Vocal Tract Length Perturbation for Speech Recognition with DNN-HMMs

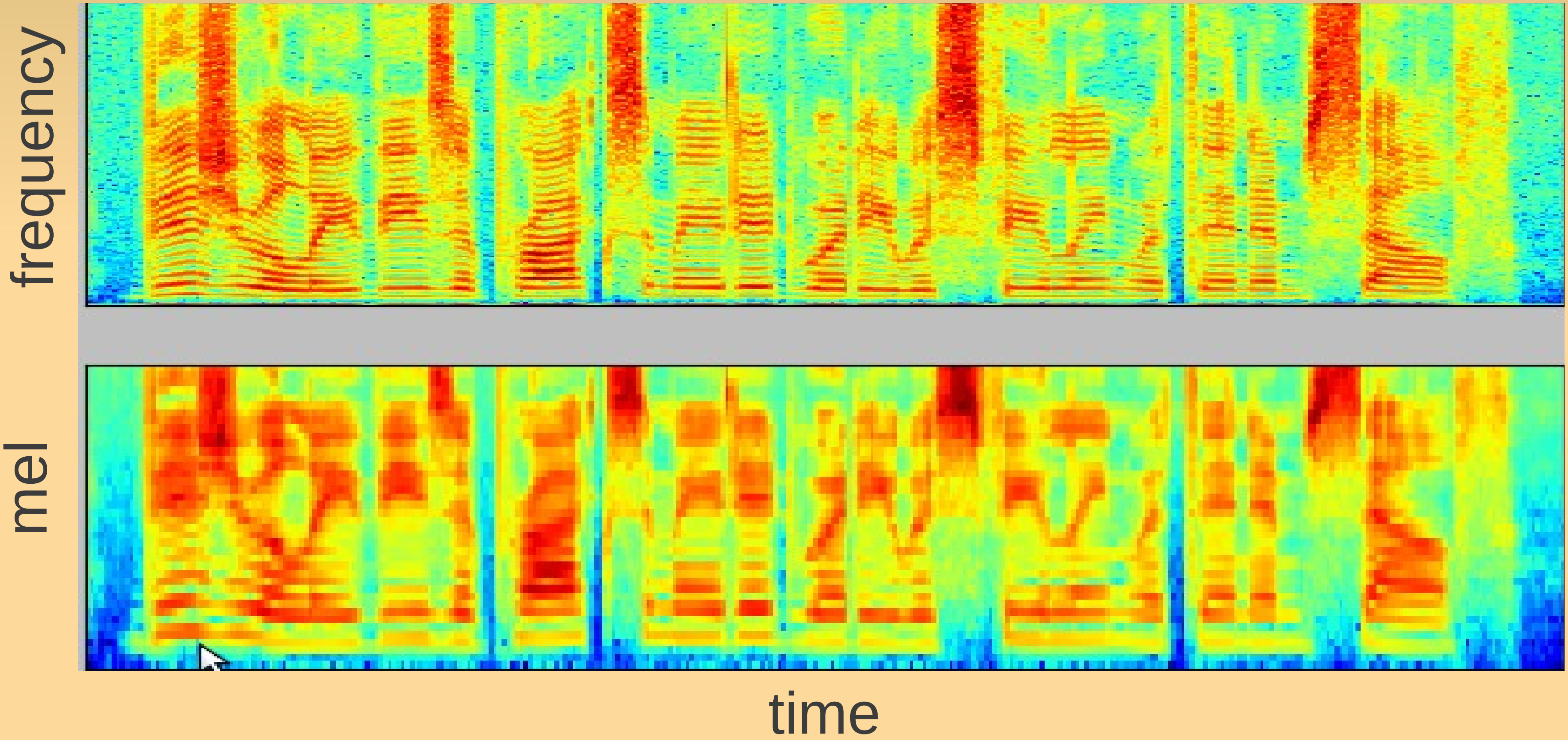
- Navdeep Jaitly
- Geoffrey Hinton

Outline

- Background on Mel Filterbanks
- Vocal Tract Length Normalization
- Vocal Tract Length Perturbation
- Results
- Avenues for Exploration

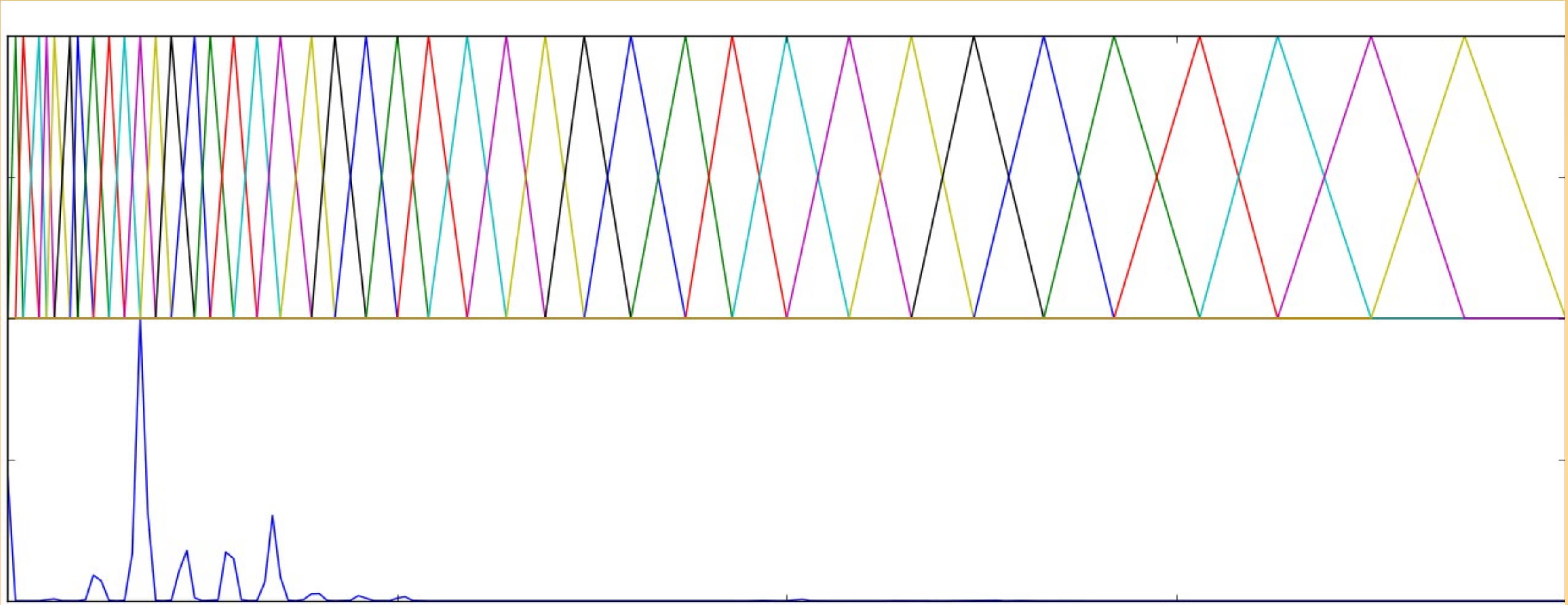
Mel log Filterbanks

- Low Resolution pre-processing of spectrograms



Under the hood

- Each frame of a spectrogram is processed by multiple filters, each of which look at a frequency subbands

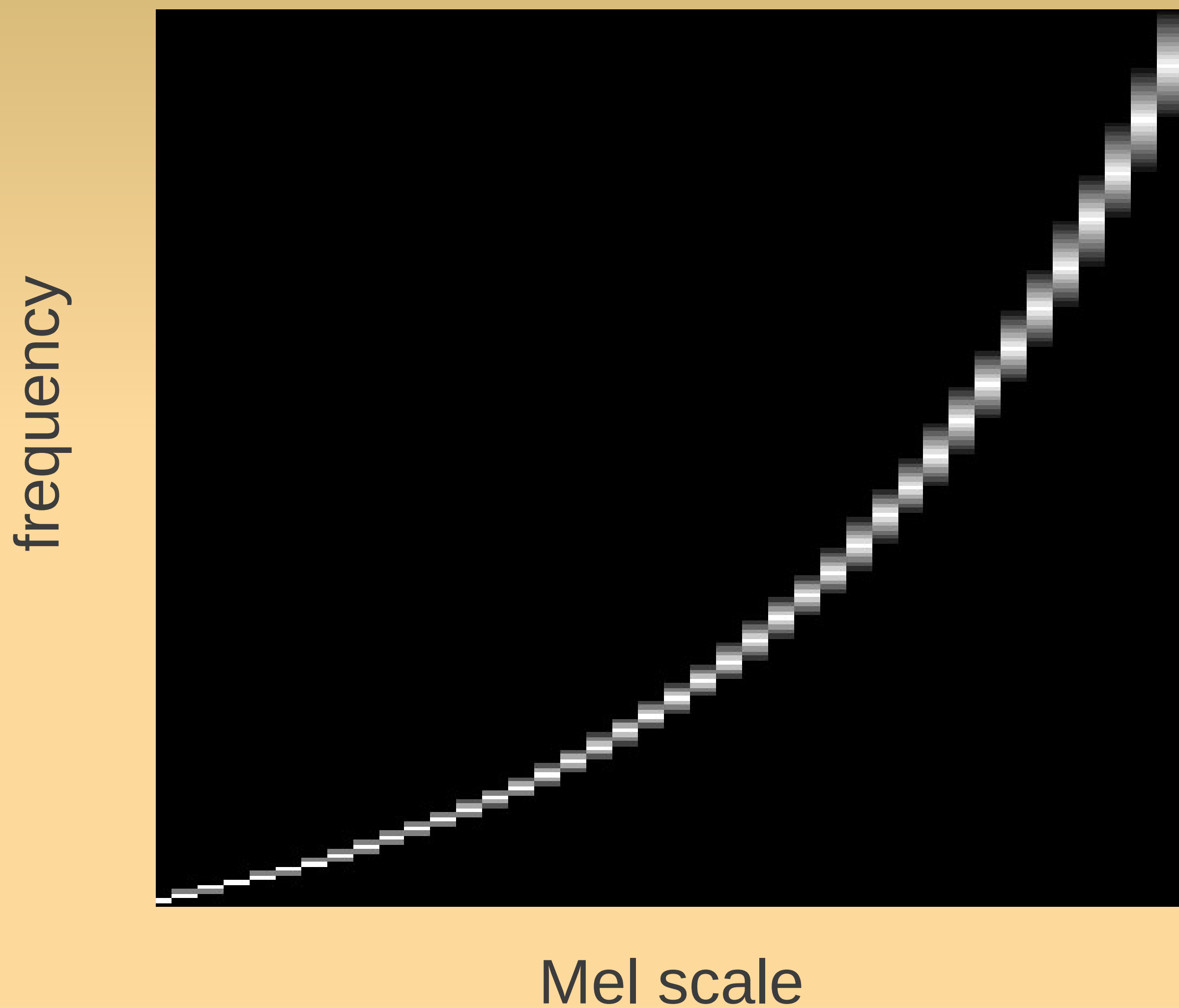


frequency

Some comments

- Filterbanks are just a linear layer of a neural networks – with a very specific, fixed architecture
  - fixed local filters, whose location, and window size depends on their center frequency
  - fixed weights (typically triangular)

Mel-Filterbanks are Fixed Layers



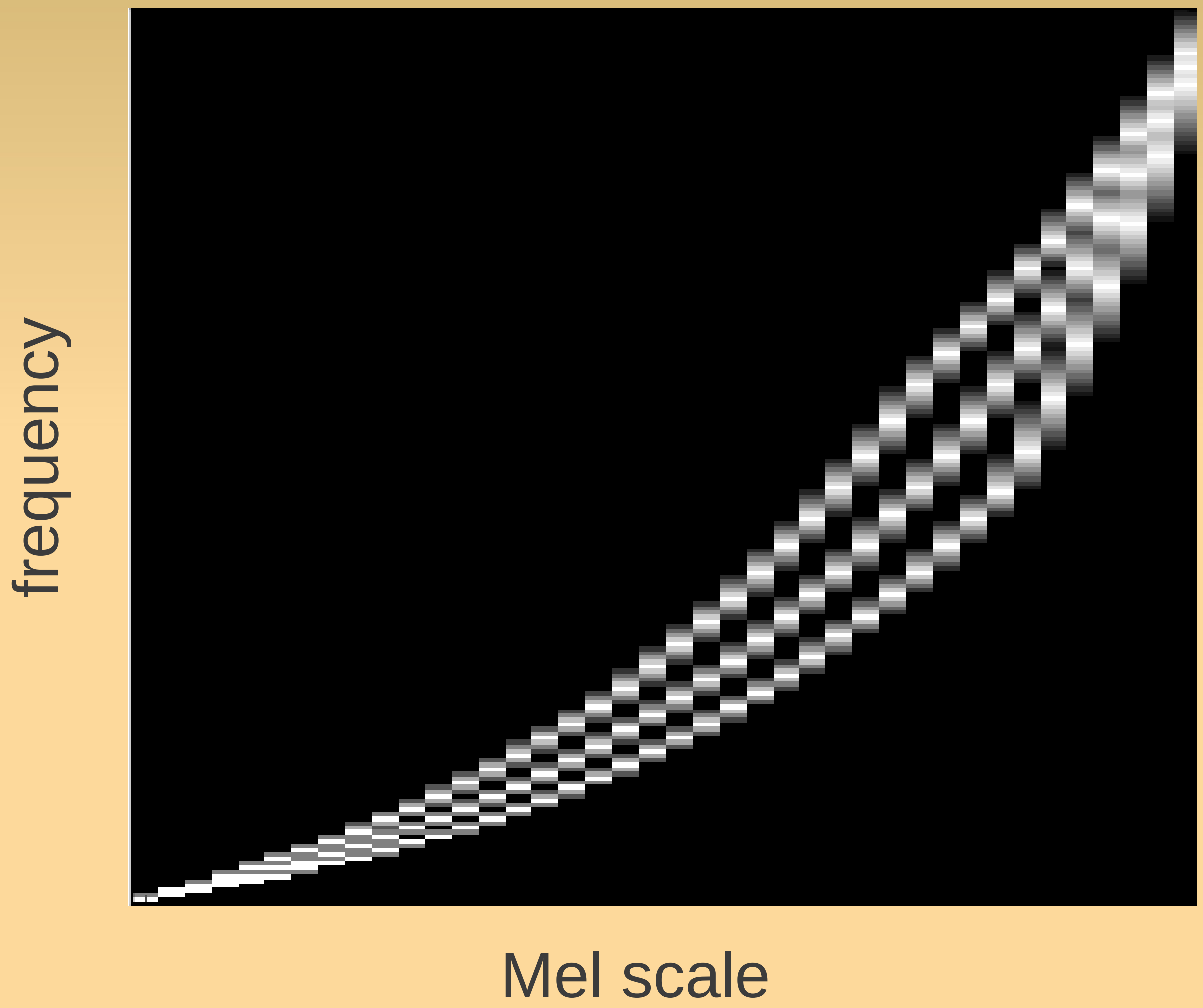
Some comments

- Applying log to the output of the filters on raw spectrograms is very similar to max-pooling, followed by log because intensities in a raw spectrogram vary over many orders of magnitude, and the log is dominated by the maximum intensity frequency

Vocal Tract Length Normalization

- Fixed Pre-processing of spectrograms to remove some degree of speaker variation
  - Parameterized by a warp factor which changes how and where the filters are applied, smoothly.
- Warping can be applied straight to the construction of the filterbanks by changing where the centers of the filters are located

Projection Matrices for different warp factors



Warp factors – 0.8, 1, 1.2

Some VTLN comments

- Requires some amount of training data per speaker to fit the warp factors
- The normalized data ”presumably” is more consistent so a better model can be built focussing on the ”true underlying structure”
- Great for GMMs because it means we can get by with fewer gaussians

- *The data become more speaker independent*

Vocal Tract Length Perturbation

- Instead of building a preprocessing model that makes filterbanks speaker independent, make the model invariant to warp factors
  - Inject the variations into the data
- Strategy well applied on vision tasks to augment databases
  - Transform the data in reasonable ways and add to databases
  - Transformations must preserve classes

Algorithm - Training

```
procedure PERTURBED_FEATURES(lst_spec)
  lst_f ← []
  for each spec ∈ lst_spec
      do { α ← RANDOM_NUMBER_IN_RANGE(0.9, 1.1)
           fb ← FILTERBANKS(α)
           APPEND(lst_f, LOG(fb * spec)) }
  return (lst_f)

main
  while stopping criterion not reached
      do { lst_spec ← LOAD RAW SPECTROGRAMS()
           lst_f ← PERTURBED_FEATURES(lst_spec)
           TRAINMODEL(lst_f) }
```

Use random warp for each utterance in each epoch of training

Algorithm - Testing

procedure SCORES_FOR_DNN-HMM($spec$)

$scores \leftarrow 0$

for each $\alpha \in 0.9 \cdots 1.1$

$$\begin{cases} fb \leftarrow \text{FILTERBANKS}(\alpha) \\ f \leftarrow \text{LOG}(fb * spec) \\ scores \leftarrow scores + \text{COMPUTE-DNN-SCORES}(f) \end{cases}$$

return ($scores$)

Combine posterior probability predictions from multiple warp factors and decode with HMM

Results – Simple Decoding

# of layers	Without VTLP	With VTLP
3	21.9	21.5
4	21.6	20.9
5	21.4	21.3
6	21.0	20.9
7	21.6	20.9

- Trained on TIMIT, warp factors generated with mean 1, stdev 0.1, truncated at 0.9, 1.1
- Simple decoding with warp factor = 1.0

Results - Averaging

# of layers	Without averaging	With averaging
3	21.5	21.1
4	20.9	20.6
5	21.3	21.2
6	20.9	20.2
7	20.9	20.9

- VTLP trained model, without and with averaging at test time (over 5 warp factors 0.95-1.05)

Results – Averaging with non-VTLP models

# of layers	Without Averaging	With Averaging
3	21.9	22.0
4	21.6	21.7
5	21.4	21.8
6	21.0	21.3
7	21.6	21.6

- Model with no warps, without and with averaging at test time (over 5 warp factors 0.95-1.05)

Most Improving phones

ah
dx
eh
aa
d

Future Work

- Explore other variations around the idea of distorting filterbanks
  - Does warping really need to be linear ?
- Explore ideas on how to combine predictions from multiple warp factors, and possibly use that in the training
- Connections to sampling in convolutions
- Large Vocabulary Tasks on larger databases