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

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
500
Fluent.ai Technology

End to end Spoken Language Understanding

Conventional approaches

- Speech to Text
- NLP

End to end SLU

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

<table>
<thead>
<tr>
<th>System Req.</th>
<th>RAM</th>
<th>Storage</th>
<th>Latency</th>
<th>Min. Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arm Cortex M4</td>
<td>25 KB</td>
<td>200 KB</td>
<td>100 ms</td>
<td>48 MHz</td>
</tr>
</tbody>
</table>
Demo: Multiple WWs on NXP LPC55S69 running Arm Cortex M33

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

<table>
<thead>
<tr>
<th>System Req.</th>
<th>RAM</th>
<th>Storage</th>
<th>Latency</th>
<th>Min. Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arm Cortex M4</td>
<td>100 KB</td>
<td>550 KB</td>
<td>&lt; 200 ms</td>
<td>48 MHz</td>
</tr>
</tbody>
</table>
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

PyTorch → Fluent Transformer → Fluent.ai µCore

- Generates C++ code
- Conditional compilation
- 8-bit quantization
- Weight reordering
# Real-time processing

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

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

Demo Hardware: 216MHz STM32F769 microcontroller

Model Size: 1.3 MB Storage, 128 KB RAM for 3 wake words and 500 intents in English
Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה