Exploring Filterbank Learning for Keyword Spotting

Iván López-Espejo[1], Zheng-Hua Tan[1] and Jesper Jensen[1,2]

[1] *Department of Electronic Systems, Aalborg University, Denmark*
[2] *Oticon A/S, Denmark*

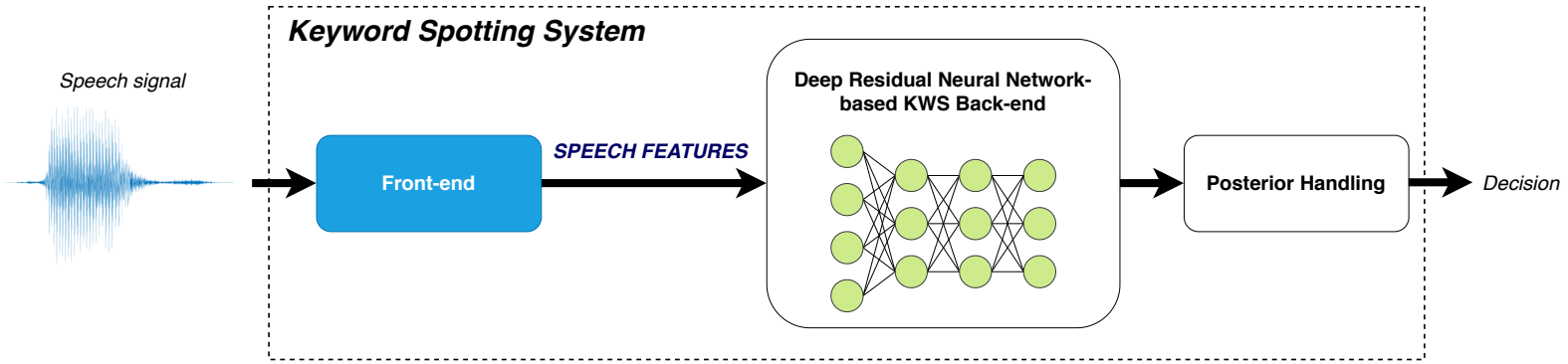
{ivl,zt,jje}@es.aau.dk, jesj@oticon.com

Monday 18th January, 2021

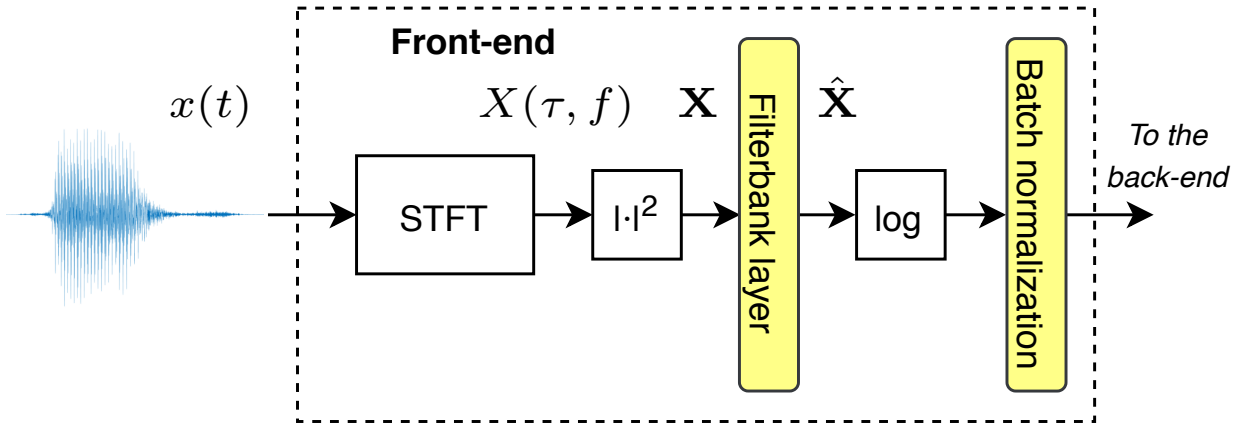
Overview

- Handcrafted speech features (e.g., MFCCs) are not necessarily optimal for any particular speech processing task.
- **Recent trend:** development of end-to-end deep learning systems where the feature extraction process is optimal according to the task and training criterion.
- For some applications, optimal filterbank learning has shown improvements with respect to using a standard Mel filterbank:
  - *Speaker verification anti-spoofing*
  - *Audio source separation and audio scene classification*
  - *...*

- To explore the possible advantages of employing *learned filterbanks over handcrafted speech features* for **keyword spotting (KWS)**:
  1. Filterbank matrix weight learning in the power spectral domain;
  2. Parameter learning of a (psychoacoustically-motivated) gammachirp filterbank.

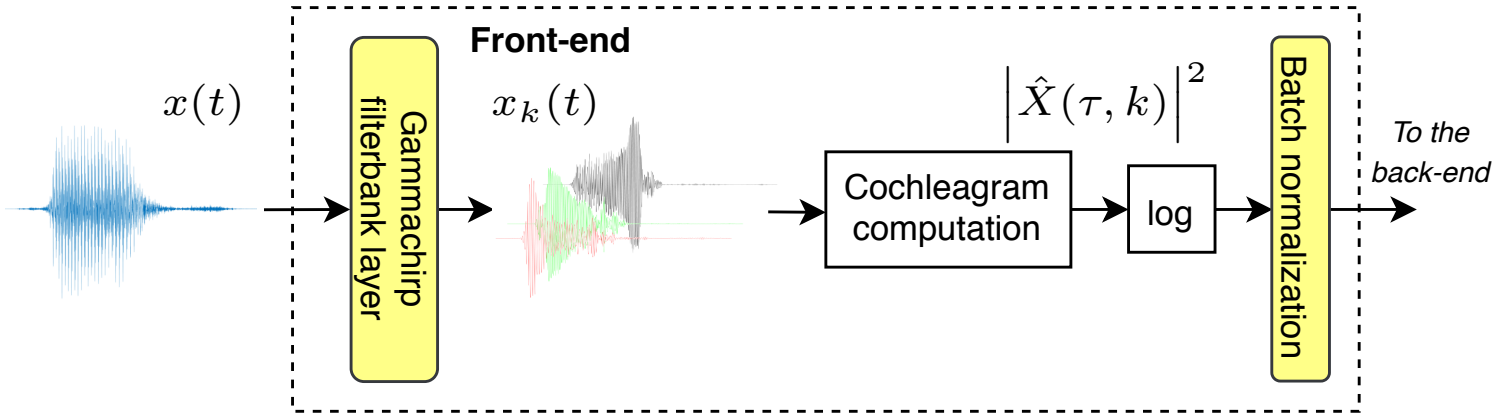


For both 1 and 2, the learnable filterbank parameters are optimized by backpropagation jointly with the KWS back-end!



- **Filterbank layer:** $\hat{\mathbf{X}} = \mathbf{X} \cdot h(\mathbf{W})$
- $\mathbf{W}$ is the learnable filterbank matrix.
- $h(\cdot) = max(\cdot, 0)$ is the element-wise applied ReLU function to ensure the positivity of the filterbank weights.

- **Gammachirp filterbank:**
  $g_c(t,k) = a_k t^{n-1} e^{-2\pi b \mathrm{ERB}(f_k)t} \cos\left(2\pi f_k t + c\log(t) + \phi\right)$
- At moderate stimulus levels, $\mathrm{ERB}\left(f_k\right) = 24.7 + 0.108 f_k$ [Hz]



- **Gammachirp filterbank layer:** $x_k(t) = x(t) * g_c(t,k)$
- **Trainable parameters:** a_k, n, b, c, f_k and the ERBs.
- To preserve their physical meaning, the ReLU function is applied to a_k, b, f_k and the ERBs, whereas n is constrained to be $\max\left(n,\ 1\right)$

Experimental Framework

- We use the Google Speech Commands Dataset (**GSCD**) for KWS experiments:
  - 105,829 one-second long speech files
  - Each file comprises one word among 35 possible candidate words

- A deep residual neural network-based KWS back-end[1] is trained to spot the **10 keywords** "yes", "no", "up", "down", "left", "right", "on", "off", "stop" and "go".

- Utterances with the remaining 25 words of the GSCD (i.e., non-keywords) are used to define the **filler class** $\Rightarrow$ the KWS back-end has to solve an **11-class classification problem**.

[1] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in *Proc. of ICASSP 2018*

Experimental Framework

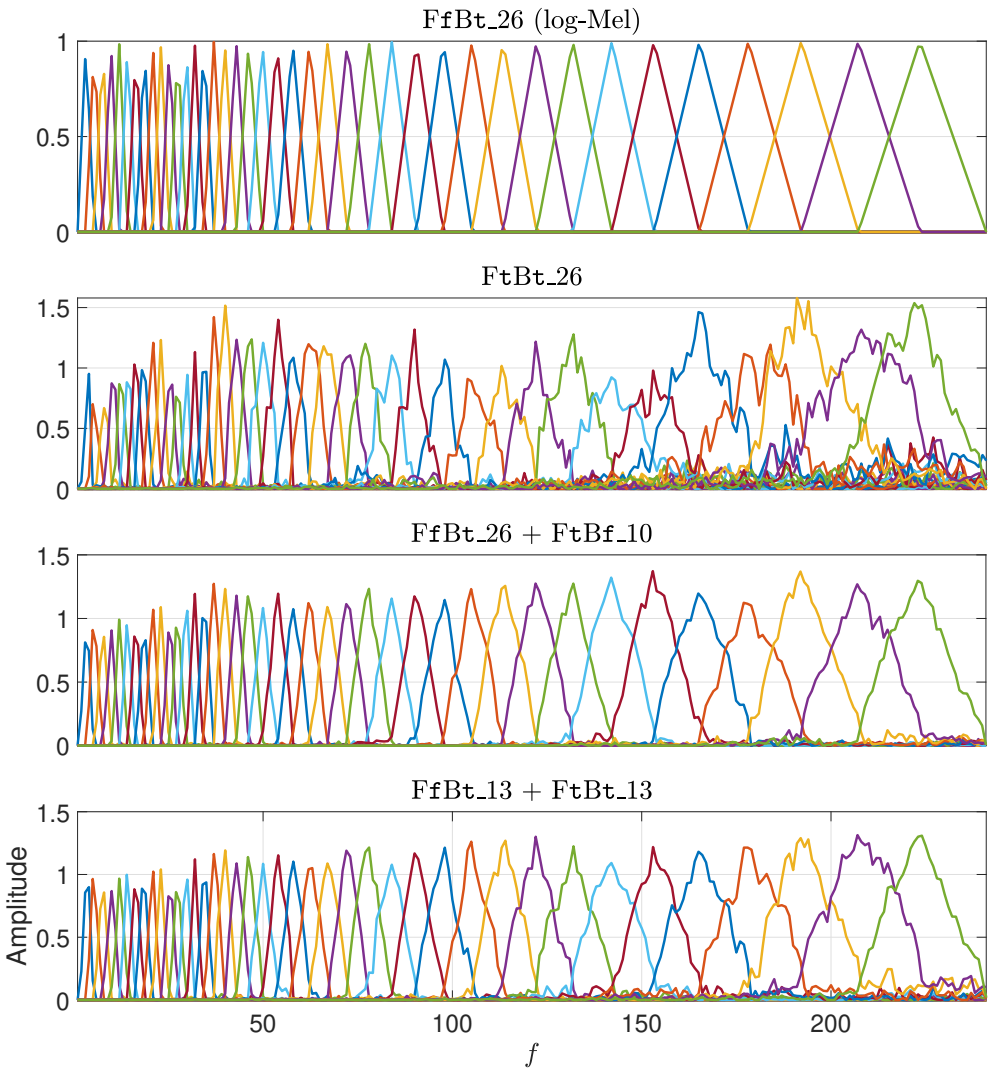
- The number of filterbank channels is 40.
- The back- and front-end are trained using categorical cross-entropy and Adam.
- As a KWS performance metric, we employ **accuracy**: the ratio of the number of correct predictions over the total number of them.
- Accuracy results are provided along with 95% confidence intervals calculated from outputs of 10 different back-end realizations trained with different random parameter initialization.

Results

Filterbank Matrix Learning

- **W** is initialized by a Mel filterbank.
- Naming FxBy_z:
  1. x $\in \{\texttt{t}, \texttt{f}\}$ indicates whether the front-end is trained, `t`, or not, `f`.
  2. y $\in \{\texttt{t}, \texttt{f}\}$ is the same, but for the back-end.
  3. z is the number of training epochs.

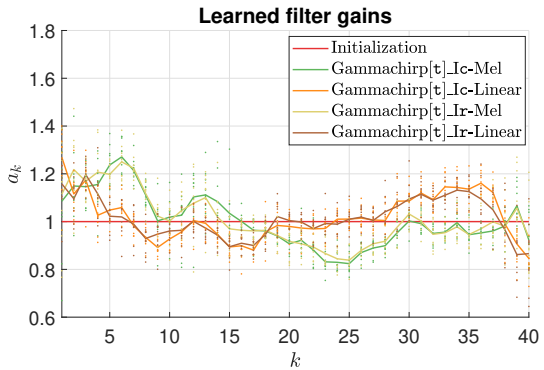
Test	Accuracy (%)
FfBt_26 (log-Mel)	95.64 $\pm$ 0.33
FtBt_26	95.73 $\pm$ 0.24
FfBt_26 + FtBf_10	95.73 $\pm$ 0.38
FfBt_13 + FtBt_13	95.30 $\pm$ 0.82

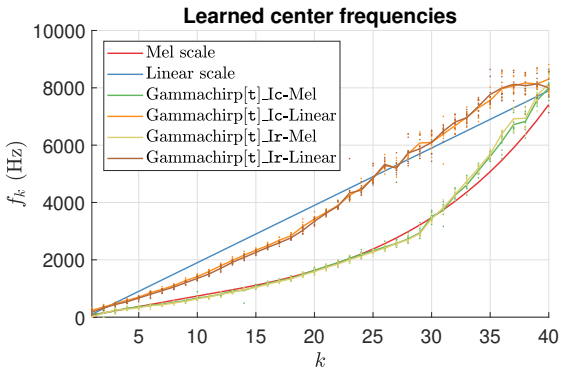


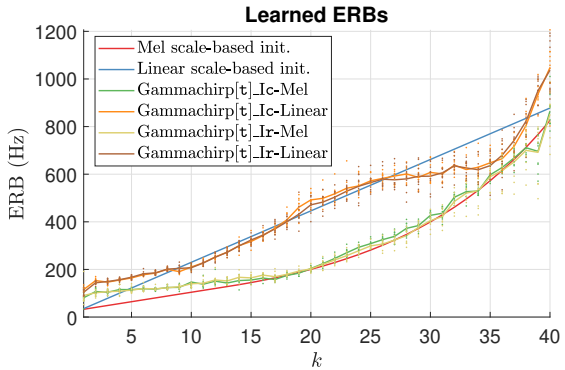
- Naming GC[x]_Iy-z:
  1. x $\in \{\texttt{t}, \texttt{f}\}$ indicates whether the front-end is trained, `t`, or not, `f` (back-end is always trained).
  2. y $\in \{\texttt{c}, \texttt{r}\}$ refers to the initialization of n, b and c, which can be constant, `c`, or random, `r`.
  3. z tells whether f_k and the ERBs are initialized by a Mel or a linear scale.

Test	Accuracy (%)	n	b	c
GT[f]_Ic-Mel	95.47 ± 0.36	4	1.019	0
GC[f]_Ic-Mel	95.45 ± 0.58	4	1.019	-1
GC[t]_Ic-Mel	95.12 ± 0.42	4.69 ± 0.07	0.976 ± 0.015	-0.84 ± 0.05
GC[t]_Ic-Linear	95.19 ± 0.52	4.44 ± 0.05	0.866 ± 0.019	-0.88 ± 0.02
GC[t]_Ir-Mel	94.68 ± 0.52	4.90 ± 0.51	0.976 ± 0.115	-0.97 ± 0.32
GC[t]_Ir-Linear	94.93 ± 0.45	4.65 ± 0.41	0.861 ± 0.075	-0.98 ± 0.38

Same KWS accuracy trends when using lighter back-end models!







Results

Feature Fusion

- *In speech acoustic modeling*: Sainath *et al.*[2] achieve to beat log-Mel features **only** by fusing learnable front-end features with them $\Leftarrow$ *it is unclear if this improvement is statistically significant!*

Test	**Accuracy (%)**
FfBt_26 (log-Mel)	95.64 $\pm$ 0.33
GC[t]_Ic-Linear	95.19 $\pm$ 0.52
Fusion	95.65 $\pm$ 0.43

- The learned gammachirp filterbank conveys no additional information for KWS.
- Other fusion combinations lead to the same conclusion.

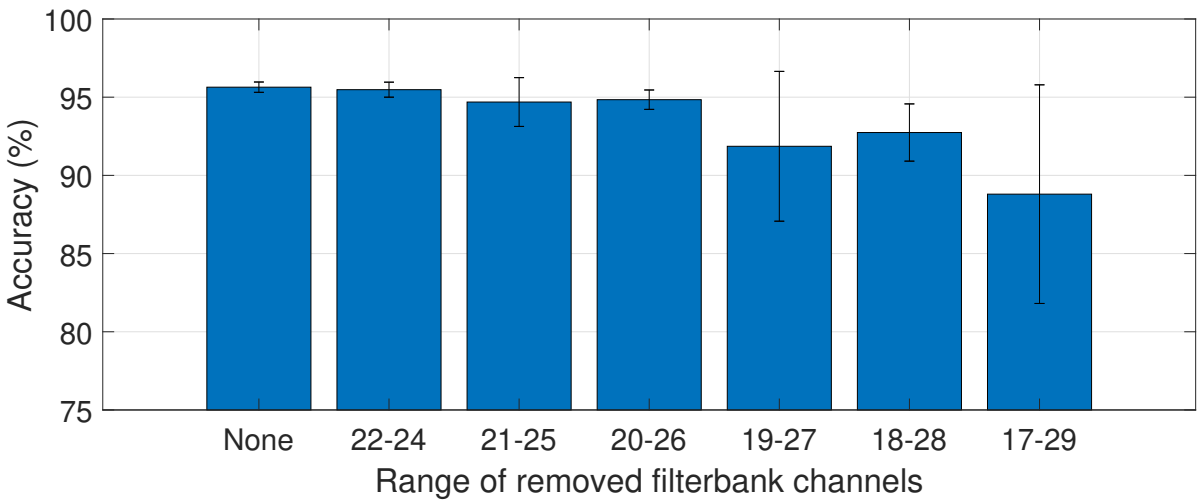
[2] T. N. Sainath *et al.*, "Learning the speech front-end with raw waveform CLDNNs," in *Proc. of Interspeech 2015*

Results

Filter Removal

- Is the filterbank and, in general, the speech feature design actually a crucial part of modern KWS systems?

Using log-Mel features:



$f_{k=23} \approx 2,000$ Hz, $\quad [20, 26] \equiv [1,626,\ 2,564]$ Hz

- KWS systems are fed with a great amount of redundant information.
- This gives clues on why the performance of learned filterbanks and handcrafted speech features is comparable.

Conclusion

- We have explored two different filterbank learning approaches for keyword spotting.
- In general, there are **no** statistically significant differences in terms of KWS accuracy between using a learned filterbank and handcrafted speech features $\Rightarrow$ the latter are still a good choice when employing modern KWS back-ends.
- The above could be a symptom of information redundancy $\Rightarrow$ new possibilities in the field of small-footprint KWS regarding the design of much more compact speech features.

Thanks for your attention!