Track 7

Inteligencia Artificial Aplicada

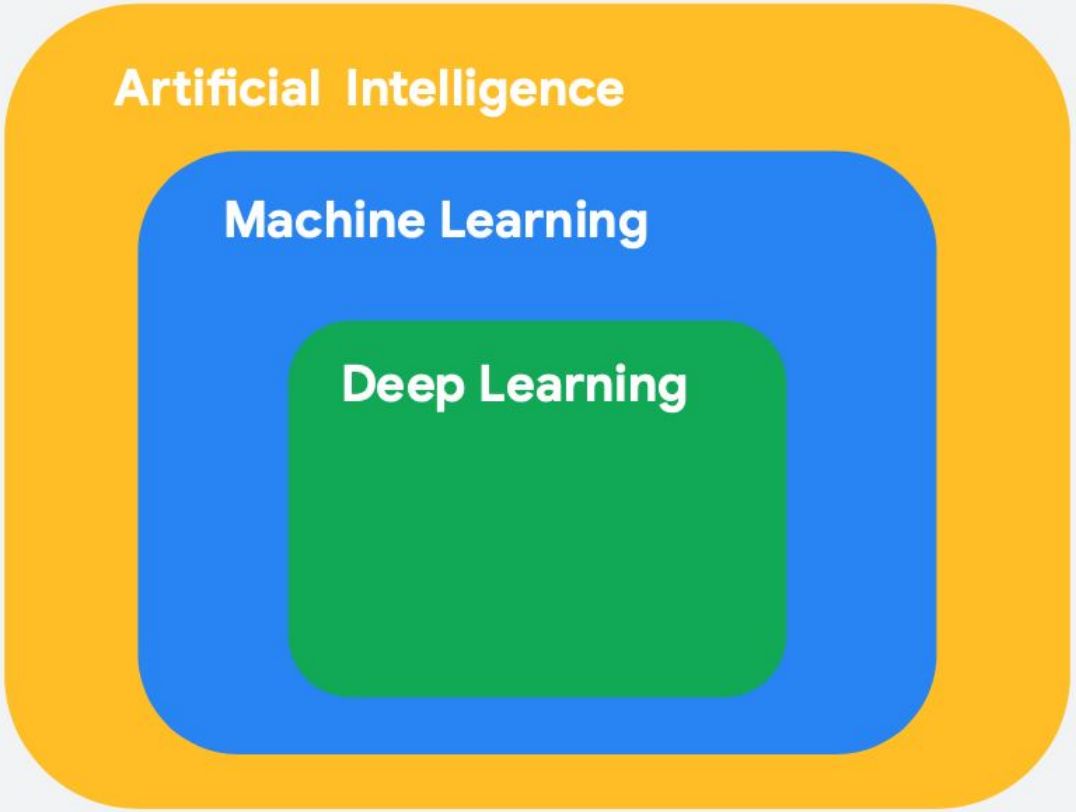
10. Embedded ML (TinyML) Intro & Applications

Prof. Marcelo José Rovai
rovai@unifei.edu.br

UNIFEI - Universidade Federal de Itajubá, Brazil

Embedded ML (TinyML)

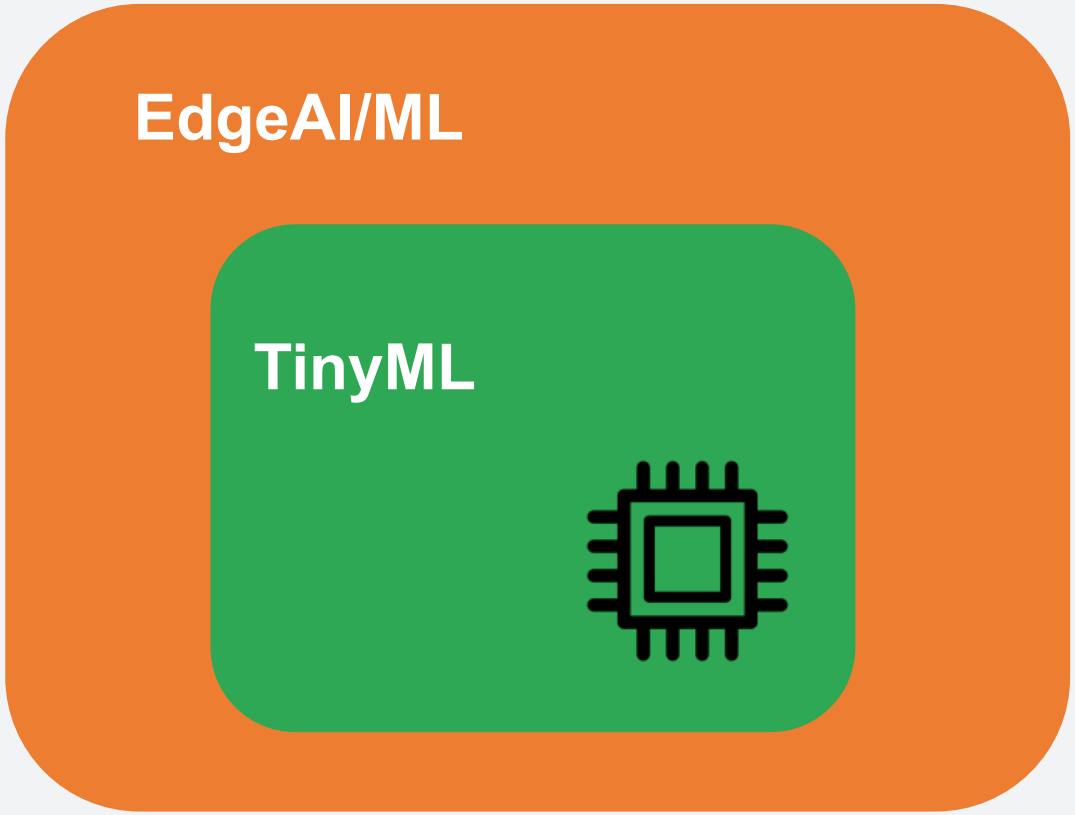
Introduction



AI: Any technique that enables computers to mimic human behavior

ML: Ability to learn without explicitly programming

DL: Extract patterns from data using neural networks



EdgeAI (or EdgeML) is the processing of Artificial Intelligence algorithms on **the network edge**, that is, on users' devices. The concept derives from Edge Computing, which starts from the same premise: data is stored, processed, and managed directly at the Internet of Things (IoT) endpoints.

TinyML is a subset of EdgeML, where sensors generate data at ultra-low power consumption (battery operated), so that they can operate continuously **("always on devices")**

What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

5

What is Tiny Machine Learning (TinyML)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

6

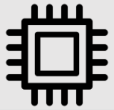
What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

On-device sensor analytics



7

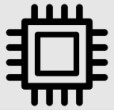
What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

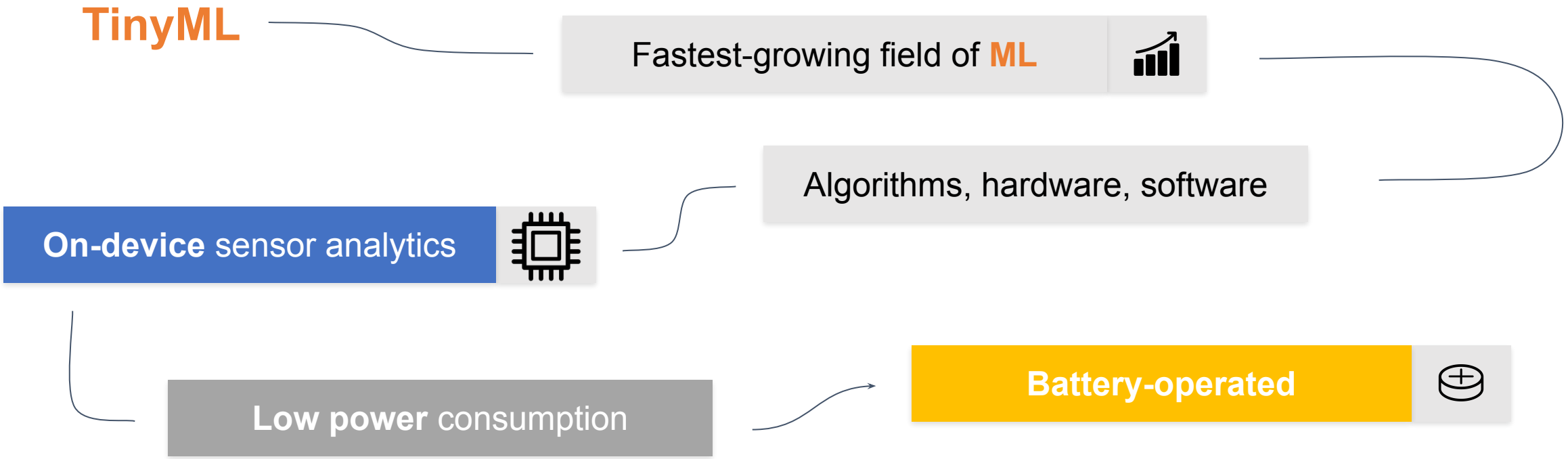
On-device sensor analytics



Low power consumption

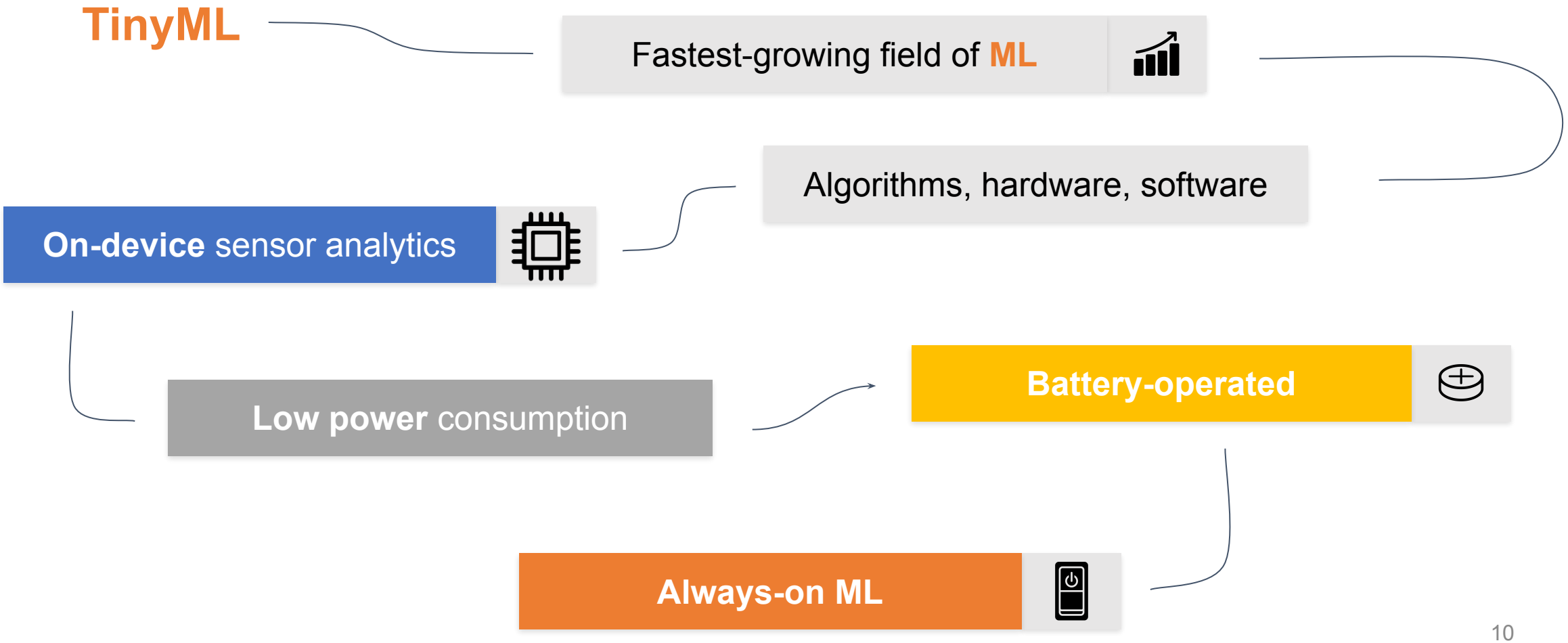
8

What is Tiny Machine Learning (TinyML)?



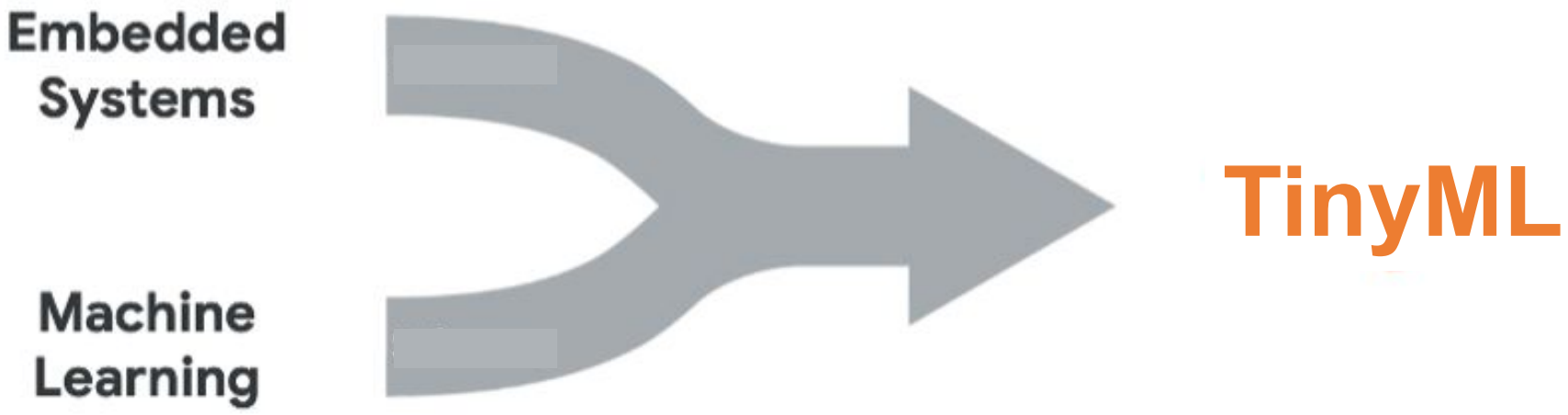
9

What is Tiny Machine Learning (TinyML)?

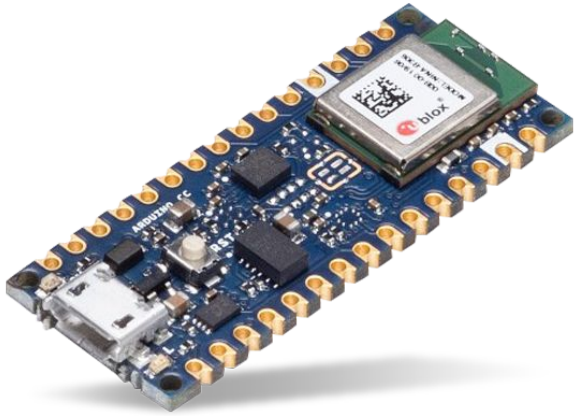


10

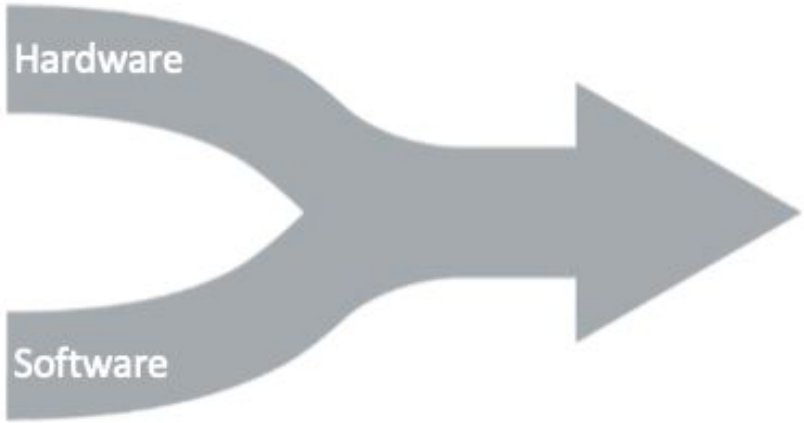
What Makes TinyML ?



What Makes TinyML ?



TensorFlow Lite



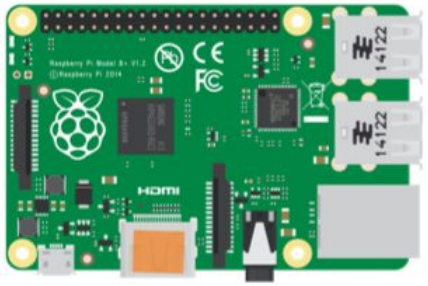
TinyML

Software

TensorFlow

TensorFlow Lite

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



Raspberry Pi

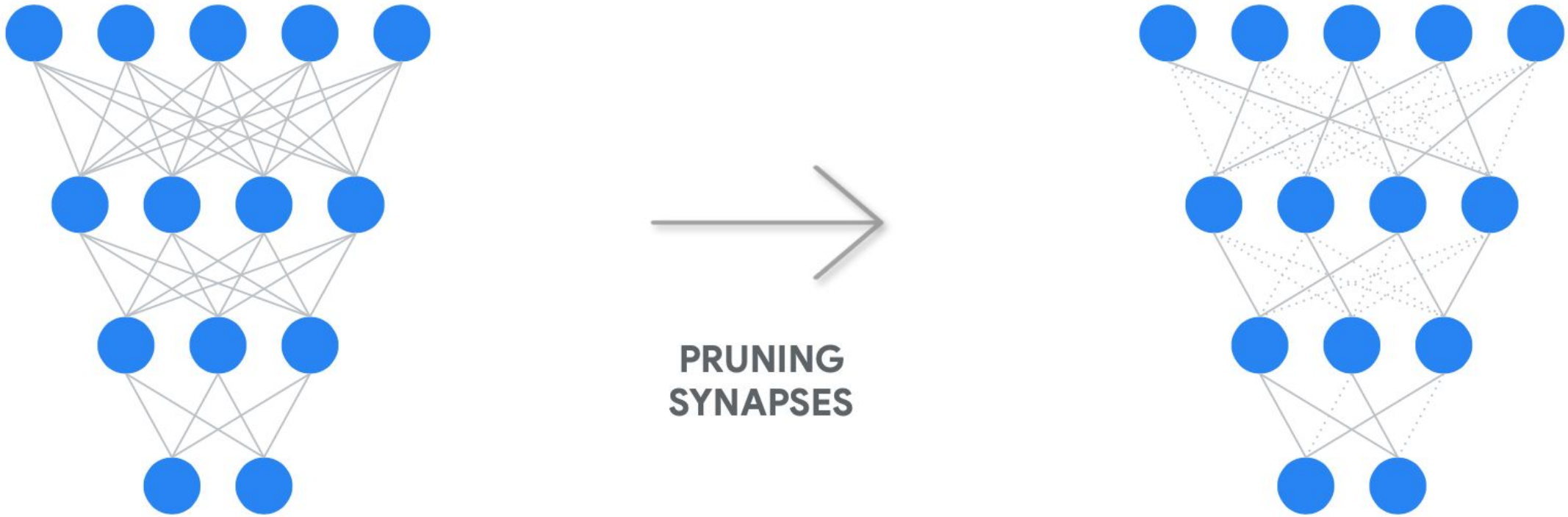
Linux

iOS

TensorFlow Lite Micro

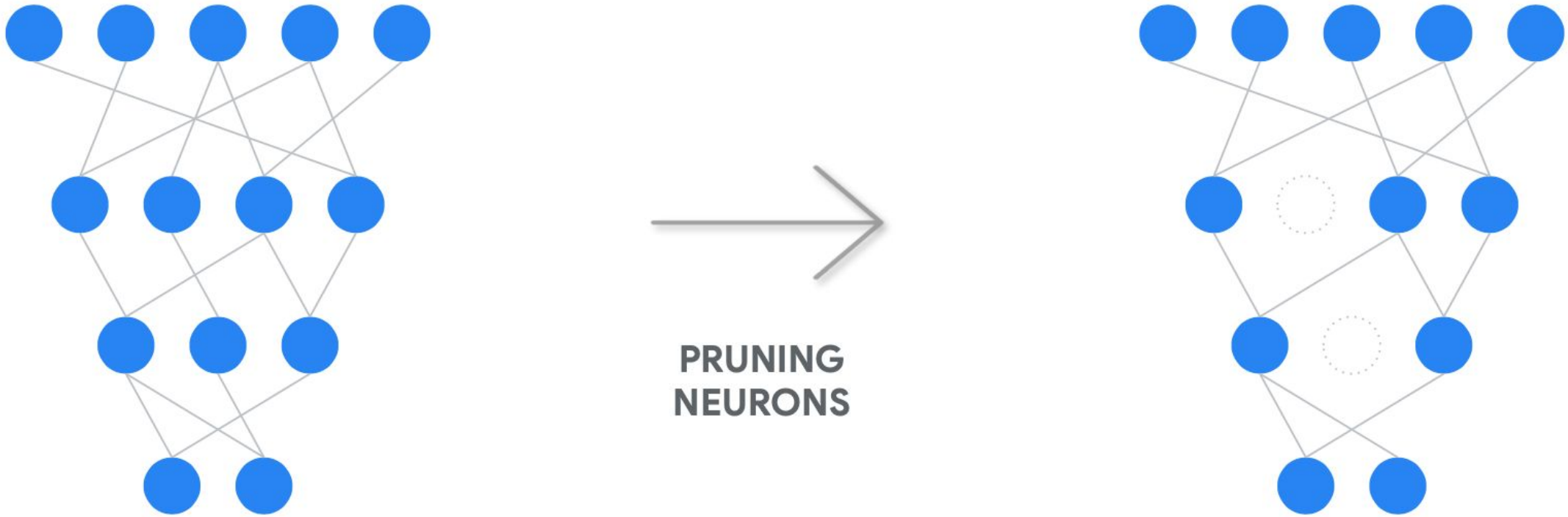
Microcontroller

Pruning



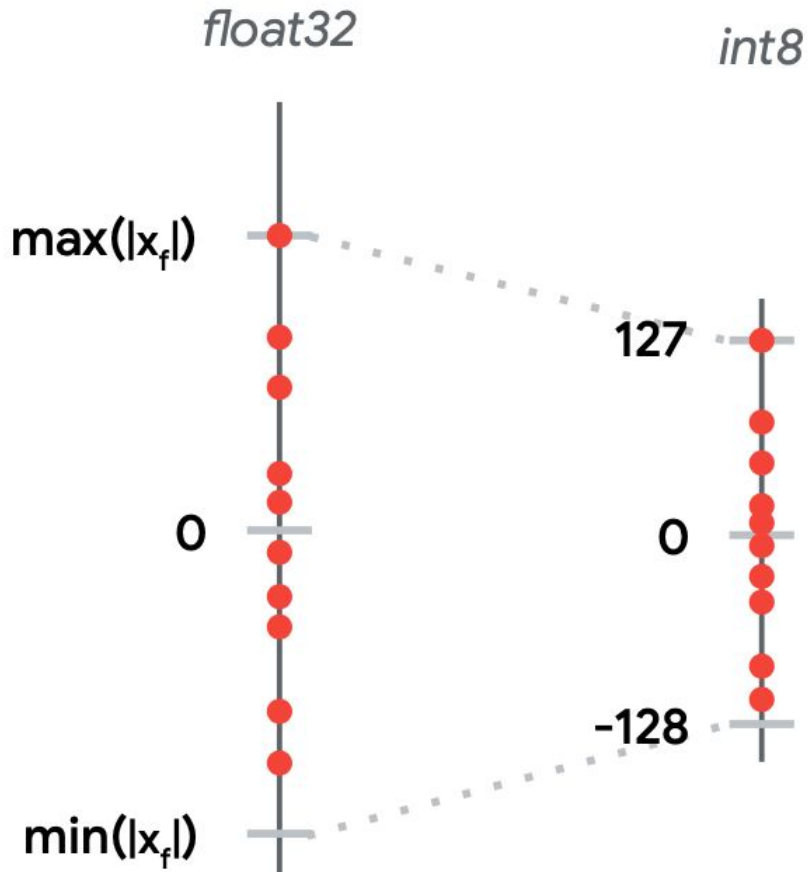
TensorFlow Lite

Pruning



TensorFlow Lite

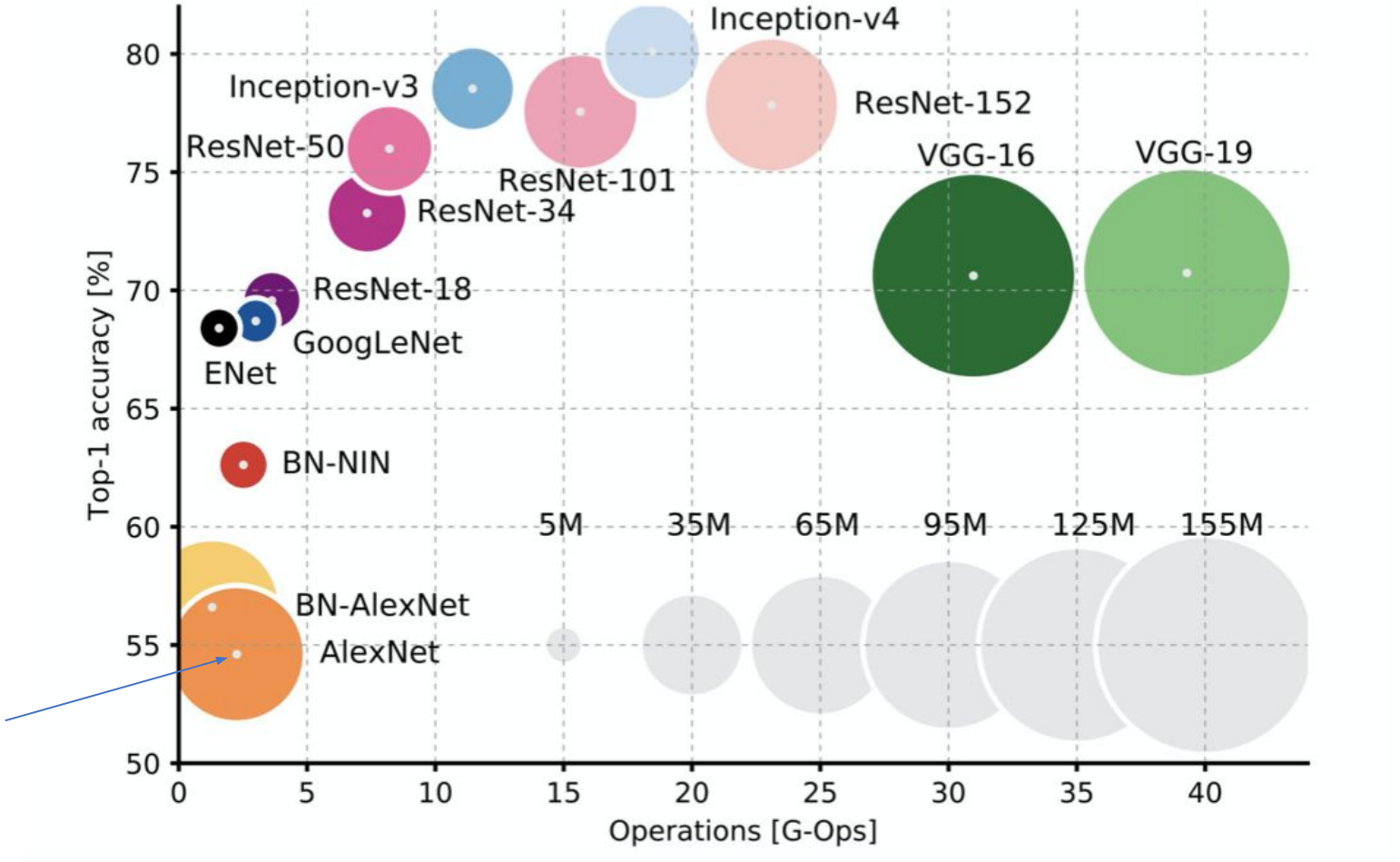
Quantization



TensorFlow Lite


Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
TensorFlow
TensorFlow Lite
TensorFlow Lite Micro

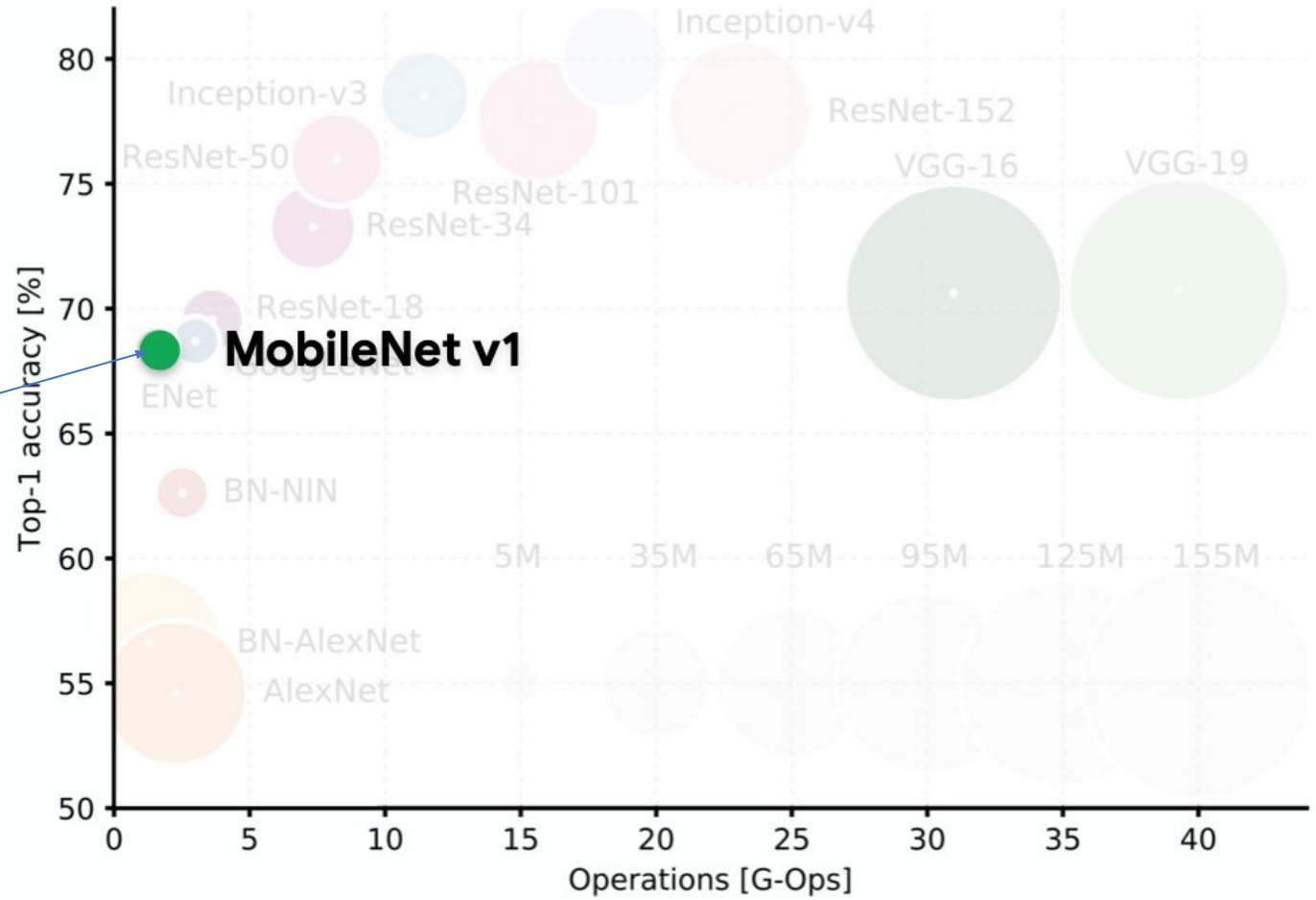
Model **Evolution**



(2012)

Model **Evolution**

(2017)



MobileNet v1

Model	Size	Top-1 Accuracy
MobileNet v1	***16 MB*** *	0.713

* Not Quantized

Fine for mobile phones or Rpi with GB of RAM, but not for microcontroller

Our Arduino Nano only has 256KB of memory RAM

Further **Optimizations**

Multiply-Accumulates

α	Image Size	MACs (millions)	Params (millions)	Top-1 Accuracy
1	224	569	**4.24**	**70.7**
1	128	186	4.14	64.1
0.75	224	317	**2.59**	**68.4**
0.75	128	104	2.59	61.8
0.5	224	150	**1.34**	**64.0**
0.5	128	49	1.34	56.2
0.25	224	41	**0.47**	**50.6**
0.25	128	14	0.47	41.2

MobileNet v1

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

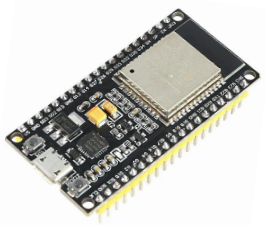
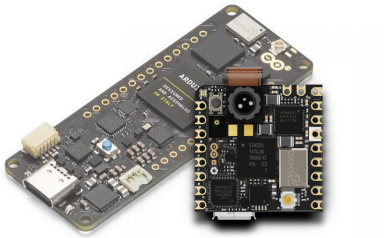
Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko
Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam

Google Inc.

{howarda,menglong,bochen,dkalenichenko,weijunw,weyand,anm,hadam}@google.com

https://arxiv.org/pdf/1704.04861.pdf

Hardware

	 Raspberry Pico (W)	 Arduino Nano Sense	 ESP 32	 Seeed XIAO BLE Sense	 Arduino Pro
32Bits CPU	Dual-core Arm Cortex-M0+	Arm Cortex-M4F	Xtensa LX6 Dual Core	Arm Cortex-M4F	Dual Core Arm Cortex M7/M4
CLOCK	133MHz	64MHz	240MHz	64MHz	480/240MHz
RAM	264KB	256KB	520KB	256KB	1MB
ROM	2MB	1MB	2MB	2MB	2MB
Radio	(Yes for W)	BLE	BLE/WiFi	BLE	BLE/WiFi
Sensors	No	Yes	No	Yes	No (Portenta) Yes (Nicla)
Price	$	$$$	$$	$$	$$$$

Application Complexity vs. HW

Power

EdgeML

TinyML

Video Classification 2 MB+

Object Detection Complex Voice Processing 1 MB+

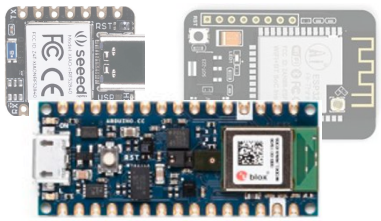
Image Classification 250 KB+

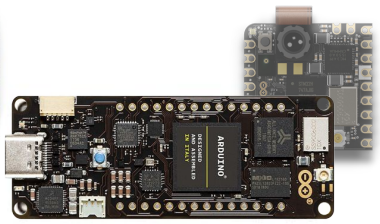
KeyWord Spotting Audio Classification 50 KB

Anomaly Detection Sensor Classification 20 KB

XIAO

ESP32





Rpi-Pico (Cortex-M0+)

Arduino Nano (Cortex-M4)

Arduino Pro (Cortex-M7)

RaspberryPi SmartPhone (Cortex-A)

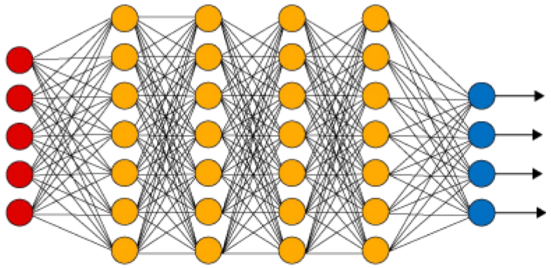
Jetson Nano (Cortex-A + GPU)

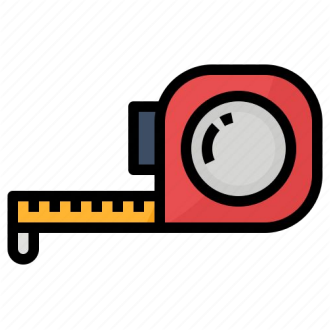
Application Complexity

CPU Power / Memory

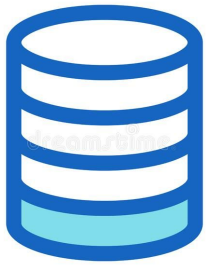
How to Train a ML Model?

Machine Learning Workflow ("What")

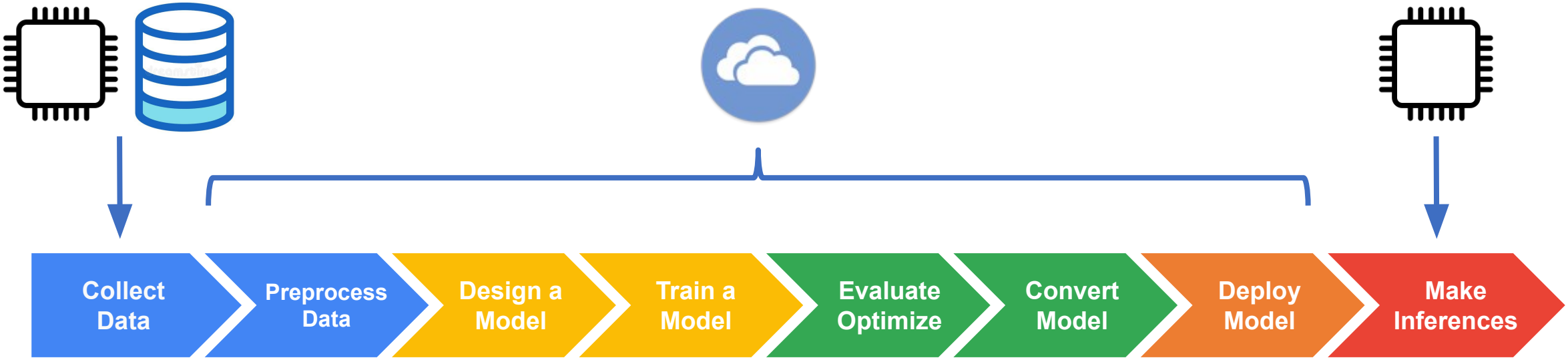




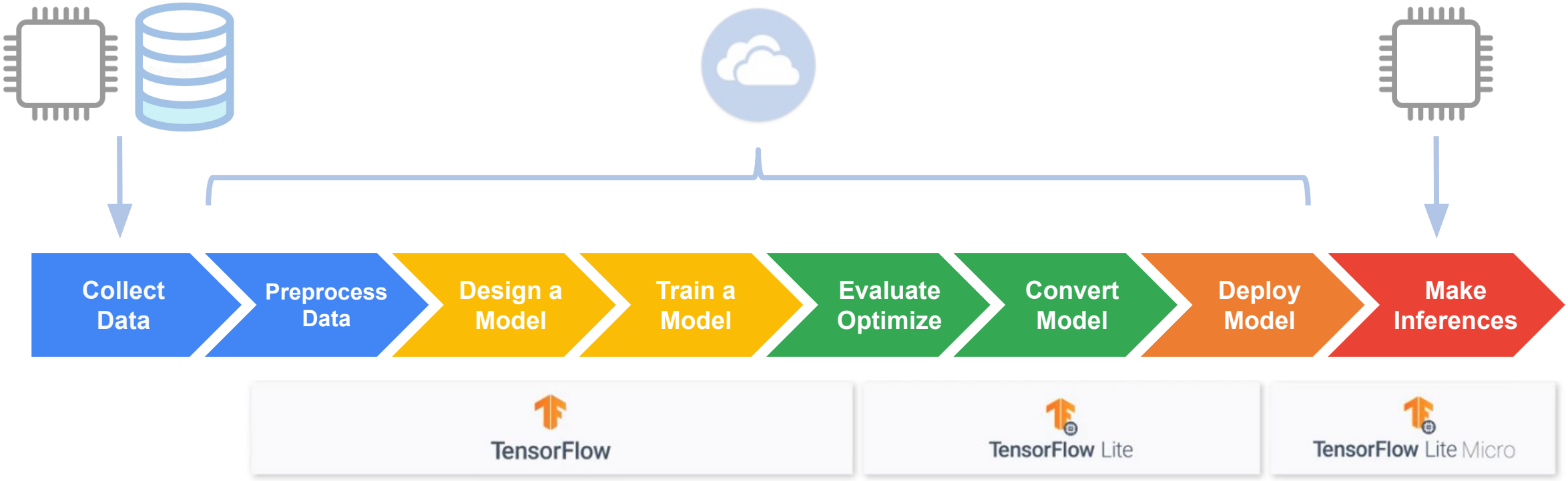
Collect Data → Preprocess Data → Design a Model → Train a Model → Evaluate Optimize → Convert Model → Deploy Model → Make Inferences



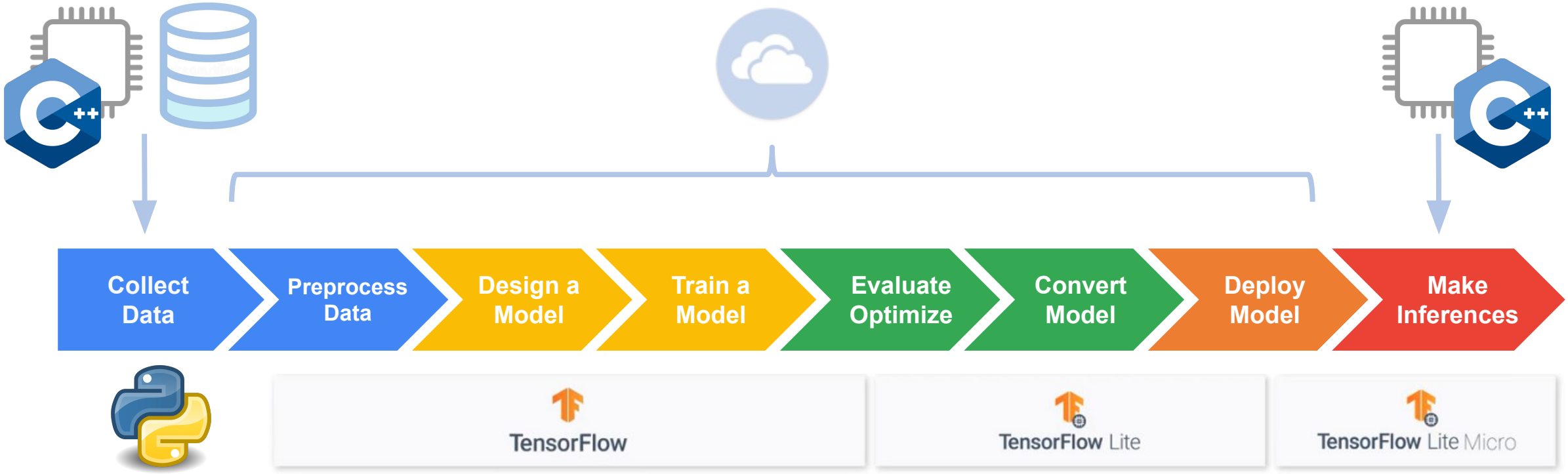
Machine Learning Workflow ("Where")



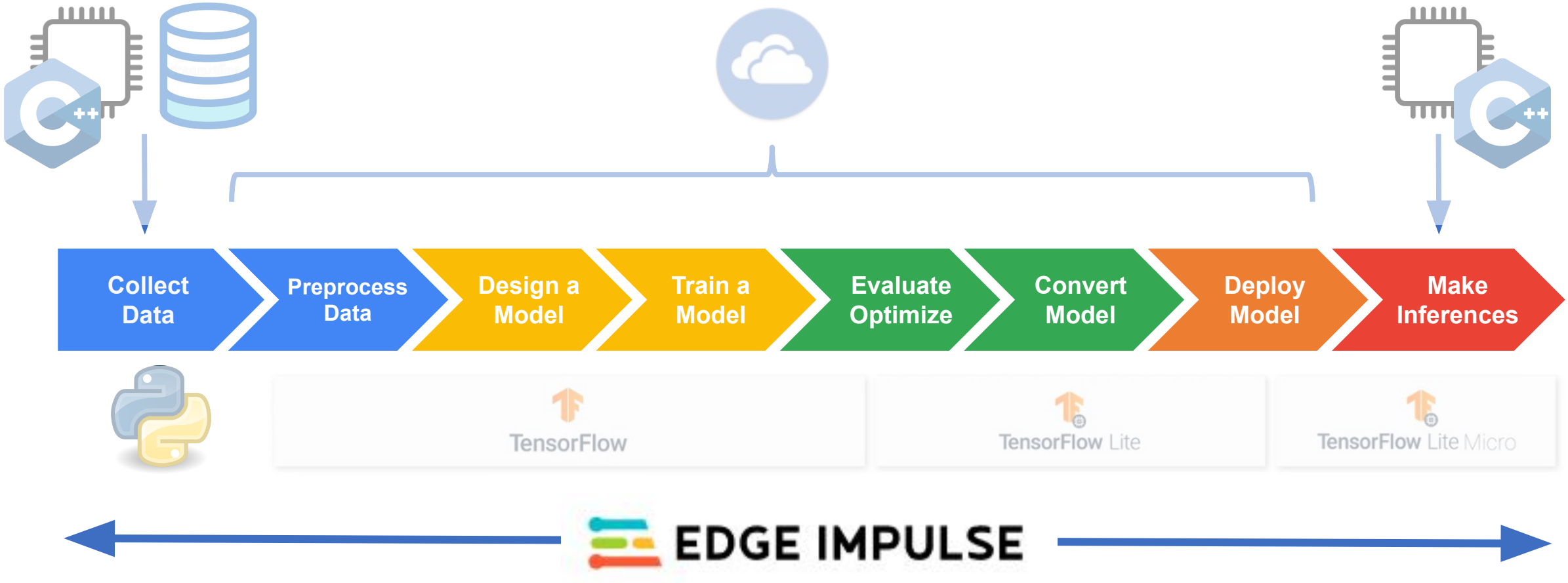
Machine Learning Workflow ("How")



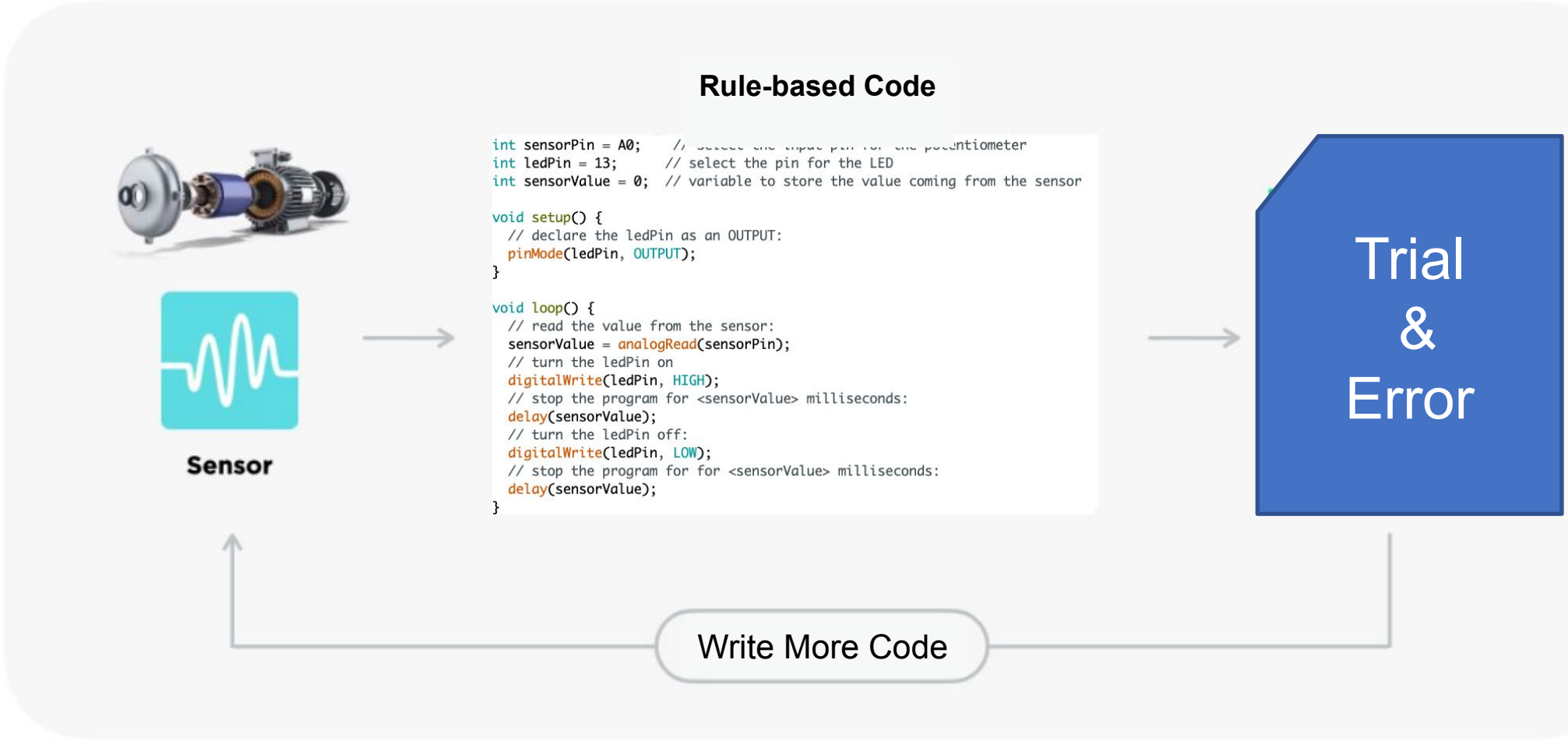
Machine Learning Workflow ("How")



Machine Learning Workflow ("How")

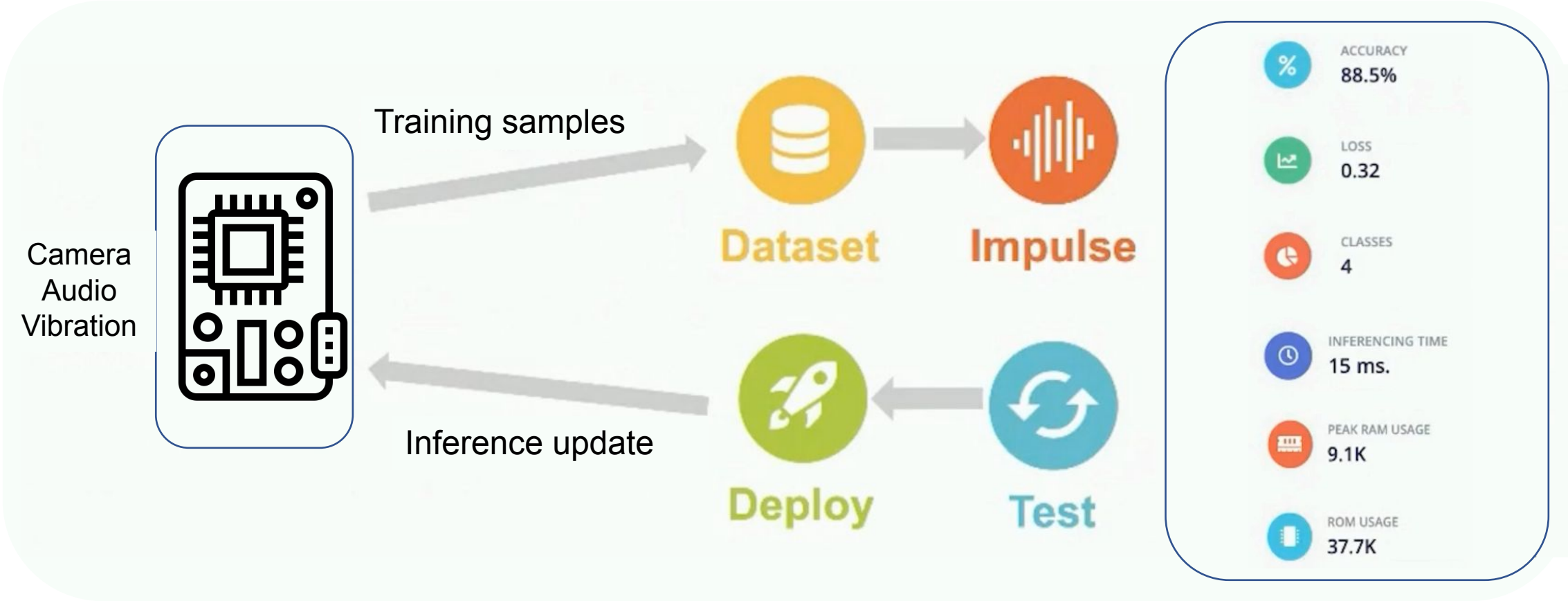


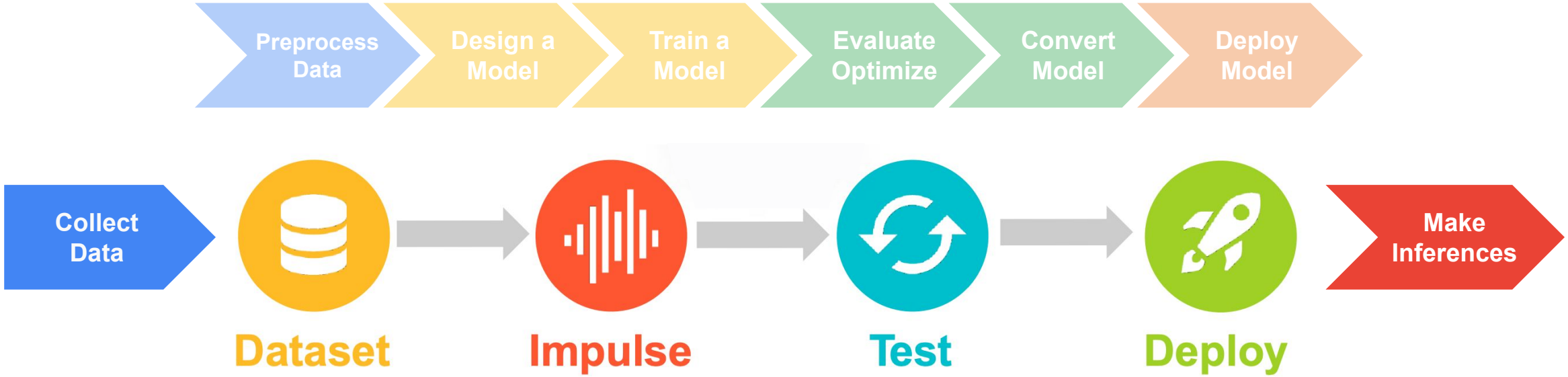
From rule-based engineering to…



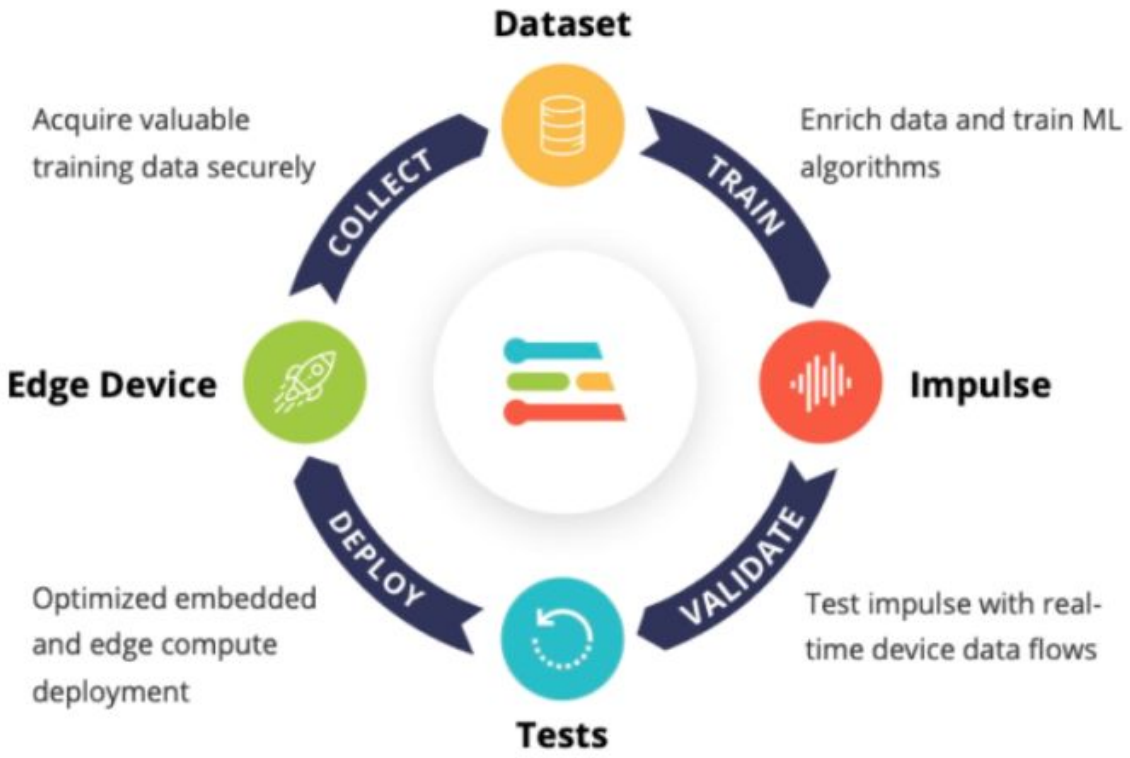
Data-driven engineering

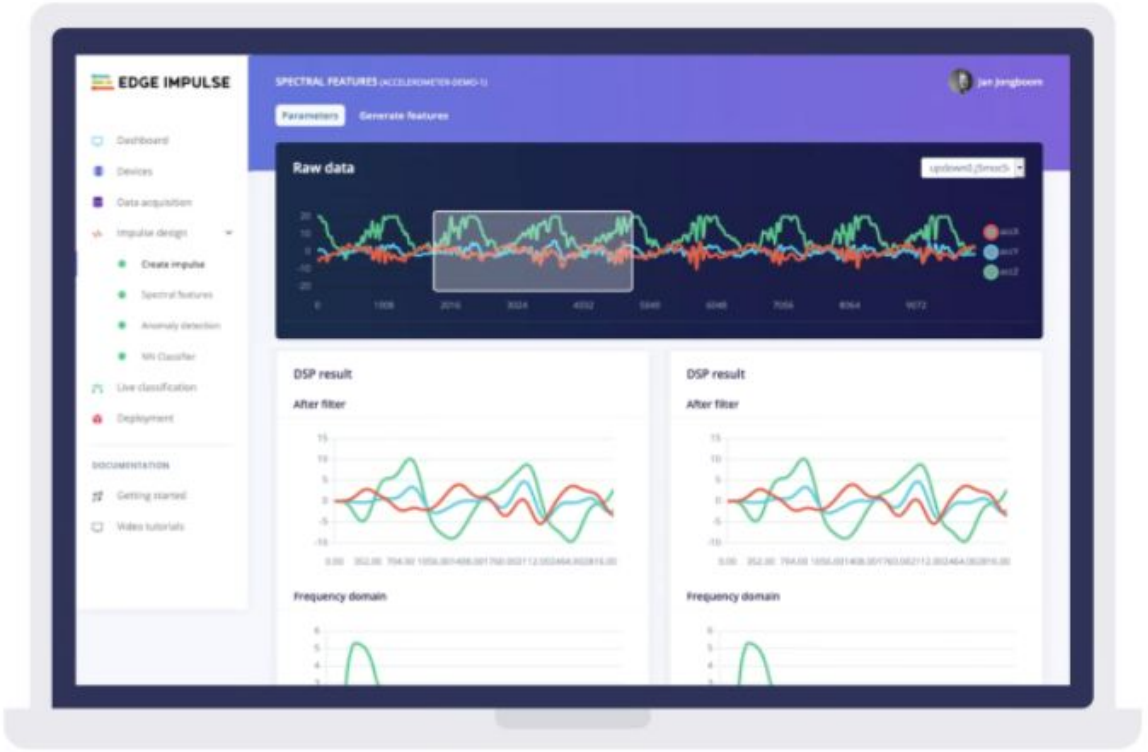
Data-driven engineering




Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Collect Data
Dataset
Impulse
Test
Deploy
Make Inferences

EI Studio - Embedded ML platform ("AutoML")





Learn more at http://edgeimpulse.com

TinyML Application

Examples

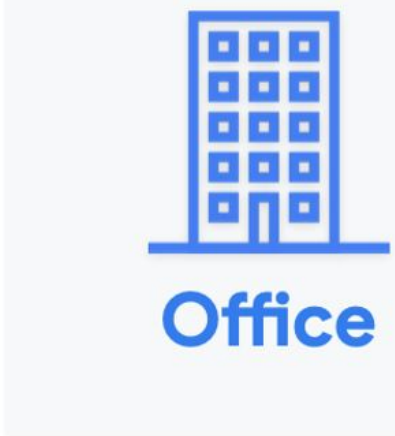
TinyML Application Areas

Home

Office

Industry

TinyML Application Areas



Endpoints Have **Sensors**, Tons of Sensors

Motion Sensors
Gyroscope, Radar, Accelerometer

Acoustic Sensors
Ultrasonic, Microphones, Geophones, Vibrometers

Environmental Sensors
Temperature, Humidity, Pressure, IR, etc.

Touchscreen Sensors
Capacitive, IR

Image Sensors
Thermal, Image

Biometric Sensors
Fingerprint, Heart rate, etc.

Force Sensors
Pressure, Strain

Rotation Sensors
Encoders

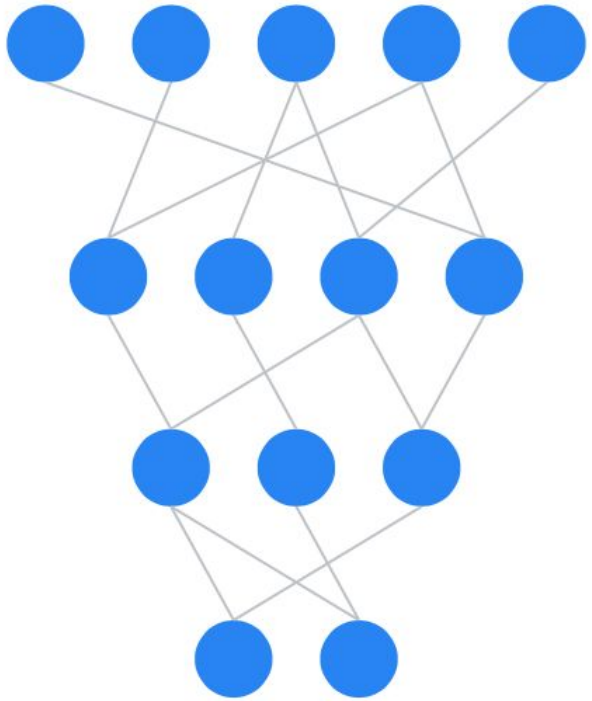
Sensors Metrics

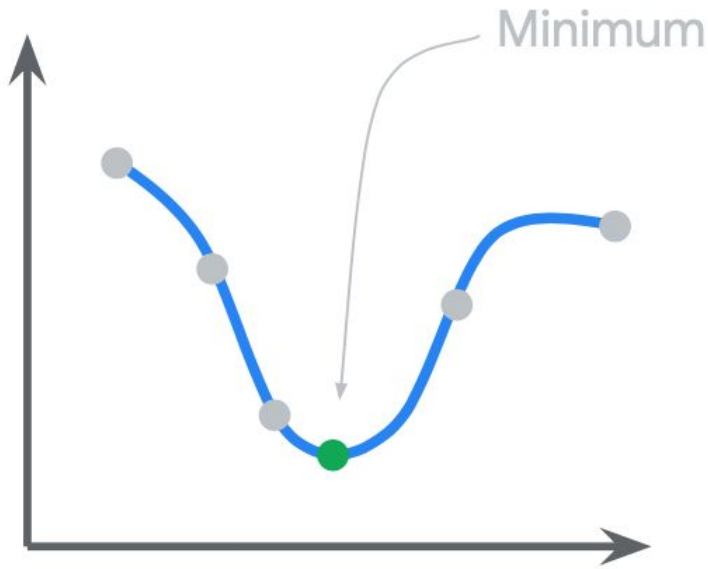
Acoustic Sensors
Ultrasonic, **Microphones**, Geophones, Vibrometers

Image Sensors
Thermal, **Image**

Motion Sensors
Gyroscope, Radar, **Accelerometer**

Models





End-to-end **TinyML** application design

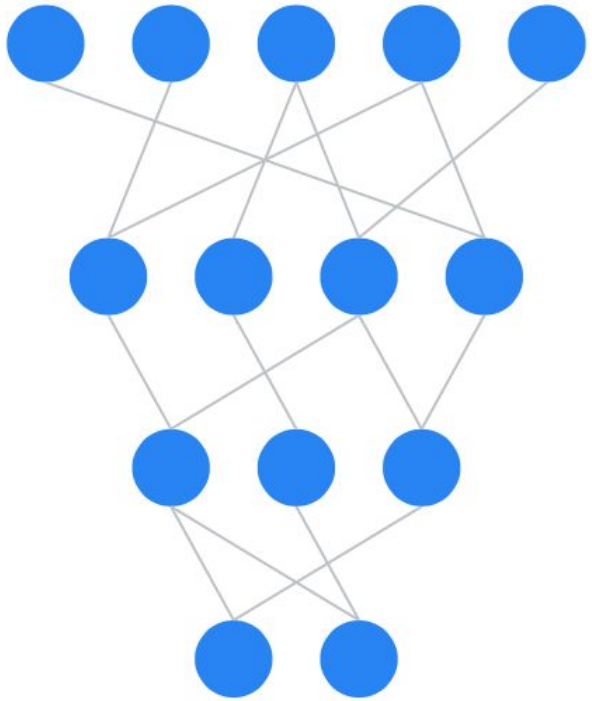
Datasets Preprocessing

Sound

Vision

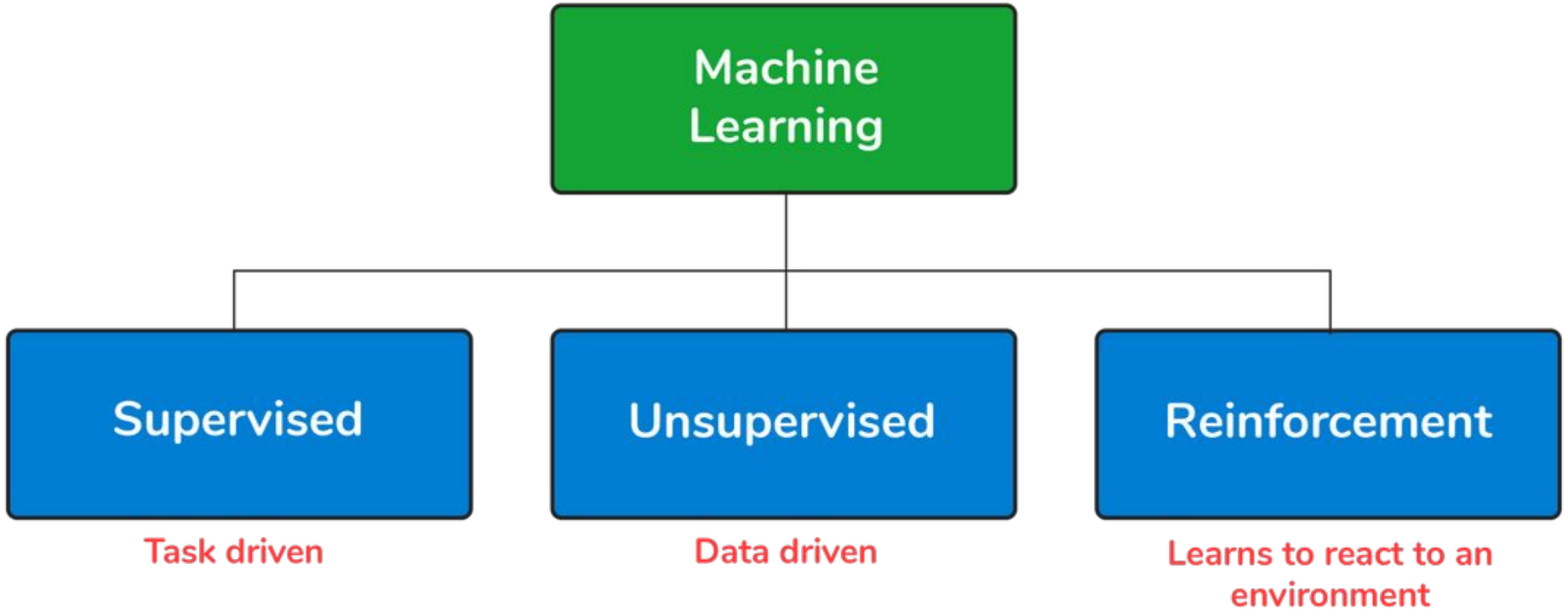
Vibration

Quantization Pruning



Resource constraints

End-to-end **TinyML** application design

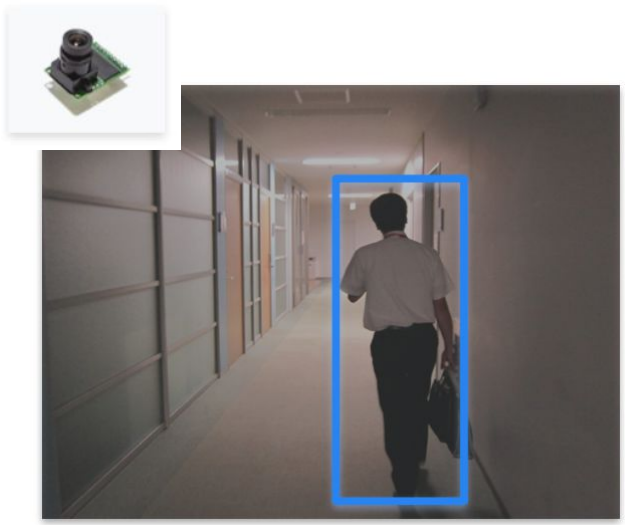


- Regression
- Classification
- Object Detection

- Anomaly Detection

- Autonomous Navigation

Vision



Sound

Vibration

Thanks

UNIFEI