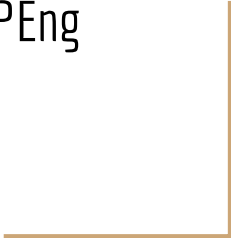


Embedded Machine Learning

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

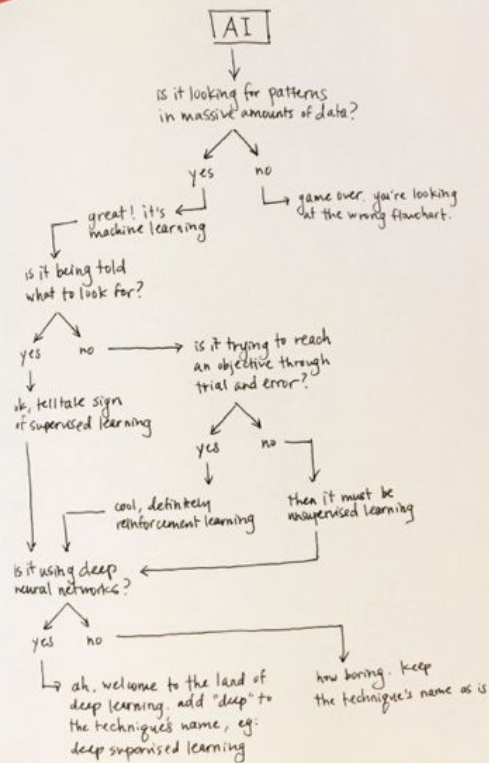
github.com/knud/CompSciAI202002

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



"What kind of machine learning is this?"
The Algorithm,
MIT Technology Review
By: Karen Hao

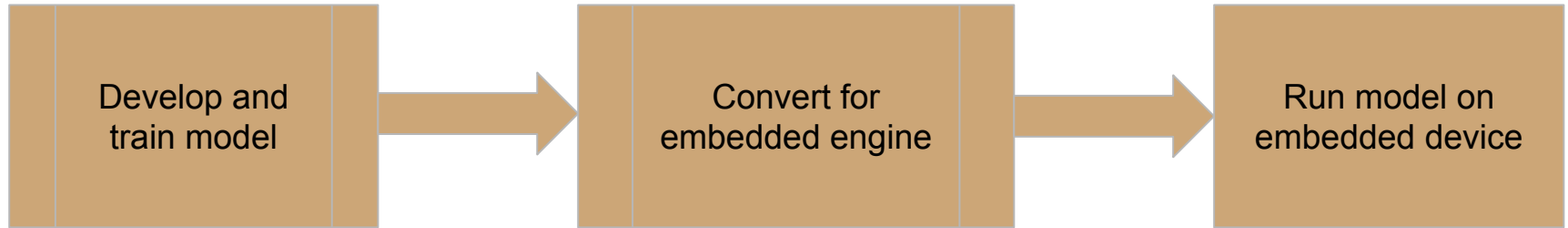


Karen Hao - technologyreview.com

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.

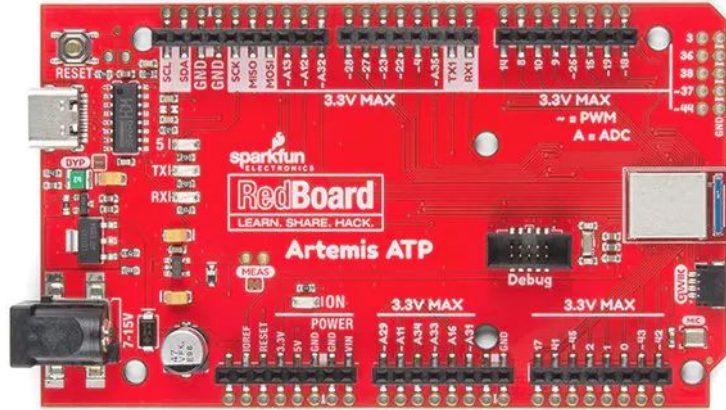
Wake-word Detection

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 - $6\mu\text{A}/\text{MHz}$, RX/TX 3 mA

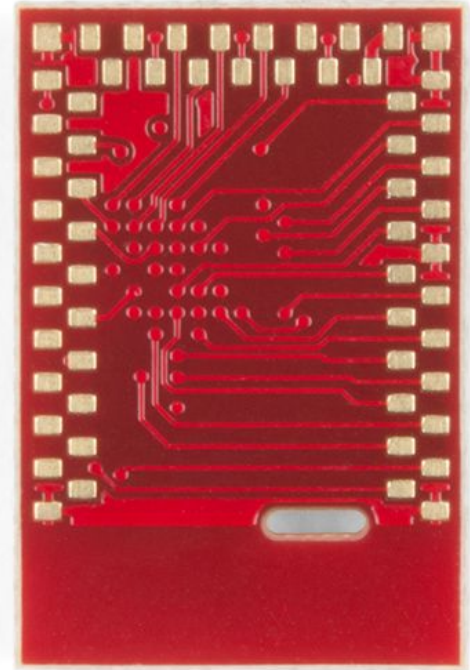
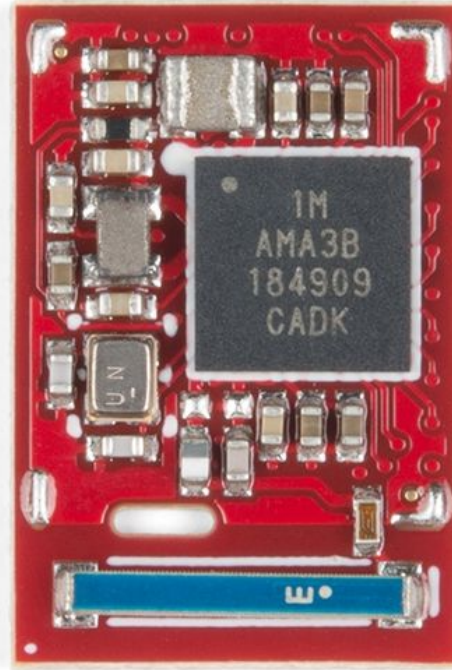
48 MHz, 96 MHz turbo mode

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

Convert FlatBuffer to C array

Integrate with TF Lite for MCU C++ library

Deploy to device

Prep MCU dev
platform; Arduino
IDE



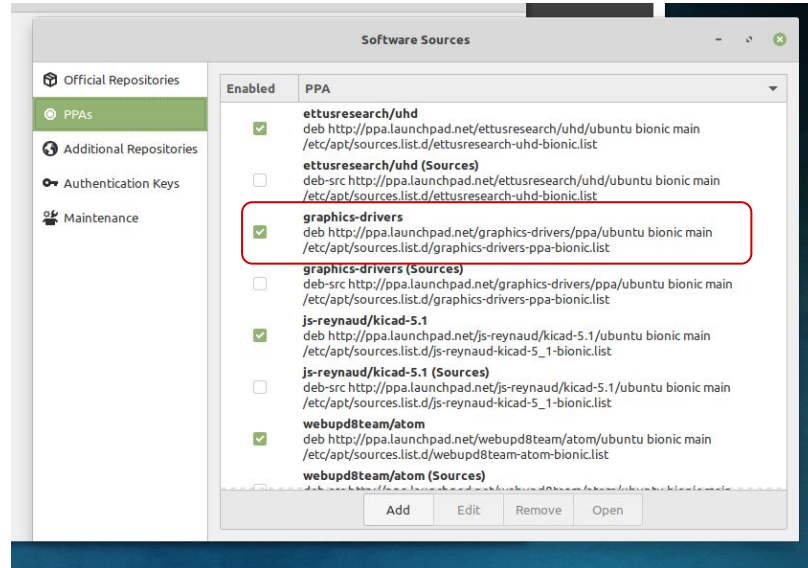


Model Development



Enable GPU support

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



Enable GPU support cont'd

```
$subuntu-drivers devices
```

```
knud 20:33:28 $subuntu-drivers devices
== /sys/devices/pci0000:00/0000:00:1c.0/0000:02:00.0 ==
modalias : pci:v00008086d00003165sv00008086sd00004010bc02sc80i00
vendor    : Intel Corporation
model     : Wireless 3165 (Dual Band Wireless AC 3165)
manual_install: True
driver    : backport-iwlwifi-dkms - distro free

== /sys/devices/pci0000:00/0000:00:01.0/0000:01:00.0 ==
modalias : pci:v000010DEd0000139Bsv00001462sd0000115Aabc03sc02i00
vendor    : NVIDIA Corporation
model     : GM107M [GeForce GTX 960M]
driver    : nvidia-driver-410 - third-party free
driver    : nvidia-driver-435 - distro non-free
driver    : nvidia-driver-440 - third-party free recommended
driver    : nvidia-driver-430 - third-party free
driver    : nvidia-driver-415 - third-party free
driver    : nvidia-driver-390 - third-party free
driver    : xserver-xorg-video-nouveau - distro free builtin

knud 21:57:20 $
```

```
$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

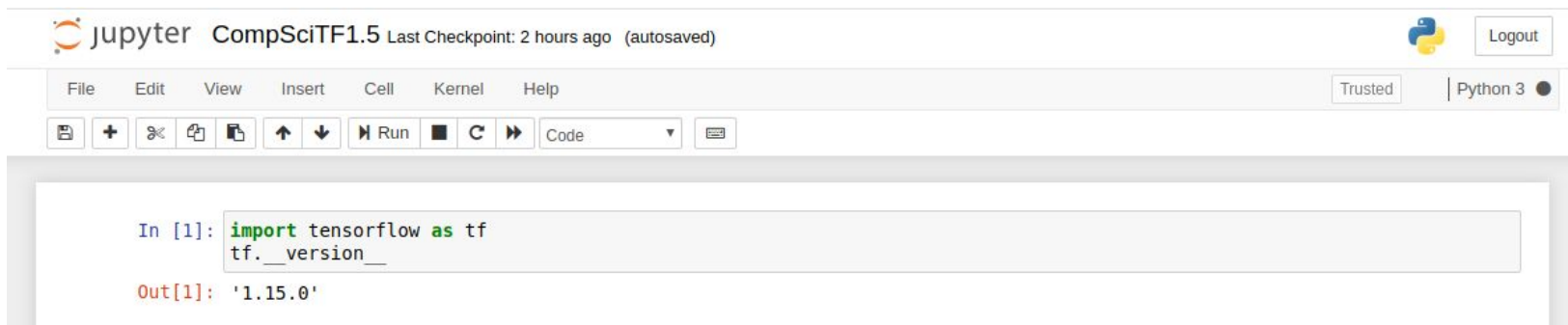
+-----+
| NVIDIA-SMI 440.48.02      Driver Version: 440.48.02      CUDA Version: 10.2      |
+-----+-----+
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+
|    0  GeForce 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”

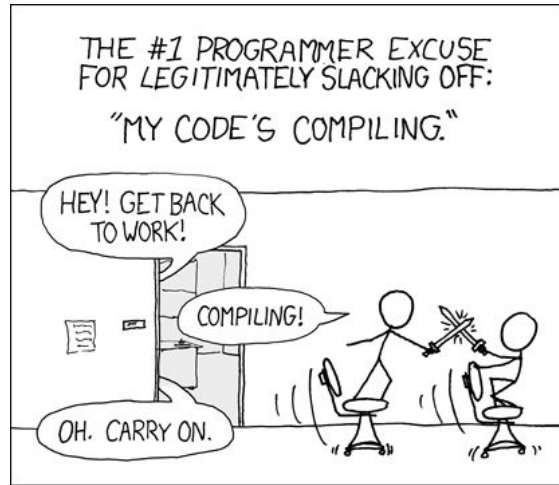
Training for “stop”

- 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 screenshot displays a Jupyter Notebook titled 'CompSciAITF115'. The interface includes a top navigation bar with various tabs, a left sidebar with file explorer and search, and a main content area. The notebook contains two plots:

- accuracy**: A line plot showing accuracy over time. The y-axis ranges from 0.6 to 0.85, and the x-axis ranges from 0 to 800. The accuracy starts at approximately 0.6 and increases to about 0.85 by step 800. A red dot is marked at approximately 400 steps.
- cross_entropy**: A line plot showing cross-entropy over time. The y-axis ranges from 0.35 to 0.85, and the x-axis ranges from 0 to 800. The cross-entropy starts at approximately 0.85 and decreases to about 0.35 by step 800. A red dot is marked at approximately 400 steps.

Below the plots, there is a section for 'Runs' with a table showing the status of different runs. The table has columns for 'Run', 'Status', and 'Action'. The 'Run' column lists 'data', 'train', and 'validation'. The 'Status' column shows a red dot for 'data' and a blue dot for 'train' and 'validation'. The 'Action' column shows a red dot for 'data' and a blue dot for 'train' and 'validation'.

Next, run the following script to begin training. The script will first download the training data:

Prepare the MCU environment

- Main reference

<https://www.hackster.io/javagoza/artemis-atp-wake-word-detection-d95f08>

1. `git clone https://github.com/knud/CompSciAI202002.git`
 - a. Interested in `micro_speech` source
2. As per main reference
 - a. Install Arduino TensorFlow Lite Library
 - b. Install SparkFun boards package and then package for Artemis boards
 - c. Set the board and bootloader
3. Test using `micro_speech.ino`, which is set up for “stop”



```

micro_speech
  return;
}

// Prepare to access the audio spectrograms from a microphone or other source
// that will provide the inputs to the neural network.
// NOLINTNEXTLINE(runtime-global-variables)
static FeatureProvider static_feature_provider(kFeatureElementCount,
                                              model_input->data.uint8);

feature_provider = &static_feature_provider;

static RecognizeCommands static_recognizer(error_reporter);
recognizer = &static_recognizer;

previous_time = 0;
}

// The name of this function is important for Arduino compatibility.
void loop() {
  // Fetch the spectrogram for the current time.
  const int32_t current_time = LatestAudioTimestamp();
  int how_many_new_slices = 0;
  TfLiteStatus feature_status = feature_provider->PopulateFeatureData(
    error_reporter, previous_time, current_time, &how_many_new_slices);
  if (feature_status != kTfLiteOk) {
    error_reporter->Report("Feature generation failed");
    return;
  }
  previous_time = current_time;
  // If no new audio samples have been received since last time, don't bother
  // running the network model.
  if (how_many_new_slices == 0) {
    return;
  }
}

```

```

/home/knud/Arduino/libraries/Arduino_TensorFlowLite/src/tensorflow/lite/experimental/microfrontend/lib/filterbank.c: in function 'filterbanksqrt':
/home/knud/Arduino/libraries/Arduino_TensorFlowLite/src/tensorflow/lite/experimental/microfrontend/lib/filterbank.c:121:25: warning: pointer targets in initialization of 'const int64_t*' {aka 'const long long int*'}
   const int64_t* work = state->work + 1;
                        ~~~~~

```

Sketch uses 139460 bytes (14%) of program storage space. Maximum is 960000 bytes.

Artemis SVL Bootloader

Got SVL Bootloader Version: 3

[#####]Upload failed

Got SVL Bootloader Version: 3

[#####]Upload failed

Got SVL Bootloader Version: 3

[#####]Upload complete


```

/dev/ttyUSB0
|
Send

rj
h FPU Enabled.
j
j PDM DMA Threshold = 16
j Heard stop (219) @5552ms
u
u STOP
j Heard stop (228) @9328ms
g
g STOP
j Heard unknown (207) @10512ms
r
r UNKNOWN
j Heard stop (212) @13824ms
o
o STOP
e Heard stop (248) @15472ms
t
t STOP
h Heard stop (226) @17120ms
e
e STOP
r Heard stop (244) @18784ms
j
j STOP
r Heard stop (226) @20432ms
f
f STOP
h Heard unknown (225) @22560ms
t
t UNKNOWN
j Heard stop (201) @23984ms
e
e STOP
j Heard unknown (202) @26336ms
s
s UNKNOWN
j Heard unknown (230) @28464ms
s
s UNKNOWN
j Heard unknown (231) @30128ms
s
s UNKNOWN
j Heard unknown (207) @32016ms
s
s UNKNOWN
j Heard unknown (235) @33680ms
s
s UNKNOWN
j Heard unknown (235) @33680ms

```

☐ Autoscroll ☐ Show timestamp

NewLine 9600 baud Clear output

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.

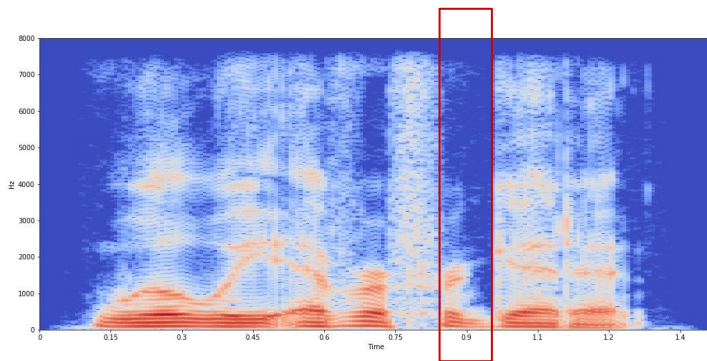
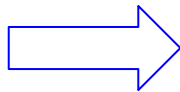
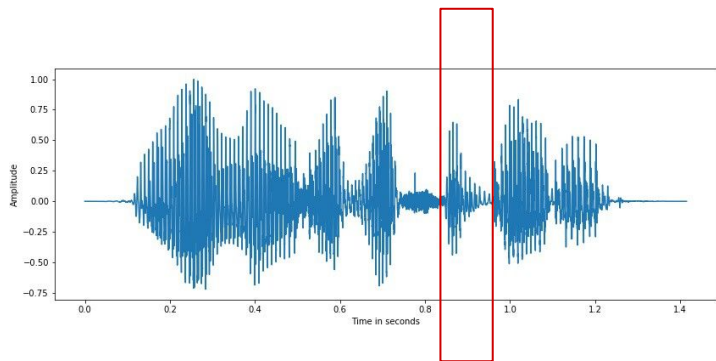


So what is actually happening?



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



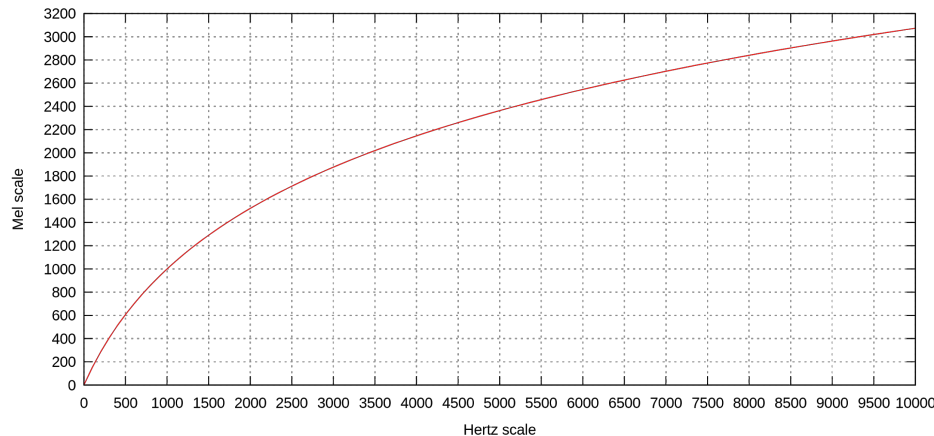
from <https://towardsdatascience.com/beginners-guide-to-speech-analysis-4690ca7a7c05>

Mel Frequency Cepstral Coefficients (MFCC)

Found that some processing is needed to aid recognition.

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

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

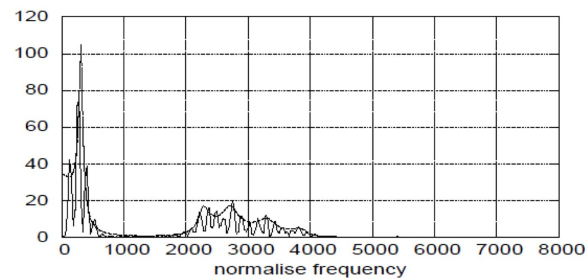


MFCC cont'd

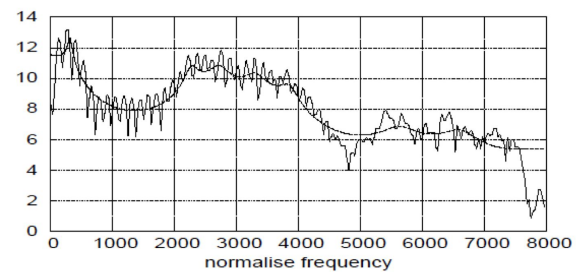
Cepstral comes from cepstrum, which is defined as the inverse DFT of the log magnitude of the DFT of a signal

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

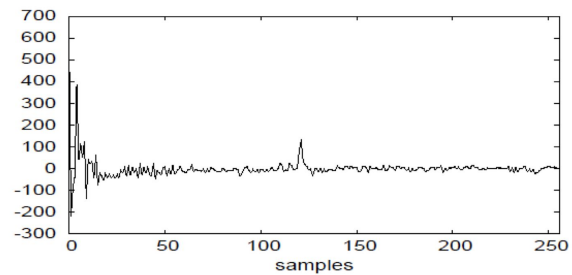
$$\mathcal{F}\{x[n]\}$$



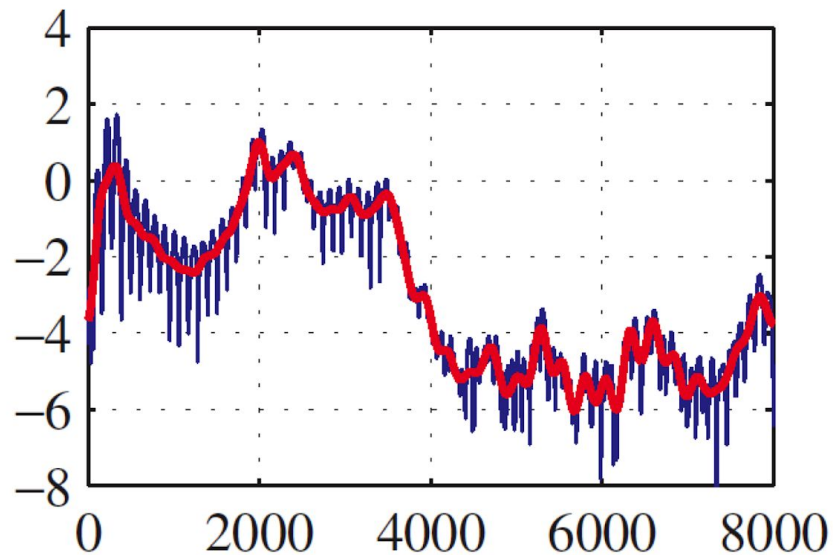
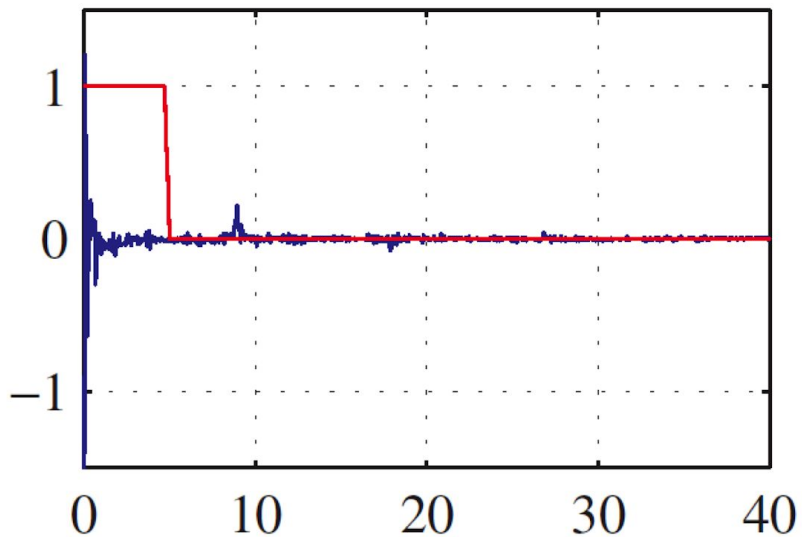
$$\log |\mathcal{F}\{x[n]\}|$$



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



Liftering in the cepstral domain



[Rabiner & Schafer, 2007]

MFCC cont'd

Main steps

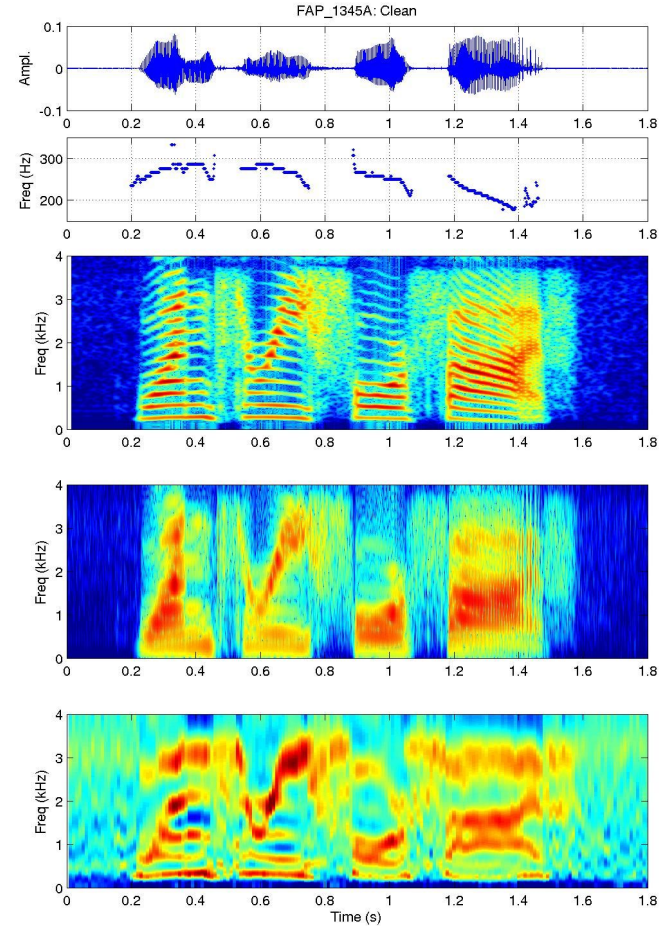
1. Divide signal into “frames”
2. Calculate power spectrum using DFT
3. Map to Mel frequencies (using a filter bank)
4. Compute the log of the spectral coefficients
5. Computer the discrete cosine transform (DCT)

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

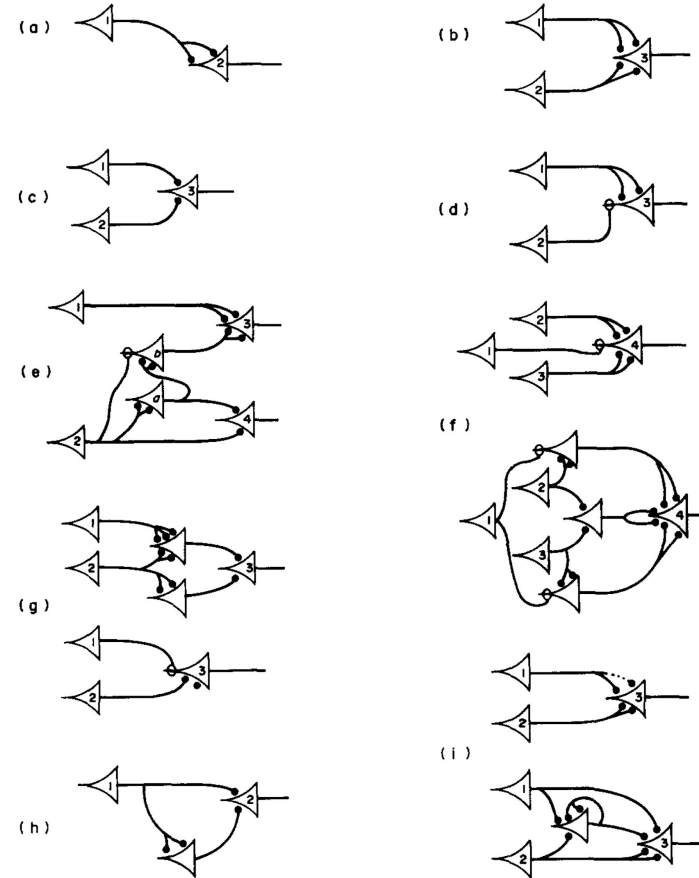


Figure 1. The neuron c_i is always marked with the numeral i upon the body of the cell, and the corresponding action is denoted by “ N ” with i ’s subscript, as in the text:

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"

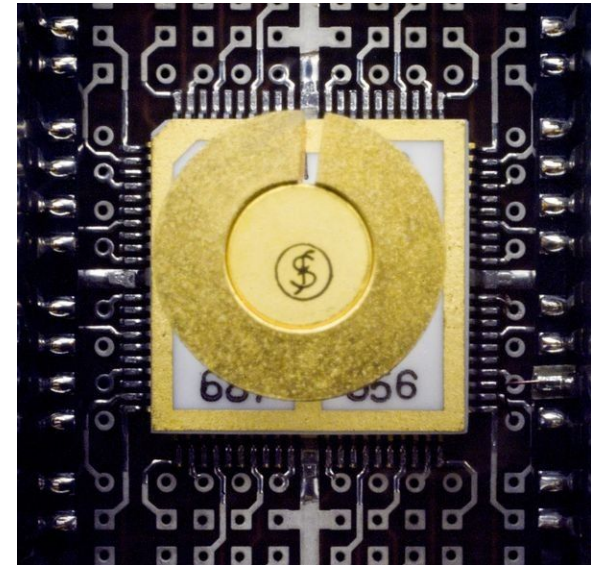
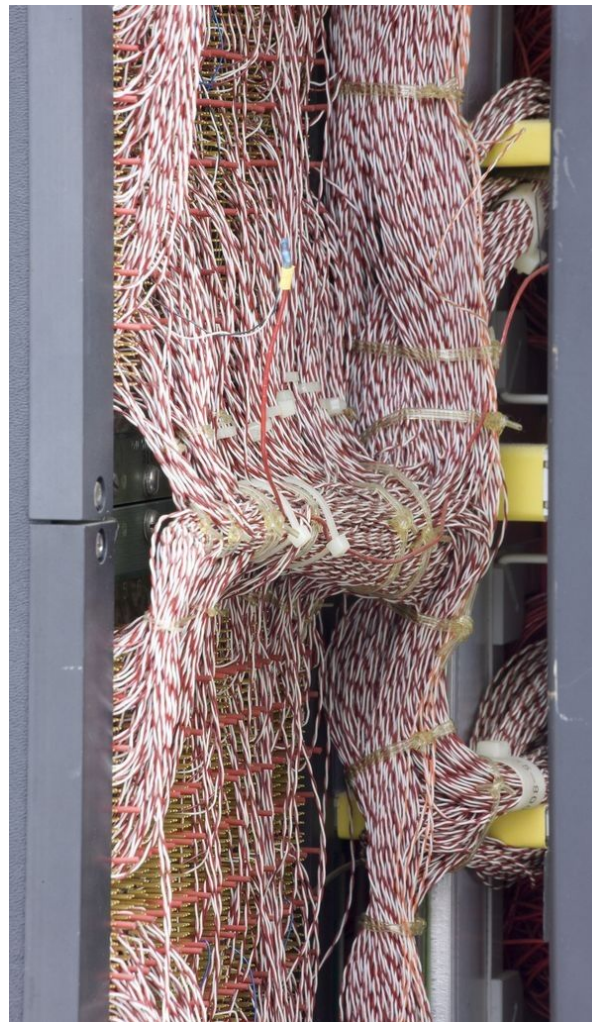
Keywords : Clones; Plug compat; IBM

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