arm AI

AI Virtual Tech Talks Series

fluent.ai™

Speech recognition on Arm Cortex-M

Vikrant Singh Tomar and Sam Myer

July 2020

arm AI

AI Virtual Tech Talks Series

Date	Title	Host
Today	Speech recognition on Arm Cortex-M	Fluent.ai
August, 11	Getting started with Arm Cortex-M software development and Arm Development Studio	Arm
August, 25	Efficient ML across Arm from Cortex-M to Web Assembly	Edge Impulse
September 8, 2020	Running Accelerated ML Applications on Mobile and Embedded Devices using Arm NN	Arm
September 22, 2020	How To Reduce AI Bias with Synthetic Data for Edge Applications	Dori AI

Visit: developer.arm.com/solutions/machine-learning-on-arm/ai-virtual-tech-talks

arm DEV SUMMIT

Join Us at Arm DevSummit

devsummit.arm.com/teamarm

Speakers

Vikrant Singh Tomar

Sam Myer

Overview

- About Fluent.ai
- Fluent.ai µCore
- Demos

About Fluent.ai

- Founded in 2015 after over 7 years of ground-breaking machine learning/AI research by international thought-leaders
- Research partnerships with many leading research labs and institutions
- Strong and experienced team of leading scientists, engineers, sales staff and managers/executives (~25)
- Working with customers in North America, Europe and Asia (robust, multilingual and off-line).

Strong Institutional Backers

Desjardins Capital

bdc

generation ventures

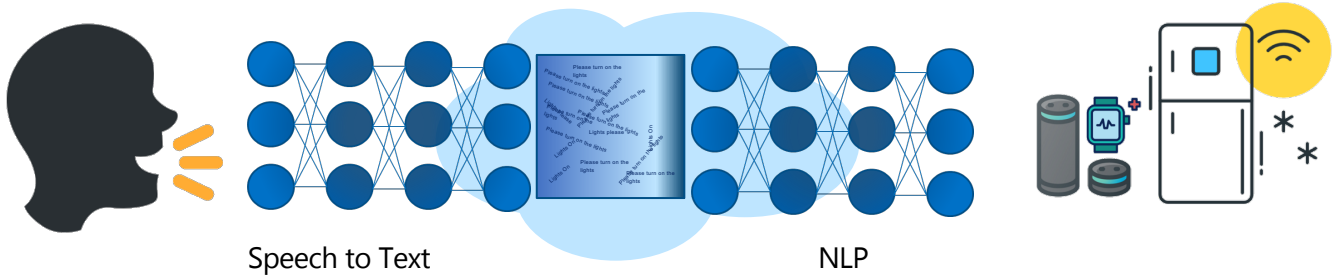
MLA MAPLE LEAF ANGELS

500

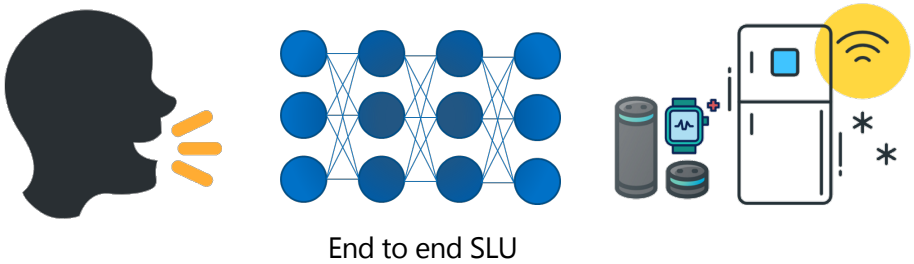
Fluent.ai Technology

End to end Spoken Language Understanding

Conventional approaches



fluent.ai™



- Smaller training data needs
- Higher accuracy and robustness against noise
- Offline and personalizable
- Any language, multi-language

EMBEDDED + ROBUST + MULTILINGUAL

Light + Works offline

Noise, Distance, Filler words

Any language, multiple languages

Large Global Opportunity & Demand

Our Right to Win

*"Fluent's **unique** models can be trained quickly to deliver the required **accuracy in many dialects, languages and noise conditions** and be **embedded** on the world's devices."*

– William Tunstall-Pedoe, Founder of Evi (Acquired by Amazon Alexa) Advisor/Investor in Fluent.ai

fluent.ai wakeword™


Smart-home Devices


Smart Toys


Robotics & Industrial IoT


Wearables


Car Infotainment


Voice Remotes

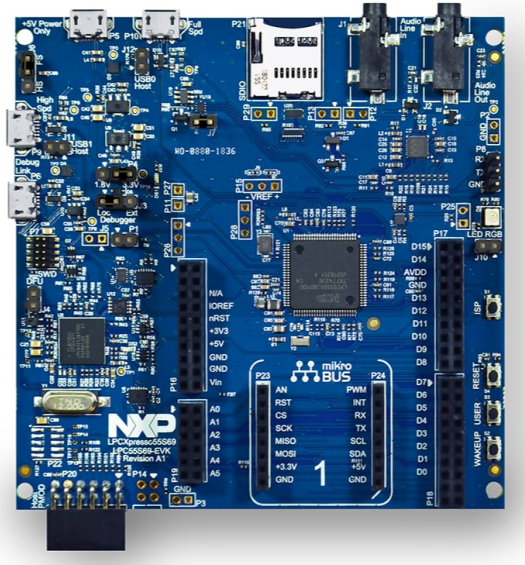
- Wake up your voice-enabled device with one of our low-power keyword spotting solutions that beat state of the art systems.
- Less than 5% FRR at 3FAs/24hrs

Single, Multiple, User-Trainable

System Req.	RAM	Storage	Latency	Min. Freq.
Arm Cortex M4	25 KB	200 KB	100 ms	48 MHz

Demo: Multiple WWs on NXP LPC55S69 running Arm Cortex M33

- Arm Cortex-M33 microcontroller running at up to 150 MHz



fluent.ai air™

Wearables

Smart Speakers

Car Infotainment

Wearables

Smart Home Hubs

Voice Remotes

Voice AI for on-device speech understanding

World's first end to end spoken language understanding system

=>faster, more flexible and more accurate voice user interfaces than conventional technologies.

- Offline/on-device for guaranteed privacy and security
- Any language and accent, multiple languages concurrently
- Personalizable by the end-user
- Lower development cost, faster Time-to-Market

System Req.	RAM	Storage	Latency	Min. Freq.
Arm Cortex M4	100 KB	550 KB	< 200 ms	48 MHz

Demo: Multiple WWs and multilingual intent on Arm Cortex M4

- Arm Cortex-M4 microcontroller running at up to 100 MHz.

  bit.ly/fluent_ww_air_cortexm4

Community contributions

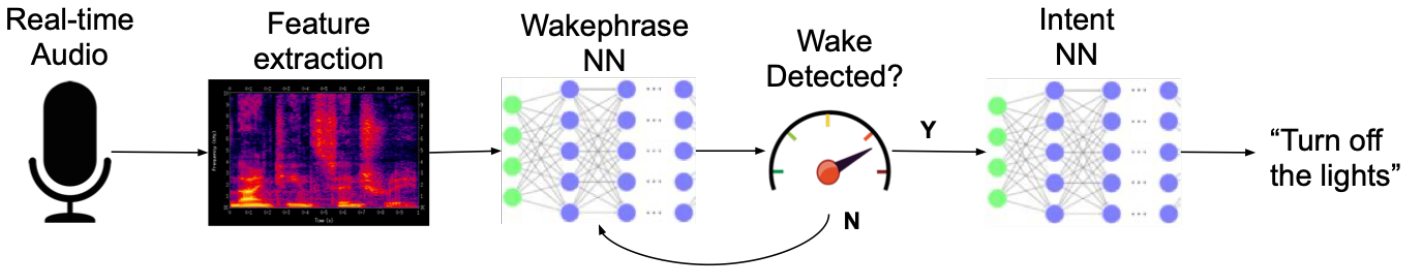
- Fluent Speech Commands Dataset
  - Speech to intent dataset
  - ~28,000 utterances from ~100 speakers
  - 31 intents, 254 commands
  - Link: bit.ly/fluent-speech-commands
  - Downloaded over 500 times. Many research papers.

- Speech Brain Project at MILA
  - A PyTorch based speech toolkit

Fluent.ai µCore

Fluent.ai µCore

- Proprietary low-resource spoken language understanding library
- Detects wakephrase(s) or keywords + commands
- Optimized for Arm Cortex-M series, e.g., M4, M33, M7



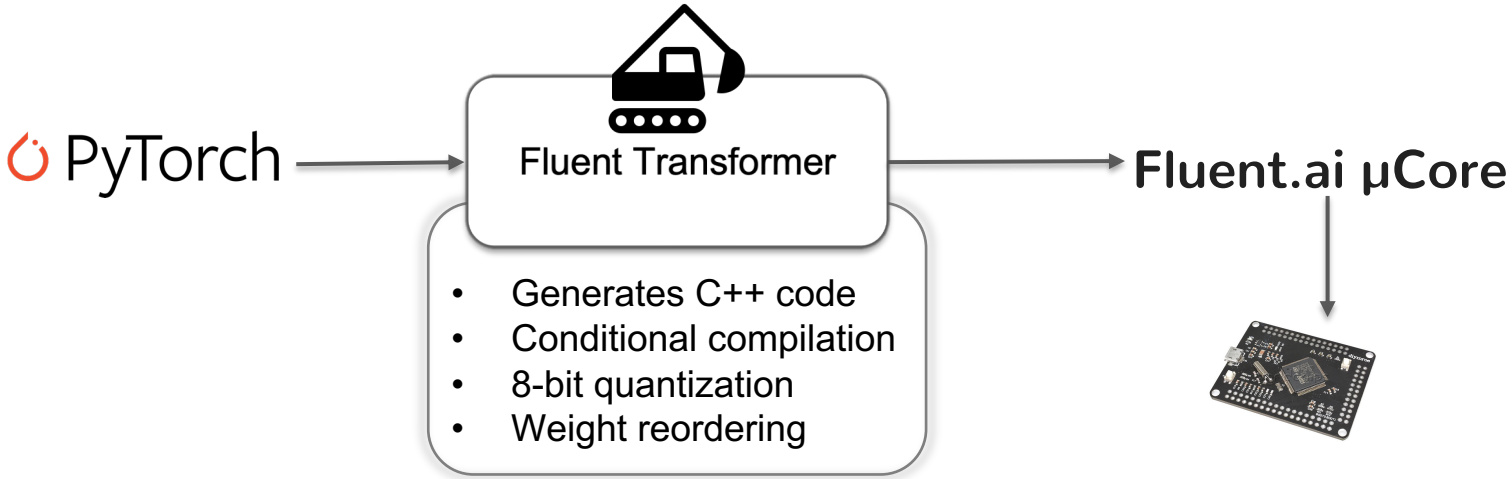
Challenges

- Taking neural networks from GPUs to MCUs: } **Fluent.ai Transformer**

- Low-footprint (memory and CPU usage)
- Real-time processing for low latency recognition } **Fluent.ai µCore**

- Cross-platform & modular
  - Uses Arm CMSIS-DSP & CMSIS-NN
    - Take advantage of assembly optimizations on Arm platforms
    - Liberal Apache 2.0 license

Transforming models



- Trained model on GPU using PyTorch
- Perform post-processing
- Generate C++ code describing model
- Compile model C++ code with library

Real-time processing

Convolution in existing libraries designed for images, not time-series

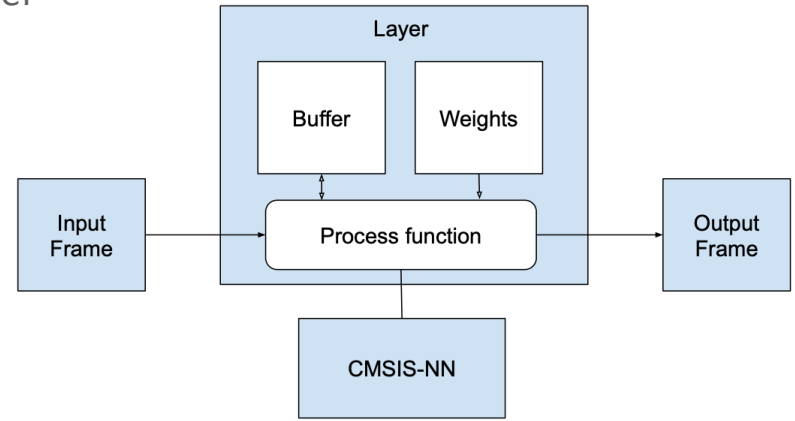
Training - batched utterances	Inference - streaming audio
Entire utterance is available	Audio streamed one frame at a time
Decoding latency not considered	Latency minimized for good user experience
Finite utterance length	Continuously listening, NN applied in overlapping windows
Activations for entire network stored in memory	Memory usage must be minimized

Layer types

- Streaming layer types
  - Unidirectional recurrent layers (GRU, LSTM)
  - Convolution / depthwise-separable
  - Windowed functions (e.g. MaxPool)
  - Streaming cumulative functions (e.g. GlobalMaxPool)
  - Skip connections
  - Activation functions (ReLU, sigmoid, tanh)

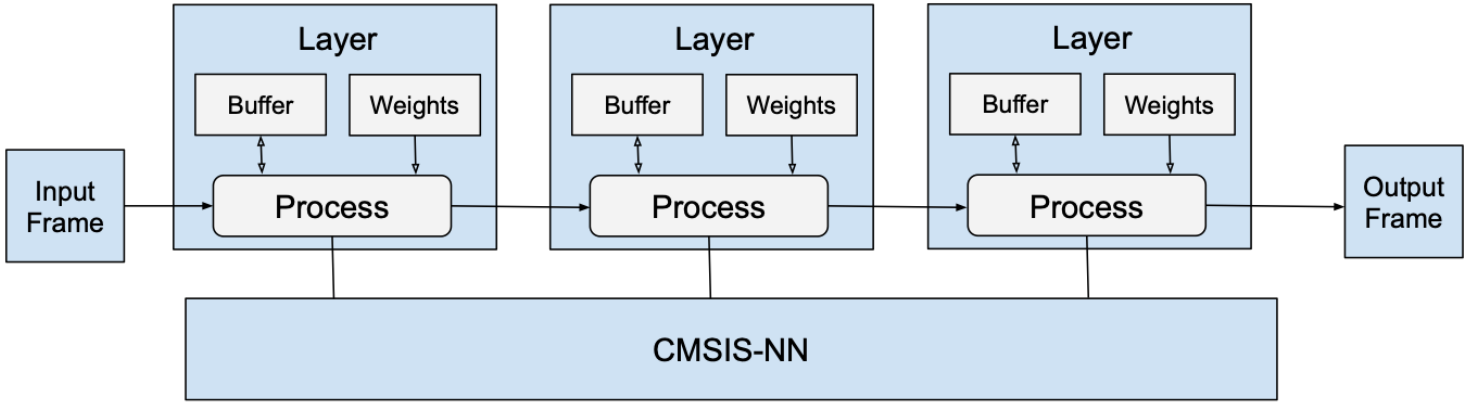
NN Layer structure

- All NN weights are stored in Flash
  - Arm MCU platform allows network weights to be fetched layer by layer
  - Only activation buffer is stored in RAM
- Process function
  - Uses CMSIS
  - Calculates activations as data is received and updates buffer
  - 1 frame input/output



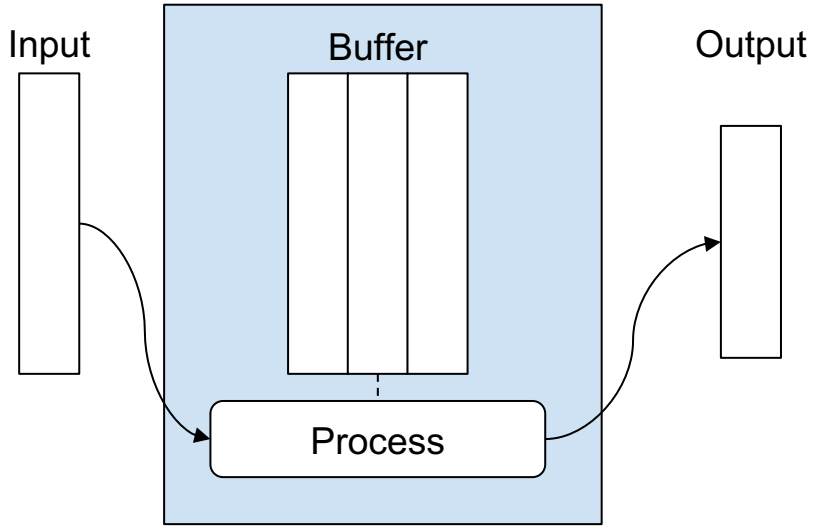
Sequence of layers

- Layers joined in sequence
- Input frame propagates through layers
- Layers are independent



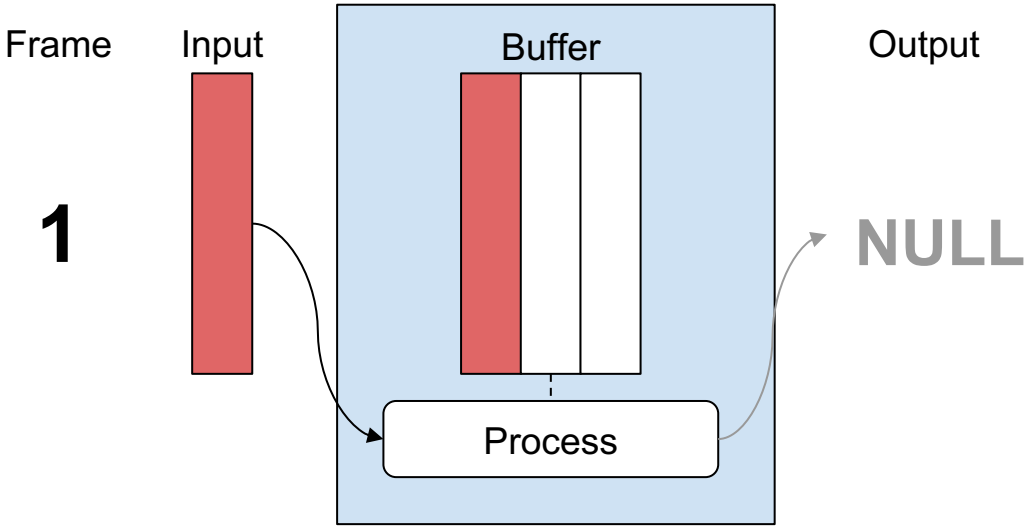
Convolution example

Streaming convolution, kernel size = 3, stride = 2



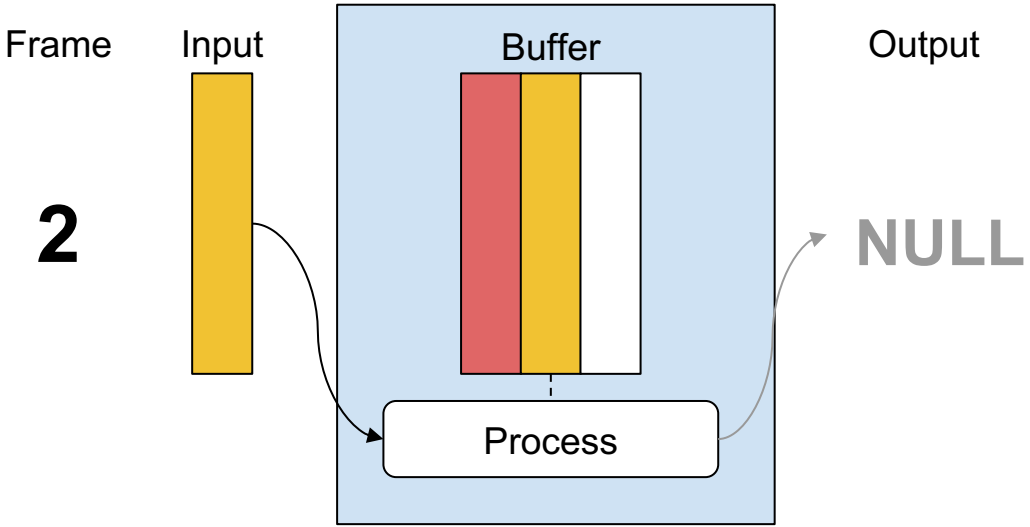
Convolution example

Streaming convolution, kernel size = 3, stride = 2



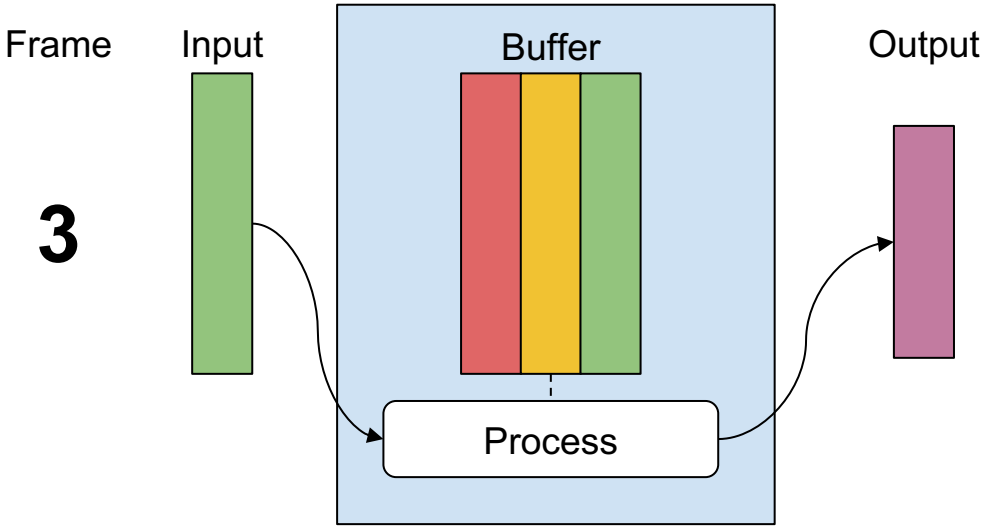
Convolution example

Streaming convolution, kernel size = 3, stride = 2



Convolution example

Streaming convolution, kernel size = 3, stride = 2

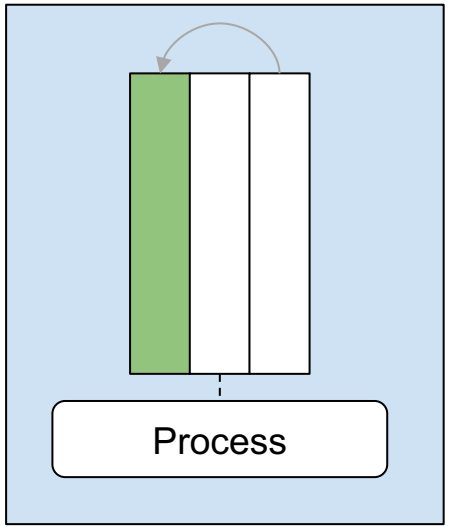


Convolution example

Streaming convolution, kernel size = 3, stride = 2

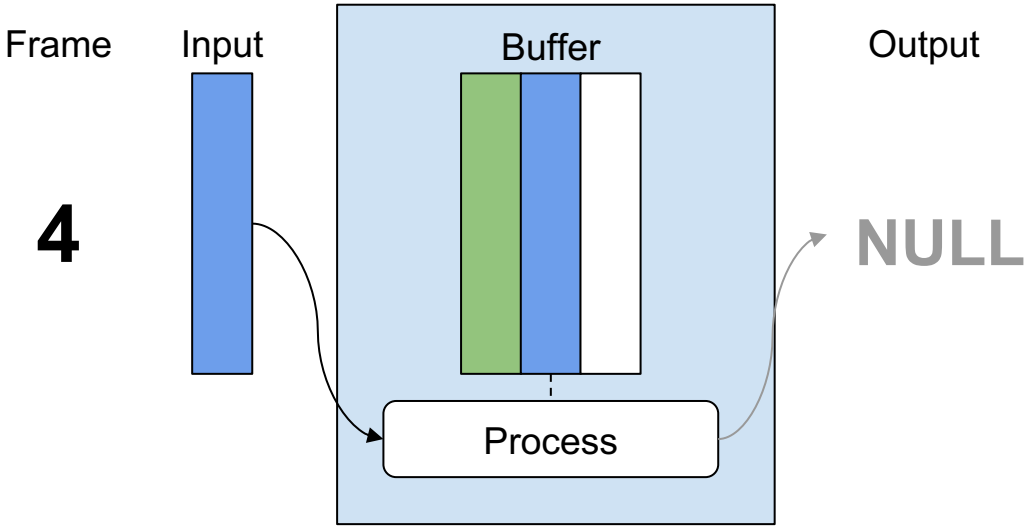
Frame

3



Convolution example

Streaming convolution, kernel size = 3, stride = 2

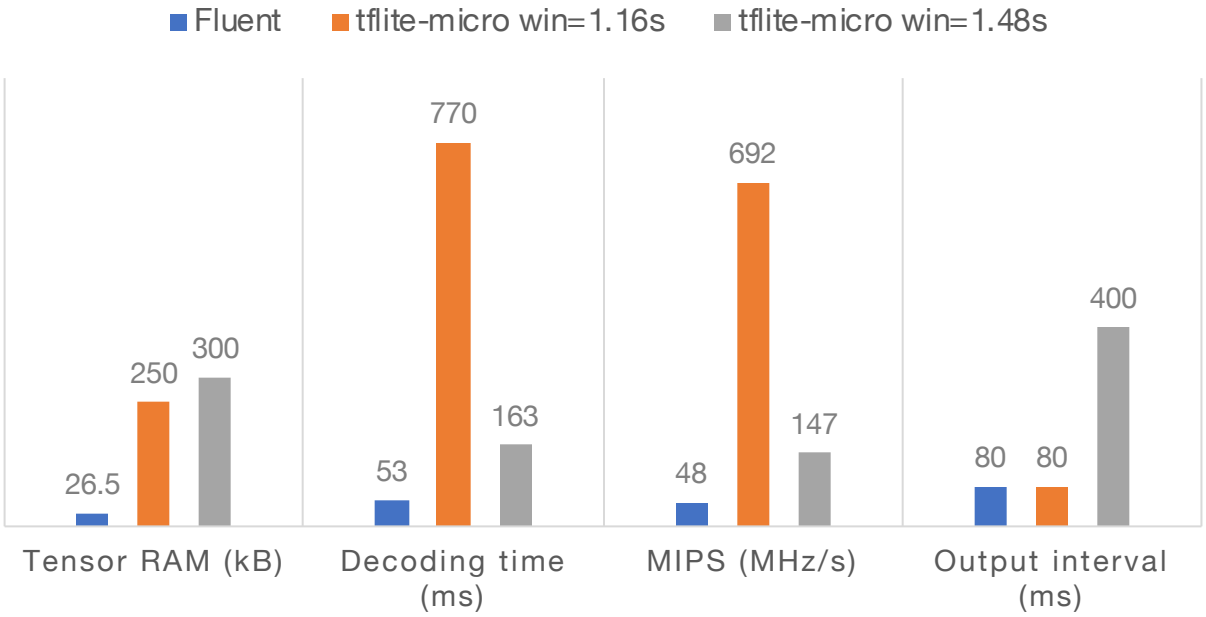


Advantages of streaming NN

- No need to keep features/activations for entire utterance
  - => lower memory requirements
- Live processing while the user is speaking
  - => lower latency
- Redundant calculations eliminated by not using overlapping windows
  - => lower CPU usage

µCore vs tflite-micro

- Same Fluent wakeword model running on µCore and tflite on a Linux machine



µCore Summary

- Low footprint
  - CPU efficient code, reduced model size & memory load operations
  - small code size — to fit into limited flash memory
  - small memory footprint (RAM)
  - e.g., Arm Cortex M4, M33 @48 MHz
- Streaming NN: optimized for low latency / real-time operations
- Cross platform

! Cheaper yet effective device designs !

more demos

Demo: Smart-home voice control on Cortex M7

fluent.ai™

Smart Home
Voice Control Demo

Demo Hardware: 216MHz STM32F769 microcontroller

Model Size: 1.3 MB Storage, 128 KB RAM for
3 wake words and 500 intents in English

arm AI

AI Virtual Tech Talks Series

fluent.ai™

Thank You

Danke

Merci

谢谢

ありがとう

Gracias

Kiitos

감사합니다

धन्यवाद

شكرًا

תודה