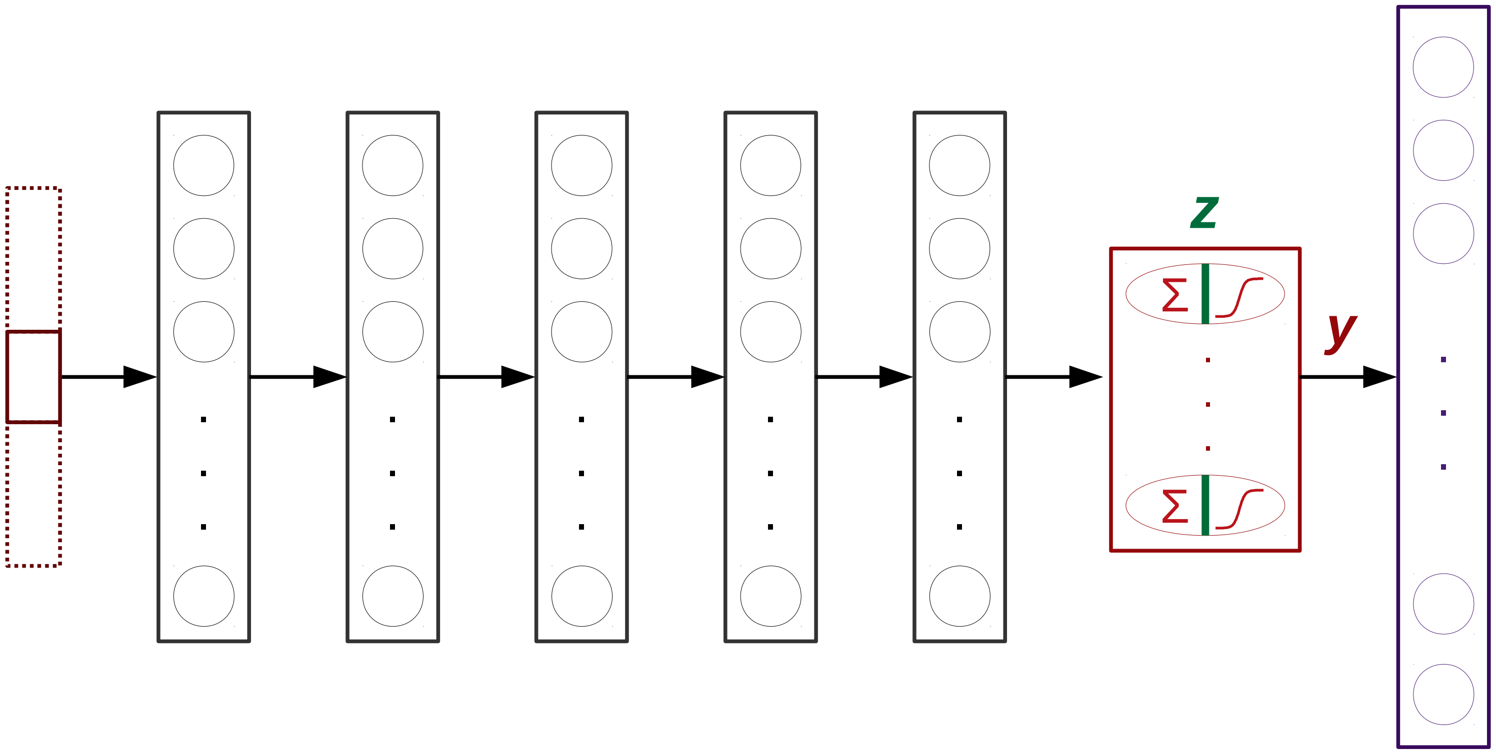
Understanding and Interpreting DNNs for Speech Recognition

Erfan Loweimi, Peter Bell and Steve Renals

Centre for Speech Technology Research (CSTR),
University of Edinburgh

DNNs are GREAT



DNNs are GREAT
BUT are a black box

Submitted to
INTERSPEECH 2019

On Learning Interpretable CNNs with Parametric Modulated Kernel-based Filters

Erfan Loweimi, Peter Bell and Steve Renals

Centre for Speech Technology Research (CSTR), School of Informatics, University of Edinburgh
{e.loweimi, peter.bell, s.renals}@ed.ac.uk

ICASSP2019

ON THE USEFULNESS OF STATISTICAL NORMALISATION OF BOTTLENECK FEATURES FOR SPEECH RECOGNITION

Erfan Loweimi, Peter Bell and Steve Renals

Centre for Speech Technology Research (CSTR), School of Informatics, The University of Edinburgh
{e.loweimi, peter.bell, s.renals}@ed.ac.uk

Outline

- Interpreting DNN's Weights
  - CNNs with interpretable parametric kernel-based filters
  - Submitted to INTERSPEECH 2019

- Interpreting DNN's Activations
  - Statistical properties of (pre-)activations
  - ICASSP 2019

Outline

- Interpreting DNN's Weights
  - CNNs with interpretable parametric kernel-based filters
  - Submitted to INTERSPEECH 2019

- Interpreting DNN's Activations
  - Statistical properties of (pre-)activations
  - ICASSP 2019

Outline -- Part (1)

- Acoustic Modelling from Raw Waveform via SincNet
- CNNs with Parametric Kernel-based Filters
  - Sinc^{2}Net
  - GammaNet
  - GaussNet
- Perceptual/Statistical Studies on Learned Filters

Acoustic Modelling from Raw Waveform via SincNet

SPEECH AND SPEAKER RECOGNITION FROM RAW WAVEFORM WITH SINCNET

*Mirco Ravanelli, Yoshua Bengio**

Mila, Université de Montréal , *CIFAR Fellow

ABSTRACT

Deep neural networks can learn complex and abstract representations, that are progressively obtained by combining simpler ones. A recent trend in speech and speaker recognition consists in discovering these representations starting from raw audio samples directly. Differently from standard hand-crafted features such as MFCCs or FBANK, the raw waveform can potentially help neural networks discover better and more customized representations. The high-dimensional raw inputs, however, can make training significantly more challenging.

This paper summarizes our recent efforts to develop a neural architecture that efficiently processes speech from audio waveforms. In particular, we propose *SincNet*, a novel Convolutional Neural Network (CNN) that encourages the first layer to discover meaningful filters by exploiting parametrized sinc functions. In contrast to standard CNNs, which learn all the elements of each filter, only low and high cutoff frequencies of band-pass filters are directly learned from data. This inductive bias offers a very compact way to derive a customized front-end, that only depends on some parameters with a clear physical meaning.

Our experiments, conducted on both speaker and speech recognition, show that the proposed architecture converges faster, performs better, and is more computationally efficient than standard CNNs.

Index Terms— ASR, CNN, SincNet, Raw samples.

We believe that one of the most critical parts of current waveform-based CNNs is the first convolutional layer. This layer not only deals with high-dimensional inputs, but it is also more affected by van-
ishing gradient pro
architectures. As v
by CNNs often tal
especially when fev
certainly make some
to human intuition,
of the speech signal.
To help the CNN
proposed a novel c
that adds some cons
CNNs, where the fi
rameters (each elem
Net convolves the w
tions that implemen
off frequencies of t
data. This solution s
network to focus on
physical meaning.
In [18] we obta
cation and speaker v
out to outperform st
dard features. Mot
extends our recent fi
recognition experim

$$sinc(x) = \frac{\sin(\pi x)}{\pi x}$$

Interpretable Convolutional Filters with SincNet

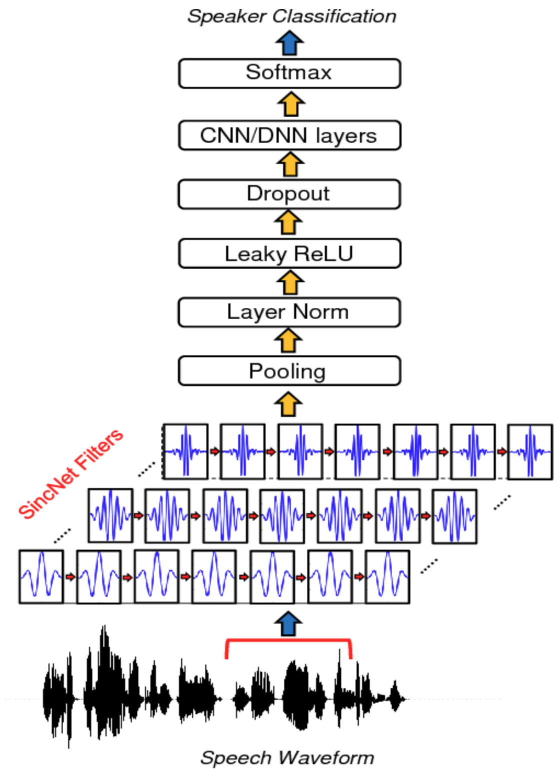
Mirco Ravanelli
Mila, Université de Montréal

Yoshua Bengio
Mila, Université de Montréal
CIFAR Fellow

Abstract

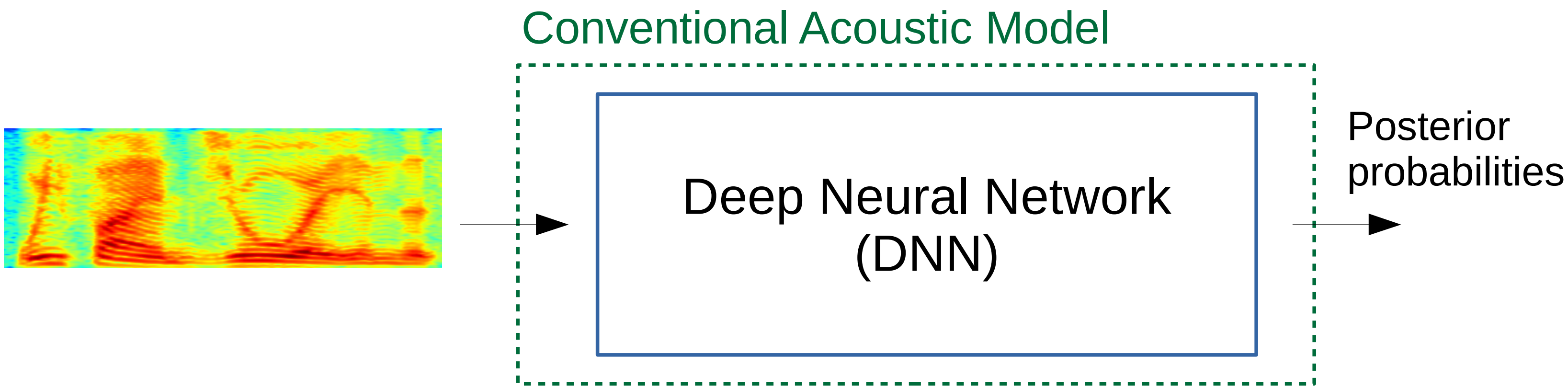
Deep learning is currently playing a crucial role toward higher levels of artificial intelligence. This paradigm allows neural networks to learn complex and abstract representations, that are progressively obtained by combining simpler ones. Nevertheless, the internal "black-box" representations automatically discovered by current neural architectures often suffer from a lack of interpretability, making of primary interest the study of explainable machine learning techniques.

This paper summarizes our recent efforts to develop a more interpretable neural model for directly processing speech from the raw waveform. In particular, we propose *SincNet*, a novel Convolutional Neural Network (CNN) that encourages the first layer to discover more meaningful filters by exploiting parametrized sinc functions. In contrast to standard CNNs, which learn all the elements of each filter, only low and high cutoff frequencies of band-pass filters are directly learned from data. This inductive bias offers a very compact way to derive a customized filter-bank front-end, that only depends on some parameters with a clear physical meaning. Our experiments, conducted on both speaker and speech recognition, show that the proposed architecture converges faster, performs better, and is more interpretable than standard CNNs.



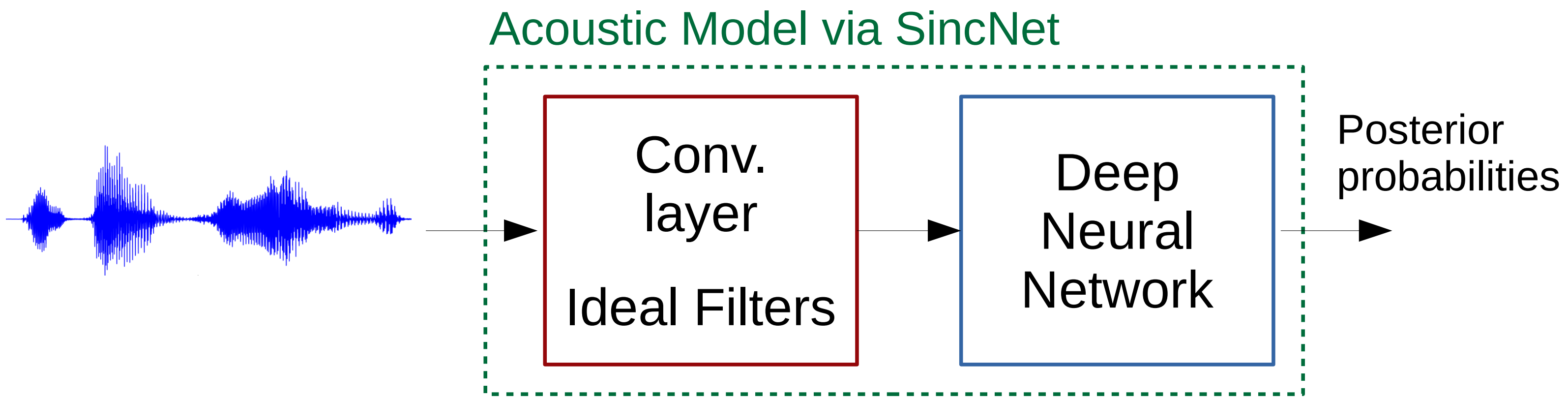
SincNet – Definition

- Convolutional acoustic modelling



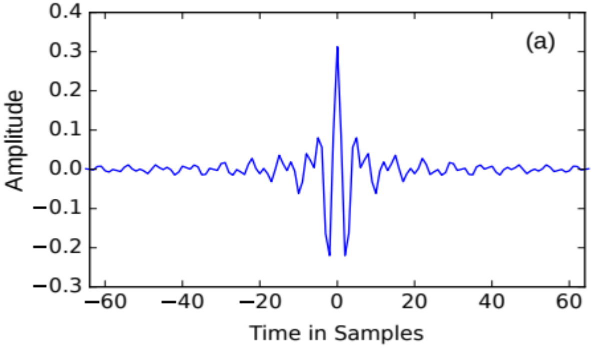
SincNet – Definition

- Convolutional layer with ideal bandpass filters
  - Impulse response ← Sinc



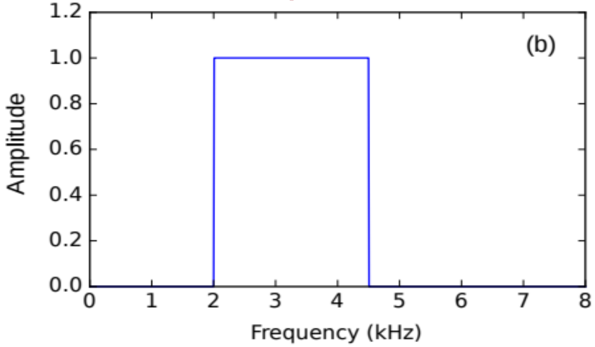
SincNet – Filters Shape

- Impulse and Frequency Responses



Impulse response
(time domain)

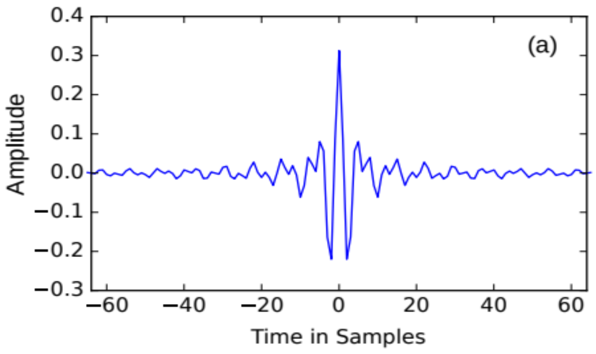
Ideal bandpass filters



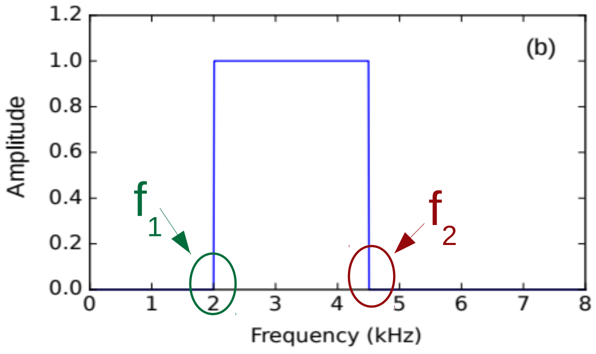
Frequency response
(frequency domain)

SincNet – Filters Shape

- Parameter Set (Θ) → cut-off frequencies: f_1 & f_2



Impulse response
(time domain)



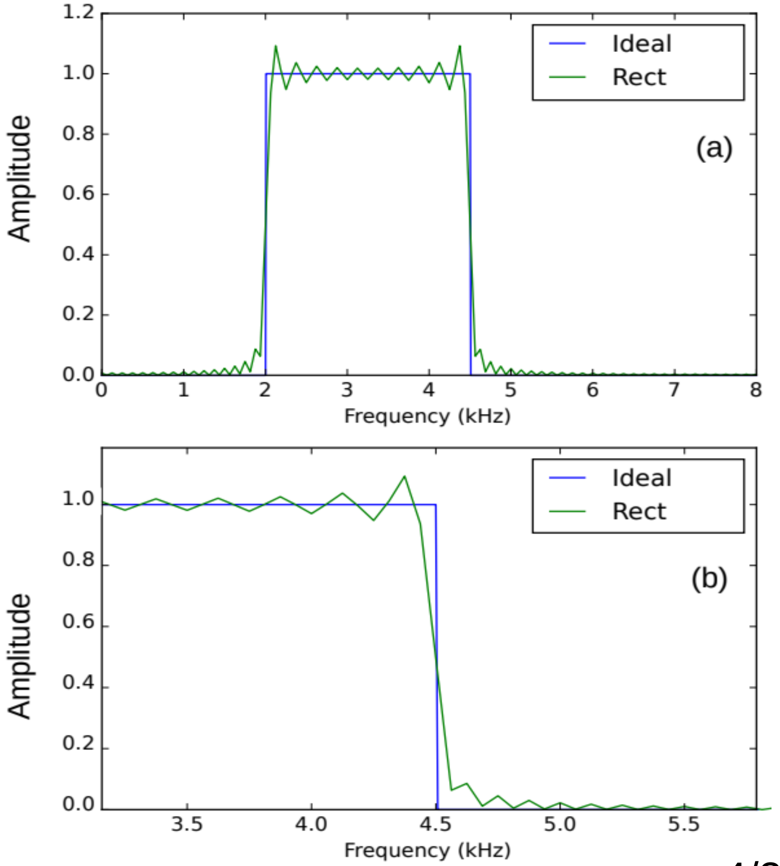
Frequency response
(frequency domain)

SincNet
Practical Considerations

Loweimi et al

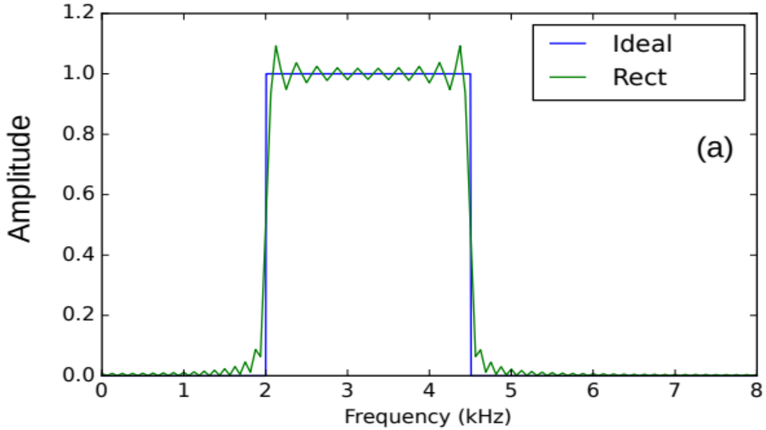
SincNet – Practical Considerations

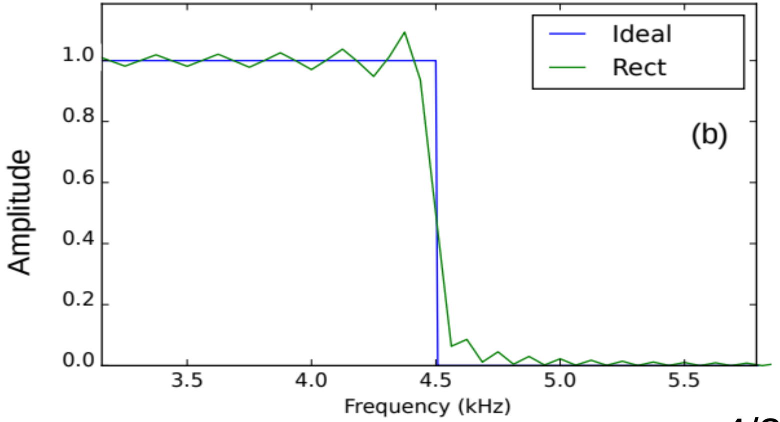
- Sinc length is finite



SincNet – Practical Considerations

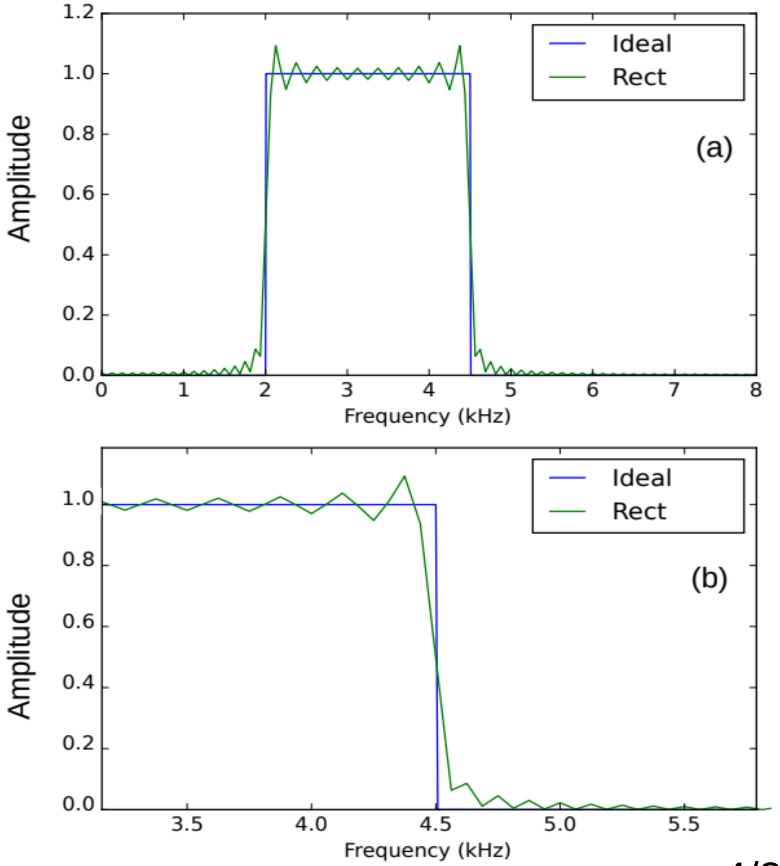
- Sinc length is finite
  - Rectangular windowing





SincNet – Practical Considerations

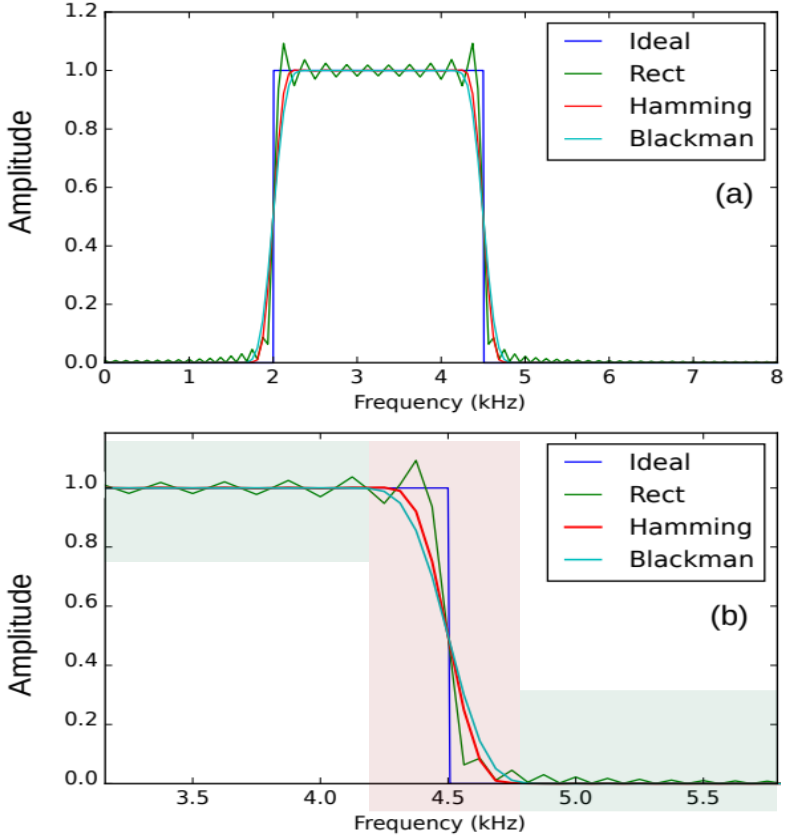
- Sinc length is finite
  - Rectangular windowing
    - Ripples



SincNet – Practical Considerations

- Sinc length is finite
  - Rectangular windowing
  - Solution:
    - Apply a tapered window

$$h(t;\theta^{(i)}) \leftarrow h(t;\theta^{(i)})\ window(t)$$



SincNet – Practical Considerations

- Sinc length is finite
  - Solution: Apply a tapered window

- Monitor the cut-off frequencies value
  - f_1 & f_2 $\rightarrow$ should be positive
  - $f_2 <$ Nyquist Rate

$$f_1 \leftarrow |f_1|$$

$$f_2 \leftarrow f_1 + |f_2 - f_1|$$

SincNet – Practical Considerations

- Sinc length is finite
  - Apply a tapered window
- Monitor the cut-off frequencies value
- Amplitude learning is not necessary
  - Higher layer’s weights → almost play the gain role

SincNet – Practical Considerations

- Sinc length is finite
  - Apply a tapered window
- Monitor the cut-off frequencies value
- Amplitude learning is not necessary
- Initialisation of Parameters (cut-off frequencies)
  - Any perceptual scale may be used, e.g. mel

CNN vs SincNet

CNN
impulse responses

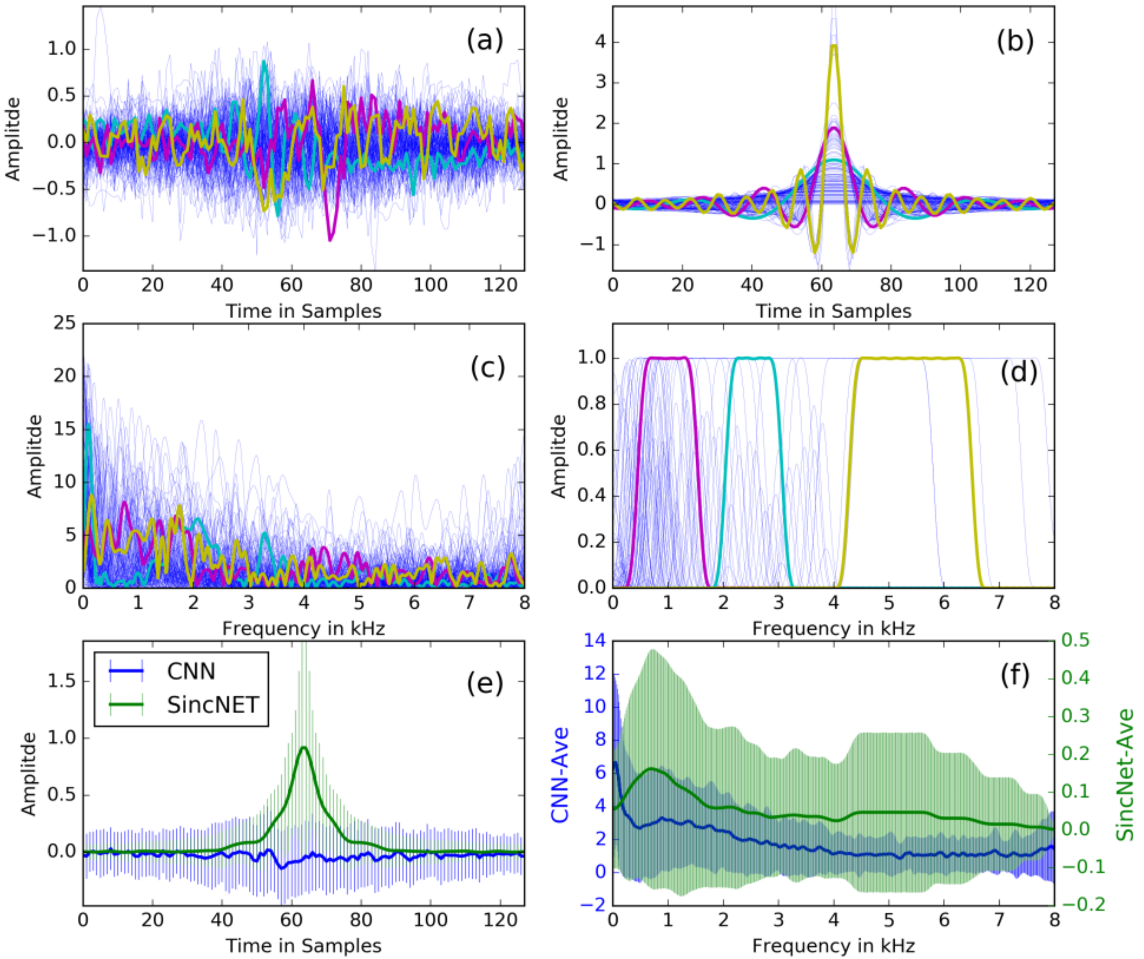
SincNet
impulse responses

CNN
Frequency responses

SincNet
Frequency responses

Average
impulse responses

Average
Frequency responses



SincNet vs CNN -- Advantages

- Parametric vs Non-parametric

SincNet vs CNN -- Advantages

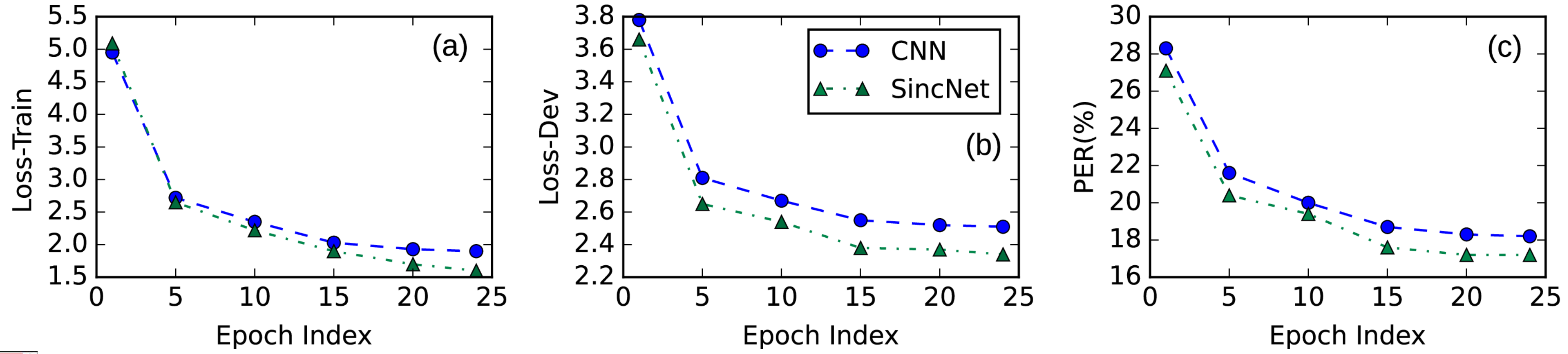
- Parametric vs Non-parametric
- Parametric model
  - More Interpretable
  - Strong constraint on hypothesis space

SincNet vs CNN -- Advantages

- Parametric vs Non-parametric
- Parametric model
  - More interpretable
  - Strong constraint on hypothesis space
    - Regularisation/Generalisation/Robustness
    - Fewer parameters
      - Less training data required
      - Faster learning/convergence

SincNet vs CNN -- Advantages

- Parametric vs Non-parametric
- Better Performance on TIMIT:
  - Lower Loss, Classification Error and PER

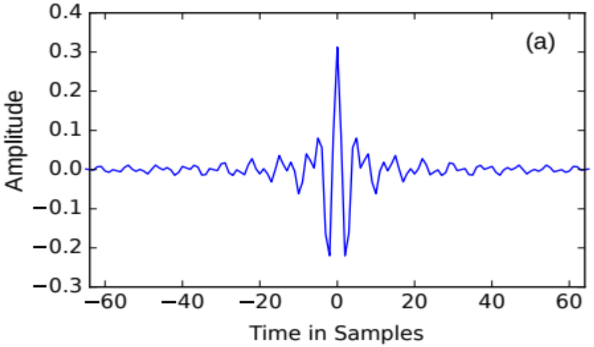


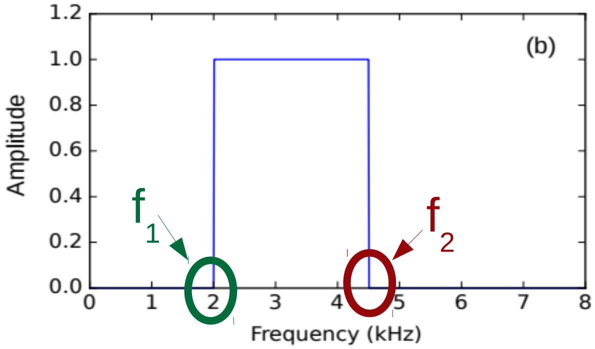
Loweimi et al

General Formulation for Interpretable Kernel-based CNNs

Interpretable Kernel-based CNNs

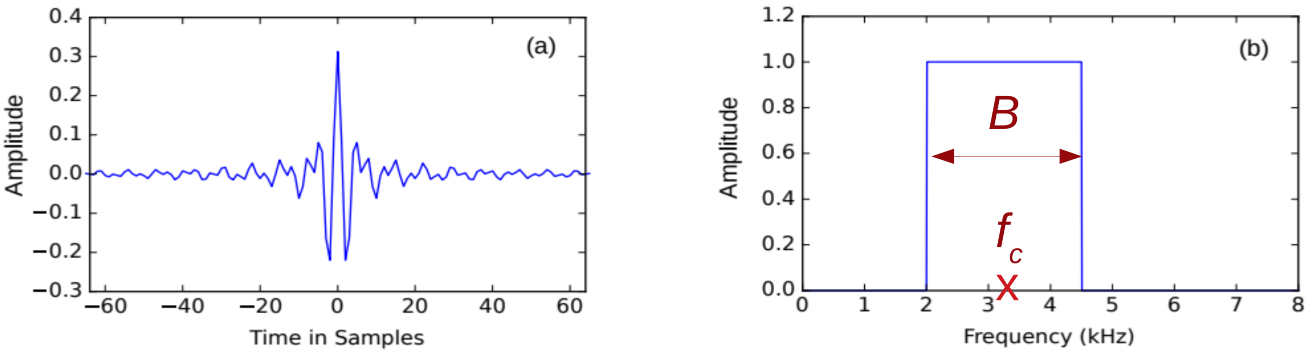
$$h(t;\theta^{(i)}) = 2f_2^{(i)} sinc(2f_2^{(i)}t) - 2f_1^{(i)} sinc(2f_1^{(i)}t)$$





Interpretable Kernel-based CNNs

$$h^{(i)}(t) = 2B^{(i)} sinc(B^{(i)} t) \quad \cos(2\pi f_c^{(i)} t)$$



Interpretable Kernel-based CNNs

$$h^{(i)}(t) = \boxed{2B^{(i)} sinc(B^{(i)}t)}\ \boxed{\cos(2\pi f_c^{(i)} t)}$$

General Formulation of Interpretable Kernel-based Filters

Kernel | Modulated Carrier

$$h^{(i)}(t) = \boxed{2B^{(i)} sinc(B^{(i)} t)}\ \boxed{\cos(2\pi f_c^{(i)} t)}$$

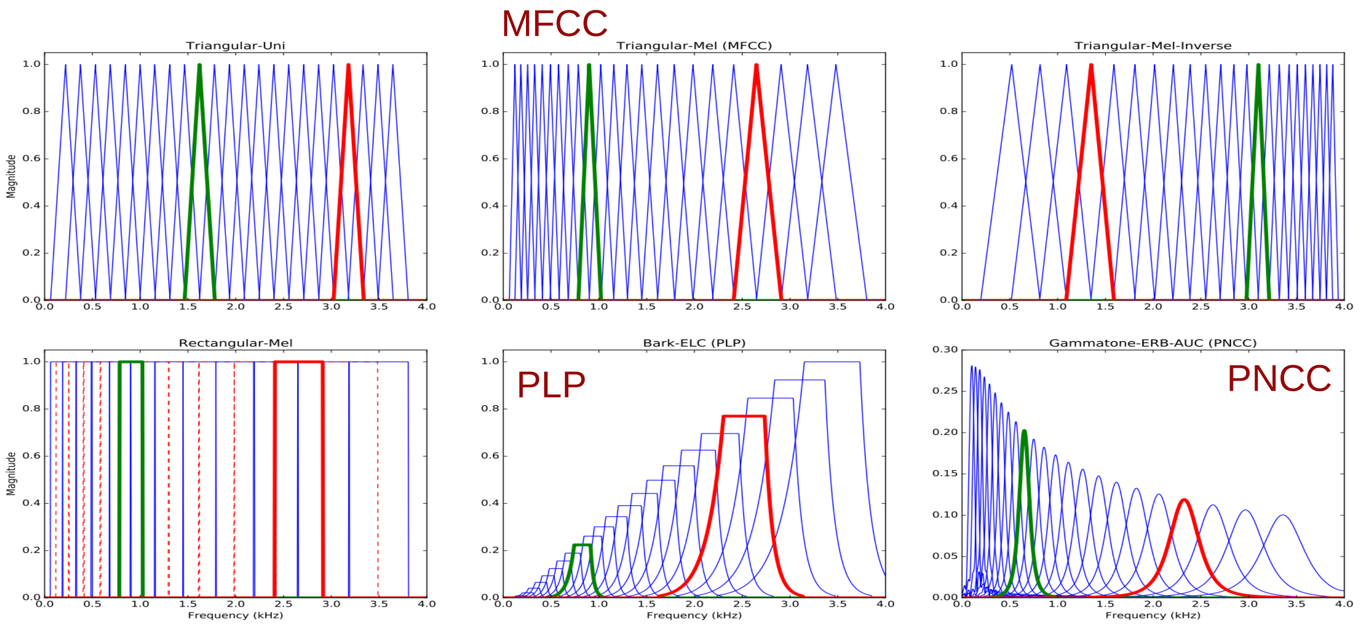
$$h^{(i)}(t;\theta^{(i)}, f_c^{(i)}) = \boxed{K(t;\theta^{(i)})}\ \boxed{carrier(t; f_c^{(i)})}$$

General Formulation of Interpretable Kernel-based Filters

$$h^{(i)}(t;\theta^{(i)},f_c^{(i)}) = \underbrace{\boxed{K(t;\theta^{(i)})}}_{\text{Kernel}} \; \underbrace{\boxed{carrier(t;f_c^{(i)})}}_{\text{Modulated Carrier}}$$

$$\text{Parameter Set: } \Theta = \{\theta^{(i)}, f_c^{(i)}\}$$

Learning Kernel-based Filterbanks



Loweimi et al

Learning Kernel-based Filterbanks

– Sinc^{2}Net

– GammaNet

– GaussNet

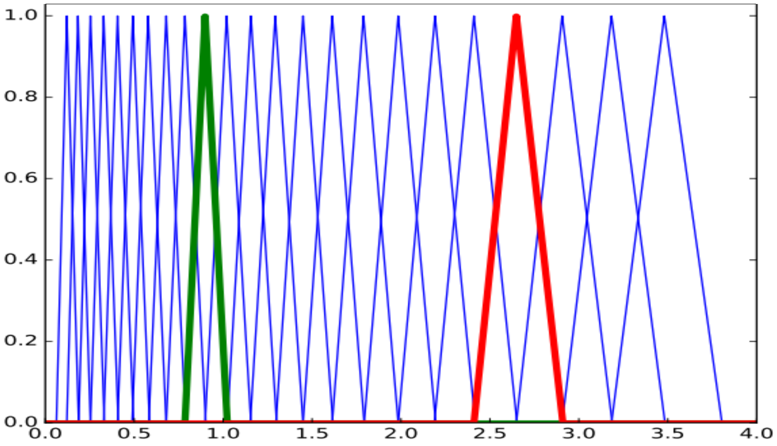
Sinc^{2}Net: Triangular Filters

- Widely used in Speech processing → MFCC
  - Perceptually more plausible than rectangular filters

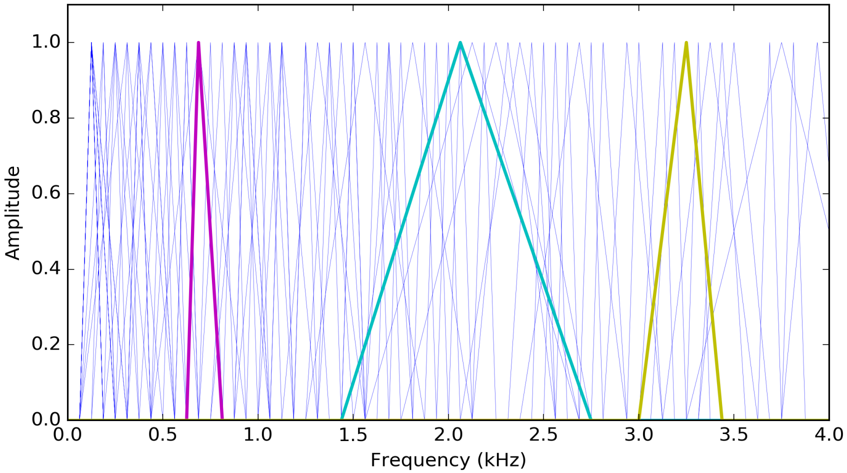
Sinc^{2}Net: Triangular Filters

- Widely used in Speech processing → MFCC
  - Perceptually more plausible than rectangular filters

Engineered



Learned

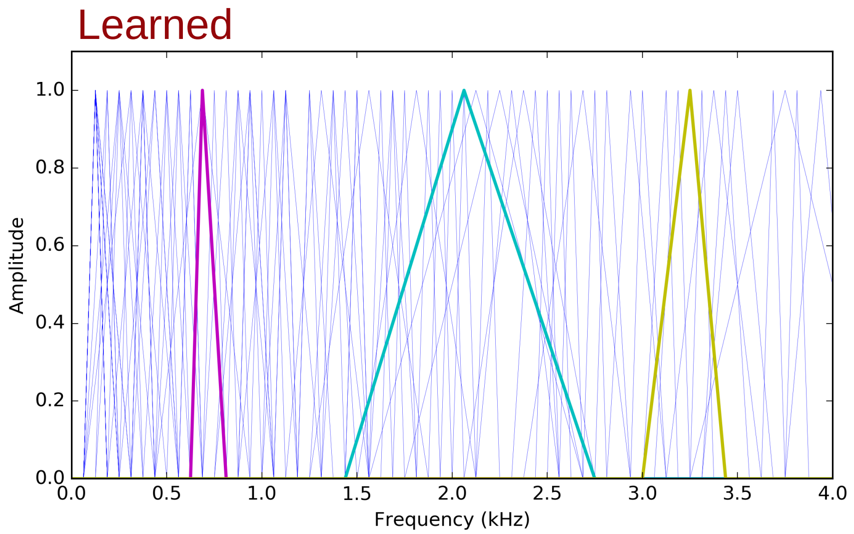


Sinc²Net: Triangular Filters

- Widely used in Speech processing → MFCC
  - Perceptually more plausible than rectangular filters

$$K(t;\theta^{(i)}) = A^{(i)} sinc^2(B^{(i)}t)$$

$$\theta^{(i)} = \{A^{(i)}, B^{(i)}\}$$



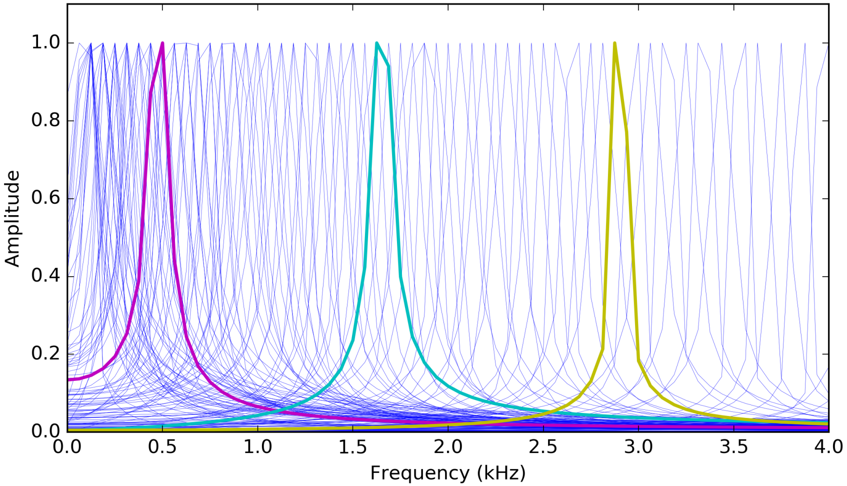
GammaNet: Gammatone Filters

- Even more biologically plausible
  - Describes impulse response of auditory filters in Cochlea

$$K(t;\theta^{(i)}) = A^{(i)} t^{(N^{(i)}-1)} e^{-2\pi B^{(i)} t}$$

$$\theta^{(i)} = \{A^{(i)}, B^{(i)}, \boxed{N^{(i)}}\}$$

Typical value: 4

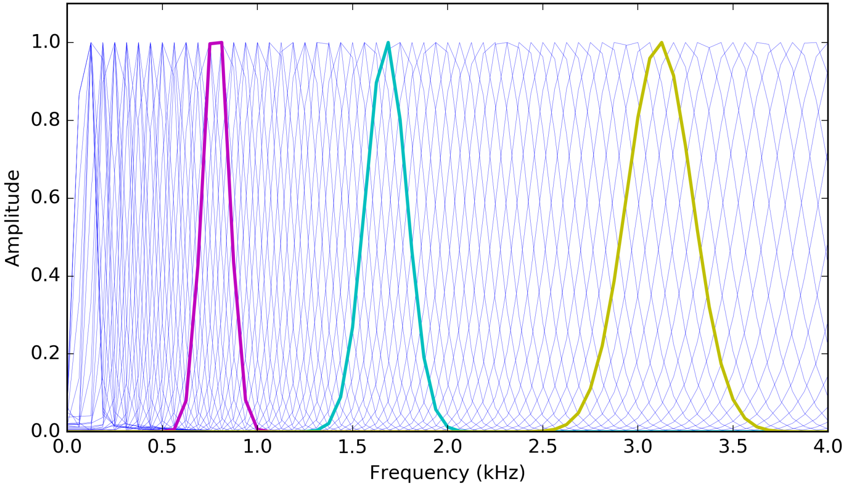


GaussNet: Gaussian Filters

- Bell-shaped Filters

$$K(t;\theta^{(i)}) = A^{(i)} \exp(-t^2/\sigma_i^2)$$

$$\theta^{(i)} = \{A^{(i)}, \sigma^{(i)}\}$$



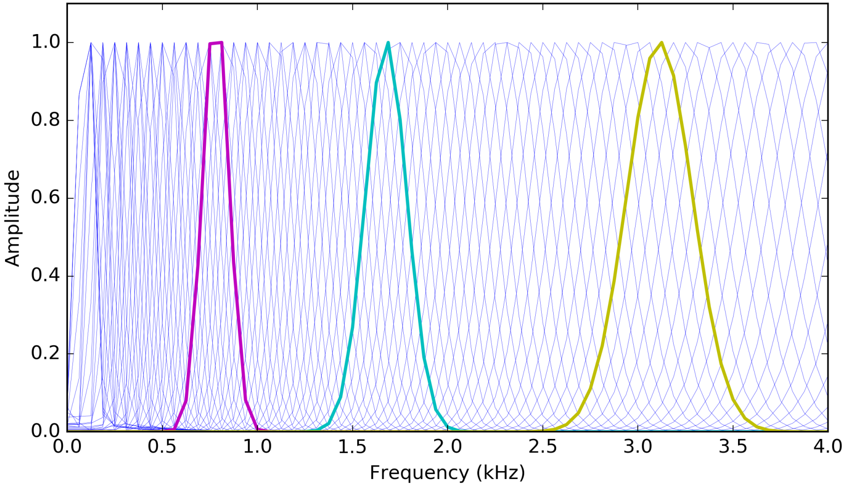
GaussNet: Gaussian Filters

- Bell-shaped Filters

$$K(t;\theta^{(i)}) = A^{(i)} \exp(-t^2/\sigma_i^2)$$

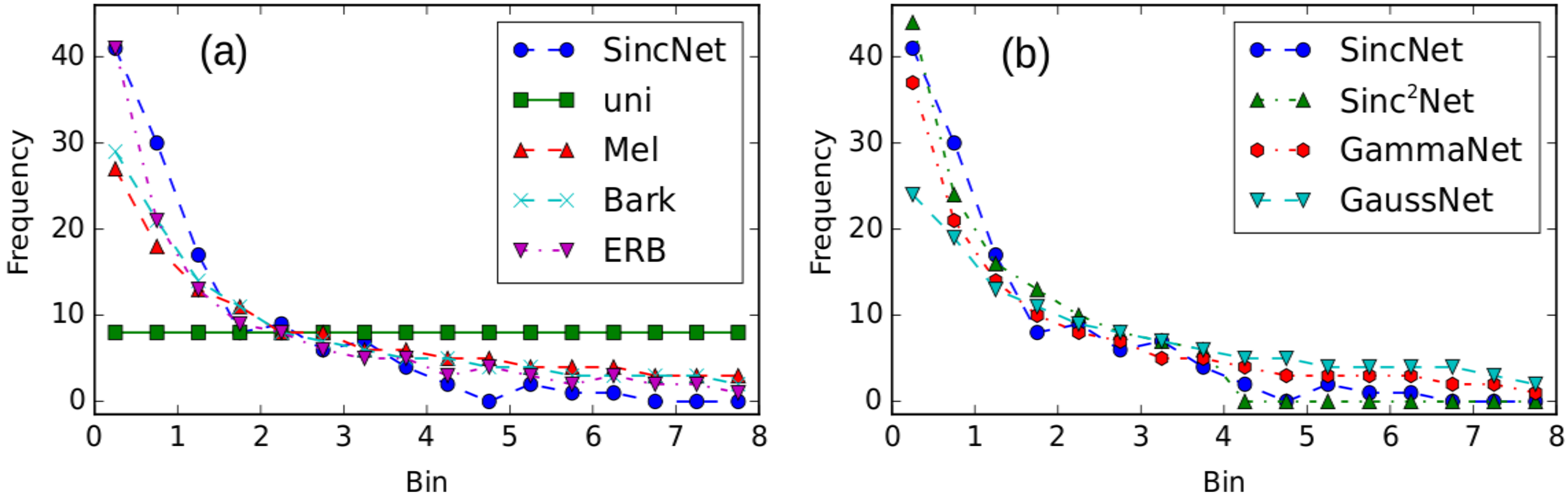
$$\sigma_i = \frac{\sqrt{log2}}{2\pi B_i}$$

3 dB bandwidth
(Hz) of the i^{th} filter

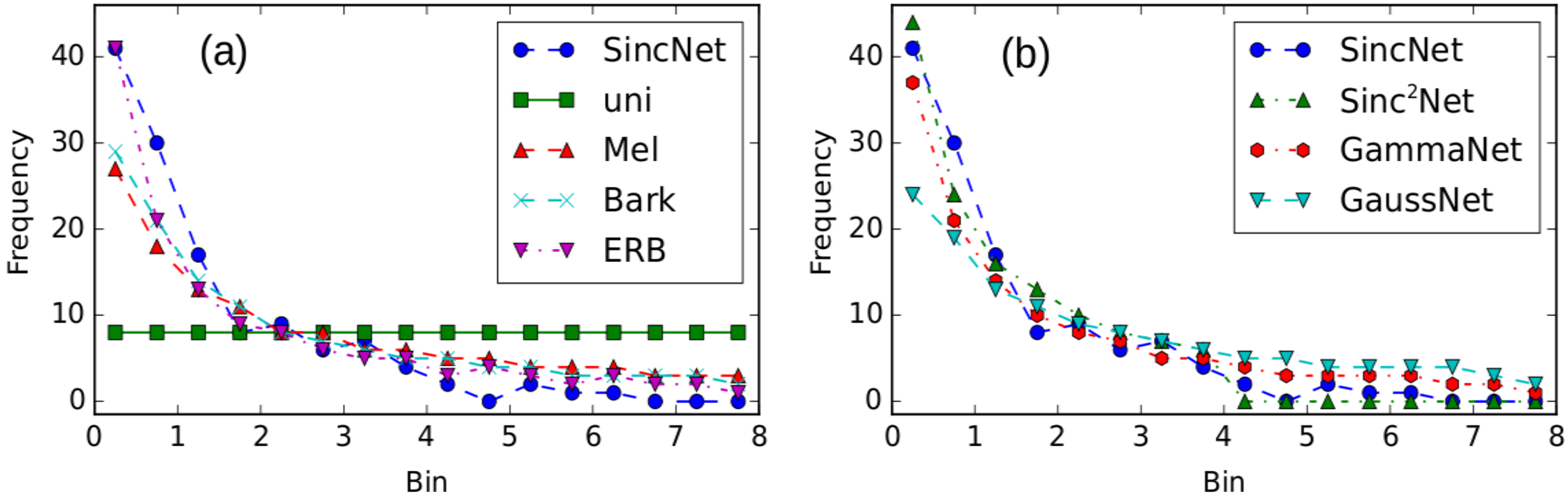


Perceptual and Statistical Studies

Filters' Centre Frequency Distribution

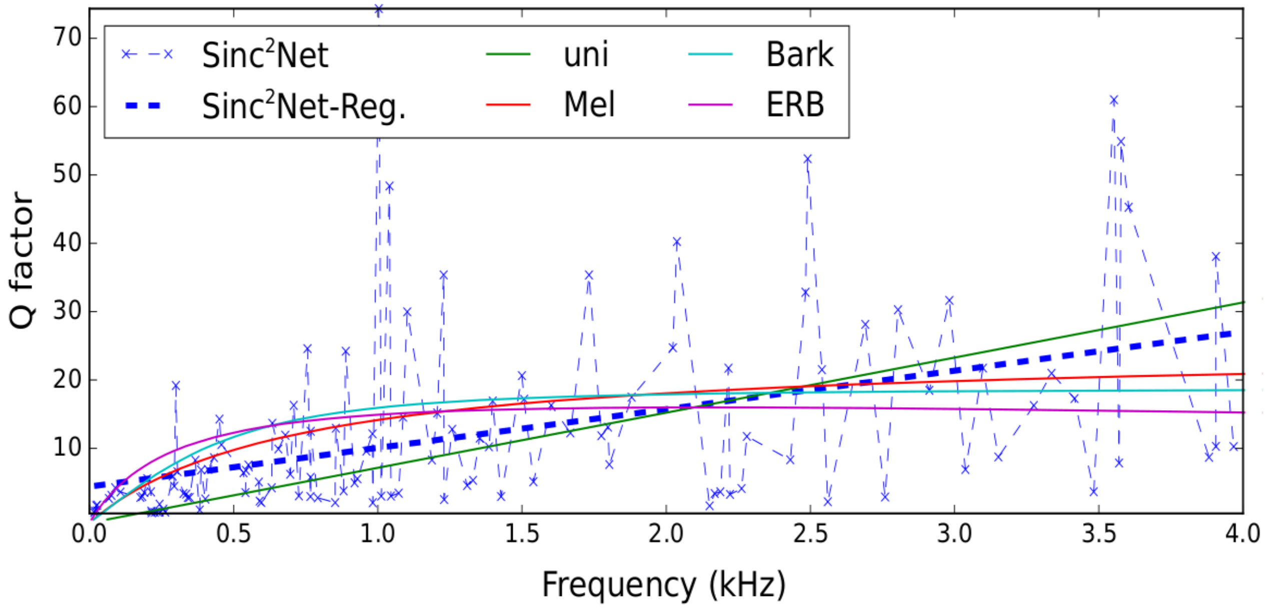


Filters' Centre Frequency Distribution

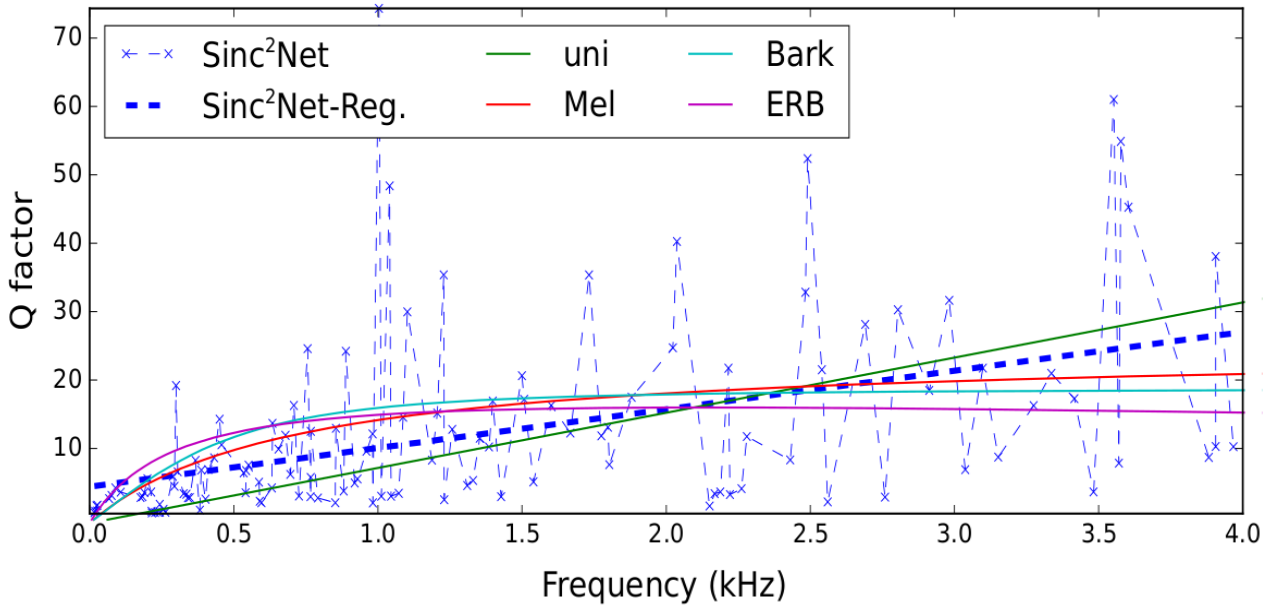


Higher filter concentration at low frequencies (< 2kHz).

Quality Factor (Q) of the Filters

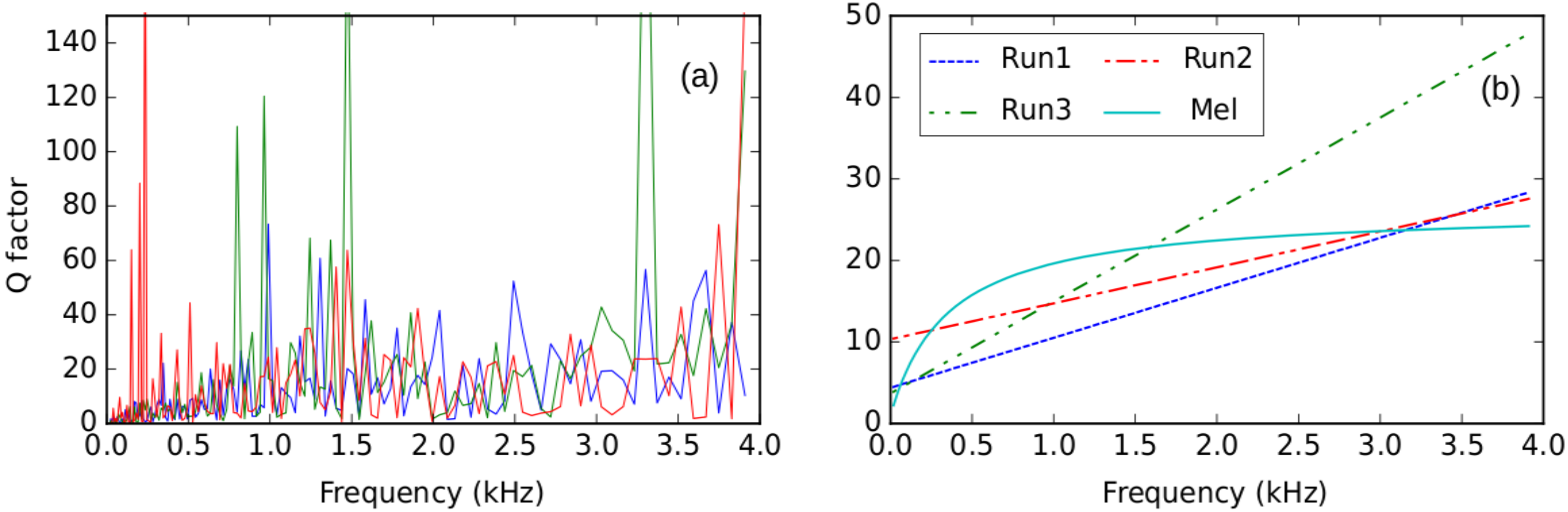


Quality Factor (Q) of the Filters



Similar trend to the perceptual measures.

Quality Factor (Q) of the Filters



It is not a random effect ...

Gammatone Filters Order
Perceptual vs Learned

Table 1: *Statistics of the GammaNet learned filters order.*

	Mean	Median	Std	Min	Max
GammaNet	4.39	4.30	0.97	1.73	6.80

– No constraint was imposed on filters order during training

Gammatone Filters Order
Perceptual vs Learned

Table 1: *Statistics of the GammaNet learned filters order.*

	Mean	Median	Std	Min	Max
GammaNet	4.39	4.30	0.97	1.73	6.80

– Matches with perceptual studies on human auditory system.

Gammatone Filters Order
Perceptual vs Learned

Table 1: *Statistics of the GammaNet learned filters order.*

	Mean	Median	Std	Min	Max
GammaNet	4.39	4.30	0.97	1.73	6.80

AN EFFICIENT AUDITORY FILTERBANK BASED ON THE GAMMATONE FUNCTION

Roy Patterson and Ian Nimmo-Smith

MRC Applied Psychology Unit
15 Chaucer Road
Cambridge CB2 2EF

John Holdsworth and Peter Rice

Cambridge Electronic Design
Science Park
Milton Road
Cambridge

December 1987

Page - 7 -

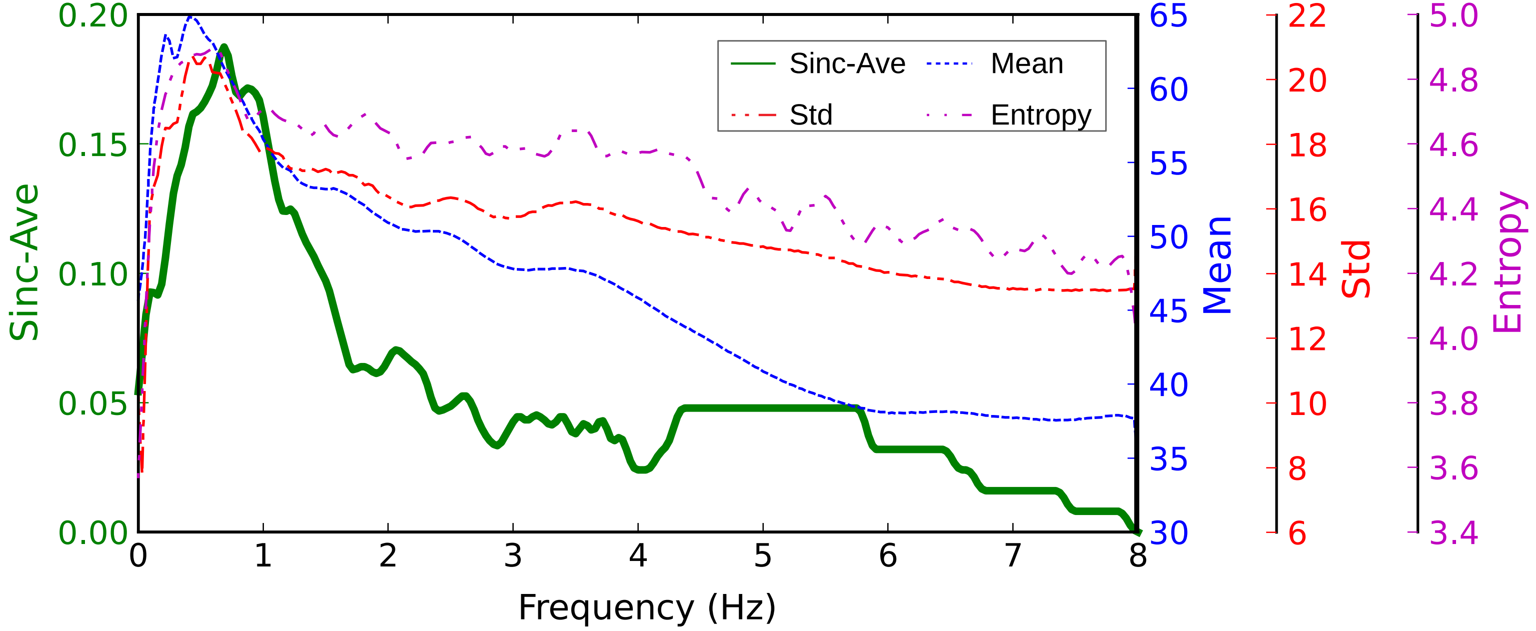
A. **A Comparison of Roex and Gammatone Amplitude Spectra**

Schofield (1985) has recently demonstrated that a gammatone filter with order 4 provides a good fit to the average auditory filters presented in Patterson (1976).

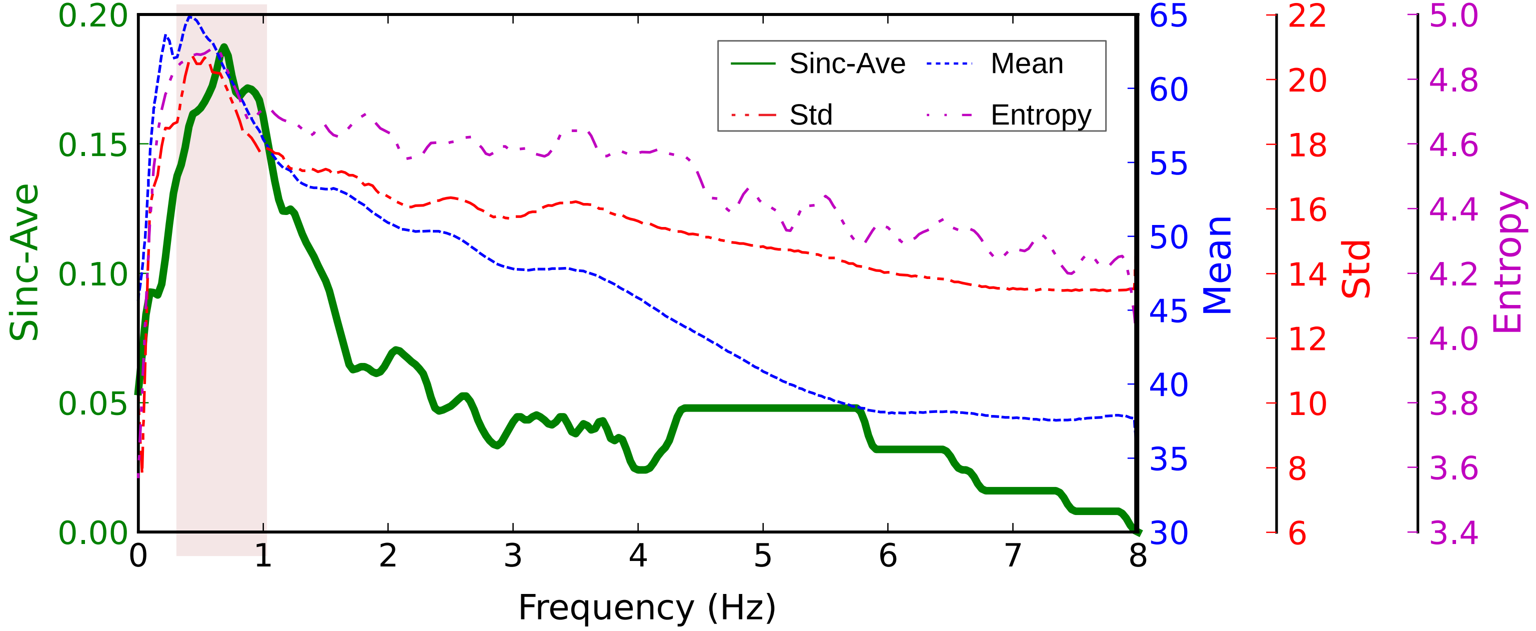
Schofield, D. (1985). Visualisations of speech based on a model of the peripheral auditory system. NPL Report DITC 62/85.

Statistical Properties of the Data and the Learned Filters

Statistical Properties of the Data and the Learned Filters



Statistical Properties of the Data and the Learned Filters



Argmax Entropy ≈ Argmax Std ≈ Argmax Ave Filter Mag.

Experimental Results

Experimental Results – Setup

- Task: TIMIT phone recognition
- Tools: Kaldi + PyTorch-Kaldi
- Frame length: 200 ms, frame shift: 10 ms
- Optimisation:
  - 24 Epochs, RMSprop
- Architecture:
  - Convolutional layer + MLP + output layer
    - MLP → 5 hidden layers, 1024nodel, ReLU

Experimental Results – PER

Table 2: *TIMIT PER for different kernels (200 ms).*

	MLP	CNN	Sinc	Sinc^2	Gamma	Gauss
PER	18.5	18.2	17.6	16.9	17.2	17.0

Experimental Results – PER

Table 2: *TIMIT PER for different kernels (200 ms).*

	MLP	CNN	Sinc	Sinc^2	Gamma	Gauss
PER	18.5	18.2	17.6	16.9	17.2	17.0

log-Filterbank

Raw Waveform models

X-Net

Experimental Results – PER

Table 2: *TIMIT PER for different kernels (200 ms).*

	MLP	CNN	Sinc	Sinc^2	Gamma	Gauss
PER	18.5	18.2	17.6	16.9	17.2	17.0

(1) Raw waveform models outperform log-Filterbank features

(2) Parametric X-Nets outperform non-parametric CNN

(3) X-Nets outperform SincNet (also more biologically plausible)

Optimal Frame Length: 200 ms

Table 3: *TIMIT PER for different frame lengths (ms).*

	25	50	100	200	300	400
CNN	30.0	21.7	18.8	18.2	18.6	19.0
SincNet	27.7	20.6	17.6	17.4	17.6	17.7
Sinc^{2}Net	27.1	20.7	17.3	16.9	17.4	17.7

Optimal Frame Length: 200 ms

Table 3: *TIMIT PER for different frame lengths (ms).*

	25	50	100	200	300	400
CNN	30.0	21.7	18.8	18.2	18.6	19.0
SincNet	27.7	20.6	17.6	17.4	17.6	17.7
Sinc^{2}Net	27.1	20.7	17.3	16.9	17.4	17.7

(1) Pros/Cons

(2) Why?

Optimal Frame Length: 200 ms
Pros/Cons

Table 3: *TIMIT PER for different frame lengths (ms).*

	25	50	100	200	300	400
CNN	30.0	21.7	18.8	18.2	18.6	19.0
SincNet	27.7	20.6	17.6	17.4	17.6	17.7
Sinc^2Net	27.1	20.7	17.3	16.9	17.4	17.7

- ✓ Suppressing harmful mid-term properties (speaker-ind. ASR)
- ✓ Preserving useful mid-term properties (speaker ID)
- ✗ More memory is required

Optimal Frame Length: 200 ms
WHY?

Table 3: *TIMIT PER for different frame lengths (ms).*

	25	50	100	200	300	400
CNN	30.0	21.7	18.8	18.2	18.6	19.0
SincNet	27.7	20.6	17.6	17.4	17.6	17.7
Sinc^{2}Net	27.1	20.7	17.3	16.9	17.4	17.7

Optimal Frame Length: 200 ms
WHY?

Table 3: *TIMIT PER for different frame lengths (ms).*

	25	50	100	200	300	400
CNN	30.0	21.7	18.8	18.2	18.6	19.0
SincNet	27.7	20.6	17.6	17.4	17.6	17.7
Sinc^{2}Net	27.1	20.7	17.3	16.9	17.4	17.7

1. Learning Temporal Masking
– Optimal combination of masker and maskee

2. Optimal syllable modelling
– Mean Syllable length in English is 200 ms (Greenberg et al, 1999)

Conclusions -- Part (1)

- Task: waveform modelling through convolutional layer
- General Formulation for interpretable CNNs with kernel-based filters was derived
  - Sinc^{2}Net, GammaNet and GaussNet were studied
- Learned filters studied statistically and perceptually
- Mid-term (200ms) processing is required for raw waveform modelling through X-Nets

Outline -- Part (2)

- Interpreting DNN’s Weights
  - CNNs with parametric kernel-based filters
  - Submitted to INTERSPEECH 2019

- Interpreting DNN’s Activations
  - Statistical Properties of (Pre-)Activations
  - ICASSP 2019

Outline -- Part (2)

- Statistical Study on (Pre-)Activations
  - Analytically and Empirically
- (Re)-Explaining some observations ...
  - Why pre-activations, NOT activations, should be used as Bottleneck feature for HMM-GMM ASR?
  - Why does ReLU give rise to sparsity?
- Statistical Normalisation of the Bottleneck Features for ASR

Come to our Poster for more ...

ON THE USEFULNESS OF STATISTICAL NORMALISATION OF BOTTLENECK FEATURES FOR SPEECH RECOGNITION

Erfan Loweimi, Peter Bell and Steve Renals

Centre for Speech Technology Research (CSTR), School of Informatics, The University of Edinburgh
{e.loweimi, peter.bell, s.renals}@ed.ac.uk

ICASSP 2019

Poster Session: MLSP-P17.11

Time: May 17, 13:30 - 15:30

That's It!

- Thanks for Your Attention
- Q/A

- Acknowledgements:
  - Supported by EPSRC Project EP/R012180/1 (*SpeechWave*)