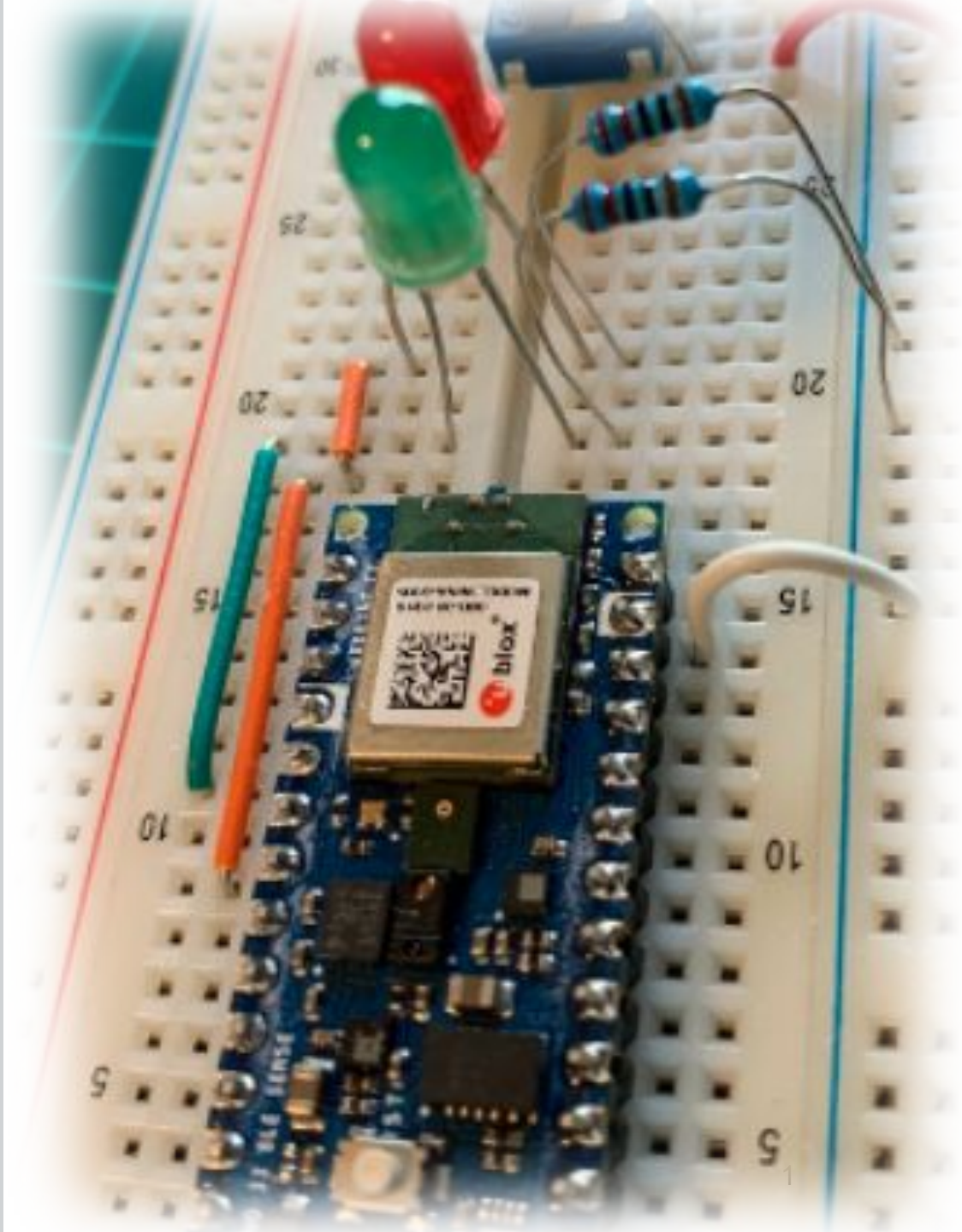
IESTI01 – TinyML

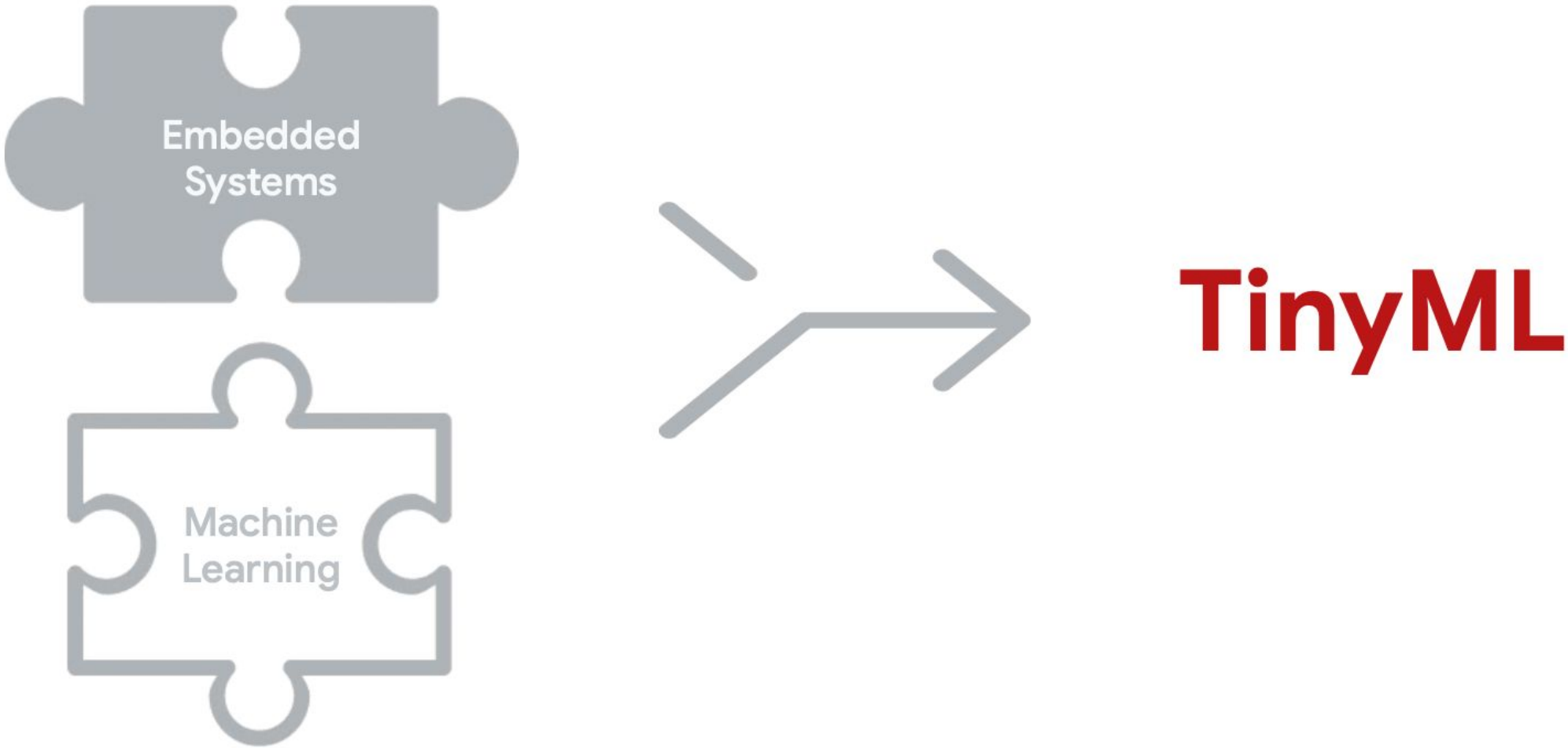
Embedded Machine Learning

3. TinyML Challenges



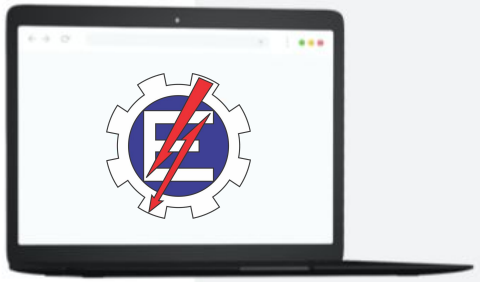
Prof. Marcelo Rovai
UNIFEI

What are the Challenges for TinyML?


Embedded
Systems
Machine
Learning
TinyML

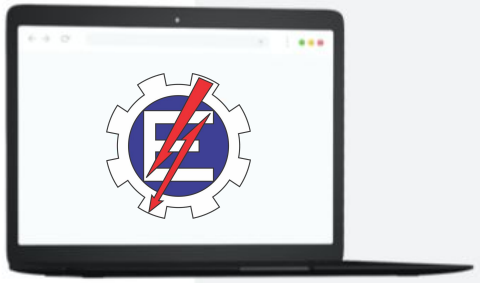
Building Blocks of Computing Hardware

Hardware



Software

Hardware



Software

Compute

Memory

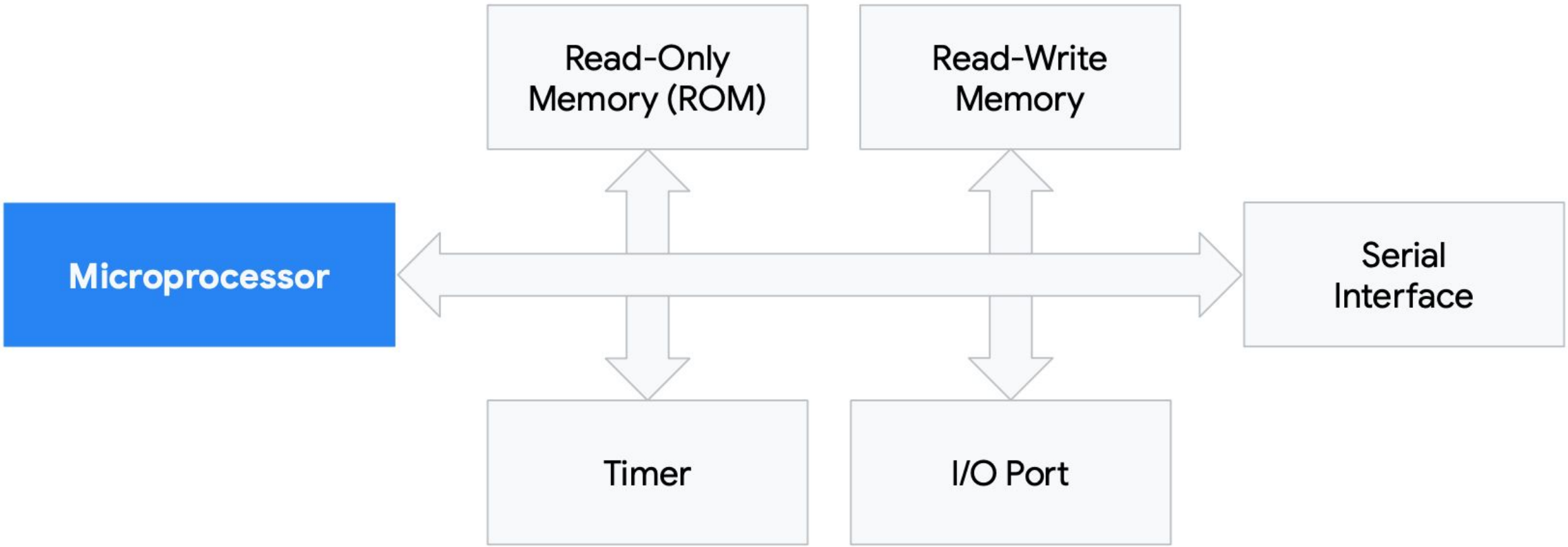
Storage

Microprocessor

v.

Microcontroller

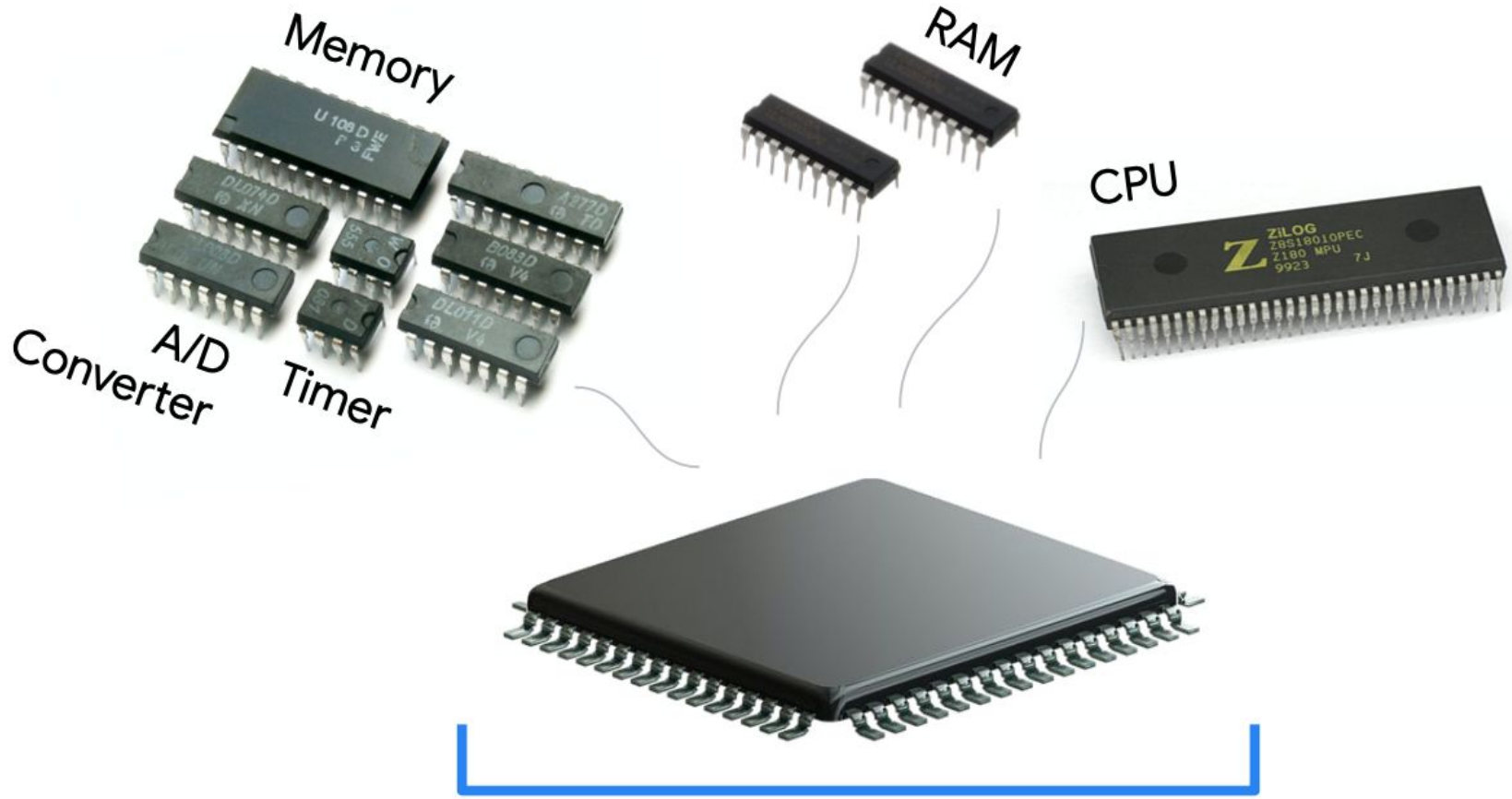
Microprocessor: only **one part** of the puzzle



Microcontroller

CPU	Read-Only Memory (ROM)	Read-Write Memory
Timer	I/O Port	Serial Interface

Microcontroller: a **complete package**



Microprocessor

- Heart of a **computer system**
- Just the processor, memory and storage are **external**
- Mainly used in **general purpose systems** like laptops, desktops and servers
- **Offers flexibility** in design
- System size is **big**

Microcontroller

- Heart of an **embedded system**
- Memory and storage are all **internal** to the system
- Mainly used in **specialized, fixed function systems** like phones, MP3 players, etc.
- **Limited flexibility** in design
- System size is **tiny**

Orders of Magnitude Difference

	Microprocessor	>	Microcontroller	
Platform				Nano
Compute	1GHz–4GHz	~10X	1MHz–400MHz	64MHz
Memory	512MB–64GB	~10000X	2KB–512KB	256KB
Storage	64GB–4TB	~100000X	32KB–2MB	1MB
Power	30W–100W	~1000X	150µW–23.5mW	

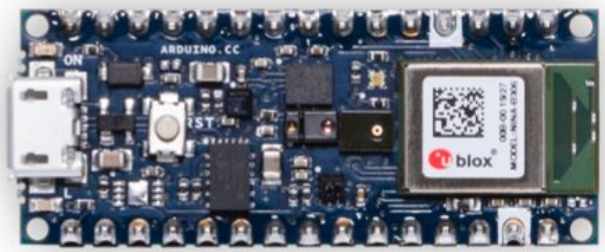
Implications

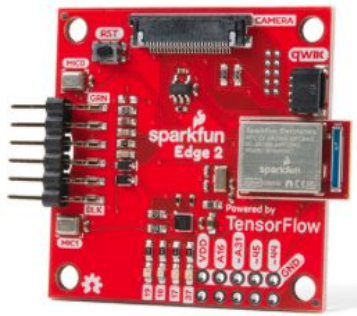
- How complicated is the running task?
- How much memory does it need to have?
- How long does the job have to perform?

Microcontroller

1MHz-400MHz
2KB - 512KB
32KB - 2MB
150μW–23.5mW

Computing Hardware





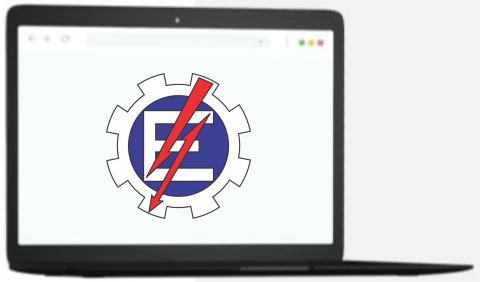
Computing Hardware

	Board	MCU / ASIC	Clock	Memory	Sensors	Radio
	Himax WE-I Plus EVB	HX6537-A 32-bit EM9D DSP	400 MHz	2MB flash 2MB RAM	Accelerometer, Mic, Camera	None
	Arduino Nano 33 BLE Sense	32-bit nRF52840	64 MHz	1MB flash 256kB RAM	Mic, IMU, Temp, Humidity, Gesture, Pressure, Proximity, Brightness, Color	BLE
	SparkFun Edge 2	32-bit ArtemisV1	48 MHz	1MB flash 384kB RAM	Accelerometer, Mic, Camera	BLE
	Espressif EYE	32-bit ESP32-D0WD	240 MHz	4MB flash 520kB RAM	Mic, Camera	WiFi, BLE

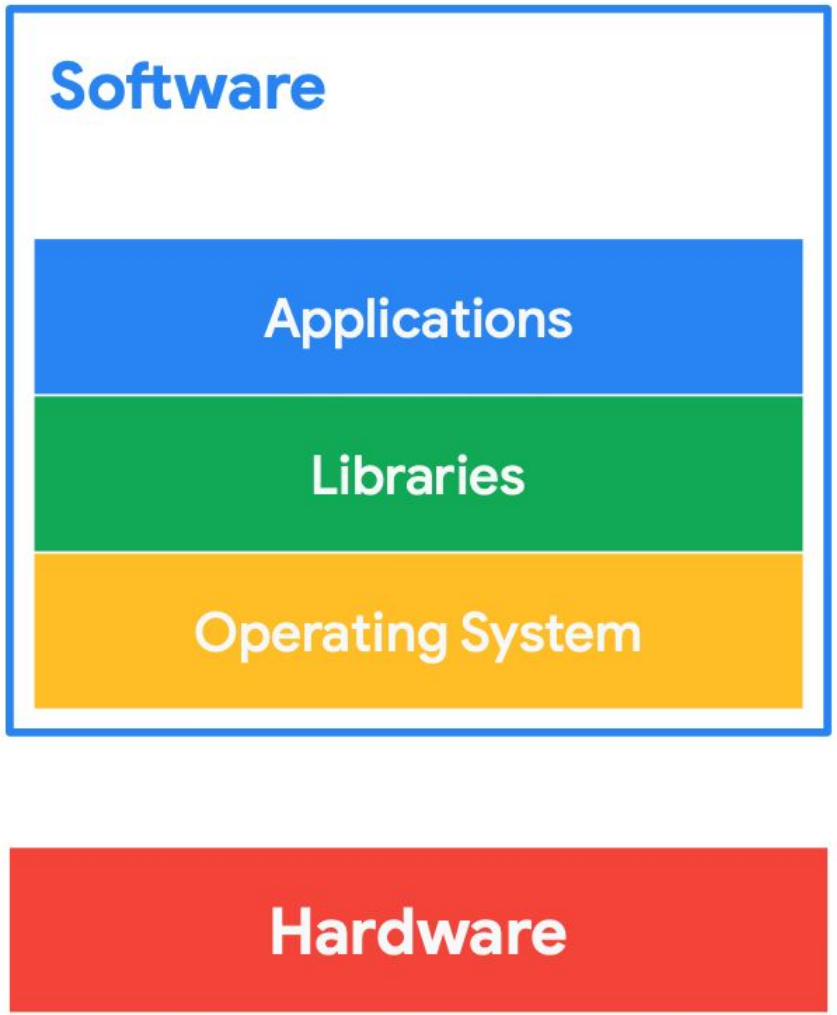
Computing Hardware

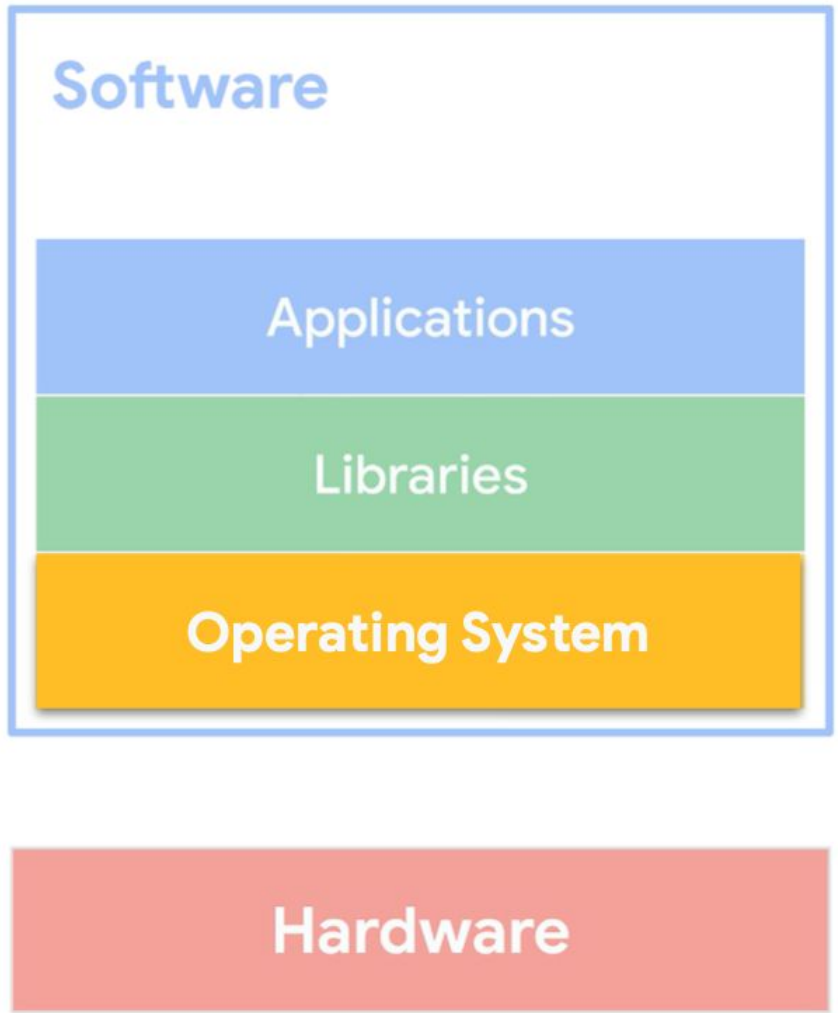
	Board	MCU / ASIC	Clock	Memory	Sensors	Radio
	Himax WE-I Plus EVB	HX6537-A 32-bit EM9D DSP	400 MHz	2MB flash 2MB RAM	Accelerometer, Mic, Camera	None
	**Arduino Nano 33 BLE Sense**	**32-bit nRF52840**	**64 MHz**	**1MB flash 256kB RAM**	**Mic, IMU, Temp, Humidity, Gesture, Pressure, Proximity, Brightness, Color**	**BLE**
	SparkFun Edge 2	32-bit ArtemisV1	48 MHz	1MB flash 384kB RAM	Accelerometer, Mic, Camera	BLE
	Espressif EYE	32-bit ESP32-D0WD	240 MHz	4MB flash 520kB RAM	Mic, Camera	WiFi, BLE

Hardware

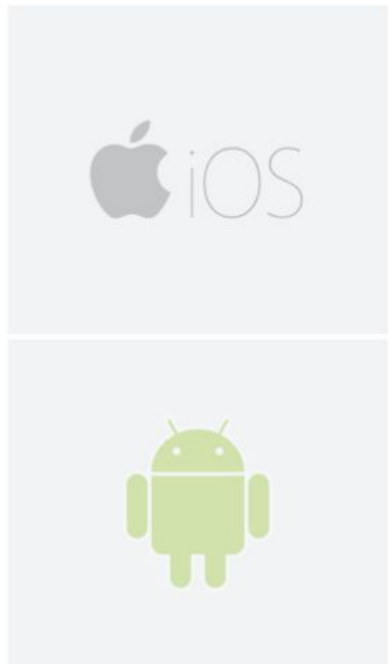


Software

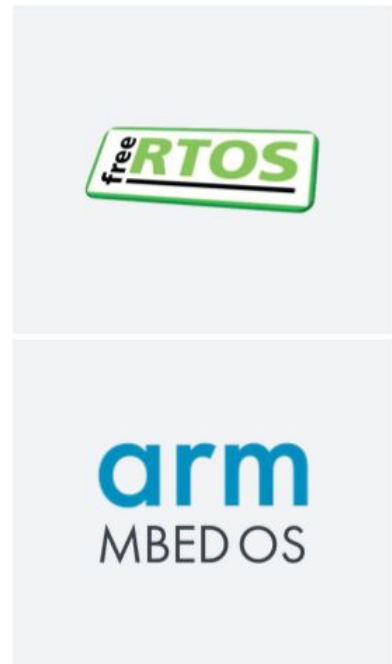

Software
Applications
Libraries
Operating System
Hardware


Software
Applications
Libraries
Operating System
Hardware

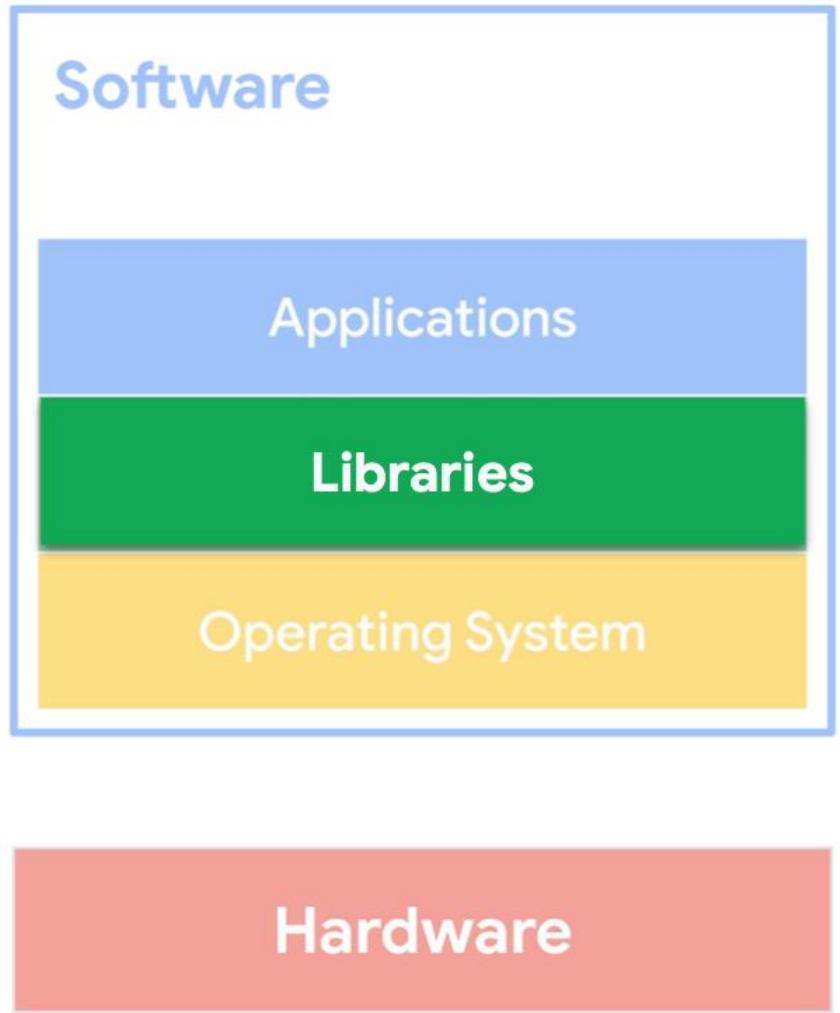
Widely Used Operating Systems

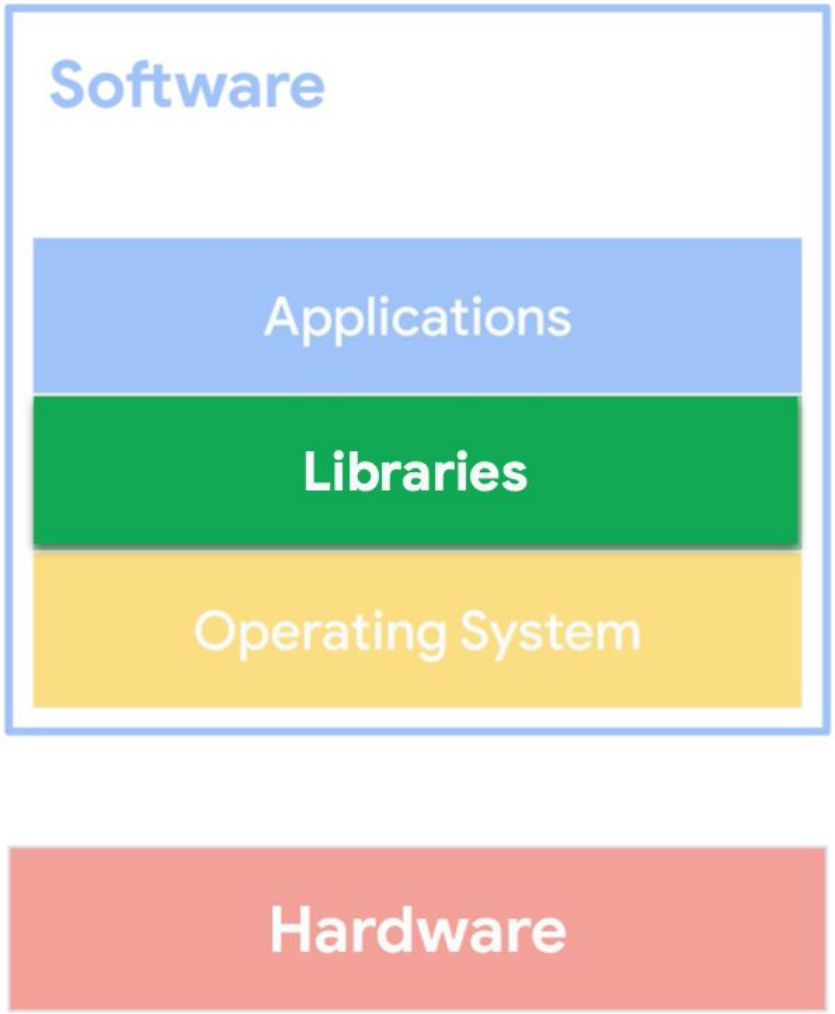


Mobile OS



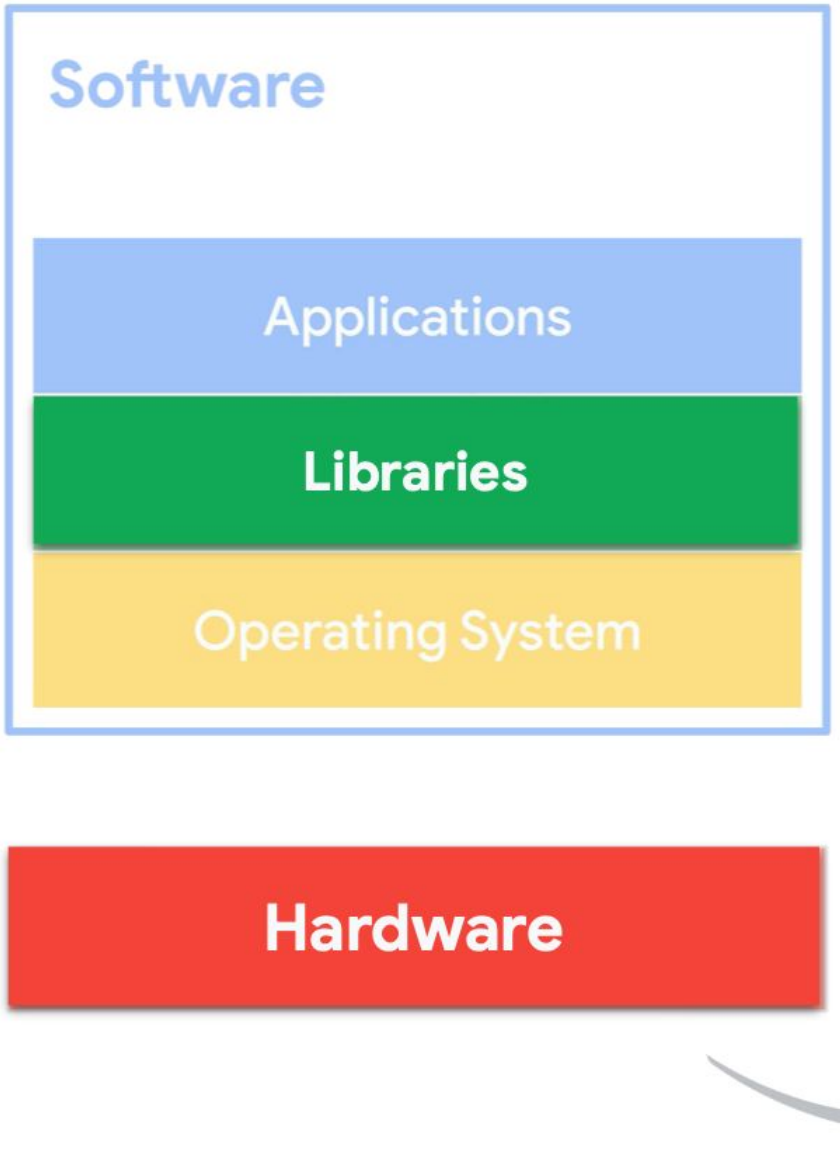
Embedded Sys.


Software
Applications
Libraries
Operating System
Hardware



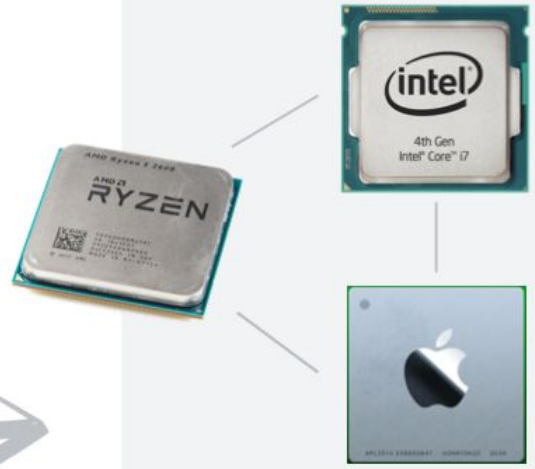

```
import numpy as np

for x in range(10):
    np.SaveTheWorld()
```

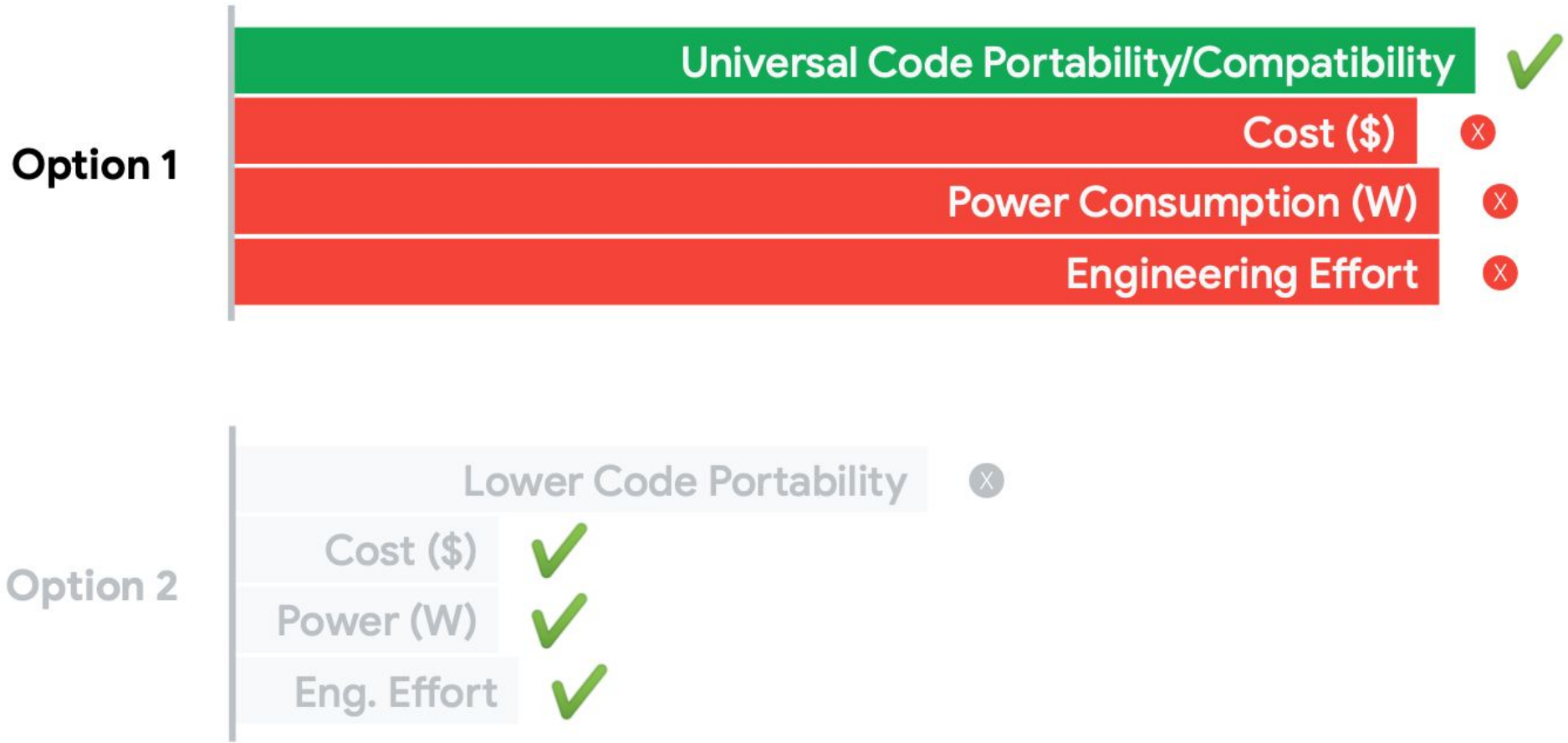


Portability Opportunity

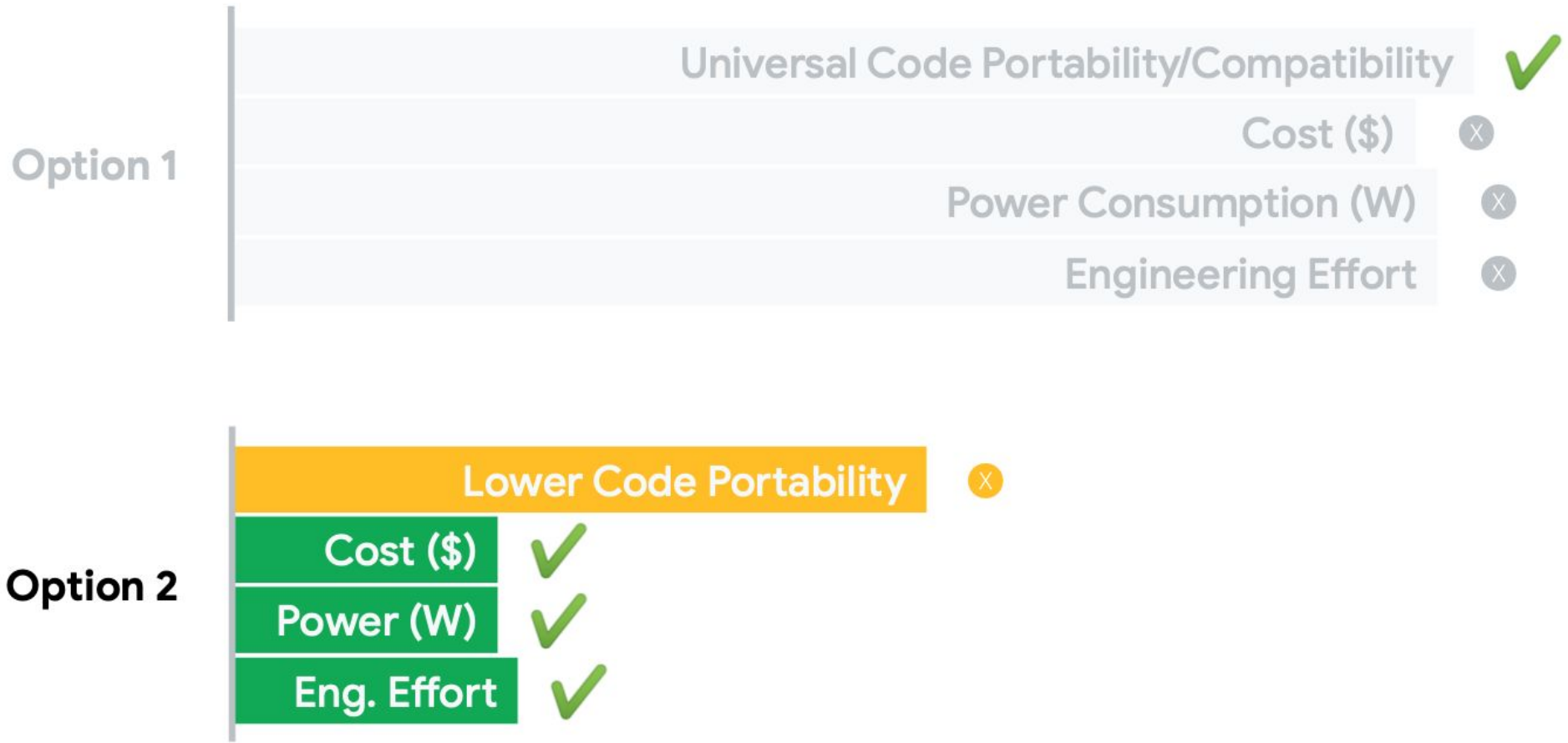
Able to execute the same code on different microprocessor hardware and architectures.



Portability **Trade-offs**

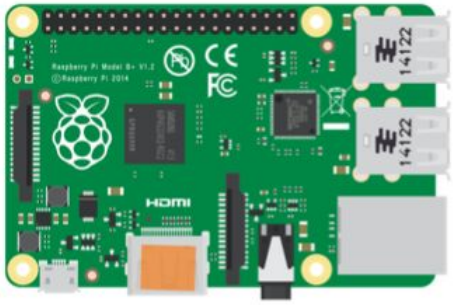


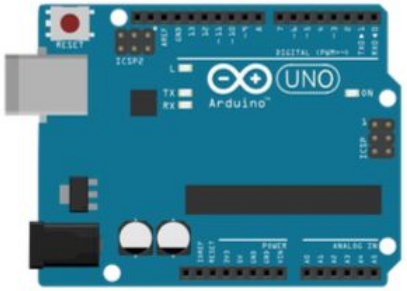
Portability **Trade-offs**



Portability **Trade-offs**

Sacrifice portability across systems for efficiency in system performance and power efficiency





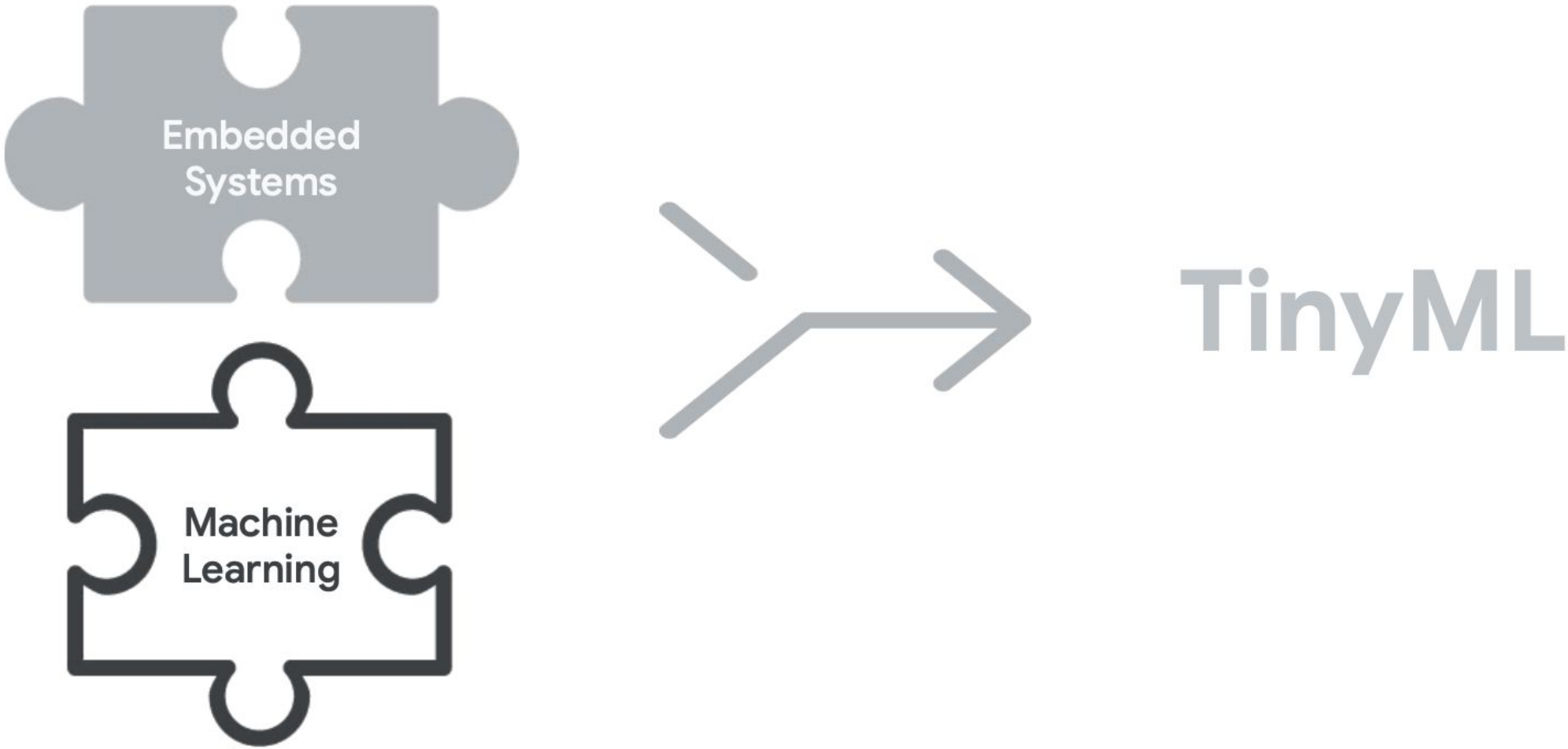
✗	✓
Specific HW Implementation of a Library	

Option 2

- Lower Code Portability ✗
- Cost ($) ✓
- Power (W) ✓
- Eng. Effort ✓

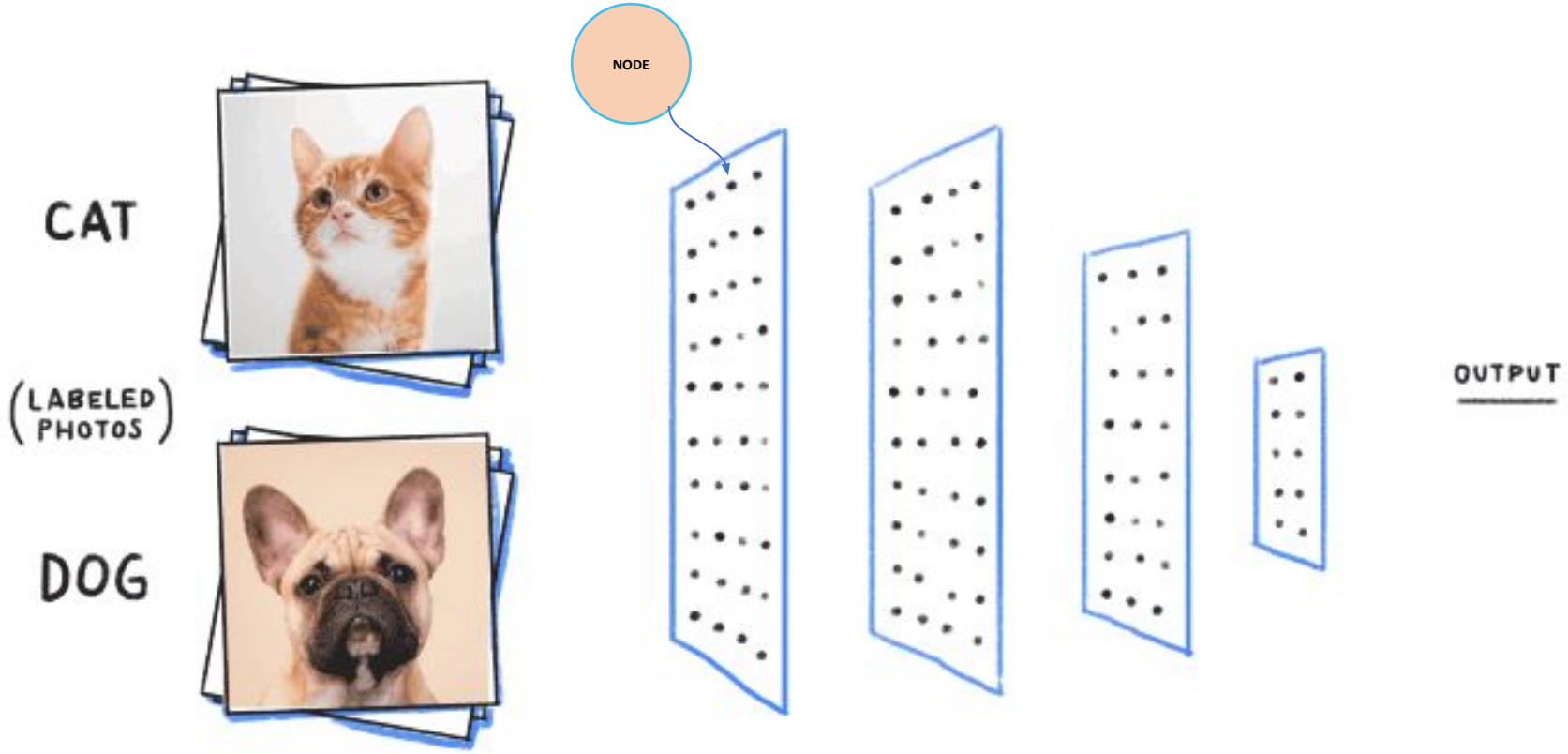
Summary

- **Embedded hardware** is extremely limited in performance, power consumption and storage
- **Embedded software** is not as portable and flexible as mainstream computing

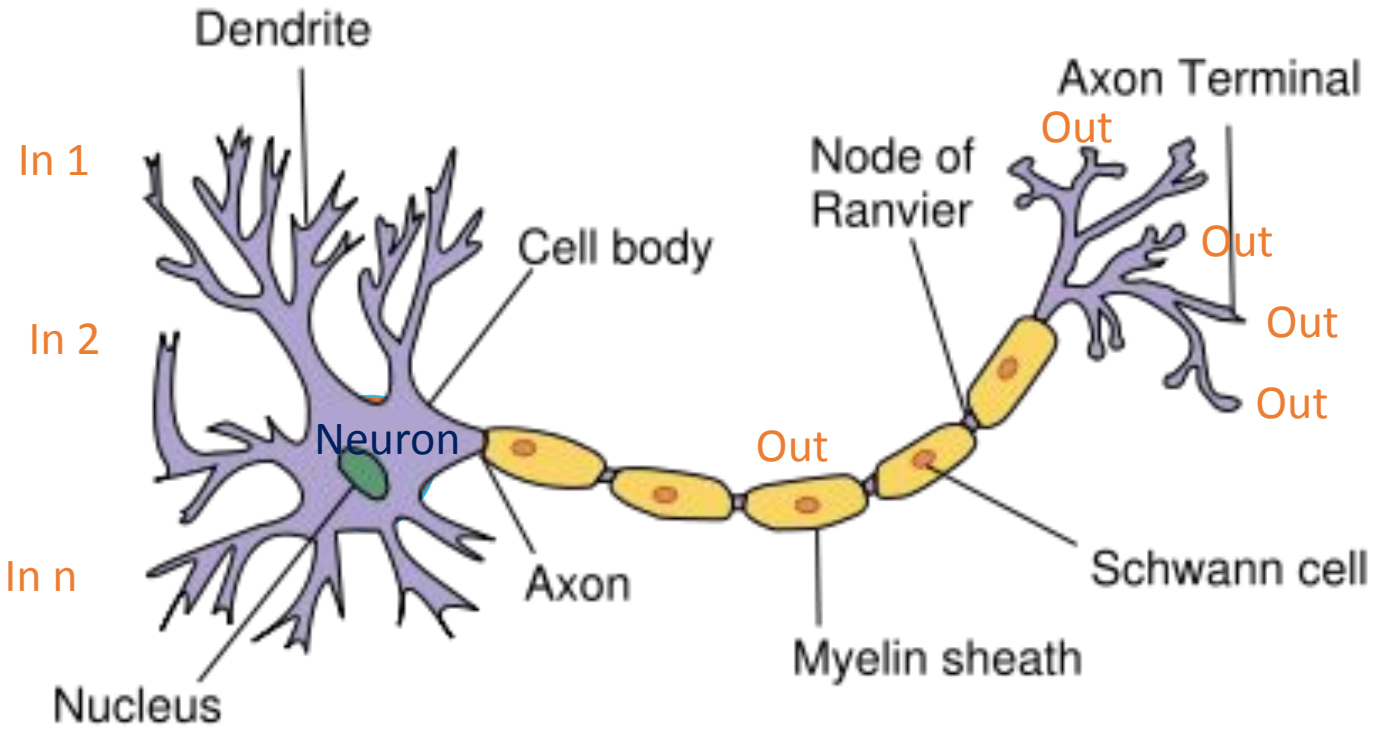

Embedded
Systems
Machine
Learning
TinyML

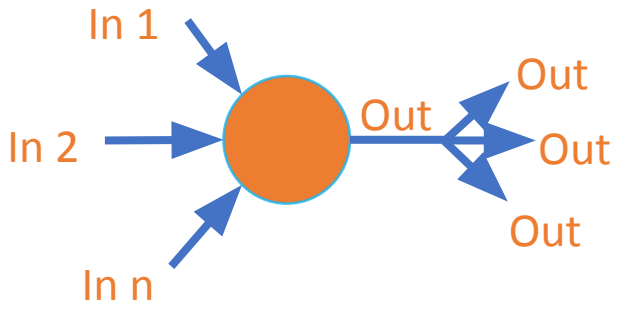
(Deep) Machine Learning

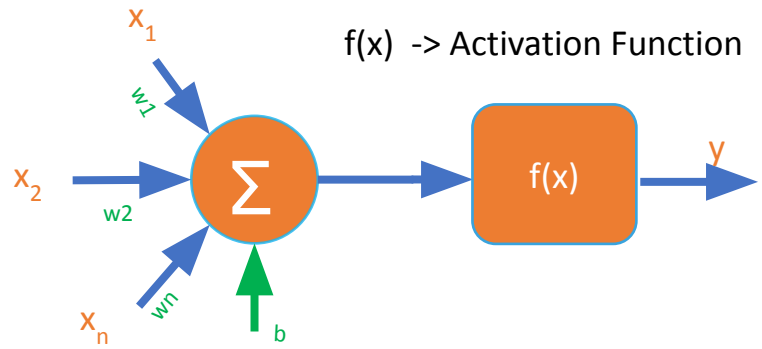
Deep Learning: Subset of Machine Learning in which multilayered neural networks learn from vast amounts of data



Neuron (Perceptron)





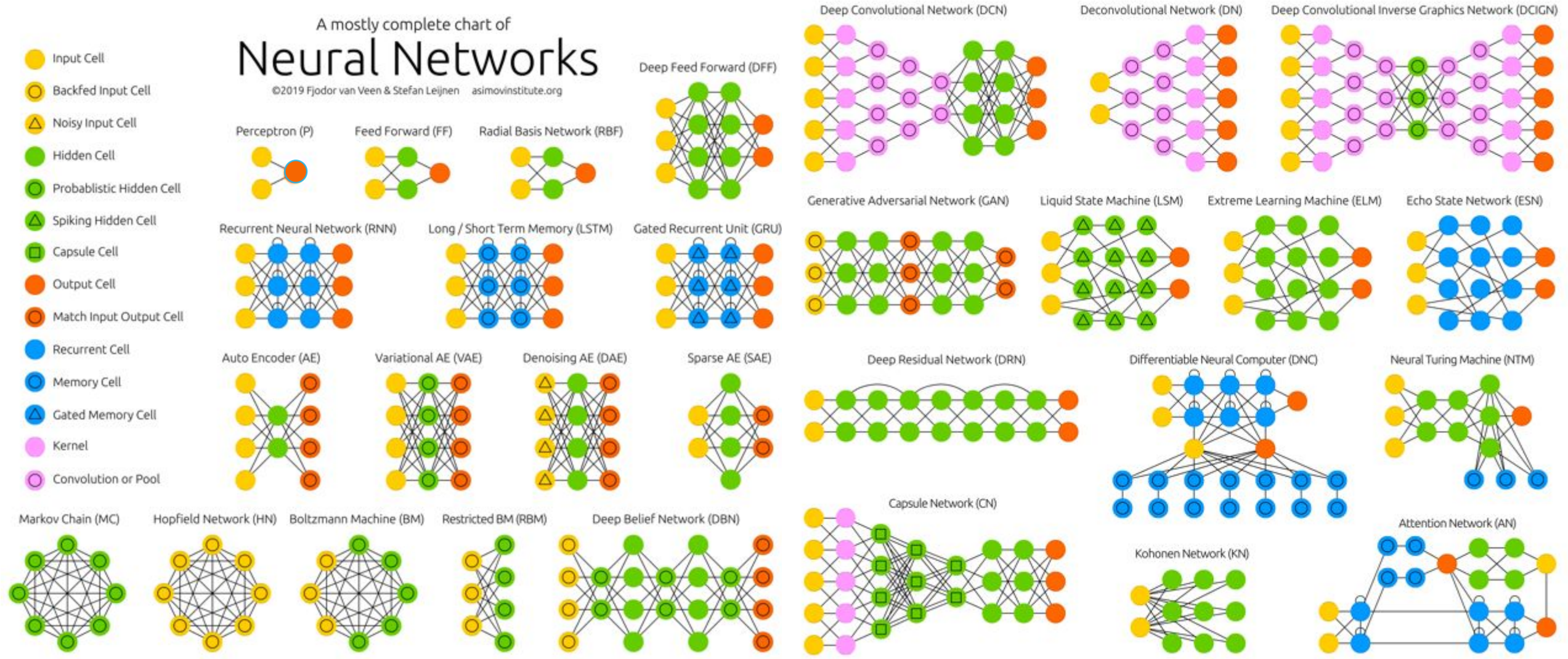


f(x) -> Activation Function

Parameters

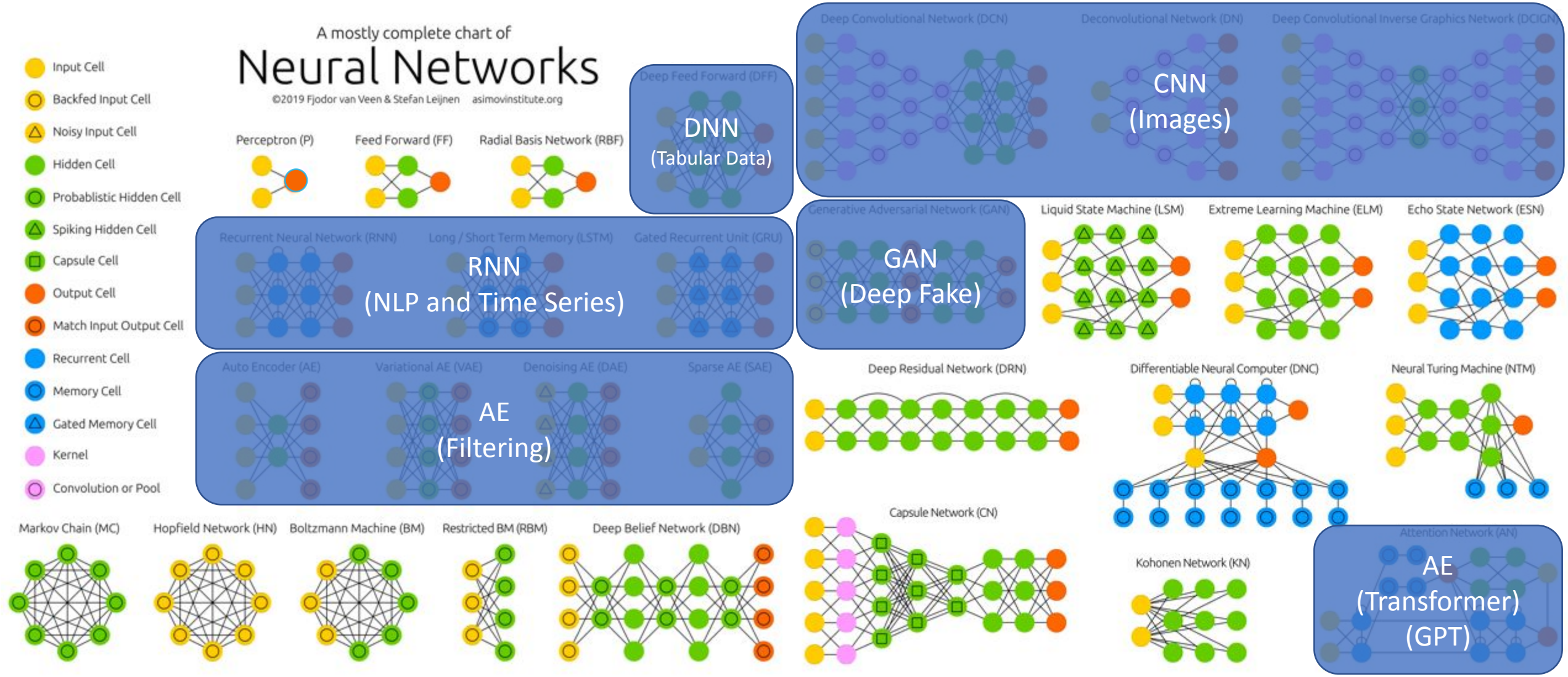
$$y = f(\sum_{i=1}^{n} x_i w_i + b)$$

The Neural Network Model Architecture



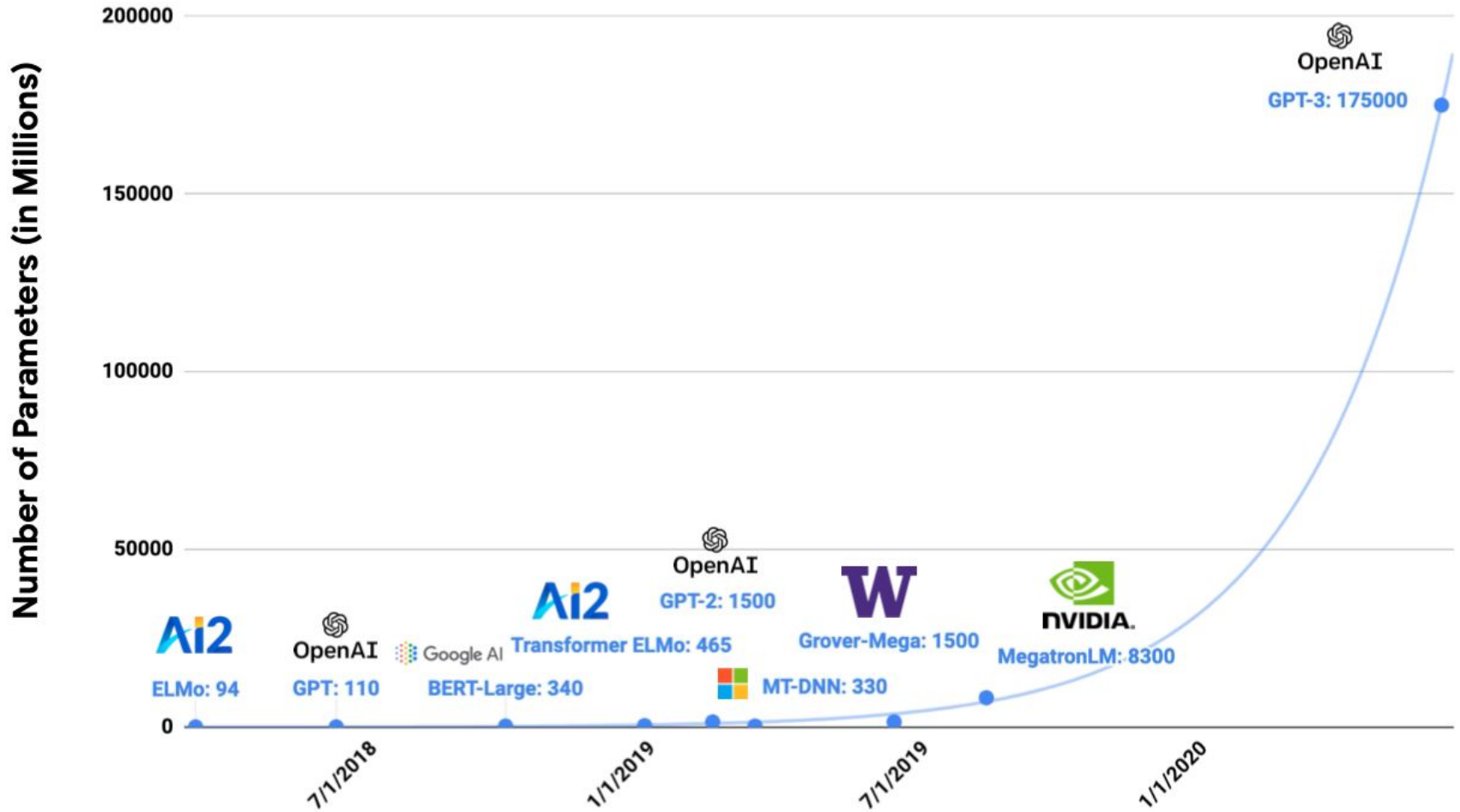
https://www.asimovinstitute.org/neural-network-zoo/

The Neural Network Model Architecture

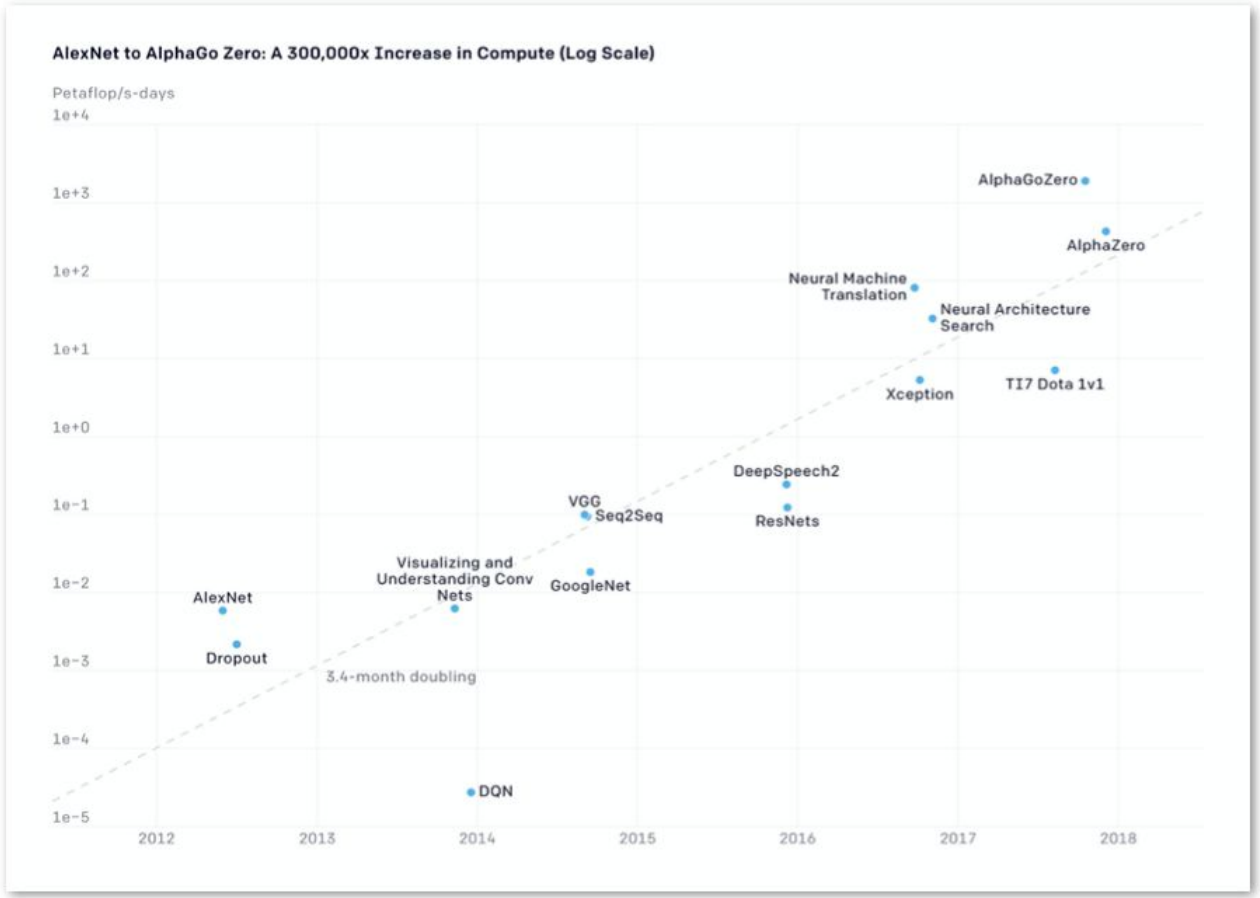


https://www.asimovinstitute.org/neural-network-zoo/

ML Model Size Growth



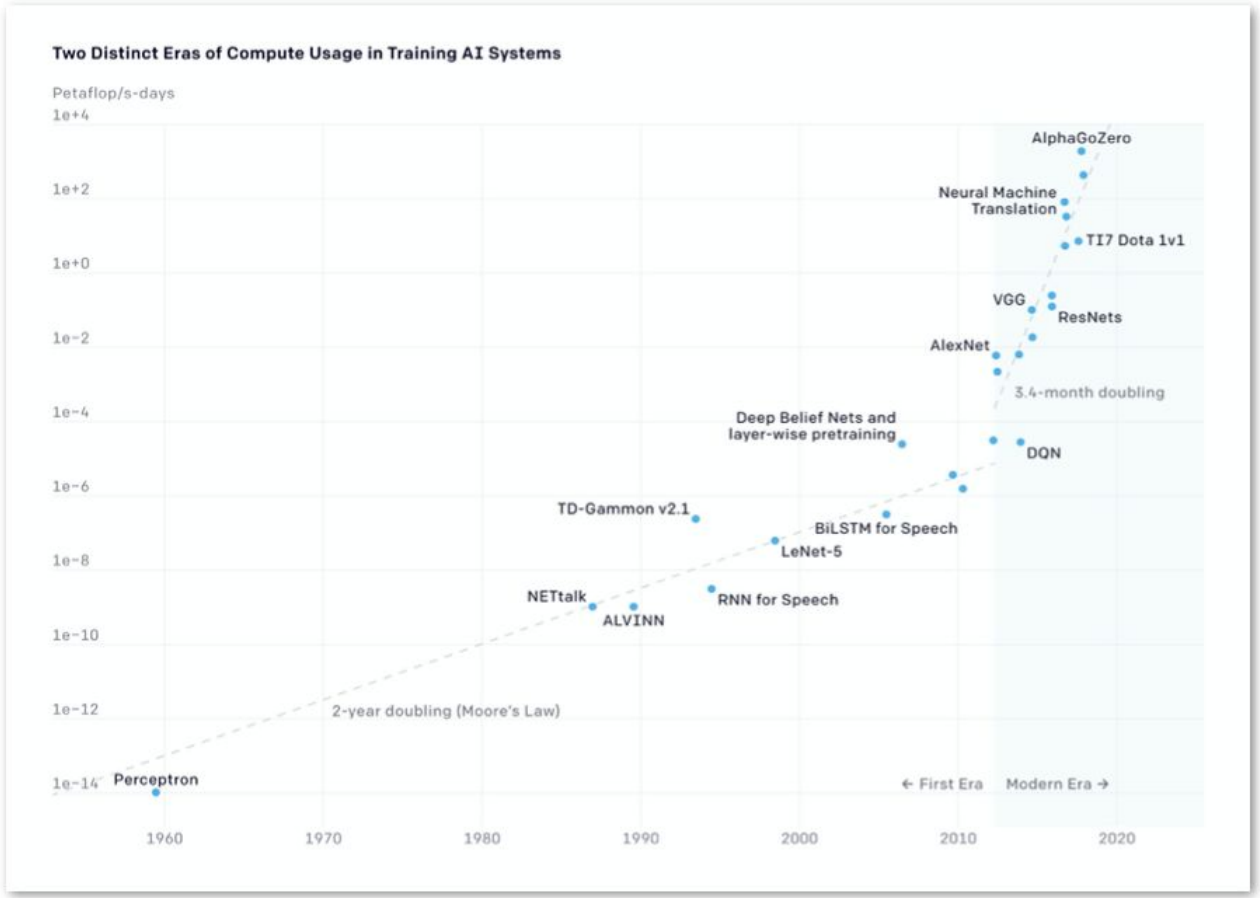
ML Compute Needs (2012 to Present Day)



Source: https://openai.com

In recent years, **computing needs grew by 300,000x** to train the machine learning models that are widely deployed in the industry

ML Compute Needs (from the 1960s)



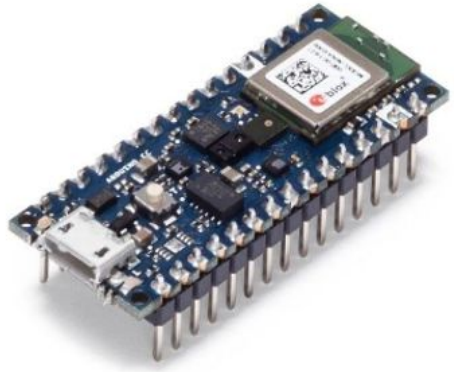
Source: https://openai.com

In recent years, the amount of computing needed has grown remarkably fast.

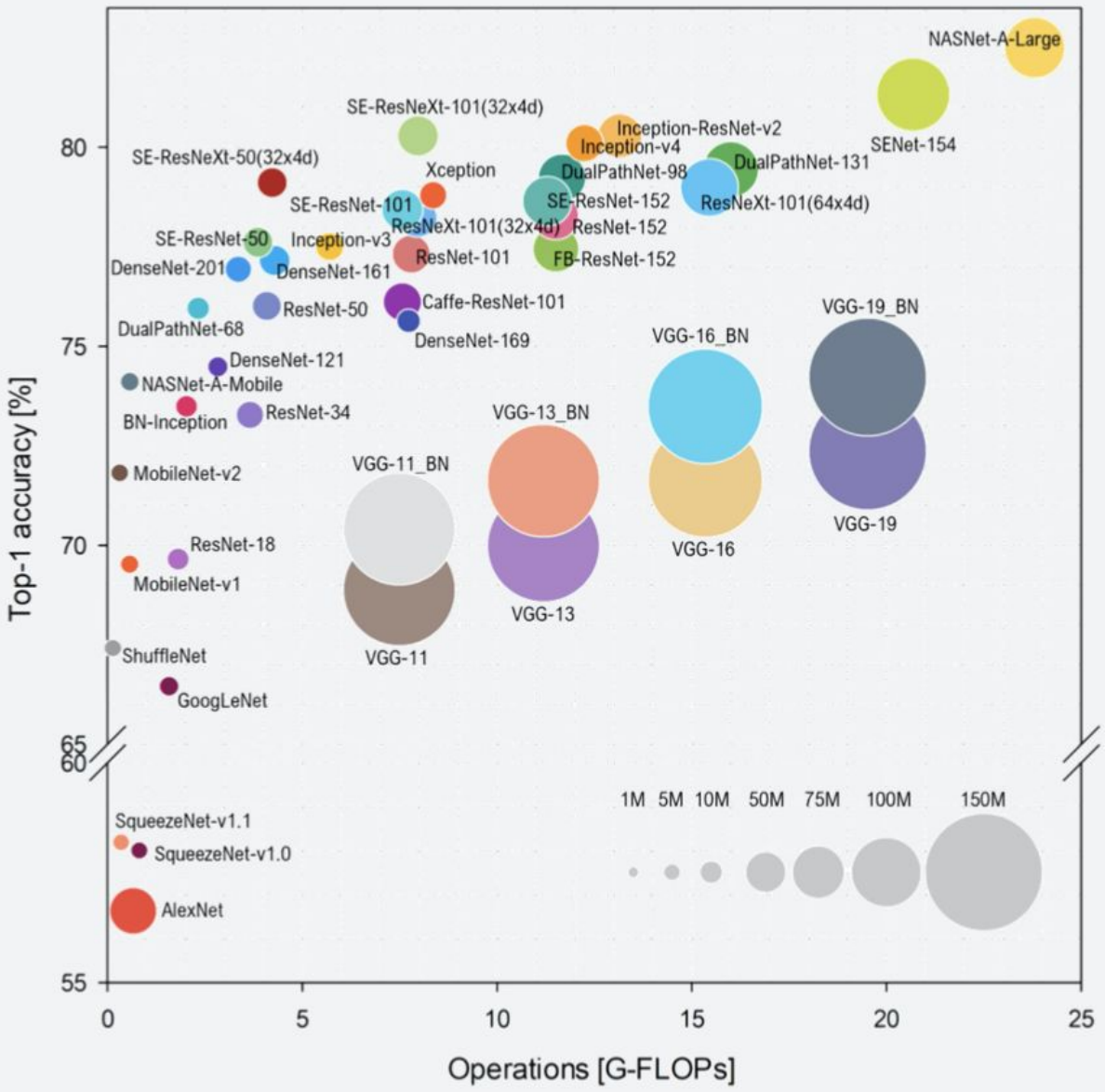
Compute requirements are **doubling nearly every 3 to 4 months**

Cloud TPU

TinyML



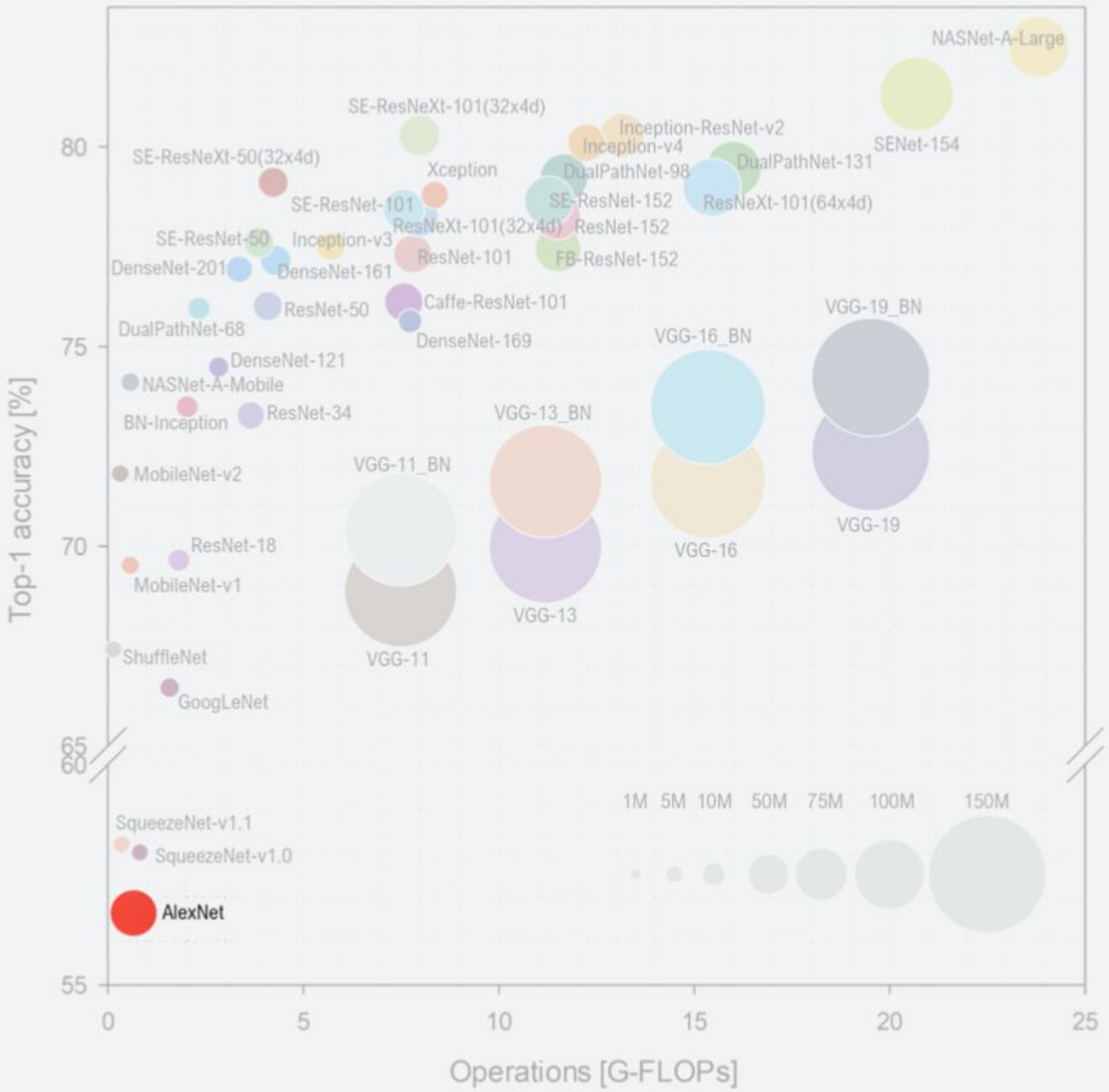
ML Model Evolution



Source: S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018

ML Model Evolution

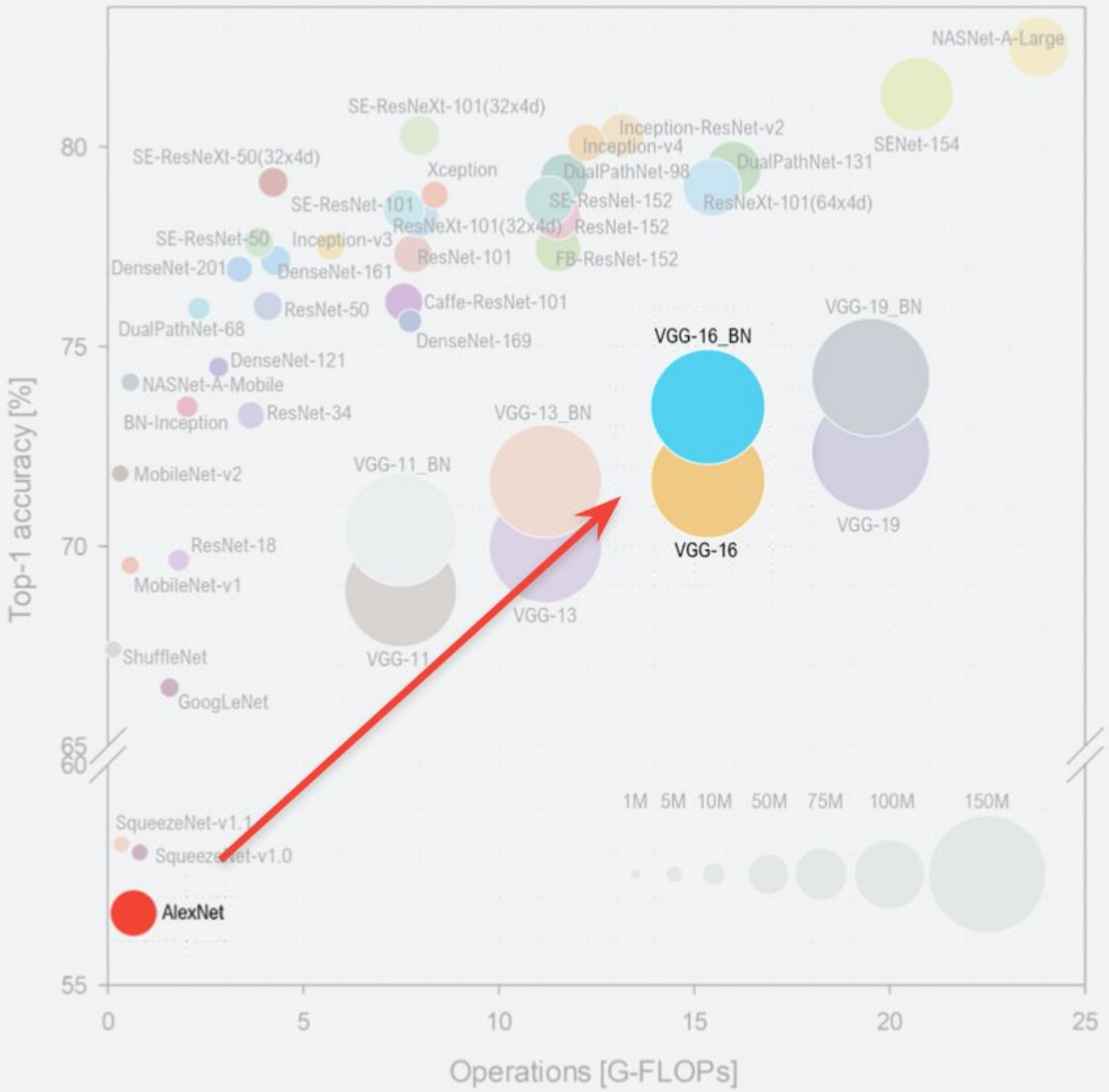
- **AlexNet** (2012)
  - 57.1% accuracy
  - 61MB in size



Source: S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018

ML Model Evolution

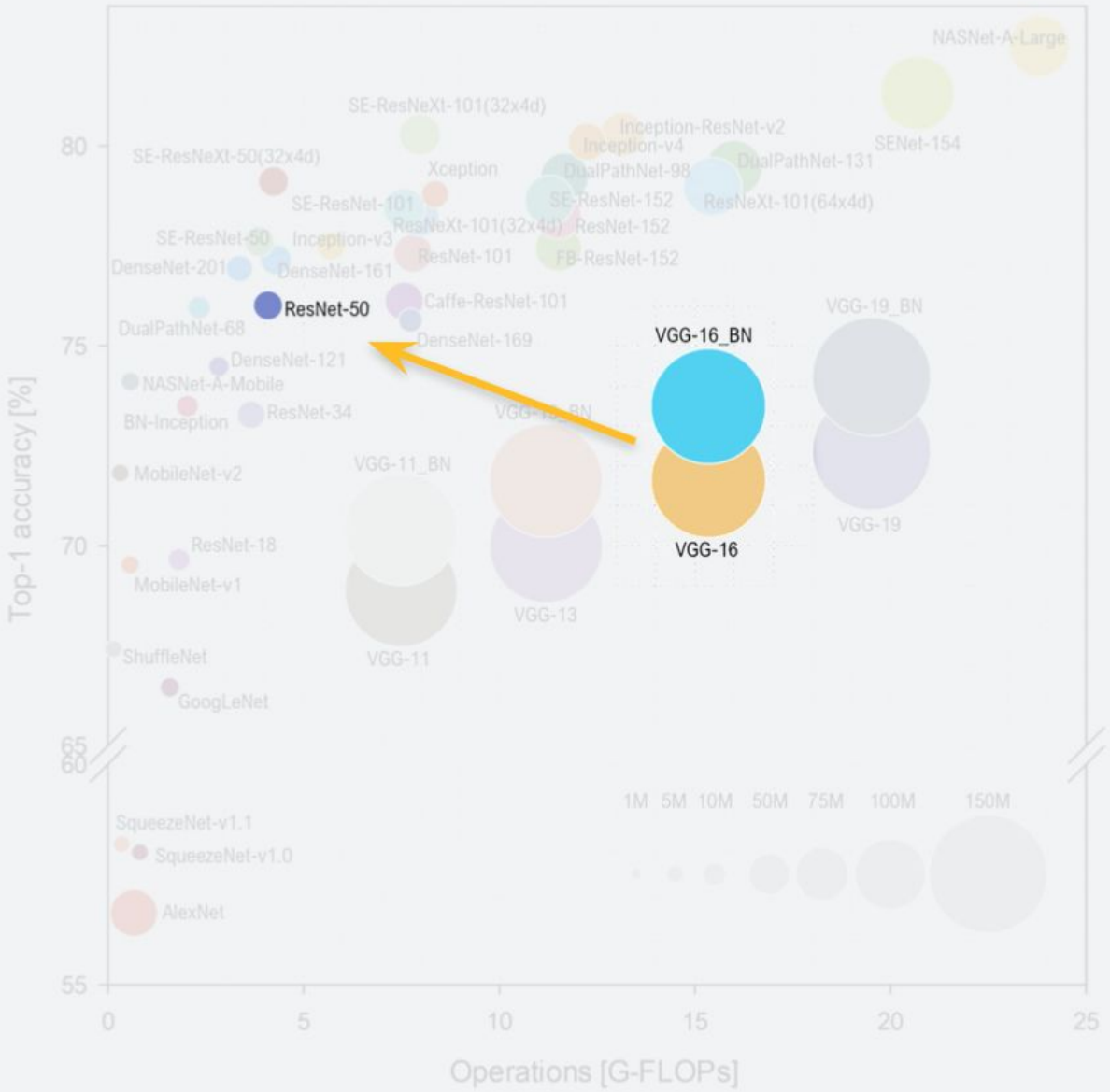
- **VGGNet** (2014) [VGG-16]
  - **71.5%** accuracy
  - **528MB** in size



Source: S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018

ML Model Evolution

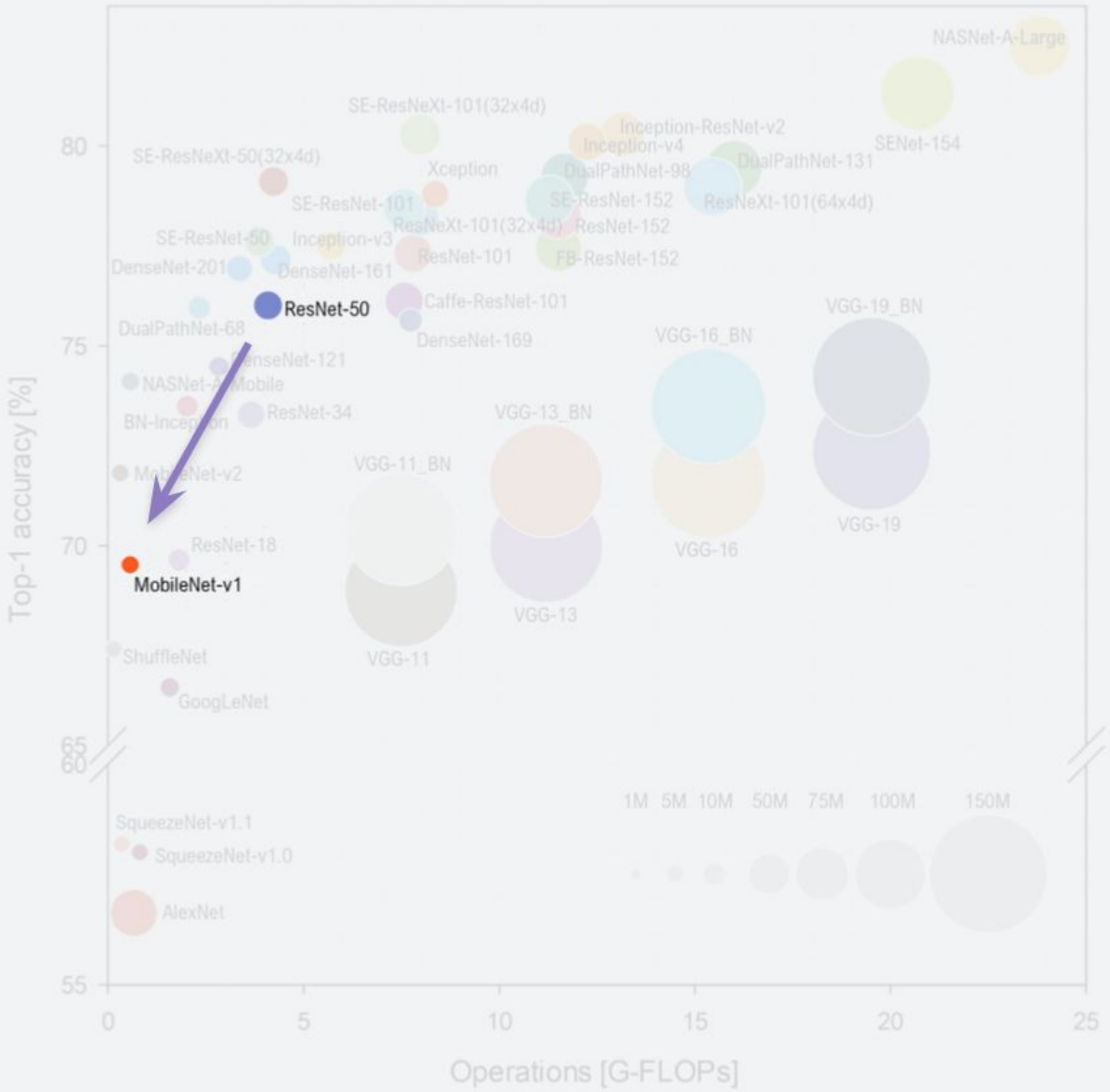
- **ResNet** (2015)
  - **75.8%** accuracy
  - **22.7MB** in size



Source: S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018

ML Model Evolution

- **MobileNet** (2015)
  - ***MobileNetv1***
    - **70.6%** accuracy
    - **16.9MB** in size



Source: S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018

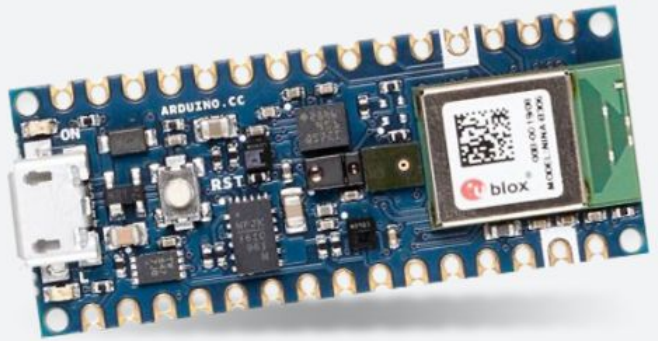
ML Model Evolution

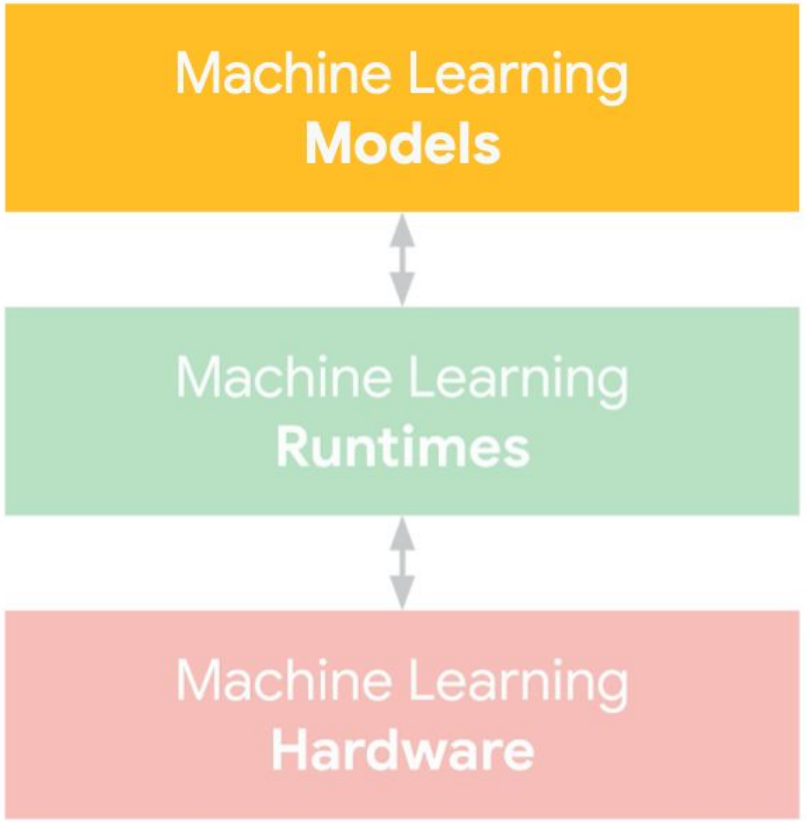
- **MobileNet** (2015)
  - ***MobileNetv1***
    - 70.6% accuracy
    - 16.9MB in size

Source: S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018

Problem:

Our board (in your kit for Course) only has **256KB** of RAM (memory) yet ***MobileNetv1*** needs **16.9MB**!





Model Compression Techniques

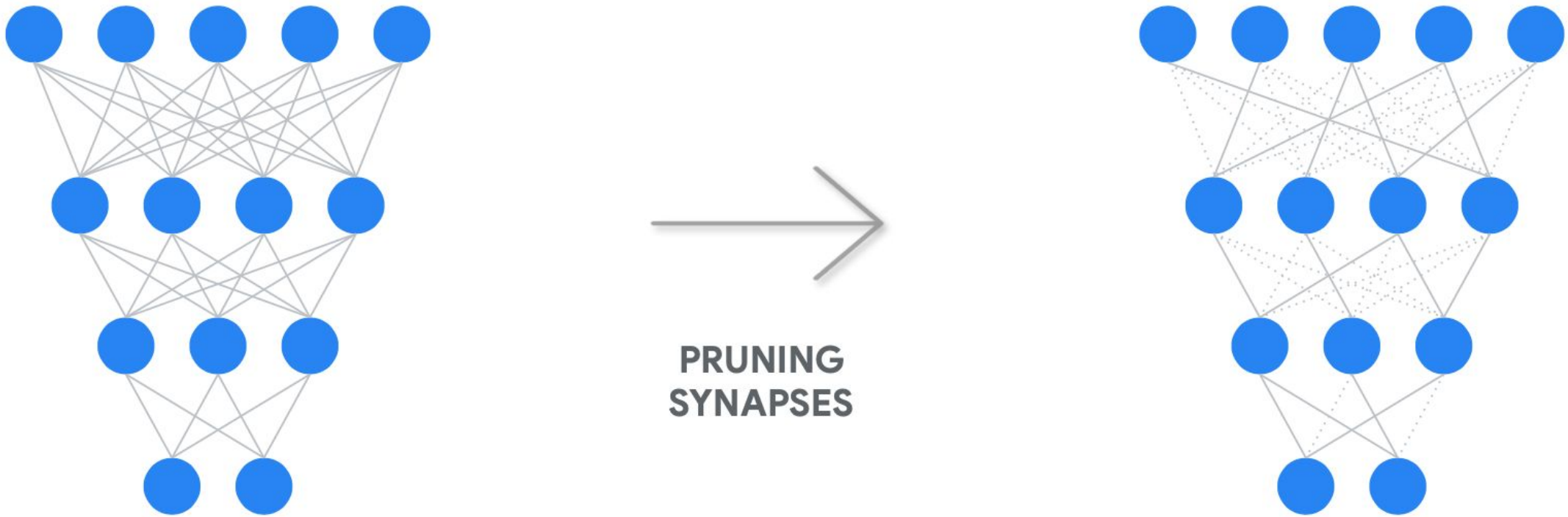
Pruning

Quantization

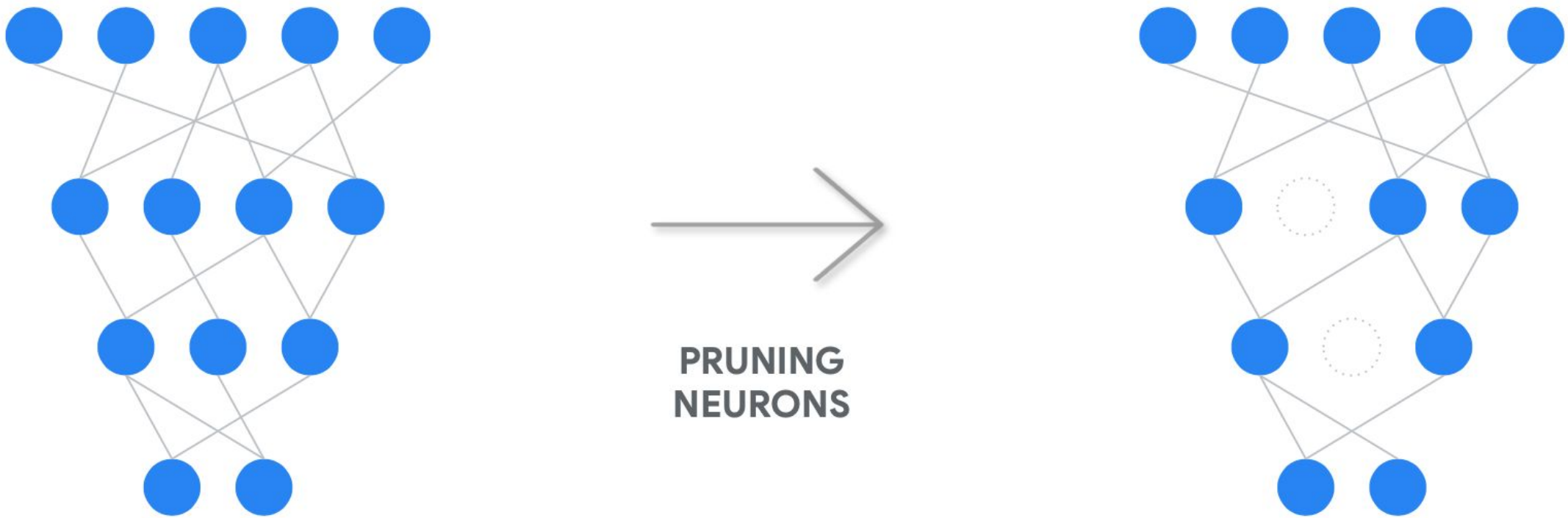
Knowledge Distillation

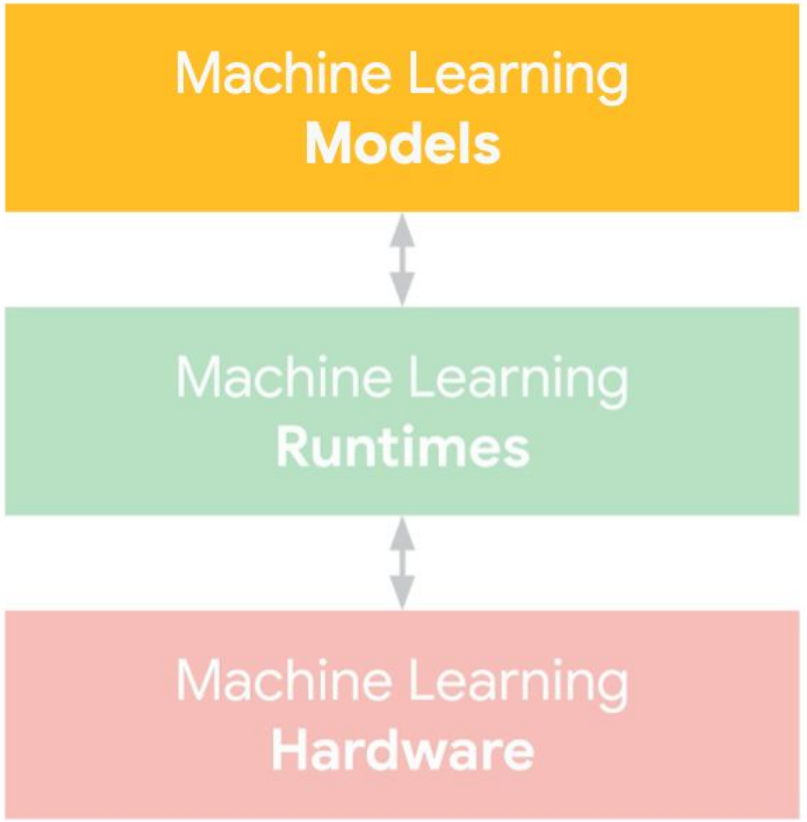
...

Pruning



Pruning





Model Compression Techniques

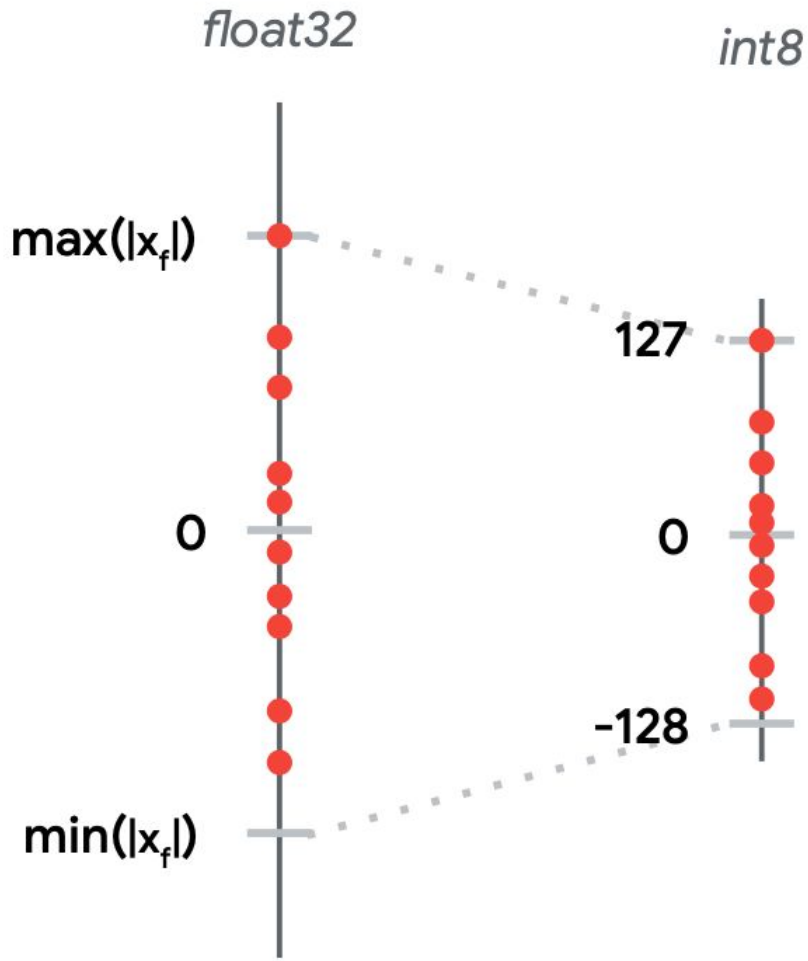
Pruning

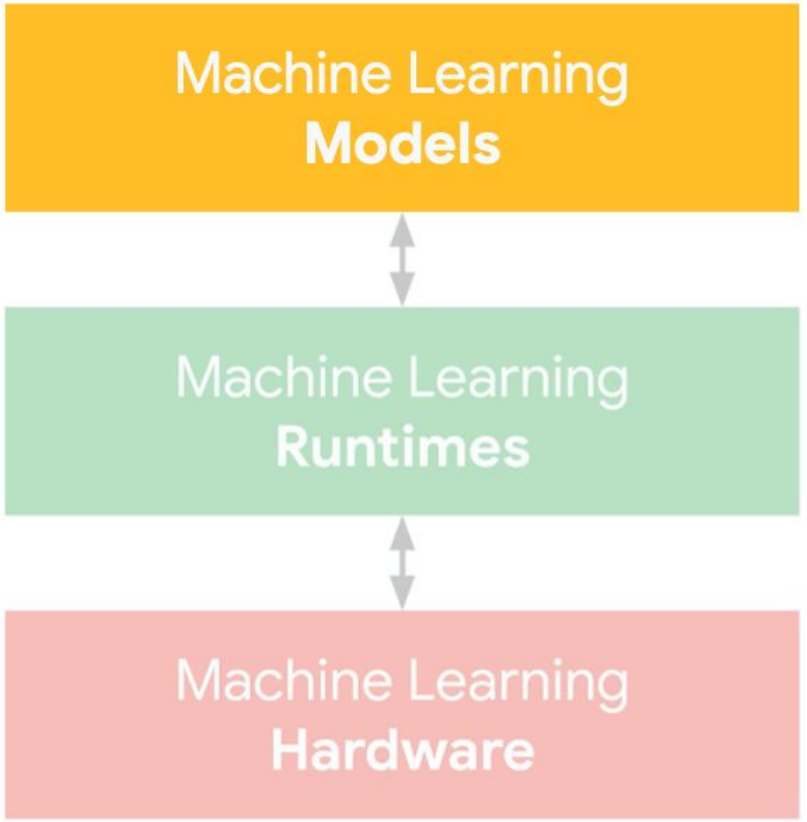
Quantization

Knowledge Distillation

...

Quantization





Model Compression Techniques

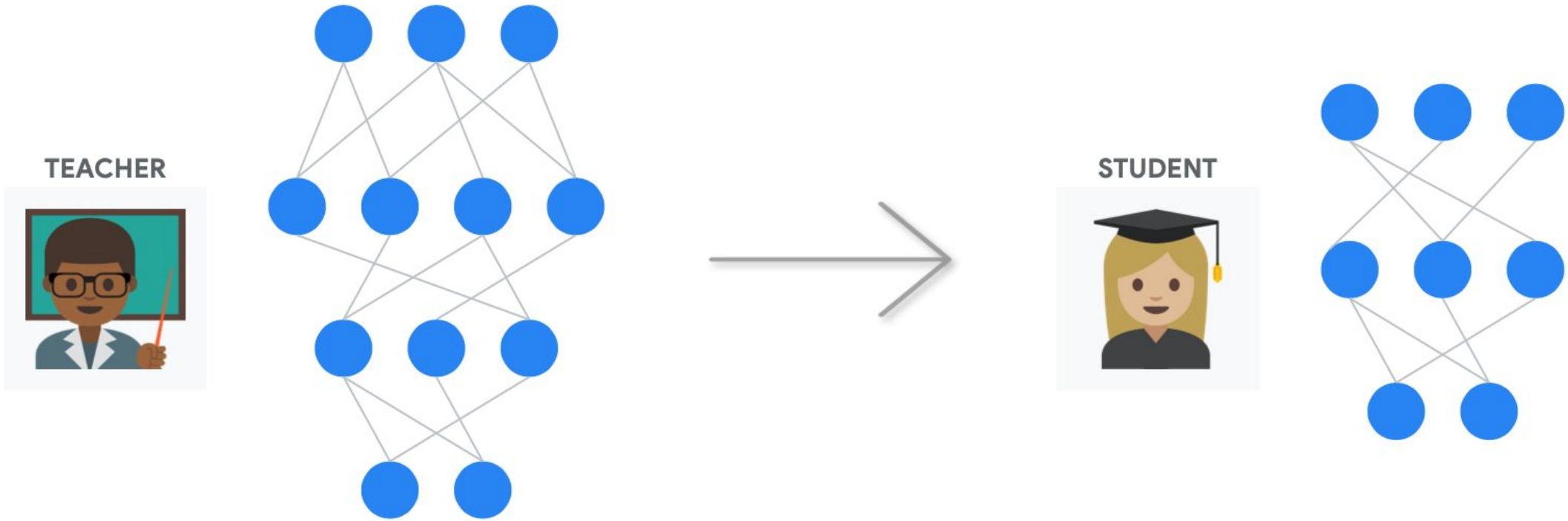
Pruning

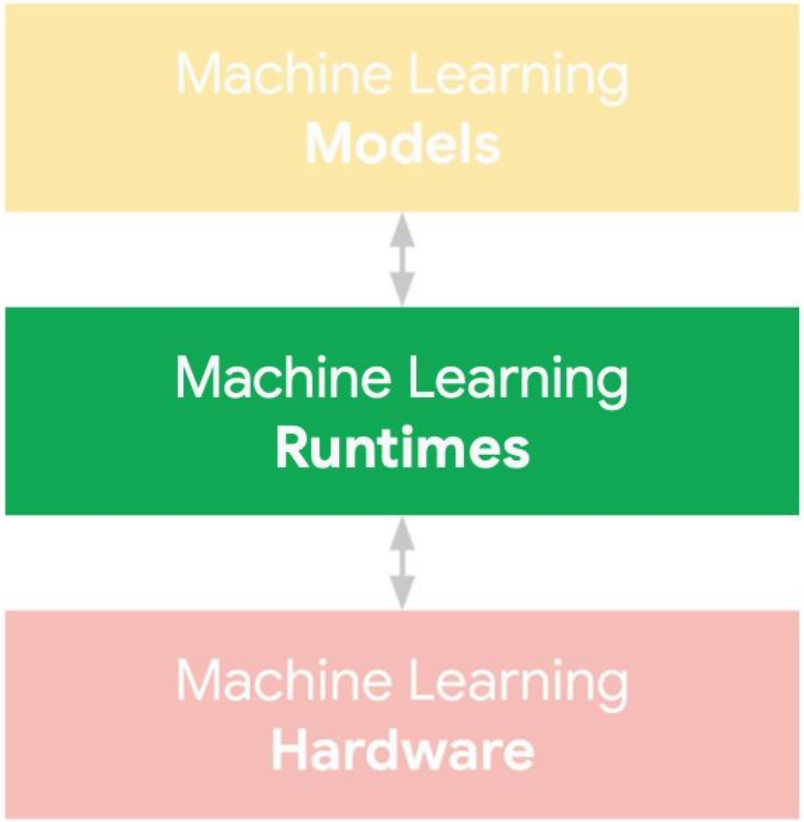
Quantization

Knowledge Distillation

...

Knowledge Distillation





TensorFlow

[TF Video]

TensorFlow

Less memory

Less compute power

Only focused on *inference*

TensorFlow Lite

Key Differences

	TensorFlow	TensorFlow Lite
Topology	Variable	Fixed
Weights	Variable	Fixed
Binary Size	Unimportant	High Priority
Distributed Compute	Needed	Not Needed
Developer Background	ML Researcher	Application Developer

Architecture

Trained TensorFlow Model

↓

TensorFlow Lite Converter

↓

TensorFlow Lite Model File (*.tflite*)

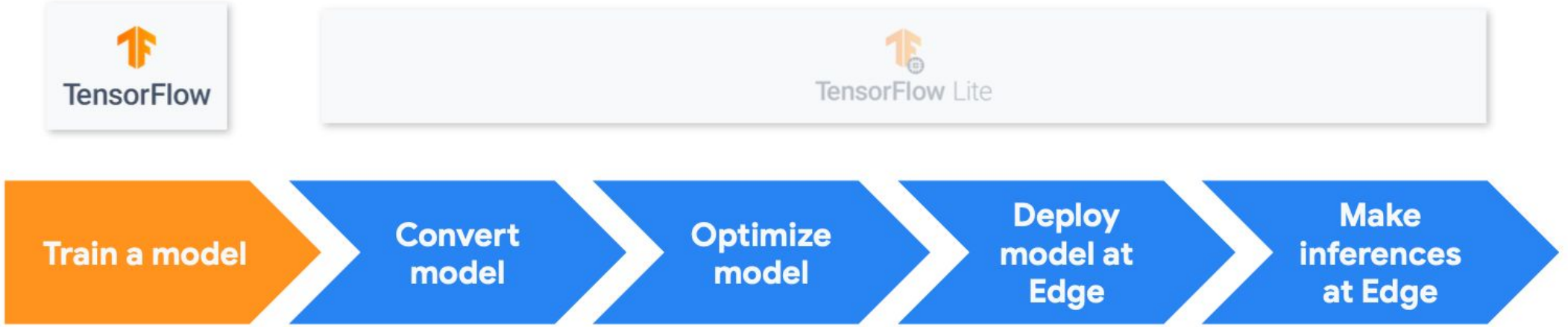
iOS

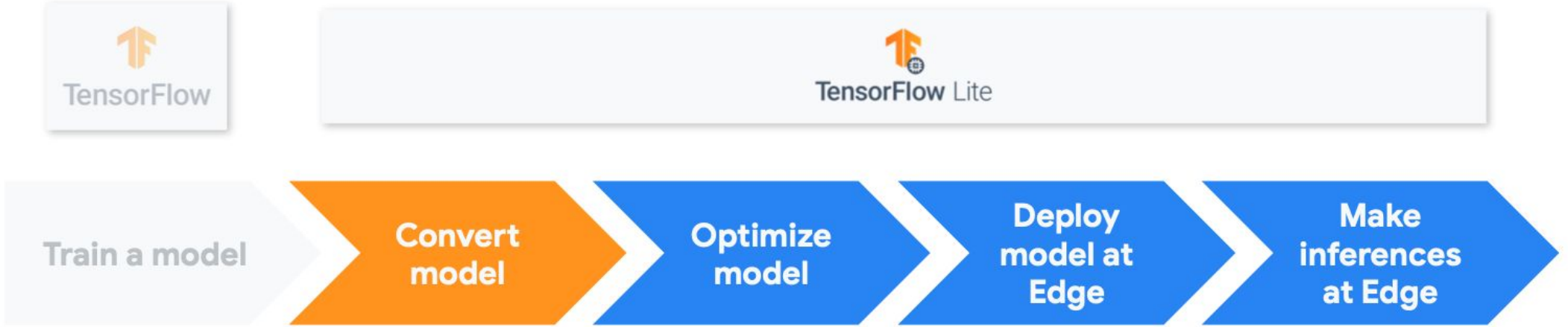
Linux

Even less memory

Even less compute power

Also, only focused on *inference*

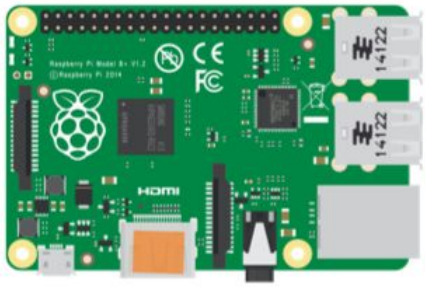

TensorFlow
TensorFlow Lite
Train a model
Convert model
Optimize model
Deploy model at Edge
Make inferences at Edge


TensorFlow
TensorFlow Lite
Train a model
Convert model
Optimize model
Deploy model at Edge
Make inferences at Edge

TensorFlow

TensorFlow Lite

Train a model → Convert model → Optimize model → Deploy model at Edge → Make inferences at Edge



Raspberry Pi

Linux

iOS

(TF Micro)

Microcontroller

TensorFlow

TensorFlow Lite

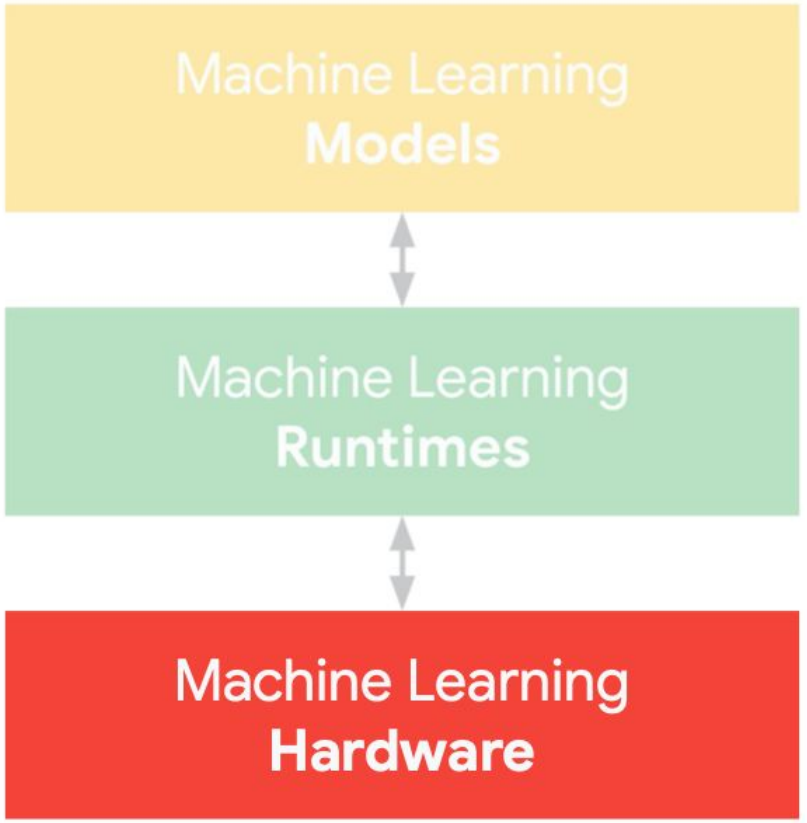
Train a model

Convert model

Optimize model

Deploy model at Edge

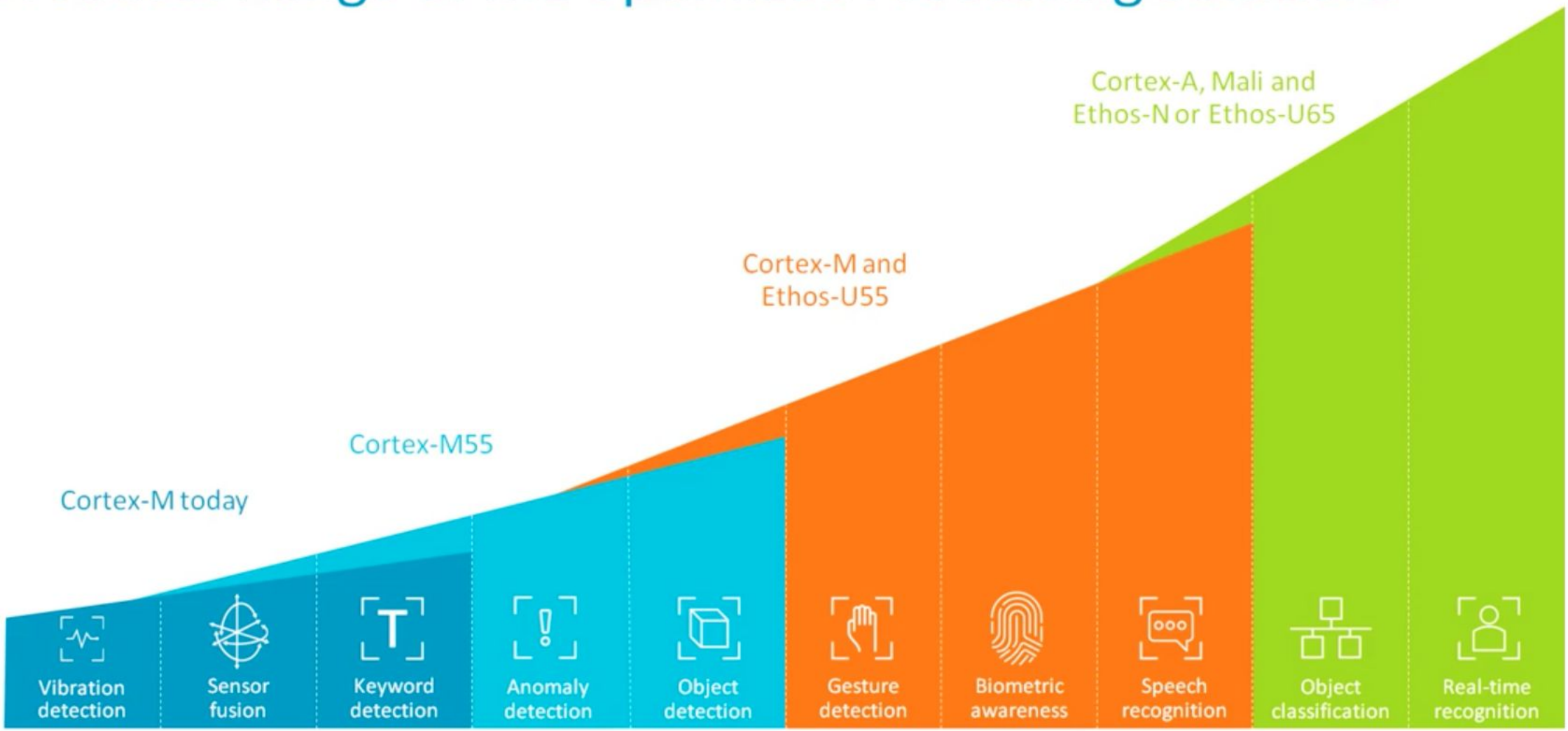
Make inferences at Edge



"Energy-efficient On-device Processing for Next-generation Endpoint ML"

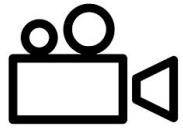
By Tomas Edso, Senior Principal Engineer (ML), ARM
At tinyML Summit 2020 presentation

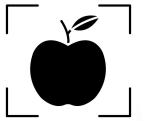
Broadest Range of ML-optimized Processing Solutions



arm

Summary





TinyML EdgeML

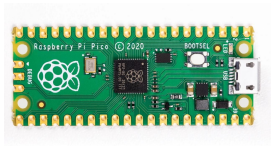
Video Classification 2 MB+

Object Detection Complex Voice Processing 1 MB+

Image Classification 250 KB+

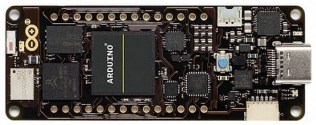
KeyWord Spotting Audio Classification 50 KB

Anomaly Detection Sensor Classification 20 KB



Rpi-Pico (Cortex-M0+)

Arduino Nano (Cortex-M4)



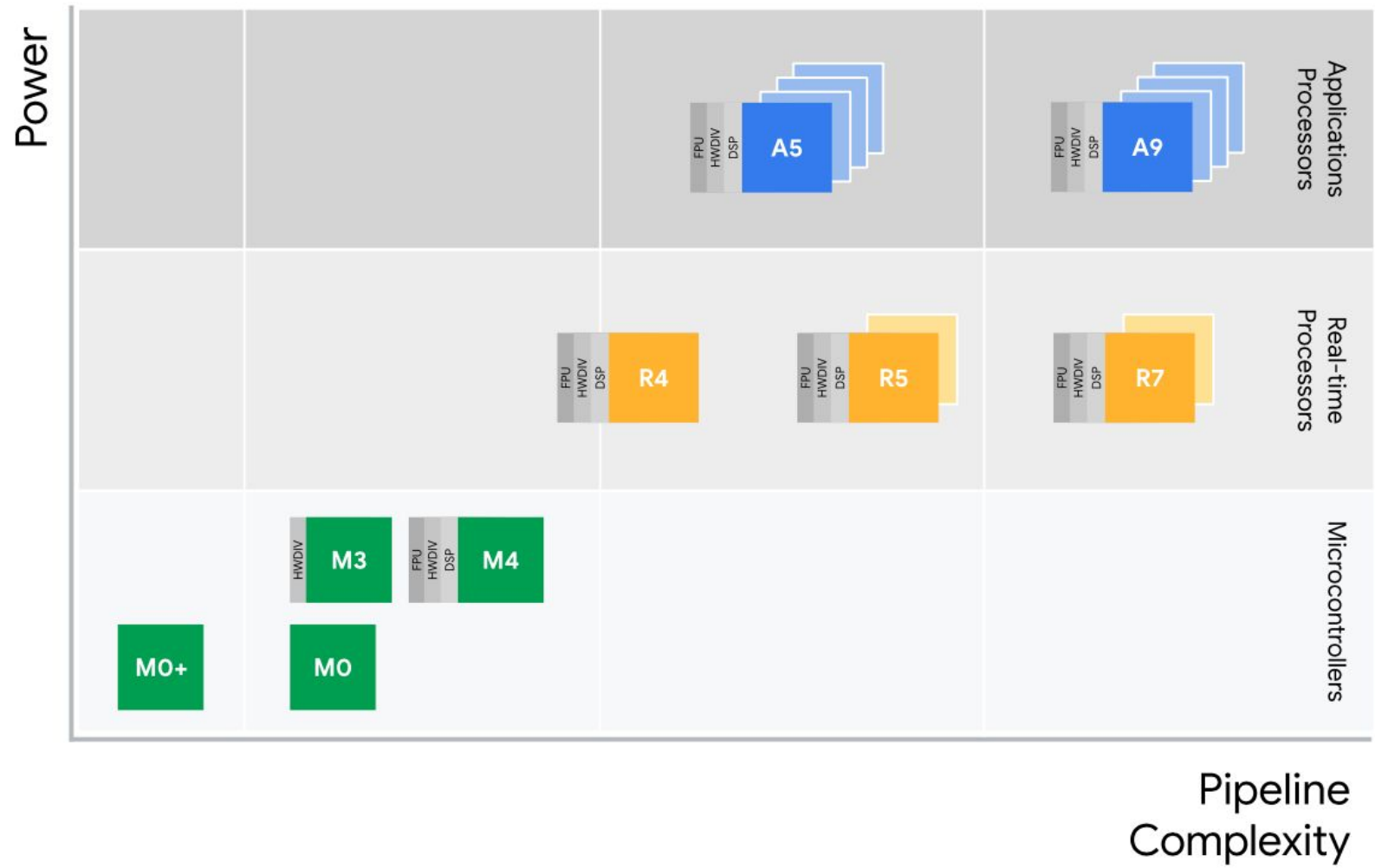
Arduino Pro (Cortex-M7)

Raspberry Pi (Cortex-A)

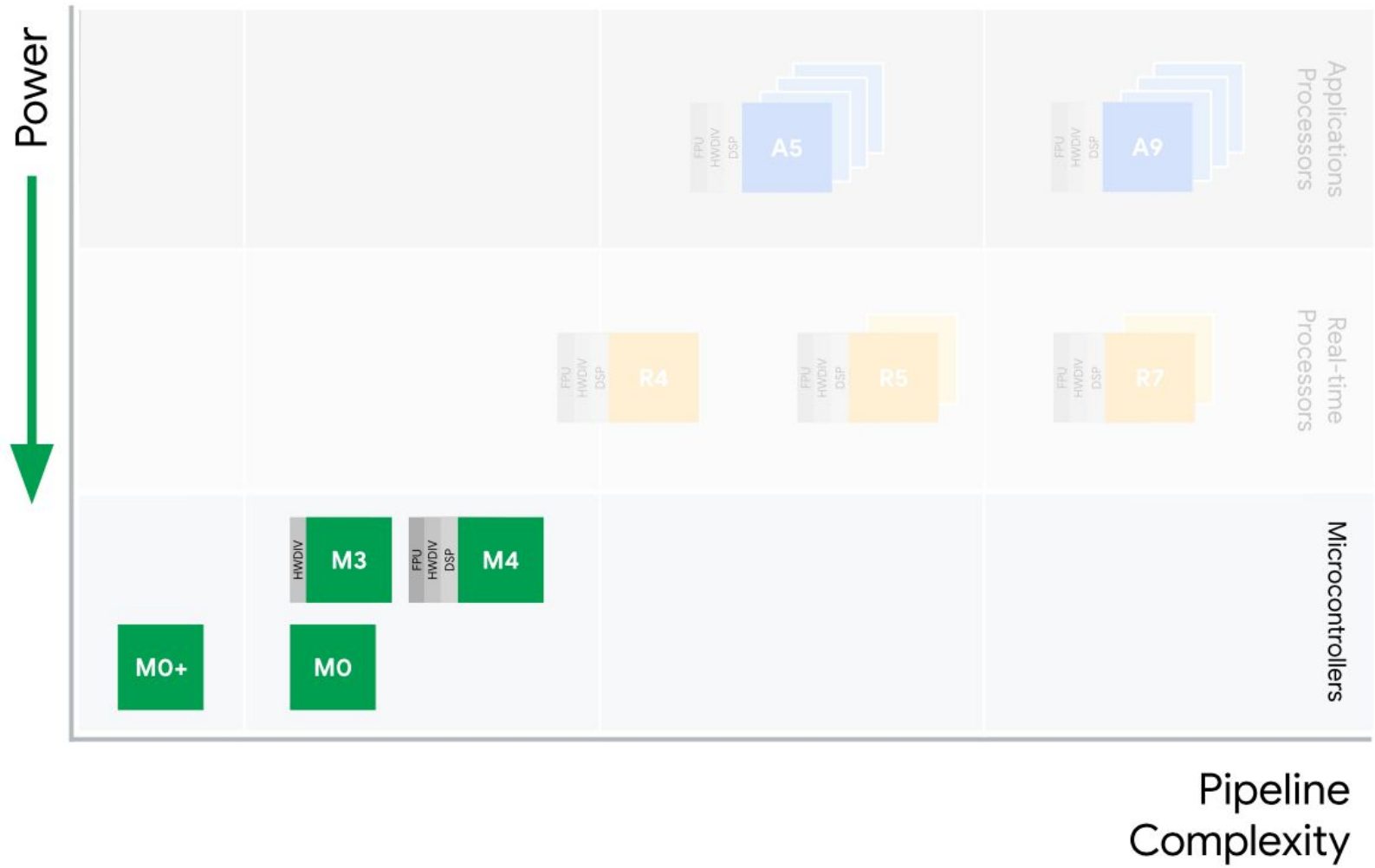
Jetson Nano (Cortex-A + GPU)

Sourse: Edge Impulse

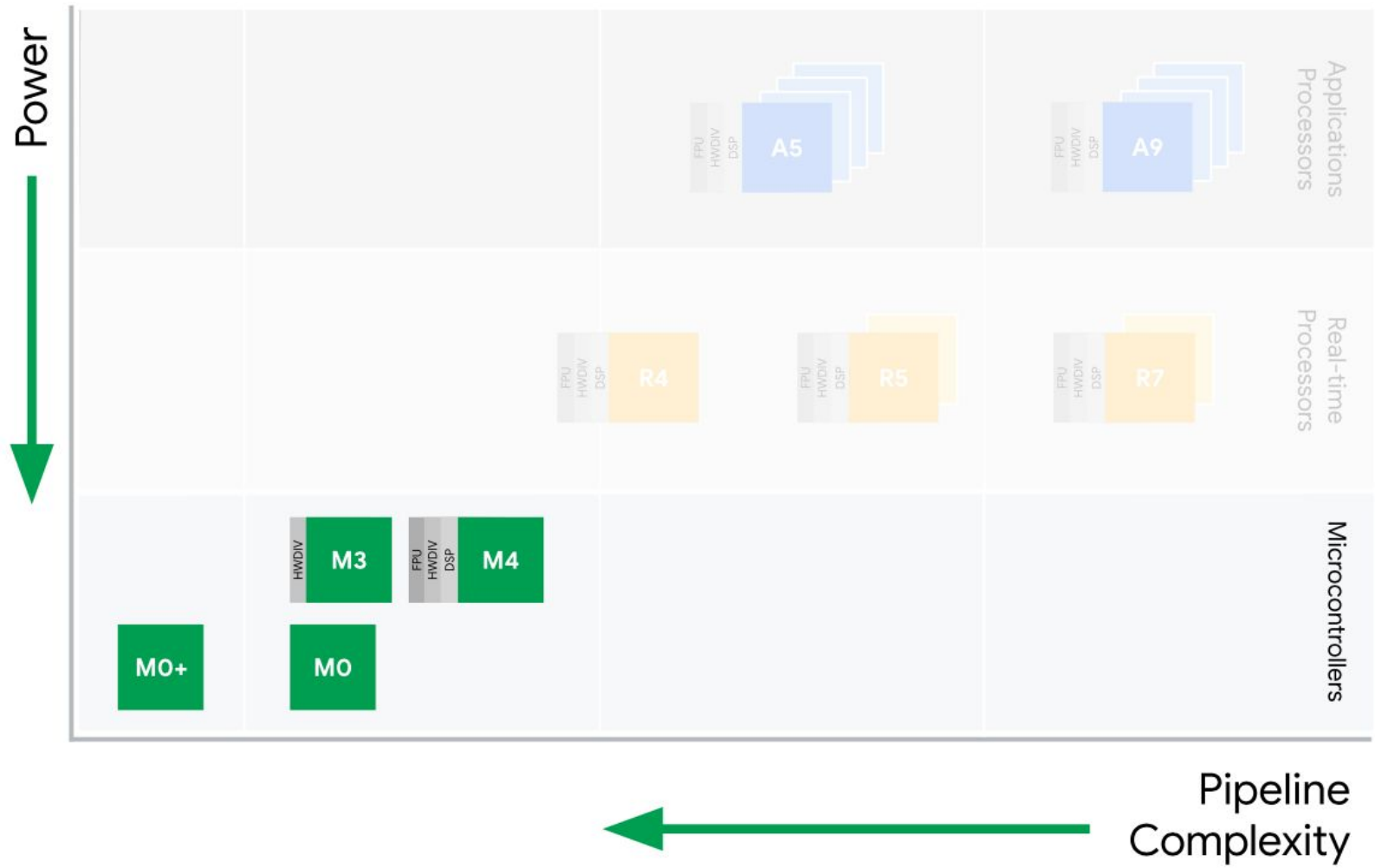
ARM Cortex Processor Profiles



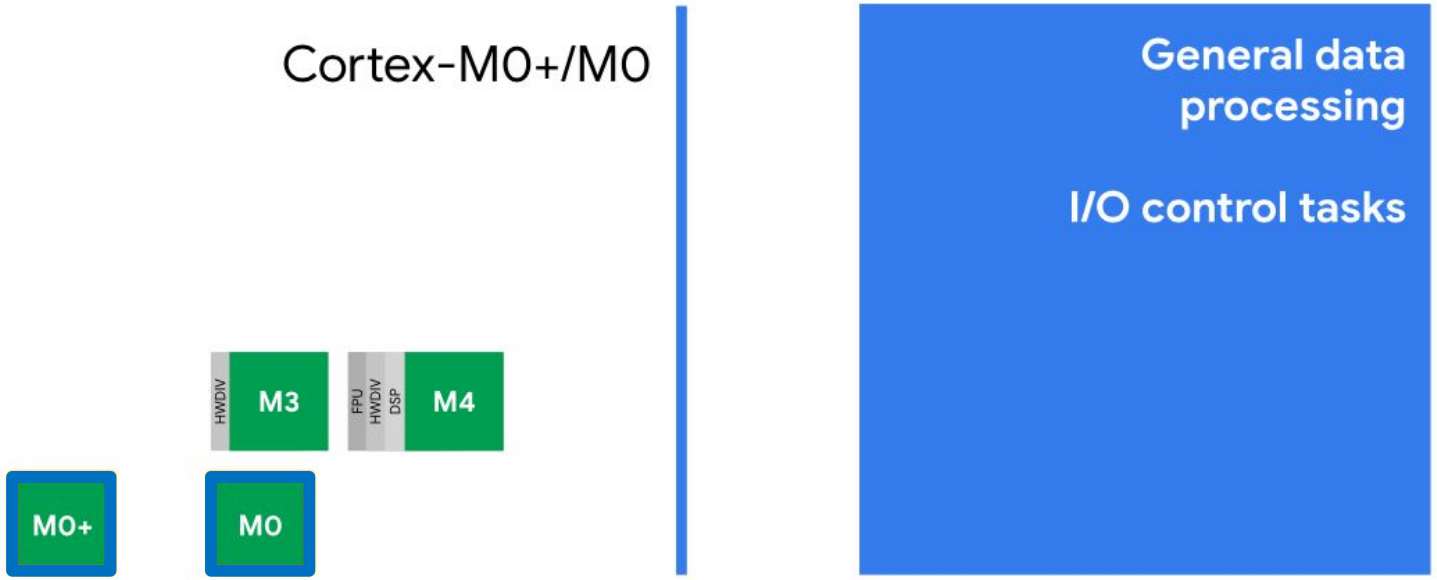
ARM Cortex Profiles



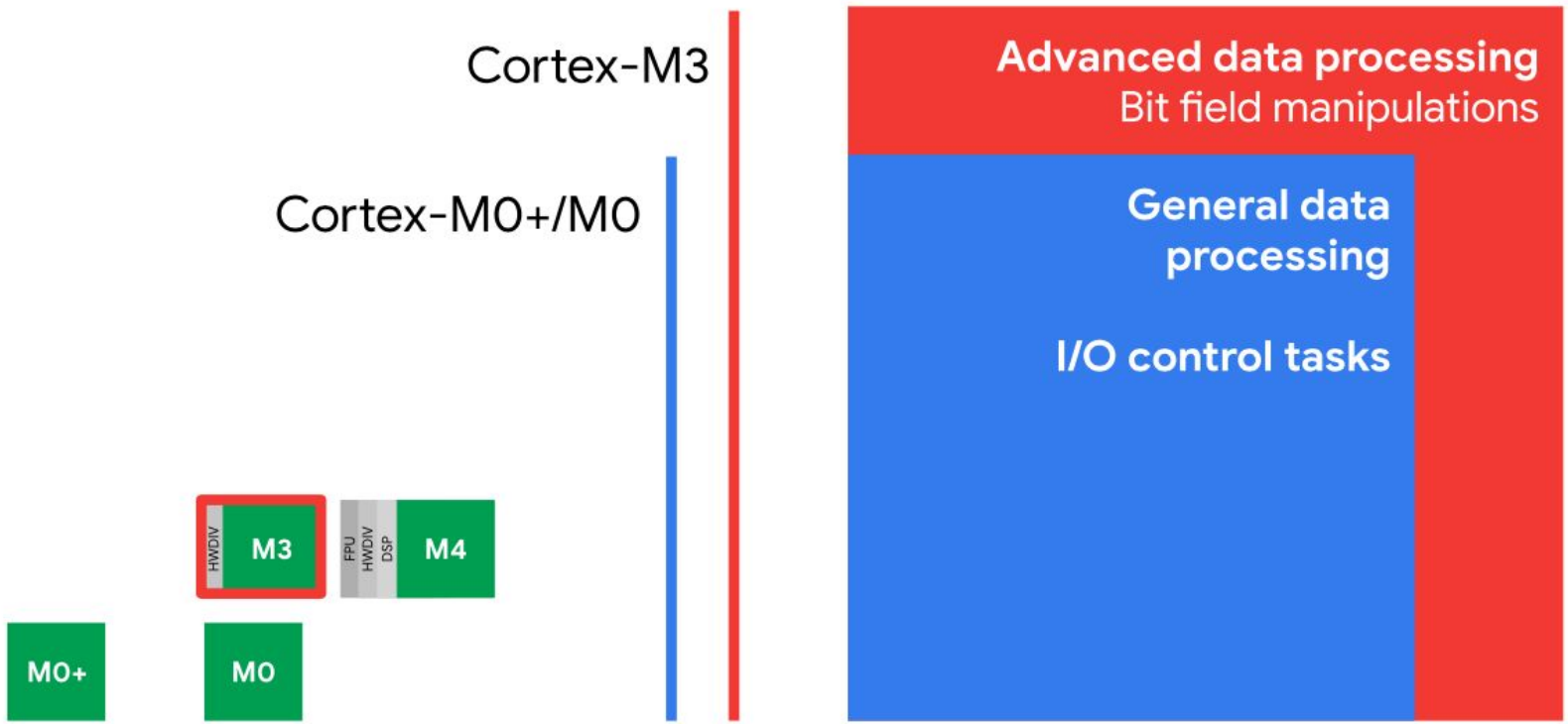
ARM Cortex Profiles



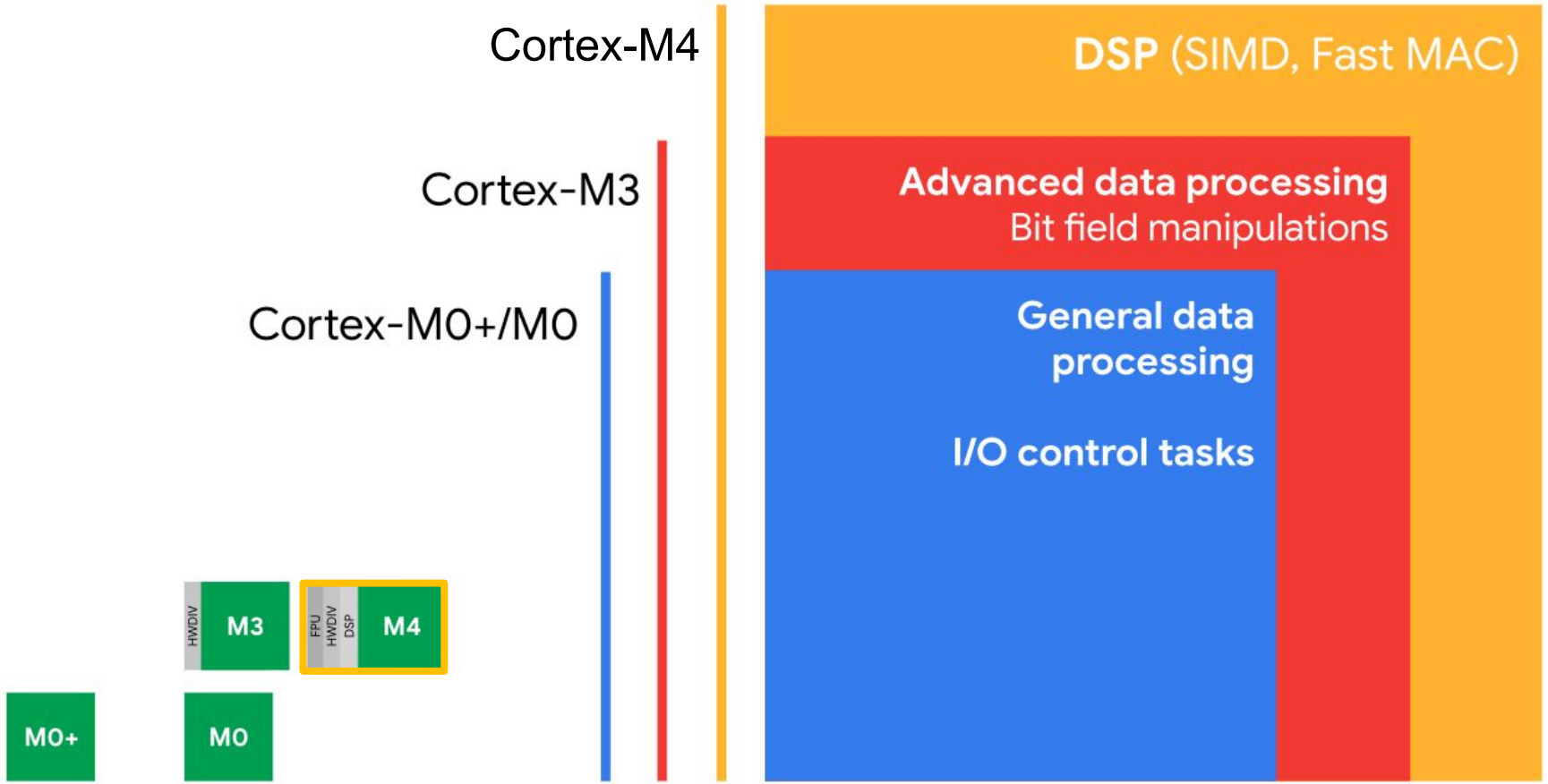
ARM **Cortex-M** ISA



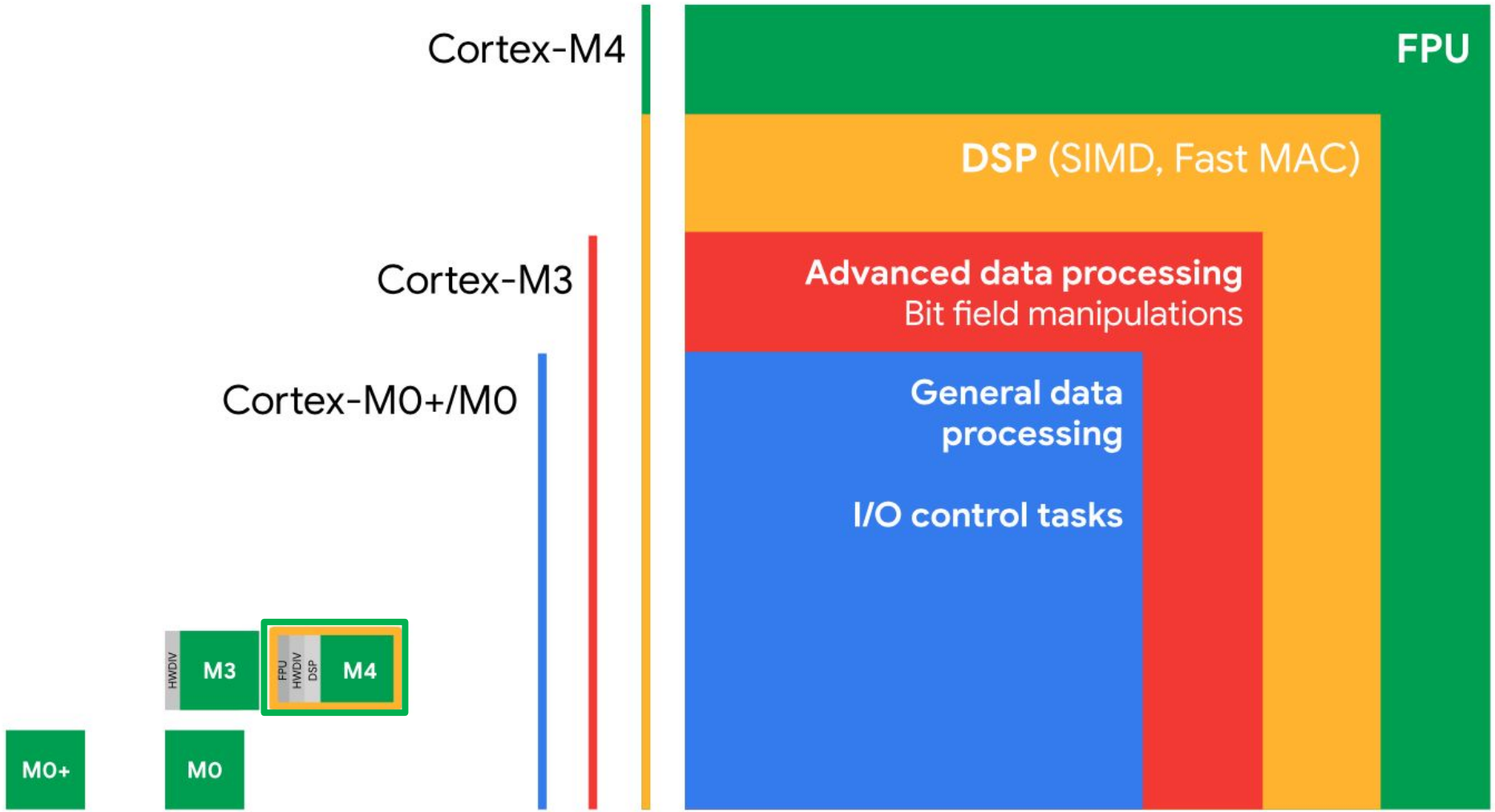
ARM **Cortex-M** ISA



ARM **Cortex-M** ISA



ARM **Cortex-M** ISA



Reading Material

Main references

- Harvard School of Engineering and Applied Sciences - CS249r: Tiny Machine Learning
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
- Introduction to Embedded Machine Learning (Coursera)
- Text Book: "TinyML" by Pete Warden, Daniel Situnayake

I want to thank Shawn Hymel and Edge Impulse, Pete Warden and Laurence Moroney from Google and specially Harvard professor Vijay Janapa Reddi, Ph.D. student Brian Plancher and their staff for preparing the excellent material on TinyML that is the basis of this course at UNIFEI.

The IESTI01 course is part of the TinyML4D, an initiative to make TinyML education available to everyone globally.

Thanks

And stay safe!

UNIFEI