Boosting keyword spotting through on-device learnable user speech characteristics

Cristian Cioflan  ETH Zürich
Lukas Cavigelli  Huawei Technologies
Luca Benini  ETH Zürich, Università di Bologna

PULP Platform
Open Source Hardware, the way it should be!
Keyword Spotting at the Extreme Edge

- Voice-controlled personal assistants
- Drones controlled remotely to investigate hard-to-reach locations
- Hearing devices adapted to the environment conditions
Accuracy degrades in real-world conditions

- **Unknown environments where pretraining (offline) ≠ target (online) data**
  - Domain shifts, differences in sensors, knowledge expansion
  - Accents, genders, background noises

- WER variation by US region in Bing Speech (left) and YouTube Captions (right) [Tatman2017]

- Feminine vs. masculine accuracy on function words for two speech translation model, in three languages [Savoldi2022]

- Keyword spotting accuracy drops by 3%-26% compared to silent environments [Cioflan2024]
How to mitigate the performance degradation?

- Server-side training on on-site data
  - Does not respect privacy
- Communication reduces device lifetime
- User-specific labeled data is scarce
How to mitigate the performance degradation?

- Server-side training on on-site data
  - Does not respect privacy
- Communication reduces device lifetime
- User-specific labeled data is scarce

- On-device training (by backpropagation)
  - Requires memory beyond tinyML constraints
  - Latency increases with #layers, #samples, #epochs
On-Device Learning of Speaker-Aware Embeddings

- Lightweight backbone, suitable for ODL [Cioflan2024]
On-Device Learning of Speaker-Aware Embeddings

- Lightweight backbone, suitable for ODL [Cioflan2024]
On-Device Learning of Speaker-Aware Embeddings

- We introduced embeddings and fuse them with the KWS backbone

- Speaker identity $\mapsto n$-dimensional vector

![Diagram showing the process of embedding and fusion with the KWS backbone.]
On-Device Learning of Speaker-Aware Embeddings

• We introduced embeddings and fuse them with the KWS backbone

• Speaker identity $\mapsto n$-dimensional vector (jointly learned)
On-Device Learning of Speaker-Aware Embeddings

- We introduced embeddings and fuse them with the KWS backbone

- Speaker identity $\mapsto$ $n$-dimensional vector (jointly learned)

- Feature-level fusion along channel dimension (projection $\propto$ speech char.)
On-Device Learning of Speaker-Aware Embeddings

• We introduced embeddings and fuse them with the KWS backbone

![Diagram of a microphone and speaker identity fusion with convolutional layers and classification]

- Speaker identity $\mapsto n$-dimensional vector (jointly learned)
- Feature-level fusion along channel dimension (projection $\propto$ speech char.)
- Late fusion minimizing on-device learning costs

$$\sum y_{gt} \log(y_n)$$
Google Speech Commands dataset

GSC–10
[Warden2018]

on stop go

off

yes left

don right

up

down six

seven eight

nine

go

one

follow

four backward

bed

visual

learn

two cat

tree

wow

marvin sheila

happy

dog

forward

house

right bird

up

on no

yes

off

down
Lessons learned during pretraining

GSC–10
[Warden2018]

GSC–35

<table>
<thead>
<tr>
<th>Error rate</th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.9%</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4.3%</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5.0%</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Pretraining: 2256, 2612, 2617
Evaluation: 362, 6, 1
Lessons learned during pretraining

Error rate 6.9% 4.3% 5.0%

Train 1 4 7
Validation 1 1 1
Test 1 1 1

On-Device Learning for 6 speakers (between 4 and 22 training samples per class)

GSC–10 [Warden2018]
Lessons learned during pretraining

GSC–10
[Warden2018]

GSC–35

Error rate

6.9%

4.3%

Train

1

4

7

1

Validation

1

1

1

Test

1

1

1

On-Device Learning for 6 speakers
(between 4 and 22 training samples per class)

Embedding

<table>
<thead>
<tr>
<th>Error rate [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>- 5.39</td>
</tr>
<tr>
<td>Addition 5.38</td>
</tr>
<tr>
<td>Multiplication 5.28</td>
</tr>
<tr>
<td>Backbone-compatible concatenation 5.32</td>
</tr>
<tr>
<td>Classifier-compatible concatenation 5.75</td>
</tr>
</tbody>
</table>

On-Device Learning for 6 speakers
(between 4 and 22 training samples per class)
Lessons learned during pretraining

Error rate

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSC–10</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>GSC–35</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

On-Device Learning for 6 speakers (between 4 and 22 training samples per class)

Multiplicative fusion of backbone features and user embeddings

<table>
<thead>
<tr>
<th>Operation</th>
<th>Error rate [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>5.39</td>
</tr>
<tr>
<td>Multiplication</td>
<td>5.28</td>
</tr>
<tr>
<td>Backbone-compatible concatenation</td>
<td>5.32</td>
</tr>
<tr>
<td>Classifier-compatible concatenation</td>
<td>5.75</td>
</tr>
</tbody>
</table>

On [Warden2018], GSC–10 and GSC–35 are used for training, validation, and testing.

[Multiplicative fusion] of backbone features and user embeddings leads to an error rate of 5.0%.
Few-shot learning of speaker embeddings

- Learning speech characteristics decreases KWS error rate
  - Evaluated on six speakers (American, British, and Indian English accents)
  - Sufficient to have four training samples per user

<table>
<thead>
<tr>
<th></th>
<th>GSC-10</th>
<th>Error rate decrease [%] ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td># classes</td>
<td>#samples = 4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.46</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.73</td>
<td></td>
</tr>
</tbody>
</table>

Pretraining error rate: 5.33%

<table>
<thead>
<tr>
<th></th>
<th>GSC-35</th>
<th>Error rate decrease [%] ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td># classes</td>
<td>#samples = 4</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>3.08</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>4.47</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>5.74</td>
<td></td>
</tr>
</tbody>
</table>

Pretraining error rate: 30.08%
Few-shot learning of speaker embeddings

- Learning speech characteristics decreases KWS error rate
  - Evaluated on six speakers (American, British, and Indian English accents)
  - Sufficient to have four training samples per user
  - More training samples → more user-specific information → better performance

<table>
<thead>
<tr>
<th>GSC-10</th>
<th>Error rate decrease [%] ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># classes</td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>0.73</td>
</tr>
</tbody>
</table>

Pretraining error rate: 5.33%

<table>
<thead>
<tr>
<th>GSC-35</th>
<th>Error rate decrease [%] ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># classes</td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.08</td>
</tr>
<tr>
<td></td>
<td>4.47</td>
</tr>
<tr>
<td></td>
<td>5.74</td>
</tr>
</tbody>
</table>

Pretraining error rate: 30.08%
Few-shot learning of speaker embeddings

- Learning speech characteristics decreases KWS error rate
  - Evaluated on six speakers (American, British, and Indian English accents)
  - Sufficient to have four training samples per user
  - More training samples → more user-specific information → better performance

- User embeddings mitigate overfitting in domain shifts

<table>
<thead>
<tr>
<th># classes</th>
<th>Update only the backbone</th>
<th>Update the backbone and the embeddings</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/GSC-10</td>
<td>1.72</td>
<td>2.06</td>
</tr>
<tr>
<td>20/GSC-35</td>
<td>3.28</td>
<td>3.54</td>
</tr>
<tr>
<td>30/GSC-35</td>
<td>4.34</td>
<td>4.67</td>
</tr>
</tbody>
</table>
On-Device Learning and tinyML constraints

• Deployment estimates on PULP Vega SoC [Rossi2022]
  • 79 GFLOP/s/W @ 50 mW
  • 128 kB L1 TCDM, 1.6 MB L2 SRAM
On-Device Learning and tinyML constraints

- Deployment estimates on PULP Vega SoC [Rossi2022]
  - 79 GFLOP/s/W @ 50 mW
  - 128 kB L1 TCDM, 1.6 MB L2 SRAM

<table>
<thead>
<tr>
<th>DS-CNN Model</th>
<th>S(mall)</th>
<th>M(edium)</th>
<th>L(arge)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters [k]</td>
<td>23.7</td>
<td>138.1</td>
<td>416.7</td>
</tr>
<tr>
<td>Error decrease [%]</td>
<td>0.73</td>
<td>0.94</td>
<td>0.3</td>
</tr>
<tr>
<td>FLOPs [M]</td>
<td>1.04</td>
<td>2.8</td>
<td>4.5</td>
</tr>
<tr>
<td>Memory [kB]</td>
<td>3.6</td>
<td>9.7</td>
<td>15.5</td>
</tr>
<tr>
<td>Energy [μJ]</td>
<td>13.22</td>
<td>35.53</td>
<td>57.01</td>
</tr>
</tbody>
</table>

- ODL under 16 kB, feasible for tinyML
- 13 μJ/epoch prolongs device lifetime
- vs. full training: 78% of the error rate, 340× fewer FLOPs
- vs. classifier update: 13× less energy efficient, 55% more accurate
Conclusions

• Learning speech characteristics increases keyword spotting accuracy
  • Multiplicative user embeddings integrated through late-level fusion
  • Error rate decreases by up to 5.74% (relative improvement of 19%)
    • Sufficient to have only 4 training samples per class
• Speech embeddings are suitable for On-Device Learning on tinyML systems
  • 16 kB of memory for backpropagation learning, 128 kB of storage per class for samples
  • 0.73% error decrease with 13 μJ/epoch (260 μJ per speaker with early stopping)
Conclusions

• Learning speech characteristics increases keyword spotting accuracy
• Speech embeddings are suitable for On-Device Learning on tinyML systems
• What are we working on now?
  • Pairing efficient on-device-learning with state-of-the-art (linear) attention-based backbones [Scherer2024]
  • On-device implementation on GAP9 – derivative of Vega
  • From user embeddings to environment embeddings – boosting performance by focusing on the domain shift
  • Domain-Aware Keyword Spotting at the Extreme Edge


Copyright Notice

This presentation in this publication was presented at the tinyML® Research Symposium 2024. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org