Embedded Machine Learning

Steven Knudsen, PhD PEng
knud@ualberta.ca

Objectives

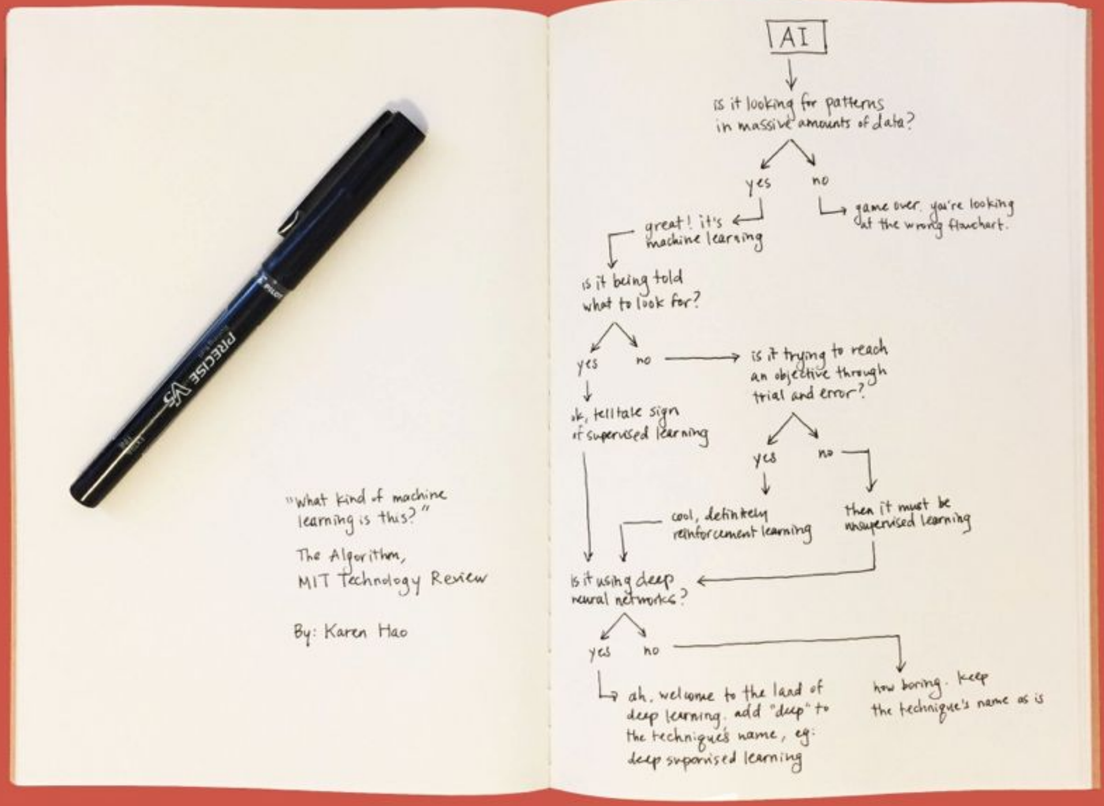
1. Introduction to ML on embedded platforms
2. Setting up learning machine
3. Setting up target platform
4. Training an example
5. Running the example

Sources for the talk

github.com/knud/CompSciAI202002

Assumptions

- Tensorflow 1.15, but discuss 2.x
- Linux for training
  - Ubuntu in general. Examples are Mint 19.3
  - Nvidia GPU
- Artemis platform, can extend to Raspberry Pi, Arduino, and others

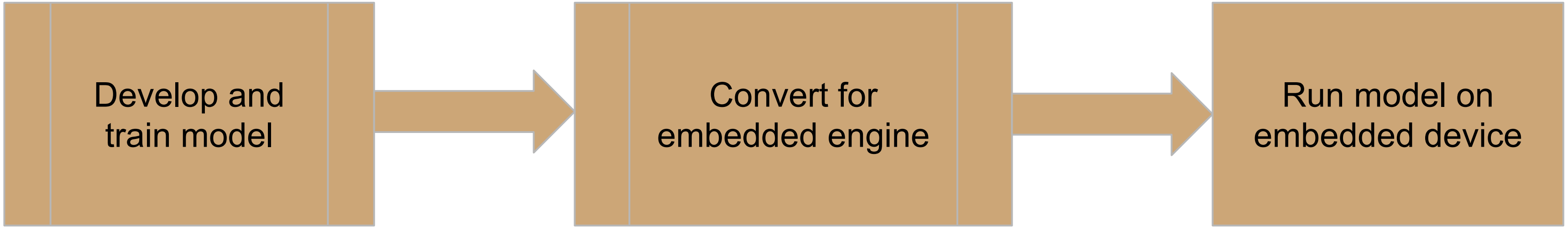

PRECISE V5
"What kind of machine learning is this?"
The Algorithm, MIT Technology Review
By: Karen Hao
AI
is it looking for patterns in massive amounts of data?
yes
no
game over. you're looking at the wrong flowchart.
great! it's machine learning
is it being told what to look for?
yes
no
is it trying to reach an objective through trial and error?
ok, telltale sign of supervised learning
yes
no
cool, definitely reinforcement learning
then it must be unsupervised learning
is it using deep neural networks?
yes
no
ah, welcome to the land of deep learning. add "deep" to the technique's name, eg: deep supervised learning
how boring. keep the technique's name as is

Karen Hao - technologyreview.com

Machine-learning algorithms use statistics to find patterns in massive* amounts of data. And data, here, encompasses a lot of things—numbers, words, images, clicks, what have you. If it can be digitally stored, it can be fed into a machine-learning algorithm.

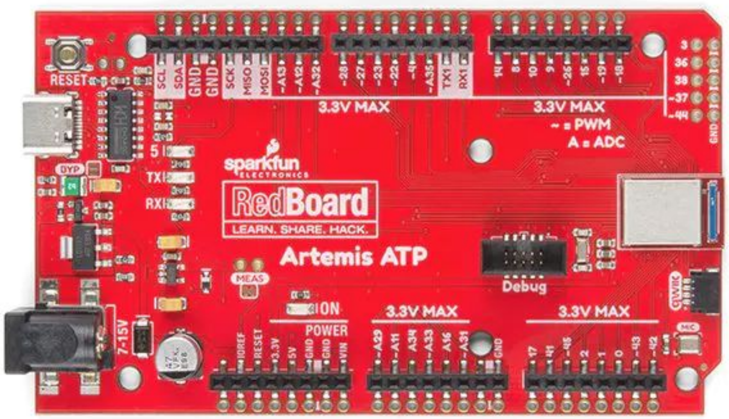
Wake-word Detection

Well-known problem with lots of examples.



Artemis ATP - Apollo3 Blue module

Low Power Machine Learning BLE Cortex-M4F



Ambiq Apollo3 Blue-based module

Low power - 6µA/MHz, RX/TX 3 mA

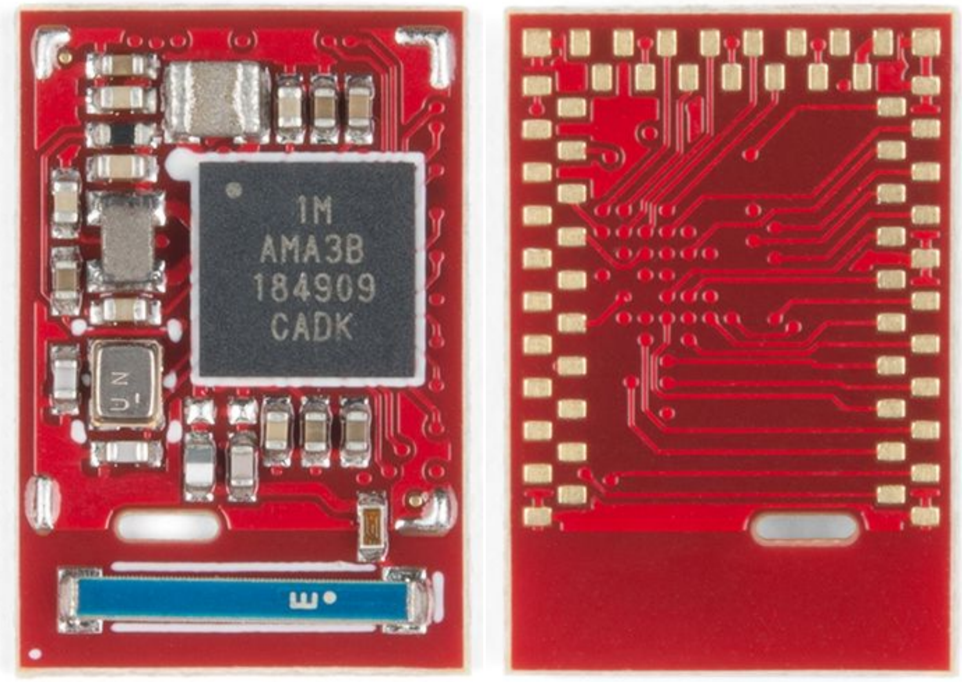
48 MHz, 96 MHz turbo mode

FPU, MPU

32 interrupts

Loads of peripherals, etc.

Loads of GPIO



TensorFlow 1.x Lite for Microcontrollers

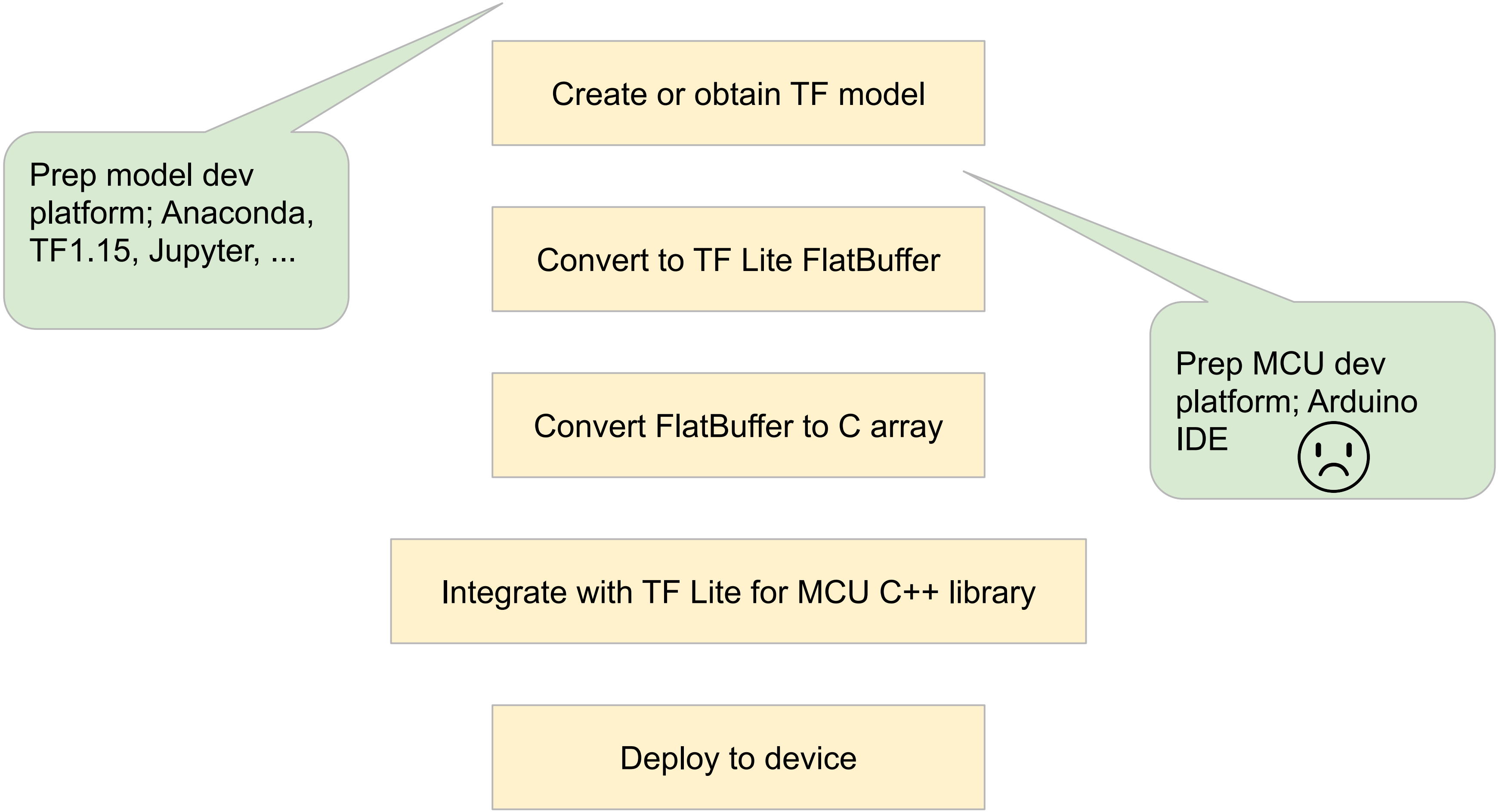
Not TensorFlow 2...

Specially designed for very small footprint devices.

- Core is ~ 16 kB
- With enough operators for speech, ~ 22 kB

Leaves lots of room for model and other functions need for your application

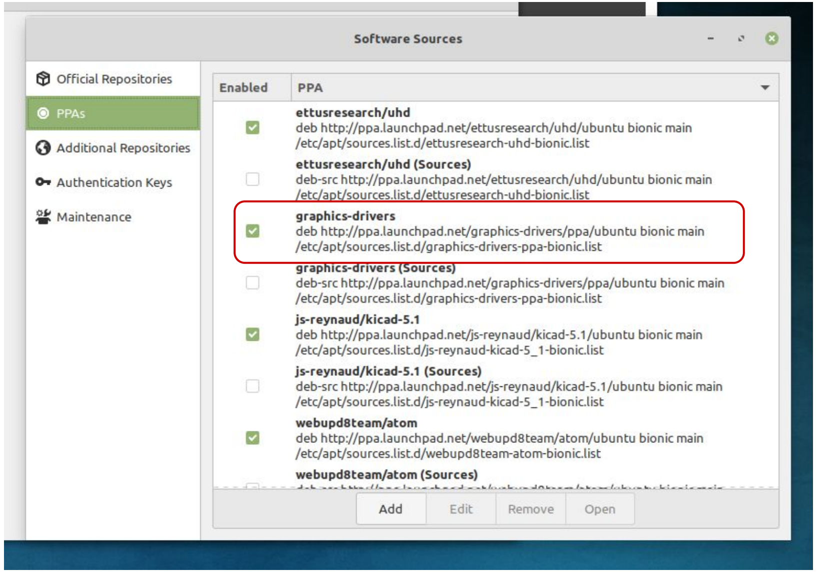
https://www.tensorflow.org/lite/microcontrollers


Prep model dev platform; Anaconda, TF1.15, Jupyter, ...
Create or obtain TF model
Convert to TF Lite FlatBuffer
Prep MCU dev platform; Arduino IDE
Convert FlatBuffer to C array
Integrate with TF Lite for MCU C++ library
Deploy to device

Model Development

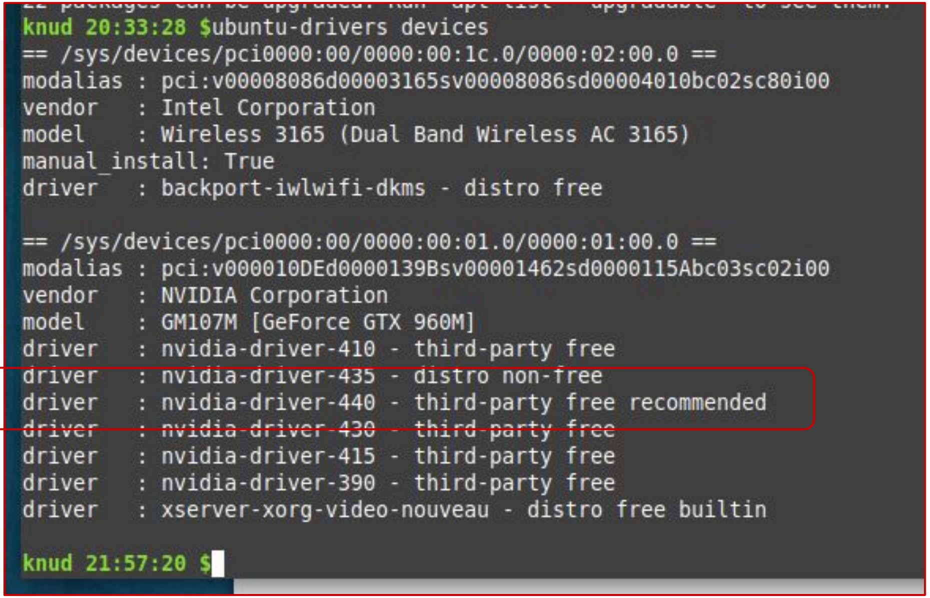
Enable GPU support

```
$sudo add-apt-repository ppa:graphics-drivers/ppa
$sudo apt-get update
```



Enable GPU support cont'd

```
$ubuntu-drivers devices
```




```
$sudo apt install nvidia-driver-440
```

Enable GPU support cont'd

Reboot

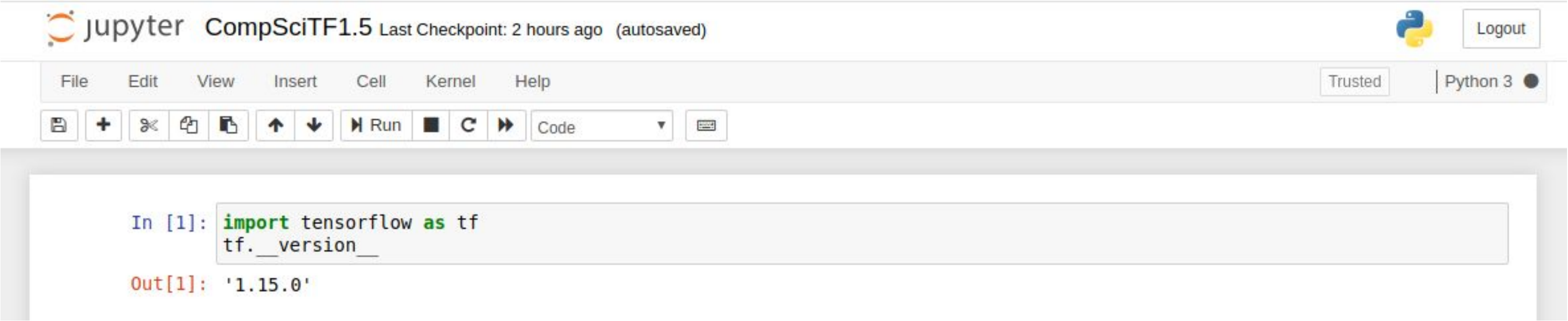
```
$nvidia-smi
```

```
knud 10:01:34 $nvidia-smi
Tue Feb  4 10:01:37 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.48.02    Driver Version: 440.48.02    CUDA Version: 10.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GTX 960M    Off  | 00000000:01:00.0 Off |                  N/A |
| N/A   35C    P8    N/A /  N/A |    211MiB /  2004MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0      3951      G   /usr/lib/xorg/Xorg                           147MiB |
|    0      7664      G   ...quest-channel-token=3152215896715755542    60MiB |
+-----------------------------------------------------------------------------+
knud 10:01:37 $
```

Anaconda Navigator, Jupyter

- For experiments and training
  - https://docs.anaconda.com/anaconda/install/linux/
  - https://docs.anaconda.com/anaconda/user-guide/tasks/tensorflow/
    - conda create -n tf15-gpu tensorflow-gpu=1.15
    - conda activate tf15-gpu
    - conda install anaconda-navigator
  - https://garywoodfine.com/set-up-anaconda-jupyter-notebook-tensorflow-for-deep-learning/



Model and training for microcontroller

- Tensorflow for Microcontrollers
  - https://www.tensorflow.org/lite/microcontrollers
    - Arduino Nano 33 BLE Sense
    - SparkFun Edge
    - STM32F746 Discovery kit
    - Adafruit EdgeBadge
    - Adafruit TensorFlow Lite for Microcontrollers Kit
    - Adafruit Circuit Playground Bluefruit
    - Espressif ESP32-DevKitC
    - Espressif ESP-EYE

  - https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite

Things to keep in mind…

- Microcontrollers have limited resources
  - Code space (FLASH)
  - RAM
  - Computational power
    - Other things may be running… e.g., Bluetooth stack
- Cannot process large data samples
  - For speech, keep the amount of data low by processing short time intervals
  - 1 second snippets typically used

May think of the microcontroller as a preprocessor for some applications.

Wake word training

- Based on the `micro_speech` example
  - https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/micro/examples
- The Jupyter notebook, `train_speech_model.ipynb`, is the starting point, but is set up to do too much if you already have things downloaded
  - Assumes access to root-level folders, so moved them
- Can select one or more words to train. Example shipped is “yes” & “no”
- Changed to train for “stop”

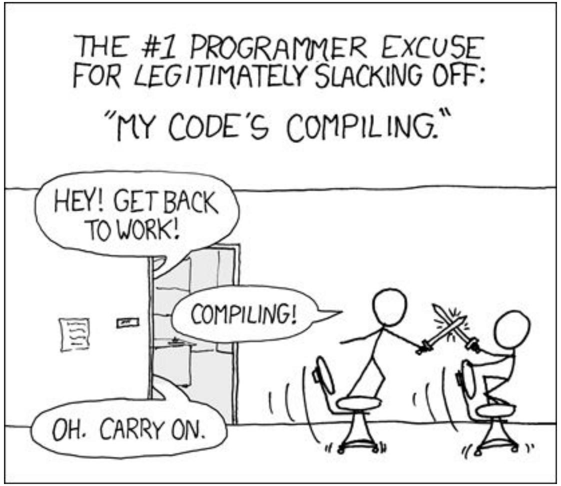
Training for “stop”

- In the Jupyter notebook `CompSciAITF115` provided in github.com/knud/CompSciAI202002, change the words to train near the top from “yes” “no” to “stop”
- Run the notebook through to the end

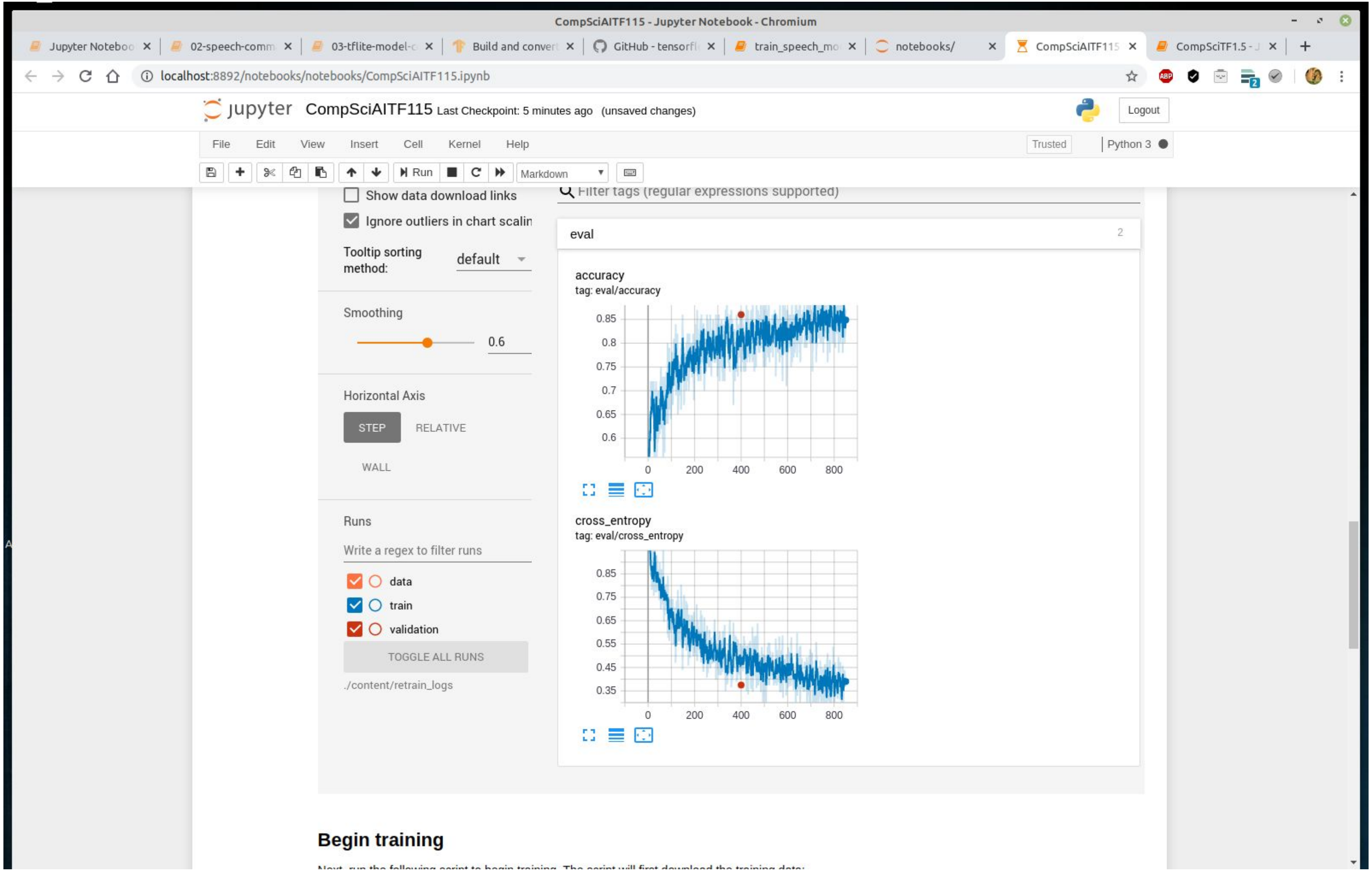
The main result at the end is a TensorFlow Lite model that has been converted into a C array suitable for deployment.

```
tiny_conv.cc
```

Training...


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"My model is training"
HEY! GET BACK TO WORK!
Training!
OH. CARRY ON.


CompSciAITF115 - Jupyter Notebook - Chromium
localhost:8892/notebooks/notebooks/CompSciAITF115.ipynb
jupyter CompSciAITF115 Last Checkpoint: 5 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Markdown
Show data download links
Ignore outliers in chart scalin
Tooltip sorting method:
default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
data
train
validation
TOGGLE ALL RUNS
./content/retrain_logs
Filter tags (regular expressions supported)
eval
2
accuracy
tag: eval/accuracy
cross_entropy
tag: eval/cross_entropy
Begin training

Modify the Artemis source for “stop”

- Open the `micro_speech` sketch in the Arduino IDE
- Copy the C array for the trained model from `tiny_conv.cc` into `micro_features_tiny_conv_micro_features_model_data.cpp`, replacing the data in the array `g_tiny_conv_micro_features_model_data`.
- Just past the array, set the size to that shown at the bottom of `tiny_conv.cc`.

Training for “stop” cont’d

- In `micro_features_micro_model_settings.h`, change `kCategoryCount` to the number of words + 2 (e.g., 3 for just “stop”)
- In `micro_features_micro_model_settings.cpp`, change the words to recognize (e.g., replace “yes” and “no” by “stop”)
- Compile and upload to the Artemis board
- Test!!!

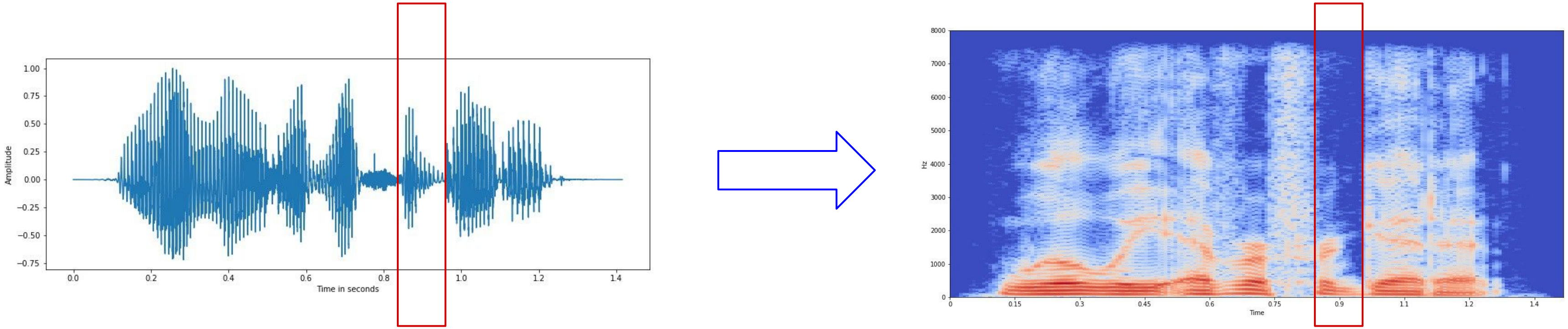
Training for "stop" optional

- In `arduino_command_responder.cpp`, can edit the `RespondToCommand` method to
  - Look for the reported word "stop"
  - Do something other than turn on an LED. E.g., turn off a motor, gather sensor data, etc.

So what is actually happening?

Speech as “image recognition”

Process 1-dimensional speech signal to get a 2-dimensional representation, aka images, that can be used for CNN network training



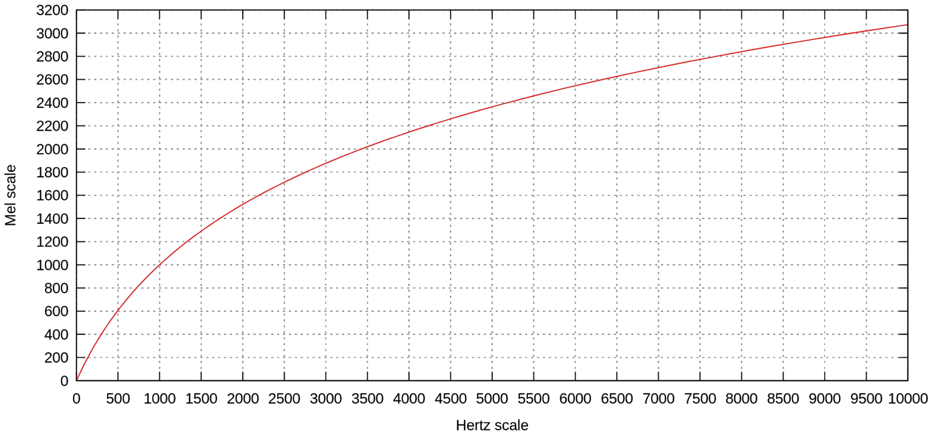
from https://towardsdatascience.com/beginners-guide-to-speech-analysis-4690ca7a7c05

Mel Frequency Cepstral Coefficients (MFCC)

Found that some processing is needed to aid recognition.

- Mel Scale - perceptual scale of pitches judged to be equidistant

$$m = 2595 log_{10}\left(1 + \frac{f}{700}\right)$$

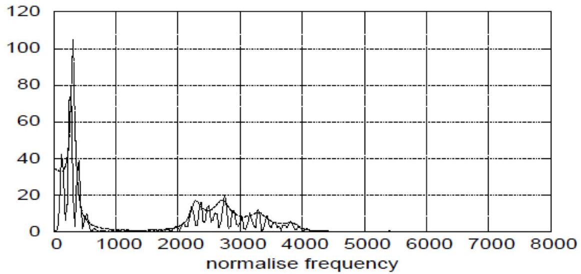


MFCC cont'd

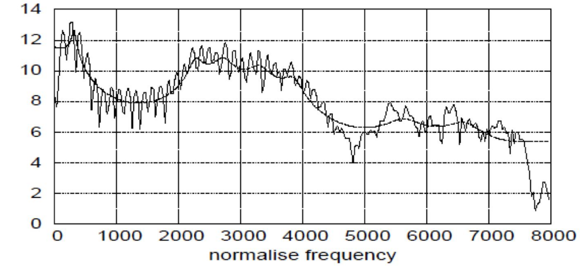
Cepstral comes from cepstrum, which is defined as the inverse DFT of the log magnitude of the DFT of a signal

$$c[n] = IDFT\left\{log_{10}\left|DFT\left\{x[n]\right\}\right|\right\}$$

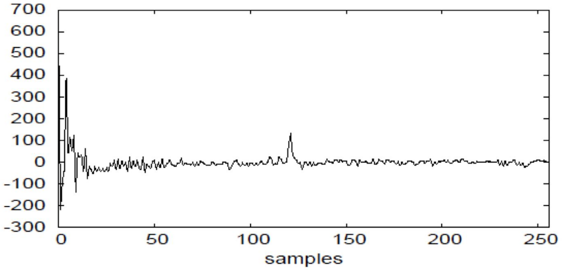
$\mathcal{F}\{x[n]\}$



$\log|\mathcal{F}\{x[n]\}|$

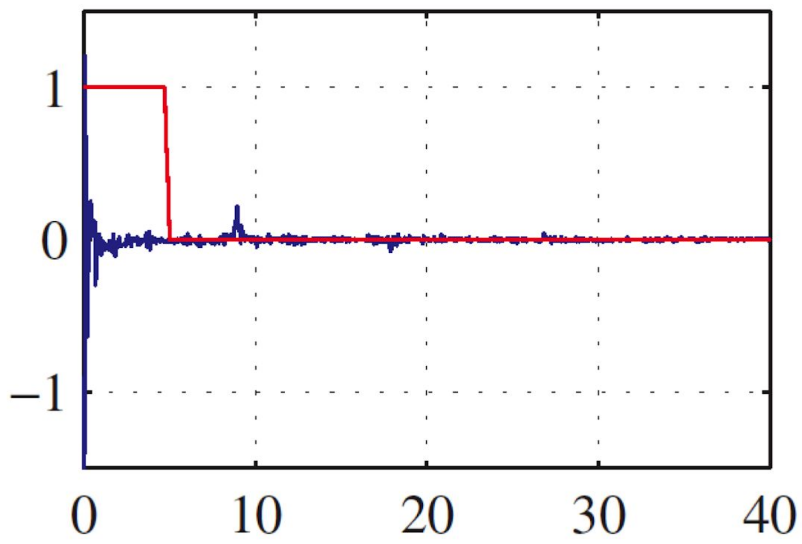


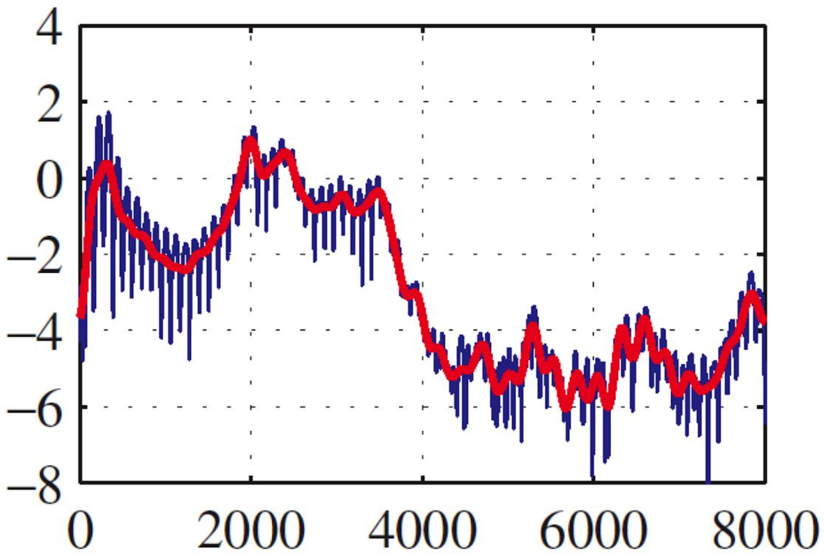
$\mathcal{F}^{-1}\{\log|\mathcal{F}\{x[n]\}|\}$



[Taylor, 2009]

<u>Lift</u>ering in the <u>ceps</u>tral domain





[Rabiner & Schafer, 2007]

MFCC cont'd

Main steps

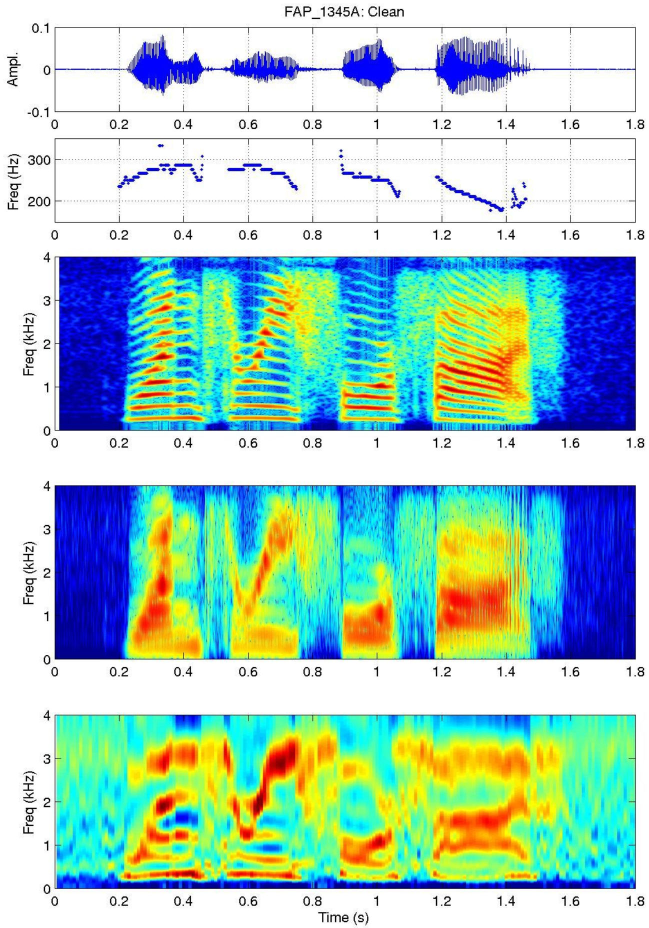
1. Divide signal into "frames"
2. Calculate power spectrum using DFT
3. Map to Mel frequencies (using a filter bank)
4. Compute the log of the spectral coefficients
5. Computer the discrete cosine transform (DCT)

DCT helps decorrelate spectral coefficients and allows pruning of high Mel frequency values that turn out not to help with recognition.

from http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs

MFCC cont’d

MFCC-based waveform and spectrogram for the utterance "one-three-four-five" by a female speaker.



from http://personal.ee.surrey.ac.uk/Personal/P.Jackson/eem.ssr/lab2.html

More details for the interested reader...

The math behind all this is nicely explained here

- https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

Trivia

...with wee prizes...

Trivia

Who created the first artificial neural network and when?

- Warren McCulloch and Walter Pitts
- 1943
- Perceptron
- Modelled using electrical circuits

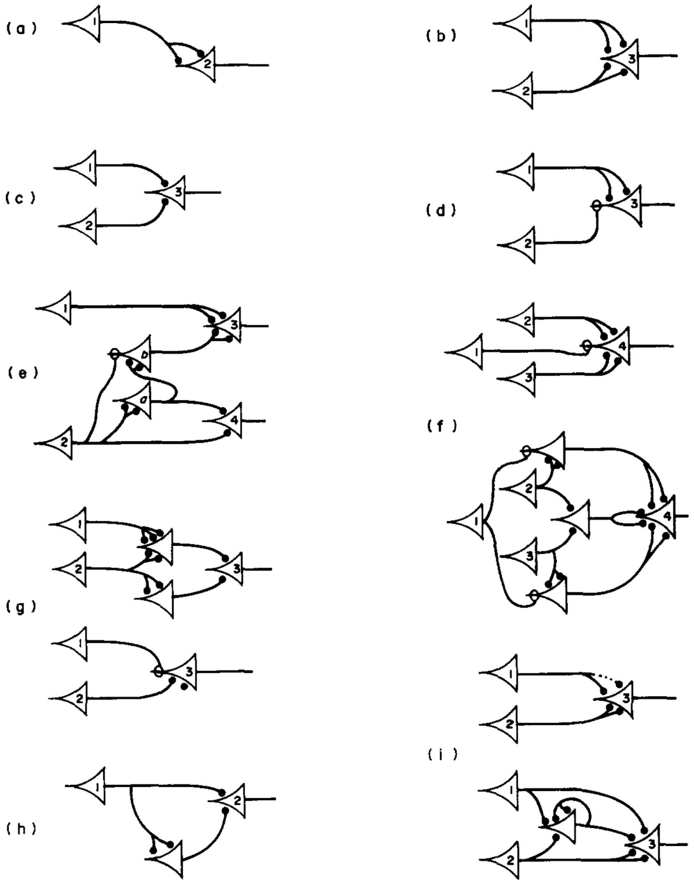


Figure 1. The neuron c_i is always marked with the numeral i upon the body of the cell, and the corresponding action is denoted by "N" with i s subscript, as in the text:

Trivia

Who is the “father” of Java and where was he born?

Trivia

What is an equivalent to the U of A’s main frame computer from the 80s?

Amdahl 470 v/6

Date Introduced : 1975

Dimensions overall: 63” x 70” x 26”

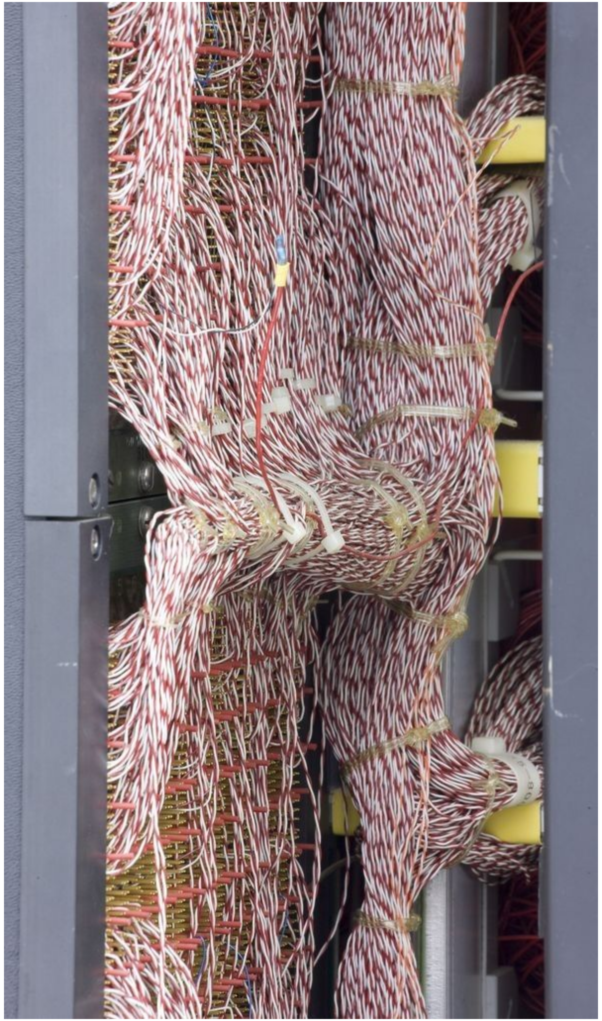
Keywords : Clones; Plug compat; IBM

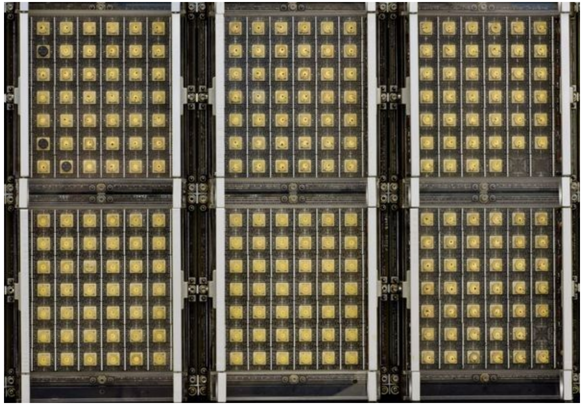
Speed : 3.5 MIPS

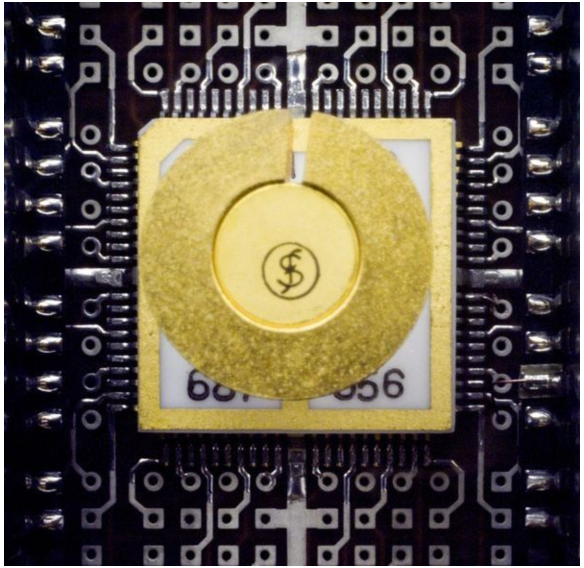
Memory Size : up to 8MB

Memory Width : 32-bit

Cost : $3,750,000 (2020 $17,981,250)







Arduino

Date Introduced : 2010

Dimensions overall: 2.7” x 2.1” x 0.6”

Speed : 8 - 11 MIPS

Memory Size : 32k FLASH 2k SRAM

Cost : ~$20

Cost to make : < $5

Trivia

What was the most dreaded language in the 2019 Stackoverflow survey