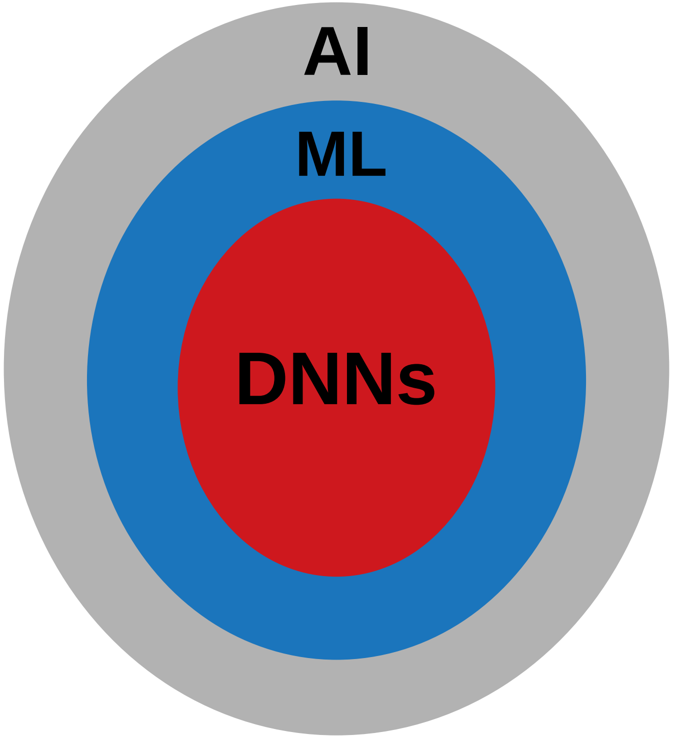
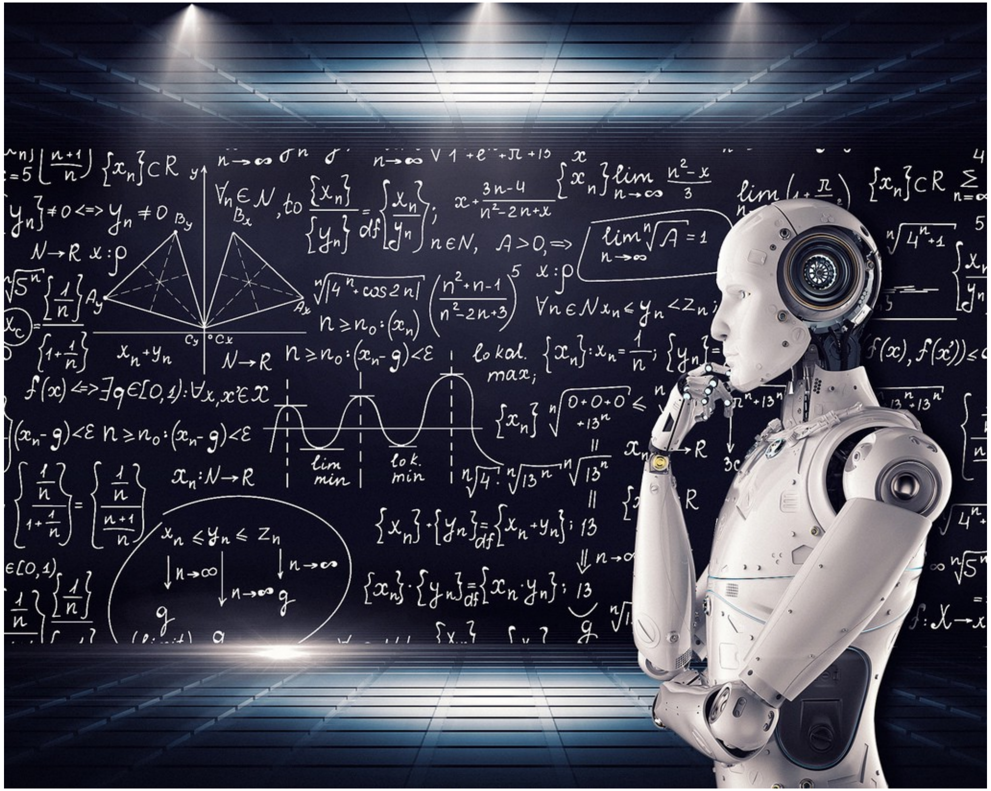
Understanding and Interpreting DNNs for Speech Recognition

Erfan Loweimi, Peter Bell and Steve Renals

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

DNNs are GREAT ...

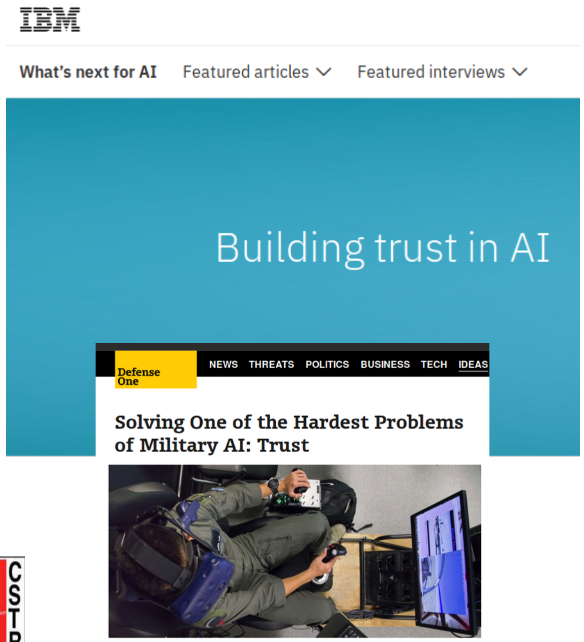


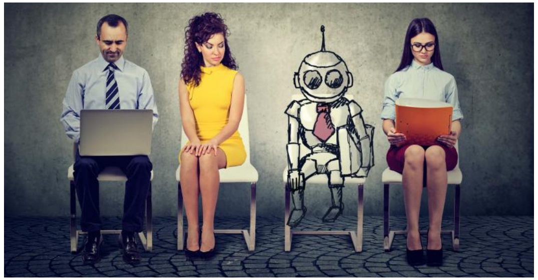


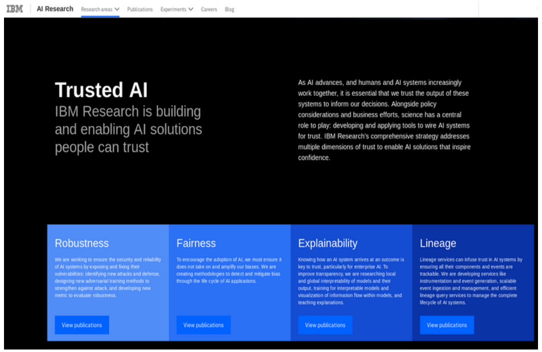
DNNs are GREAT ...
BUT are a **black box**

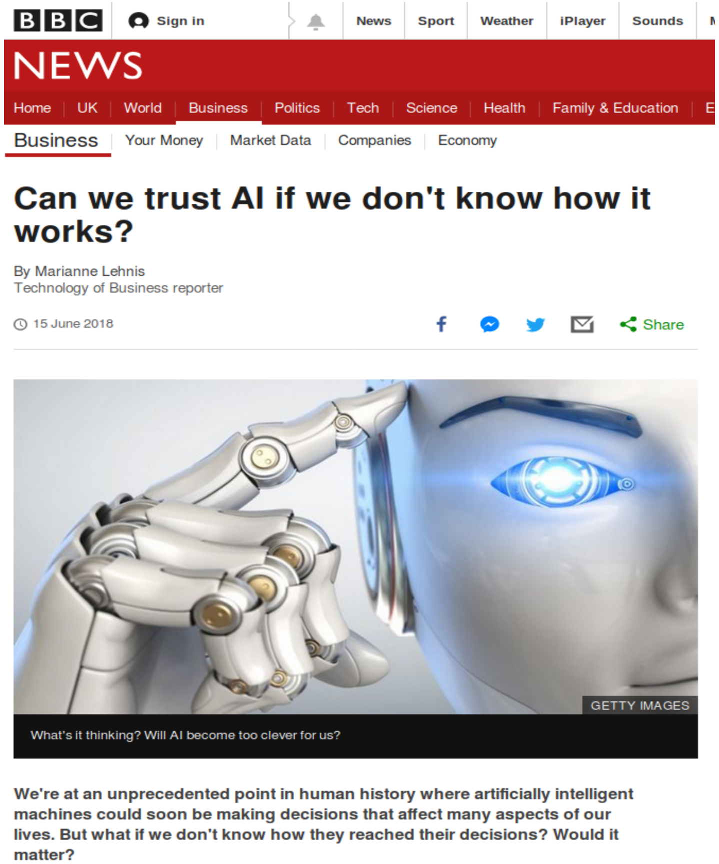
Importance of Understanding

- Trust









BBC | Sign in | News | Sport | Weather | iPlayer | Sounds

NEWS

Home | UK | World | Business | Politics | Tech | Science | Health | Family & Education

Business | Your Money | Market Data | Companies | Economy

Can we trust AI if we don't know how it works?

By Marianne Lehnis
Technology of Business reporter

15 June 2018

Share

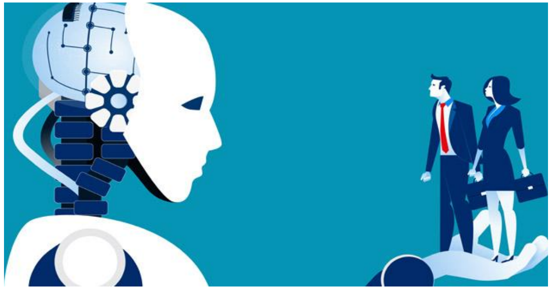
GETTY IMAGES

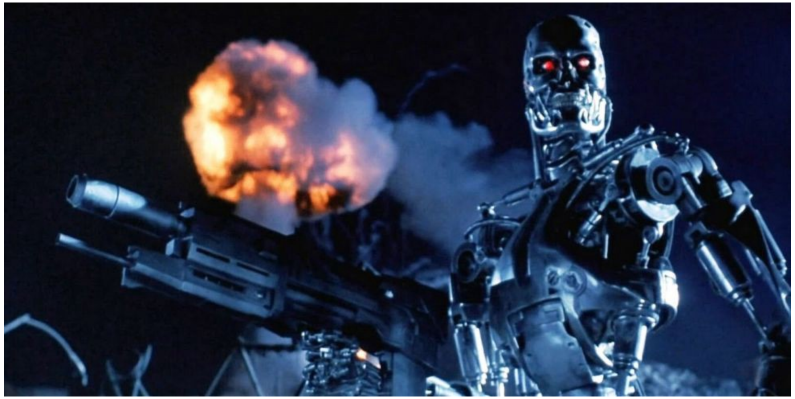
What's it thinking? Will AI become too clever for us?

We're at an unprecedented point in human history where artificially intelligent machines could soon be making decisions that affect many aspects of our lives. But what if we don't know how they reached their decisions? Would it matter?

Importance of Understanding

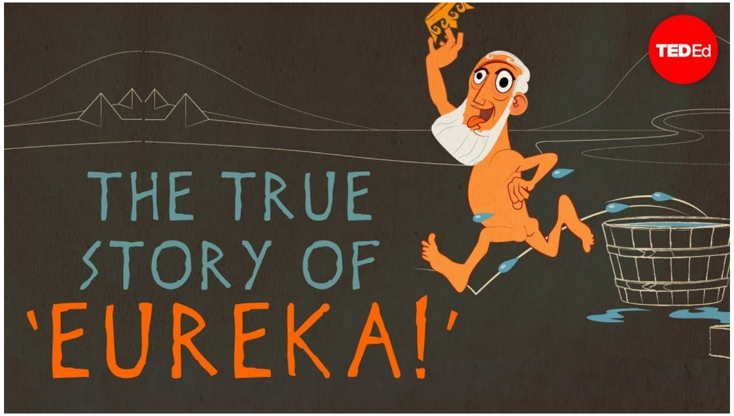
- Control

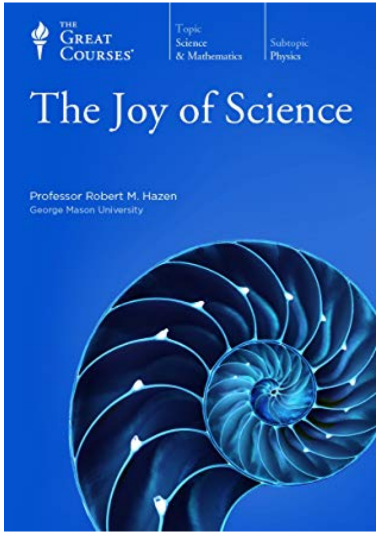




Importance of Understanding

- The Joy of Science
- Better practice





SpeechWave

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

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

Outline

- PART I
  - Interpreting DNN's Activations
    - ICASSP 2019
- PART II
  - Interpreting DNN's Weights
    - Submitted to INTERSPEECH 2019

Outline

- **PART I**
  - Interpreting DNN's Activations
    - ICASSP 2019
- PART II
  - Interpreting DNN's Weights
    - Submitted to INTERSPEECH 2019

“I have always thought the actions of men the best interpreters of their thoughts.”

— John Locke

“I have always thought the activations of NNs the best interpreters of their thoughts.”

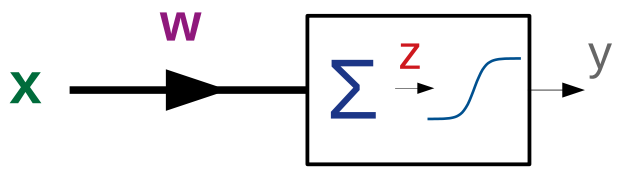
– Unknown :-)

Part I Outline

- Conduct a series of statistical studies on activations
- (Re)-Explaining some observations
- Statistical Normalisation of bottleneck features for ASR

DNNs from Statistical Standpoint

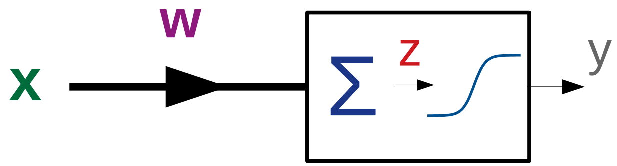
- Effect of the activation function *f*(.) on the density ...
  - **x**: input   **w**: weights
  - z: pre-activation   y: activation



$$\mathrm{y} = \mathrm{f}(\underbrace{\mathbf{w}^{\mathbf{T}}\mathbf{x}}_{z}) = f(z) \Rightarrow z = f^{-1}(y)$$

DNNs from Statistical Standpoint

- Effect of the activation function *f*(.) on the density ...
  - **x**: input **w**: weights
  - z: pre-activation y: activation

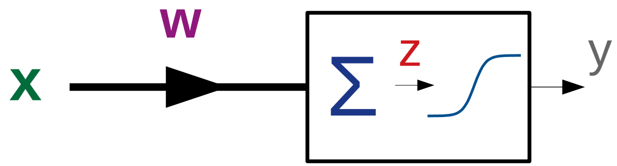


$$y = f(\underbrace{\mathbf{w}^{\mathbf{T}}\mathbf{x}}_{z}) = f(z) \Rightarrow z = f^{-1}(y)$$

$$P_Y(y) = \left| \frac{d}{dy} f^{-1}(y) \right| \; P_Z(f^{-1}(y))$$

DNNs from Statistical Standpoint

- Effect of the activation function *f*(.) on the density ...
  - **x**: input   **w**: weights
  - z: pre-activation   y: activation



$$\mathrm{y} = \mathrm{f}(\underbrace{\mathbf{w}^{\mathbf{T}}\mathbf{x}}_{z}) = f(z) \Rightarrow z = f^{-1}(y)$$

$$\mathrm{P}_Y(y) = \left|\frac{d}{dy} f^{-1}(y)\right| \; P_Z(f^{-1}(y))$$

Derivation of Distribution of Bottleneck Features Analytically – Tanh

$$\mathrm{P}_Y(y) = \left|\frac{d}{dy}f^{-1}(y)\right| \; P_Z(f^{-1}(y))$$

$$y = f(z) = \tanh(z)$$

Derivation of Distribution of Bottleneck Features Analytically – Tanh

$$\mathrm{P}_Y(y) = \boxed{\left|\frac{d}{dy} f^{-1}(y)\right|} \; \boxed{P_Z(f^{-1}(y))}$$

$$y = f(z) = \tanh(z)$$

$$f^{-1}(y) = \frac{1}{2}\log\frac{1+y}{1-y} \quad , \quad \frac{d}{dy} f^{-1}(y) = \frac{1}{1-y^2}$$

Derivation of Distribution of Bottleneck Features Analytically – Tanh

$$\mathrm{P}_Y(y) = \left|\frac{d}{dy} f^{-1}(y)\right| \; P_Z(f^{-1}(y))$$

$$y = f(z) = \tanh(z)$$

$$f^{-1}(y) = \frac{1}{2}\log\frac{1+y}{1-y} \quad , \quad \frac{d}{dy} f^{-1}(y) = \frac{1}{1-y^2}$$

$$\Rightarrow P_Y^{\tanh}(y) = \frac{1}{1-y^2} P_Z\left(\frac{1}{2}\log\frac{1+y}{1-y}\right)$$

Derivation of Distribution of Bottleneck Features Analytically – Tanh

$$\mathrm{P}_Y(y) = \left| \frac{d}{dy} f^{-1}(y) \right| P_Z(f^{-1}(y))$$

$$P_Y^{\tanh}(y) = \frac{1}{1-y^2} P_Z\left(\frac{1}{2} \log \frac{1+y}{1-y}\right)$$

Derivation of Distribution of Bottleneck Features Analytically – Tanh

$$\mathrm{P}_Y(y) = \left|\frac{d}{dy} f^{-1}(y)\right| P_Z(f^{-1}(y))$$

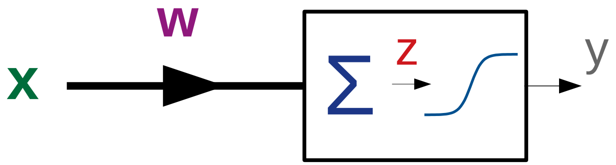
$$P_Y^{\tanh}(y) = \frac{1}{1-y^2} P_Z(\frac{1}{2}\log\frac{1+y}{1-y})$$

Distribution of the pre-activation, $P_Z(z)$, is required!

Distribution of the Pre-Activation (Z)

- *Pre-activation, Z,* is a weighted sum ...

$$z = \mathbf{w}^T \mathbf{x} = w_1 x_1 + w_2 x_2 + ... + w_N x_N$$

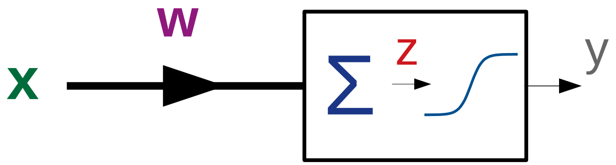


Distribution of the Pre-Activation (Z)

- *Pre-activation, Z,* is a weighted sum
  - Z is approximately Gaussian $\leftarrow$ CLT*

$$z = \mathbf{w}^T\mathbf{x} = w_1 x_1 + w_2 x_2 + \ldots + w_N x_N$$

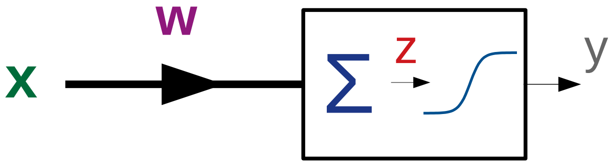
$$\mathrm{z} \boxed{\dot{\sim}} \mathcal{N}(z; \mu_z, \sigma_z^2)$$



Distribution of the Pre-Activation (Z)

- *Pre-activation, Z,* is a weighted sum
  - *Z* is approximately Gaussian $\leftarrow$ CLT
  - $\mu_z \rightarrow 0$: No preference for positive/negative values

$$z \stackrel{\cdot}{\sim} \mathcal{N}(z; 0, \sigma_z^2)$$



Distribution of the Activation (y)

- Now we can work out this ...

$$P_Y^{\tanh}(y) = \frac{1}{1-y^2} P_Z(\frac{1}{2}\log\frac{1+y}{1-y})$$

$$\boxed{\mathrm{z} \mathrel{\dot\sim} \mathcal{N}(z; 0, \sigma_z^2)}$$

Distribution of the Activation (y)

- After some algebraic manipulation ...

$$P_Y^{\tanh}(y) = \frac{1}{1-y^2}\,\mathcal{N}\Big(\frac{1}{2}\log\frac{1+y}{1-y};0,\sigma_z^2\Big)$$

$$= \underbrace{\frac{1}{1-y^2}}_{F_Y^{<1>}(y)}\;\underbrace{\frac{1}{\sqrt{2\pi}\sigma_z}\Big(\frac{1+y}{1-y}\Big)^{-\frac{1}{8\sigma_z^2}\log\frac{1+y}{1-y}}}_{F_Y^{<2>}(y,\sigma_z)}$$

Distribution of the Activation (y)

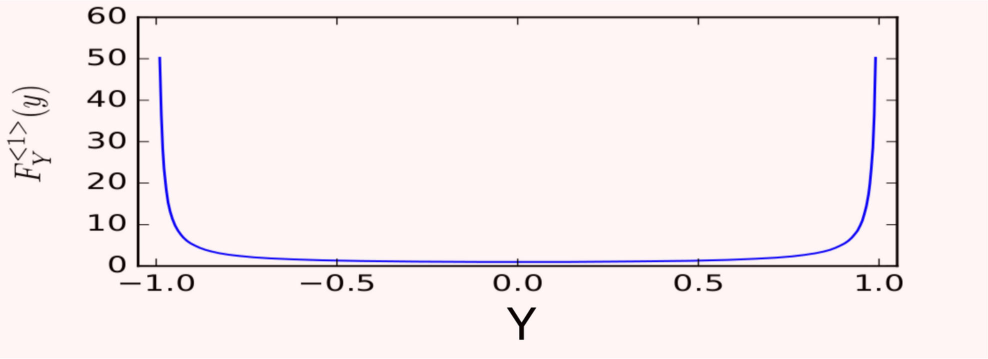
- After some algebraic manipulation ...

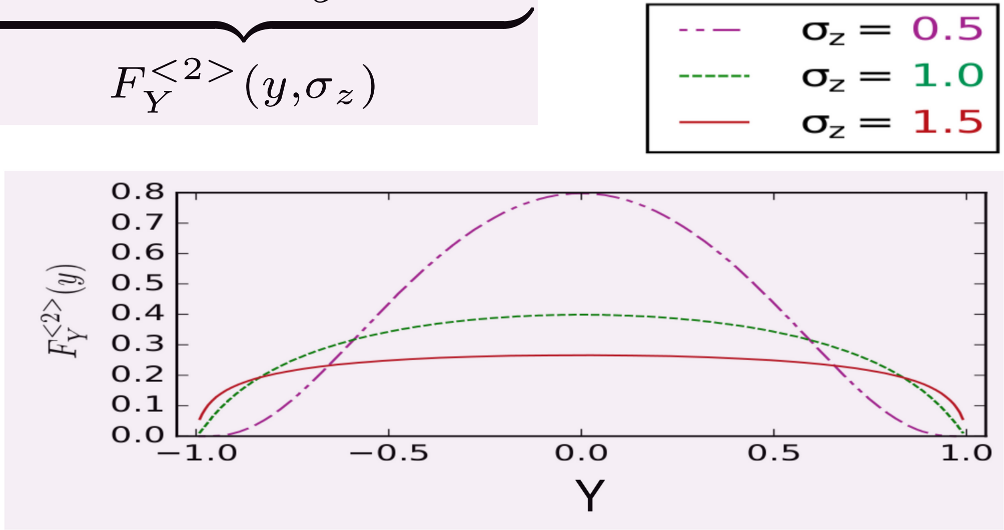
$$P_Y^{\text{tanh}}(y) = \underbrace{\frac{1}{1-y^2}}_{F_Y^{<1>}(y)} \underbrace{\mathcal{N}(\frac{1}{2}\log\frac{1+y}{1-y}; 0, \sigma_z^2)}_{F_Y^{<2>}(y,\sigma_z)}$$

Factor 1 Factor 2 → function of σ_z

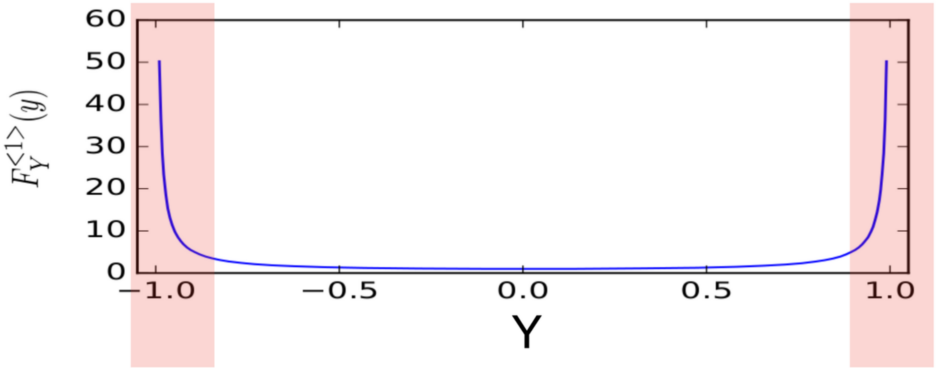
Activation Distribution Factors – Tanh

$$P_Y^{\tanh}(y) = \underbrace{\frac{1}{1-y^2}}_{F_Y^{<1>}(y)} \underbrace{\mathcal{N}\left(\frac{1}{2}\log\frac{1+y}{1-y};0,\sigma_z^2\right)}_{F_Y^{<2>}(y,\sigma_z)}$$

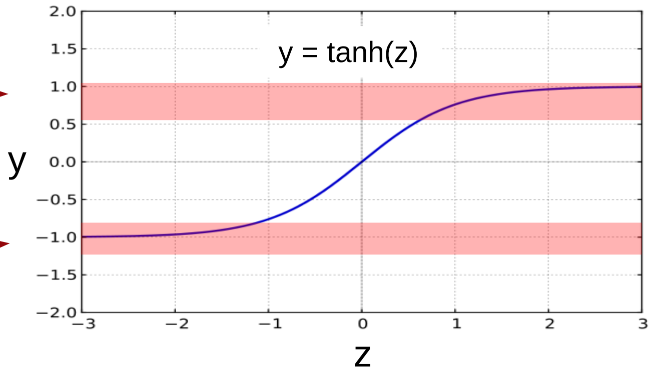




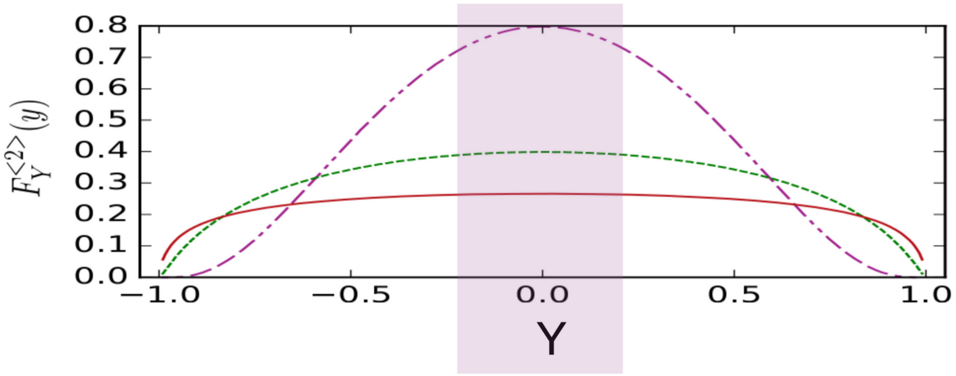
Statistical Interpretation (1)

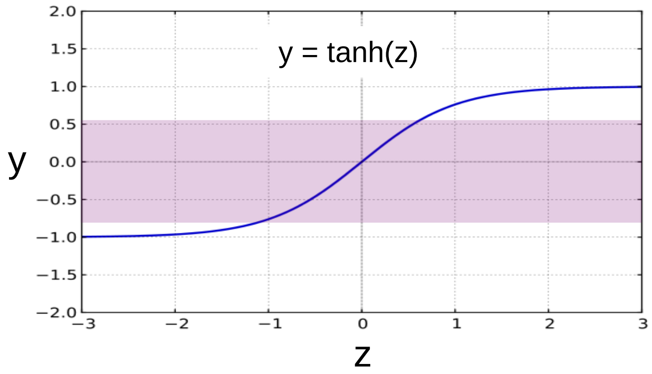


$F^{<1>}$ dominant



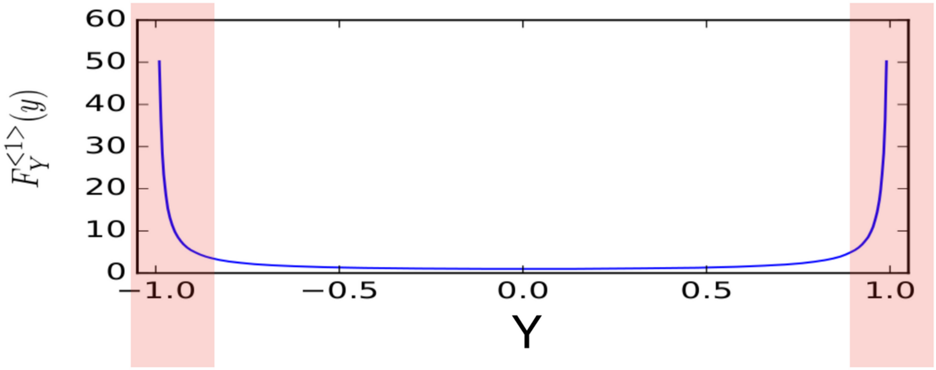
Statistical Interpretation (1)

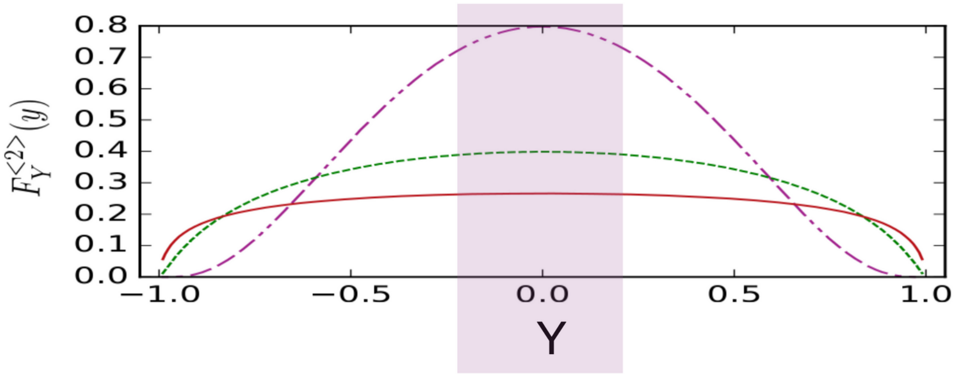




$F^{<2>}$ dominant

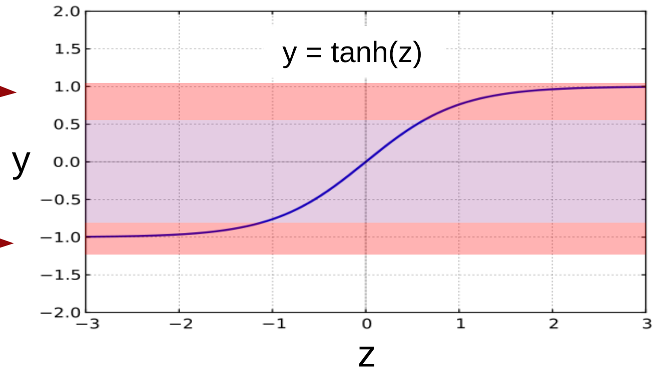
Statistical Interpretation (1)





$F^{<1>}$ dominant

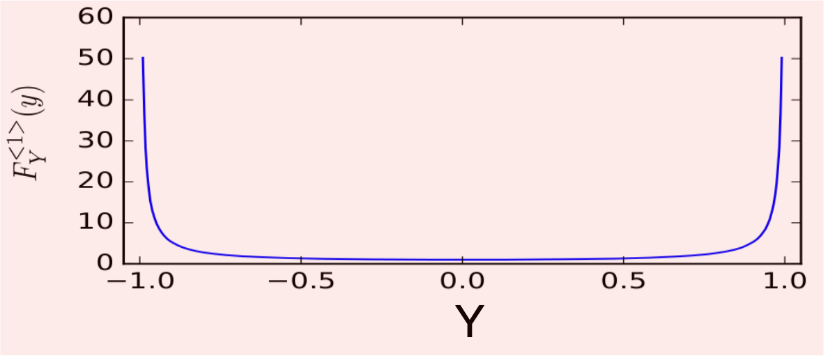
Non-linear region

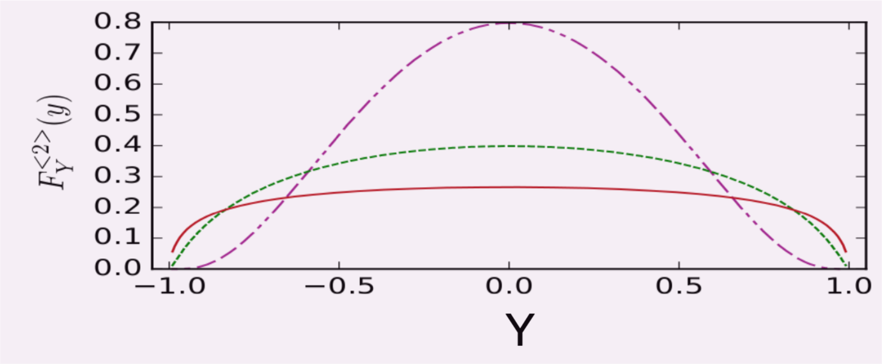


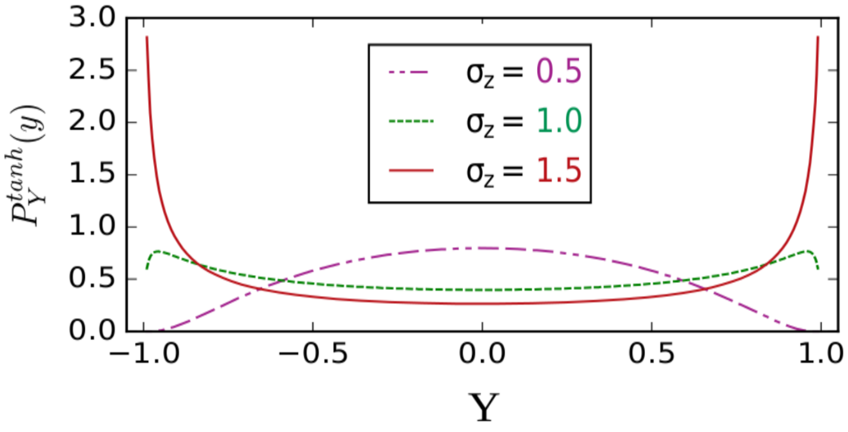
$F^{<2>}$ dominant

Linear region

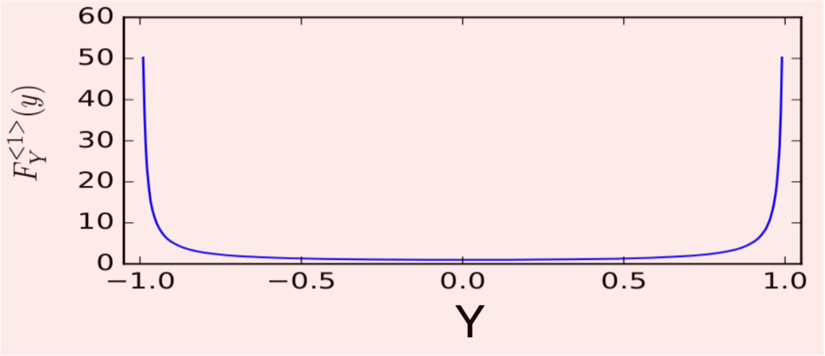
Statistical Interpretation (2) – Product

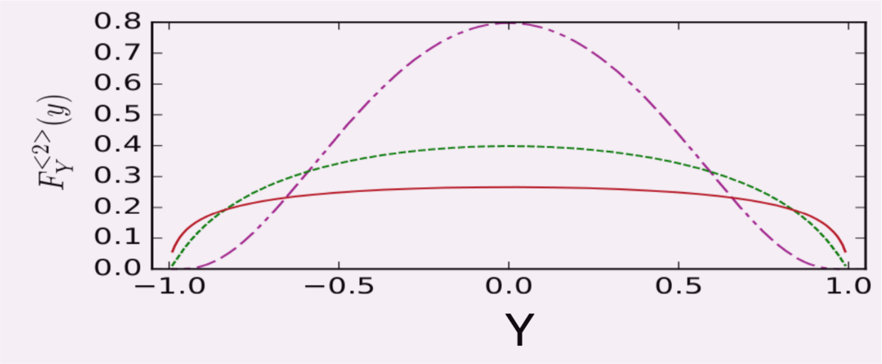




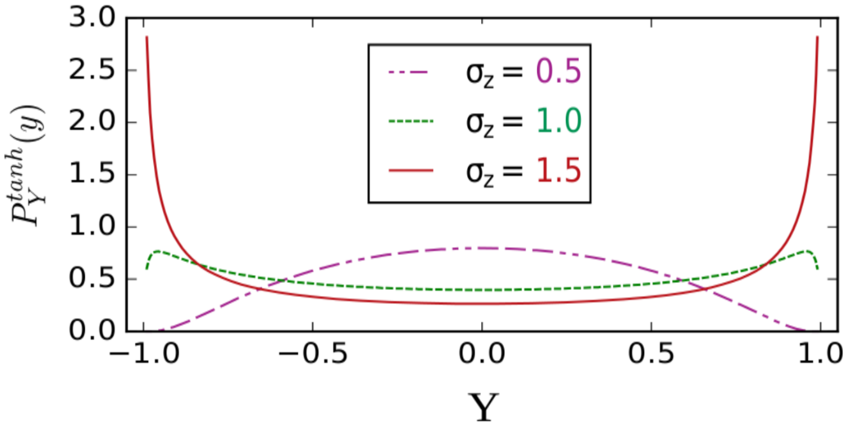


Statistical Interpretation (2) – Product

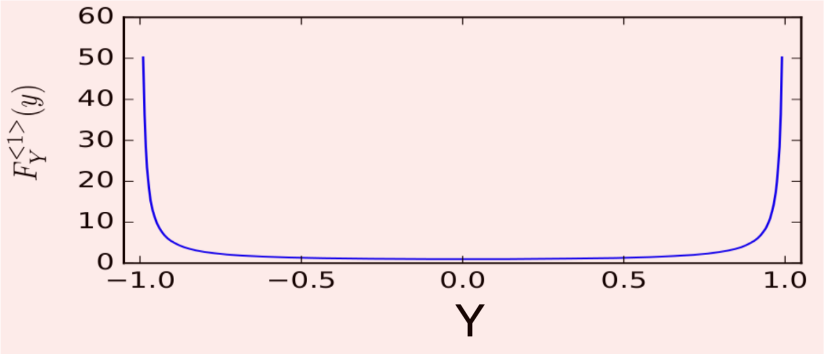


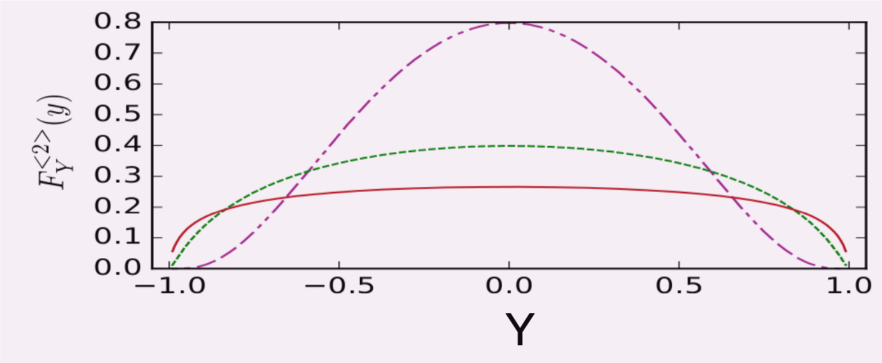


σ_z is a *shape* parameter



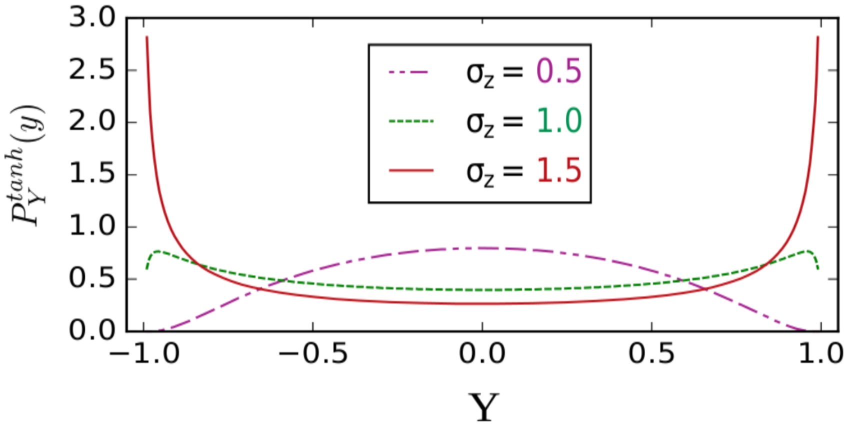
Statistical Interpretation (2) – Product



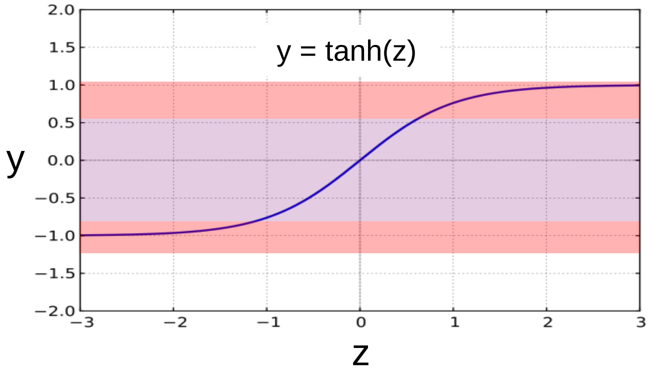


$\sigma_z > 1 \Rightarrow F^{<1>}$ is dominant

$\sigma_z < 1 \Rightarrow F^{<2>}$ is dominant



Statistical Interpretation (2) – Product

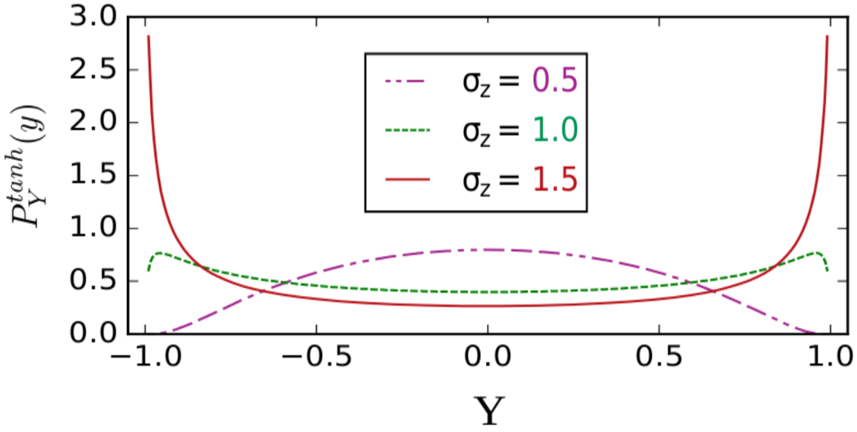


Non-linear

$\sigma_z > 1 \Rightarrow F^{<1>}$ is dominant

$\sigma_z < 1 \Rightarrow F^{<2>}$ is dominant

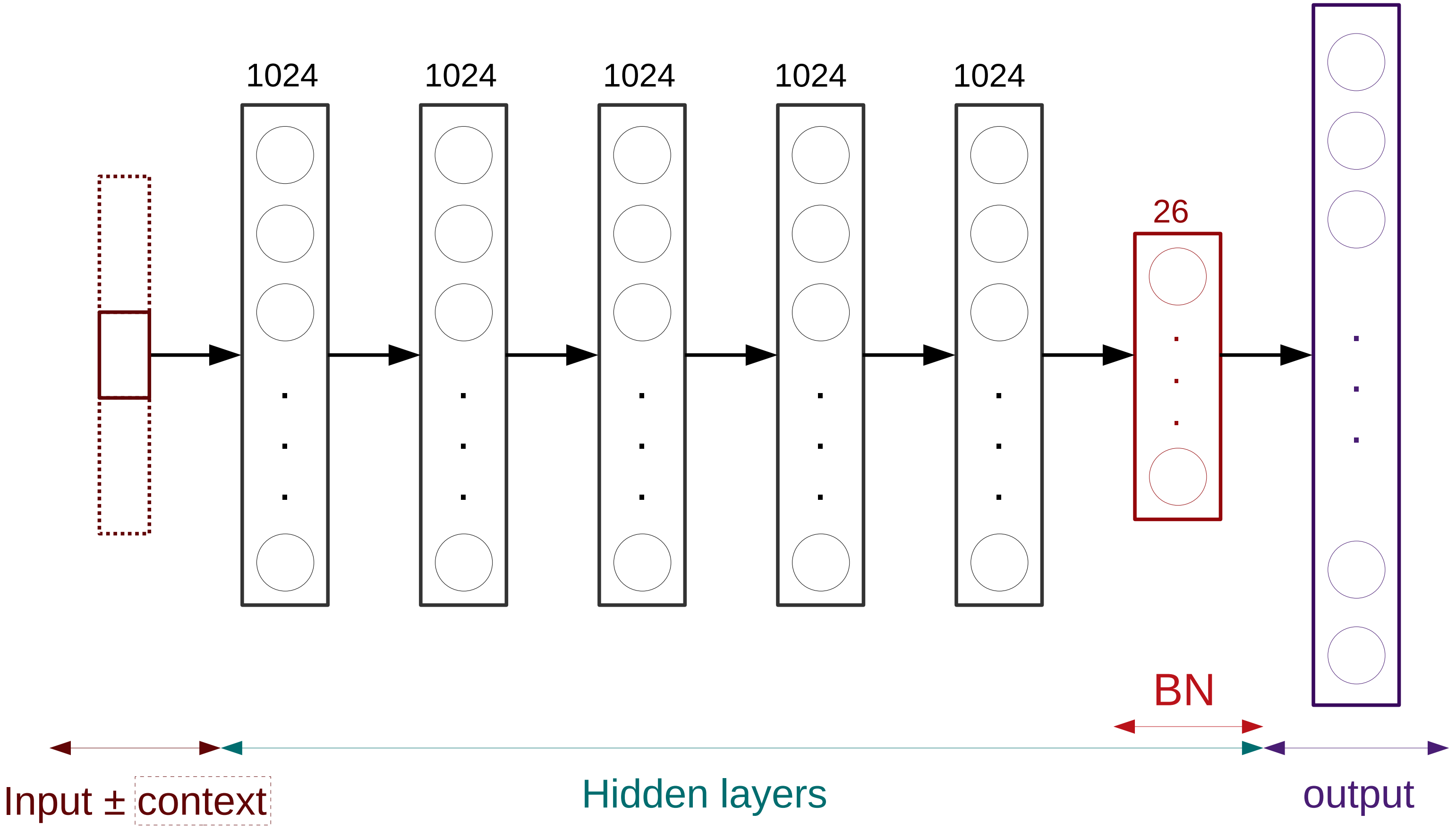
Linear



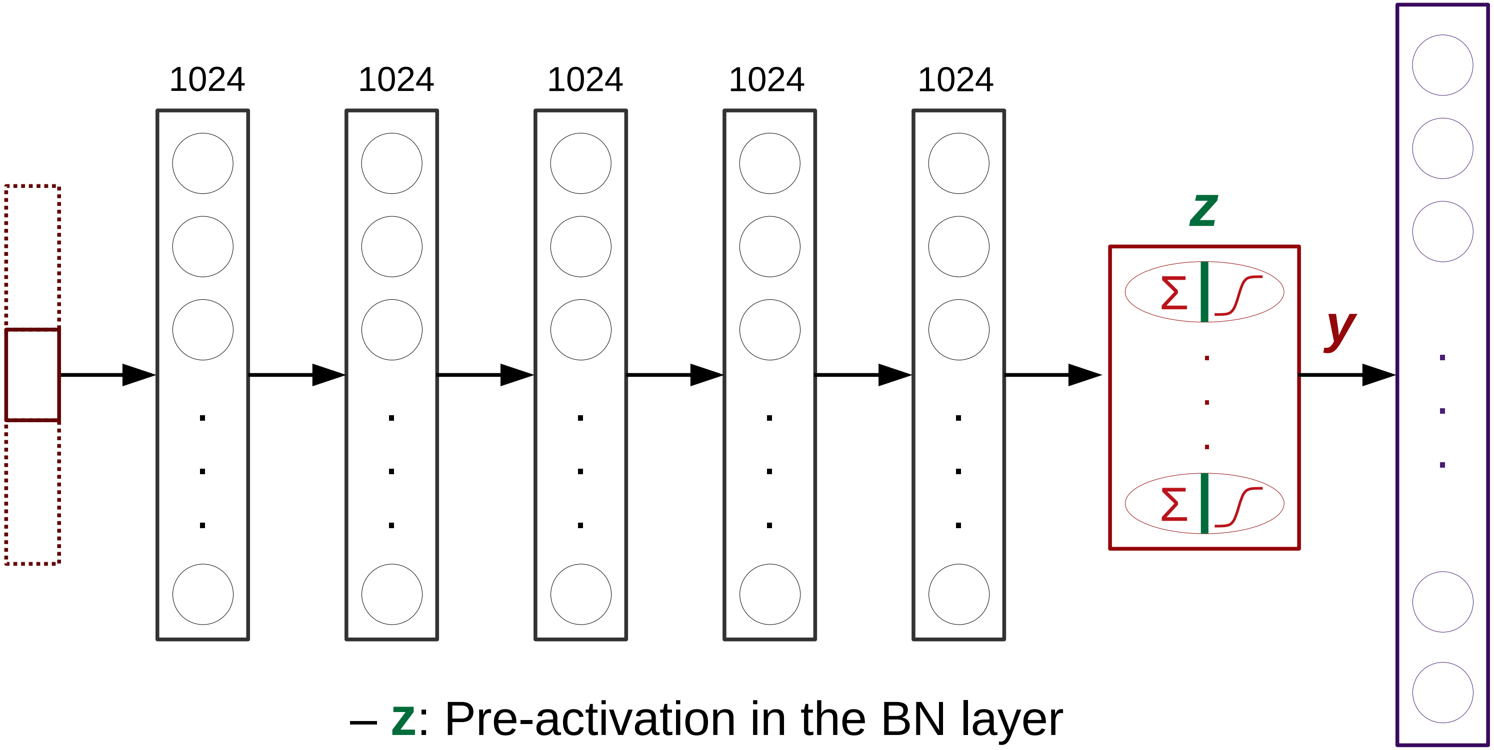
Empirical Studies

- Database: WSJ-5k (SI-84)
- Bottleneck (BN) features extracted using Kaldi
- DNN: TDNN (nnet3), 7 layers: 5x1024 → BN(26D) → output
- Features: log-Filterbank, mean-var normalised, ±5 frames
- #Frames: 5.4 M

DNN Architecture



DNN Architecture



– **z**: Pre-activation in the BN layer

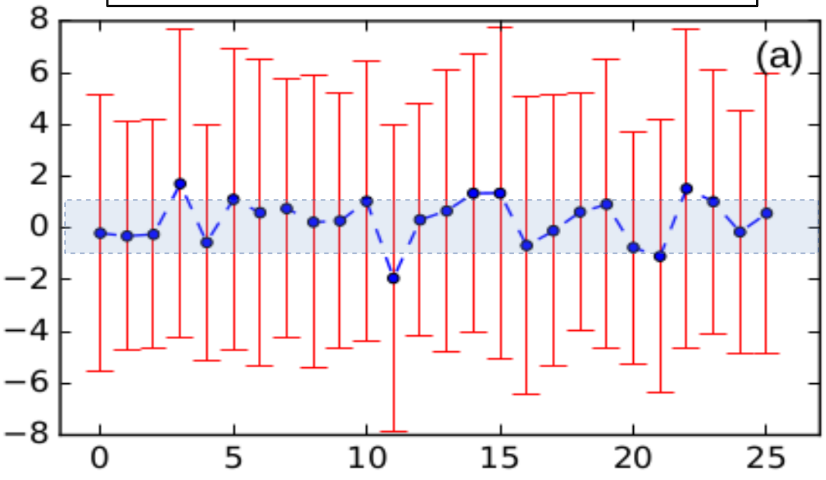
– **y**: Activations of the BN layer

Mean/Std of Pre-activation Z – Error-bar

- $\mu_z \rightarrow 0$ approximation is reasonable ...

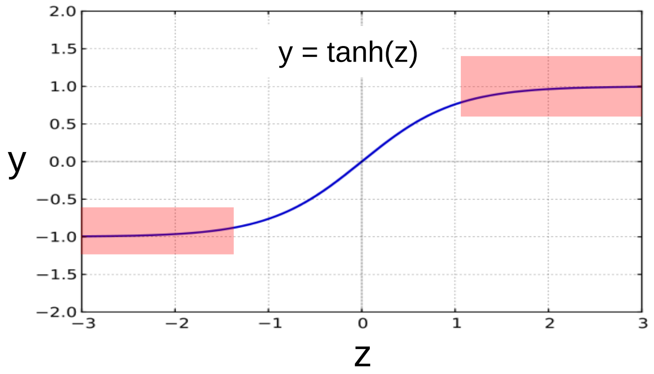
$$z \dot{\sim} \mathcal{N}(z; 0, \sigma_z^2)$$

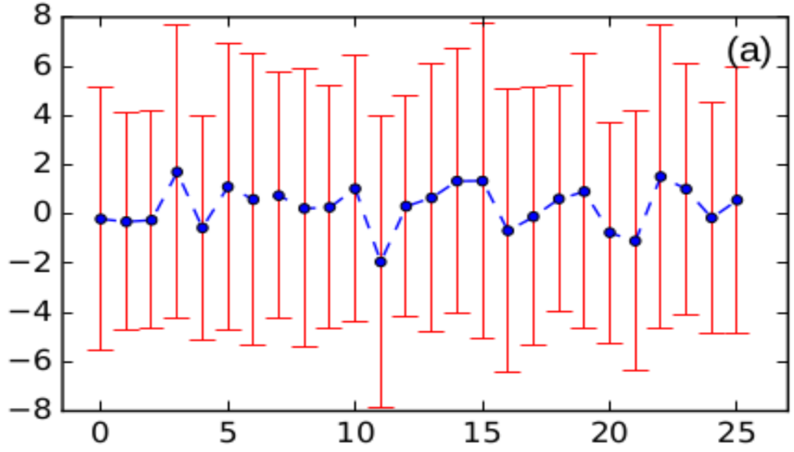
Each bar denotes a node



Mean/Std of Pre-activation Z – Error-bar

- $\mu_z \rightarrow 0$ approximation is reasonable ...
- $\sigma_z > 1$ ==>> DNN operates in the *non-linear mode*



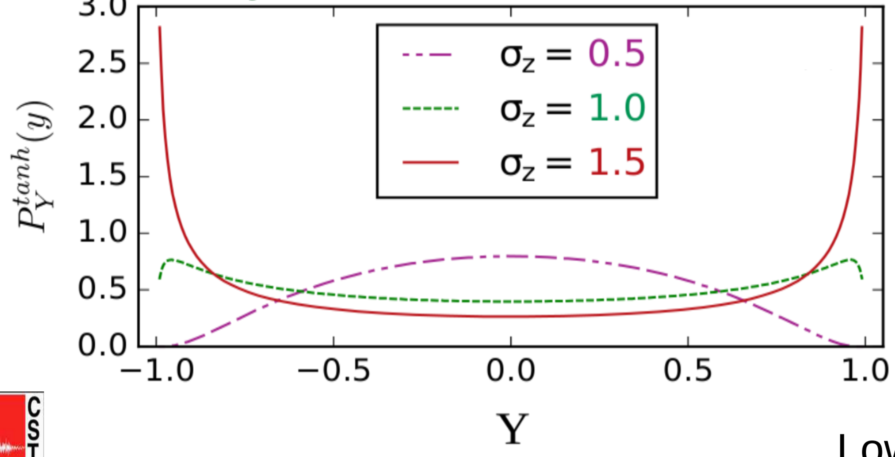


Properties of Z and Y – Empirical Study

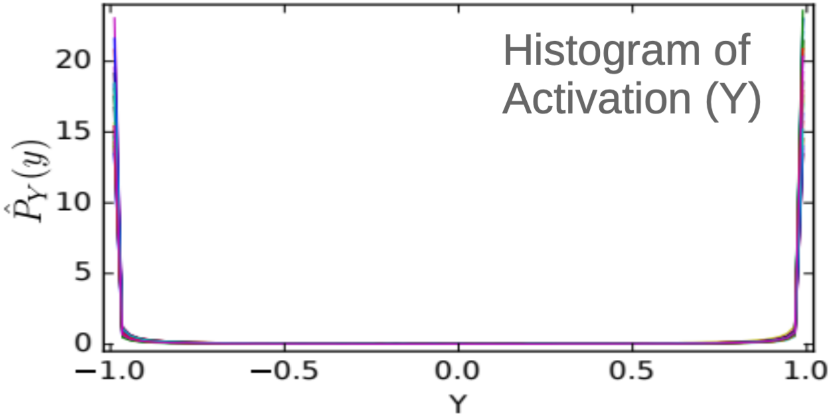
- Distribution of Y matches the derived equation ($\sigma_z > 1$)

$$P_Y^{\tanh}(y) = \frac{1}{1-y^2}\,\mathcal{N}\Big(\frac{1}{2}\log\frac{1+y}{1-y}; 0, \sigma_z^2\Big)$$

Analytic



Empirical



Explaining Two Side Observations

- As bottleneck (BN) features for ASR ...
  - Pre-activation (Z) or activation (Y)?

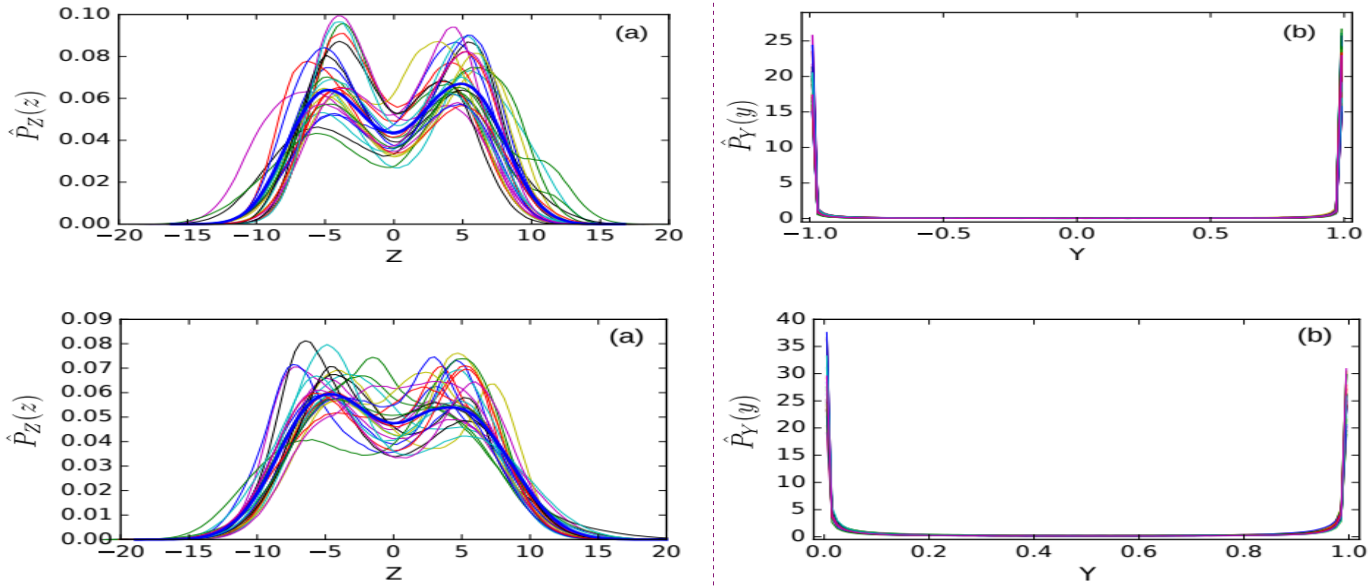
- Sparsity of ReLU; Why and how?

As BN Features, Pre-activation or Activation?

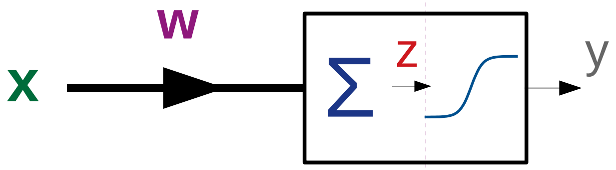
Pre-activation (Z) | Activation (Y)

Tanh

Sigmoid



Each colour denotes a node.



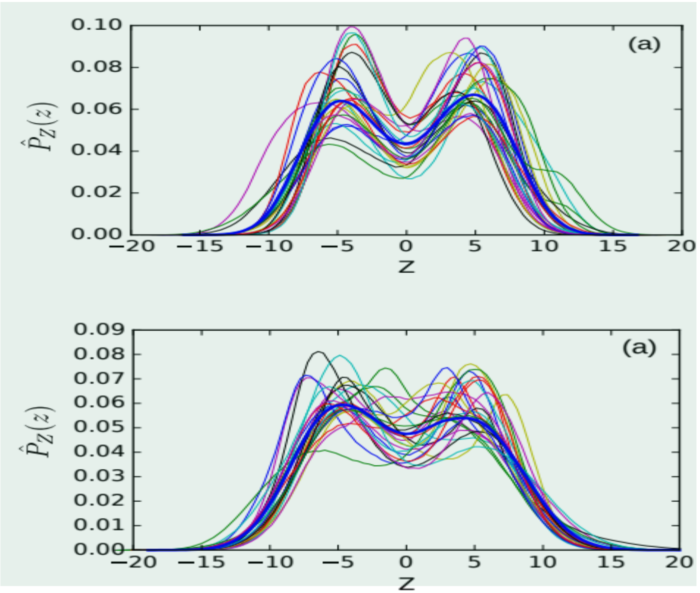
Loweimi et al

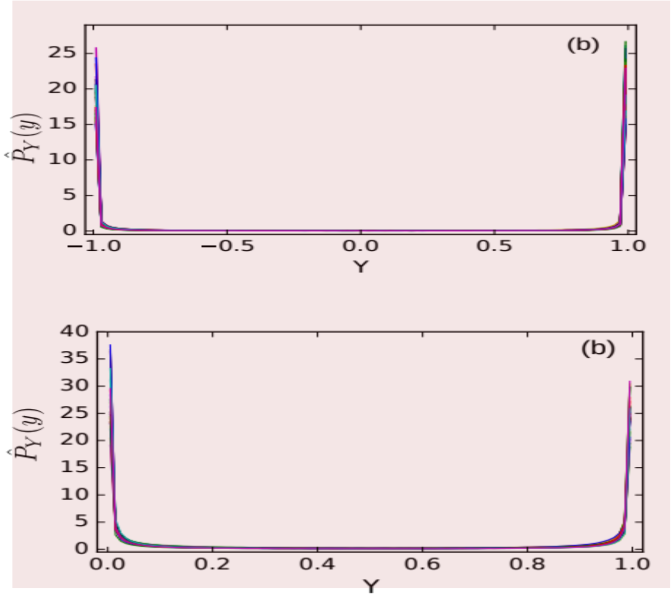
As BN Features, Pre-activation or Activation?

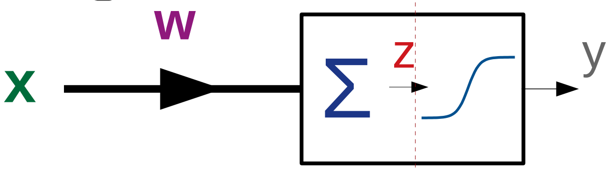
Pre-activation (Z) | Activation (Y)

Tanh

Sigmoid

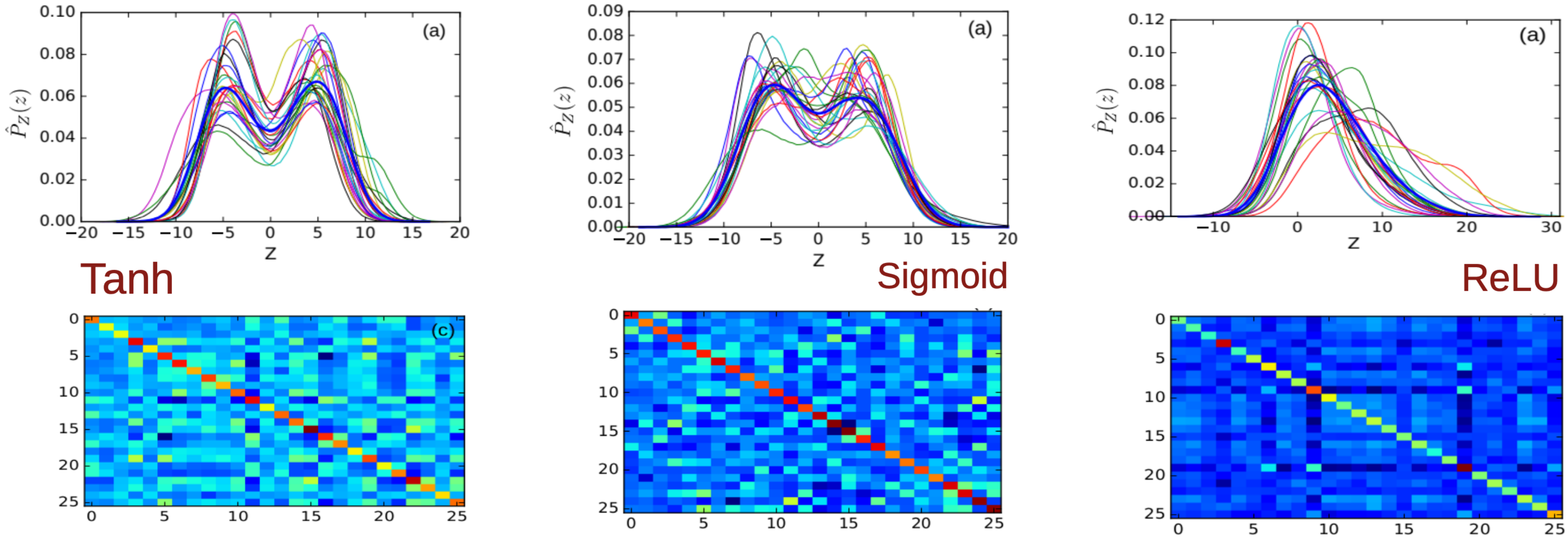






Loweimi et al

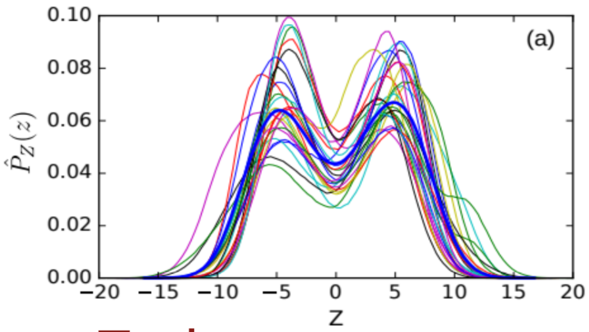
As BN Features, Pre-activation or Activation?



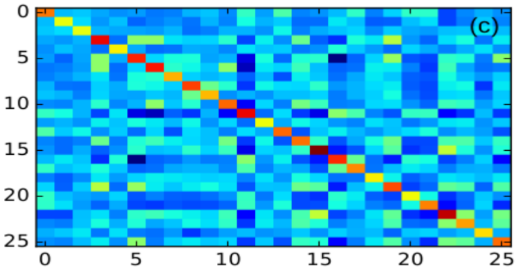
Covariance matrix of Z ...

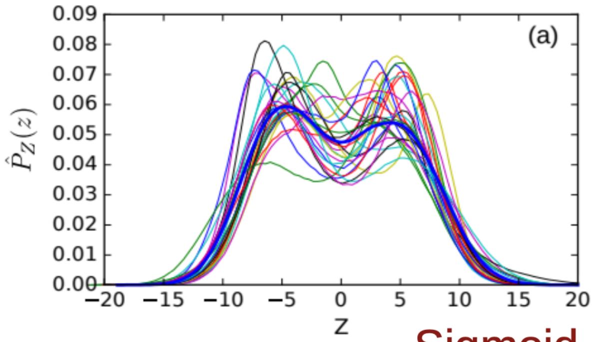
Loweimi et al

As BN Features, Pre-activation or Activation?

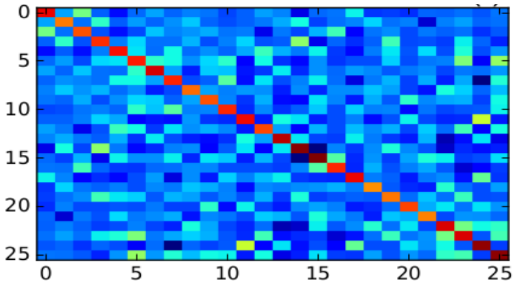


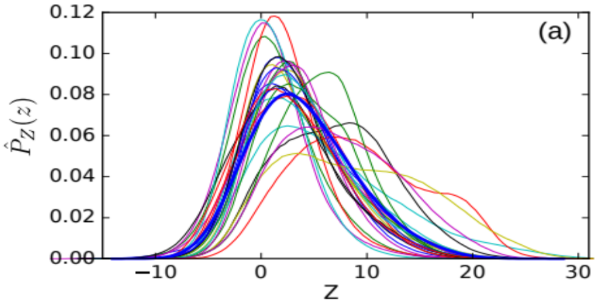
Tanh



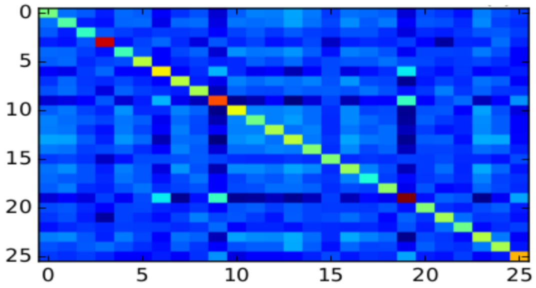


Sigmoid





ReLU



Z Distribution is easily fitted with *diagonal GMMs.*

Sparsity of ReLU

- **ReLU** **Advantages** **…**

  - ✓ **Sparse activations**
    - Biologically plausible
    - Info disentanglement
  - ✓ Gradient propagation
  - ✓ Efficient computation

Deep Sparse Rectifier Neural Networks

Xavier Glorot
DIRO, Université de Montréal
Montréal, QC, Canada
glorotxa@iro.umontreal.ca

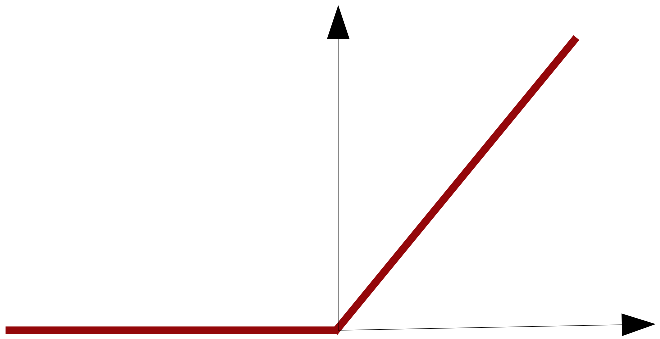
Antoine Bordes
Heudiasyc, UMR CNRS 6599
UTC, Compiègne, France
and
DIRO, Université de Montréal
Montréal, QC, Canada
antoine.bordes@hds.utc.fr

Yoshua Bengio
DIRO, Université de Montréal
Montréal, QC, Canada
bengioy@iro.umontreal.ca

AISTATS 2011

Sparsity of ReLU; Why?

- Glorot et al., "*Deep Sparse Rectifier Neural Networks*", AISTATS 2011
  - Assuming the probability of positive and negative pre-activations is equal, half of the activations (50%) will be zero ...

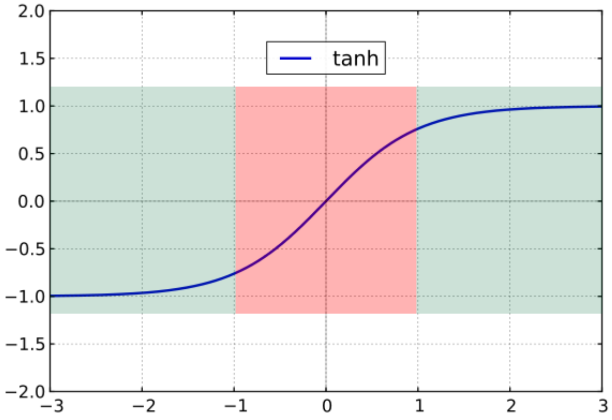


Sparsity of ReLU; Why?

- **Our argument ...**
  - Remember Tanh, linear and non-linear zones ...

✓ Non-linear mode

⊗ Linear mode



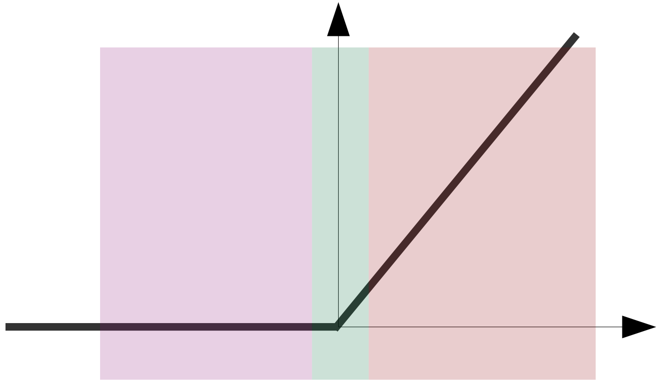
Sparsity of ReLU; Why?

- **Our argument ...**
  - To operate in the non-linear mode, activations must be around $\mathbf{0^+}$

⊗ Blocks information

✓ Non-linear mode (switch)

⊗ Linear mode



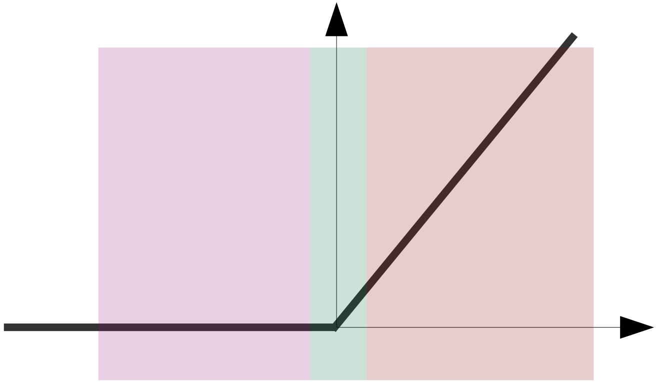
Sparsity of ReLU; Why?

- **Coincidence** of non-linear operating zone with $\mathbf{0^+}$

❌ Blocks information

✅ Non-linear mode (switch)

❌ Linear mode

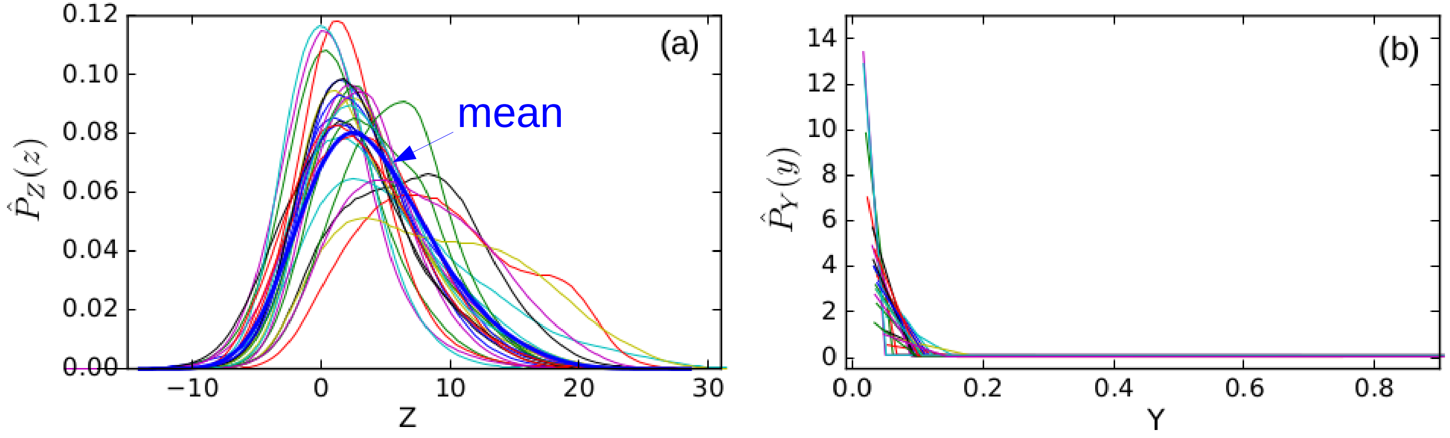


Sparsity of ReLU -- Empirical Results

- Distribution of the pre-activation (z) and activation (y) for ReLU ...

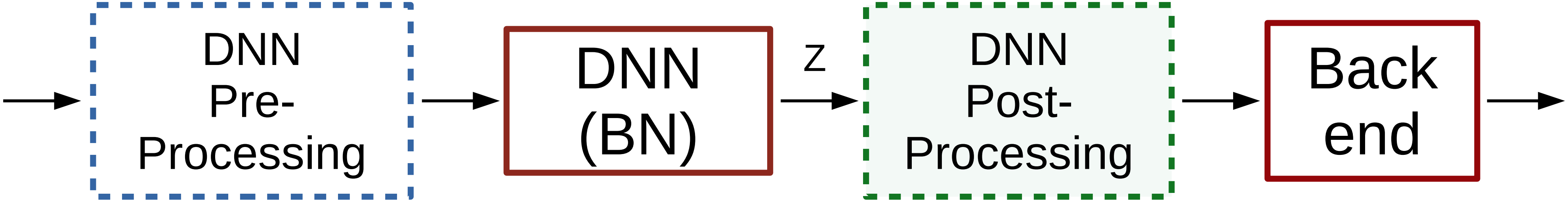
Pre-activation (z) pdf

Activation (y) pdf



Each colour denotes a node.

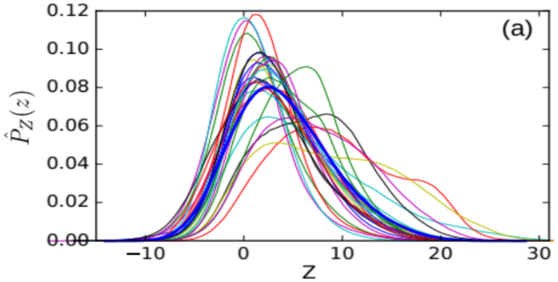
Feature Normalisation for DNNs



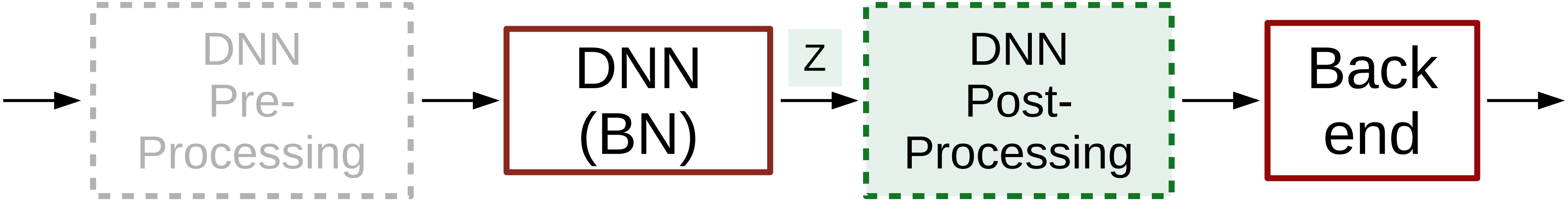
Loweimi et al., ICASSP 2018
(gVTS → DNN)

ICASSP 2019

Feature Normalisation for DNNs



Z is amenable to statistical normalisation



Loweimi et al., ICASSP 2018
(gVTS → DNN)

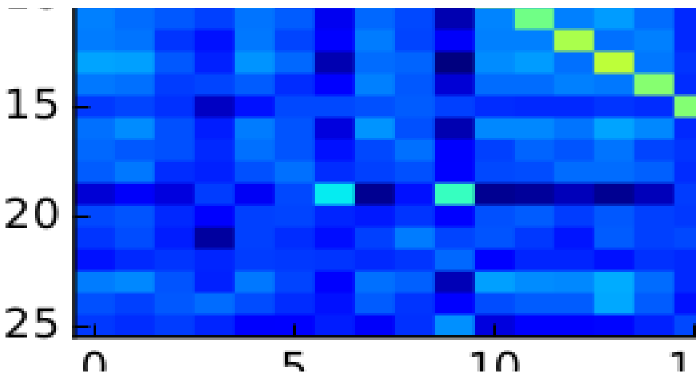
ICASSP 2019

Feature Post-processing for ASR

- Minimise test/train mismatch
  - mean(-variance) normalisation, Gaussianisation

Feature Post-processing for ASR

- Minimise test/train mismatch
  - mean(-variance) normalisation, Gaussianisation
- Orthogonalisation or Decorrelation
  - PCA or DCT



Feature Post-processing for ASR

- Minimise test/train mismatch
  - mean(-variance) normalisation, Gaussianisation
- Orthogonalisation or Decorrelation
  - PCA or DCT
- Feature Enhancement (Noise Robustness)
  - Cannot do (g)VTS → Environment model is not available!
  - Histogram Equalisation (HEQ)

ASR Experiments – Aurora-4 Noisy Training Set (only Additive)

Aurora-4 Train Set:
- Clean+Additive noise

Aurora-4 Test Sets:
- A: Clean (match)
- B: Additive noise (match)
- C: Channel (mismatch)
- D: Additive+Channel (match)

- MN: Mean normalised
- MVN: Mean-variance normalised
- Gauss: Gaussianisation
- HEQ: Histogram Equalisation

Table 1: *WER for Aurora-4 (Kaldi-LDA-MLLT).*

Feature	A	B	C	D	*Ave4*
BN (baseline)	3.87	7.96	21.80	32.72	16.58
BN+MN	3.64	7.66	21.02	32.20	16.13
BN+MVN	4.07	8.31	20.34	33.04	16.44
BN+Gauss	4.15	8.12	20.18	32.67	16.28
BN+HEQ	3.96	7.43	19.76	30.87	15.50
BN+PCA	3.75	7.88	21.56	32.46	16.41
BN+DCT	3.77	7.77	21.76	32.49	16.44

ASR Experiments – Aurora-4
Noisy Training Set (only Additive)

BN post-processing is helpful.

MN is consistently useful.

HEQ → highest WER reduction
– Testset C: 2% WER reduction

Decorrelation has no effect.

Table 1: *WER for Aurora-4 (Kaldi-LDA-MLLT).*

Feature	A	B	C	D	*Ave4*
BN (baseline)	3.87	7.96	21.80	32.72	16.58
BN+MN	3.64	7.66	21.02	32.20	16.13
BN+MVN	4.07	8.31	20.34	33.04	16.44
BN+Gauss	4.15	8.12	20.18	32.67	16.28
BN+HEQ	3.96	7.43	19.76	30.87	15.50
BN+PCA	3.75	7.88	21.56	32.46	16.41
BN+DCT	3.77	7.77	21.76	32.49	16.44

– MN: Mean normalised
– MVN: Mean-variance normalised
– Gauss: Gaussianisation
– HEQ: Histogram Equalisation

Wrap-up for Part I

- *Statistical* properties of Z and Y was investigated ...
  - *Analytically & Empirically*
- Re-explanations for ...
  - Pre-activation → easily fitted with *diagonal GMMs*
  - *Sparsity* of ReLU → Non-linearity
- *Post-processing* of BN features was investigated …
  - Up to 2% absolute (9% relative) WER reduction achieved

Outline

- PART I
  - Interpreting DNN's Activations
    - ICASSP 2019
- **PART II**
  - Interpreting DNN's Weights
    - Submitted to INTERSPEECH 2019

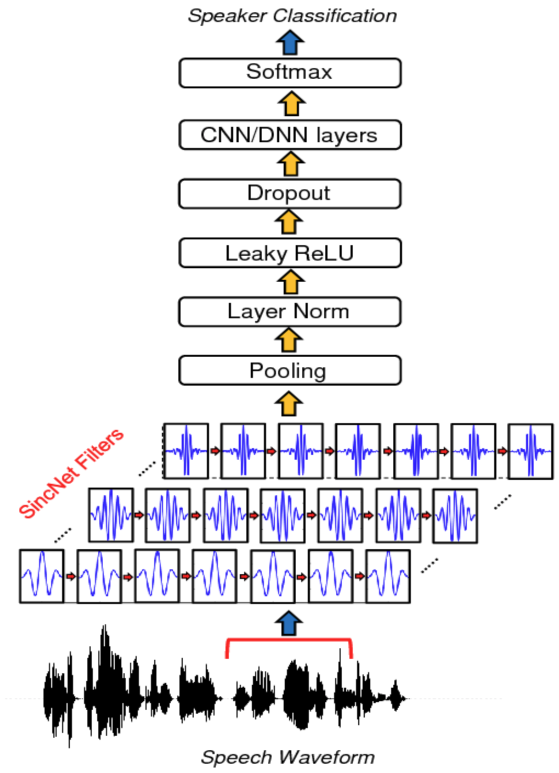
Part II Outline

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

Raw Waveform Modelling via SincNet

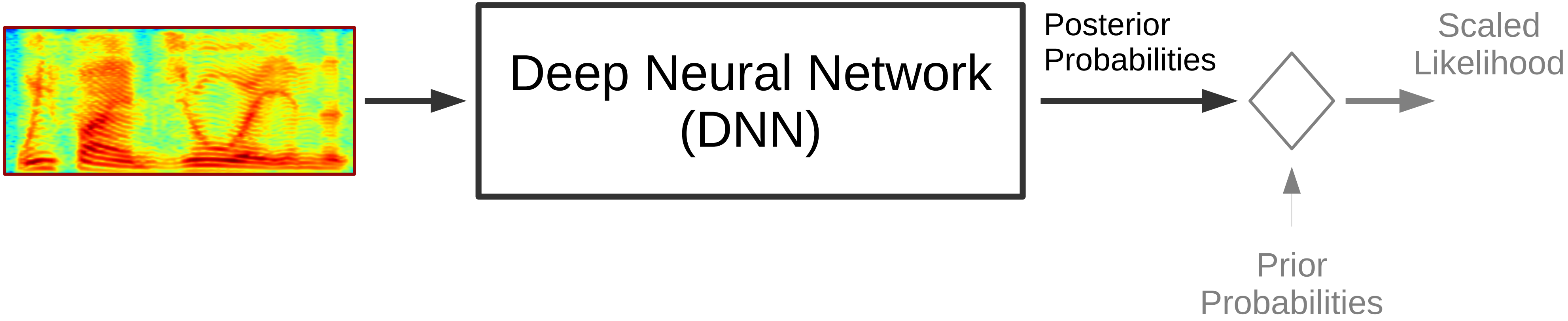
SincNet

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



Acoustic Modelling from Conventional Features

- Conventional Features ↔ Fourier magnitude-based



Acoustic Modelling from Raw Waveform

- Advantages w.r.t. Fourier-based features

Acoustic Modelling from Raw Waveform

- Advantages w.r.t. Fourier-based features
  - Learned vs handcrafted pipeline
    - Task-oriented → optimal for the given tasks/labels
    - Employ all signal info → including *all-pass* and *phase* spec.
    - Learning basis functions → instead of Fourier's *complex exp.*

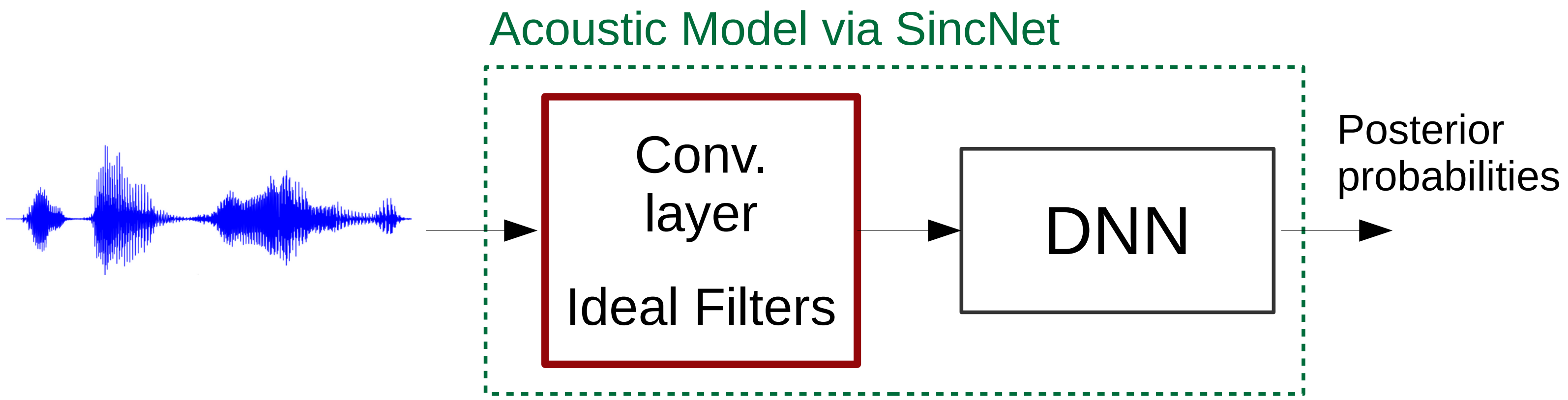
Acoustic Modelling from Raw Waveform

- Advantages w.r.t. Fourier-based features
  - Learned vs handcrafted pipeline
    - Task-oriented → optimal for the given tasks/labels
    - Employ all signal info → including *all-pass* and ***phase*** spec.
    - Learning basis functions → instead of Fourier's *complex exp.*

E. Loweimi, "**Robust Phase-based Speech Signal Processing; From Source-filter Separation to Model-based Robust ASR**," Ph.D. dissertation, University of Sheffield, Sheffield, UK, Feb 2018. [Online]. Available: http://etheses.whiterose.ac.uk/19409/

SincNet – Definition

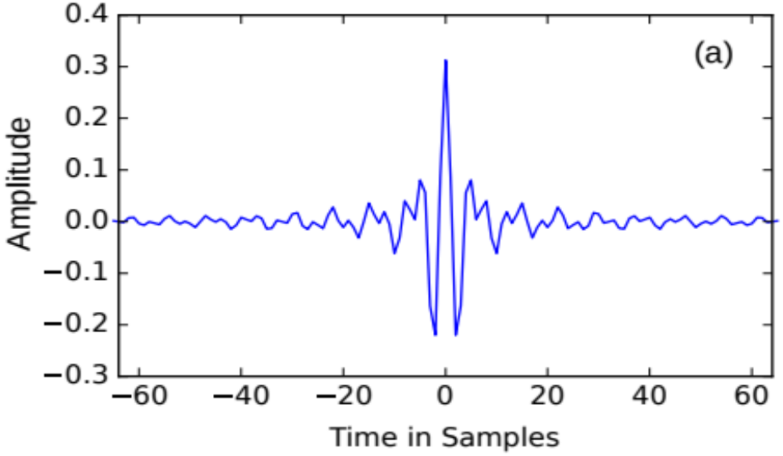
- Convolutional layer with ideal bandpass filters
  - Impulse response $\leftarrow$ Sinc



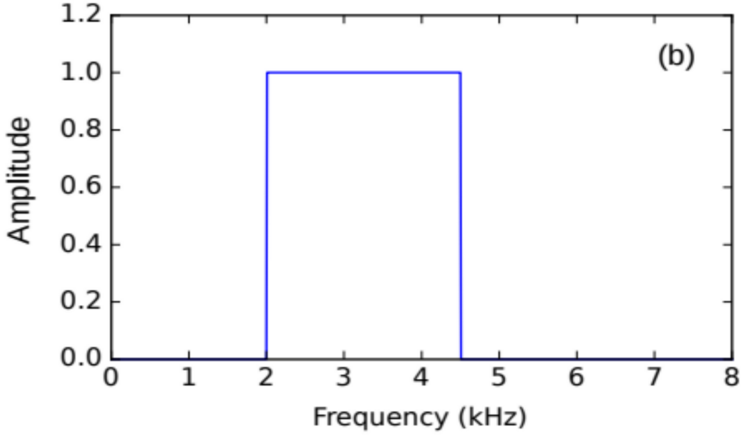
SincNet – Filters

- Impulse & Frequency Responses

Impulse response



Frequency response
(ideal bandpass filter)



SincNet – Filters

- Filters' mathematical definition ...

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

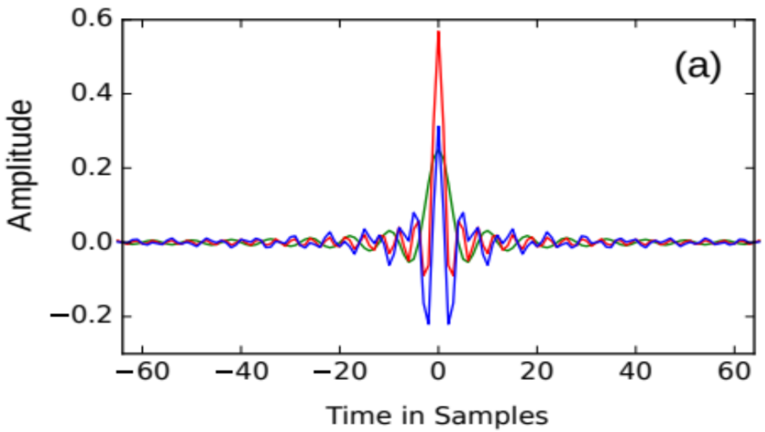
$$H(f;\theta^{(i)}) = \Pi(\frac{f}{2f_2^{(i)}}) - \Pi(\frac{f}{2f_1^{(i)}})$$

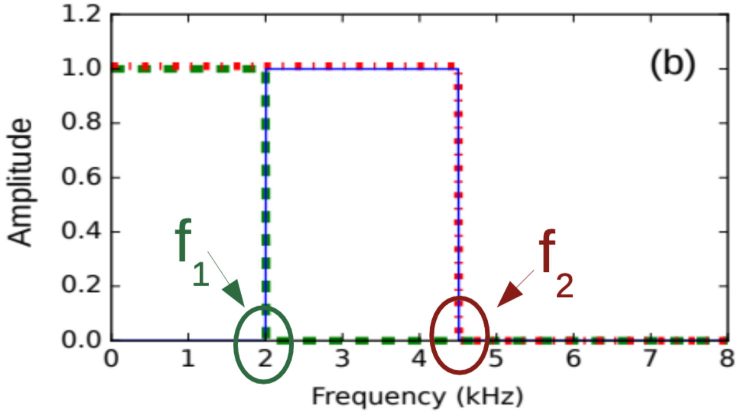
i: filter index in the filterbank

SincNet – Filters Shape

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

$$H(f;\theta^{(i)}) = \Pi(\frac{f}{2f_2^{(i)}}) - \Pi(\frac{f}{2f_1^{(i)}})$$





SincNet – Parameters

- Parameter Set (Θ) $\rightarrow$ cut-off frequencies: f_1 & f_2

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

$$H(f;\theta^{(i)}) = \Pi(\frac{f}{2f_2^{(i)}}) - \Pi(\frac{f}{2f_1^{(i)}})$$

$$\Theta = \{\theta^{(i)}\} = \{f_1^{(i)}, f_2^{(i)}\}$$

Backprop

Advantages of SincNet

SincNet vs CNN -- Advantages

- Parametric vs Non-parametric

SincNet vs CNN -- Advantages

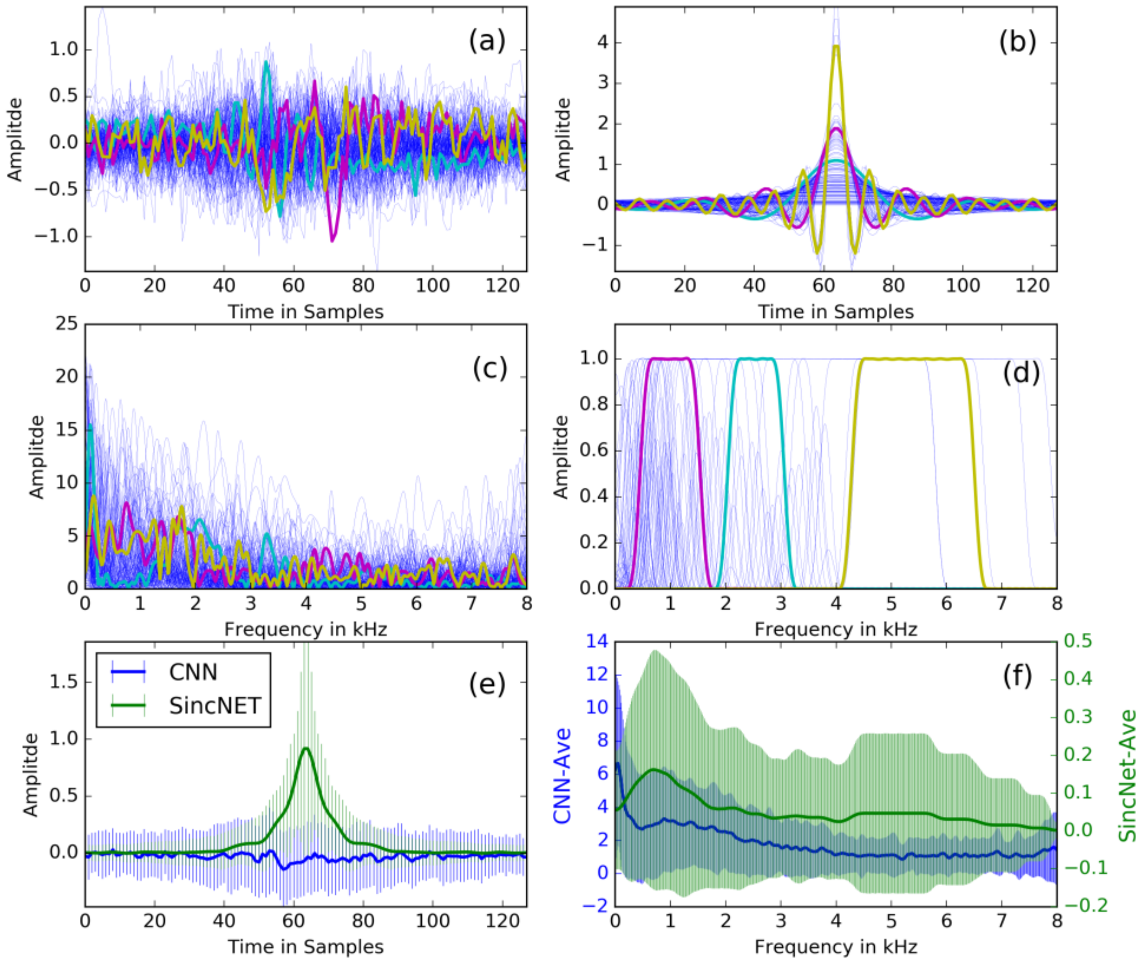
- **Parametric** vs Non-parametric
  - More Interpretable

CNN vs SincNet

CNN
impulse responses

CNN
Frequency responses

Average
impulse responses



SincNet
impulse responses

SincNet
Frequency responses

Average
Frequency responses

SincNet vs CNN -- Advantages

- **Parametric** vs Non-parametric
  - More interpretable
  - Constraint on hypothesis space
    - Regularisation → better generalisation

SincNet vs CNN -- Advantages

- **Parametric** vs Non-parametric
  - More interpretable
  - Constraint on hypothesis space
    - Regularisation → better generalisation
  - Fewer parameters
    - Less training data required
    - Faster learning/convergence

SincNet vs CNN -- Advantages

- Parametric vs Non-parametric
- Better performance on TIMIT ...
  - Lower loss and phone error rate (PER)

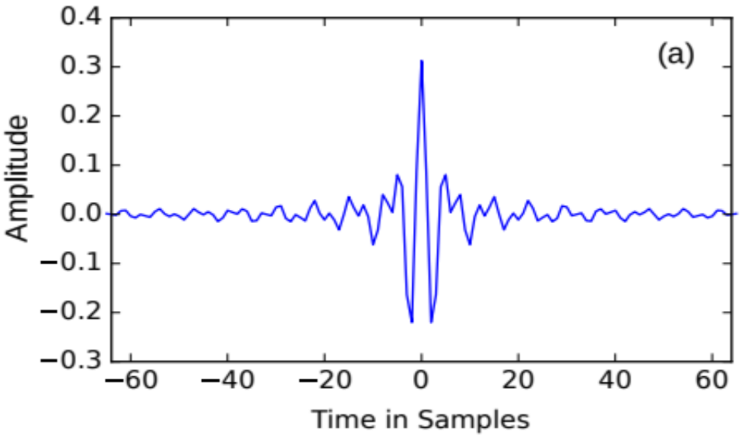
Creator:LibreOffice 6.0
LanguageLevel:2

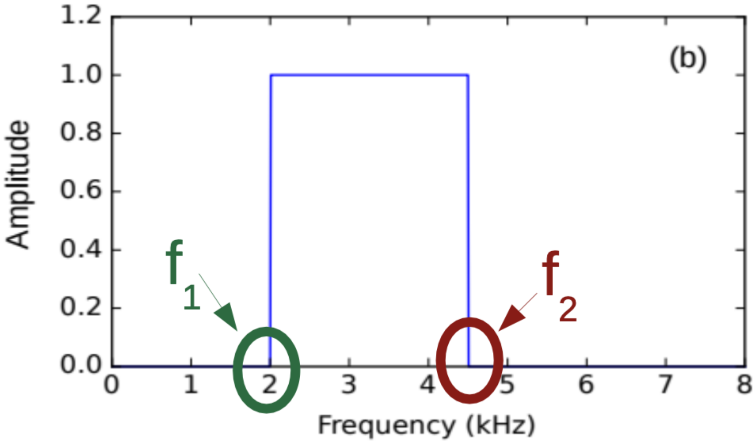
General Formulation for Interpretable Kernel-based CNNs

Loweimi et al

Interpretable Kernel-based CNNs

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





Loweimi et al

Interpretable Kernel-based CNNs

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

$$h(t;\theta^{(i)}) = \frac{1}{\pi t}(\sin(2\pi f_2^{(i)}t) - \sin(2\pi f_1^{(i)}t))$$

Interpretable Kernel-based CNNs

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

$$h(t;\theta^{(i)}) = \frac{1}{\pi t}(\sin(2\pi f_2^{(i)}t) - \sin(2\pi f_1^{(i)}t))$$

$$\sin\alpha - \sin\beta = 2\sin\frac{\alpha-\beta}{2}\cos\frac{\alpha+\beta}{2}$$

Interpretable Kernel-based CNNs

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

$$h(t;\theta^{(i)}) = \frac{1}{\pi t}(\sin(2\pi f_2^{(i)}t) - \sin(2\pi f_1^{(i)}t))$$

$$\sin\alpha - \sin\beta = 2\sin\frac{\alpha-\beta}{2}\cos\frac{\alpha+\beta}{2}$$

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

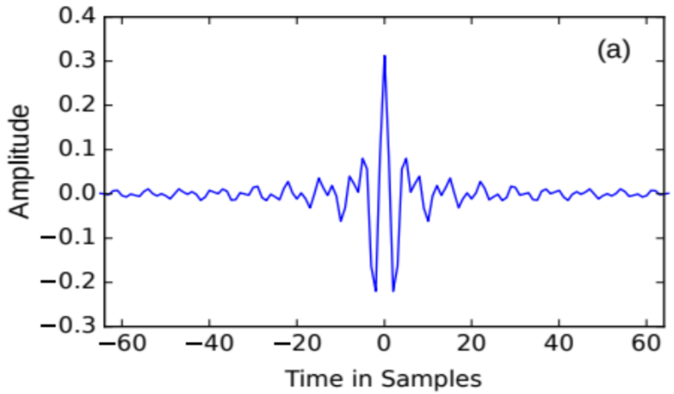
$$B^{(i)} = f_2^{(i)} - f_1^{(i)} \quad , \quad f_c^{(i)} = \frac{f_1^{(i)} + f_2^{(i)}}{2}$$

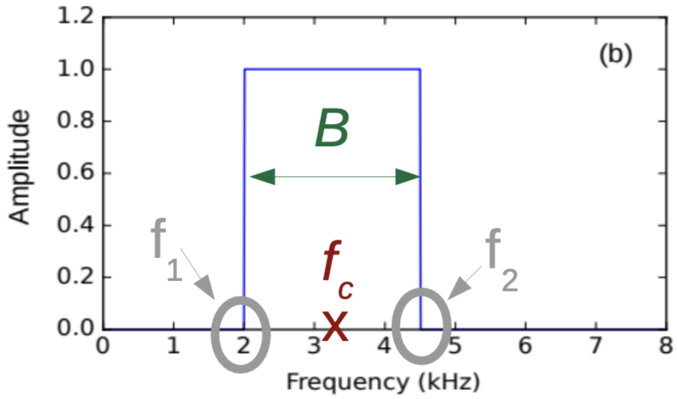
Interpretable Kernel-based CNNs

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

Baseband filter ≡ Kernel

Carrier





Interpretable Kernel-based CNNs

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

Baseband filter ≡ Kernel   Carrier

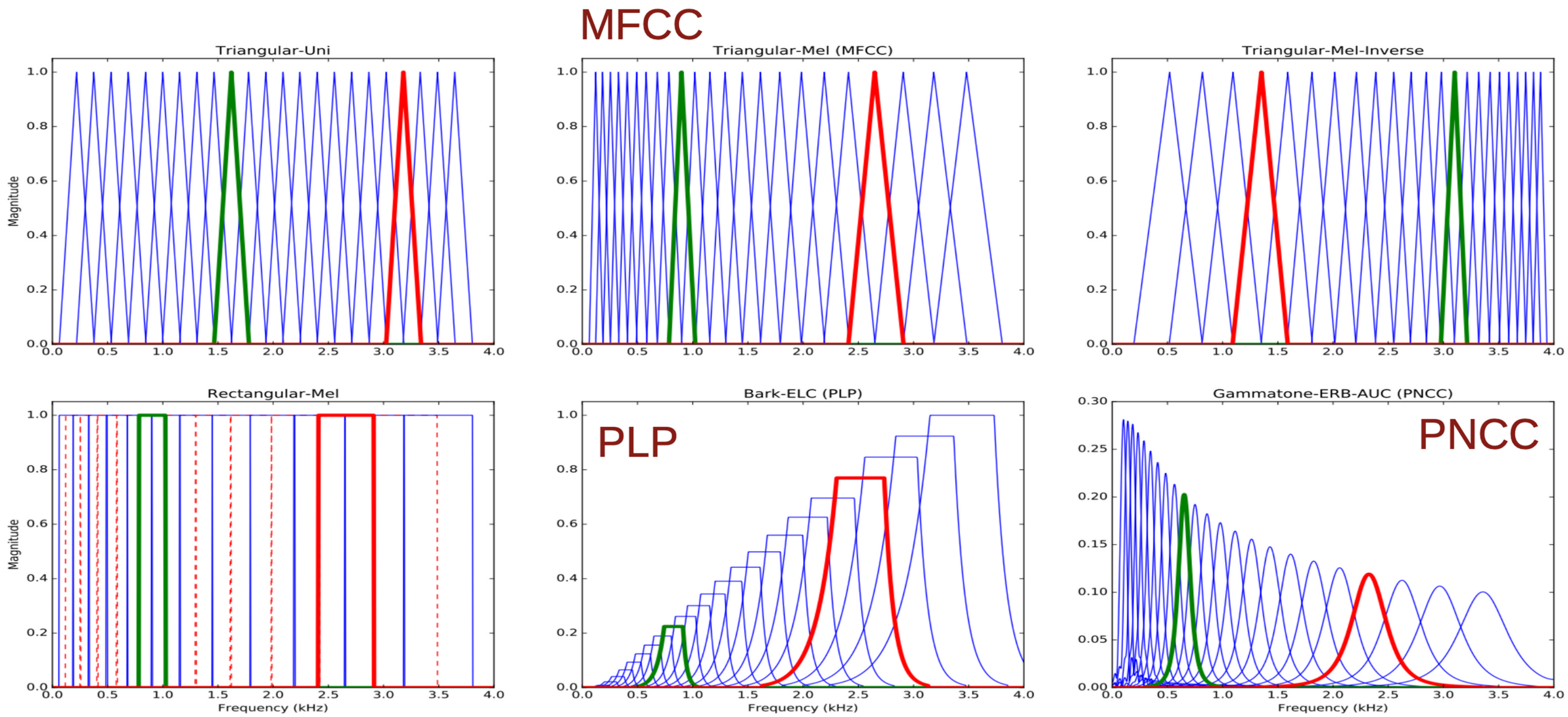
$$h^{(i)}(t; \theta^{(i)}, f_c^{(i)}) = \boxed{K(t; \theta^{(i)})} \; \boxed{carrier(t; f_c^{(i)})}$$

Interpretable Kernel-based CNNs

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

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

Learning Kernel-based Filterbanks



Loweimi et al

Learning Kernel-based Filterbanks

SincNet

Interpretable CNNs with Parametric Kernel-based Filters

Sinc^{2}Net

GammaNet

GaussNet

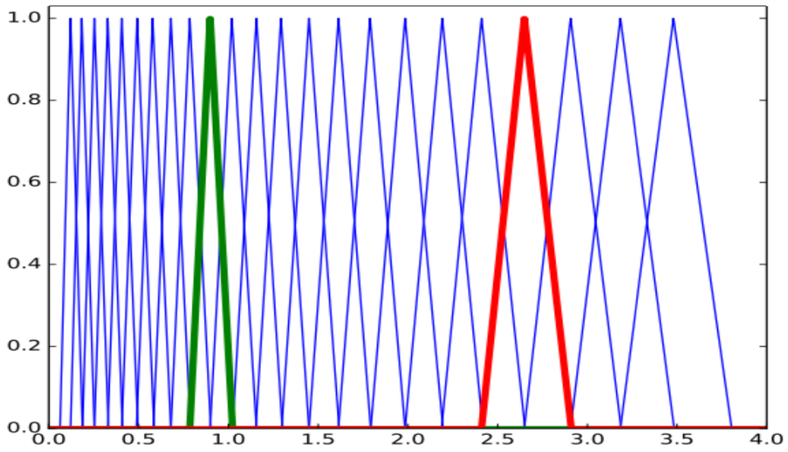
Sinc^{2}Net: Triangular Filters

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

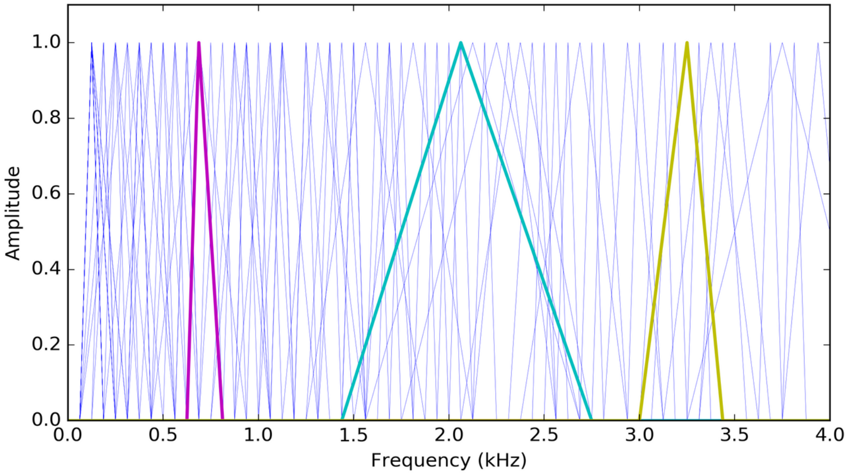
Sinc²Net: Triangular Filters

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

Engineered



Learned

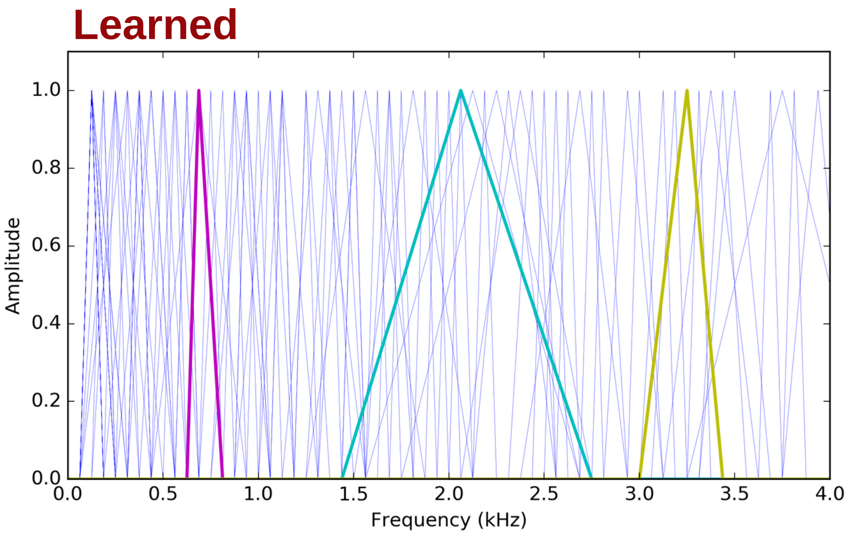


Sinc²Net: Triangular Filters

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

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

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



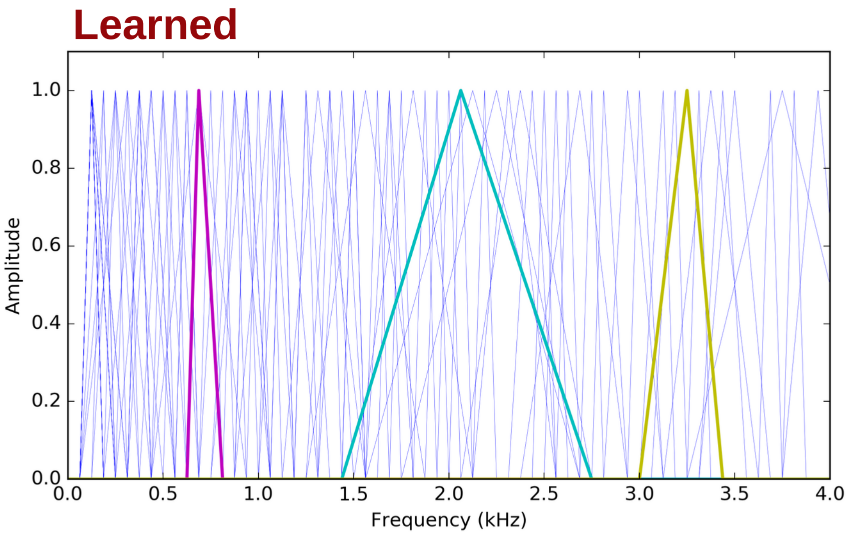
Sinc²Net: Triangular Filters

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

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

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

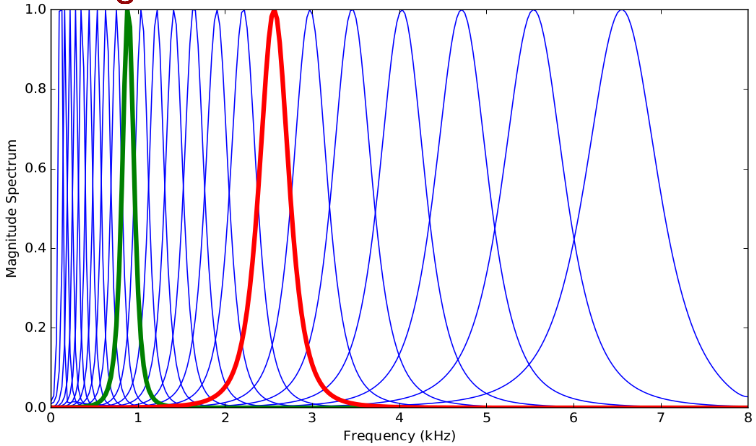
Amplitude　　Bandwidth



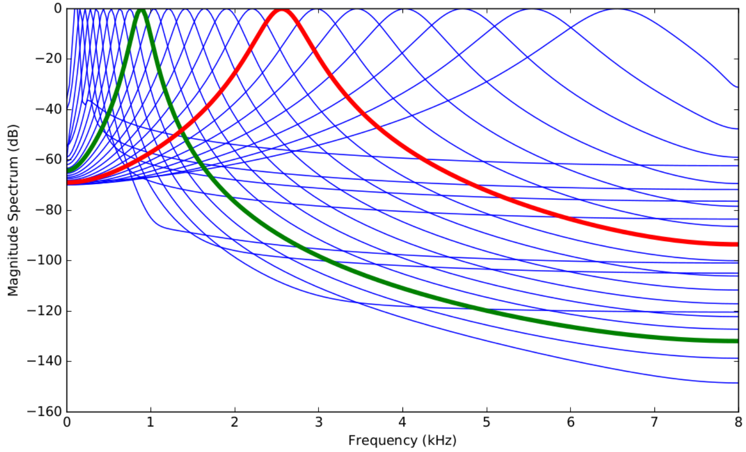
GammaNet: Gammatone Filters

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

Engineered

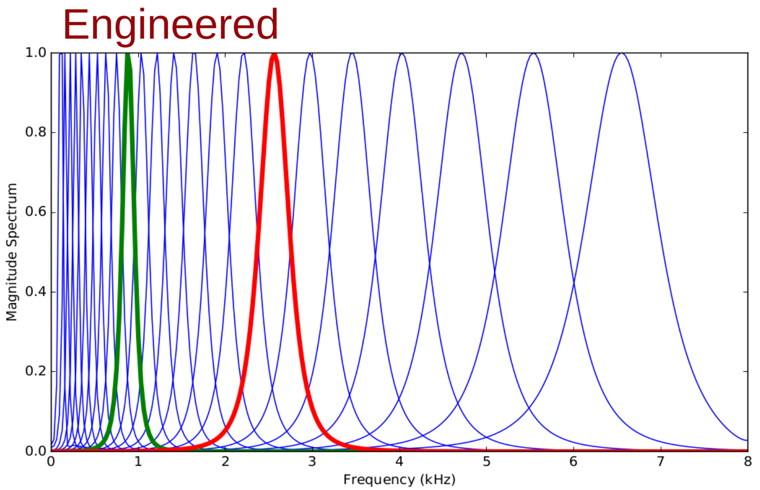


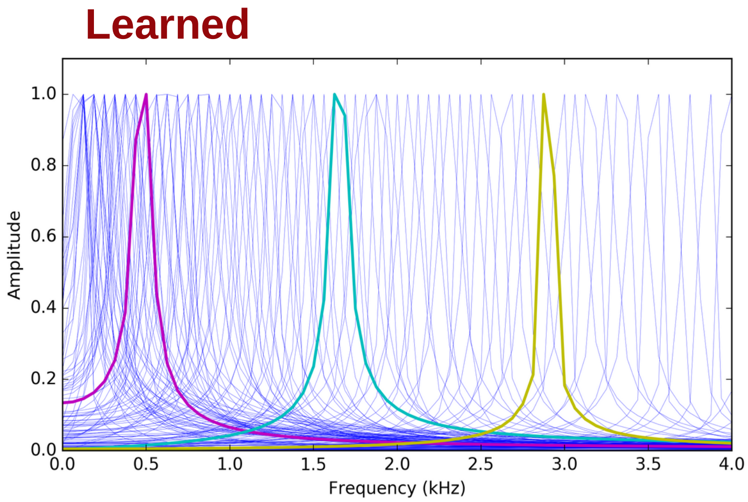
Engineered (dB)



GammaNet: Gammatone Filters

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



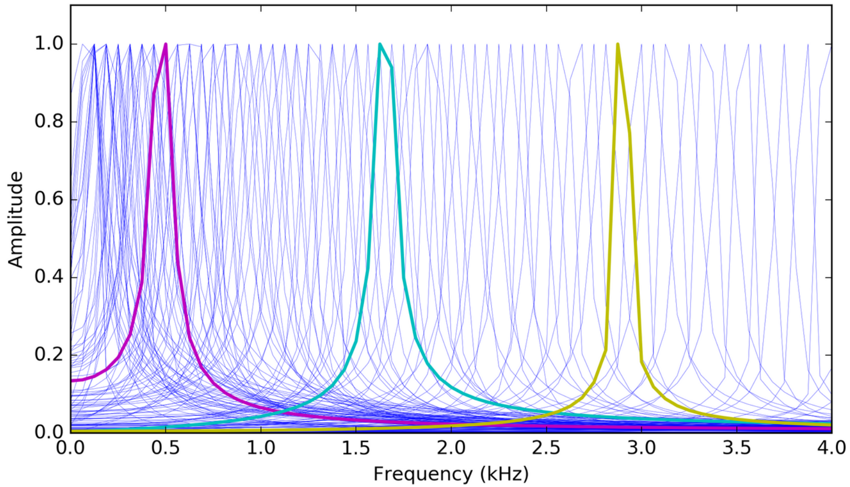


GammaNet: Gammatone Filters

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

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

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



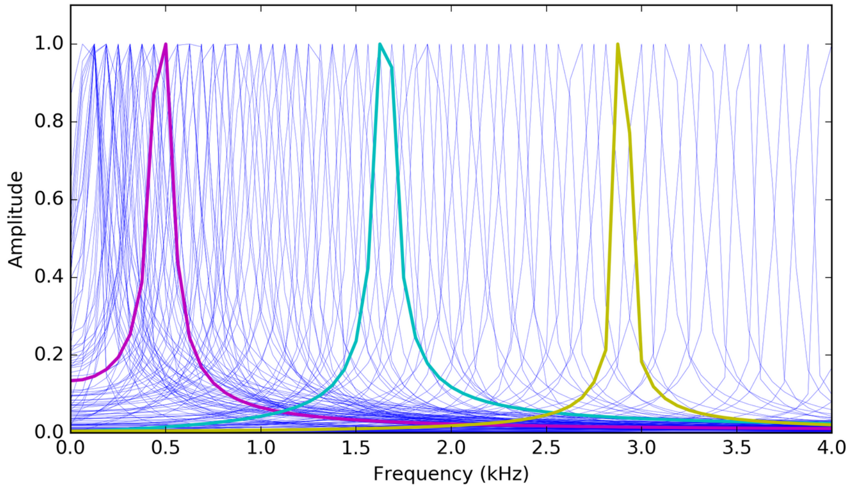
GammaNet: Gammatone Filters

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

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

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

Order



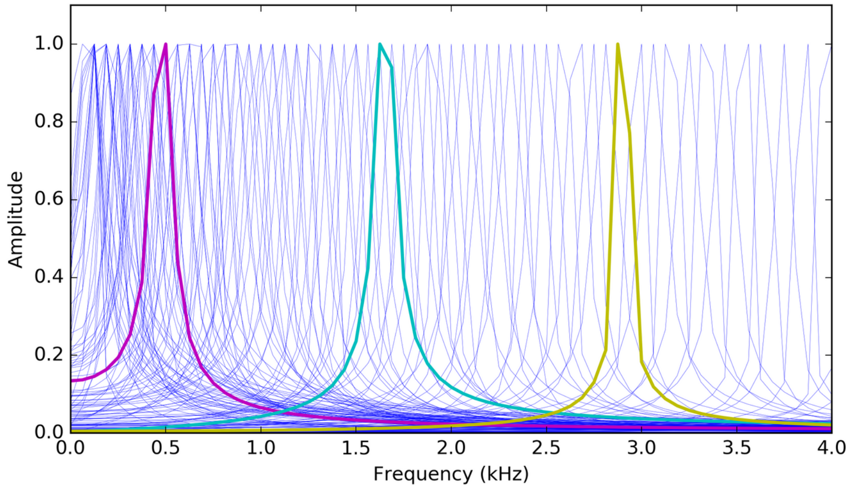
GammaNet: Gammatone Filters

- 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

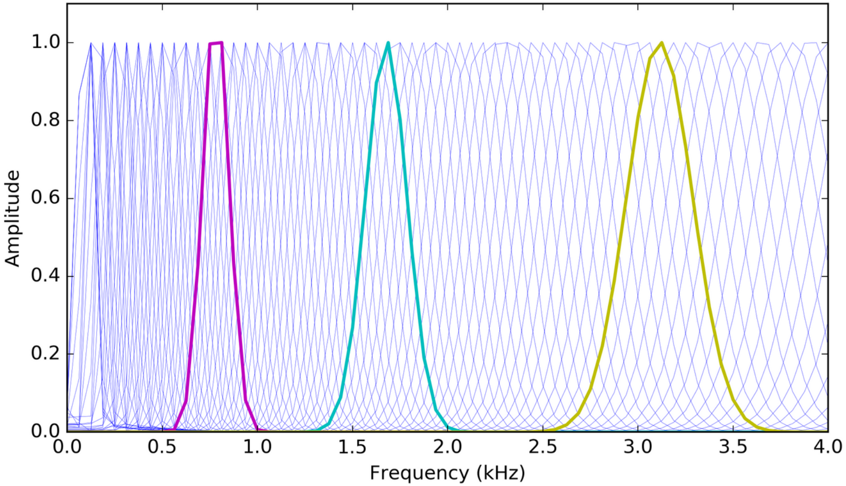


GaussNet: Gaussian Filters

- Bell-shaped Filters

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

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



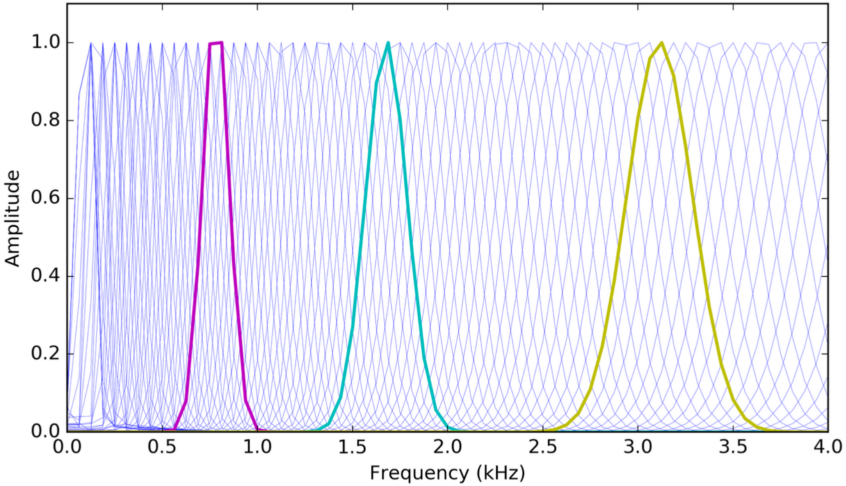
GaussNet: Gaussian Filters

- Bell-shaped Filters

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

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

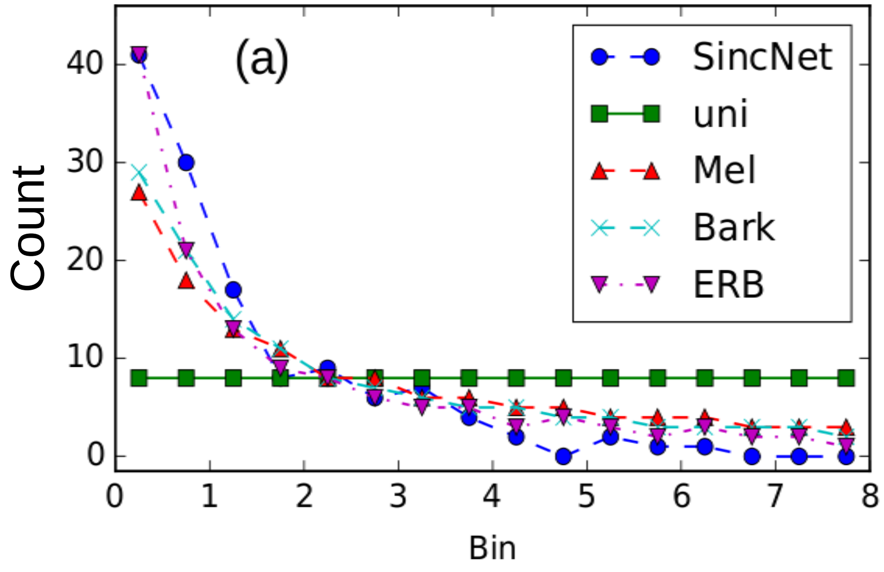
3 dB bandwidth
(Hz) of the i^{th} filter



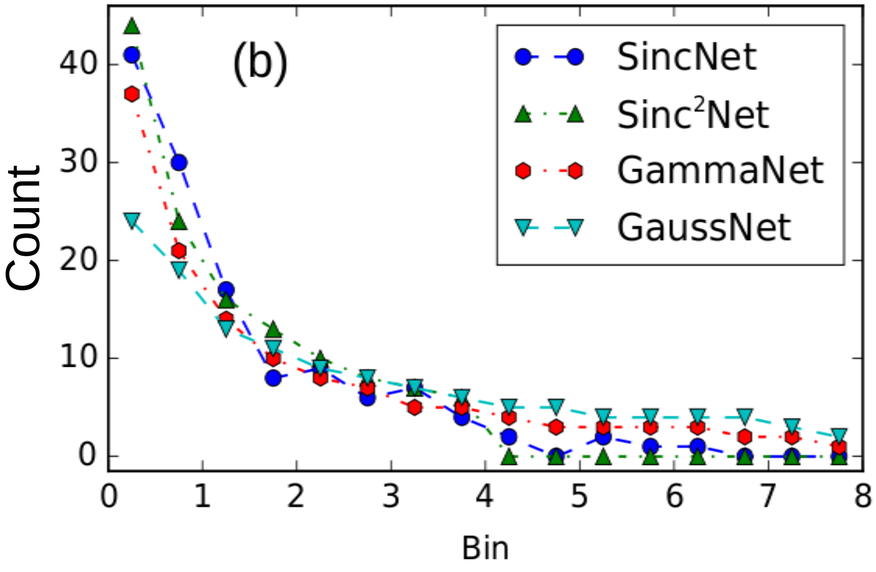
Perceptual and Statistical Studies

Filters' Centre Frequency Distribution

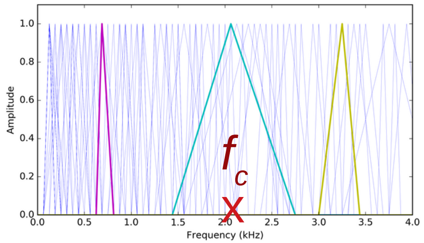
Histograms



Histograms

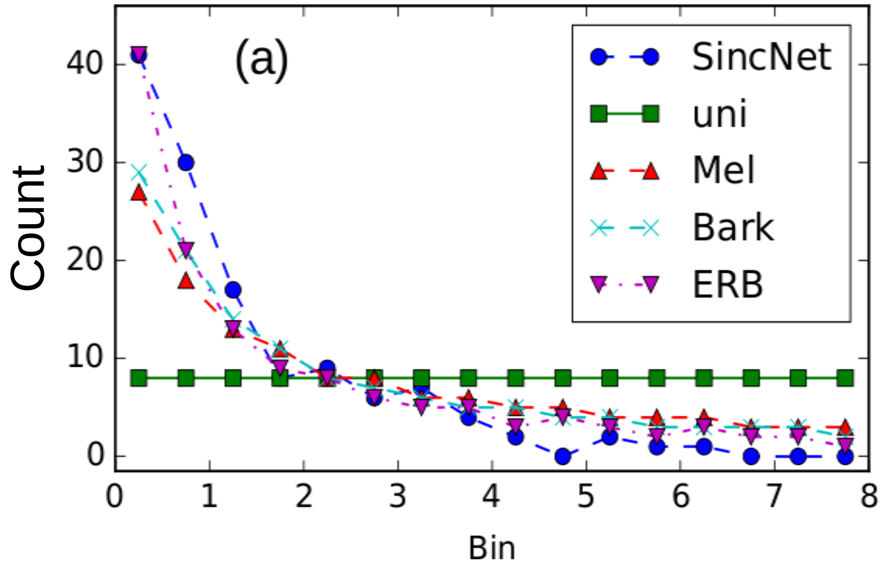


Centre frequency in kHz

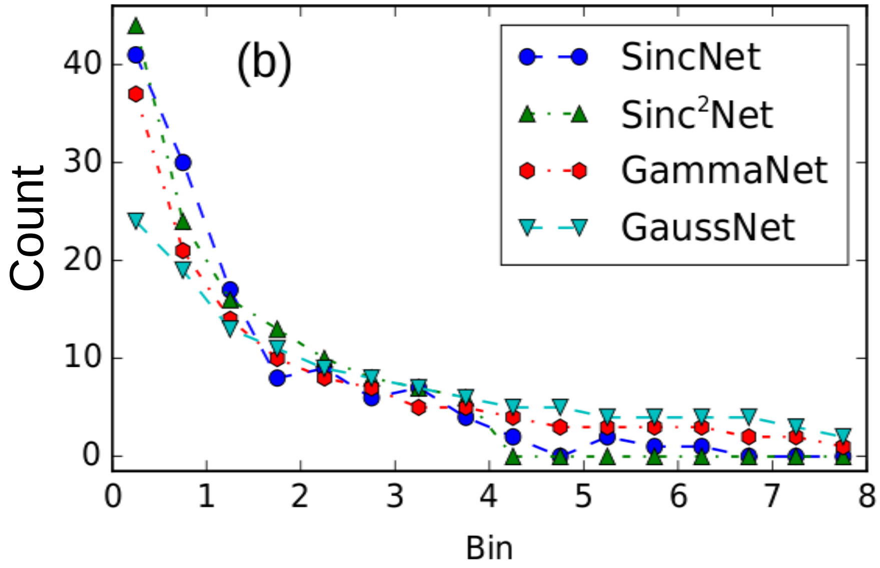


Filters’ Centre Frequency Distribution

Histograms



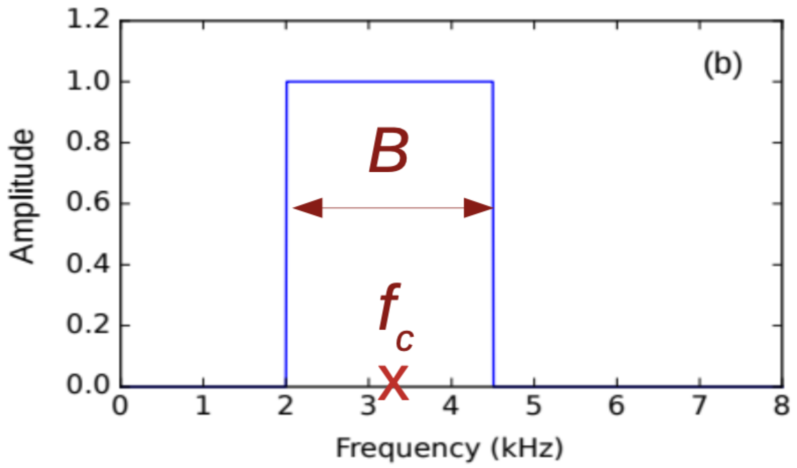
Histograms



Higher filter concentration at low frequencies (< 2 kHz).
==>> More discriminative and selective filtering ...

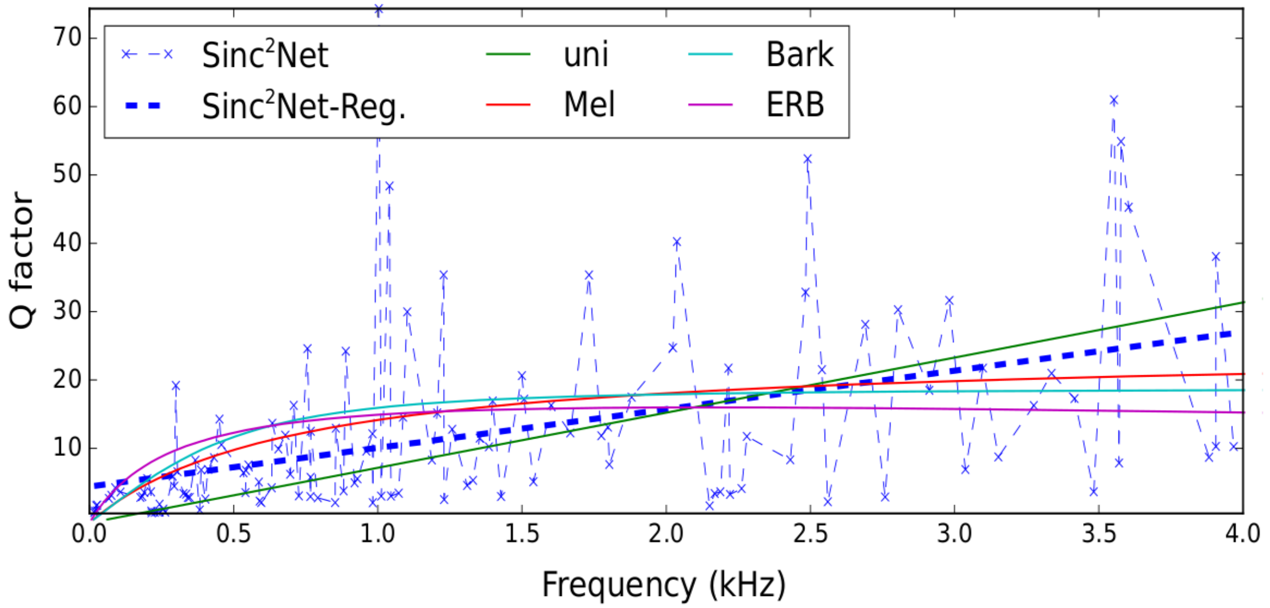
Quality Factor (Q) of the Filters

$$Q^{(i)} = \frac{f_c^{(i)}}{B^{(i)}}$$



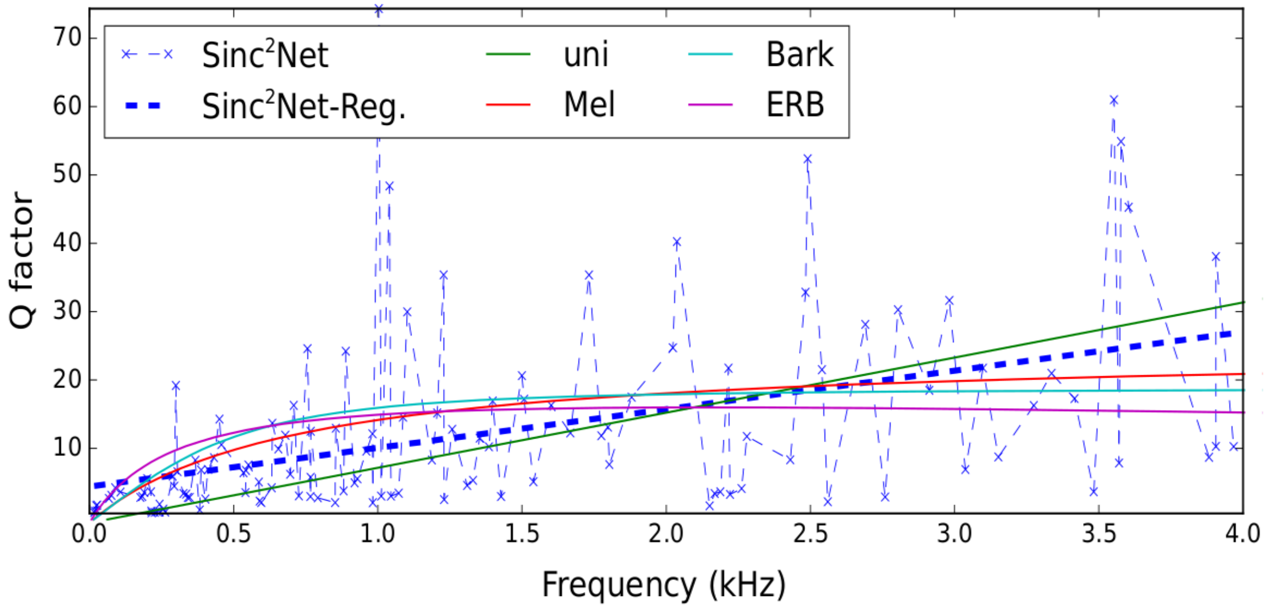
Quality Factor (Q) of the Filters

$$Q^{(i)} = \frac{f_c^{(i)}}{B^{(i)}}$$



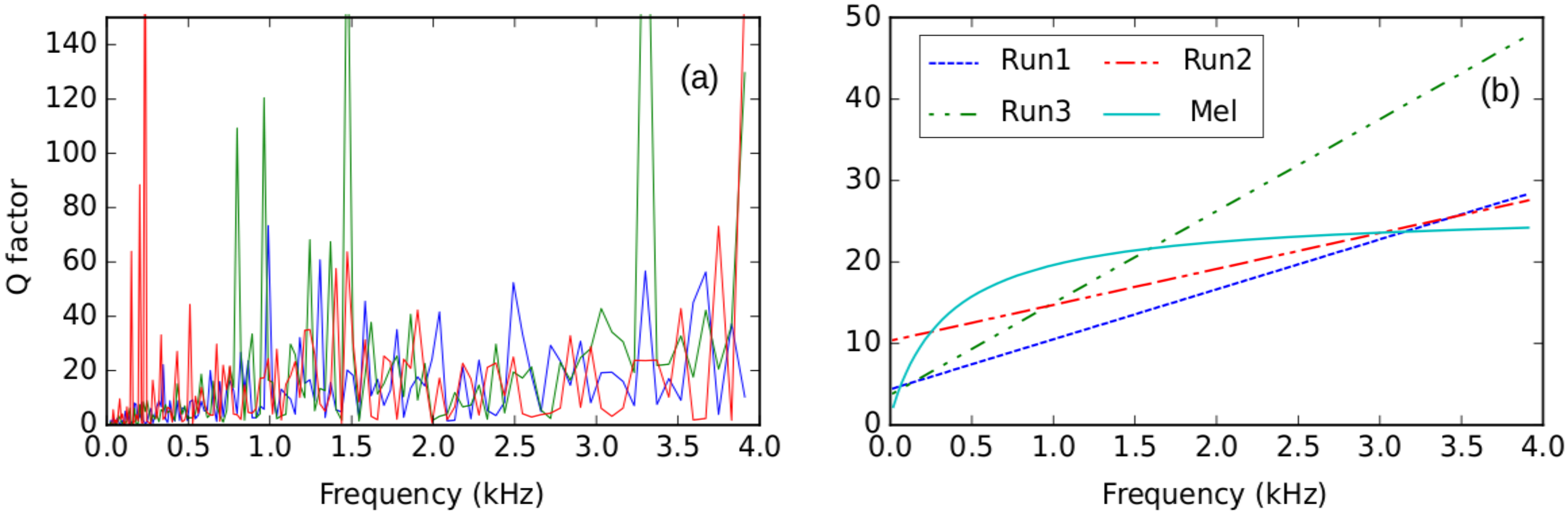
Quality Factor (Q) of the Filters

$$Q^{(i)} = \frac{f_c^{(i)}}{B^{(i)}}$$



Similar trend to the perceptual measures.

Quality Factor (Q) of the Filters



It is not a random effect ...

Gammatone Filters Order -- Perceptual Studies

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).

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

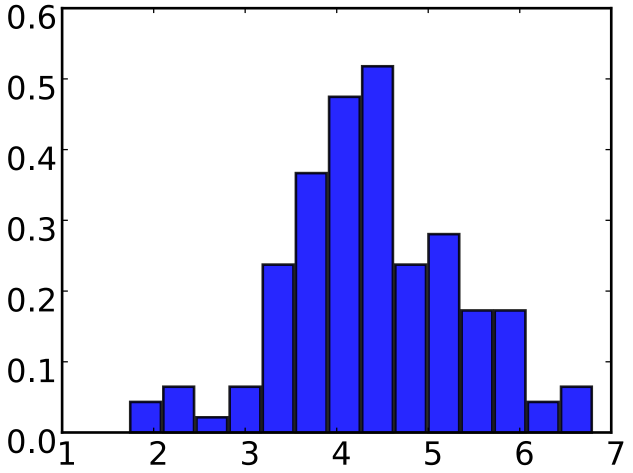
Schofield, D. (1985). Visualisations of speech based on a model of the peripheral auditory system. NPL Report DITC 62/85.

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

“Order” Histogram



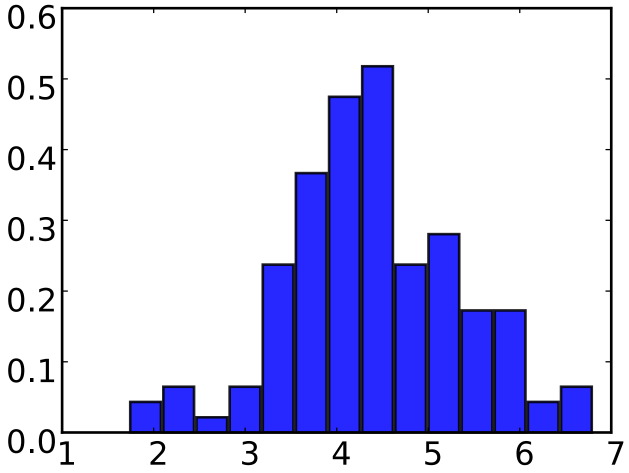
No constraint was imposed on filters order during training.

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

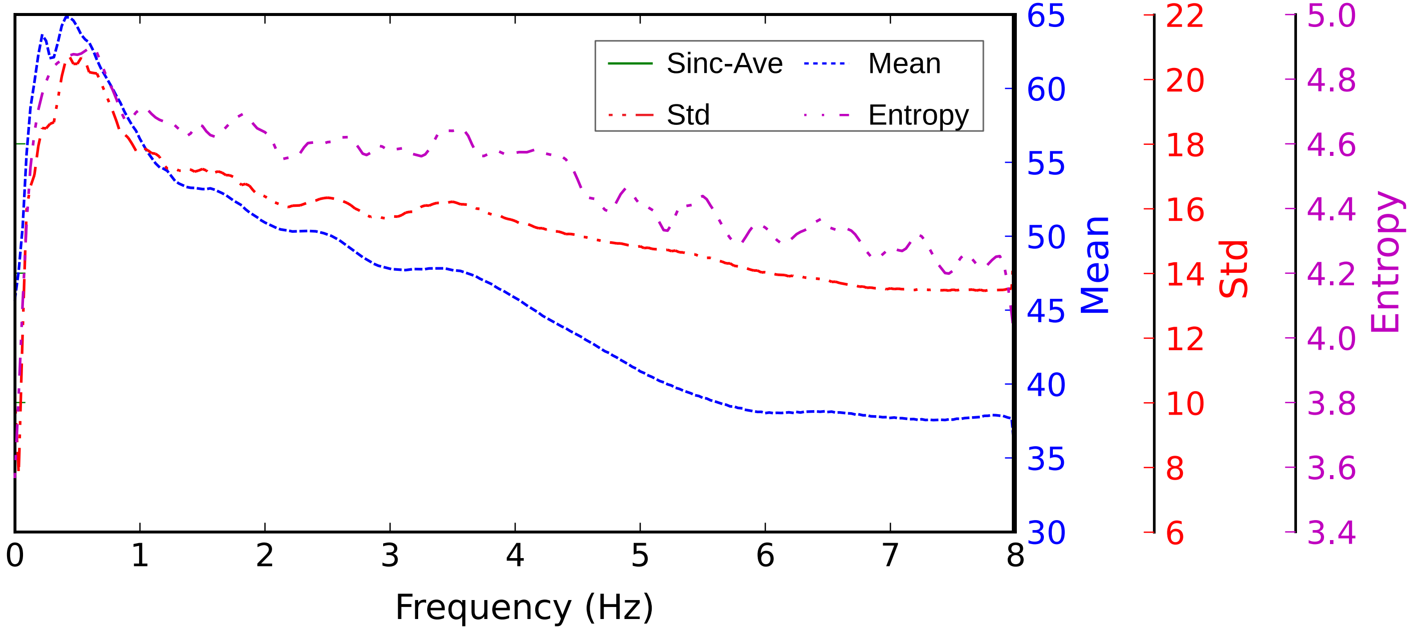
“Order” Histogram



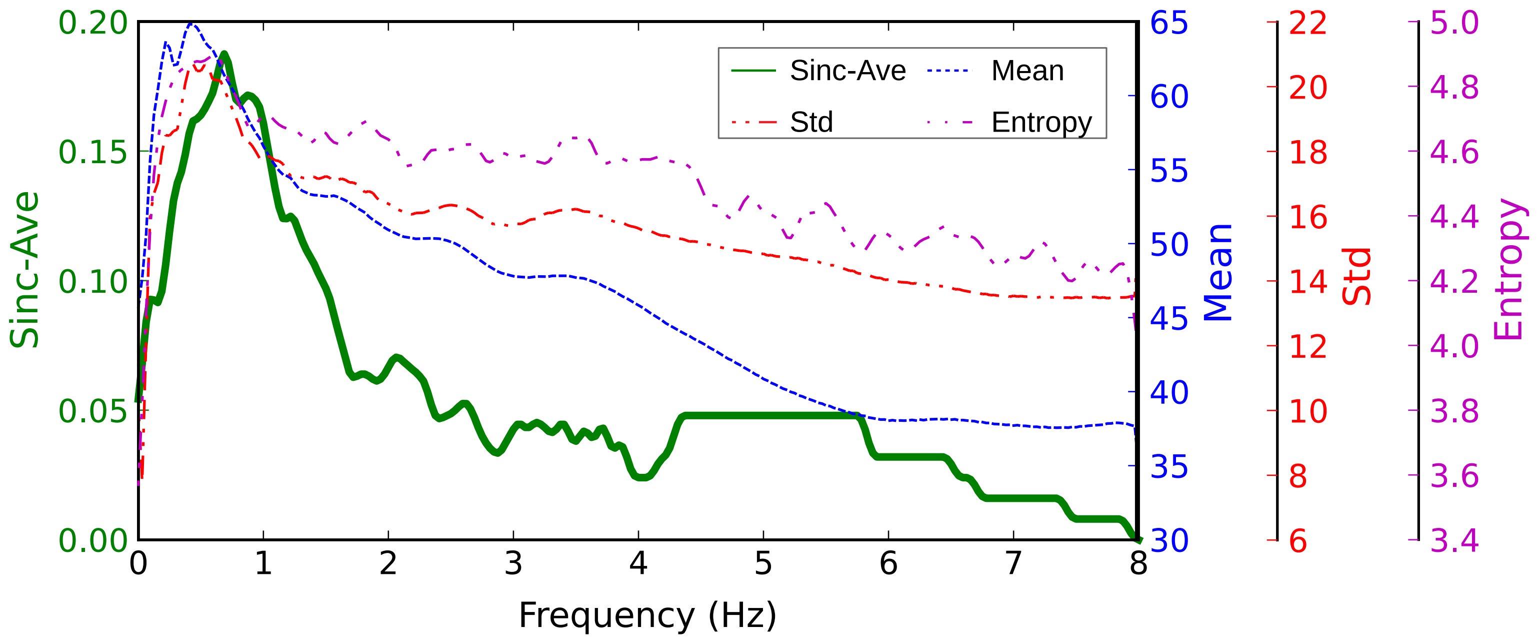
Matches with perceptual studies on human auditory system.

Statistical Properties of the Data and the Learned Filters

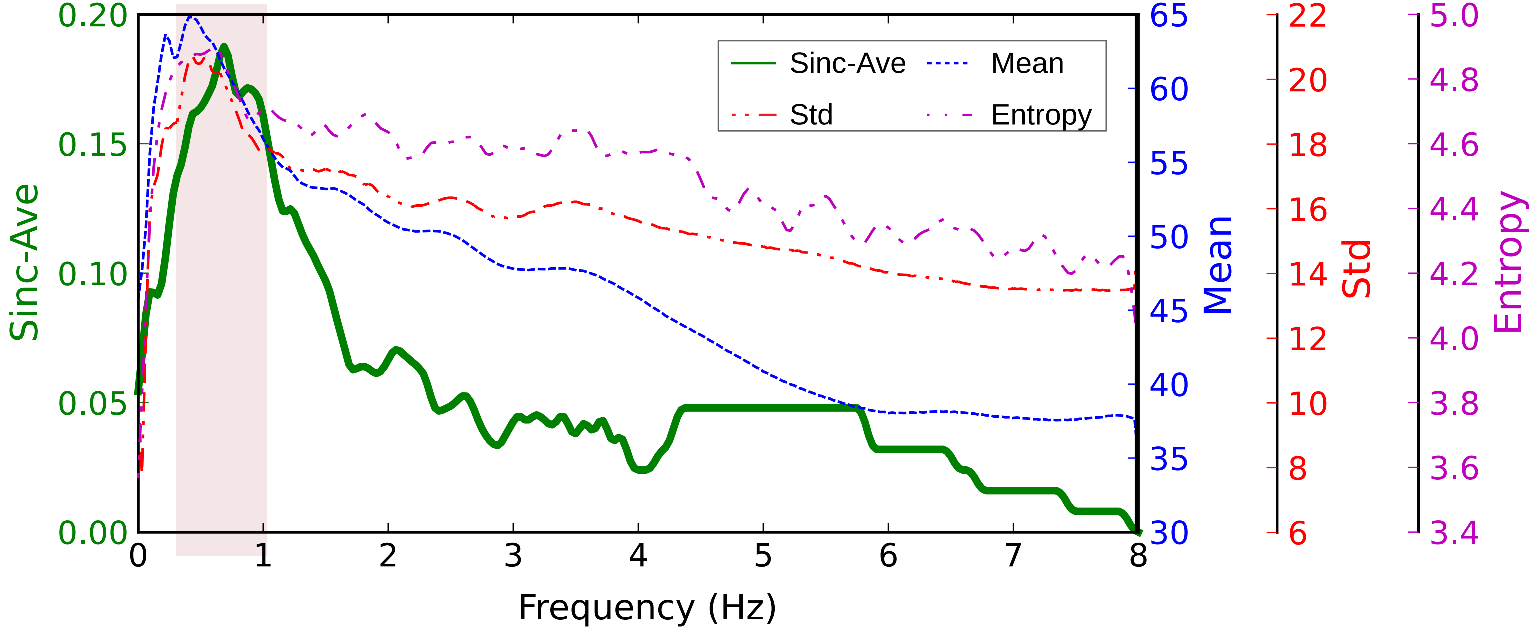
Statistical Properties of the Data and the Learned Filters



Statistical Properties of the Data and the Learned Filters



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, 1024 nodes, 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

Raw waveform models outperform log-Filterbank features.

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

Non-Parametric Parametric

Parametric Nets outperform non-parametric CNN.

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

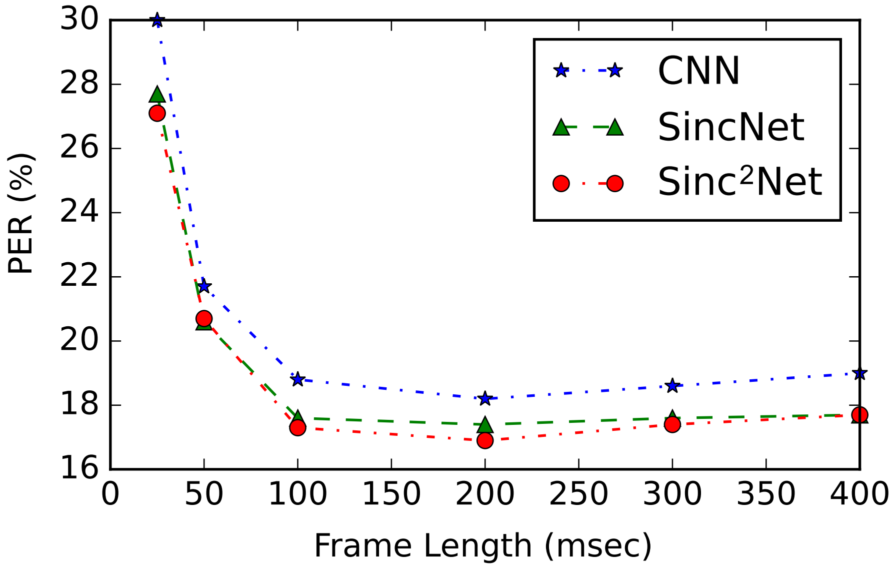
SincNet X-Nets

X-Nets outperform SincNet (also are more biologically plausible).

Frame Length Effect Investigation

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

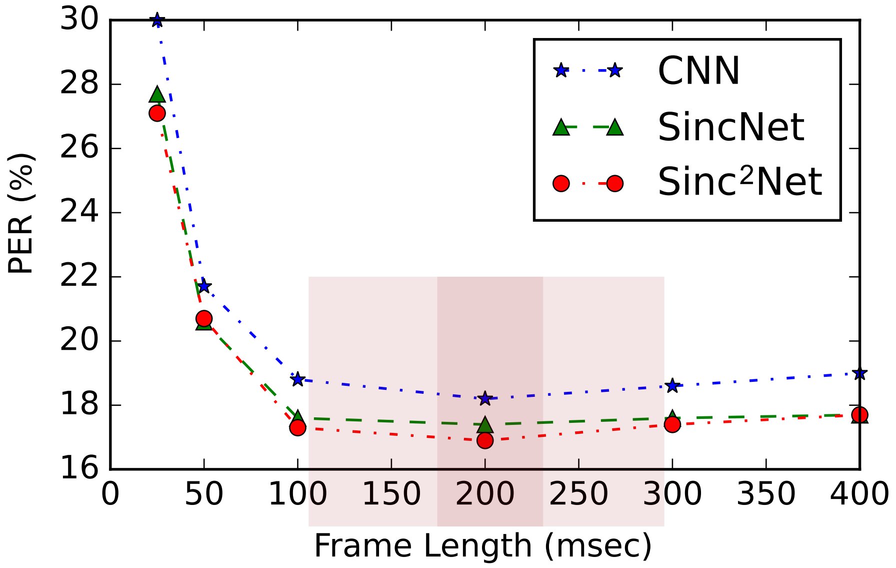
	25	50	100	200	300	400
CNN	30.0	21.7	18.8	18.2	18.6	19.0
SincNet	27.7	20.6	17.6	17.4	17.6	17.7
Sinc^{2}Net	27.1	20.7	17.3	16.9	17.4	17.7



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^2Net	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^{2}Net	27.1	20.7	17.3	16.9	17.4	17.7

- ✓ Suppressing harmful **mid-term** properties (e.g. speaker-ind. ASR)
- ✓ Preserving useful **mid-term** properties (e.g. speaker/emotion ID)
- ✗ Higher 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^2Net	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).*

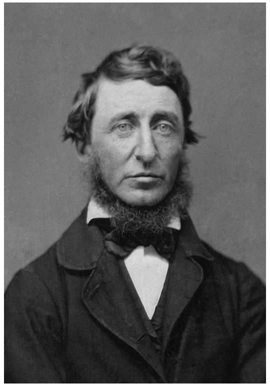
	25	50	100	200	300	400
CNN	30.0	21.7	18.8	18.2	18.6	19.0
SincNet	27.7	20.6	17.6	17.4	17.6	17.7
Sinc^{2}Net	27.1	20.7	17.3	16.9	17.4	17.7

1. Optimal Temporal Masking or Coarticulation Modelling
– Optimal combination of masker and maskee

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

Wrap-up for Part II

- Waveform acoustic modelling via convolutional layer
- A general formulation for interpretable CNNs with modulated kernel-based filters (X-Net) was derived
- Learned filters were studied statistically/perceptually
- Mid-term (~ 200ms) processing is required for raw waveform modelling through X-Nets



"Not until we are lost do we begin to understand …"

- Henry David Thoreau

That’s It!

- Thanks for Your Attention
- Q/A

- Acknowledgements:
  - Supported by EPSRC Project EP/R012180/1 (*SpeechWave*)
  - Benefited from discussion with Zoran Cvetkovic (KCL)

SincNet – Practical Considerations

- Sinc length is finite → imperfect response
  - Apply a tapered window
- Monitor the cut-off frequencies value ($0 < f_i < f_S/2$)
- Amplitude learning is not necessary
- Initialisation of Parameters (cut-off frequencies)
  - Any perceptual scale may be used, e.g. Mel, Bark, ERB
  - Using random initialisation is still fine ($0 < f_i < f_S/2$)