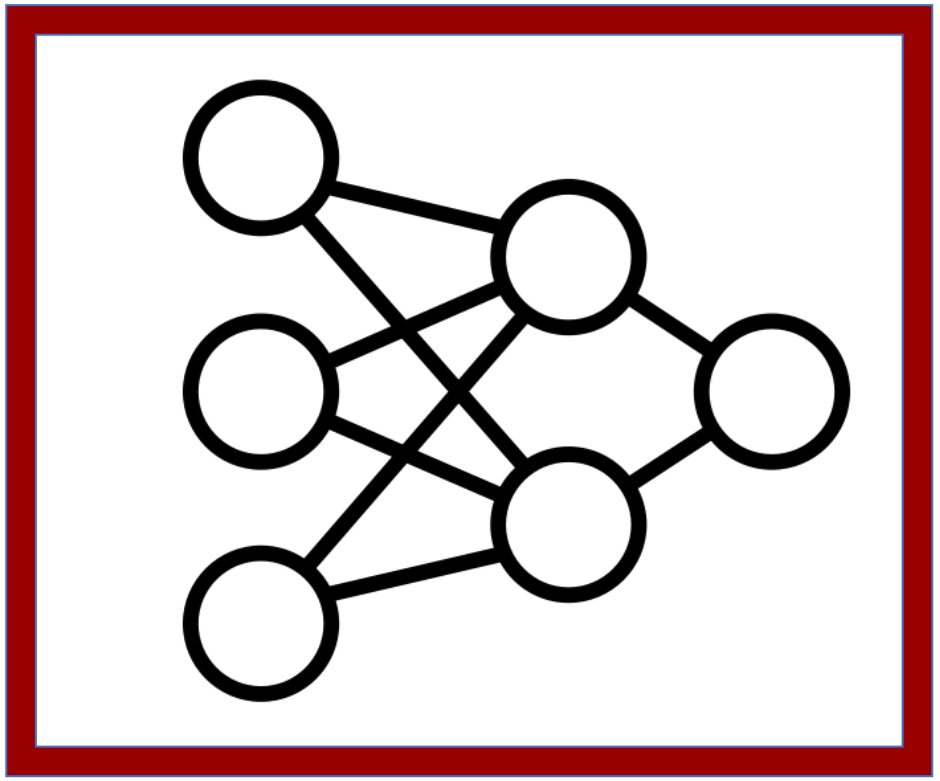


Black Box

Types of Machine Learning

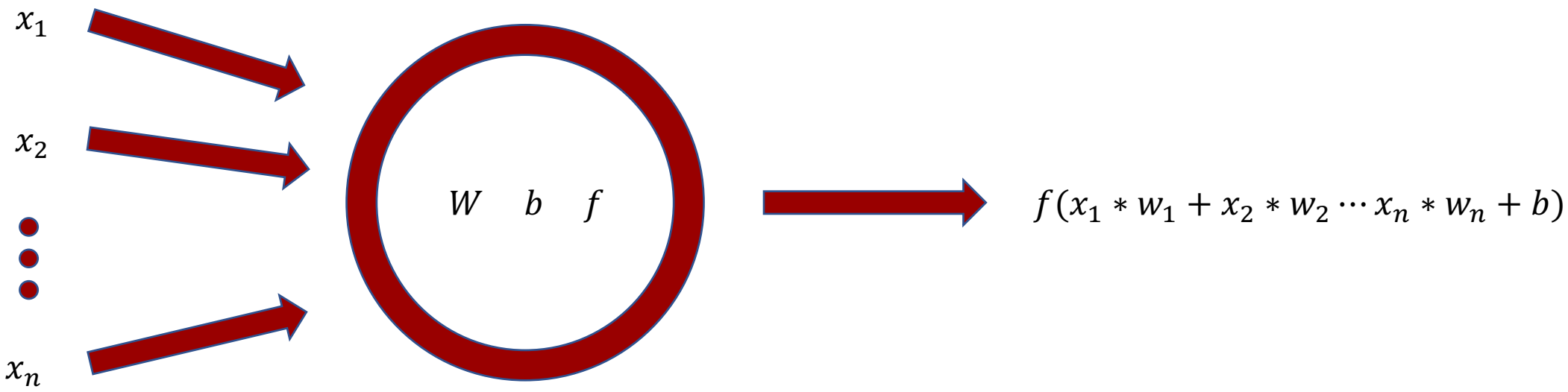


Traditional Machine Learning

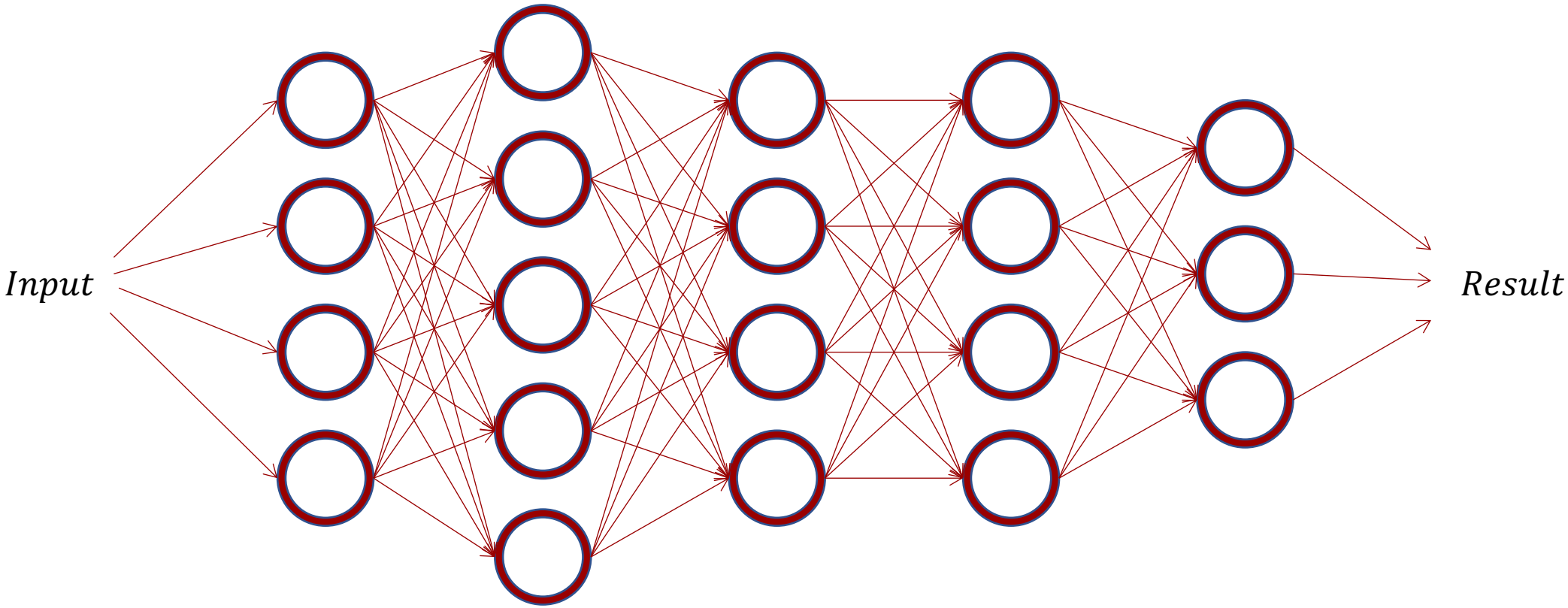


Deep Learning

The Neuron

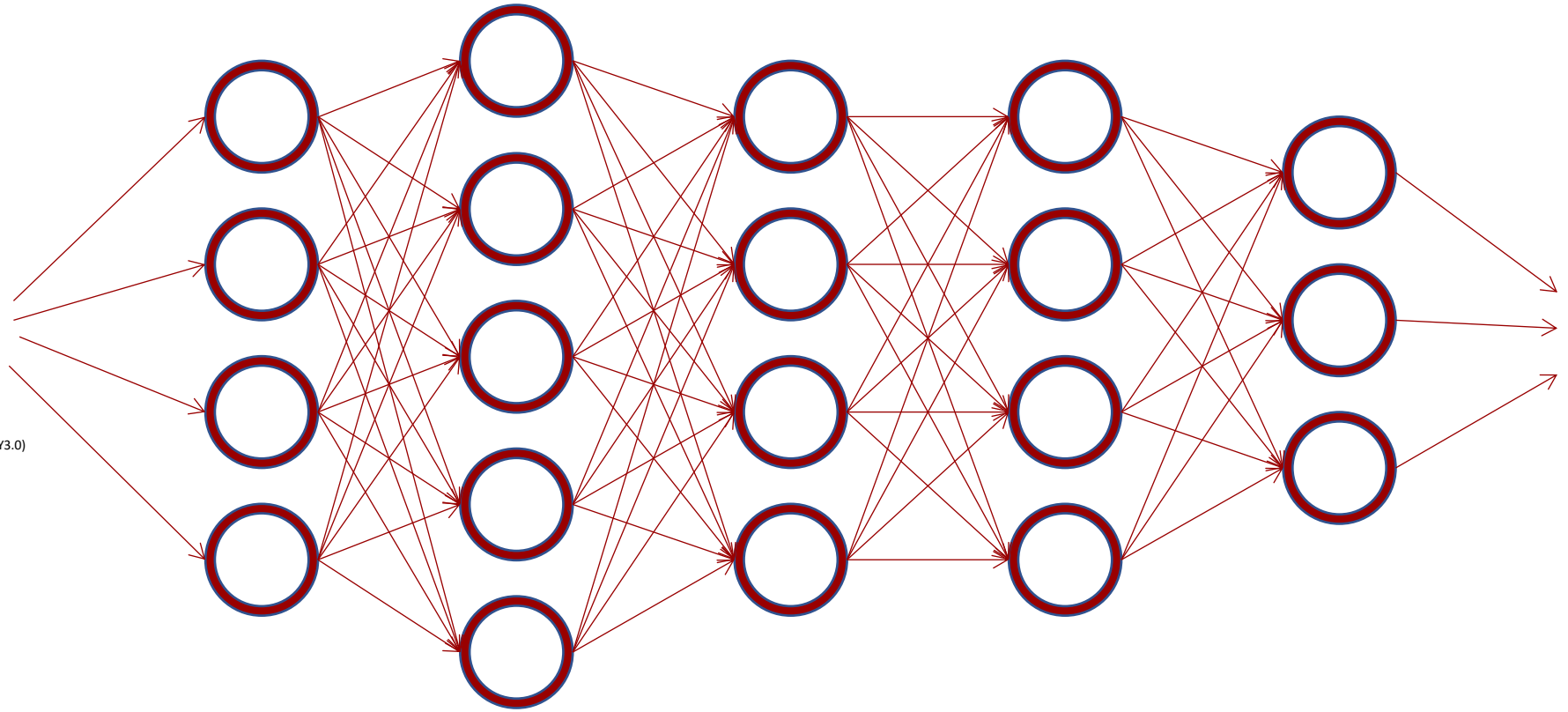


A Neural Network



Untrained Neural Networks

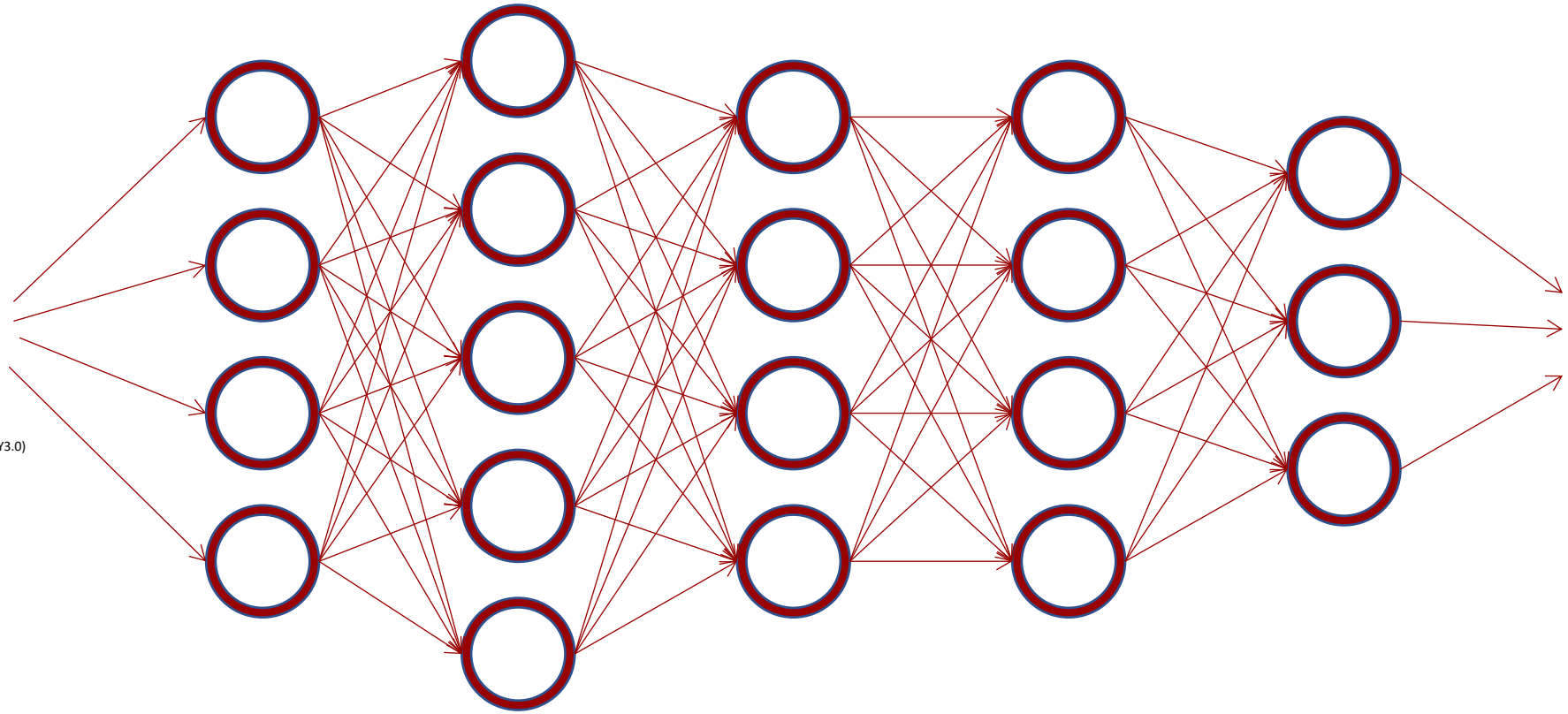
Cow by Rfourtytwo from Noun Project (CCBY3.0)



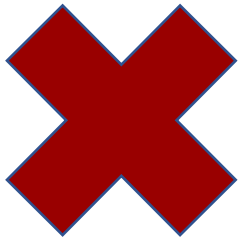
Cat

Untrained Neural Networks

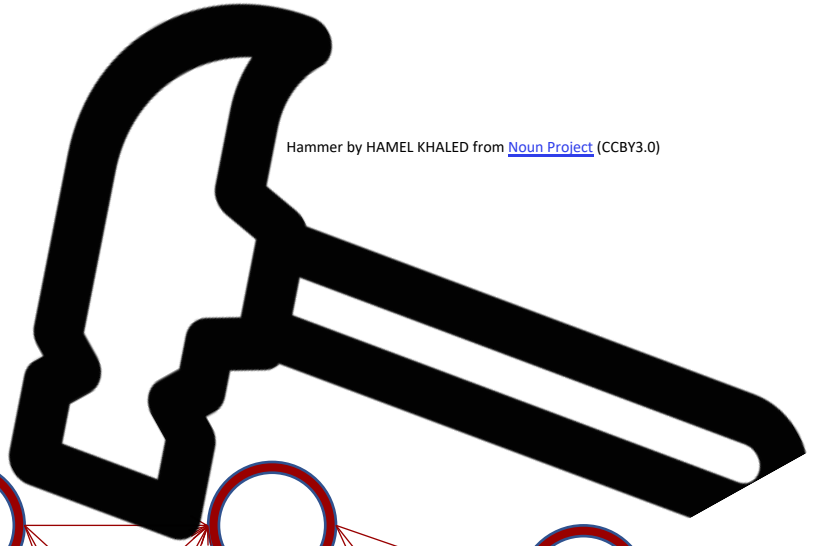
Cow by Rfourtytwo from Noun Project (CCBY3.0)



Cat

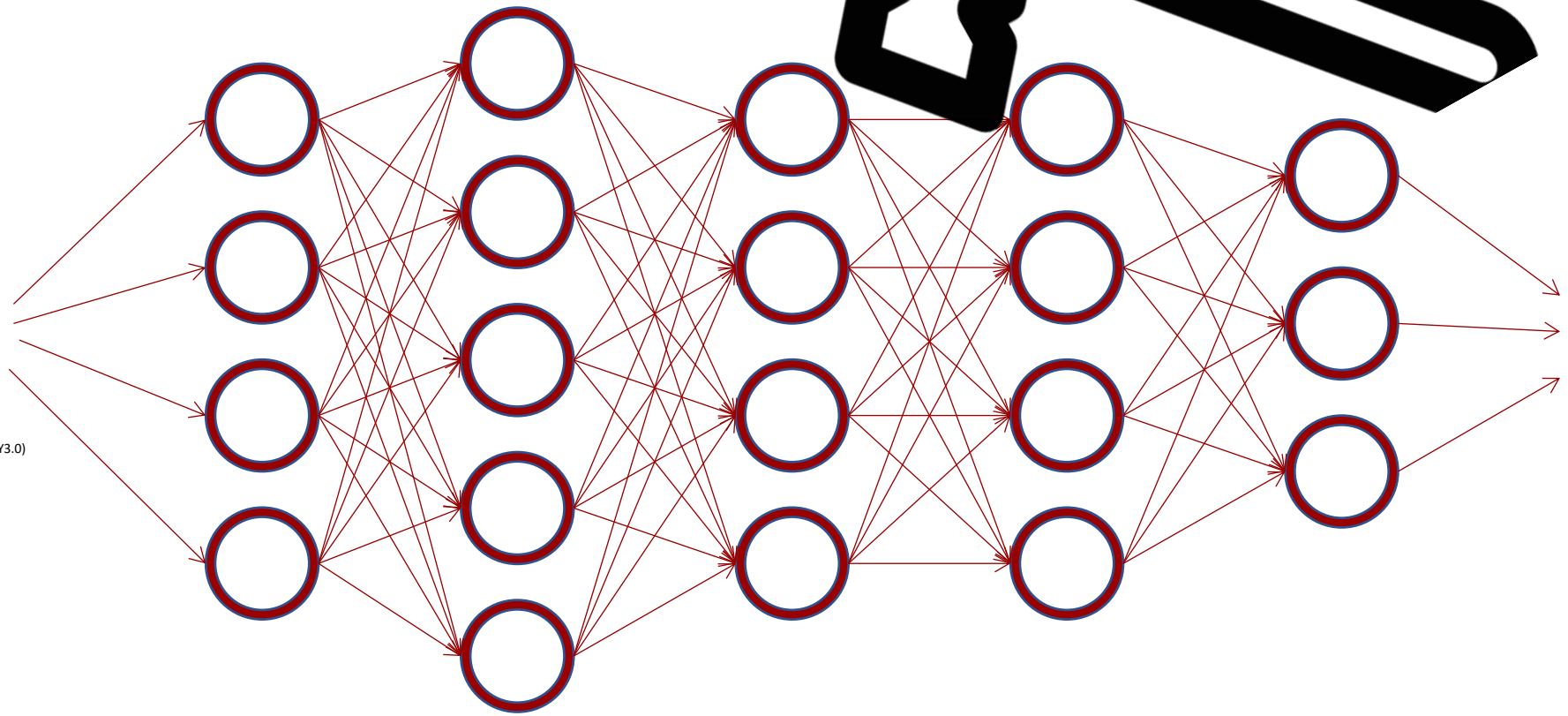


Training Neural Networks

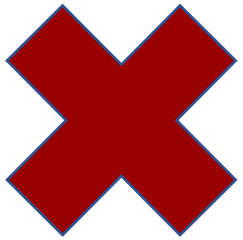


Hammer by HAMEL KHALED from Noun Project (CCBY3.0)

Cow by Rfourtytwo from Noun Project (CCBY3.0)

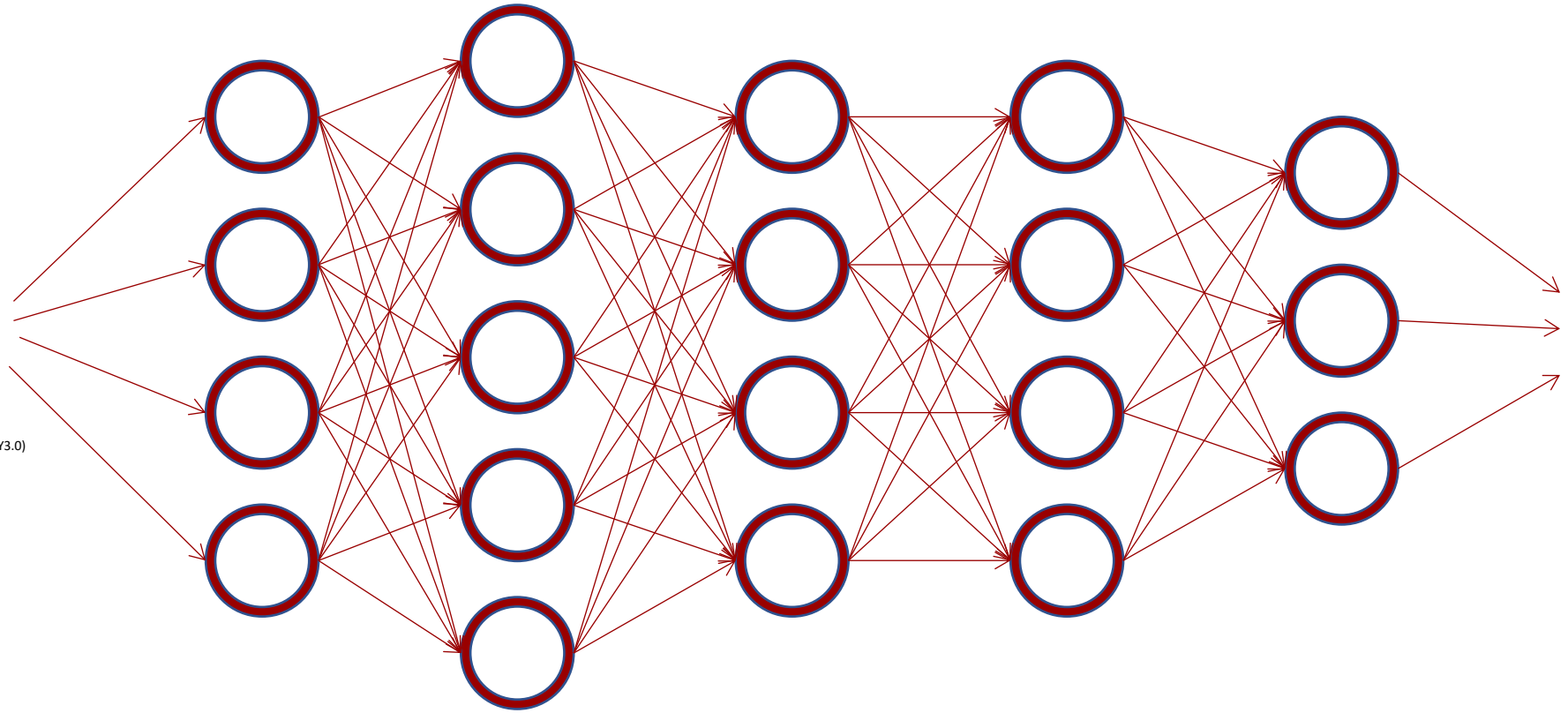


Cat



Trained Neural Networks

Cow by Rfourtytwo from Noun Project (CCBY3.0)



Cow

Backpropagation Example

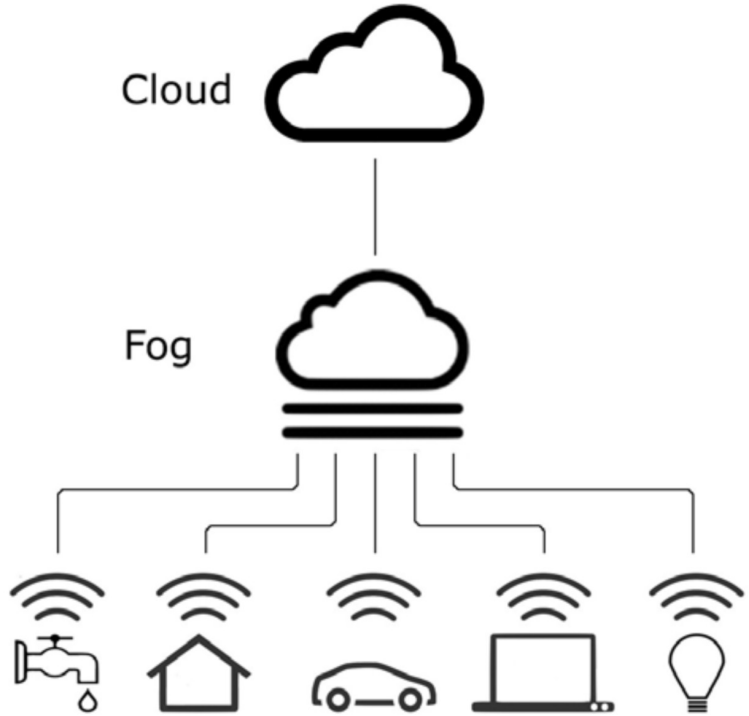
https://www.javatpoint.com/pytorch-backpropagation-process-in-deep-neural-network

Machine Learning Systems

The Cloud Based ML System

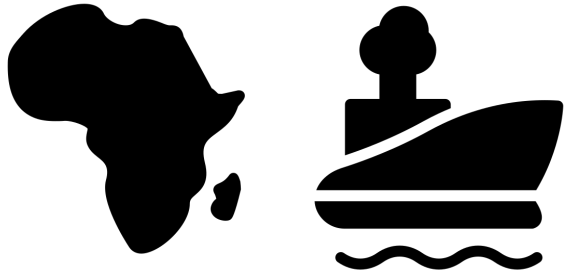
The archetype ML System

- Lots of computing power
- Lots of storage
- Possible to train really accurate but very resource-consuming deep learning models

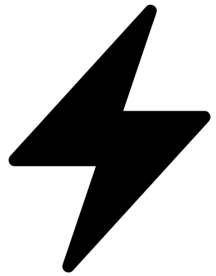


Why can a Cloud Based ML System be a bad idea?

Why can a Cloud Based ML System be a bad idea?

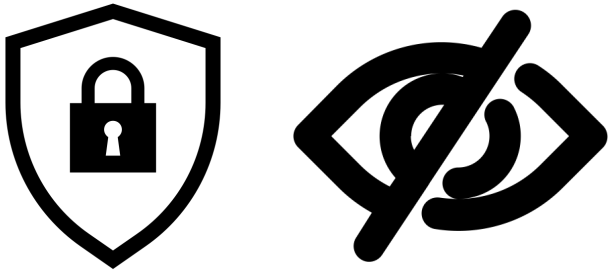


Remote Areas



Energy Consumption

Latency

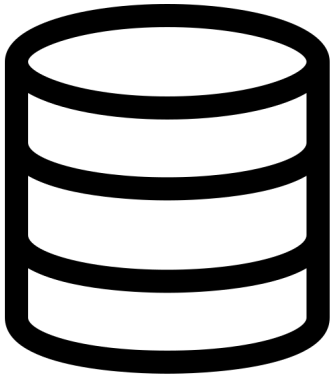


Security & Privacy

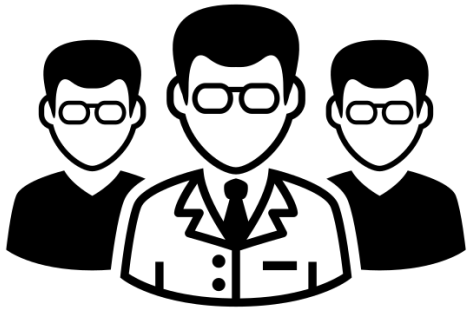
Reliability

Cost

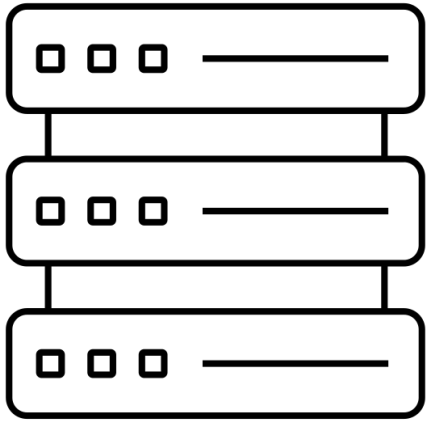
Sidetrack:
Resource Consumption of Modern ML Systems



Data



Personnel

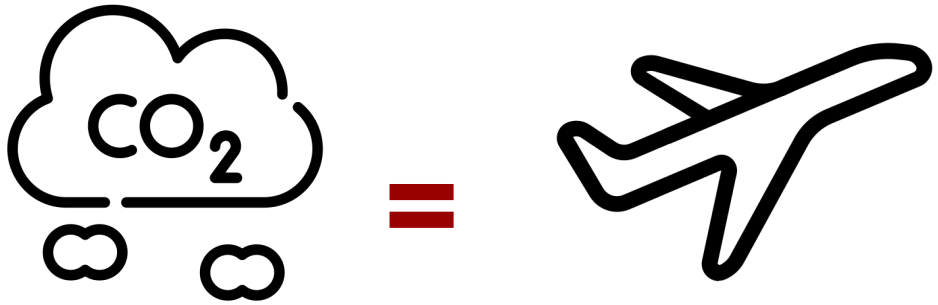


Hardware

Less accessible to small businesses

Sidetrack:
Sustainability of Machine Learning

To train one large model:



300.000 kg CO_2

=

125 Round-Trip flights
from New York to Beijing

The combined use:

Embedded AI

- Embedded Systems are resource-constrained
- Multiple (often conflicting) performance objectives
  - Predictive performance (accuracy, precision, recall)
  - Resource-efficiency (processing, memory, bandwidth, energy)
- Traditional AI is like a sprint
  - Use all resources to maximise predictive performance
- Embedded AI is like a marathon
  - Efficient use of resources for sustained performance

(Almost) Aliases

Embedded Machine Learning

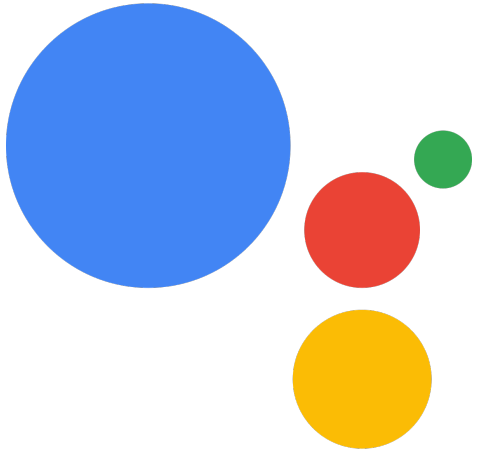
AIoT

Resource-efficient Machine Learning

Resource-constrained Machine Learning

Digital Assistants (Keyword Spotting)

Siri
(Apple)



Google Assistant
(Google)

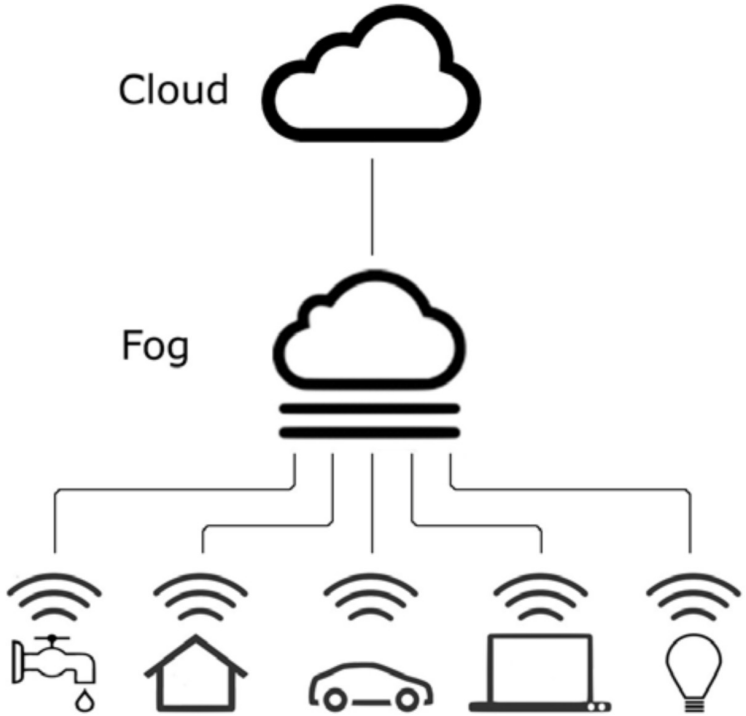
Alexa
(Amazon)

PingMonitor.co (Predictive Maintenance)

HackThePoacher.com (Object Detection)

Cloud-Fog-Edge Continuum

- Computing and AI is distributed in a cloud-fog-edge continuum as needed by the applications
- Embedded AI
  - AI comes to Embedded Systems (inference, training)
  - Embedded System become more autonomous (sensing, knowledge extraction, planning, decision making)
- Local Fog Servers are employed when more computing power or data fusion is needed
  - Privately-owned or offered as a service
- Cloud Data Centres used for really demanding tasks (e.g. training NLP models)

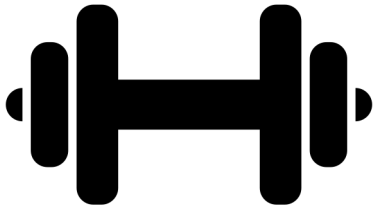


Designing Embedded ML Systems

System Setup

- A typical Embedded AI pipeline assumes off-board training and on-board inference
  - Example: Tensor Flow Lite Micro (TFLM)
    - Train neural network using TF off-board
    - Convert a TF model in TFLM model format
    - Interpreter that runs on microcontrollers and executes the TFLM model

- On-device training
  - Active Learning: get feedback from the user
  - Transfer Learning: adapt a generic model to local data



System Design Pipeline

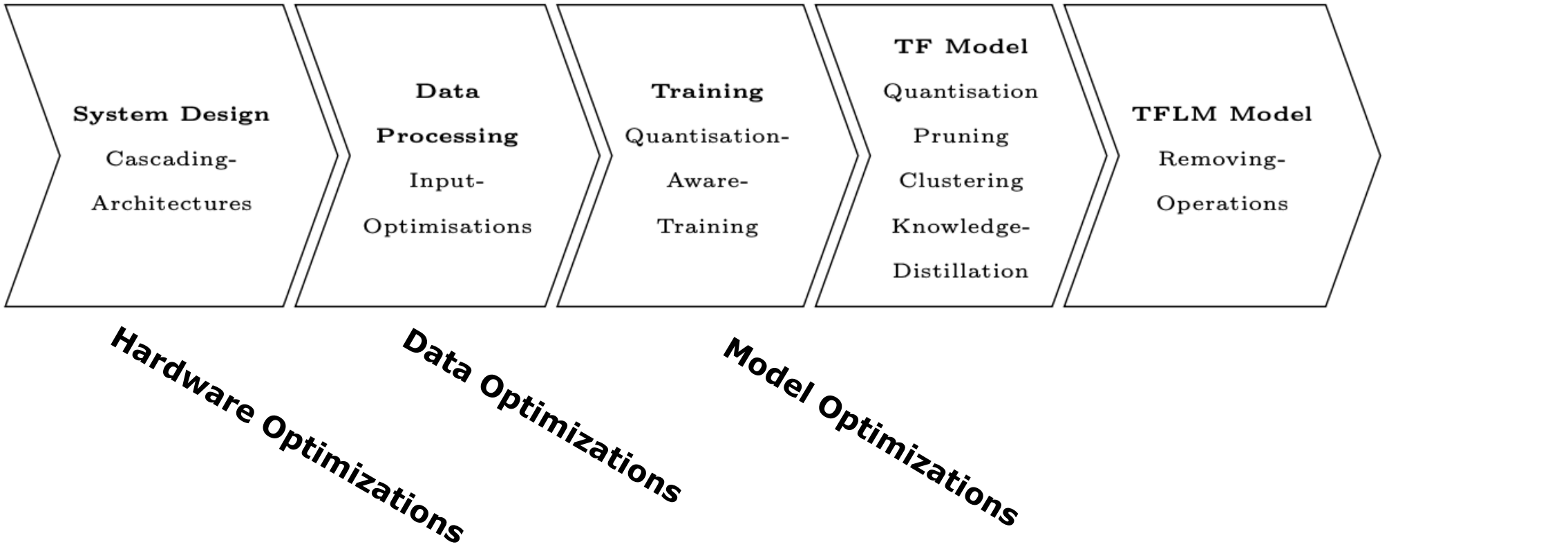
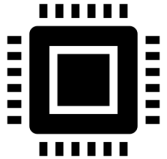


Image source: https://doi.org/10.1007/978-3-031-08337-2_6

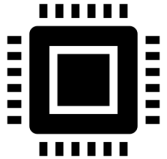
Hardware Optimizations

Central Processing Unit



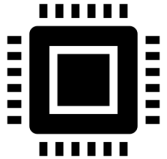
- Implements a number of basic and commonly used instructions
  - Complex operations are compiled into a sequence of basic instructions
- Fast memory usage (e.g. cache memory) is designed according to the principle of locality
  - Reduces overall memory access on the average program in execution
- Implement some degree of parallelisation
  - Pipelining, parallel instruction execution, SIMD (single instruction, multiple data), multicore architectures
- CPUs for embedded applications may have reduced functionality (see ARM Cortex-M series)

Special-Purpose Processor



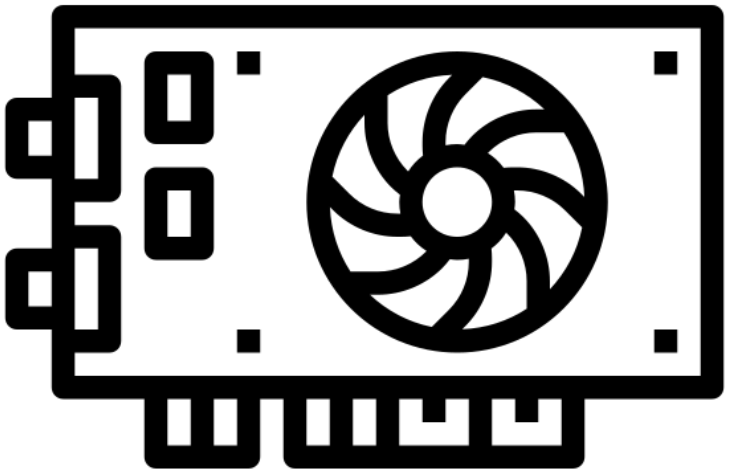
- The challenge with general-purpose processors is that we want them to be able to execute any program
- What what if we optimise a process for a specific program?
- We can create specialised instructions that perform rare and complex operations used by our program
- We can optimise memory access for the specific data input pattern used by our program
- We can take full advantage of any parallelisation opportunity appears in our program
- We can remove instructions and functionality that is useless

Special-Purpose Processor: Pros and Cons



- Special-Purpose Processor or Hardware Accelerators, Engines, xPU…
- Advantages
  - Faster than a CPU
  - More energy efficient than a CPU
- Disadvantages
  - Not flexible, not updatable, suitable for fixed functions
  - Increase cost of chips/boards/platforms, suitable for repetitive functions
  - Higher costs and slower development

GPUs for Machine Learning



gpu by Template from Noun Project (CCBY3.0)

Graphics Processing Units (GPUs):

- Support only a limited amount of operations compared to CPUs.
- Are great at parallel processing of SIMD instructions.
- Have caches optimized for non-predictable memory access.

Comparison between CPU and GPU

Feature	Multicore with SIMD	GPU
SIMD Processors	4–8	8–32
SIMD Lanes/Processor	2–4	up to 64
Multithreading hardware support for SIMD Threads	2–4	up to 64
Typical ratio of single-precision to double-precision performance	2:1	2:1
Largest cache size	40 MB	4 MB
Size of memory address	64-bit	64-bit
Size of main memory	up to 1024 GB	up to 24 GB
Memory protection at level of page	Yes	Yes
Demand paging	Yes	Yes
Integrated scalar processor/SIMD Processor	Yes	No
Cache coherent	Yes	Yes on some systems

Figure 4.23 Similarities and differences between multicore with multimedia SIMD extensions and recent GPUs.

Computer Architecture: A Quantitative Approach, Sixth Edition, page 335

Arguments for Special-Purpose Processors

CPUs & GPUs spend a lot of time and energy shuffling memory.

Operation	*Energy (pJ)*	Operation	Energy (pJ)	Operation	Energy (pJ)
8b DRAM LPDDR3	125.00	8b SRAM	1.2–17.1	16b SRAM	2.4–34.2
32b Fl. Pt. muladd	2.70	8b int muladd	0.12	16b int muladd	0.43
32b Fl. Pt. add	1.50	8b int add	0.01	16b int add	0.02

Memory operations are ridiculously expensive compared to arithmetic operations

Machine Learning applications have very predictable memory access patterns

However!

- The target domain needs to be large enough to justify the non recurrent engineering costs of a custom ship and software.
- Here FPGAs are an alternative which both lessen the benefits and the drawbacks

What is needed?

Matrix Multiplication unit to do the core matrix multiplication of deep learning.

Local Memory (SRAM)

Interconnect (to connect different processing units and memory)

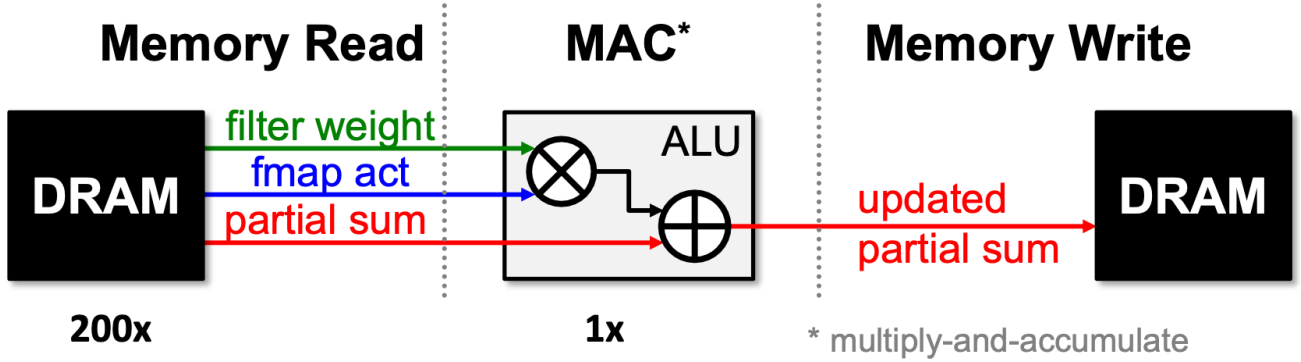
Interfaces to external processor and DRAM

Controller

Useful properties of ML

- High parallelism
  - Means that high throughput is possible

- Memory access is the Bottleneck



Worst Case: all memory R/W are **DRAM** accesses

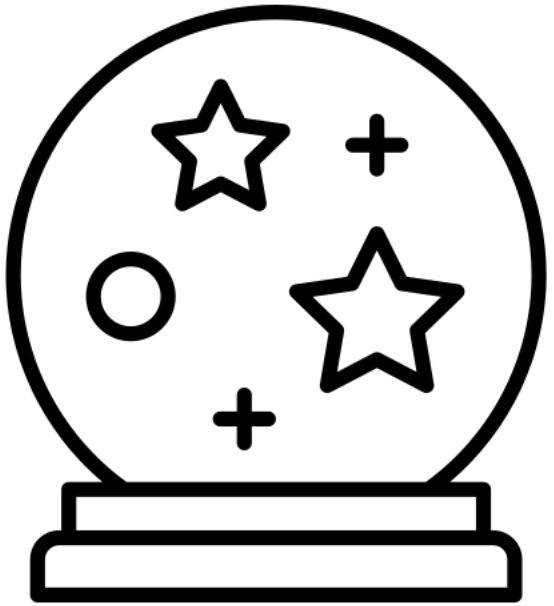
Example: AlexNet has **724M** MACs → **2896M** DRAM accesses required

V. Sze. «Efficient Processing of Deep Neural Networks:from Algorithms to Hardware Architectures». Thirty-third Conference on Neural Information Processing Systems (NeurIPS 2019). Online (08.09.2020): http://eyeriss.mit.edu/2019_neurips_tutorial.pdf

Useful Properties of ML

- High parallelism
  - Means that high throughput is possible

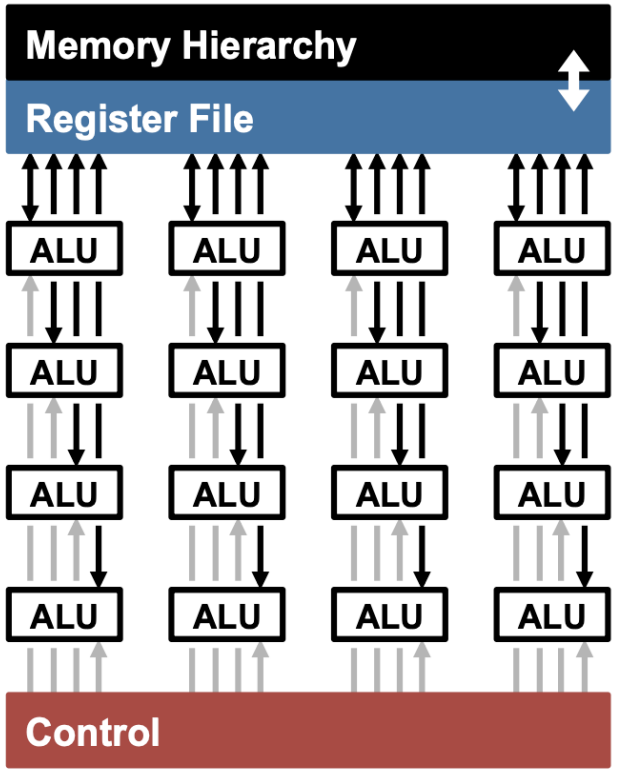
- Memory access is predictable



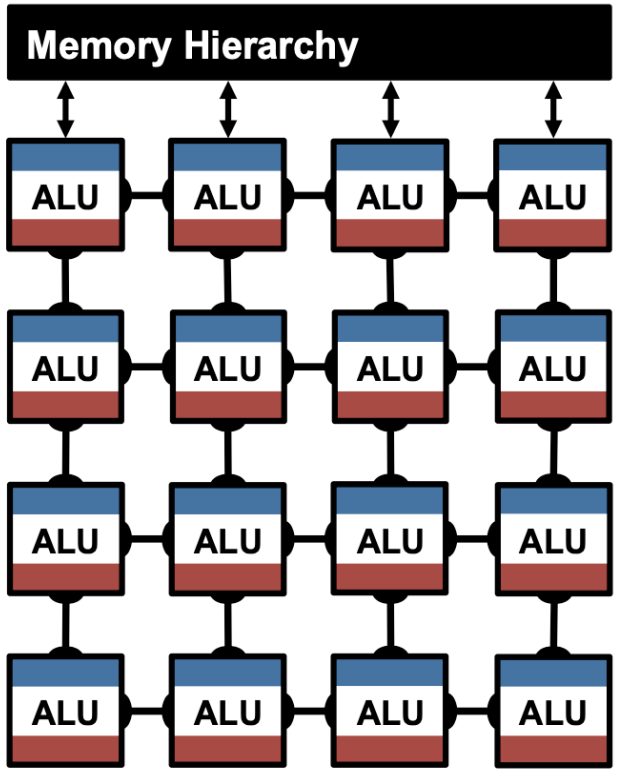
fortune teller by Made from Noun Project (CCBY3.0)

ML Computing Architecture

Temporal Architecture (SIMD/SIMT)

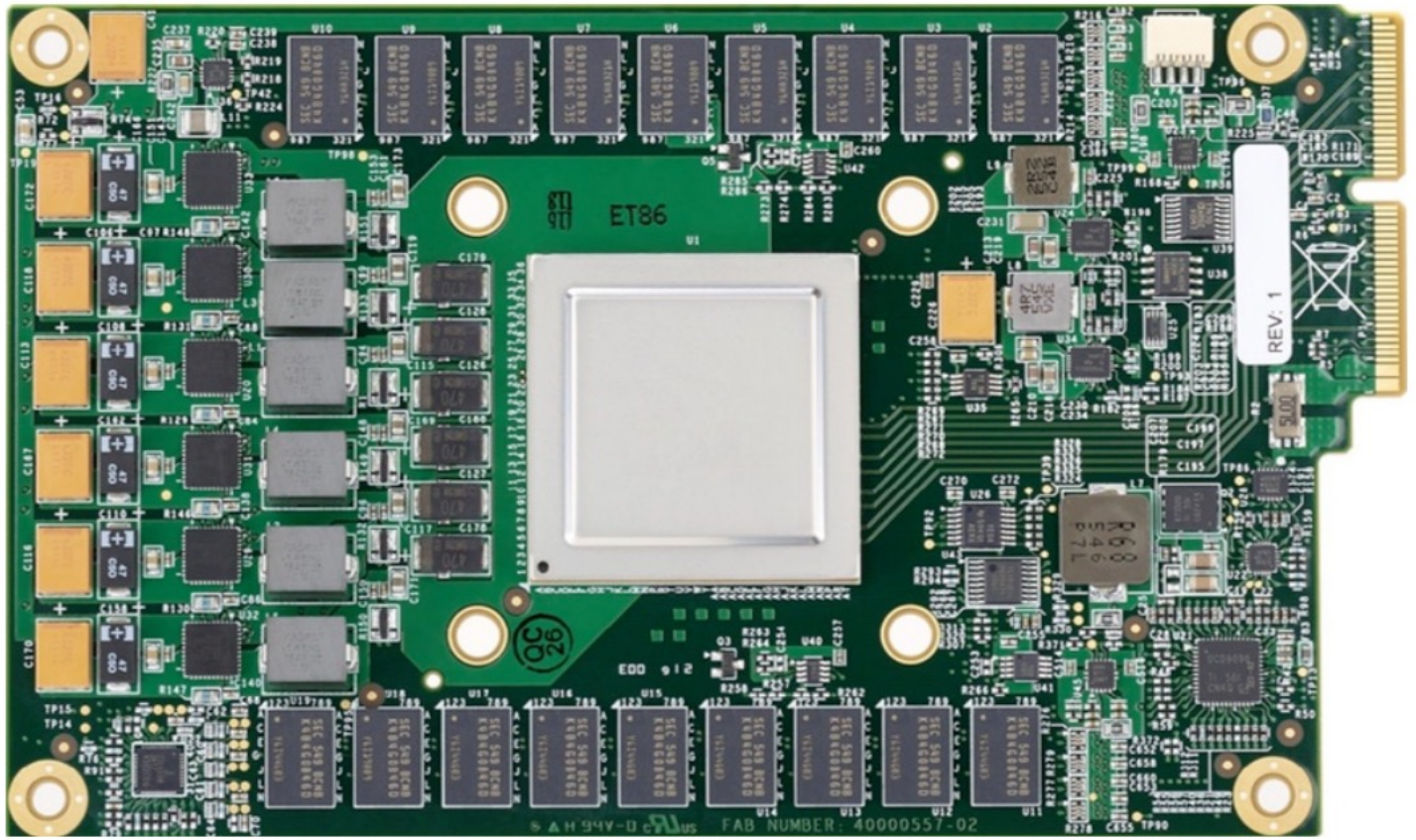


Spatial Architecture (Dataflow Processing)

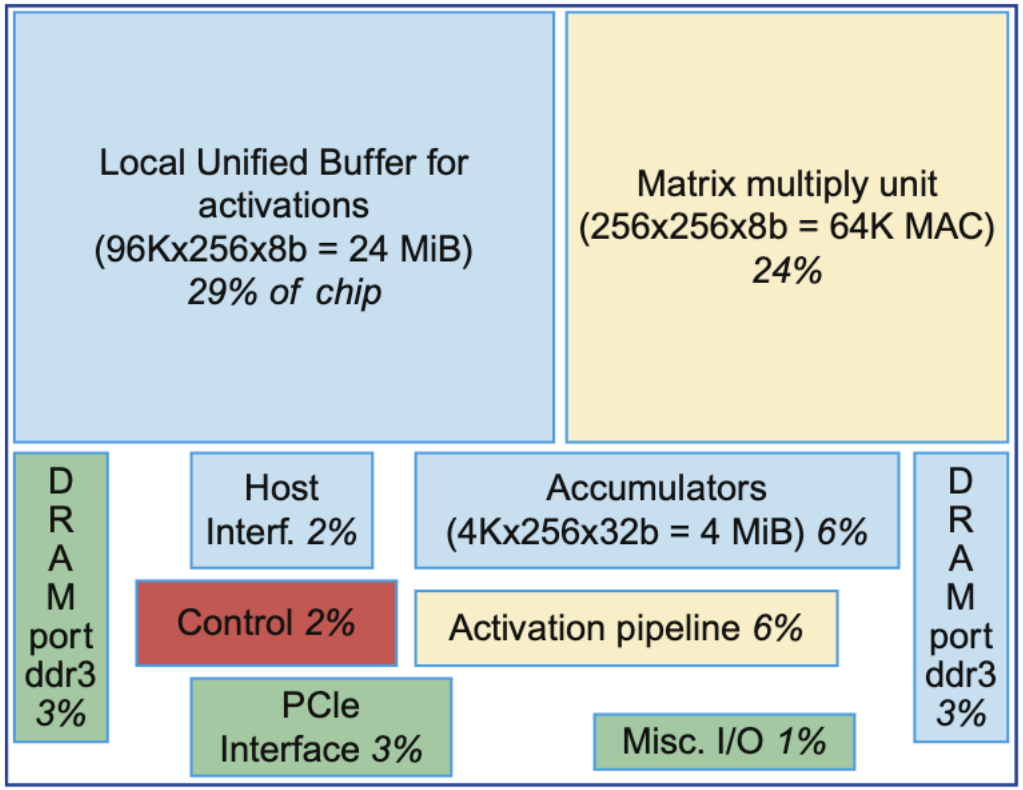


V. Sze. «Efficient Processing of Deep Neural Networks:from Algorithms to Hardware Architectures». Thirty-third Conference on Neural Information Processing Systems (NeurIPS 2019). Online (08.09.2020): http://eyeriss.mit.edu/2019_neurips_tutorial.pdf

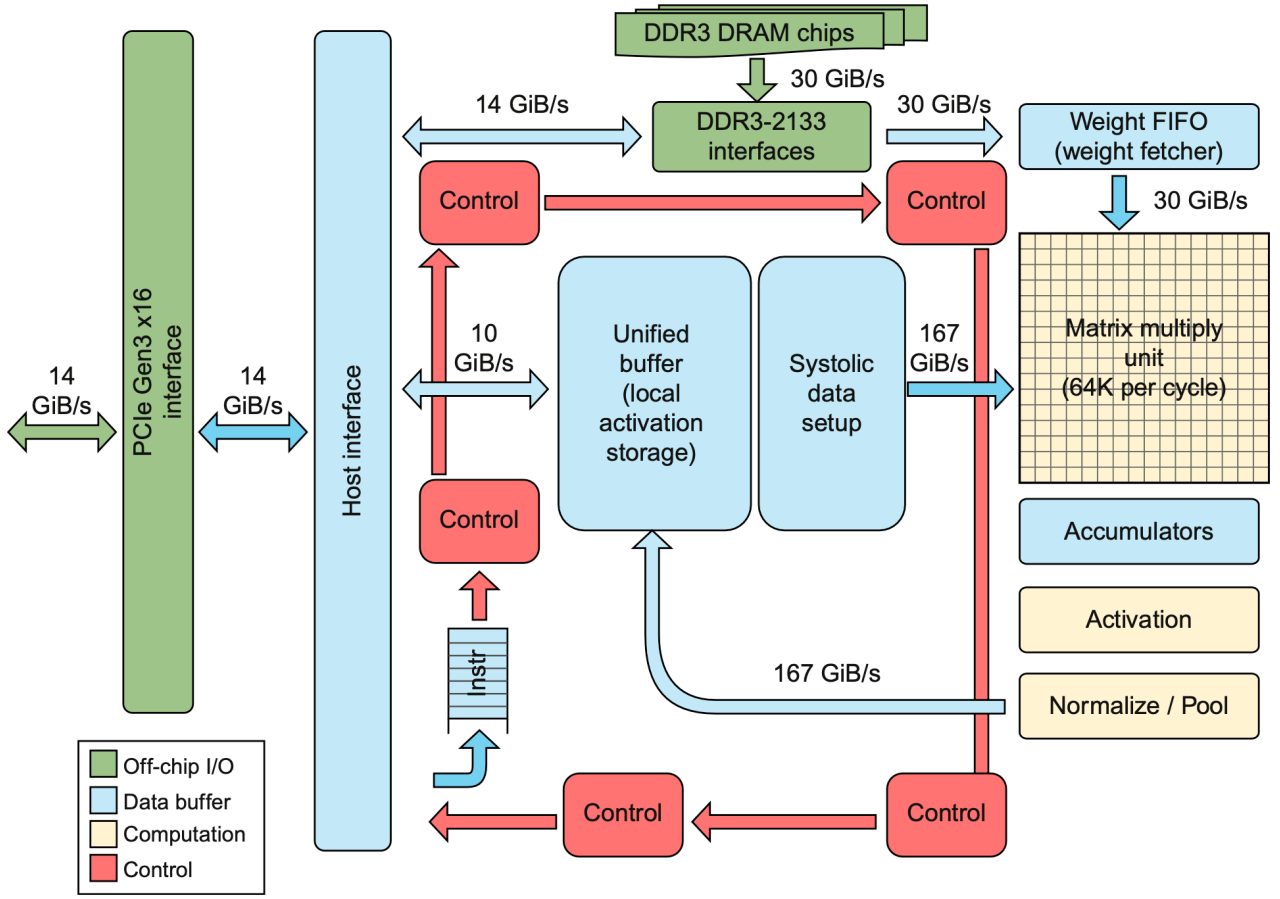
Google TPU



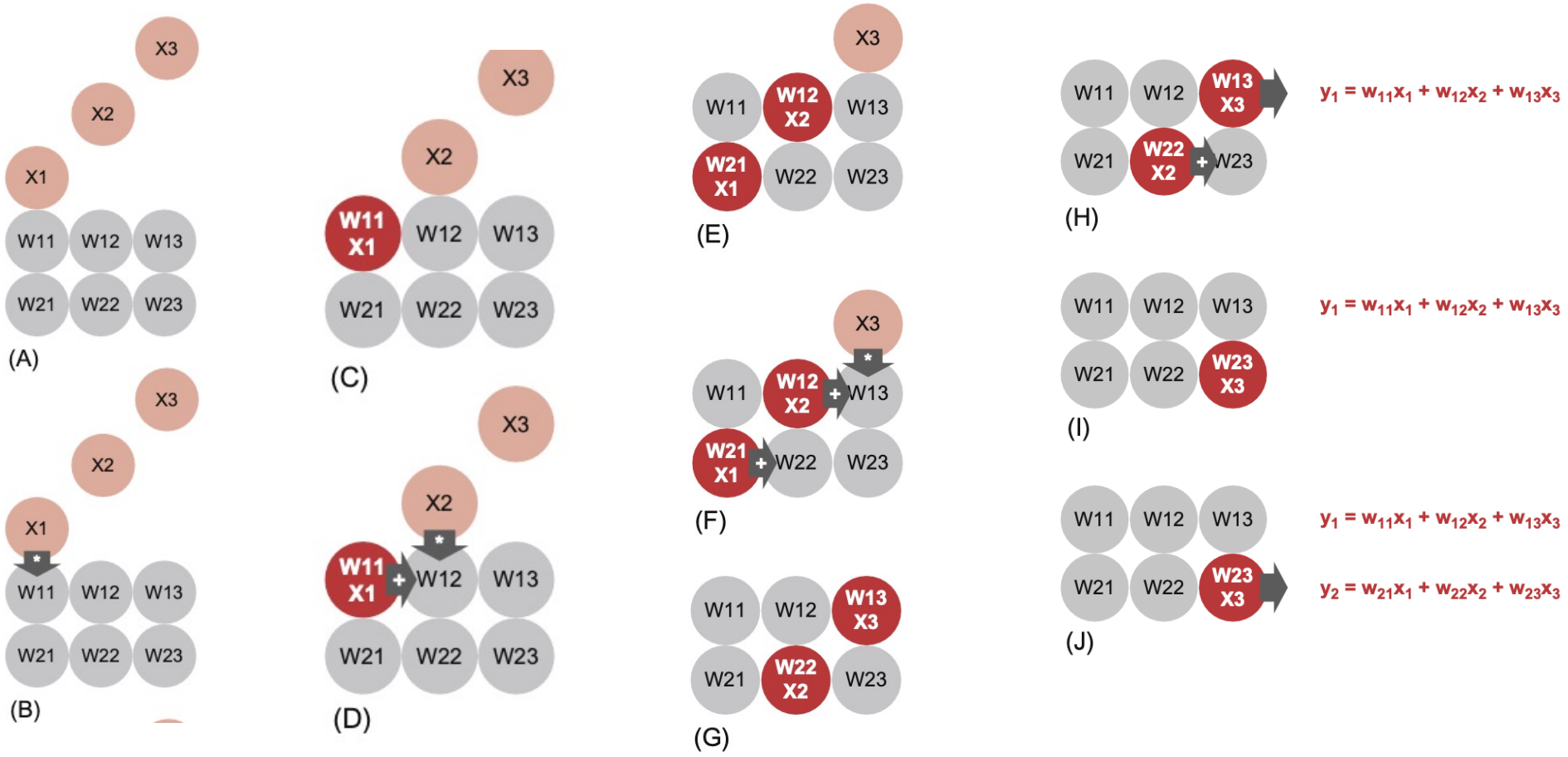
Google TPU Floor Plan



Google TPU Overview



The systolic array



For Embedded Systems

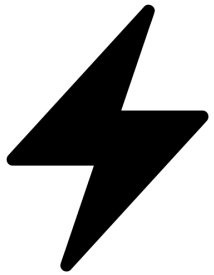
Sidetrack: The New Computer

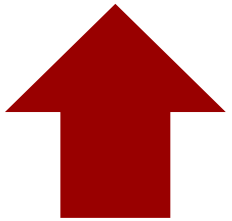
- Consists of hetrogeneous processors
- A standard CPU for conventional programs
- Domain specific architectures for a narrow range of common tasks.
- Compare to a hospital with general practitioners and medical specialists.

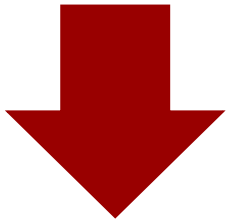
	Google Tensor	Snapdragon 888	Exynos 2100
CPU	2x Arm Cortex-X1 (2.80GHz) 2x Arm Cortex-A76 (2.25GHz) 4x Arm Cortex-A55 (1.80GHz)	1x Arm Cortex-X1 (2.84GHz, 3GHz for Snapdragon 888 Plus) 3x Arm Cortex-A78 (2.4GHz) 4x Arm Cortex-A55 (1.8GHz)	1x Arm Cortex-X1 (2.90GHz) 3x Arm Cortex-A78 (2.8GHz) 4x Arm Cortex-A55 (2.2GHz)
GPU	Arm Mali-G78 MP20	Adreno 660	Arm Mali-G78 MP14
RAM	LPDDR5	LPDDR5	LPDDR5
ML	Tensor Processing Unit	Hexagon 780 DSP	Triple NPU + DSP
Media Decode	H.264, H.265, VP9, AV1	H.264, H.265, VP9	H.264, H.265, VP9, AV1
Modem	4G LTE 5G sub-6Ghz & mmWave	4G LTE 5G sub-6Ghz & mmWave 7.5Gbps download 3Gbps upload (integrated Snapdragon X60)	4G LTE 5G sub-6Ghz & mmWave 7.35Gbps download 3.6Gbps upload (integrated Exynos 5123)
Process	5nm	5nm	5nm

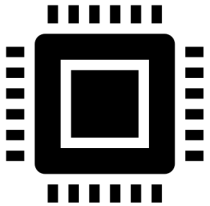
https://www.androidauthority.com/google-tensor-3060818/

Sidetrack^2: Is this sustainable?









Cascading Architectures

- Use a small (cheap) as a filter before activating a larger (expensive) model
- Example: Personal Assistant
  - Tiny embedded model to detect activation keyword
  - Send the remaining speech to the cloud model
- Example: Expensive Sensors
  - Use model that is based on data from a cheap sensor to decide when to activate the model that is based on data from an expensive sensor

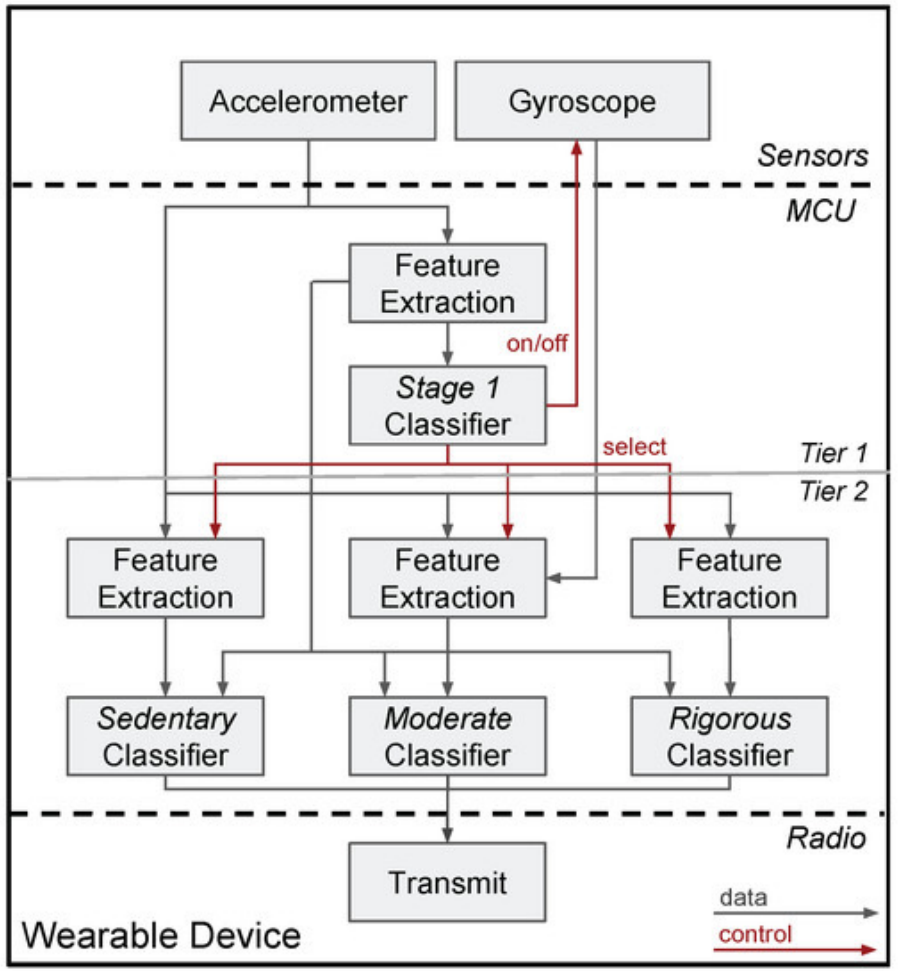


Image source: https://doi.org/10.3390/s20061655

Data Optimizations

Input Data Processing and Optimisation

- Tune the input data that the AI system uses to make predictions
- Trade predictive performance for resource-efficiency
- Input properties
  - Sampling frequency
  - Sampling resolution
  - Number of sensors
  - Activate/deactivate sensors

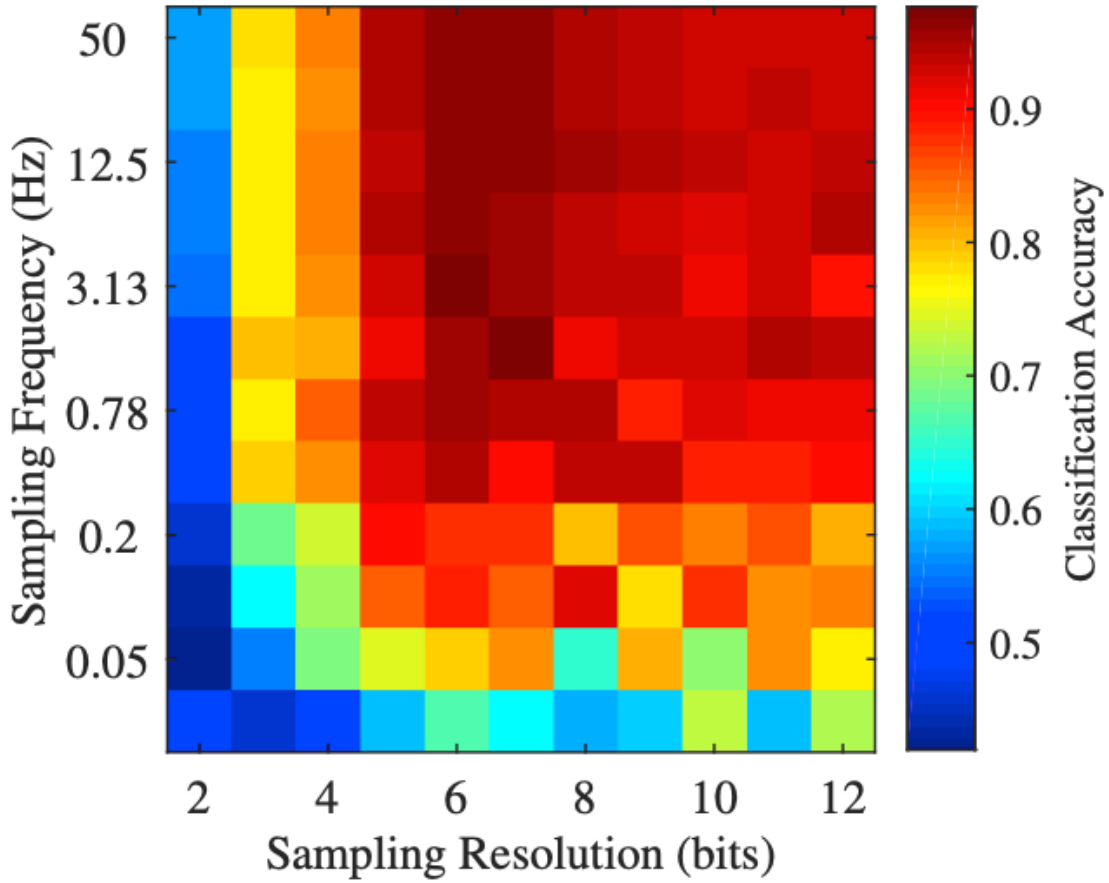
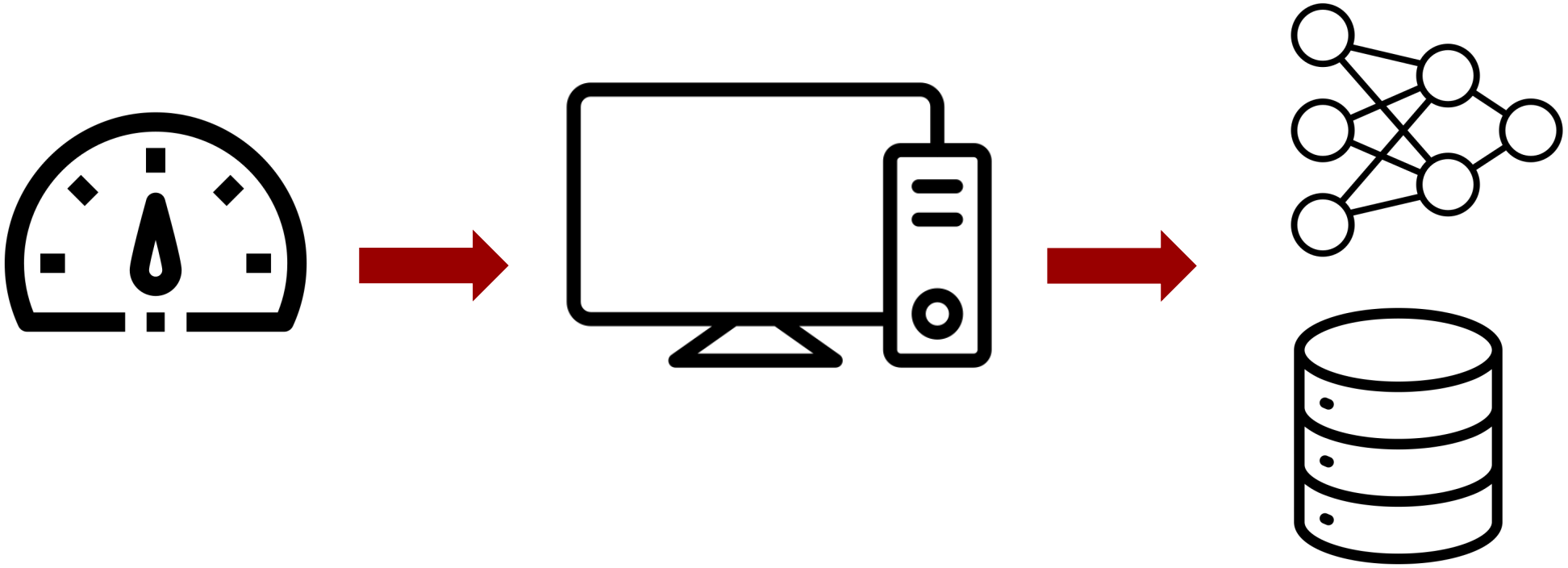


Image source: https://doi.org/10.1109/WF-IoT.2018.8355116

Data Aware Neural Architecture Search



Specific Datasets

(a) ‘Person’

(b) ‘Not-person’

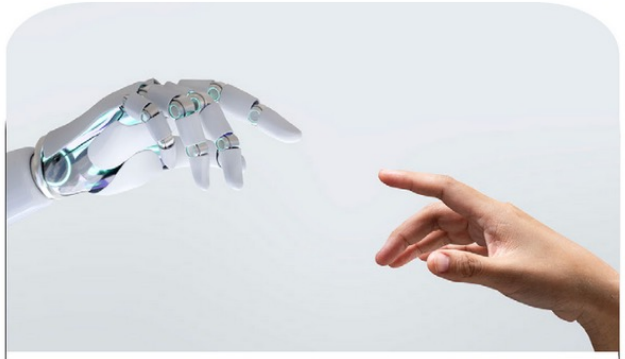
https://doi.org/10.48550/arXiv.1906.05721

Dataset Selection and Cleaning

Vision DataPerf

A multilabel image classification task, with a train set selection competition.

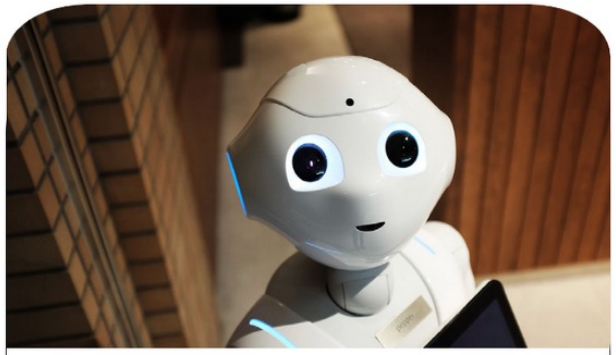
Vision



Debugging DataPerf

An image classification task, where submitters decide in which order they would clean a noisy training set.

Vision



Data Acquisition DataPerf

A benchmark for modeling data markets designed for ML applications.

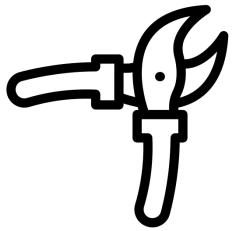
NLP

Model Optimizations

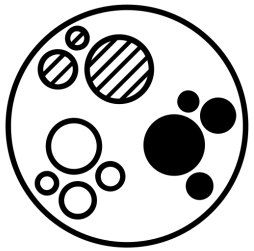
Overview

Float 32 $\rightarrow$ Int 8

Quantization



Pruning



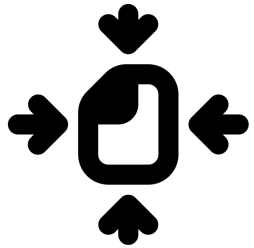
Clustering

Neural Architecture Search

Cascading Architectures

Knowledge Distillation

Early Exit Networks



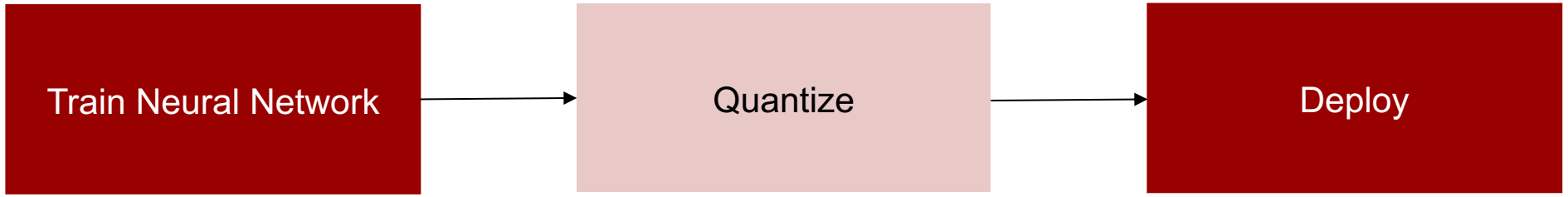
Compression

Quantization

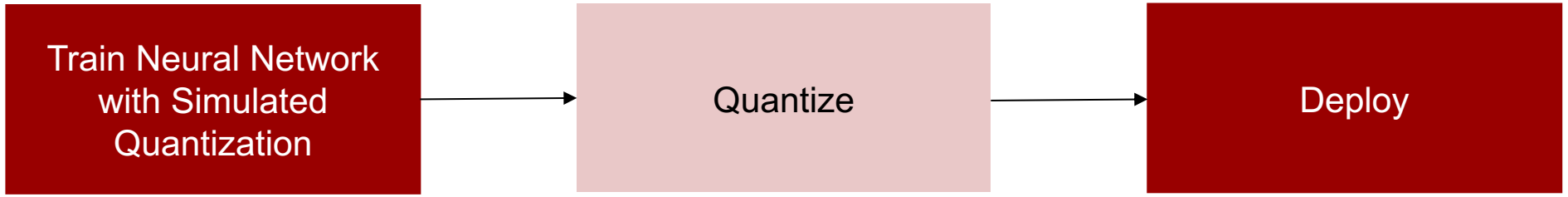
5.53	0.63	3.29
-0.43	3.21	1.05
-2.32	4.56	0.51

28	3	16
-2	16	5
-11	23	3

Post Training Quantization

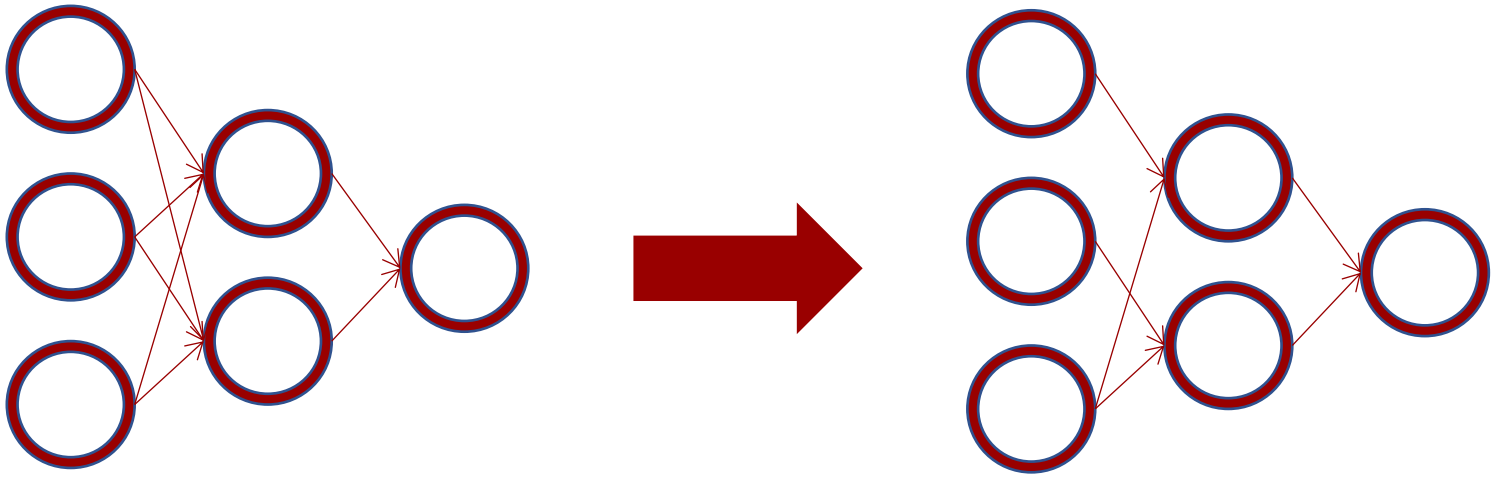


Quantization Aware Training



Quantization Tutorial

Pruning



Remove the least useful weights from model

What does “least useful” mean?

Magnitude of Weights

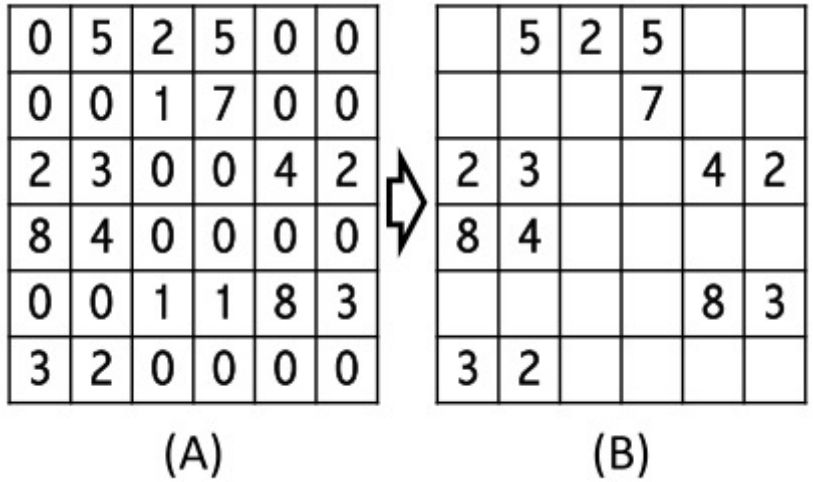
Magnitude of Activations

Gradient Based

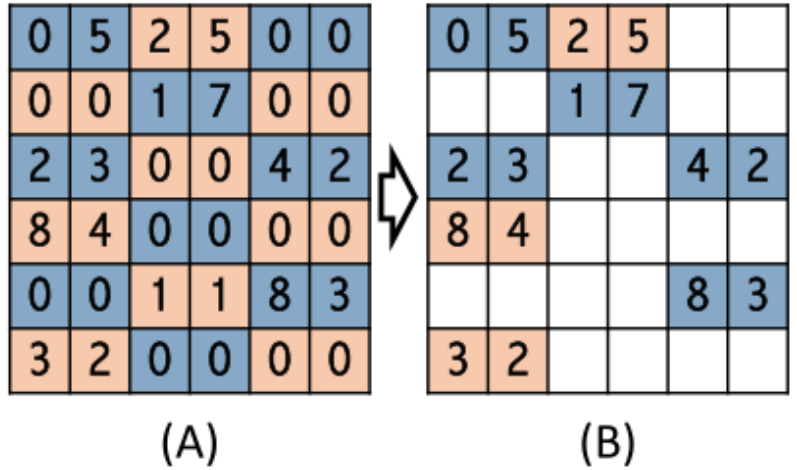
Redundancy

Energy

Unstructured and Structured Pruning



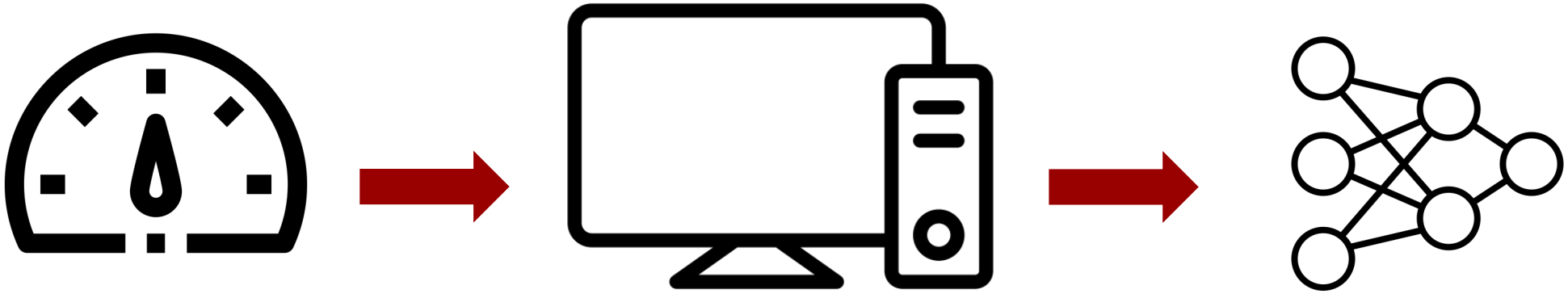
Unstructured



Structured

Pruning Tutorial

Neural Architecture Search

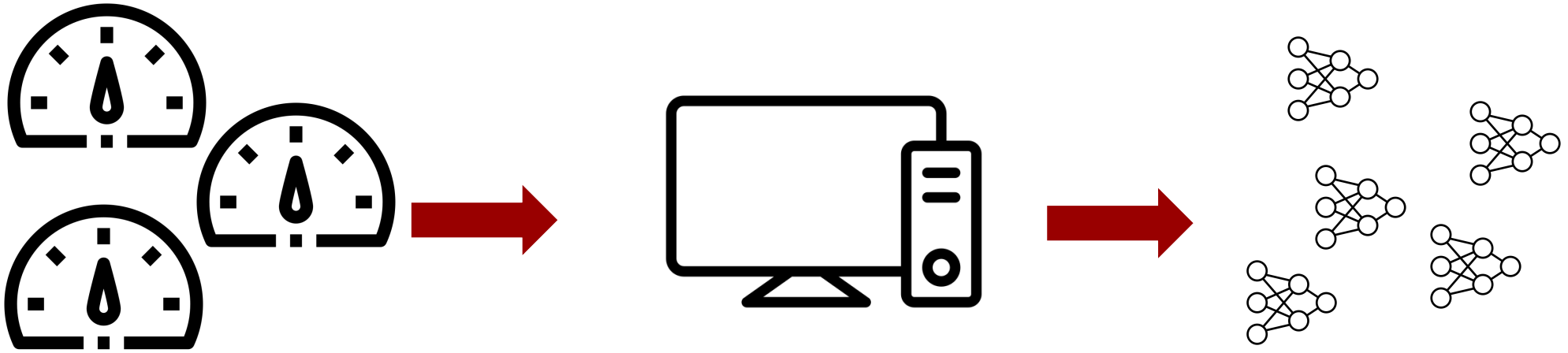


Single Objective

Traditional NAS

Single Output

Hardware Aware Neural Architecture Search



Multiple Objectives

Hardware Aware NAS

Multiple Models

Tools: Machine Learning Libraries

Tools: Compilers & Interpreters

BayesWitnesses/
m2cgen

Transform ML models into a native code (Java, C, Python, Go, JavaScript, Visual Basic, C#, R, PowerShell, PHP, Dart, Haskell,...

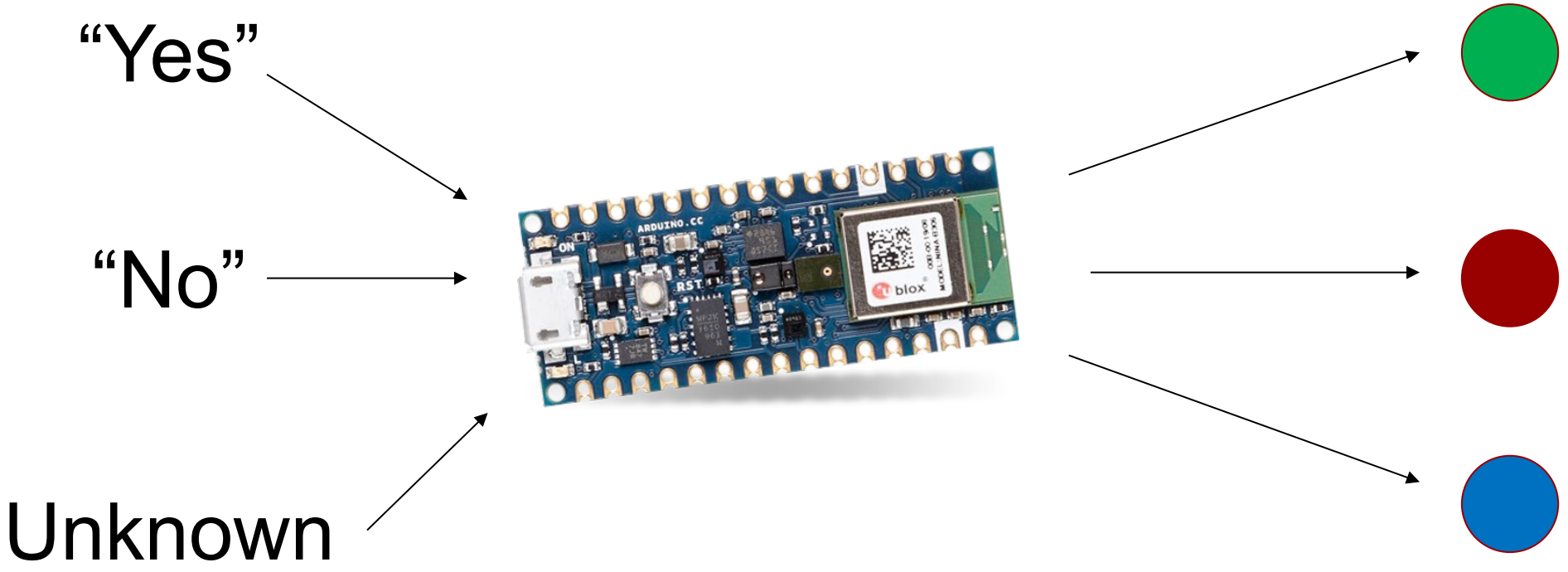
14 Contributors | 113 Used by | 2k Stars | 211 Forks

TensorFlow Lite

µtvm

Practical Showcase

Keyword Spotting



Keyword Spotting

Method Used

Float 32 → Int 8

More could have been used!

Keyword Spotting

Let’s see it in action!

Attributions

In order of first appearance:

Ai Brain by Smashing Stocks from Noun Project
Linear Regression by papergarden from Noun Project
Neural Network by Laymik from Noun Project
bot by Round Icons from Noun Project (CCBY3.0)
Data by Mello from Noun Project (CCBY3.0)
resources by Ahmad Roaayala from Noun Project (CCBY3.0)
Plane by Enya from Noun Project (CCBY3.0)
medical personnel by Andy M from Noun Project (CCBY3.0)
Server by nugra from Noun Project (CCBY3.0)

Attributions

co2 by Gung Yoga from Noun Project (CCBY3.0)
africa by HeadsOfBirds from Noun Project (CCBY3.0)
Sea by DinosoftLab from Noun Project (CCBY3.0)
Energy by shashank singh from Noun Project (CCBY3.0)
Self-driving car by Guilhem from Noun Project (CCBY3.0)
Security by Adrien Coquet from Noun Project (CCBY3.0)
Privacy by notplayink! from Noun Project (CCBY3.0)
open sign by Uciha iconic from Noun Project (CCBY3.0)
cost by abdul rohman from Noun Project (CCBY3.0)
Pruning by Brickclay from Noun Project
Clustering by Meaghan Hendricks from Noun Project
computer search by Hrbon from Noun Project

Attributions

Dominoes by Erik Arndt from Noun Project

Teacher by Adrien Coquet from Noun Project

Exit by Adrien Coquet from Noun Project

Compression by elastic1studio from Noun Project

metrics by SBTS from Noun Project

Computer by Adeel rehman from Noun Project

Neural Network by Laymik from Noun Project

Yu, Jiecao, et al. "Scalpel: Customizing dnn pruning to the underlying hardware parallelism." *ACM SIGARCH Computer Architecture News* 45.2 (2017): 548-560.

The End

Thank you
for your time!

Connect with me on LinkedIn: