Machine Learning on MCUs

uTensor

A Mbed Labs Project

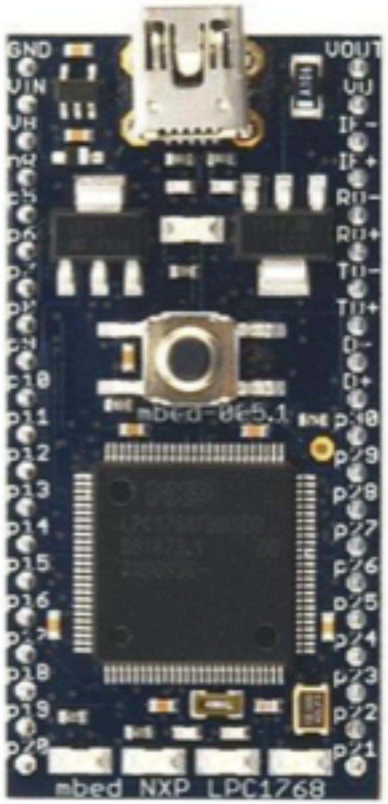
Neil Tan
July 2018

Machine learning

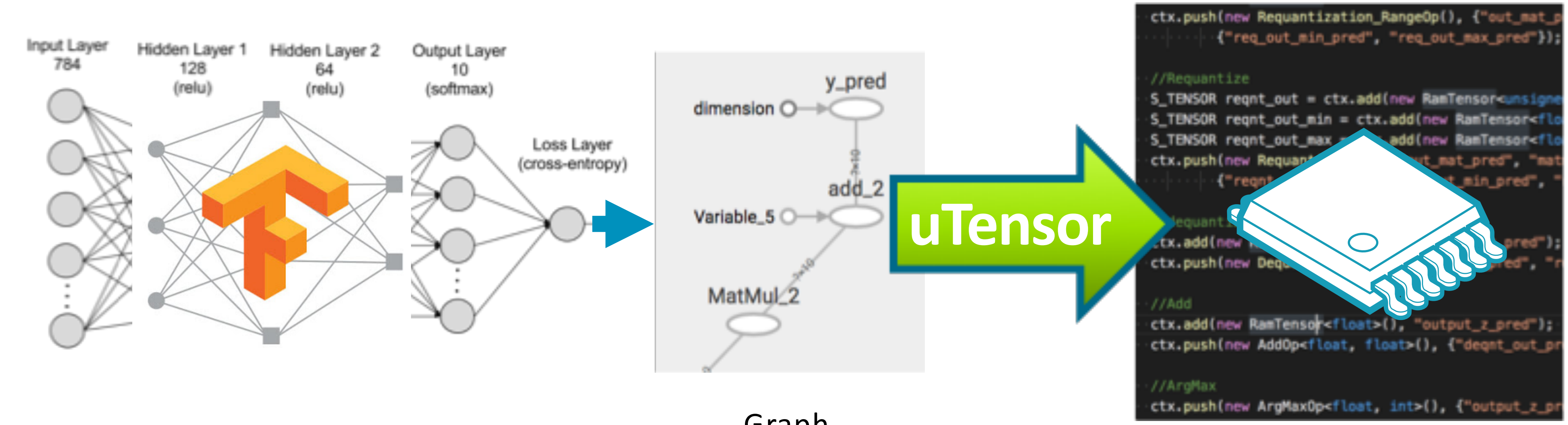
uTensor

Machine Learning for Microcontrollers

- Runs in <256K RAM
- Runs at ~100 MHz
- TensorFlow Compatible
- Inference Only
- Open source, Apache 2.0 license



uTensor



```
ctx.push(new Requantization_RangeOp(), {"out_mat_p
        {"req_out_min_pred", "req_out_max_pred"});

//Requantize
S_TENSOR reqnt_out = ctx.add(new RamTensor<unsigne
S_TENSOR reqnt_out_min = ctx.add(new RamTensor<flo
S_TENSOR reqnt_out_max =      .add(new RamTensor<flo
ctx.push(new Requant                  ut_mat_pred", "mat
        {"reqnt                      t_min_pred", "

equanti
ctx.add(new                               pred");
ctx.push(new Deq                          red", "r

//Add
ctx.add(new RamTensor<float>(), "output_z_pred");
ctx.push(new AddOp<float, float>(), {"deqnt_out_pr

//ArgMax
ctx.push(new ArgMaxOp<float, int>(), {"output_z_pr
```

Graph

C++

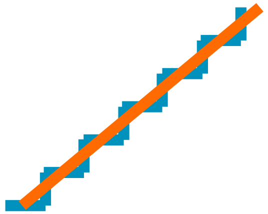
Tensorflow

Mbed

109 YEARS
HOUSE
4.3% ALC BY VOL
12
MB1274 D-01

Technologies

Quantization



Float to 8-bit
75% memory saving
Faster Computation

Code Generation

Copy and Paste
Easy Integration

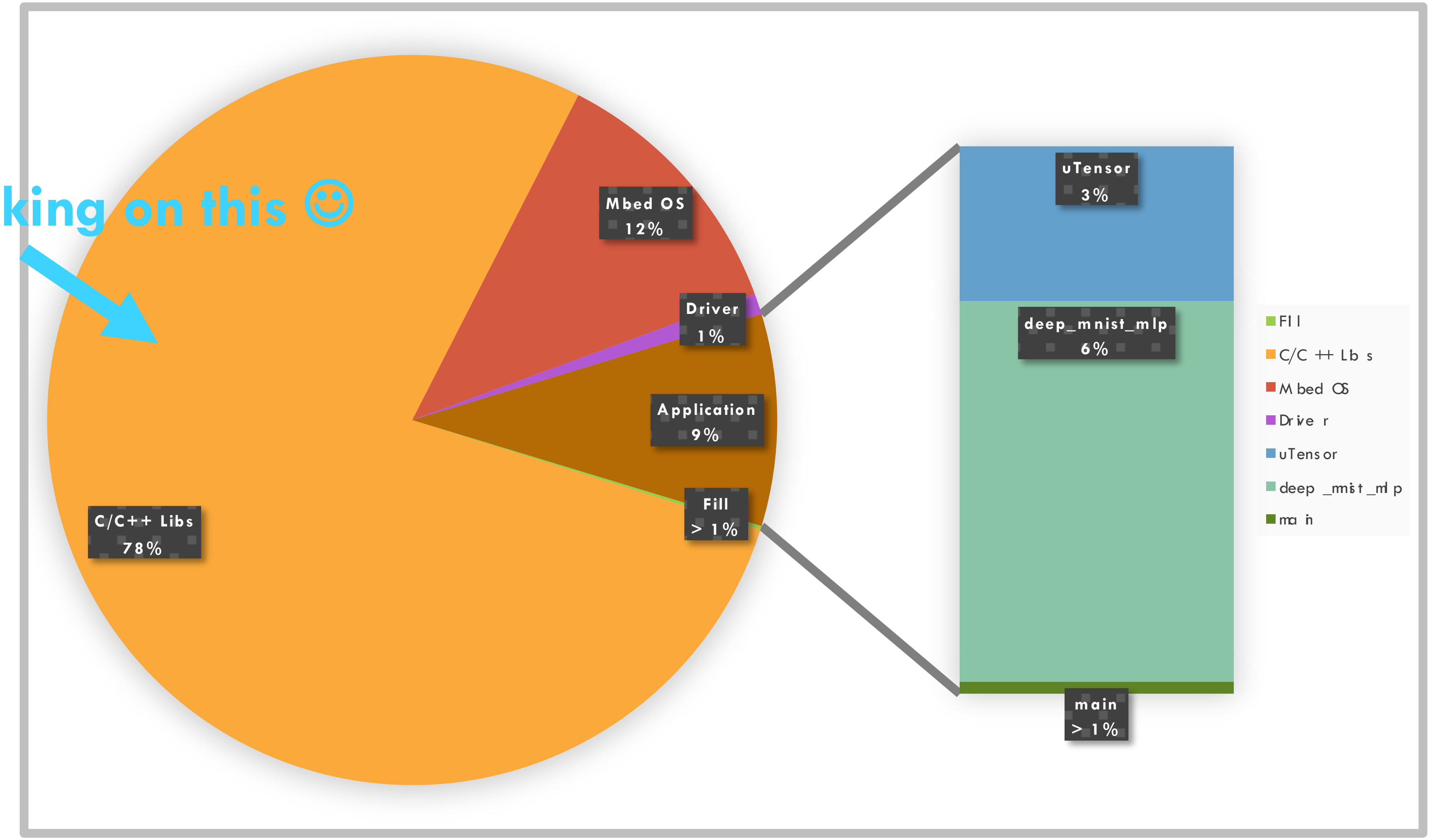
Intermediate Representation

Cross-Framework
Multi-Language

Mbed

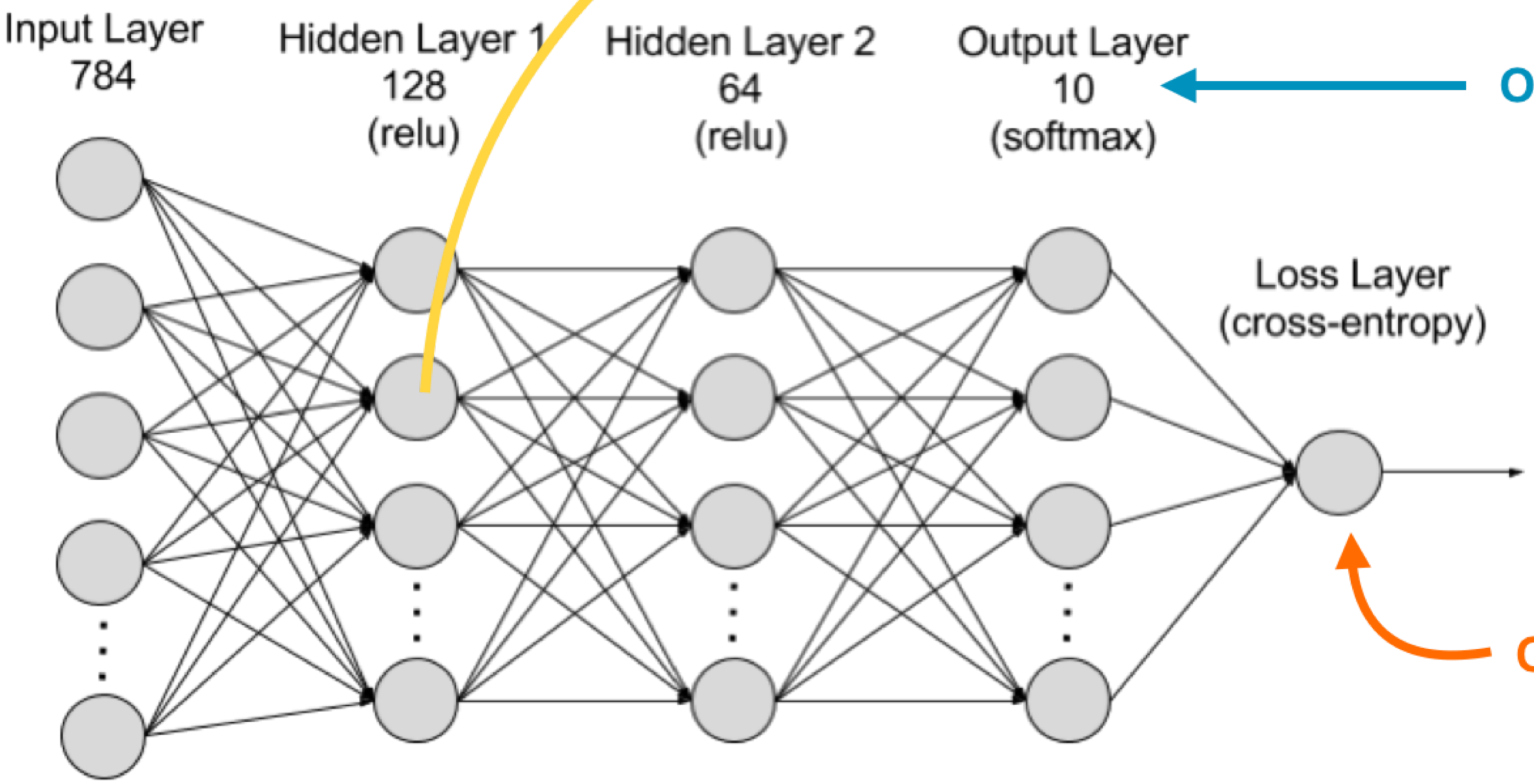
RTOS
CMSIS-NN
Connectivity
Production Ready

Binary



We're working on this ☺

This is a Neural Network



Matrix multiplication (weight), bias, then activation

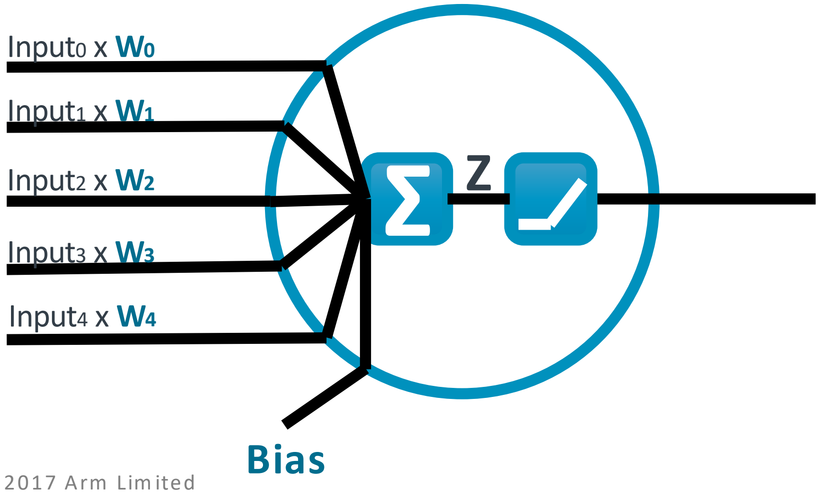
Output Classes

Output

Neuron

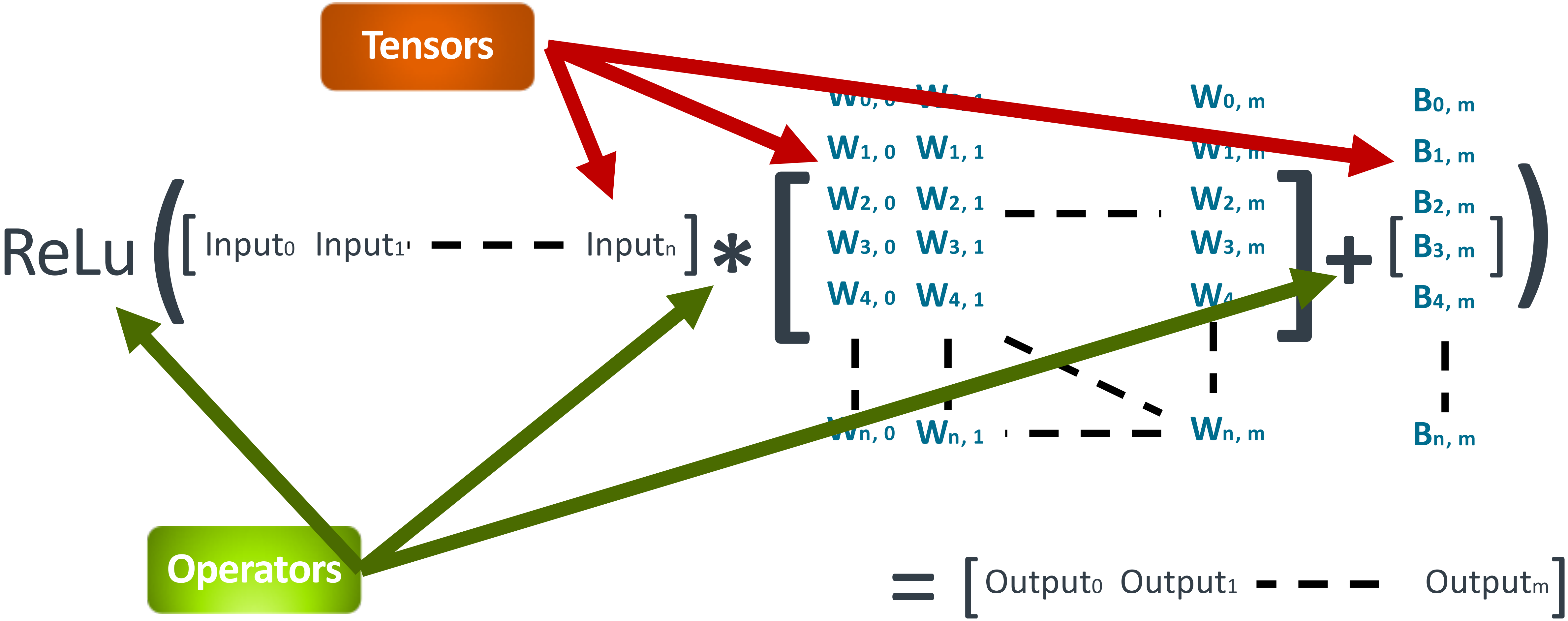
Why Matrix Multiplication

$$\mathrm{ReLu}\left(\begin{bmatrix} \mathrm{Input}_0 & \mathrm{Input}_1 & \cdots & \mathrm{Input}_n \end{bmatrix} * \begin{bmatrix} W_{0,0} & W_{0,1} & & W_{0,m} \\ W_{1,0} & W_{1,1} & & W_{1,m} \\ W_{2,0} & W_{2,1} & \cdots & W_{2,m} \\ W_{3,0} & W_{3,1} & & W_{3,m} \\ W_{4,0} & W_{4,1} & & W_{4,m} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n,0} & W_{n,1} & \cdots & W_{n,m} \end{bmatrix} + \begin{bmatrix} B_{0,m} \\ B_{1,m} \\ B_{2,m} \\ B_{3,m} \\ B_{4,m} \\ \vdots \\ B_{n,m} \end{bmatrix}\right)$$



$$\mathbf{Output} = \begin{bmatrix} \mathrm{Output}_0 & \mathrm{Output}_1 & \cdots & \mathrm{Output}_m \end{bmatrix}$$

Tensors and Operators



Tensors

Operators

$$\mathrm{ReLu}\left(\begin{bmatrix} \mathrm{Input}_0 & \mathrm{Input}_1 & \cdots & \mathrm{Input}_n \end{bmatrix} * \begin{bmatrix} W_{0,0} & W_{0,1} & & W_{0,m} \\ W_{1,0} & W_{1,1} & & W_{1,m} \\ W_{2,0} & W_{2,1} & \cdots & W_{2,m} \\ W_{3,0} & W_{3,1} & & W_{3,m} \\ W_{4,0} & W_{4,1} & & W_{4,m} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n,0} & W_{n,1} & \cdots & W_{n,m} \end{bmatrix} + \begin{bmatrix} B_{0,m} \\ B_{1,m} \\ B_{2,m} \\ B_{3,m} \\ B_{4,m} \\ \vdots \\ B_{n,m} \end{bmatrix}\right)$$

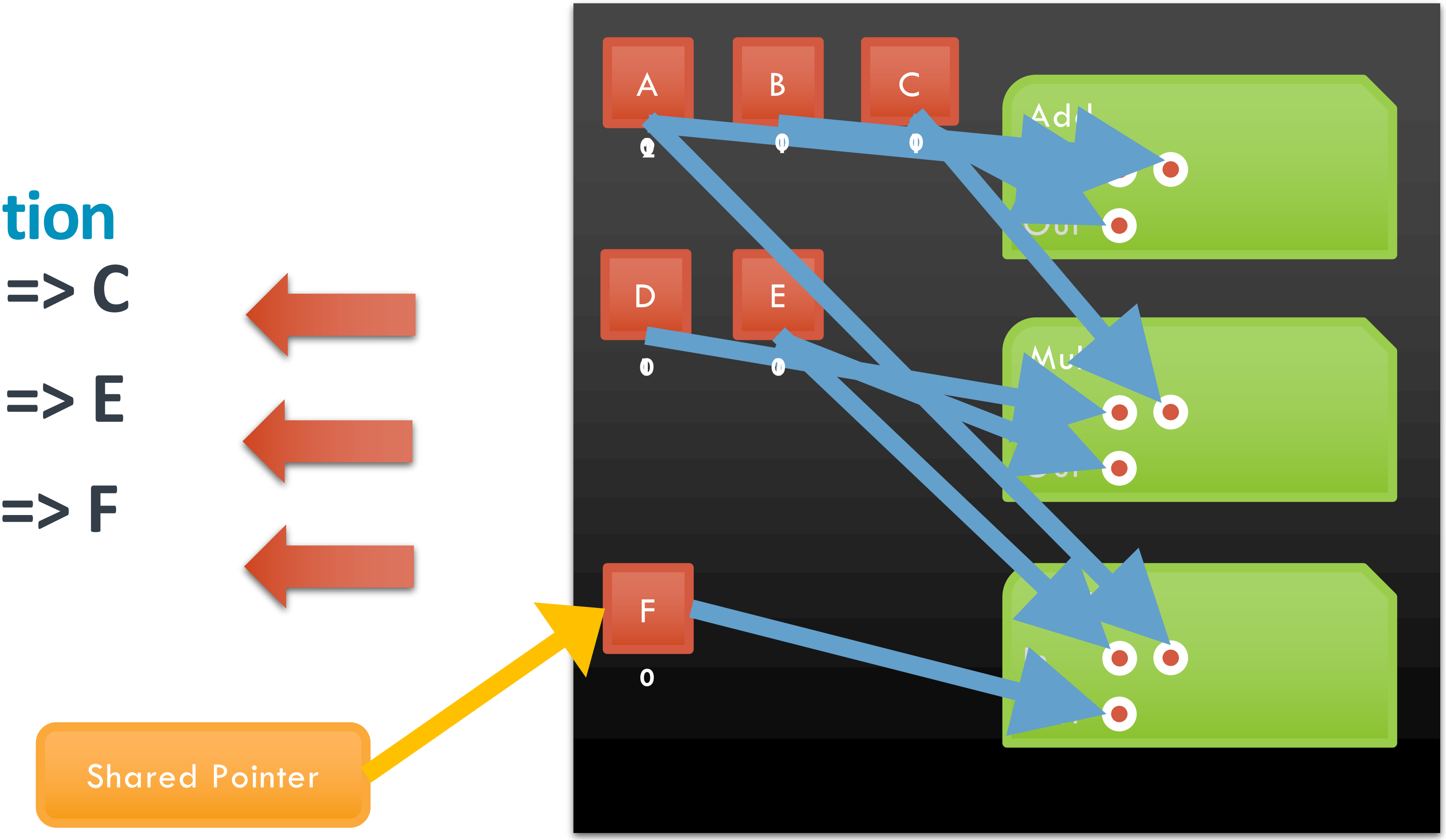
$$= \begin{bmatrix} \mathrm{Output}_0 & \mathrm{Output}_1 & \cdots & \mathrm{Output}_m \end{bmatrix}$$

Execution

A + B => C

D * C => E

A + E => F



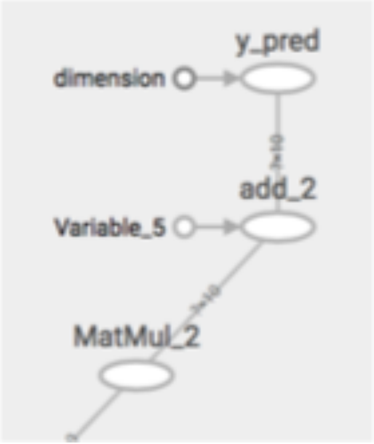
Code generator

A Python Tool

Turns Graph into C++ source

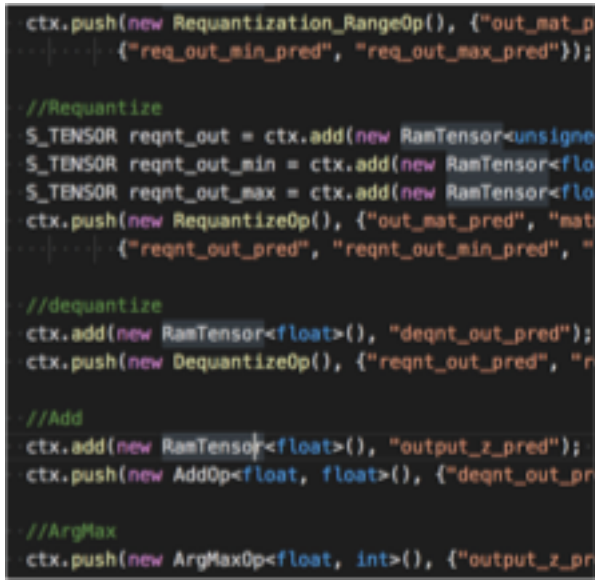
- Graph Traversal
- Optimizer
- Template Engine

TensorFlow



Graph

Mbed

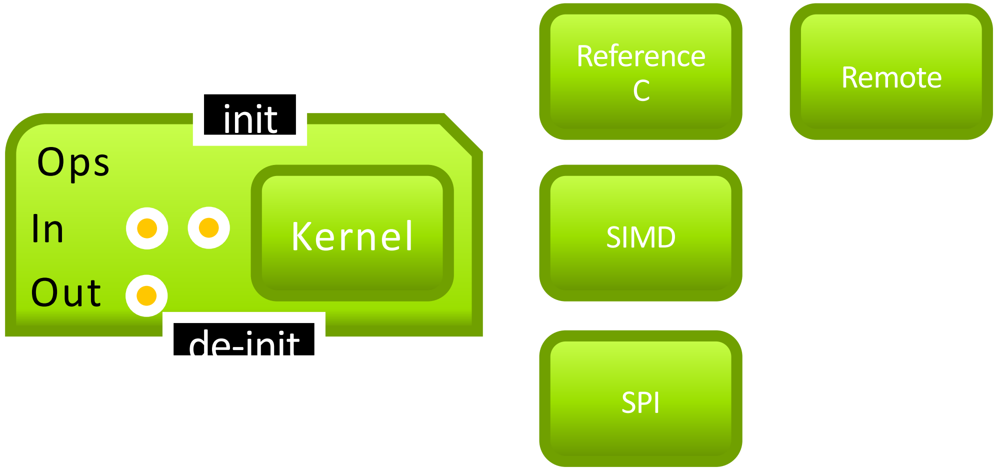


```
ctx.push(new Requantization_RangeOp(), {"out_mat_p
        {"req_out_min_pred", "req_out_max_pred"});

//Requantize
S_TENSOR reqnt_out = ctx.add(new RamTensor<unsigne
S_TENSOR reqnt_out_min = ctx.add(new RamTensor<flo
S_TENSOR reqnt_out_max = ctx.add(new RamTensor<flo
ctx.push(new RequantizeOp(), {"out_mat_pred", "mat
        {"reqnt_out_pred", "reqnt_out_min_pred", "

//dequantize
ctx.add(new RamTensor<float>(), "deqnt_out_pred");
ctx.push(new DequantizeOp(), {"reqnt_out_pred", "r

//Add
ctx.add(new RamTensor<float>(), "output_z_pred");
ctx.push(new AddOp<float, float>(), {"deqnt_out_pr

//ArgMax
ctx.push(new ArgMaxOp<float, int>(), {"output_z_pr
```

C++

Graph

Operator Class



init

Ops

In

Out

Kernel

de-init

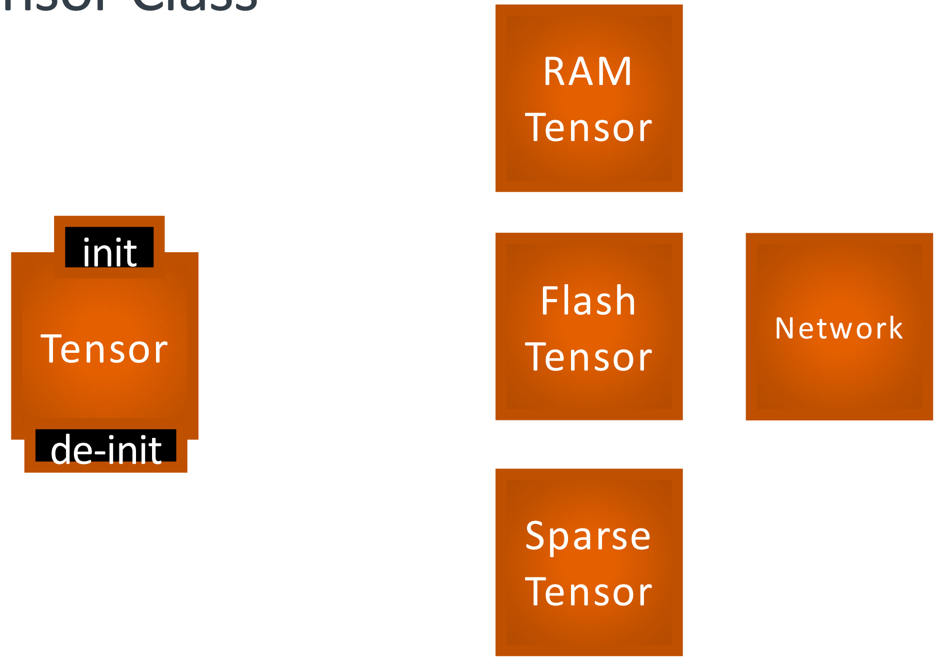
Reference C

Remote

SIMD

SPI

Tensor Class



init

Tensor

de-init

RAM Tensor

Flash Tensor

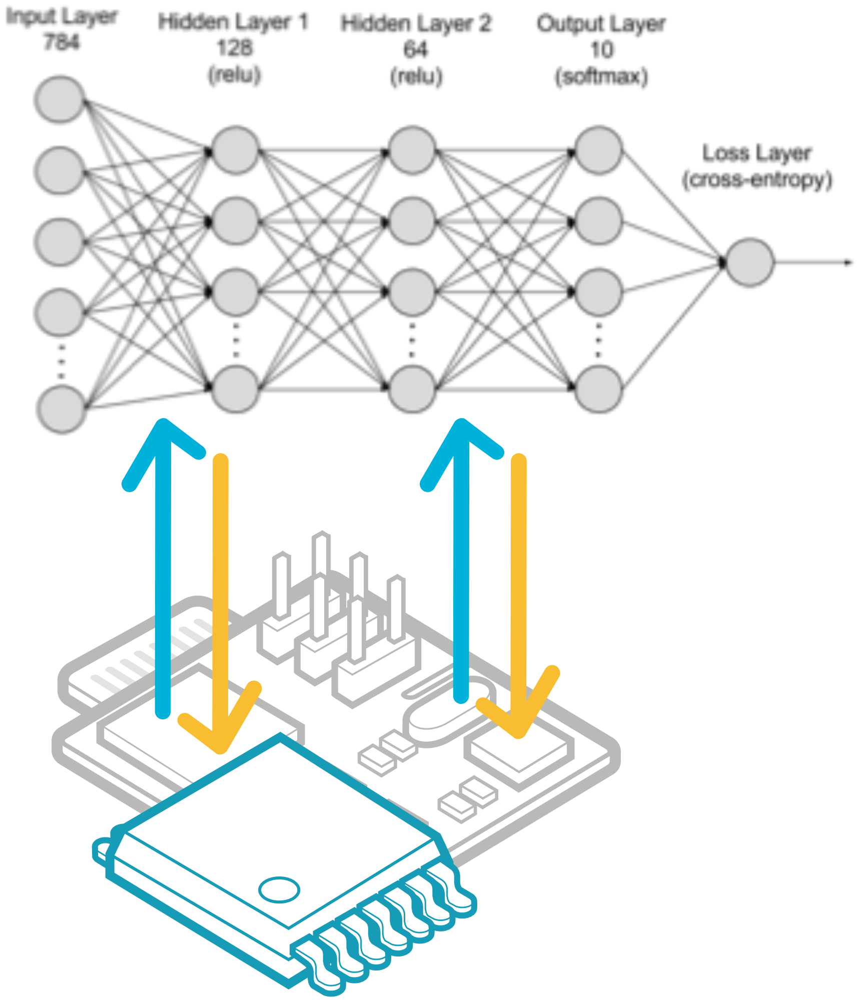
Network

Sparse Tensor

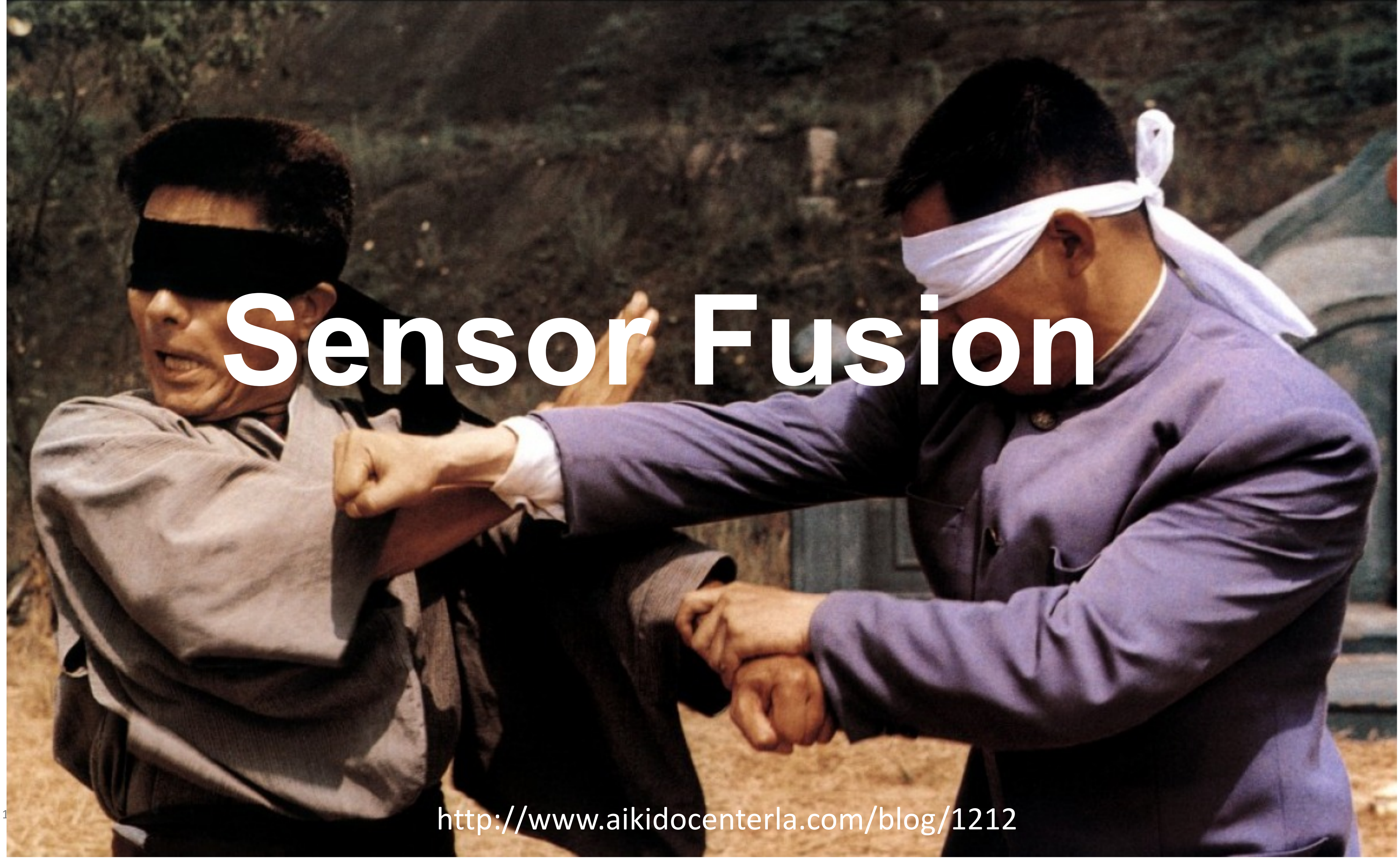
FOTA Graph Update

uTensor’s design allows the graph to be embedded in the binary

- Graph is in Firmware
- Firmware Over The Air
- Weights stored in SD or Firmware

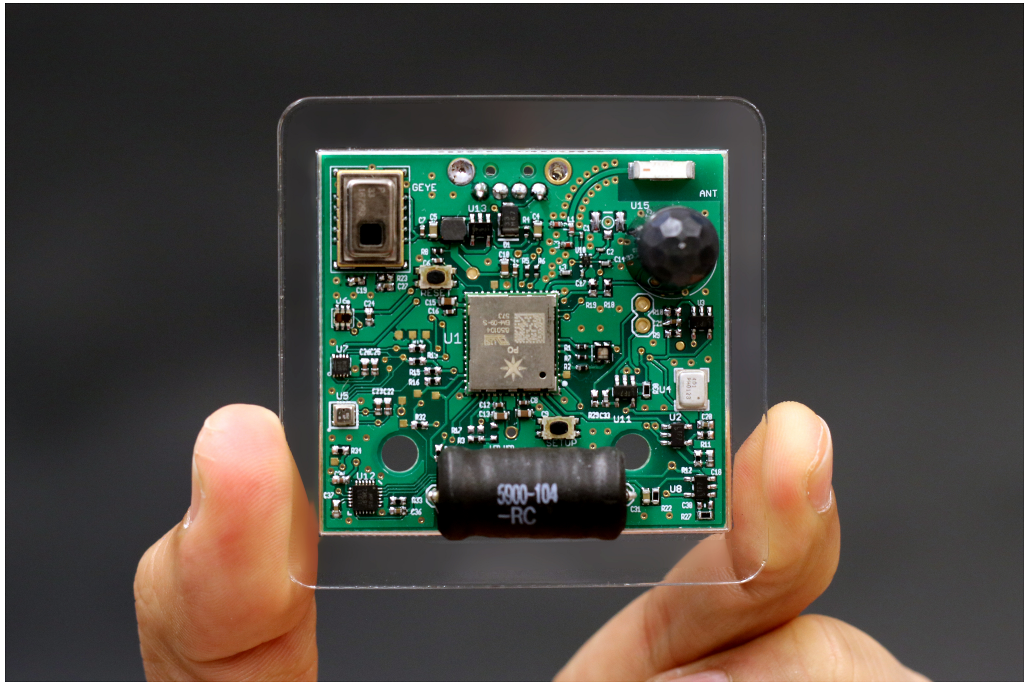


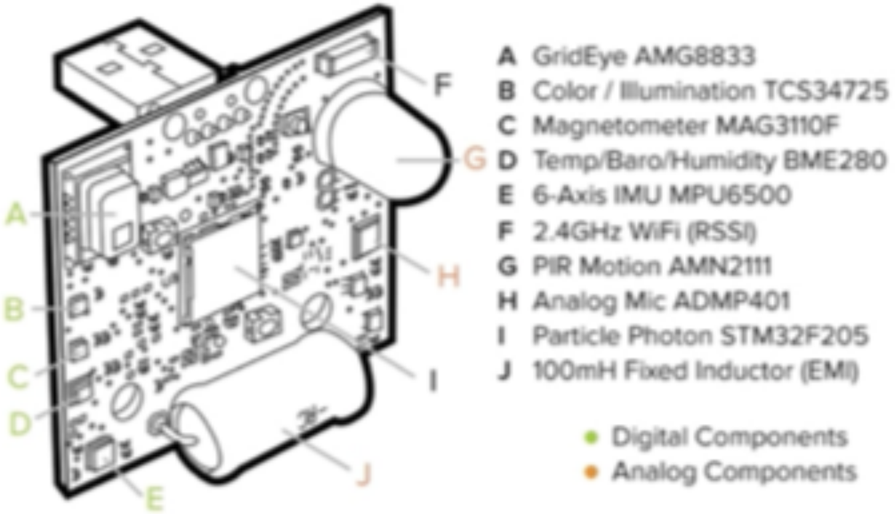
Sensor Fusion



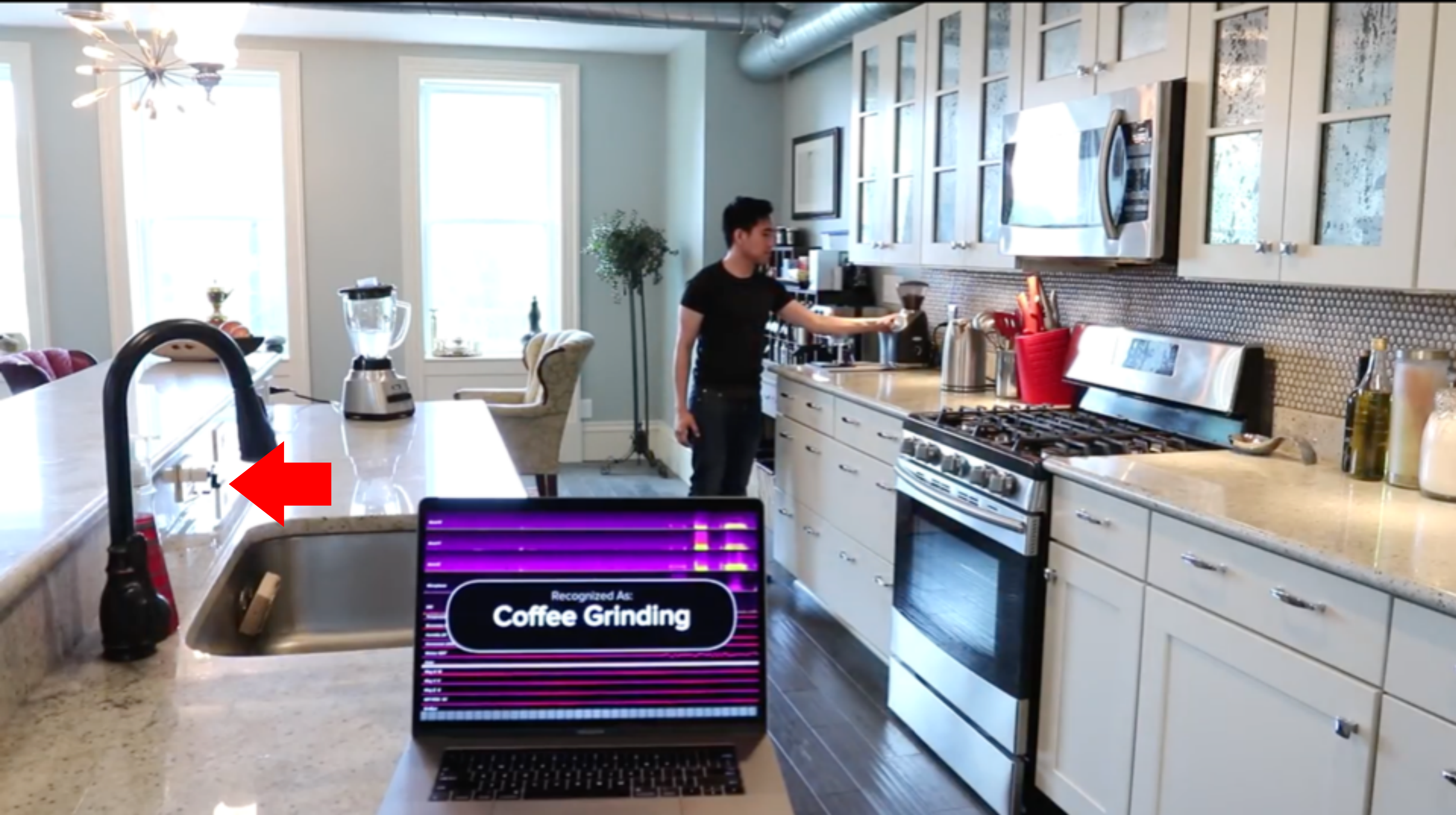
http://www.aikidocenterla.com/blog/1212

Synthetic Sensors

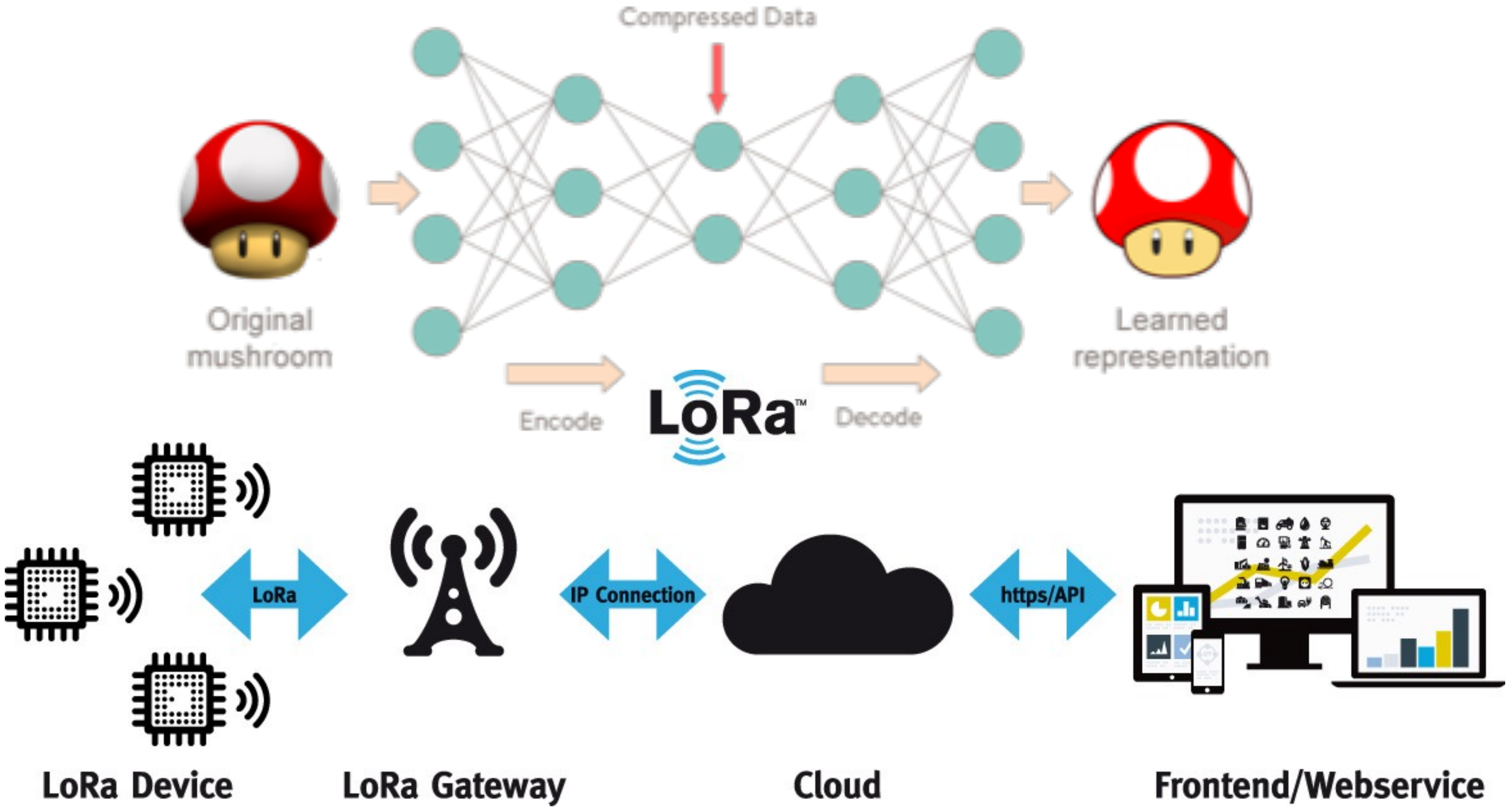




http://www.gierad.com/projects/supersensor/


Recognized As:
Coffee Grinding

Compression + LPWAN

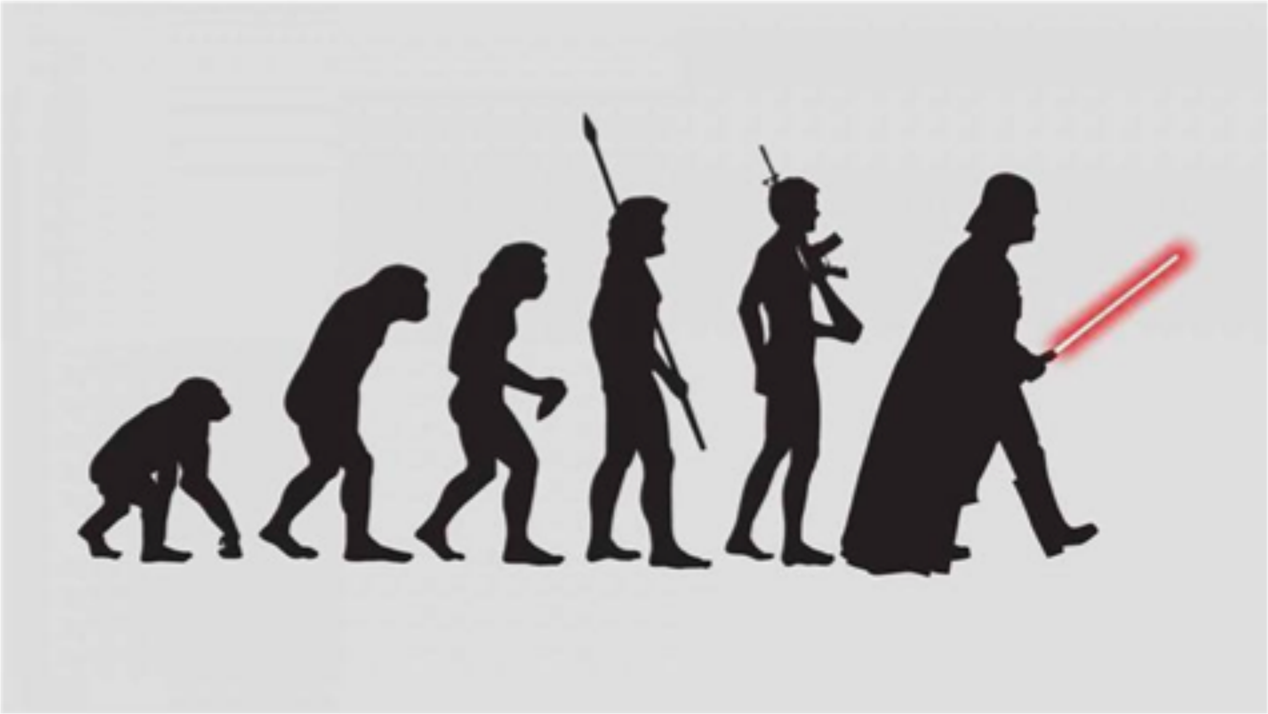


http://curiousily.com/data-science/2017/02/02/what-to-do-when-data-is-missing-part-2.html
https://blog.microtronics.com/lora-and-2g-in-one-module/

utensor.ai

uTensor Timeline

- Test Release
  - Fully Connected, Dropout, Documentation
- Alpha Release
  - Convolution, SD Free
- Beta Release
  - CMSIS-NN



https://futurism.com/common-misunderstandings-of-evolution-part-2/

Thank You!