

# IESTI01 - TinyML

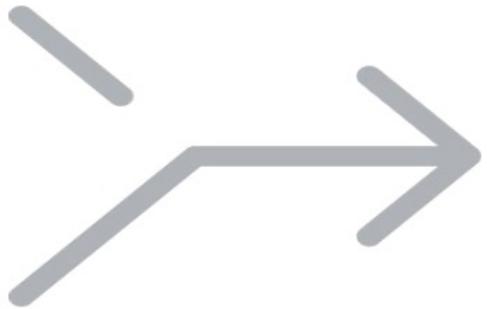
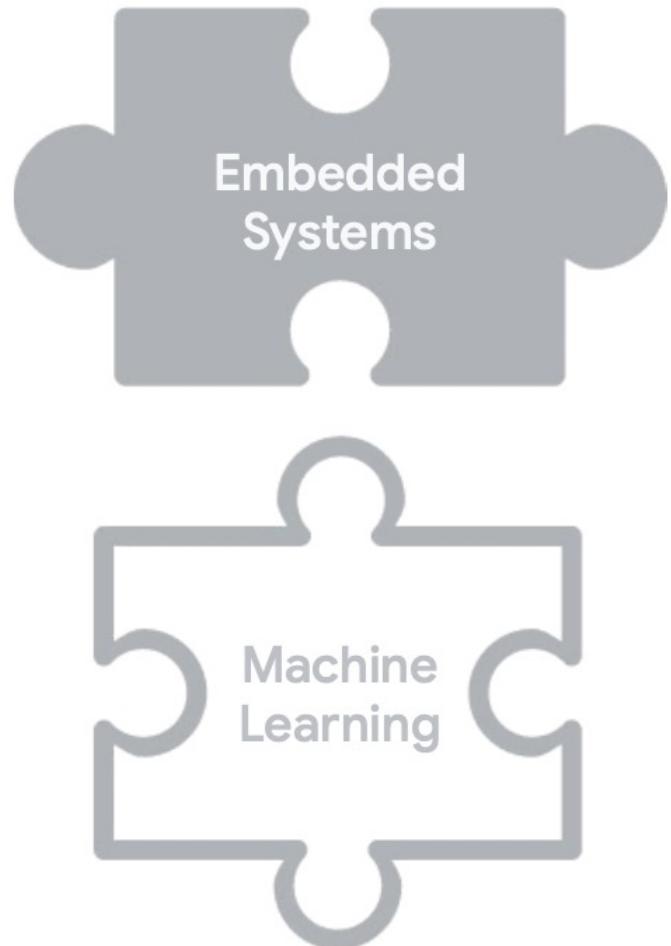
TinyML Challenges

Prof. Marcelo Rovai

May 5<sup>th</sup>, 2021

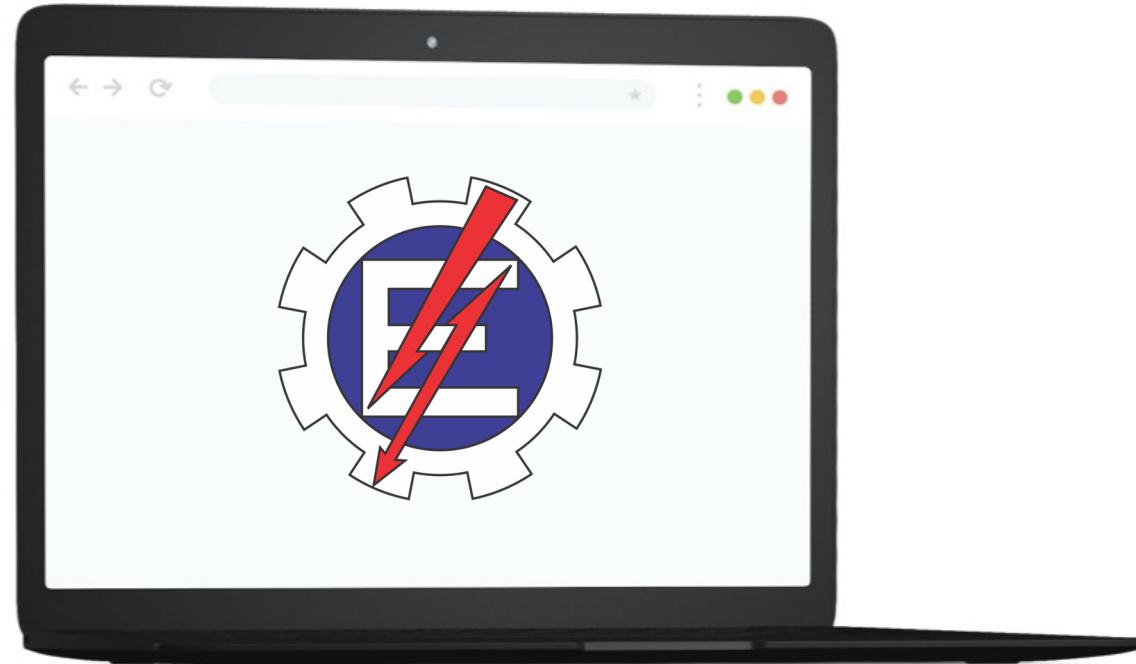


# What are the Challenges for TinyML?



**TinyML**

# Building Blocks of Computing Hardware



# Hardware



# Software

# Compute



# Memory



# Storage

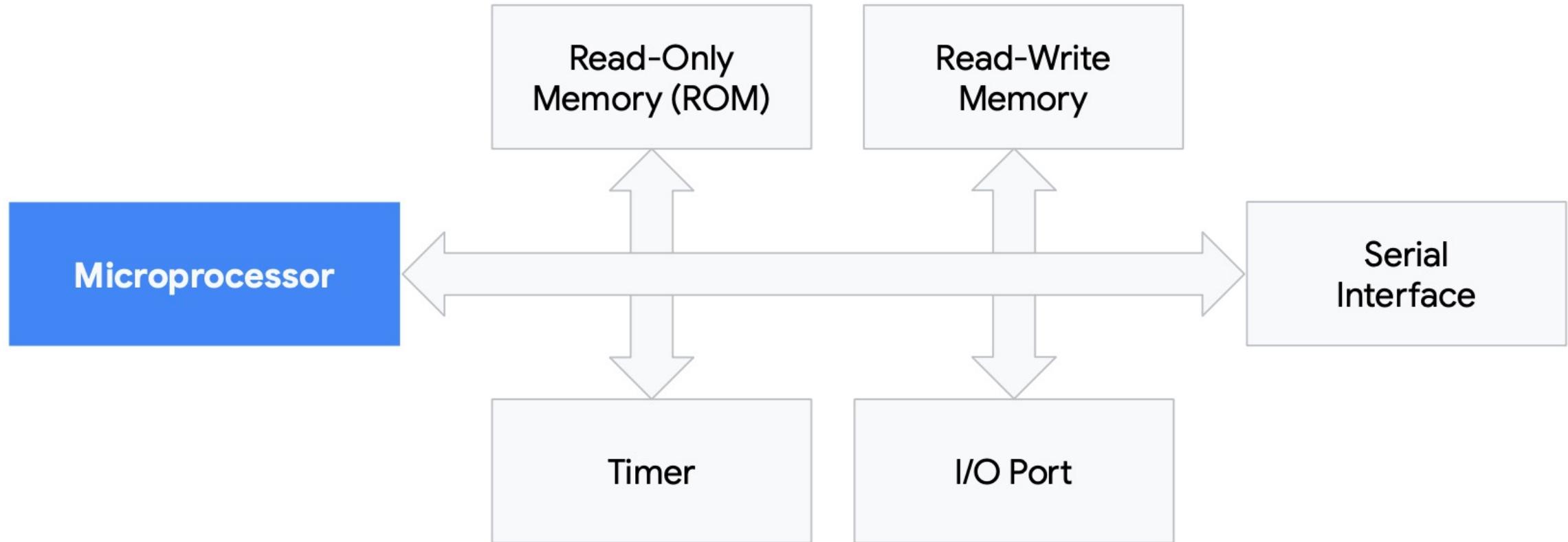


# **Microprocessor**

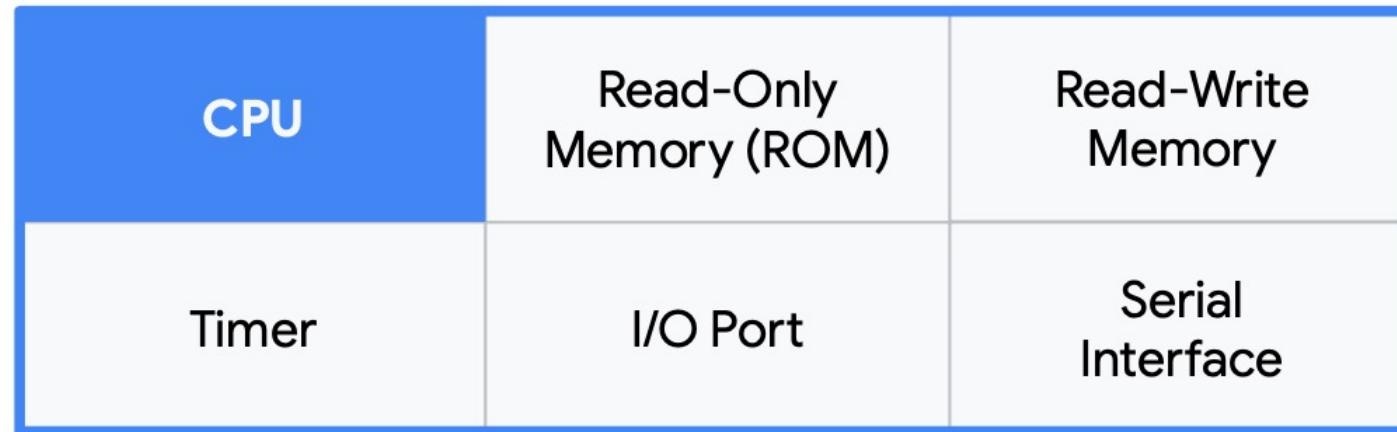
v.

# **Microcontroller**

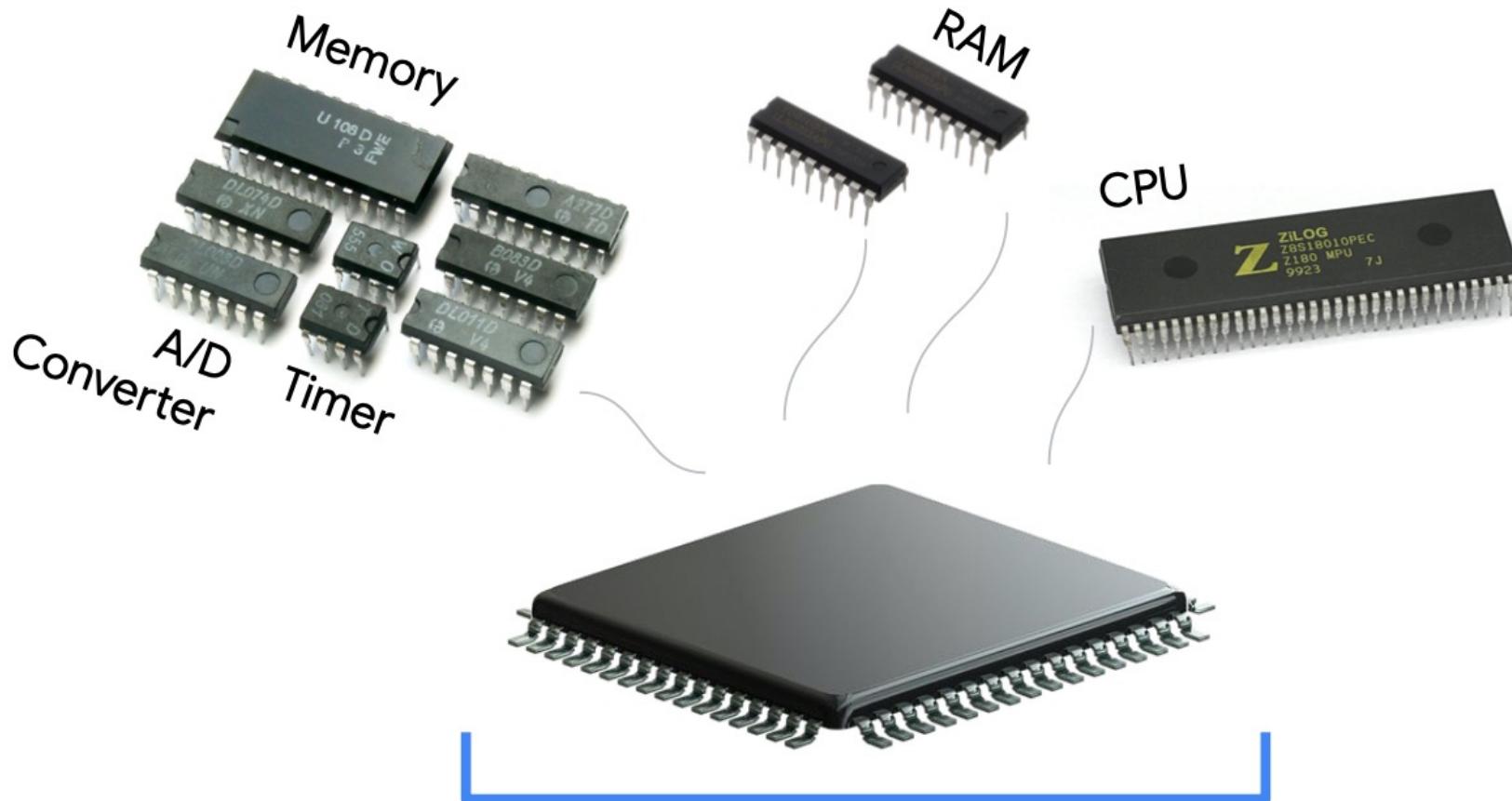
# Microprocessor: only **one part** of the puzzle



# Microcontroller



# 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**

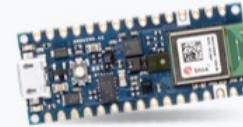
# Orders of Magnitude Difference

	<b>Microprocessor</b>	<b>&gt;</b>	<b>Microcontroller</b>	
<b>Platform</b>				Nano
<b>Compute</b>	1GHz–4GHz	~10X	1MHz–400MHz	48MHz
<b>Memory</b>	512MB–64GB	~10000X	2KB–512KB	32KB
<b>Storage</b>	64GB–4TB	~100000X	32KB–2MB	256KB
<b>Power</b>	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

150 $\mu$ W-23.5mW

# Hardware



# Software

## Software

Applications

Libraries

Operating System

## Hardware

## **Software**

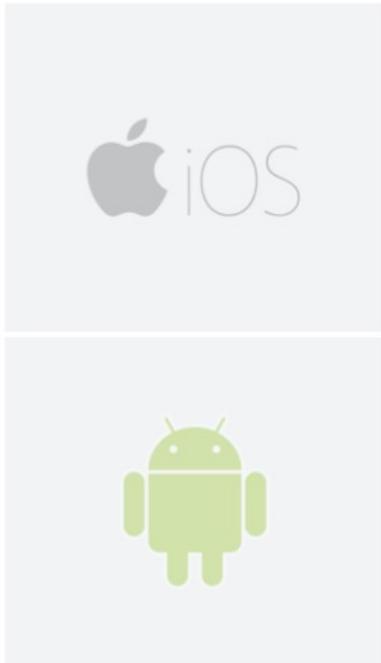
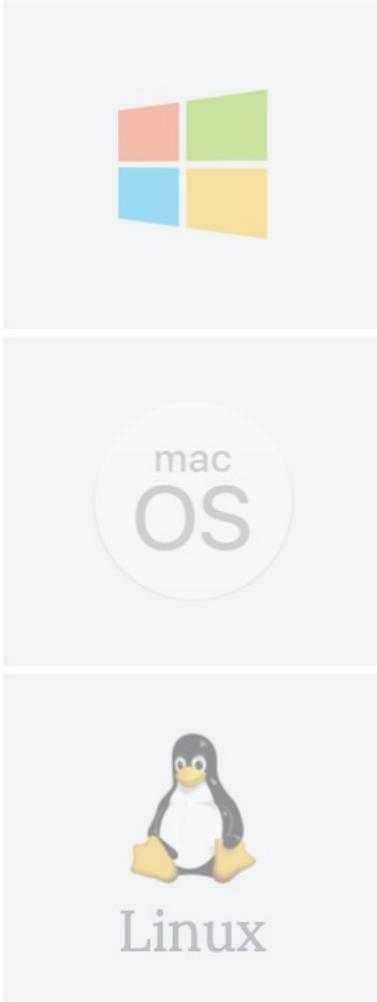
**Applications**

**Libraries**

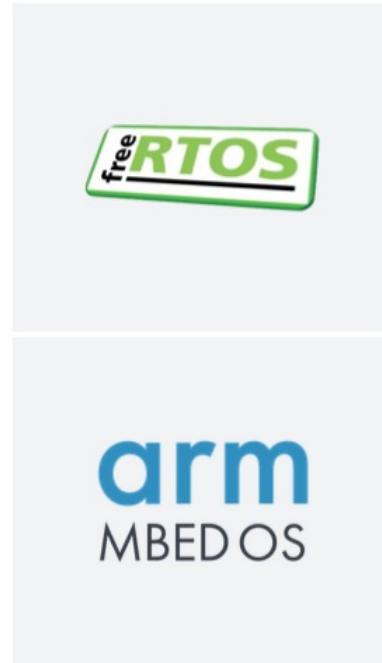
**Operating System**

**Hardware**

# Widely Used Operating Systems



Mobile OS



**Embedded Sys.**

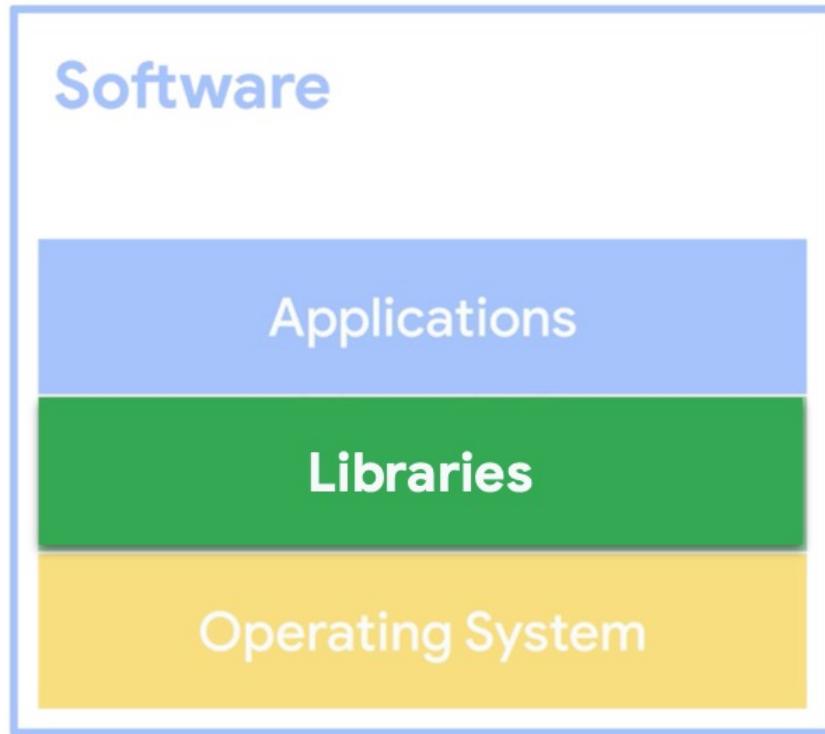
## Software

Applications

Libraries

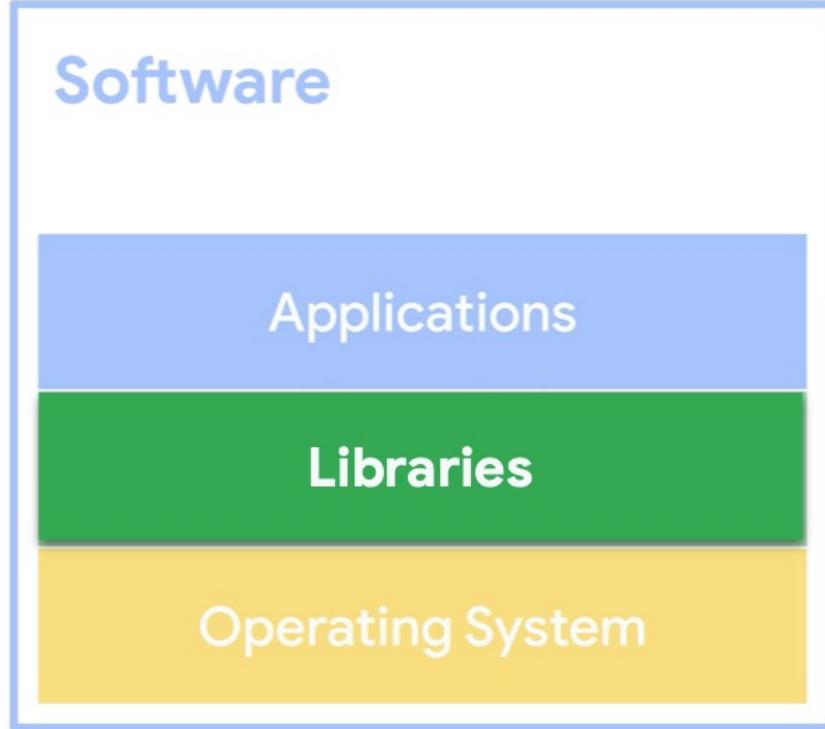
Operating System

Hardware



**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

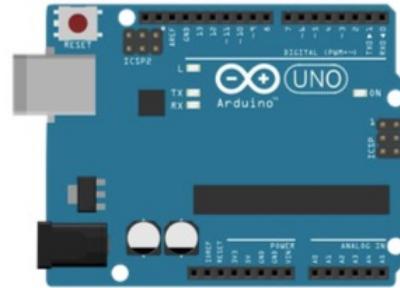
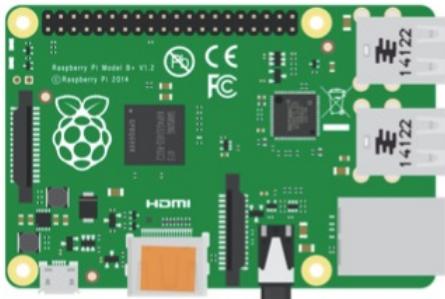
	Universal Code Portability/Compatibility	Cost (\$)	Power Consumption (W)	Engineering Effort
Option 1	✓	✗	✗	✗
	Lower Code Portability	✗		
Option 2	✓	✓	✓	✓

# Portability Trade-offs

	Universal Code Portability/Compatibility	Cost (\$)	Power Consumption (W)	Engineering Effort
Option 1	✓	✗	✗	✗
Option 2	✗	✓	✓	✓
	Lower Code Portability	✗		

# Portability Trade-offs

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



Specific HW Implementation of a Library

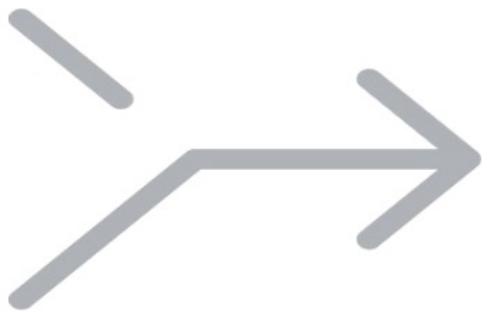
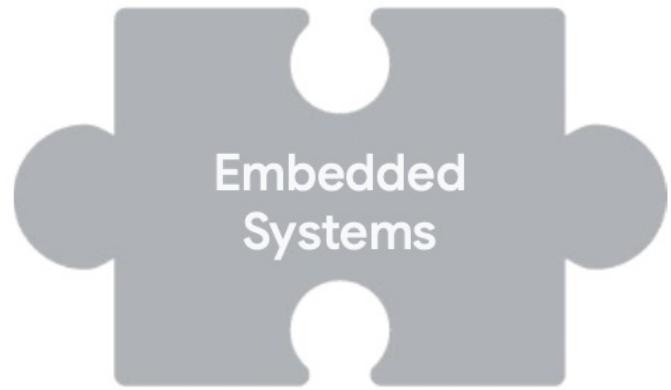
Option 2

Lower Code Portability	X
Cost (\$)	✓
Power (W)	✓
Eng. Effort	✓



# Summary

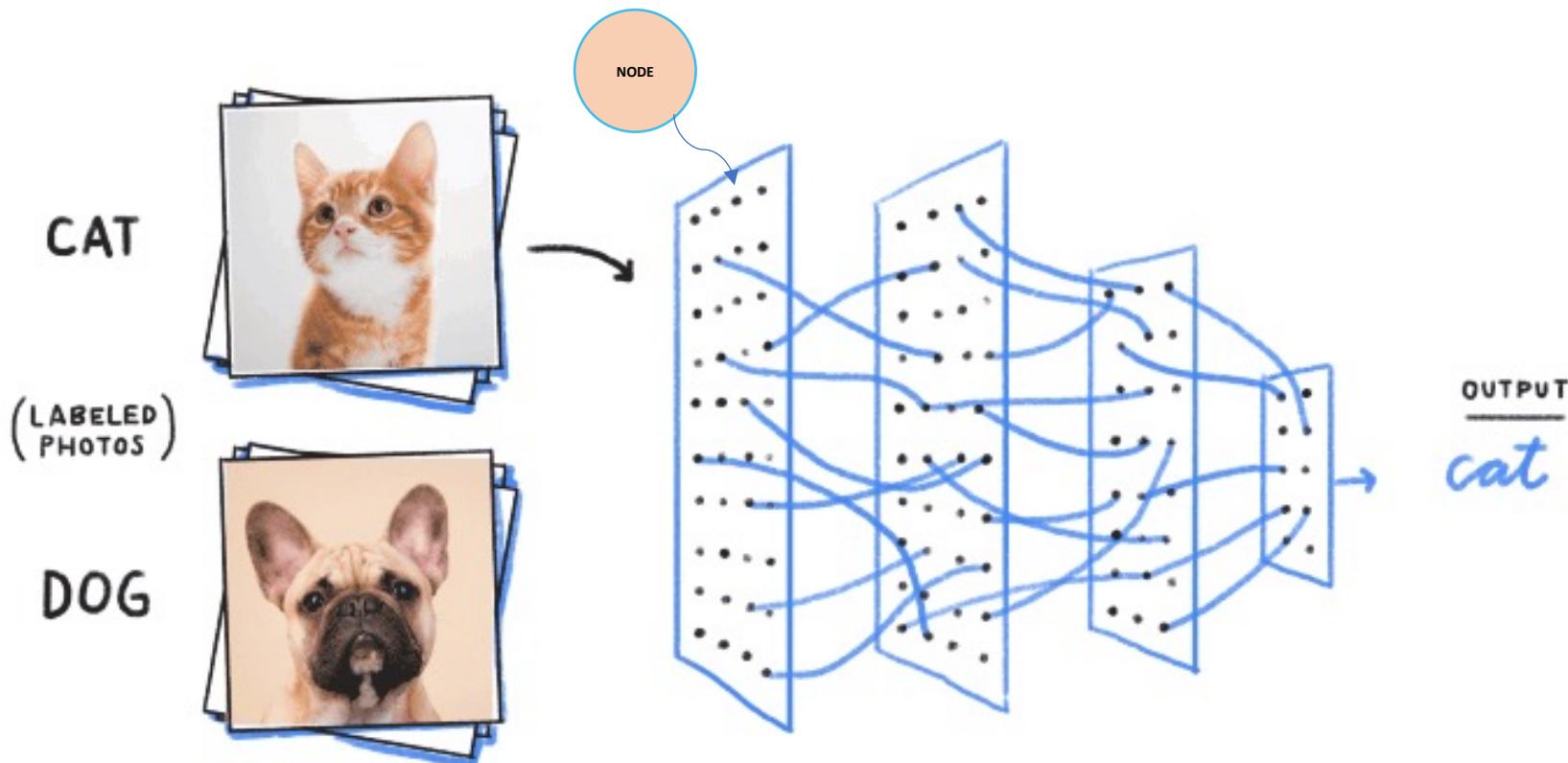
- **Embedded hardware** is extremely limited in performance, power consumption and storage
- **Embedded software** is not as portable and flexible as mainstream computing



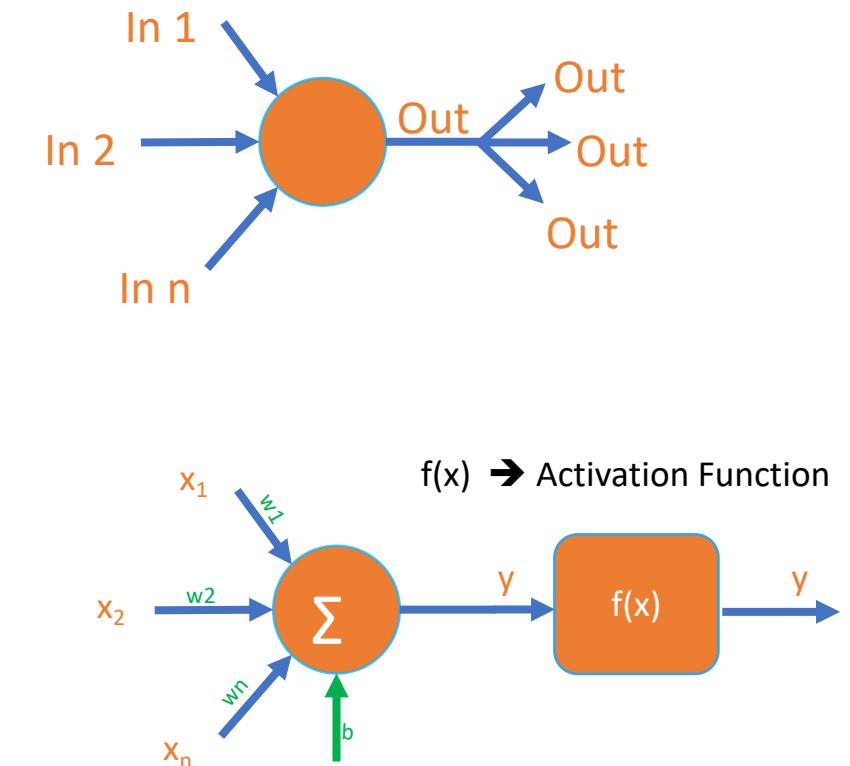
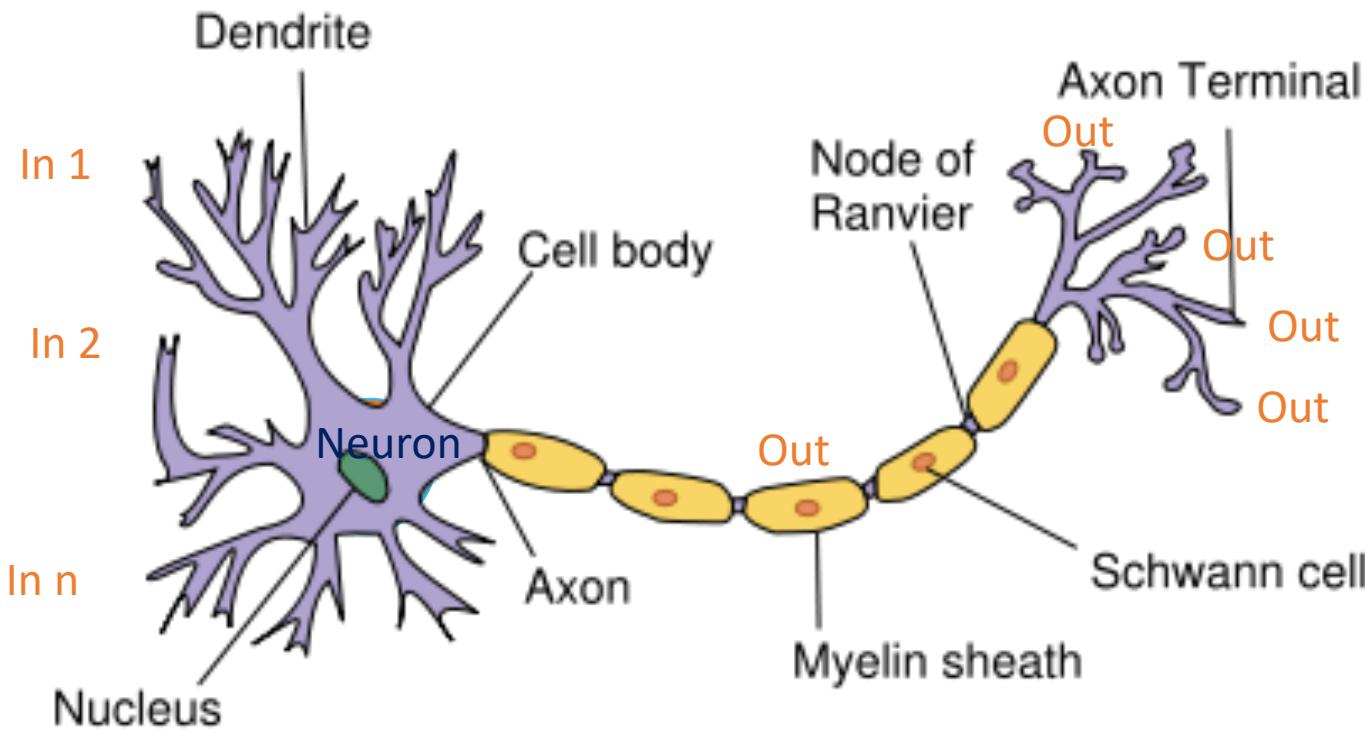
TinyML

# (Deep) Machine Learning

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



# Neuron (Perceptron)



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

# The Neural Network Model Architecture

- Input Cell
- Backfed Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Capsule Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Gated Memory Cell
- Kernel
- Convolution or Pool

## A mostly complete chart of Neural Networks

©2019 Fjodor van Veen & Stefan Leijnen asimovinstitute.org

Perceptron (P)



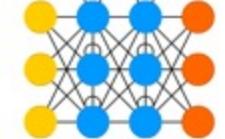
Feed Forward (FF)



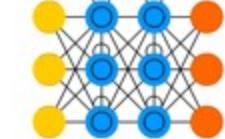
Radial Basis Network (RBF)



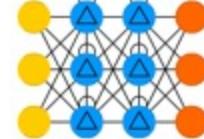
Recurrent Neural Network (RNN)



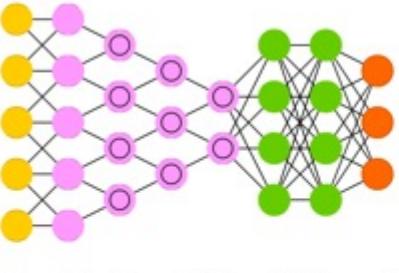
Long / Short Term Memory (LSTM)



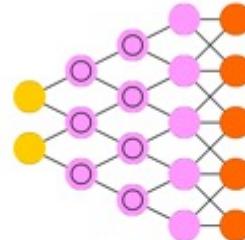
Gated Recurrent Unit (GRU)



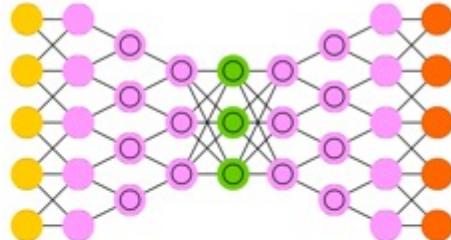
Deep Convolutional Network (DCN)



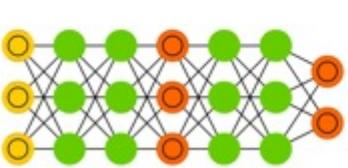
Deconvolutional Network (DN)



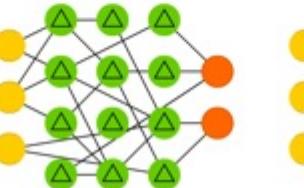
Deep Convolutional Inverse Graphics Network (DCIGN)



Generative Adversarial Network (GAN)



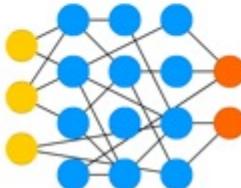
Liquid State Machine (LSM)



Extreme Learning Machine (ELM)



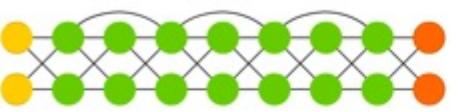
Echo State Network (ESN)



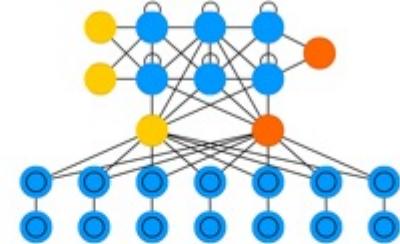
Sparse AE (SAE)



Deep Residual Network (DRN)



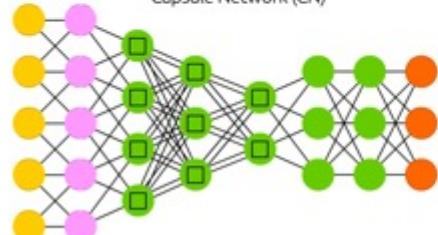
Differentiable Neural Computer (DNC)



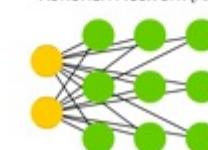
Neural Turing Machine (NTM)



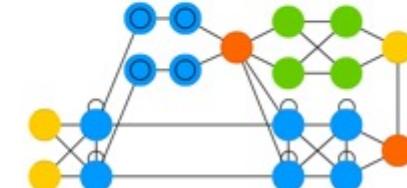
Capsule Network (CN)



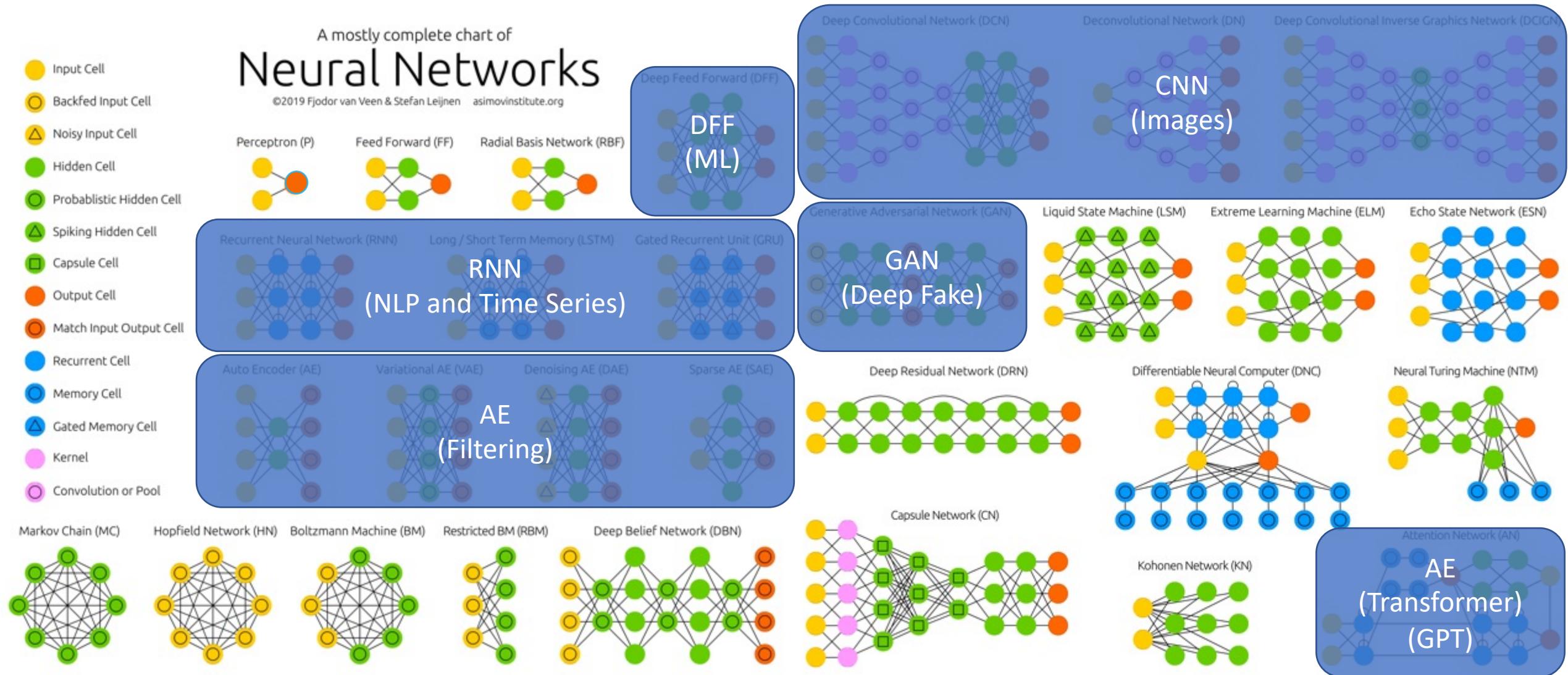
Kohonen Network (KN)



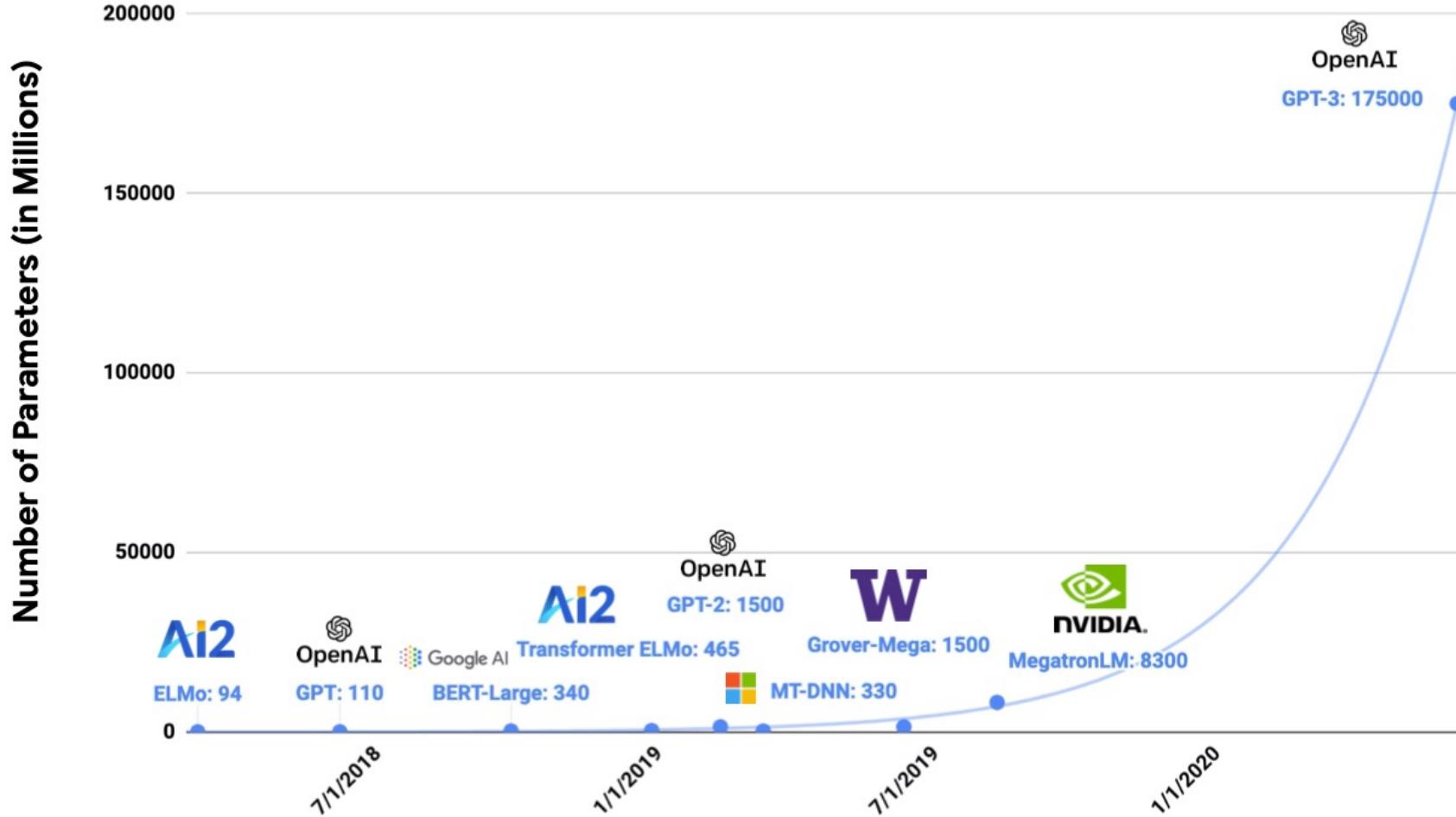
Attention Network (AN)



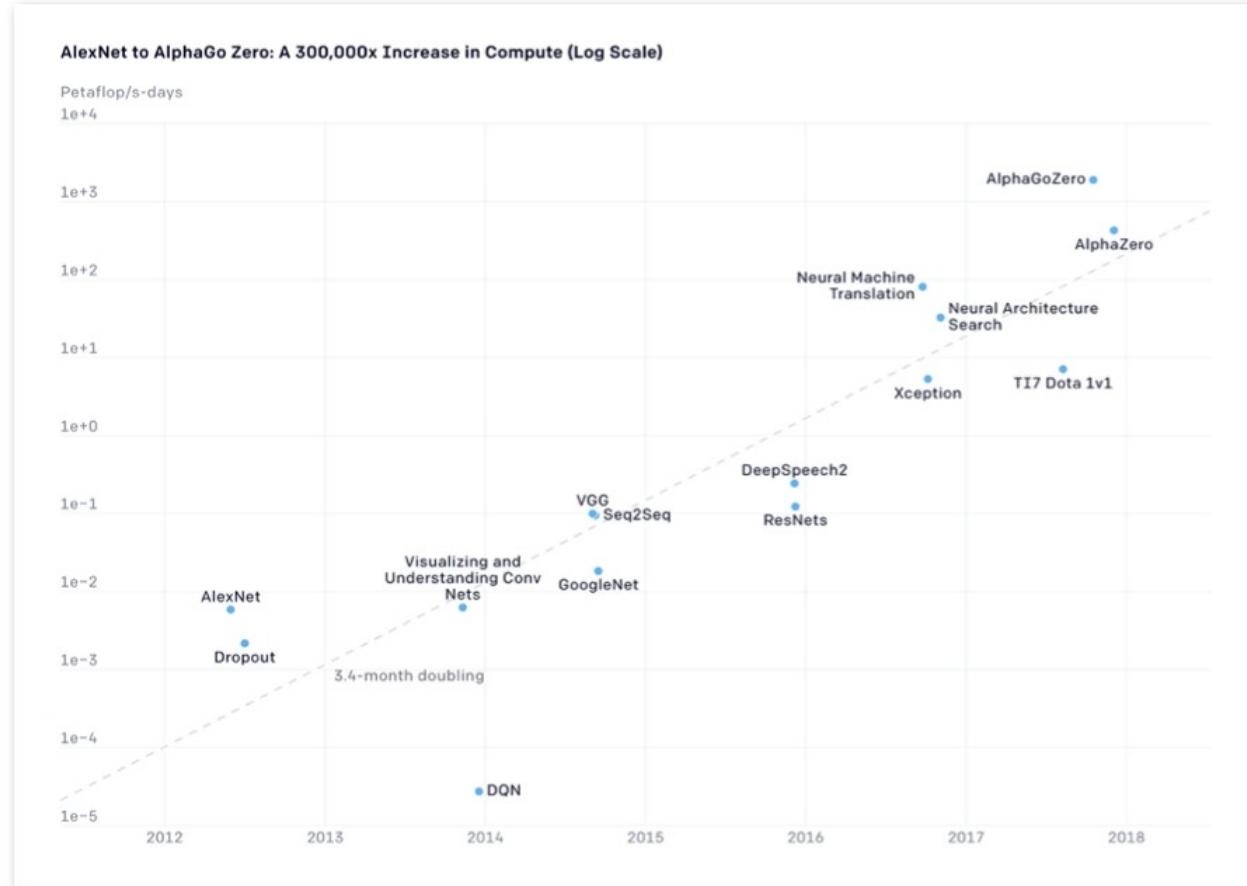
# The Neural Network Model Architecture



# ML Model Size Growth



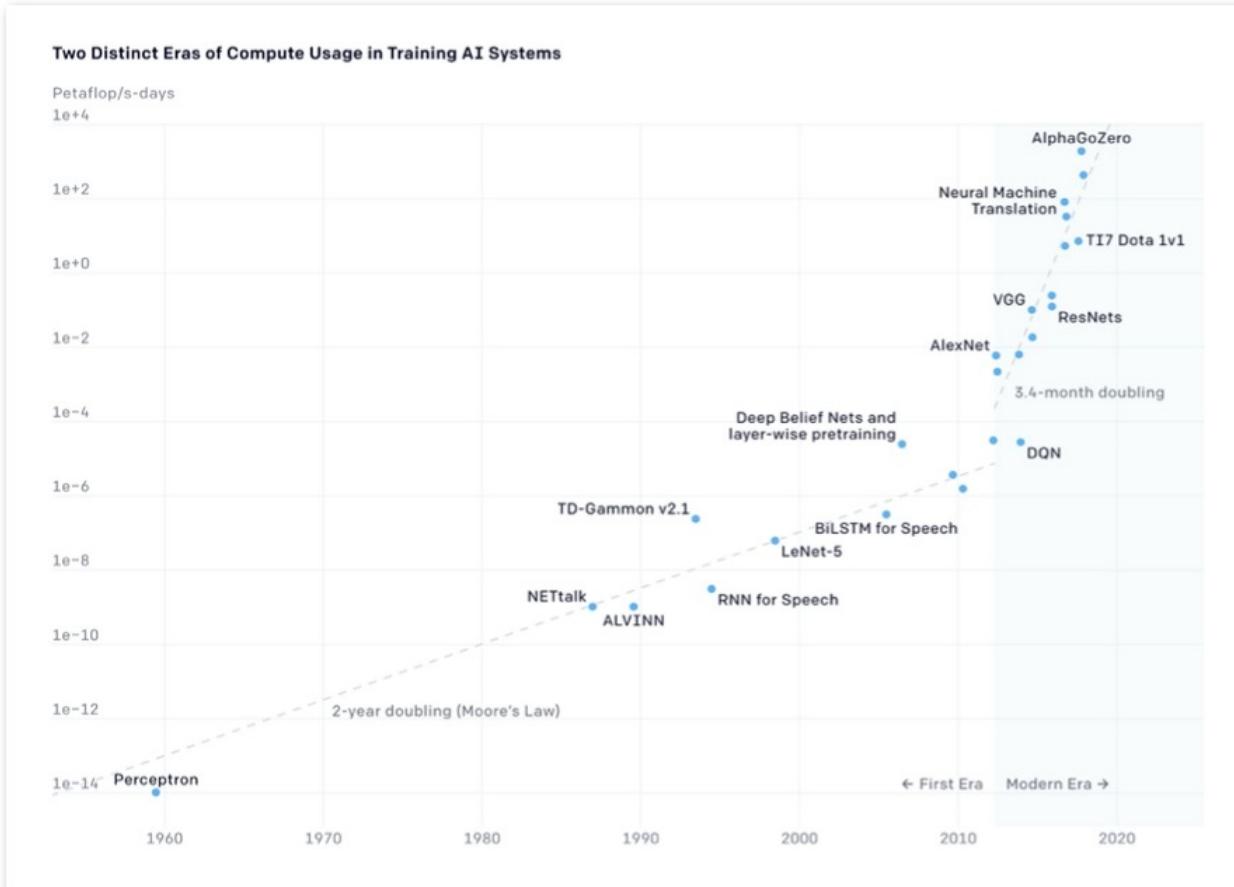
# ML Compute Needs (2012 to Present Day)



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

Source: <https://openai.com>

# ML Compute Needs (from the 1960s)



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

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

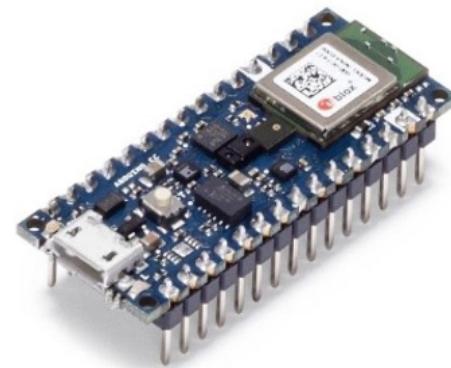


A large, light-grey arrow pointing downwards, indicating a flow or connection between the data center above and the Cloud TPU below.

Cloud TPU

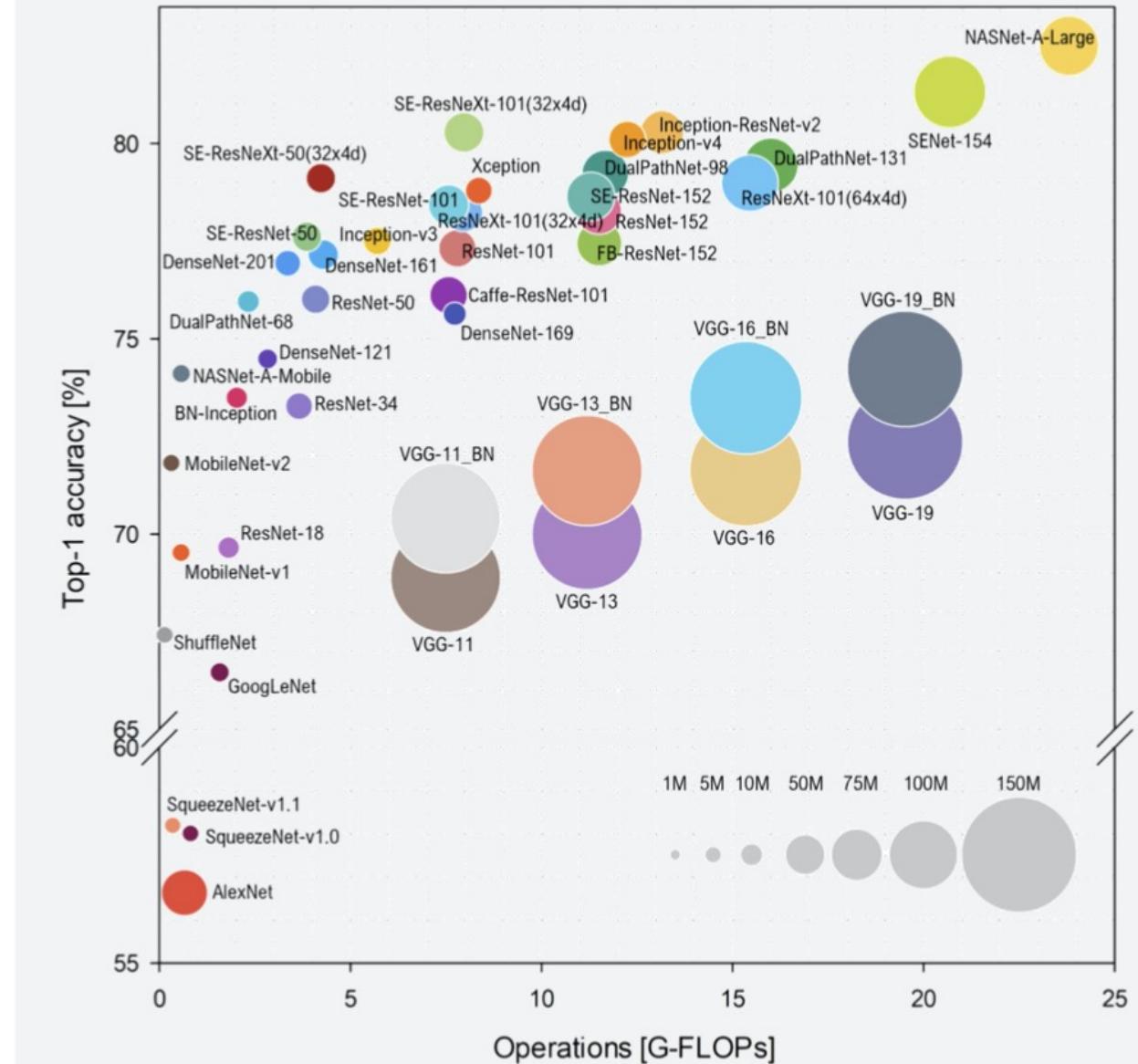


TinyML



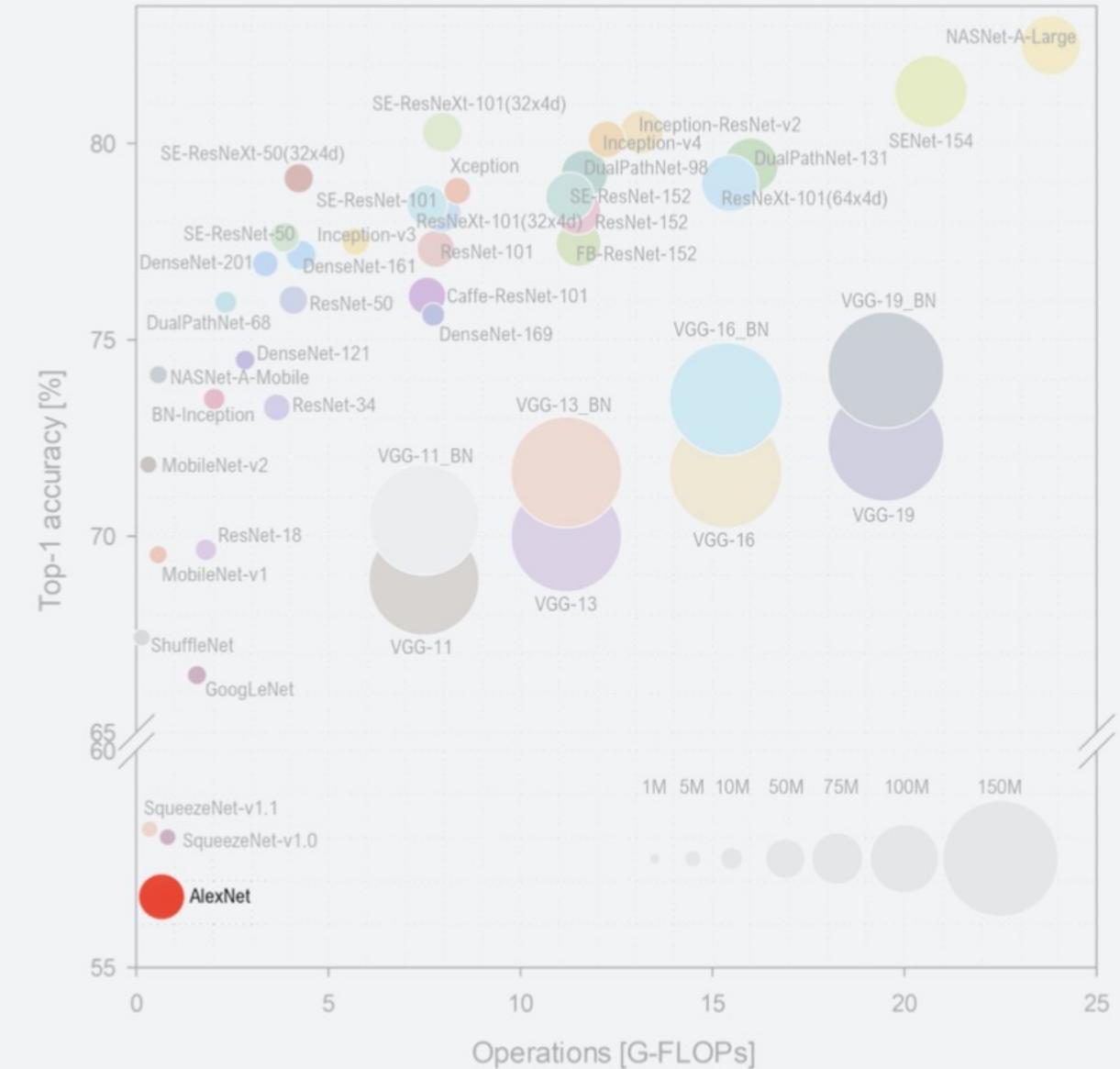
# ML Model Evolution

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

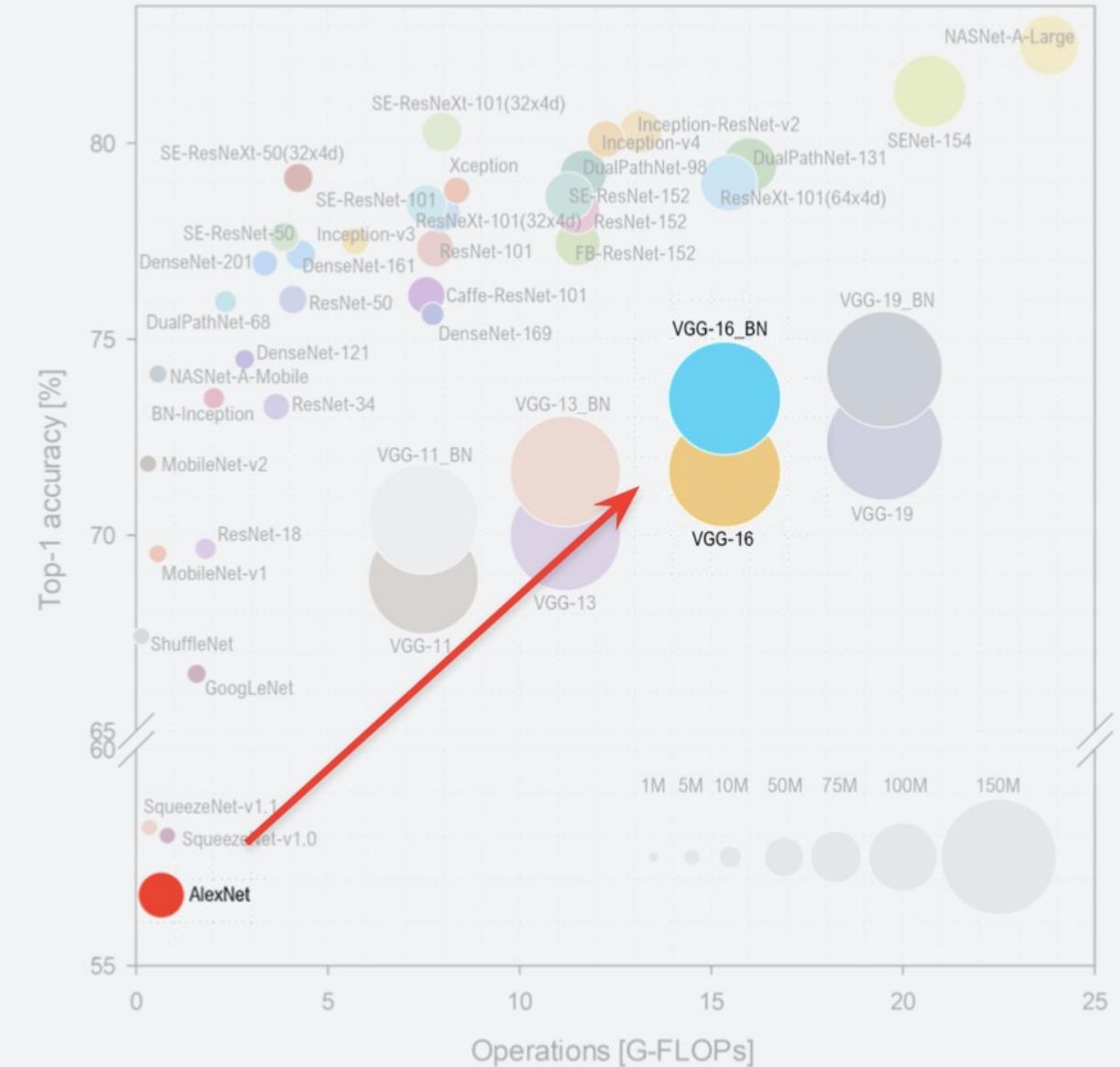
- **AlexNet (2012)**
  - 57.1% accuracy
  - 61MB in size



**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

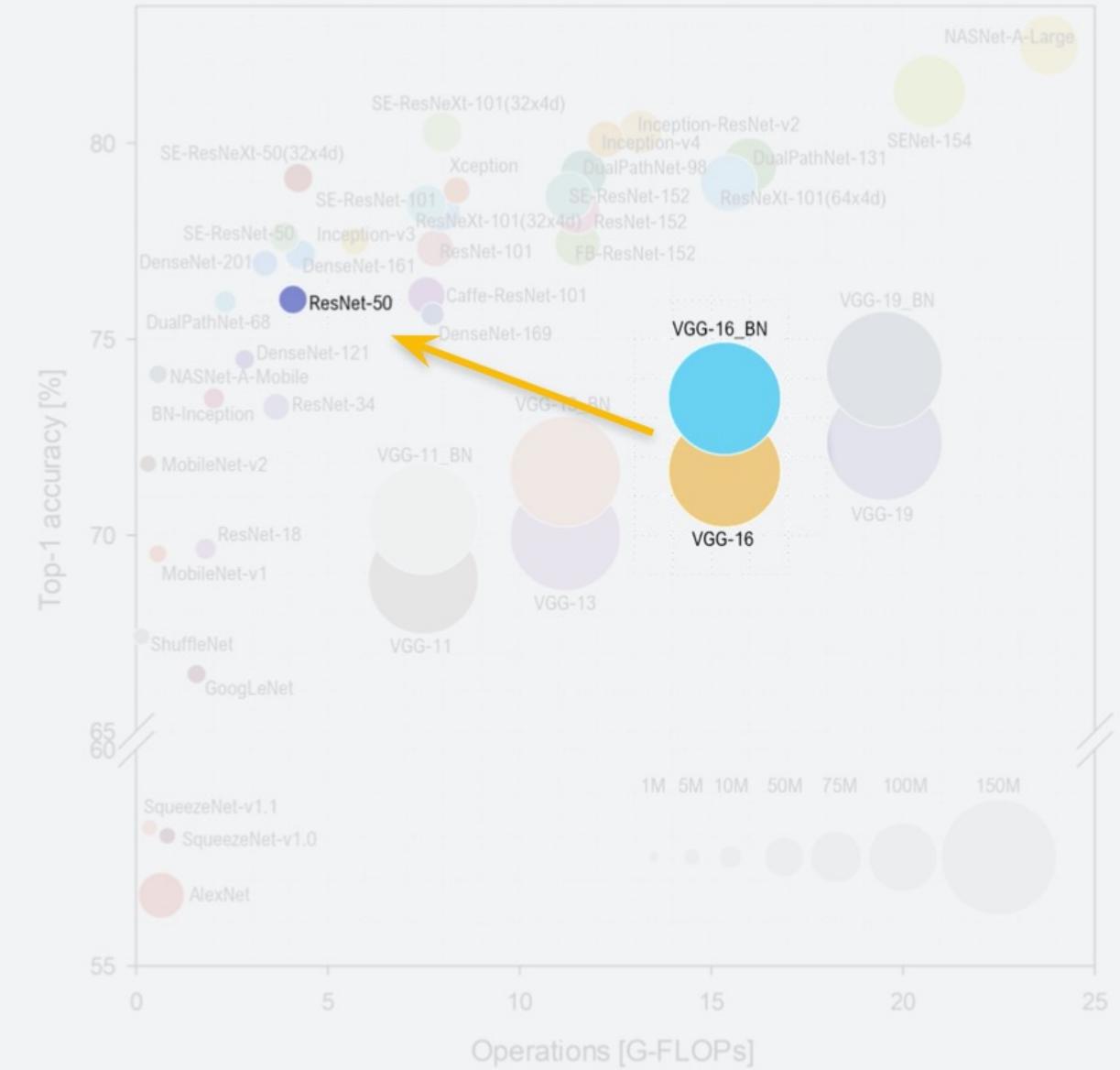
- **VGGNet (2014) [VGG-16]**
  - **71.5% accuracy**
  - **528MB** in size



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

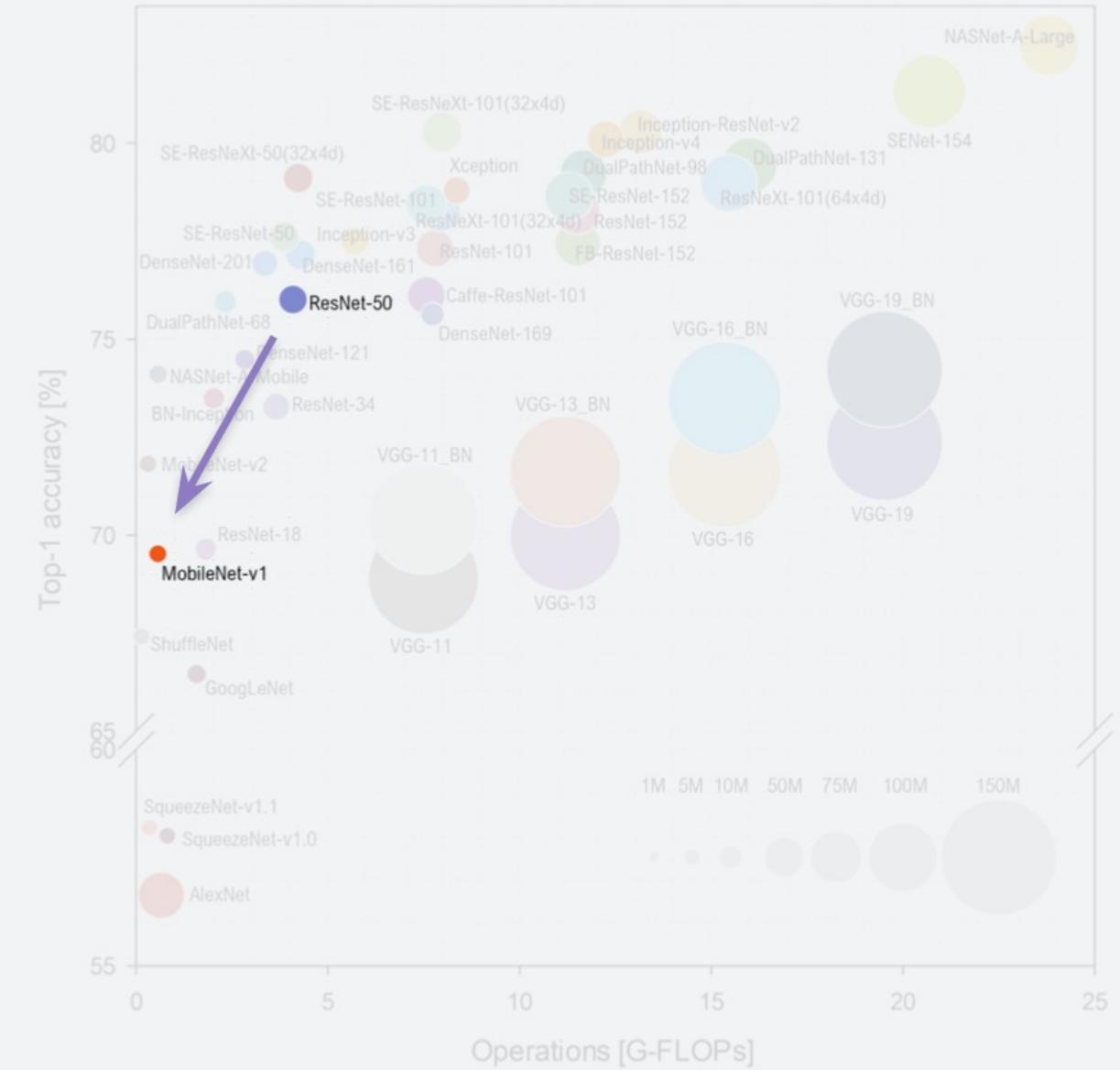
- **ResNet (2015)**
  - **75.8% accuracy**
  - **22.7MB** in size



**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



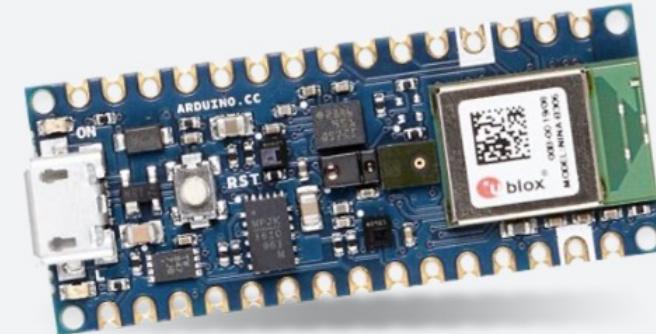
**Source:** S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018

# ML Model Evolution

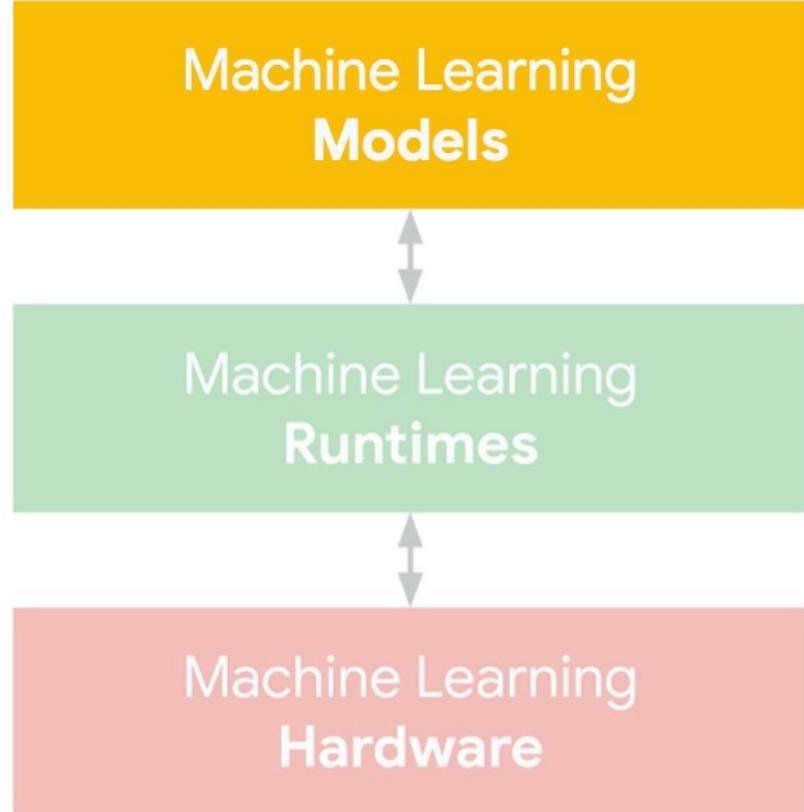
- **MobileNet (2015)**
  - **MobileNetv1**
    - 70.6% accuracy
    - 16.9MB in size

## Problem:

Our board (in your kit for Course 3) only has **256KB** of RAM (memory) yet **MobileNetv1** needs **16.9MB!**



**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



# Model Compression Techniques

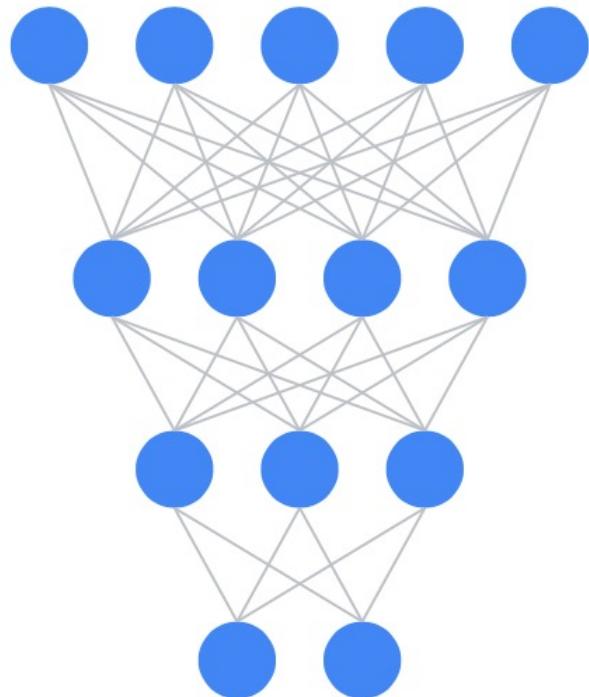
**Pruning**

Quantization

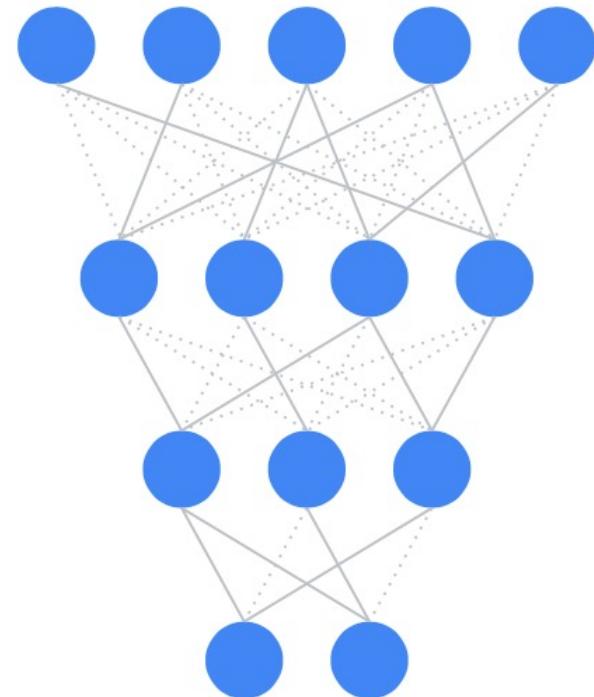
Knowledge Distillation

...

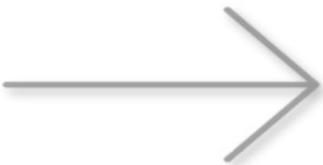
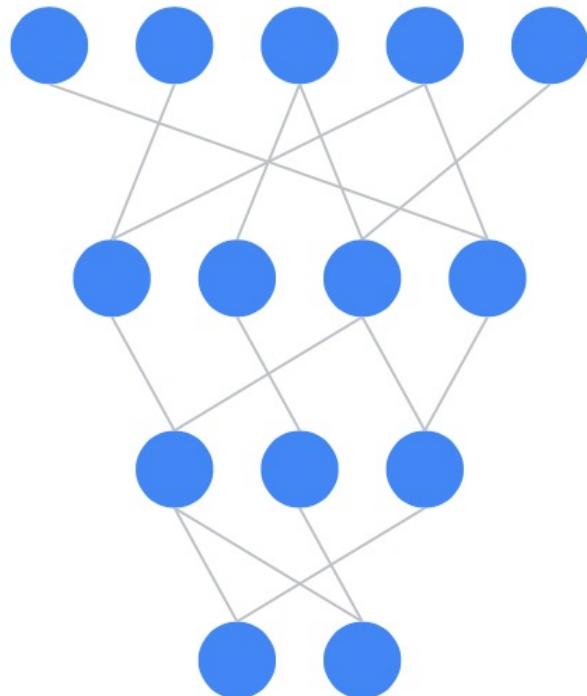
# Pruning



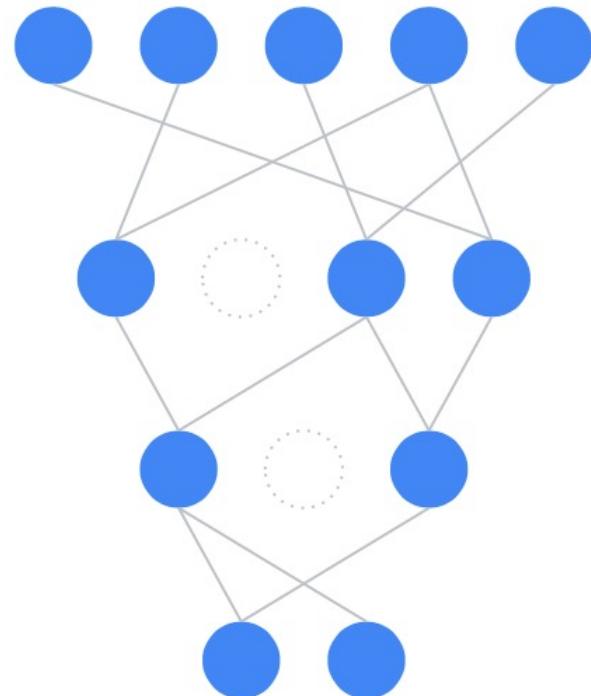
**PRUNING  
SYNAPSES**

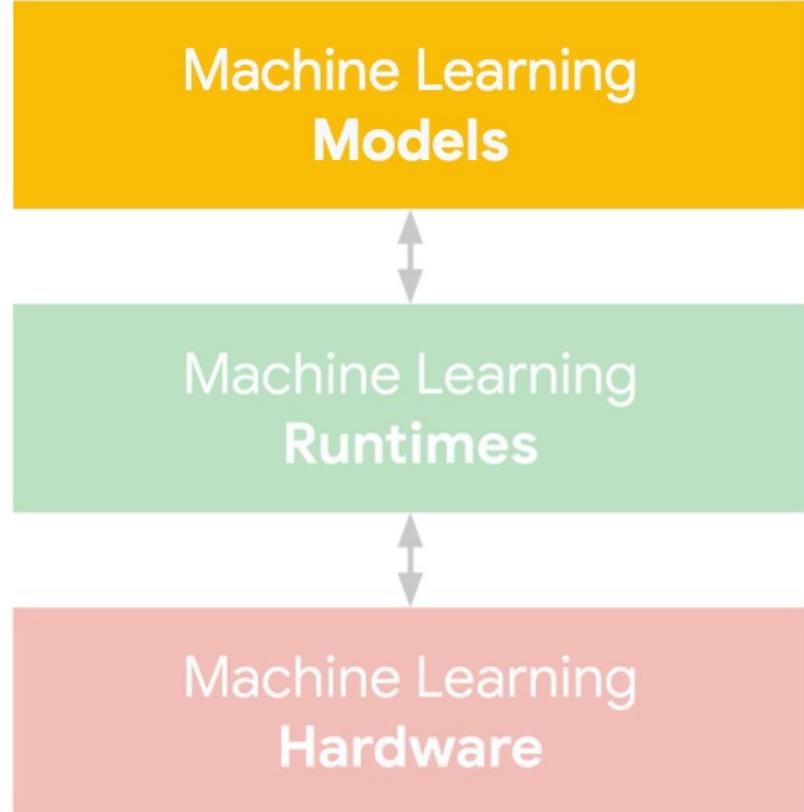


# Pruning



**PRUNING  
NEURONS**





# Model Compression Techniques

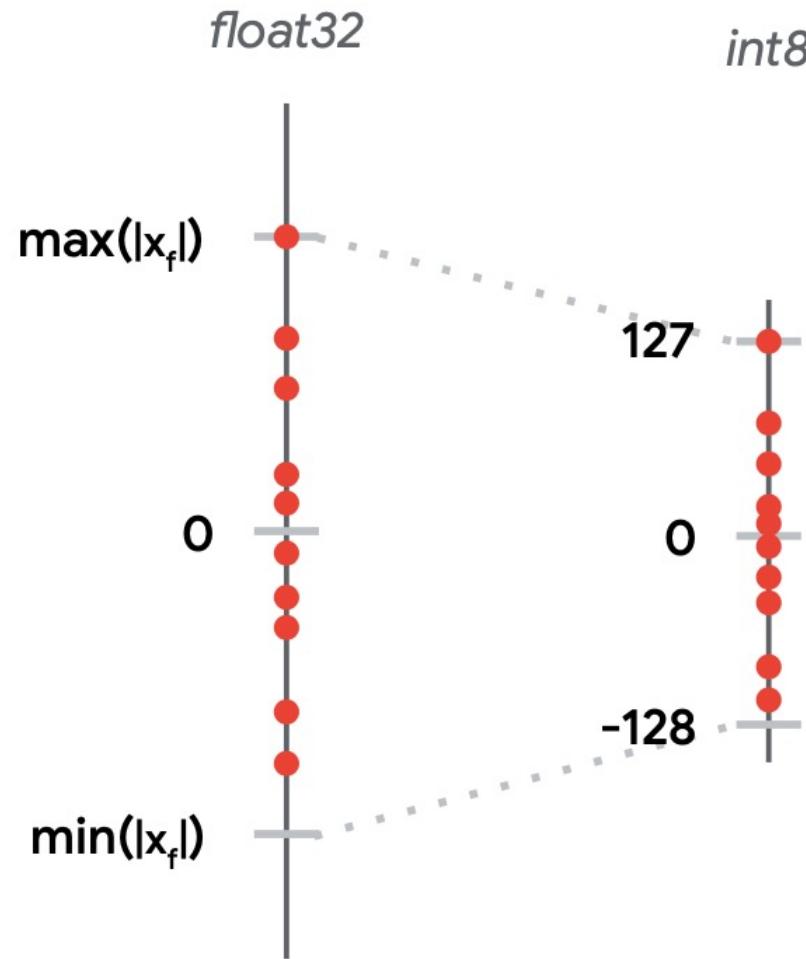
Pruning

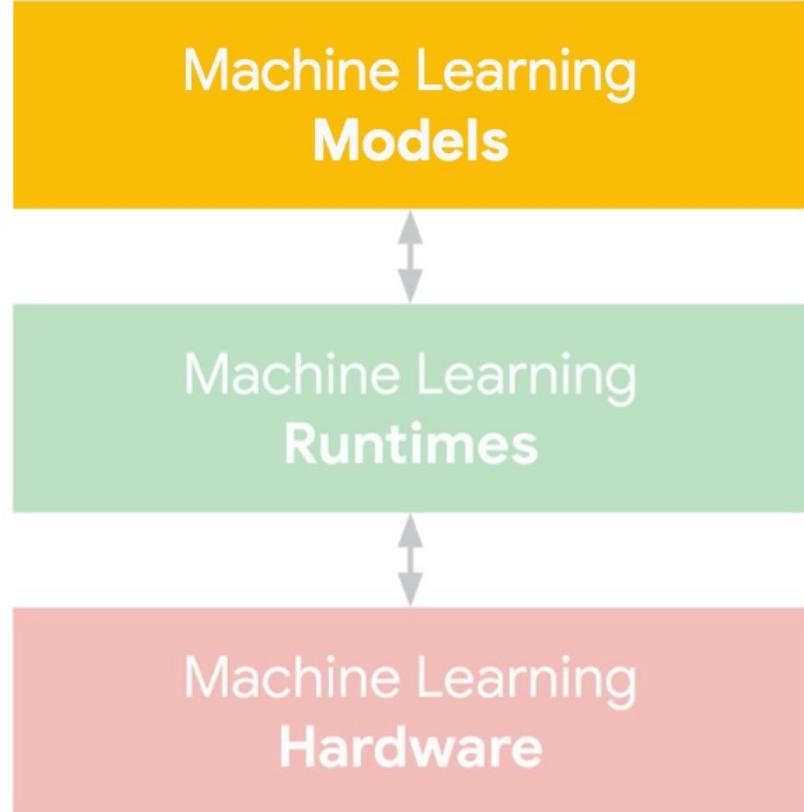
**Quantization**

Knowledge Distillation

...

# Quantization





# Model Compression Techniques

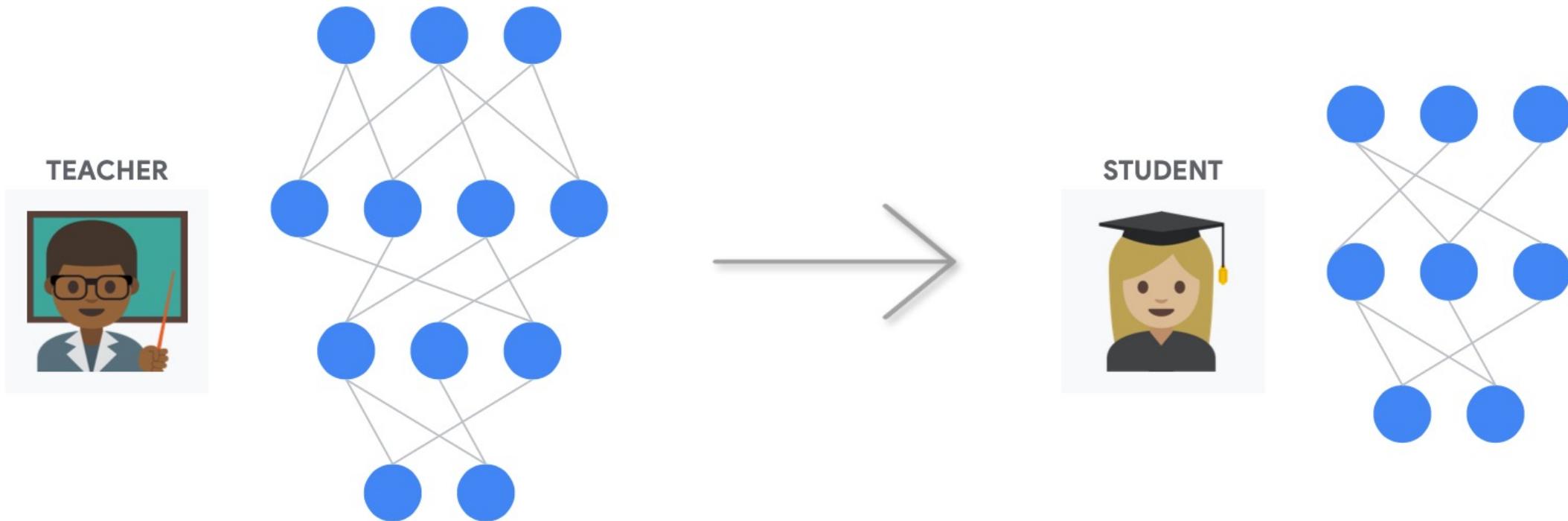
Pruning

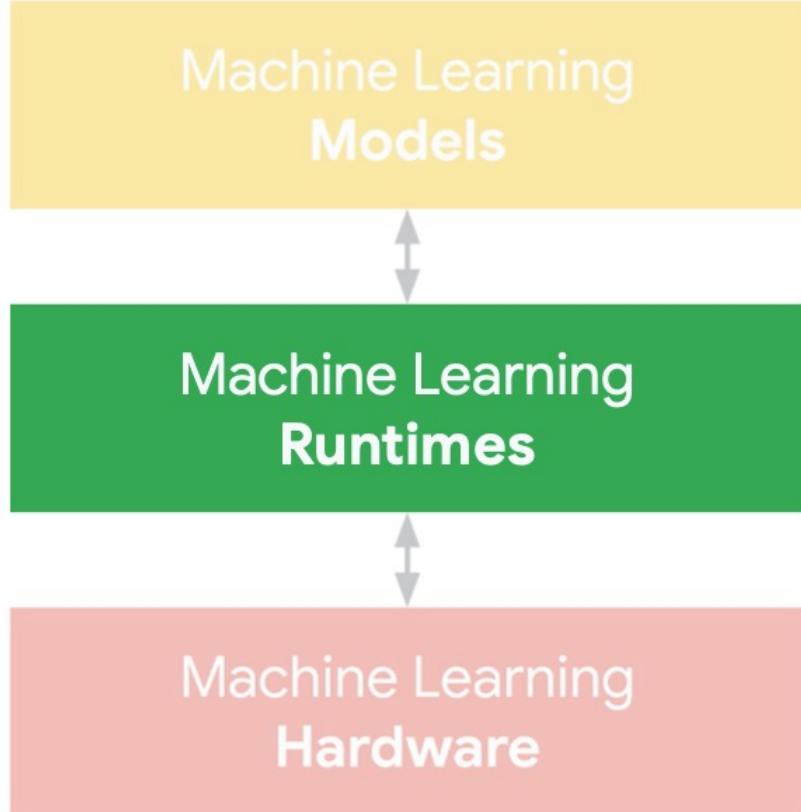
Quantization

**Knowledge Distillation**

...

# Knowledge Distillation





[\[TF Video\]](#)





Less memory

Less compute power

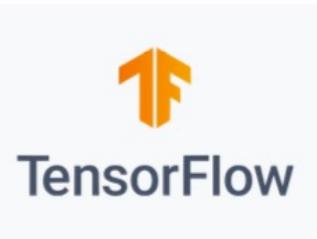
Only focused on *inference*



TensorFlow Lite

# Key Differences

**Topology**  
**Weights**  
**Binary Size**  
**Distributed Compute**  
**Developer Background**

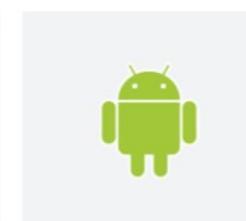
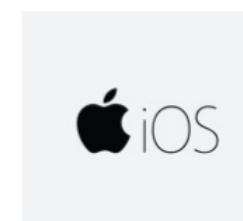
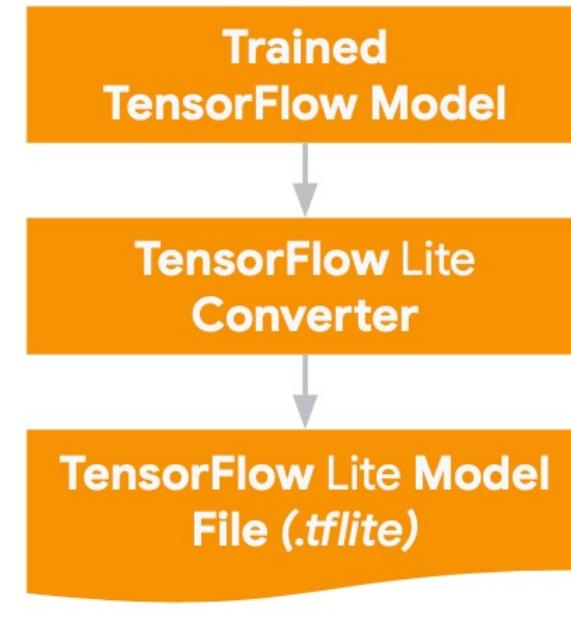
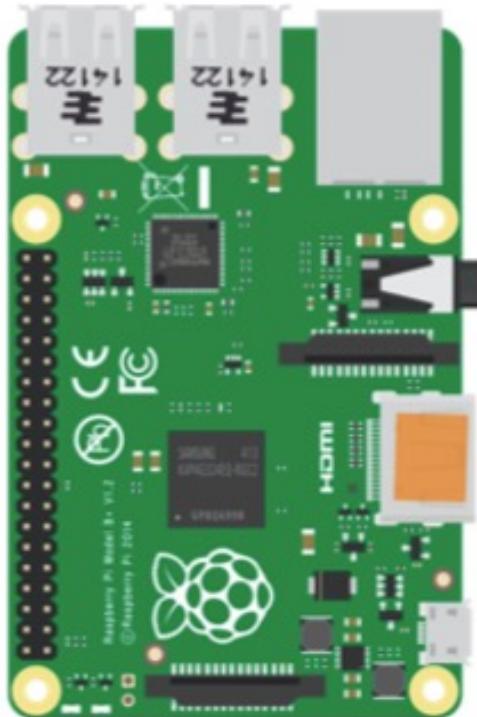


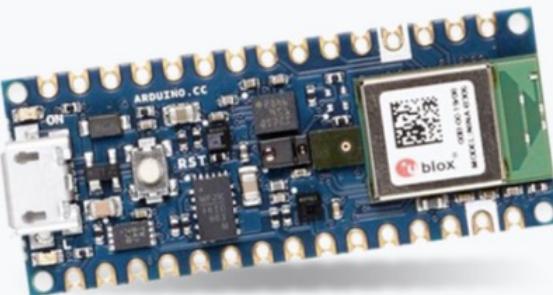
**Variable**  
**Variable**  
**Unimportant**  
**Needed**  
**ML Researcher**



**Fixed**  
**Fixed**  
**High Priority**  
**Not Needed**  
**Application Developer**

# Architecture





**Even less memory**

**Even less compute power**

**Also, only focused on *inference***



TensorFlow



TensorFlow Lite

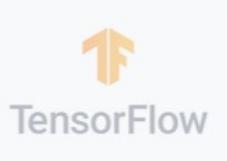
Train a model

Convert  
model

Optimize  
model

Deploy  
model at  
Edge

Make  
inferences  
at Edge



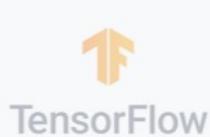
Train a model

Convert  
model

Optimize  
model

Deploy  
model at  
Edge

Make  
inferences  
at Edge



TensorFlow



TensorFlow Lite

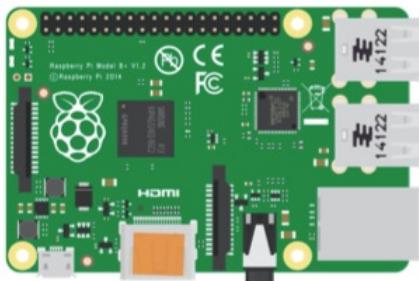
Train a model

Convert  
model

Optimize  
model

Deploy  
model at  
Edge

Make  
inferences  
at Edge



Raspberry Pi



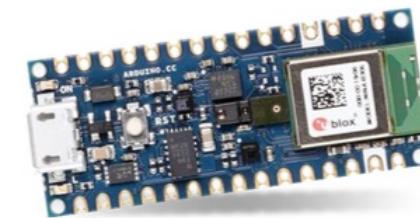
Linux



iOS



(TF Micro)



Microcontroller



TensorFlow



TensorFlow Lite

Train a model

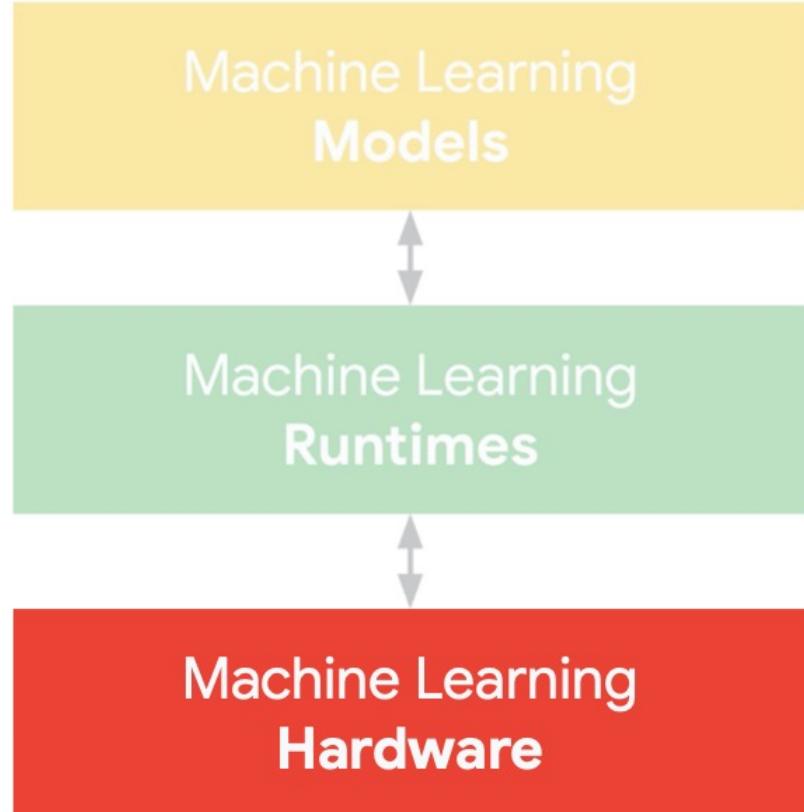
Convert  
model

Optimize  
model

Deploy  
model at  
Edge

Make  
inferences  
at Edge

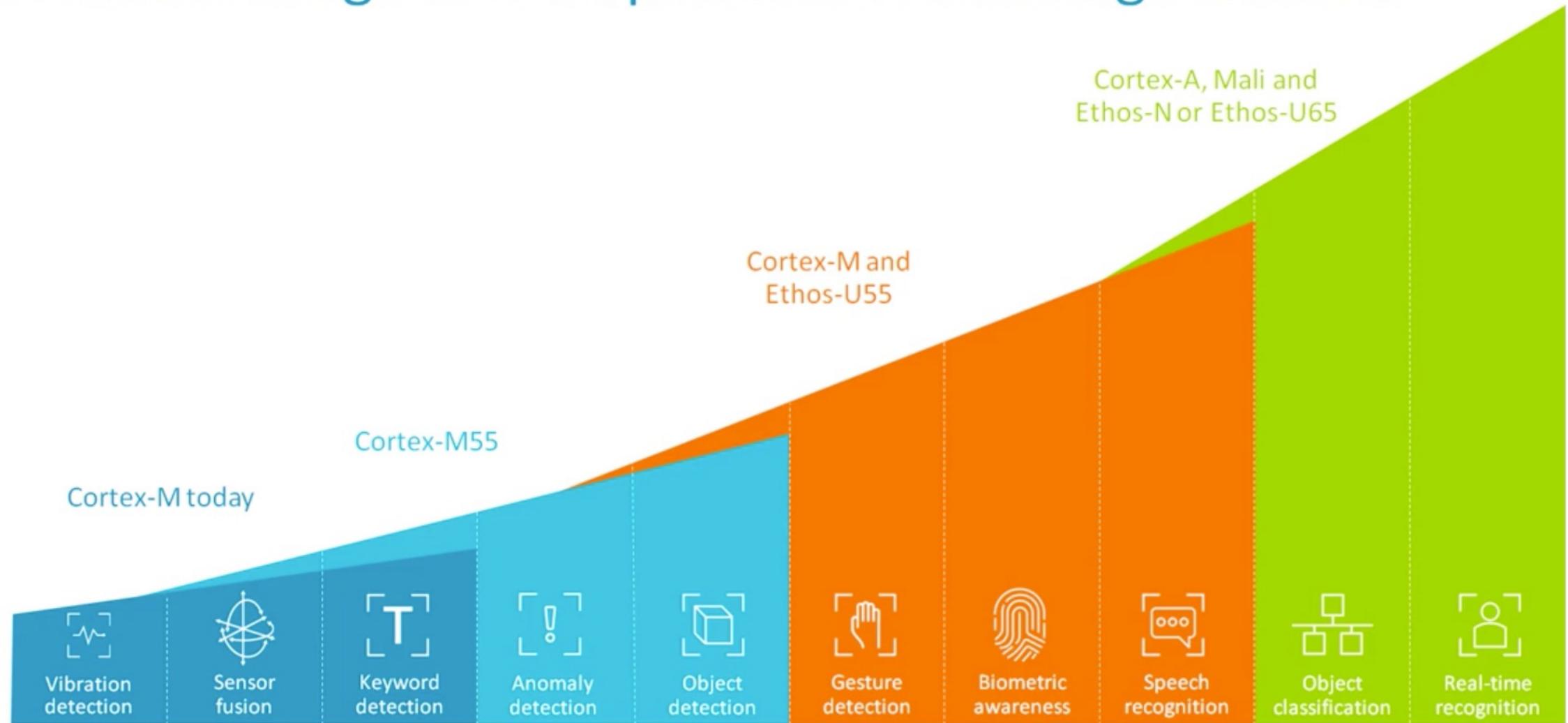




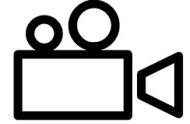
## "Energy-efficient On-device Processing for Next-generation Endpoint ML"

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

# Broadest Range of ML-optimized Processing Solutions



# Summary



Anomaly Detection  
Sensor Classification  
20 KB



Rpi-Pico  
(Cortex-M0+)



Arduino Nano  
(Cortex-M4)



Arduino Pro  
(Cortex-M7)

TinyML

Image  
Classification  
250 KB+



KeyWord Spotting  
Audio Classification  
50 KB

Video  
Classification  
2 MB+

EdgeML

Object  
Detection  
Complex Voice  
Processing  
1 MB+



Raspberry Pi  
(Cortex-A)



Jetson Nano  
(Cortex-A + GPU)

# 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 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!

