Some Mathematical Aspects of Deep Learning

Monika Dörfler

NuHAG, Faculty of Mathematics, University of Vienna

Peyresc, France

July 4 & 6, 2018

universität wien

Contents

1. Lecture 1: Introduction and Motivation
   - The Power of Convolutional Neural Networks
   - What is learning in a mathematical sense? - a simple example
2. Lecture 2: Elements of Mathematical Learning Theory
   - Approximation Error and Sample Error
   - Generalization
3. Lecture 3: Approximation by (Deep) Neural Networks and the Idea of Locality in CNNs
   - Approximation Results for Shallow and Deep Networks
   - CNNs: Extracting Local Information
4. Lecture 4: Features - Invariance, Symmetry and Stability
   - Features for Audio: (Mel-)Spectrogram and Scattering Transforms
5. Lecture 5: Adaptivity - what do we know, what must we learn?

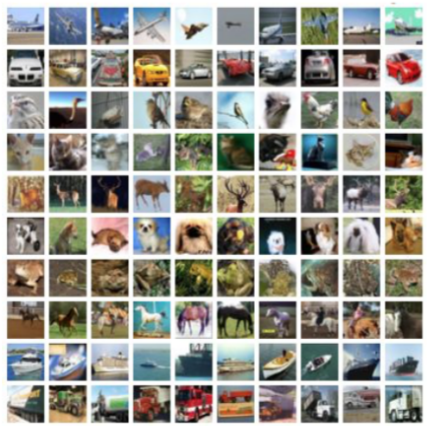
1. Lecture 1: Introduction and Motivation
   - The Power of Convolutional Neural Networks
   - What is learning in a mathematical sense? - a simple example
2. Lecture 2: Elements of Mathematical Learning Theory
   - Approximation Error and Sample Error
   - Generalization
3. Lecture 3: Approximation by (Deep) Neural Networks and the Idea of Locality in CNNs
   - Approximation Results for Shallow and Deep Networks
   - CNNs: Extracting Local Information
4. Lecture 4: Features - Invariance, Symmetry and Stability
   - Features for Audio: (Mel-)Spectrogram and Scattering Transforms
5. Lecture 5: Adaptivity - what do we know, what must we learn?

The Power of Convolutional Neural Networks

- Convolutional Neural Networks (CNNs)
  introduced in Image Processing, e.g. for image classification.

Task: Recognize hand-written digits

Task: Recognize photographed objects
(with a fixed set of possible answers)



The Power of Convolutional Neural Networks

- Convolutional Neural Networks (CNNs) introduced in Image Processing, e.g. for image classification.

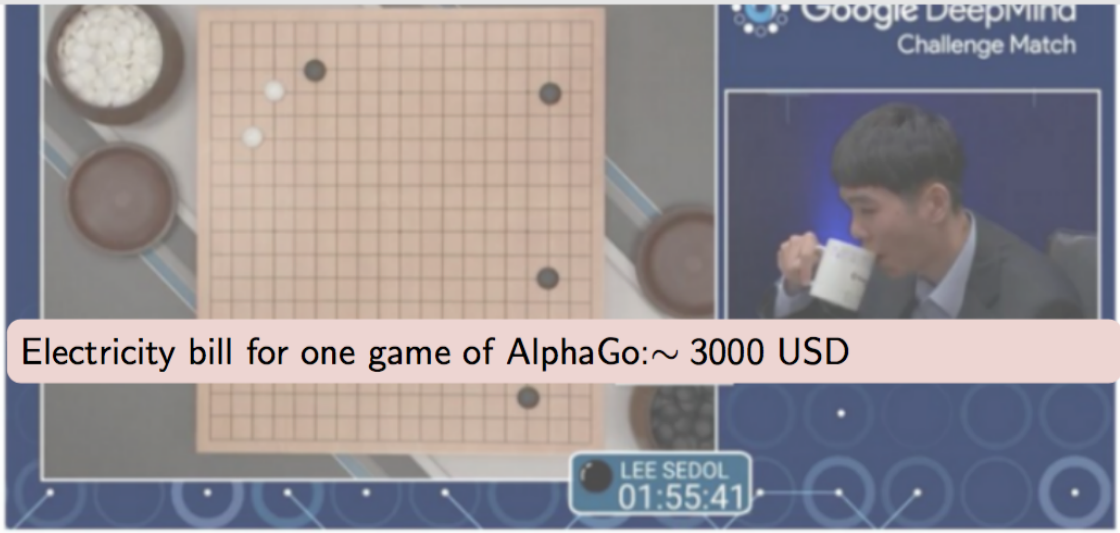
  

  LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. *Gradient-based learning applied to document recognition.* . Proceedings of the IEEE, 86(11), 2278 – 2324. (1998).

- My Project: SALSA
  (Semantic Annotation by Learned, Structured and Adaptive signal representations) Goal of SALSA is to bridge the *semantic gap* in music information research (MIR) by using adaptive and structured signal representations.

CNNs seem to be able "to do anything" - but at what cost?

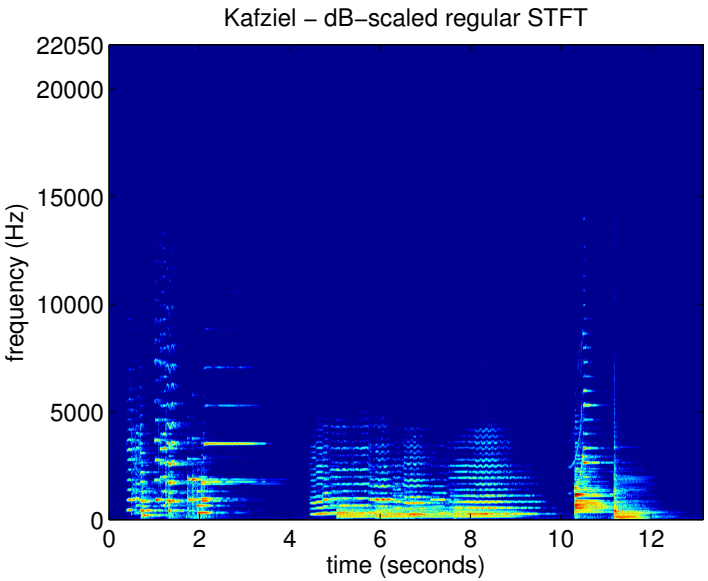
Be More Efficient



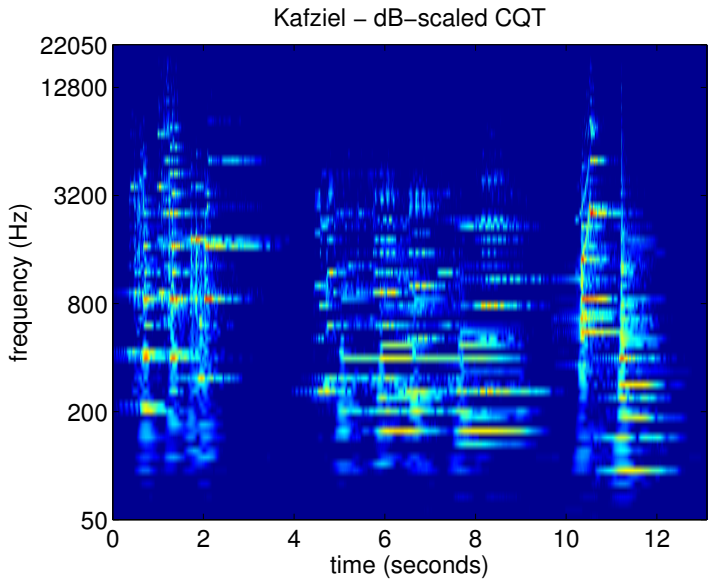
AlphaGO	Lee Se-dol
1202 CPUs, 176 GPUs,	1 Human Brain,
100+ Scientists.	1 Coffee.

The Power of Convolutional Neural Networks

- For complex problems (with "semantic flavour") need huge architectures, thus a lot of data points. What can we do to reduce amount of necessary data to learn?
- In Image Processing usually learn directly from "raw data", i.e., no pre-processing (feature engineering) takes place.
- In Audio, "end-to-end" learning is much less common - instead, some kind of FFT-based *time-frequency* signal representation is used as a pre-processing step; it will turn the audio signal into an image ...

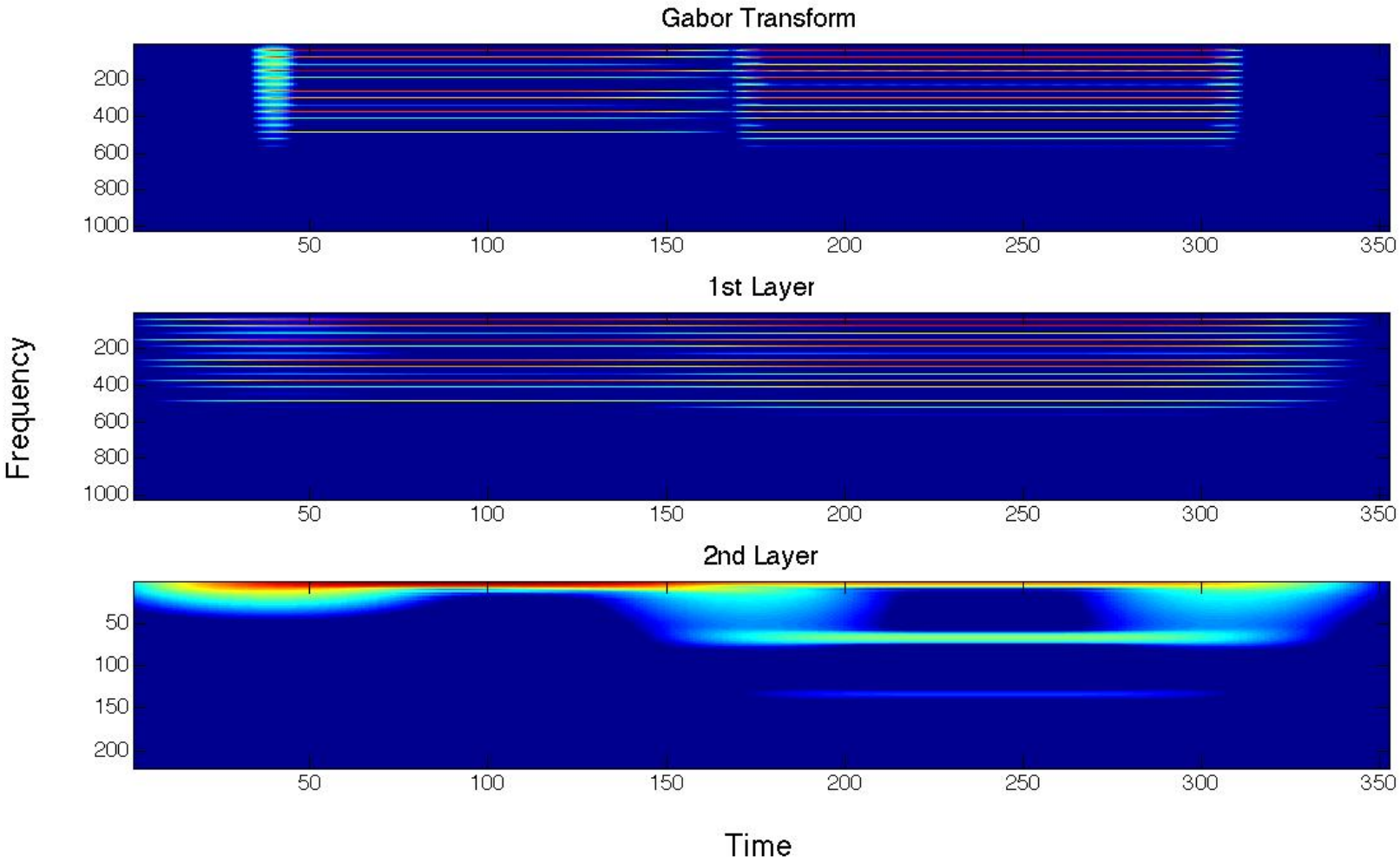


(a) Standard Spectrogram of music excerpt



(b) CQ-Spectrogram of music excerpt

- Which representation would a Neural Network learn?
- Can end-to-end learning improve performance if sufficient amount of data is available?
- Can a representation which encodes known symmetries reduce necessary network size?



Gabor Scattering of simple music signal: separating signal characteristics

Linear Regression - a simple example

Learning from data: look for a function $f : \mathcal{X} \mapsto \mathcal{Y}$, which describes with sufficient accuracy the "nature of data". ... Learning means "improving with experience" (Mitchell, Machine Learning, 1997)

Two important examples:

1. Regression: $\mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \mathbb{R}$
2. Classification: $\mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{c_1, \dots, c_n\}$, $c_j \in \mathbb{R}$

look at the regression problem in more detail!!

Linear Regression - a simple example

We can observe three central questions:

1. What is the nature of the function f we need to learn? $\rightarrow$ Choice of model or hypothesis class $\mathcal{H}$
2. Can I determine a close approximation of the "best" function $f_{\mathcal{H}}$ from the available data points? $\rightarrow$ Sampling problems.
3. How will the learned function f perform on unseen data? $\rightarrow$ generalization properties.

1. Lecture 1: Introduction and Motivation
   - The Power of Convolutional Neural Networks
   - What is learning in a mathematical sense? - a simple example
2. Lecture 2: Elements of Mathematical Learning Theory
   - Approximation Error and Sample Error
   - Generalization
3. Lecture 3: Approximation by (Deep) Neural Networks and the Idea of Locality in CNNs
   - Approximation Results for Shallow and Deep Networks
   - CNNs: Extracting Local Information
4. Lecture 4: Features - Invariance, Symmetry and Stability
   - Features for Audio: (Mel-)Spectrogram and Scattering Transforms
5. Lecture 5: Adaptivity - what do we know, what must we learn?

Some Probability Theory

For mathematical description, need some elements from probability theory...

1. In learning theory, assume that data are "drawn according to some (unknown) probability density(measure)"

$$\rho : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$$

2. For any function $f : \mathcal{X} \mapsto \mathcal{Y}$, define the "true error", "expected risk":

$$\mathcal{E}(f) = \mathcal{E}_\rho(f) := \int_{\mathcal{X}\times\mathcal{Y}} (f(x) - y)^2 d\rho(x, y)$$

Which function f minimizes $\mathcal{E}(f)$?

Some Probability Theory

For mathematical description, need some elements from probability theory...

1. From ρ, derive marginal measures $\rho_{\mathcal{X}}, \rho_{\mathcal{Y}}$ and conditional measures $\rho(y|x), \rho(x|y)$.
2. Then for any random variable φ defined on $\mathcal{X} \times \mathcal{Y}$, we can write:

$$\int_{\mathcal{X}\times\mathcal{Y}} \varphi(x,y) d\rho(x,y) = \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} \varphi(x,y) d\rho(y|x) \right) d\rho_{\mathcal{X}}(x)$$

3. The regression function f_ρ, given by

$$f_\rho(x) := \int_{\mathcal{Y}} y \, d\rho(y|x)$$

minimizes $\mathcal{E}(f)$!

We show some facts...

1. The expected risk of f_ρ is given by the integral of its quadratic loss; this is σ_ρ^2, the "condition number" of ρ.

2. For any function $f : \mathcal{X} \mapsto \mathcal{Y}$ we have

$$\mathcal{E}(f) = \int_{\mathcal{X}} (f(x) - f_\rho(x))^2 d\rho_{\mathcal{X}}(x) + \sigma_\rho^2$$

Now consider sampling!

1. Let $\mathcal{Z}_m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be a *sampling set* of examples/data points, drawn independently according to ρ.
2. The empirical error (or risk) of f with respect to $\mathcal{Z}_m$ is defined as
$$\mathscr{E}_{\mathcal{Z}}(f) = \frac{1}{m}\sum_{i=1}^{m}(f(x_i) - y_i)^2.$$
3. *Defect Function or Generalization Error:* $L_{\mathcal{Z}}(f) = \mathscr{E}(f) - \mathscr{E}_{\mathcal{Z}}(f)$.

Now consider a model - in other words, a *Hypothesis Space* $\mathcal{H} \subseteq C(\mathcal{X})$!

1. Optimal problem solution within $\mathcal{H}$:

$$f_{\mathcal{H}} := \operatorname{argmin}_{f \in \mathcal{H}} \mathcal{E}(f) = \operatorname{argmin}_{f \in \mathcal{H}} \int_x (f - f_\rho)^2 d\rho_{\mathcal{X}}$$

2. $f_{\mathcal{H}}$ - target function
3. Empirical target function: $f_{\mathcal{H},\mathcal{Z}} := \operatorname{argmin}_{f \in \mathcal{H}} \mathcal{E}_{\mathcal{Z}}(f)$
4. Error in $\mathcal{H}$: $\mathcal{E}_{\mathcal{H}}(f) = \mathcal{E}(f) - \mathcal{E}(f_{\mathcal{H}}) \geq 0$, for $f \in \mathcal{H}$.

Proposition

Let a Hypothesis class $\mathcal{H}$ and a sampling set $\mathcal{Z}$ be given and let $f_{\mathcal{H},\mathcal{Z}}$ be the minimizer of the empirical risk $\mathcal{E}_{\mathcal{Z}}$. Then the expected error of $f_{\mathcal{H},\mathcal{Z}}$ is given by

$$\mathcal{E}(f_{\mathcal{H},\mathcal{Z}}) = \mathcal{E}_{\mathcal{H}}(f_{\mathcal{H},\mathcal{Z}}) + \mathcal{E}(f_{\mathcal{H}})$$

- $\mathcal{E}_{\mathcal{H}}(f_{\mathcal{H},\mathcal{Z}})$ ... *Sample Error*
- $\mathcal{E}(f_{\mathcal{H}})$ ... *Approximation Error*

Formalization of the BIAS-VARIANCE Trade-off!

Definition

Let a Hypothesis class $\mathcal{H}$ be given and equipped with a metric. Then, for some $\varepsilon > 0$, the *covering number* $\mathcal{N}(\mathcal{H}, \varepsilon)$ is given by the minimal number $l \in \mathbb{N}$, such that there exist l disks of radius ε, which cover $\mathcal{H}$.

Using covering numbers, we can now give bounds (in probability) for the defect $L_{\mathcal{Z}}(f) = \mathcal{E}(f) - \mathcal{E}_{\mathcal{Z}}(f)$ and the sample error $\mathcal{E}_{\mathcal{H}}(f_{\mathcal{H},\mathcal{Z}}) = \mathcal{E}(f_{\mathcal{Z}}) - \mathcal{E}(f_{\mathcal{H}})$.

Challenges: understanding why CNNs perform so well

- Generalization is not well-understood in deep learning; data structure (geometry) as well as invariances and symmetries with respect to the learning task (semantic) seem to play an important role.
  "Mystery of generalization in deep learning"?
- Similarly the design of architectures often based on previous experience; what are we willing to pay and how do we distribute the available weights between number of layers (depth) and size of layers (width)?

In Lectures 4/5, we will see how these questions are related to signal representations - in particular: how to balance adaptivity vs. fixed parameters?

Challenges: understanding why CNNs perform so well

K.Kawaguchi, L.P.Kaelbling and Y.Bengio.
Generalization in deep learning.
arXiv:1710.05468, 2017.

suggest an approach to generalization results for deep learning, which is more motivated by practical use..

Proposition

Let R_{val} be a validation data set and F_{val} a finite set of models, proposed independently from R_{val}. Further, set $z_{f,i} = \mathscr{E}(f) - (f(x_i) - y_i)^2$. If $\mathbb{E}(z_{f,i}^2) \leq \gamma^2$ and $|z_{f,i}| \leq C$ a.e. $\forall(f,i)$, then $\forall \delta > 0$ with probability at least $1-\delta$, have for all $f \in F_{val}$:

$$\mathscr{E}(f) \leq \mathscr{E}_{R_{val}}(f) + \frac{2C\log(\frac{|F_{val}|}{\delta})}{3m_{val}} + \left(\frac{2\gamma^2\log(\frac{|F_{val}|}{\delta})}{m_{val}}\right)^{1/2}$$

1. Lecture 1: Introduction and Motivation
   - The Power of Convolutional Neural Networks
   - What is learning in a mathematical sense? - a simple example
2. Lecture 2: Elements of Mathematical Learning Theory
   - Approximation Error and Sample Error
   - Generalization
3. Lecture 3: Approximation by (Deep) Neural Networks and the Idea of Locality in CNNs
   - Approximation Results for Shallow and Deep Networks
   - CNNs: Extracting Local Information
4. Lecture 4: Features - Invariance, Symmetry and Stability
   - Features for Audio: (Mel-)Spectrogram and Scattering Transforms
5. Lecture 5: Adaptivity - what do we know, what must we learn?

Most basic building block in a general neural network may be written as

$$x_{n+1} = \sigma(A_n x_n + b_n)$$

- x_n – data vector (array) in the n-th layer
- A_n – linear operator
- b_n – vector of biases in the n-th layer
- nonlinearity σ (applied component wise).

For convolutional layers of CNNs: A_n are block-Toeplitz.
General A_n: dense layers.
Parameters (weights) $\theta = (A_n, b_n)_{n=1}^{N_p}$ are learned by gradient descent algorithms.

Notation:

- $W_n = A_n x_n + b_n$ (affine mapping)
- d – input dimension
- x_n – number of layers
- N_l – width of each layer (number of neurons in each layer)
- The Network:

$$\Phi(x_0) = W_L \sigma(W_{L-1} \sigma(\dots (W_2 \sigma(W_1(x_0)))))$$

- $N(\Phi) := \sum_{l=1}^{L} N_l$ –over-all number of neurons
- $W(\Phi) := \sum_{l=1}^{L} \|A_l\|_0 + \|b_l\|_0$ –over-all number of weights ("connectivity")

Proposition (Cybenko, G. (1989), K.Hornik (1991))

Even shallow networks are universal approximators!
Let $\sigma : \mathbb{R} \mapsto \mathbb{R}$ be a non-polynomial, continuous function and $K \subseteq \mathbb{R}^d$. Then, for all $\varepsilon > 0$ and any continuous function f on K, there exist $N \in \mathbb{N}$, $c_i, b_i \in \mathbb{R}$ and $w_i \in \mathbb{R}^d$, such that for all $x \in K$

$$|f(x) - \sum_{i=1}^{N} c_i \sigma \left(w_i^T x + b_i \right)| < \varepsilon$$

Proof uses either Stone-Weierstrass Theorem (Hornik) or Hahn-Banach Theorem (Cybenko).

So why use deep networks at all?

"The number of linear regions grows exponentially with depth of a neural network"

Montufar, G., Pascanu, R., Cho, K. and Bengio, Y.
On the Number of Linear Regions of Deep Neural Networks.
Advances in Neural Information Processing Systems 27, 2014
papers.nips.cc/paper/5422-on-the-number-of-linear-regions-of-deep-neural-networks.pdf

"There are deep neural networks of size n that can only be approximated by shallow networks whose size is exponential in n."

Telgarsky, M.
Benefits of depth in neural networks.
Conference on Learning Theory (COLT), (2016), 1517-1539.
https://arxiv.org/abs/1509.08101

Depth before width: Some Statements and References

So why use deep networks at all?

"The complexity necessary to achieve a certain accuracy is expontially smaller when using deep nets for approximating compositional functions"

Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B. and Liao,Q.
Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review
International Journal of Automation and Computing, 2017, 14/5

Caveat: here the statement only holds, if the deep NN has the same "compositional" structure as the function to be approximated.
However, the result emphasizes importance of locality, as e.g. featured by convolutional NNs!

"All affine representation systems are (effectively) representable by neural networks."

Bölcskei, H., Grohs, P., Kutyniok, G. and Petersen, P.
Optimal approximation with sparsely connected deep neural networks.
arXiv preprint arXiv:1705.01714, 2017.

Uses a different, information theoretic definition of complexity of a function class (based on work by D. Donoho):
Let $d \in \mathbb{N}$ and $\mathcal{C} \subset L^2(\Omega)$ a function class. For $\varepsilon > 0$ the minimax code length of $\mathcal{C}$ is

$$L(\varepsilon, \mathcal{C}) := \min\{\ell \in \mathbb{N} : \exists (E, D) : \sup_{f \in \mathcal{C}} \|D(E(f)) - f\|_{L^2(\Omega)} \leq \varepsilon\}$$

Furthermore, $\gamma^*(\mathcal{C}) := \inf\{\gamma \in \mathbb{R} : L(\varepsilon, \mathcal{C}) = O(\varepsilon^{-\gamma})\}$ describes the asymptotic behaviour of optimal encoder - decoder pairs.

Depth before width: Some Statements and References

- For some known $\mathcal{C}$ there exists a specific optimal dictionary that achieves the optimal tradeoff between sparsity (..codelength!) and approximation error ...
- Examples: Textures $\leftrightarrow$ Gabor frames (JPEG), point singularities $\leftrightarrow$ wavelets (JPEG2000), line/hyperplane singularities $\leftrightarrow$ ridgelets, curved/hypersurface singularities$\leftrightarrow$(α-)curvelets.

Depth before width: Some Statements and References

So why use deep networks at all?

"Let $\Omega \subseteq \mathbb{R}^d$ be open and connected and let $f \in C^3(\Omega)$ be not affine-linear. Then, for a NN Φ with L layers, there is a constant $C_f > 0$ such that for every $p \in [1, \infty]$:

$$\|f - \Phi\|_p \geq C_f \cdot \max\{(N(\Phi) - 1)^{-2L}, (W(\Phi) + d)^{-2L}\}."$$

Petersen, P. and Voigtlaender, F.,
Optimal approximation of piecewise smooth functions using deep ReLU neural networks.
arXiv preprints, arxiv.org/abs/1709.05289, 2017..

Data Structure?

What about the structure of the data?

- Geometric information in data vs. semantic information in learning problem.
- Features: Design or Learning?
- Convolutional layers as feature generating layers...

The following building blocks define a CNN:

- Convolution:

$$S * w(m,n) := \sum_{m'} \sum_{n'} S(m',n') w(m-m', n-n')$$

- Pooling: $P_p^{K,L} : \mathbb{R}^{M \times N} \mapsto \mathbb{R}^{\frac{M}{K} \times \frac{N}{L}}$

$$(P_p^{K,L} S_0)(m,n) = \|v_{S_0}^{m,n}\|_p,\ m = 1,\dots,\frac{M}{K}, n = 1,\dots,\frac{N}{L},$$

$$v_{S_0}^{m,n} = S_0[(m-1)\cdot K + 1,\dots,m\cdot K; (n-1)\cdot L + 1,\dots,n\cdot L]$$

- A nonlinearity $\sigma : \mathbb{R} \mapsto \mathbb{R}$, whose action is always to be understood component-wise.

- $S_n \in \mathbb{R}^{M_n \times N_n \times K_n}$: Input array to convolutional layer N
  K_n is the number of feature maps In layer n.
- Output of convolutional layer $n+1$ with convolutional kernels
  $w_{n+1} \in \mathbb{R}^{K_{n+1} \times K_n \times M_n \times N_n}$:

$$S_{n+1}(k_{n+1}) = P_\infty^{A_n, B_n} \sigma \left[\left(\sum_{k_n=1}^{K_n} S_n(k_n) * w_{n+1}(k_{n+1}, k_n) \right) + b^{k_{n+1}} \otimes \mathbf{1} \right]$$

- $\mathbf{1}$: all-ones array of size $M_n \times N_n$
  $b^{k_{n+1}} \in \mathbb{R}^{K_{n+1}}$
  $S_{n+1}(k_{n+1}) \in \mathbb{R}^{M_n/A_n \times N_n/B_n}$ for $k_{n+1} = 1, \dots, K_{n+1}$.

1. Lecture 1: Introduction and Motivation
   - The Power of Convolutional Neural Networks
   - What is learning in a mathematical sense? - a simple example
2. Lecture 2: Elements of Mathematical Learning Theory
   - Approximation Error and Sample Error
   - Generalization
3. Lecture 3: Approximation by (Deep) Neural Networks and the Idea of Locality in CNNs
   - Approximation Results for Shallow and Deep Networks
   - CNNs: Extracting Local Information
4. Lecture 4: Features - Invariance, Symmetry and Stability
   - Features for Audio: (Mel-)Spectrogram and Scattering Transforms
5. Lecture 5: Adaptivity - what do we know, what must we learn?

Invariance, Symmetry and Stability

- **Features** are supposed to make life for learners easier ...
- Let a family of invertible operators $\mathcal{A} = \{\mathcal{A}_g : \mathcal{X} \mapsto \mathcal{X}, g \in G\}$ be given. A function $f : \mathcal{X} \mapsto \mathcal{Y}$ is locally invariant to $\mathcal{A}$, if

$$\forall x \in \mathcal{X}, \exists C_x > 0 \text{ s.t. } \forall g \text{ with } |g| < C_x : \ f(\mathcal{A}_g(x)) = f(x).$$

- Deformation stability: $\|\Phi(\mathcal{A}_g(x)) - \Phi(x)\| \leq C|g|\|x\|$. (Locally linearizable!)
- In general, call a mapping $\Phi : \mathbb{R}^L \mapsto \mathbb{R}^{M_1 \times \ldots \times M_d}$ a *feature extractor* if $\Phi(f)$ maps raw (audio) data to "more structured" representation (e.g. in the sense of encoding known invariances).

Invariance, Symmetry and Stability

- A feature extractor $\Phi = (\Phi_k)_{k=1}^d : \mathbb{R}^L \mapsto \mathbb{R}^{M_1 \times \ldots \times M_d}$ aims at a decomposition $f(x) = f_0(\Phi(x))$ with f_0 (much) simpler than f!
- Φ *separates* f *linearly*, if $f(x)$ is sufficiently closely approximated by

$$\tilde{f}(x) = \langle \Phi(x), w \rangle = \sum_{k=1}^{d} w_k \cdot \Phi_k(x).$$

- Examples, playground.tensorflow
- Features for Audio Data??

Definition (Frame)

A sequence $\{g_j : j \in J\} \subseteq \mathcal{H}$ is called a frame if there exist $A, B > 0$ such that $\forall f \in \mathcal{H}$

$$A\|f\|^2 \leq \sum_{j \in J} |\langle f, g_j \rangle|^2 \leq B\|f\|^2.$$

A, B: frame bounds.

Definition (Frame)

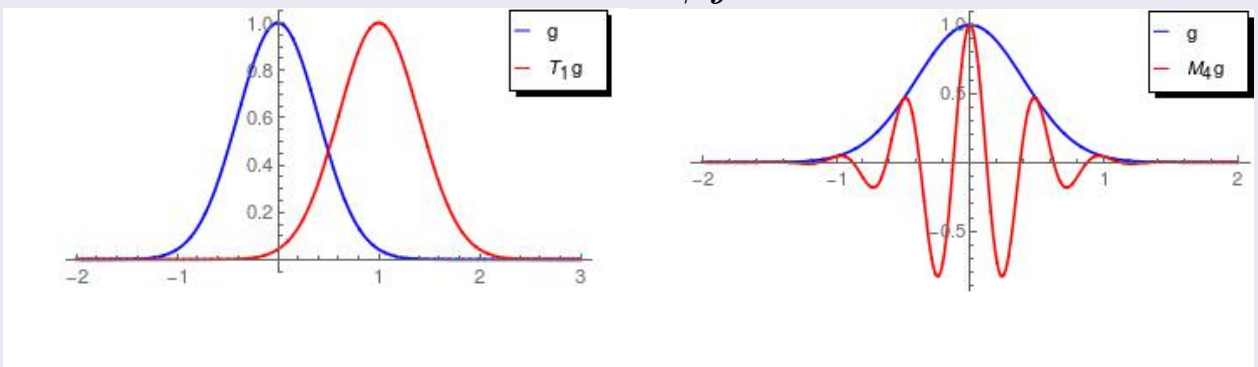
A sequence $\{g_j : j \in J\} \subseteq \mathcal{H}$ is called a frame if there exist $A, B > 0$ such that $\forall f \in \mathcal{H}$

$$A\|f\|^2 \leq \sum_{j \in J} |\langle f, g_j \rangle|^2 \leq B\|f\|^2.$$

A, B: frame bounds.

Definition (Gabor frame)

$\mathcal{G}(g, \alpha, \beta) = \{M_{\beta j} T_{\alpha k} g : j, k \in \mathbb{Z}\}$.
$T_{\alpha k} g(t) = g(t - \alpha k)$ and $M_{\beta j} g(t) = g(t) \cdot e^{2\pi i \beta j}$.



STFT of f with respect to a time-localized window g (e.g. Gaussian):

$$\mathcal{V}_g f(b,k) = \mathcal{F}(f \cdot T_b g)(k)$$

Spectrogram: $S_0(lb_0, k\nu_0) = |\mathcal{V}_g f(lb_0, k\nu_0)|^2 = |\langle f, g_{k,l}\rangle|^2$ if we use elements of a Gabor frame

$$\mathcal{G}(g,\alpha,\beta) = \{g_{j,k} = M_{\beta j} T_{\alpha k} g : j,k \in \mathbb{Z}\}$$

In practice, linear sampling in frequency leads to $S_0 \in \mathbb{R}^{M\times N}$ with N time-samples and M frequency channels with $M \gg N$, most energy accumulated in lower frequency channels. Alternatives?

For non-stationary Gabor frames, windows with adaptive bandwidth replace modulated versions of a fixed window g:

$$\{h_{\nu,l} = T_{lb_\nu} h_\nu : l \in \mathbb{Z}, \nu \in \mathcal{I}\}$$

Time-shift parameters b_ν may be chosen separately for each band in some index set $\mathcal{I}$.

N. Holighaus, M. Dörfler, G. A. Velasco, and T. Grill, "A framework for invertible, real-time constant-Q transforms," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 4, pp. 775 –785, 2013.

Often use $b_\nu = b_0$ for all channels, thus S_a of size $M \times N$ containing the coefficients of f with respect to the non-stationary Gabor frame, i.e.

$$S_a(l,k) = |\langle f, T_l h_\nu \rangle|^2.$$

Now $M = |\mathcal{I}|$ can be chosen such that $M \approx N$.

Feature Extractor and Mel Spectrogram

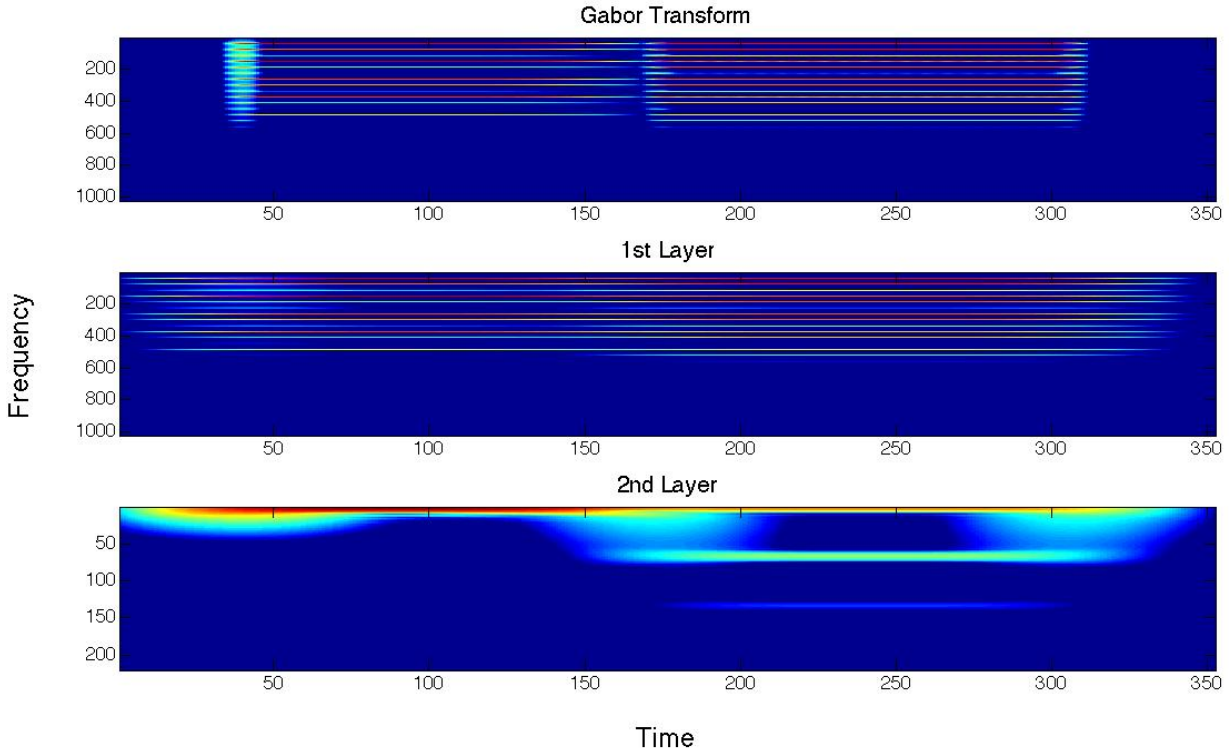
- Spectrogram expresses essential signal properties much more clearly, or sparsely, than raw audio data.
- In general, call a mapping $\Phi : \mathbb{R}^L \mapsto \mathbb{R}^{M_1 \times \ldots \times M_d}$ a *feature extractor* if $\Phi(f)$ maps raw (audio) data to "more structured" representation (e.g. in the sense of encoding known invariances).

Example (Mel spectrogram)

The mel spectrogram is derived from S_0 by taking weighted averages over frequency channels defined by the *mel-scale*:

$$\mathrm{MS}_g(f)(l,\nu) = \sum_k S_0(l,k) \cdot \Lambda_\nu(k).$$

S. S. Stevens, "A scale for the measurement of the psychological magnitude pitch," *Acoustical Society of America Journal*, vol. 8, 1937.



We show, that 1st layer in Gabor scattering is invariant to (smooth) amplitude modulations, while the 2nd layer is invariant to frequency variations.

R.Bammer, MD, "Gabor frames and deep scattering networks in audio processing ," *preprint*, https://arxiv.org/abs/1706.08818, 2017

Triplet Sequence $\Omega = \left((\Psi_\ell, \sigma_\ell, S_\ell)\right)_{\ell\in\mathbb{N}}$:

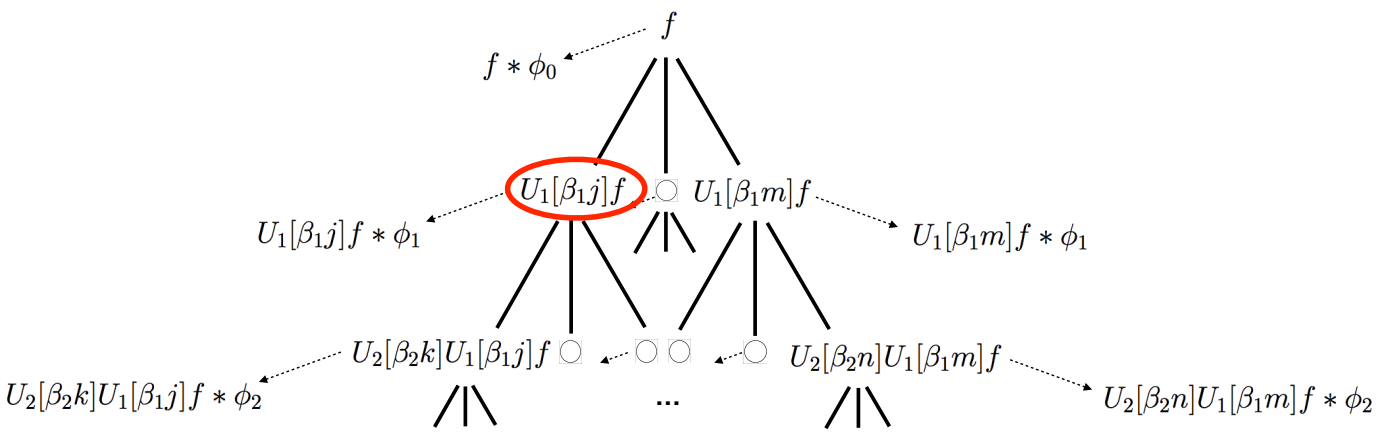
- $\Psi_\ell := \{g_{\lambda_\ell}\}_{\lambda_\ell\in\Lambda_\ell}$ with $g_{\lambda_\ell} = M_{\beta_\ell j} T_{\alpha_\ell k} g_\ell$, $\lambda_\ell = (\alpha_\ell k, \beta_\ell j)$, is a Gabor frame indexed by a lattice $\Lambda_\ell = \alpha_\ell\mathbb{Z} \times \beta_\ell\mathbb{Z}$.
- A pointwise non-linearity function $\sigma_\ell : \mathbb{C} \to \mathbb{C}$, Lipschitz-continuous, i.e. $\|\sigma_\ell f - \sigma_\ell h\|_2 \leq L_\ell \|f - h\|_2$ for all $f, h \in L^2(\mathbb{R})$.
  Here: modulus function with Lipschitz constant $L_\ell = 1$.
- Pooling factor $S_\ell > 0$, which leads to dimensionality reduction.
  Here: covered by choosing specific lattices Λ_ℓ in each layer, i.e. $S_\ell = \alpha_\ell$.

Definition (Gabor Scattering $\ell-$th Layer Element)

Let $\Omega = \left((\Psi_\ell, \sigma_\ell, \Lambda_\ell)\right)_{\ell\in\mathbb{N}}$ be a triplet-sequence. Then the ℓ-th layer of the Gabor scattering transform is defined as the output of the operator $U_\ell : \beta_\ell\mathbb{Z} \times \mathcal{H}_{\ell-1} \to \mathcal{H}_\ell$:

$$f_\ell := U_\ell[\beta_\ell j] f_{\ell-1}(k) := \sigma_\ell(\langle f_{\ell-1}, M_{\beta_\ell j} T_{\alpha_\ell k} g_\ell \rangle_{\mathcal{H}_{\ell-1}}),$$

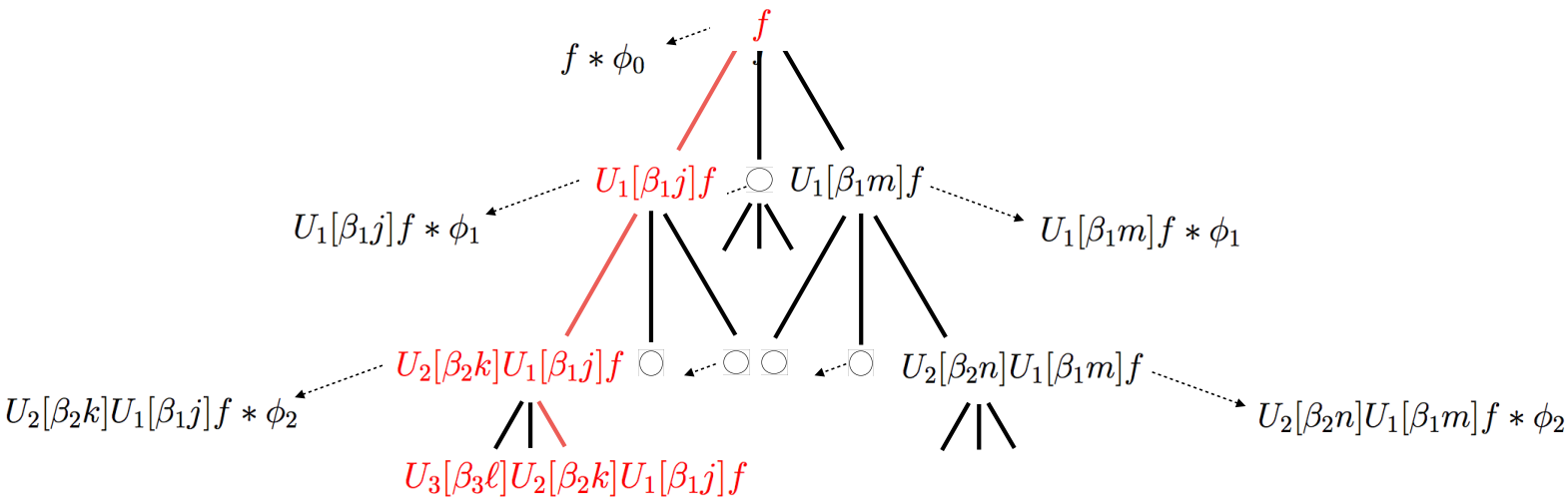
where $f_{\ell-1}$ is the output-vector of the previous layer. Here $\mathcal{H}_0 = L^2(\mathbb{R})$ and $\mathcal{H}_\ell = \ell^2(\mathbb{Z})\ \ \forall \ell > 0$.



Path extension:

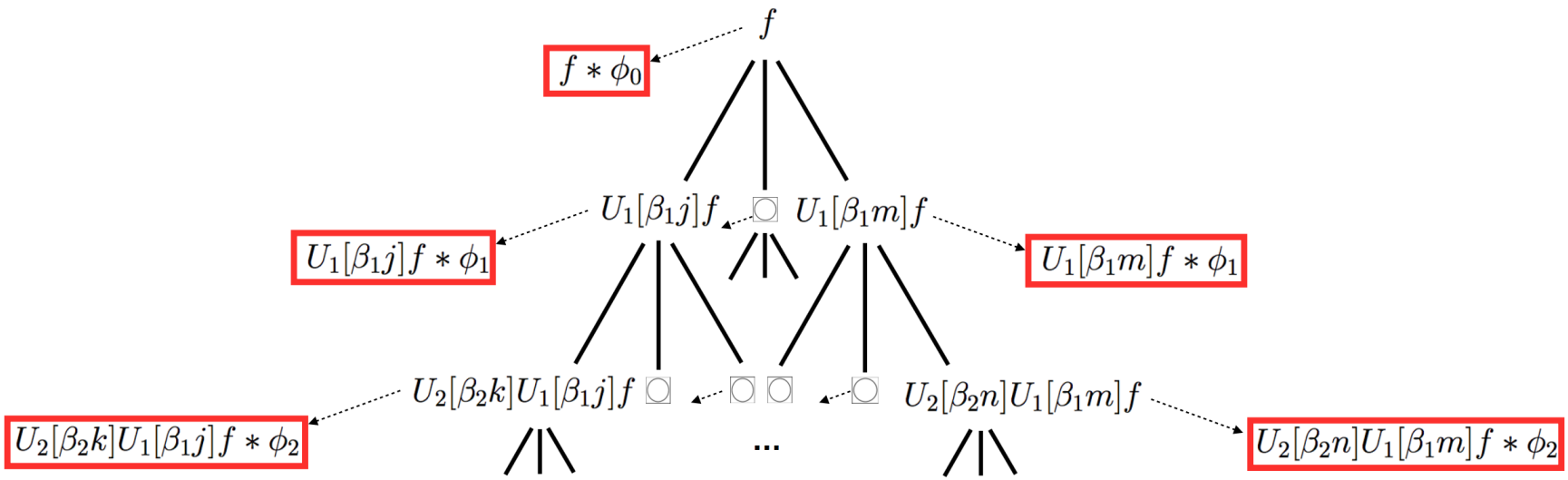
$q := (q_1, ..., q_\ell) \in \beta_1\mathbb{Z} \times ... \times \beta_\ell\mathbb{Z} =: \mathcal{B}^\ell, \ell \in \mathbb{N}$ and obtain

$$U[q]f = U[(q_1, ..., q_\ell)]f := U_\ell[q_\ell] \cdots U_1[q_1]f.$$



Output-generating atom:

$\phi_{\ell-1} := g_{\lambda_\ell^*}, \lambda_\ell^* \in \Lambda_\ell.$



Definition (Feature Extractor)

Let $\Omega = \left((\Psi_\ell, \sigma_\ell, \Lambda_\ell)\right)_{\ell\in\mathbb{N}}$ be a triplet-sequence and ϕ_ℓ the output generating atom for each layer. Then the feature extractor $\Phi_\Omega : L^2(\mathbb{R}) \to (\ell^2(\mathbb{Z}))^{\mathcal{Q}}$ is defined as

$$\Phi_\Omega(f) := \bigcup_{\ell=0}^{\infty} \{(U[q]f) * \phi_\ell\}_{q\in\mathcal{B}_1^\ell}.$$

$\mathcal{Q} := \bigcup_{\ell=0}^{\infty} \mathcal{B}^\ell$ and the space $(\ell^2(\mathbb{Z}))^{\mathcal{Q}}$ of sets $s := \{s_q\}_{q\in\mathcal{Q}},\ s_q \in \ell^2(\mathbb{Z})$ for all $q \in \mathcal{Q}$.

T. Wiatowski and H. Bölcskei

A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction.
CoRR, abs/1512.06293, (2015).

S. Mallat.

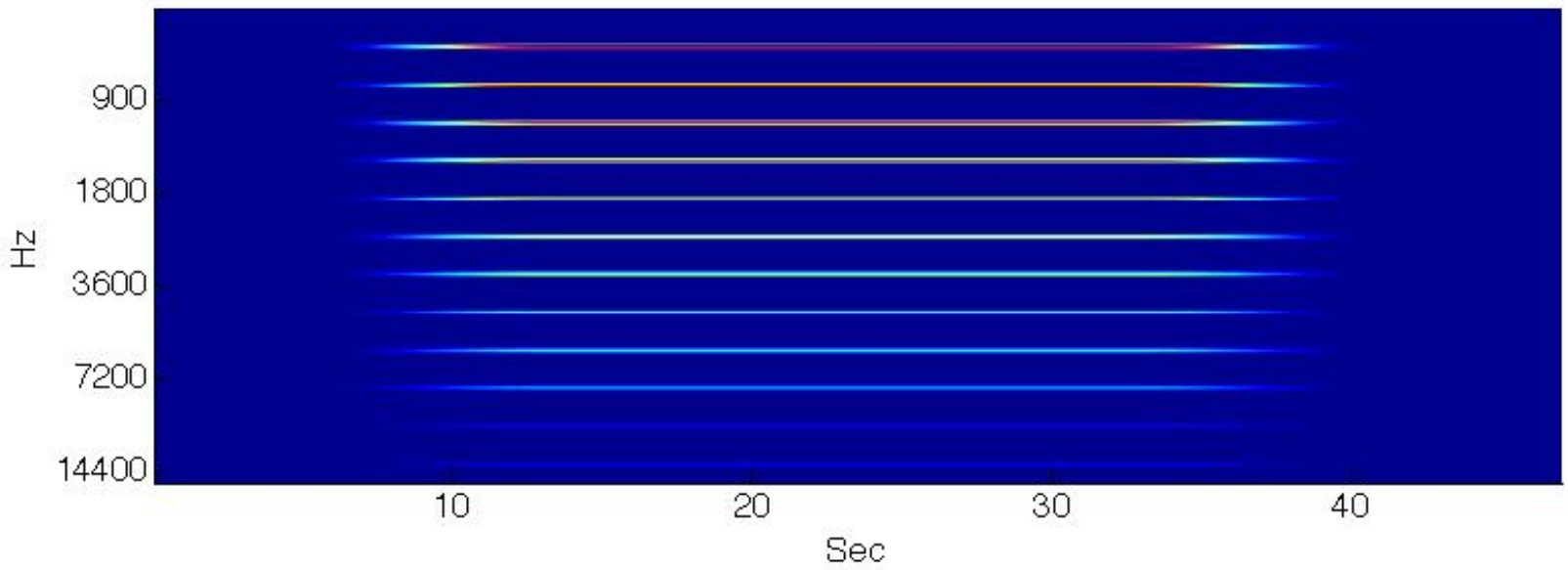
Group Invariant Scattering.
Communications on Pure and Applied Mathematics, 65(10):1331–1398, (2012).

T. Wiatowski and H. Bölcskei

Deep Convolutional Neural Networks Based on Semi-Discrete Frames.
Proc. of IEEE International Symposium on Information Theory (ISIT), Hong Kong, China:1212–1216, (2015).

Class of tones:
$\mathcal{T} = \{\sum_{n=1}^{N} A_n(t) e^{2\pi i n \xi_0 t} | A_n \in \mathcal{C}_c^\infty(\mathbb{R})\}$.

$A_n(t)$... envelope for each harmonic
ξ_0... fundamental frequency, N... number of harmonics
Want to understand invariances and deformation stability induced by the Scattering Network!



Invariance of first layer:

Proposition

Let $f(t) \in \mathcal{T}$ *with* $\|A_n\|_\infty \leq 1$, $\|A_n'\|_\infty < \infty \; \forall n \in \{1, ..., N\}$, $g_1 : |\hat{g_1}(\omega)| \leq C_{\hat{g_1}}(1 + |\omega|^s)^{-1}$ *for some* $s > 1$ *and* $\|t g_1(t)\|_1 = C_{g_1} < \infty$. *Fix* j, n_0 *and let* $n_0 = \underset{n}{\operatorname{argmin}} |\beta j - \xi_0 n|$.

Then

$$U_1[\beta_1 j](f)(k) = |\langle f, M_{\beta_1 j} T_{\alpha_1 k} g_1 \rangle| = A_{n_0}(\alpha_1 k)|\hat{g}_1(\beta_1 j - n_0 \xi_0)| + E_1(k)$$

with

$$E_1(k) \leq C_{g_1} \sum_{n=1}^{N} \|A_n' \cdot T_k \chi[-\alpha_1; \alpha_1]\|_\infty + 2 C_{\hat{g}_1} \sum_{n>0} \left(1 + |\xi_0|^s |n - \frac{1}{2}|^s\right)^{-1}.$$

Note: need slowly varying amplitude; worse separation for low frequencies.

Output:

Let $\phi_1 \in \Psi_2$, then the output of the first layer is

$$U_1[\beta_1 j] f * \phi_1(k) = |\hat{g}_1(\beta_1 j - n_0 \xi)|(A_{n_0} * \phi_1)(k) + \epsilon_1(k),$$

where

$$\epsilon_1(k) \leq C'_{g_1} \cdot \sum_{n=1}^{N} \|A'_n \cdot T_k \chi[-\alpha_1; \alpha_1]\|_\infty + C'_{\hat{g}_1} \sum_{n>0} (1 + |\xi_0|^s |n - \frac{1}{2}|^s)^{-1}.$$

- for slowly varying amplitude $A_n \rightarrow$ contribution near the frequencies of the tone's harmonics.
- ϕ_1 low pass filter and in dependence of pooling factor $\alpha_1 \rightarrow$ temporal fine-structure is averaged out.

$\Rightarrow$ first layer is invariant w.r.t. envelope changes.

Invariance of second layer:

Corollary

Let $f(t) \in \mathcal{T}$, $\sum_{k \neq 0} |\hat{A}_{n_0}(. - \frac{k}{\alpha_1})| \leq \varepsilon_{\alpha_1}$ *and* $|\hat{g}_2(h)| \leq C_{\hat{g}_2}(1 + |h|^s)^{-1}$. *Then the elements of the second layer can be expressed as*

$$\begin{aligned} U_2[\beta_2 h]U_1[\beta_1 j]f(m) =& |\hat{g}_1(\beta_1 j - \xi_0 n_0)| |\langle M_{-\beta_2 h} A_{n_0}, T_{\alpha_2 m} g_2 \rangle| \\ &+ E_2(m), \end{aligned}$$

where

$$E_2(m) \leq \varepsilon_{\alpha_1} C_{\hat{g}_2} |\hat{g}_1(\beta_1 j - \xi_0 n_0)| \sum_r \left(1 + |\beta_2 h - r|^s\right)^{-1} + \|E_1\|_\infty.$$

Output:

Let $\phi_2 \in \Psi_3$ then the second layer output is

$$U_2[\beta_2 h]U_1[\beta_1 j]f * \phi_2(m) = |\hat{g}_1(\beta_1 j - \xi_0 n_0)| |\langle M_{-\beta_2 h} A_{n_0}, T_{\alpha_2 m} g_2 \rangle| * \phi_2$$

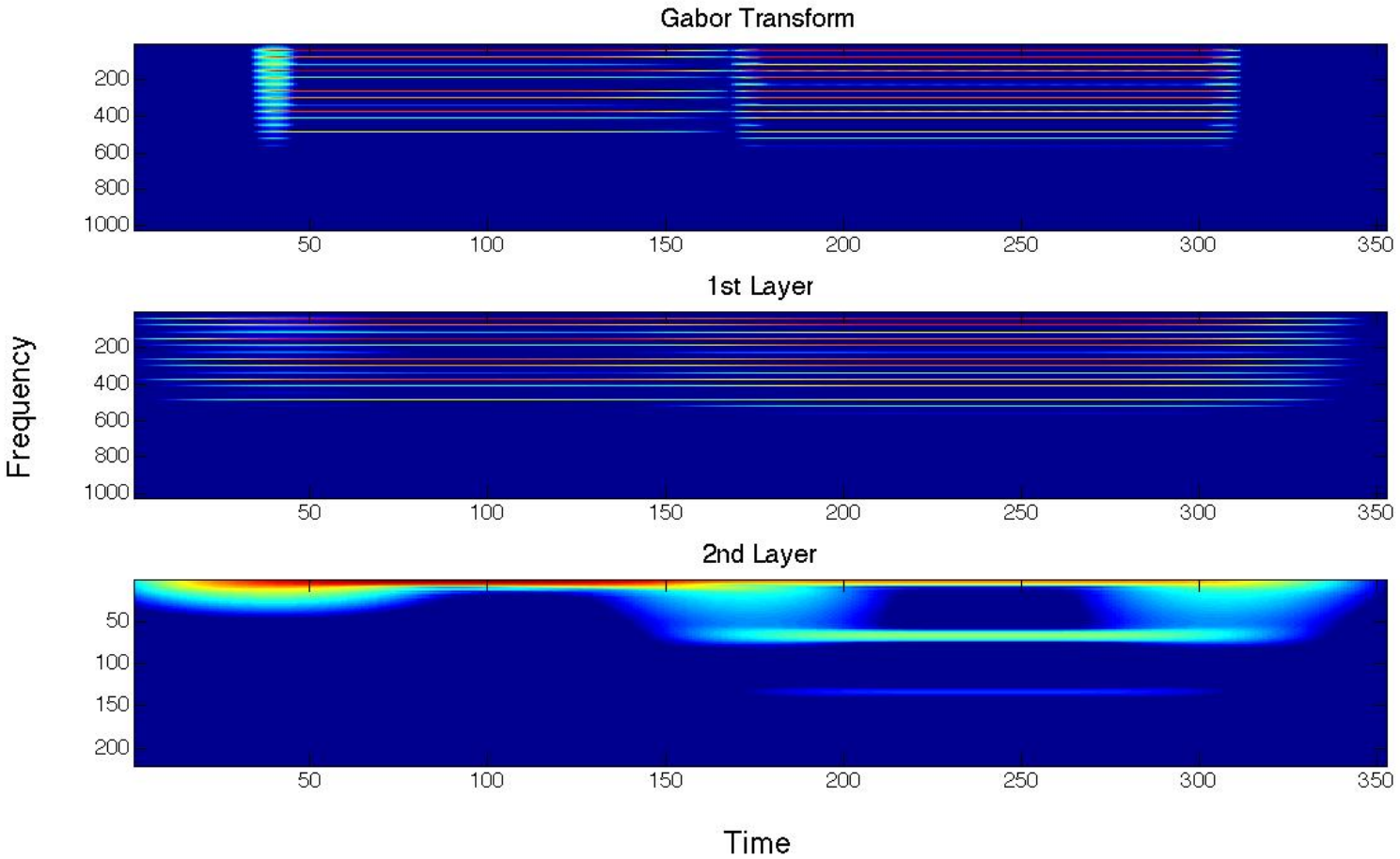
$$+\epsilon_2(m)$$

$$\epsilon_2(m) \leq \varepsilon_{\alpha_1} C'_{\hat{g}_2} |\hat{g}_1(\beta_1 j - \xi_0 n_0)| \sum_r \left(1 + |\beta_2 h - r|^s\right)^{-1} + \|E_1\|_\infty \|\phi_2\|_1.$$

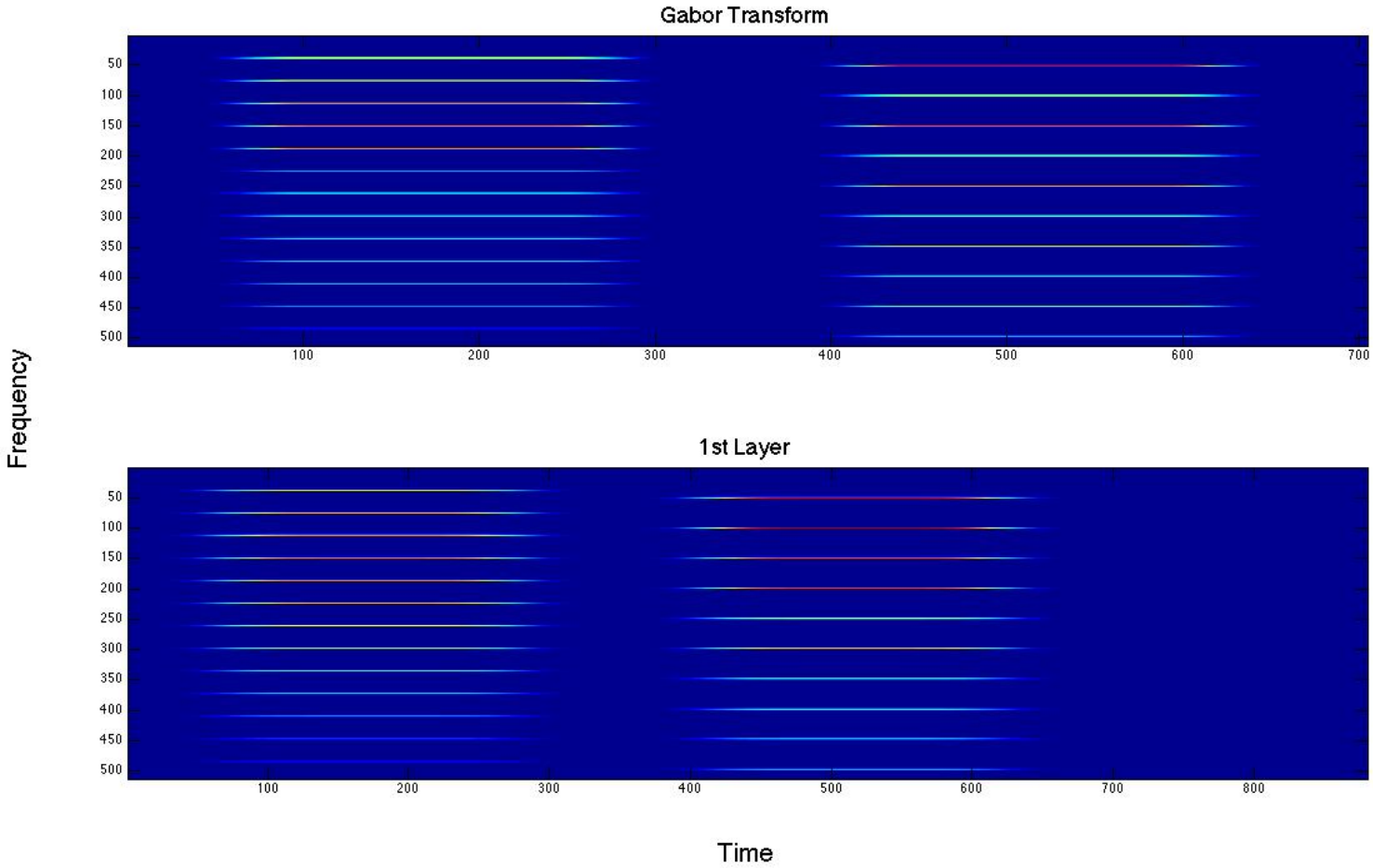
- applying $\phi_2 \rightarrow$ removes fine temporal structure.

$\Rightarrow$ second layer is invariant w.r.t. pitch and reveals information contained in the envelopes A_n.

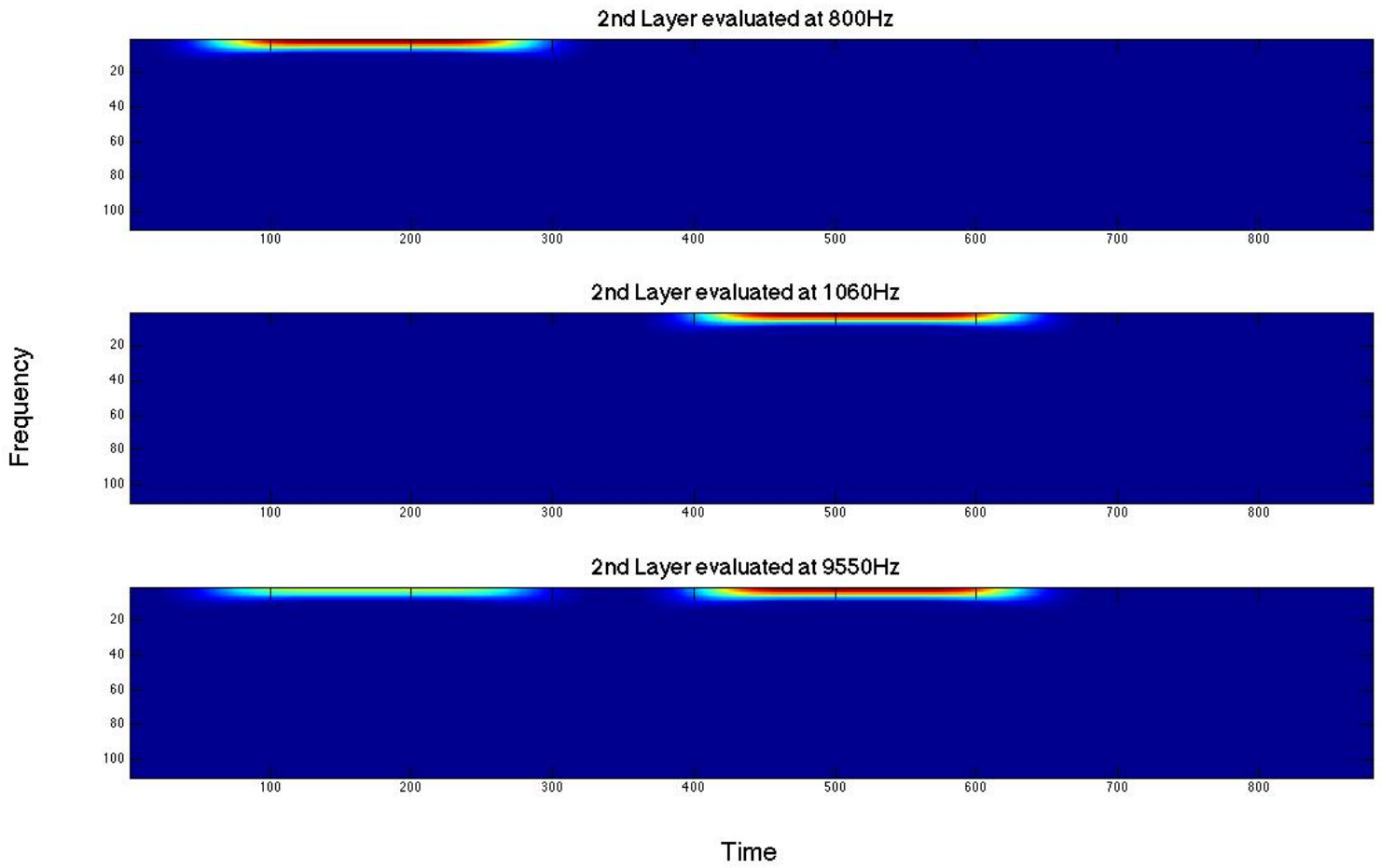
Different envelopes:



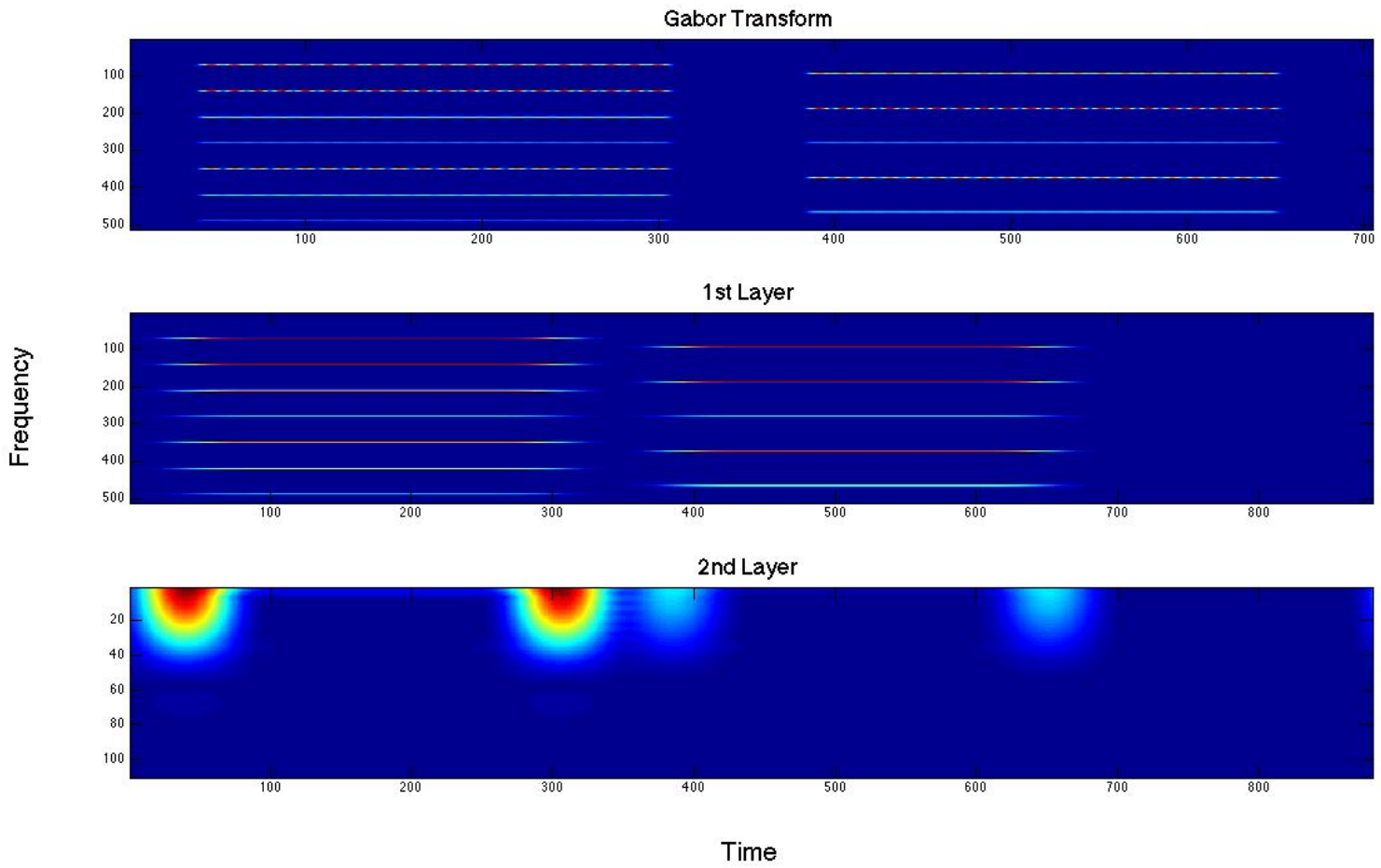
Pitch invariance (1st part):



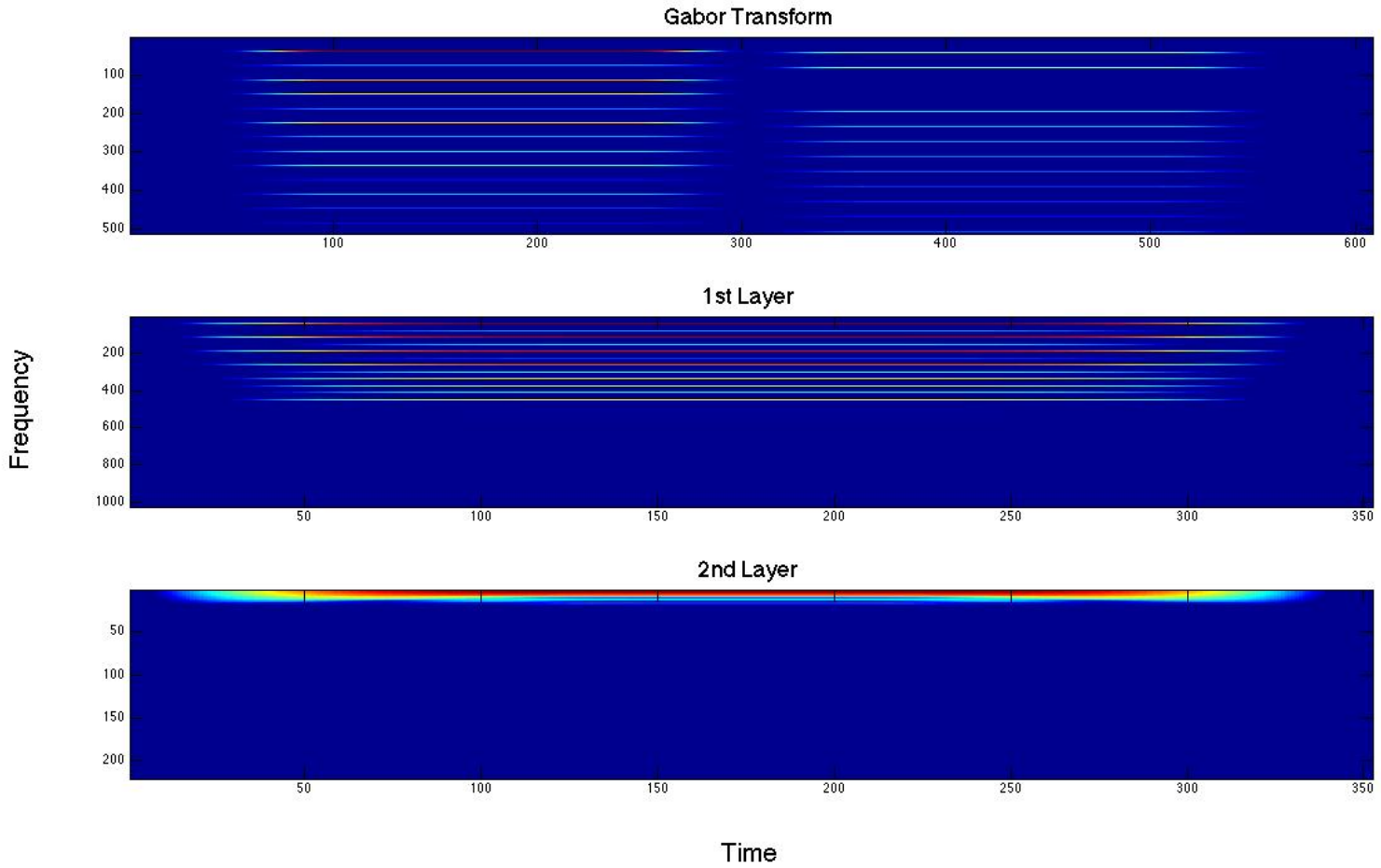
Pitch invariance (2nd part):



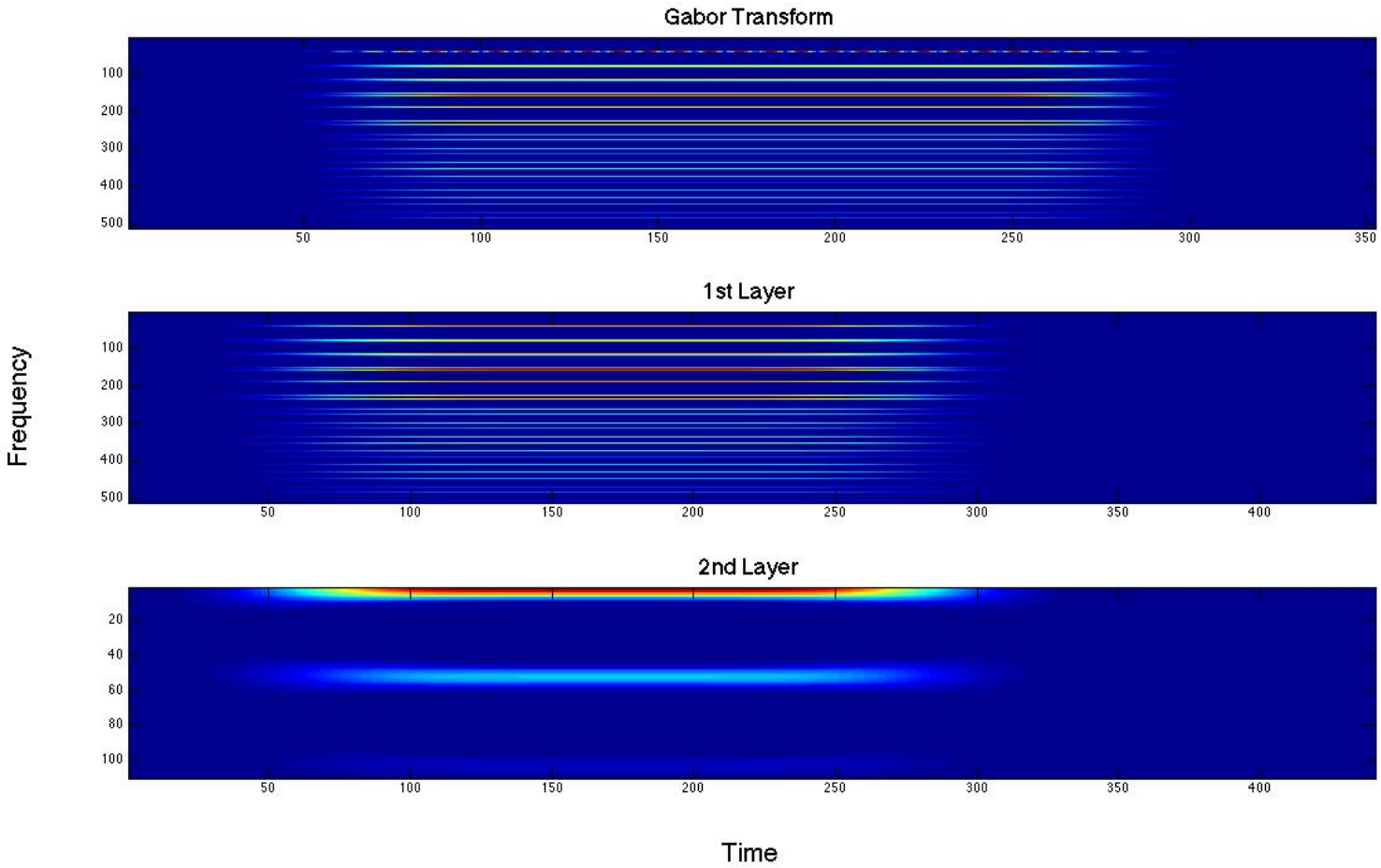
Same example with modulated envelope:



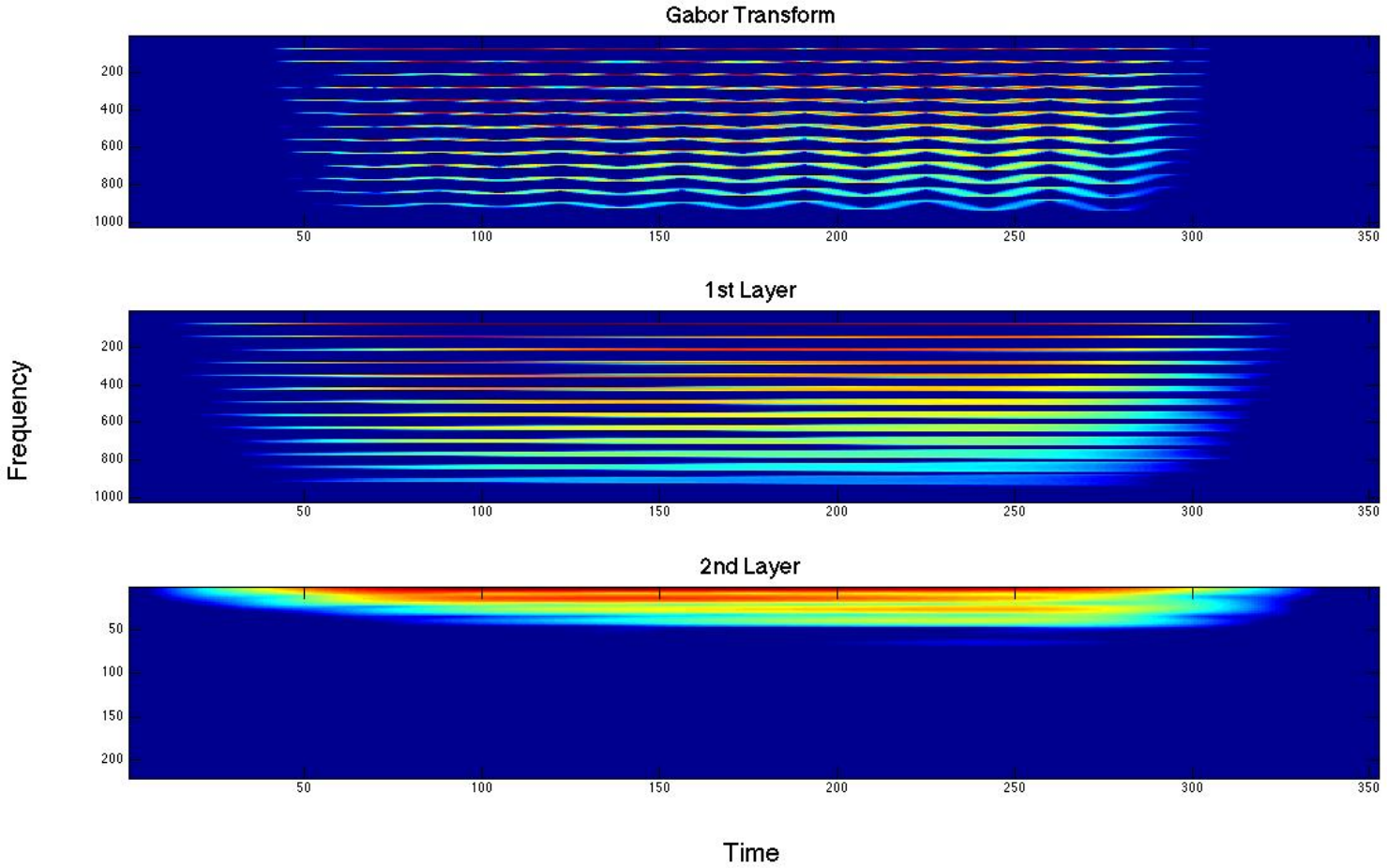
Two tones close to each other (1st part):



Two tones close to each other causing beats (2nd part):



Frequency modulation:



homepage.univie.ac.at/monika.doerfler/GaborScattering.html

Deformation stability

Stable w.r.t.:

- envelope changes $\mathfrak{F}_\tau(f)(t) = \sum_{n=1}^N A_n(t+\tau(t))e^{2\pi i n\xi_0 t}$.
- frequency modulation $\mathfrak{F}_\tau(f)(t) = \sum_{n=1}^N A_n(t)e^{2\pi i(n\xi_0 t+\tau_n(t))}$.

$\Rightarrow$ Stability is obtained by using the decoupling technique:

- contractivity of feature extractor: $\|\Phi_\Omega(f) - \Phi_\Omega(h)\|_2 \leq \|f-h\|_2$.
- error bound of signal class: $\|f - \mathfrak{F}_\tau(f)\|$, w.r.t. a small deformation τ.

P. Grohs, T. Wiatowski and H. Bölcskei
Deep convolutional neural networks on cartoon functions.
IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016:1163–1167, (2016).

Proposition (Envelope changes)

Let $f(t) \in \mathcal{T}$ *and* $|A_n'(t)| \leq C_n(1+|t|^s)^{-1}$, *for constants* $C_n > 0,\ n = 1, ..., N$ *and* $s > 1$. *Moreover let* $\|\tau\|_\infty \ll 1$. *Then*

$$\|f - \mathfrak{F}_\tau(f)\|_2 \leq D\|\tau\|_\infty \sum_{n=1}^{N} C_n,$$

for $D > 0$ *depending only on* $\|\tau\|_\infty$.

Proposition (Frequency modulation)

Let $f(t) \in \mathcal{T}$ *and assume* $\|A_n\|_\infty < \frac{1}{n}$. *Moreover let* $\|\tau_n\|_\infty < \frac{\arccos(1-\frac{\varepsilon^2}{2})}{2\pi}$. *Then*

$$\|f - \mathfrak{F}_\tau(f)\|_2 \leq \varepsilon \sum_{n=1}^{N} \frac{1}{n}.$$

R. Bammer and M. Dörfler

Invariance and Stability of Gabor Scattering for Music Signals.

Proc. of Sampling Theory and Aplication (Sampta) in Tallinn, Estonia, (2017).

R. Bammer and M. Dörfler, "Gabor frames and deep scattering networks in audio processing ," *preprint*, https://arxiv.org/abs/1706.08818, 2017

Ongoing work: using Gabor Scattering coefficients as input to various classification methods in comparison to standard features, such as Mel-coefficients.. (coming soon).

1. Lecture 1: Introduction and Motivation
   - The Power of Convolutional Neural Networks
   - What is learning in a mathematical sense? - a simple example
2. Lecture 2: Elements of Mathematical Learning Theory
   - Approximation Error and Sample Error
   - Generalization
3. Lecture 3: Approximation by (Deep) Neural Networks and the Idea of Locality in CNNs
   - Approximation Results for Shallow and Deep Networks
   - CNNs: Extracting Local Information
4. Lecture 4: Features - Invariance, Symmetry and Stability
   - Features for Audio: (Mel-)Spectrogram and Scattering Transforms
5. Lecture 5: Adaptivity - what do we know, what must we learn?

Equivalence of feature-network pairs

CNN $\mathcal{N}$ defined by

- input dimension,
- number of convolutional layers D_c,
- number of dense layers D_d,
- number and size of convolutional kernels in each convolutional layer,
- type of non-linearity and pooling in each layer.

$\theta \in \mathbb{R}^p$ ...parameter vector comprising all the weights occurring in the network; $\mathcal{N}(\theta)$... concrete realisation.
Data set: $\mathcal{D} = \{(f_i, c_i) \in \mathbb{R}^L \times \mathbb{R}, i \in \mathcal{I}\}$.

Equivalence of feature-network pairs

Since designed feature extractors may often lead to useful invariances, we need to consider both feature extractor and network architecture.

Definition (CNN equivalence)

Given two feature-network pairs $(\Phi_j, \mathcal{N}_j)$, $j = 1, 2$, we say that $(\Phi_1, \mathcal{N}_1)$ is subordinate to $(\Phi_2, \mathcal{N}_2)$ with respect to a data set $\mathcal{D}$, if for all $\theta_1 \in \mathbb{R}^{p_1}$ there exists a $\theta_2 \in \mathbb{R}^{p_2}$ such that

$$\mathcal{N}_1(\theta_1)(\Phi_1(f_i)) = c_i \Rightarrow \mathcal{N}_2(\theta_2)(\Phi_2(f_i)) = c_i \ \ \forall (f_i, c_i) \in \mathcal{D}.$$

$(\Phi_1, \mathcal{N}_1)$ and $(\Phi_2, \mathcal{N}_2)$ are equivalent with respect to $\mathcal{D}$ if they are subordinate to each other.

Anden and Mallat showed that the mel-spectrogram can be approximated by time-averaging the absolute values-squared of a wavelet transform.

J. Andén and S. Mallat, "Deep scattering spectrum,"
IEEE Transactions on Signal Processing
vol. 62, no. 16, pp. 4114–4128 (2014)

Can this be made precise? What is the consequence for representation design?

Compute filtered version of f with respect to filter bank h_ν (generating non-stationary Gabor frame $\{T_l h_\nu\}, \nu \in \mathcal{I}, k \in \mathbb{Z}$) and apply subsequent time-averaging using a time-averaging function ϖ_ν:

$$\mathrm{FB}_{h_\nu}(f)(b,\nu) = \sum_l |(f * h_\nu)(\alpha l)|^2 \cdot \varpi_\nu(\alpha l - b).$$

Recall:

$$\mathrm{MS}_g(f)(b,\nu) = \sum_k |\mathcal{F}(f \cdot T_b g)(\beta k)|^2 \cdot \Lambda_\nu(\beta k).$$

Equivalence of feature-network pairs

Proposition

For all $\nu \in \mathcal{I}$, let $g, h_\nu, \Lambda_\nu, \varpi_\nu$ be given. Let $\mathrm{MS}_g(f)$ and $\mathrm{FB}_{h_\nu}(f)$ be computed on a lattice $\alpha\mathbb{Z} \times \beta\mathbb{Z}$ and set

$$m^\nu(x) = \sum_l T_{\frac{l}{\beta}} \mathcal{F}^{-1}(\Lambda_\nu)(x) \quad \text{and} \quad m_F^\nu(\xi) = \sum_k T_{\frac{k}{\alpha}} \mathcal{F}(\varpi_\nu)(\xi).$$

Then the following estimate holds for all $(b, \nu) \in \alpha\mathbb{Z} \times \mathcal{I}$:

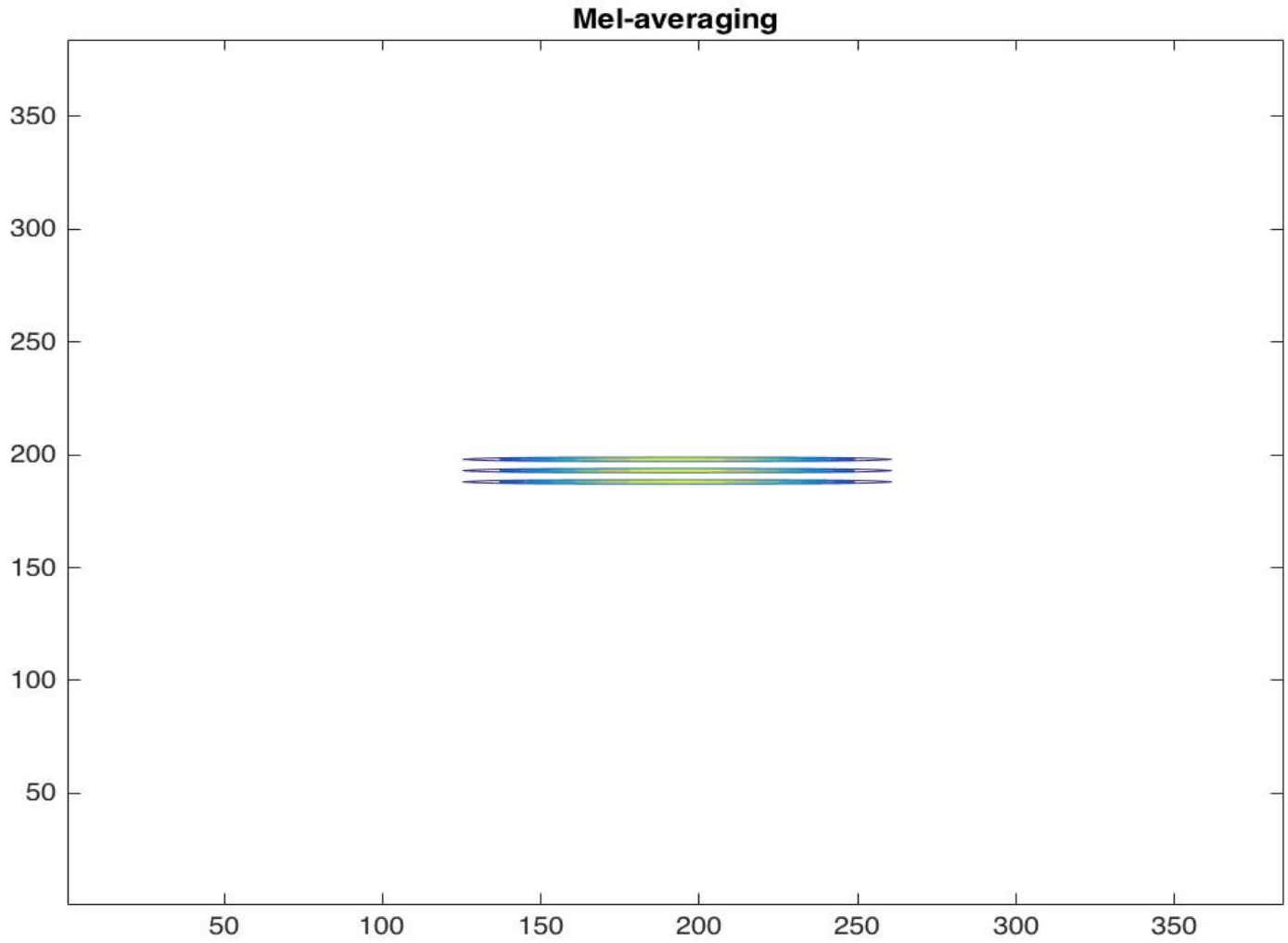
$$|\mathrm{MS}_g(f)(b,\nu) - \mathrm{FB}_{h_\nu}(f)(b,\nu)| \leq \|\mathcal{V}_g g \cdot m^\nu - \mathcal{V}_{h_\nu} h_\nu \cdot m_F^\nu\|_2 \|f\|_2^2$$

In particular, if

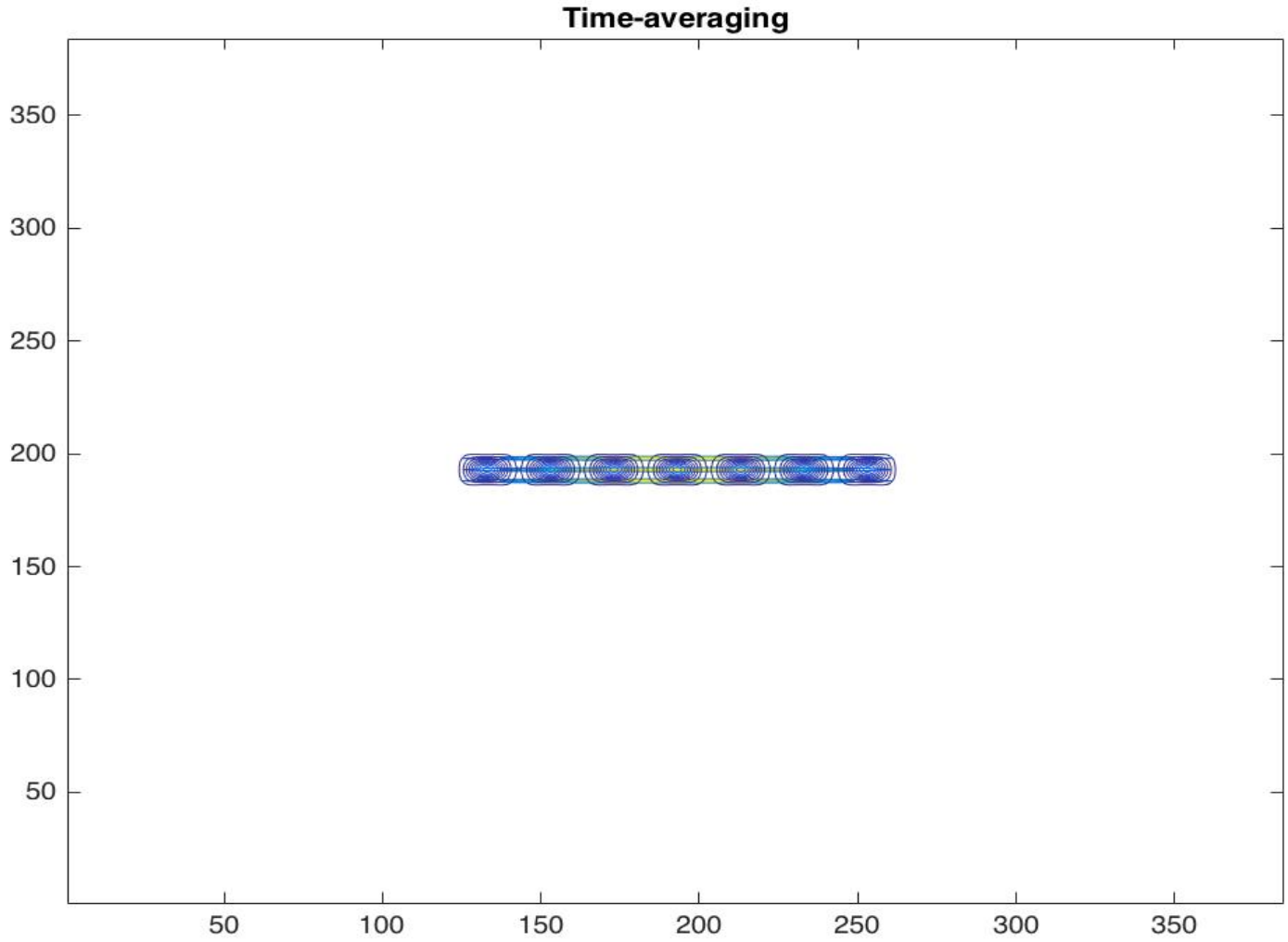
$$V_{h_{\nu_k}} h_{\nu_k}(x,\xi) \cdot \mathcal{F}(\varpi_{\nu_k})(\xi) = V_g g(x,\xi) \cdot \mathcal{F}^{-1}(\Lambda_{\nu_k})(x),$$

then $\mathrm{MS}_g(f)(l, \nu_k)$ can be obtained by time-averaging the filtered signal's absolute value squared on the full lattice $\mathbb{Z}$ ($\alpha = 1$).

Equivalence of feature-network pairs



Equivalence of feature-network pairs



Equivalence of feature-network pairs

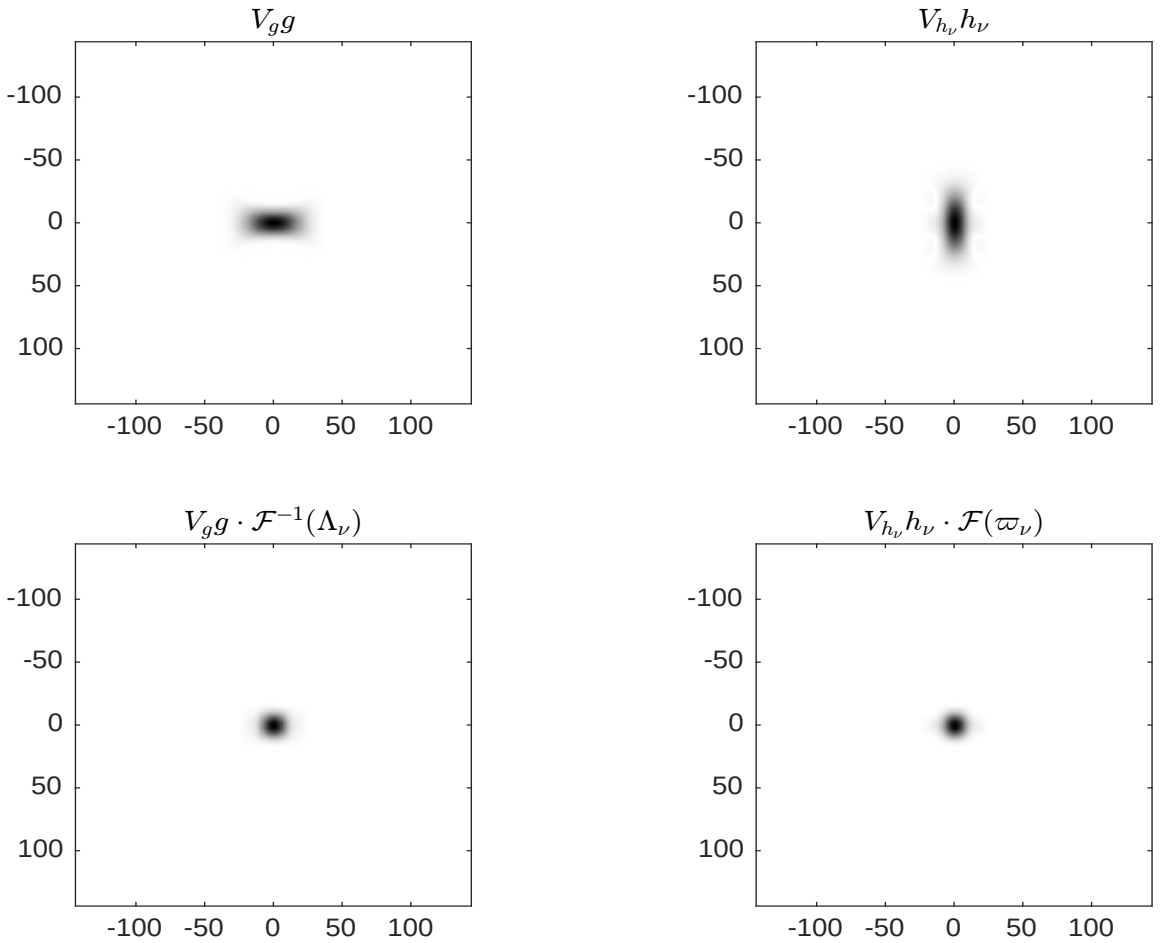


Figure: Spreading functions of operators defining different feature extractors.

Equivalence of feature-network pairs

Theorem

Consider $\mathcal{N}_1$ as a convolutional network with D_c convolutional layers.
Consider $\mathcal{N}_2$, analogue to $\mathcal{N}_1$, except for an additional convolutional layer, consisting of a finite number of convolutional kernels with sufficient length in time-direction and length 1 in frequency direction, preceding the D_c convolutional layers.
Then $(\mathrm{MS}_g, \mathcal{N}_1)$ is subordinate to $(S_a, \mathcal{N}_2)$ if the windows g, h_ν and the mel-filters Λ_ν are chosen such that $\mathrm{MS}_g = \mathrm{FB}_{h_\nu}$.

M. Dörfler, T. Grill et al. "Basic Filters for Convolutional Neural Networks Applied to Music: Training or Design?' *To appear in Neural Computation and Applications*, https://arxiv.org/abs/1709.02291, 2018.

Example: Performance on Singing Voice Detection

Singing voice detection: binary problem of presence or absence of human voice in music

Let's listen to and watch some examples!

`http://ofai.at/~jan.schlueter/pubs/2016_ismir/alexanderross/index.html`

The architecture has a total number of 1.41 million weights (91% for the dense layers).

Challenges: understanding why CNNs perform so well

How is it possible, that networks defined by more than a million of weights and trained on significantly fewer data points still *generalize well*?
Two central challenges:

- *Generalization:* performance on unseen data points drawn from same distribution.
- *Architectures:* predefined number of layers (depth), both convolutional and dense, and size of layers (width).

Keywords: Drop-out, Validation, Features

Experimental Setup:

1. Size reduction possible since we expect useful invariances captured by features
2. Four convolutional layers, two 3×3 convolutions (32 and 16 kernels), 3×3 non-overlapping max-pooling, two more 3×3 convolutions (32 and 16 kernels), 3×3 pooling.
3. Two variants for dense layer (Classification stage):
   'small-two': two dense layers of 64 and 16 units (total number of weights 94337, 85% classification stage).
   'small-one': one dense layer of 32 units (total number of weights is 53857, 73% classification stage).
4. Final dense layer is a single sigmoidal output unit.

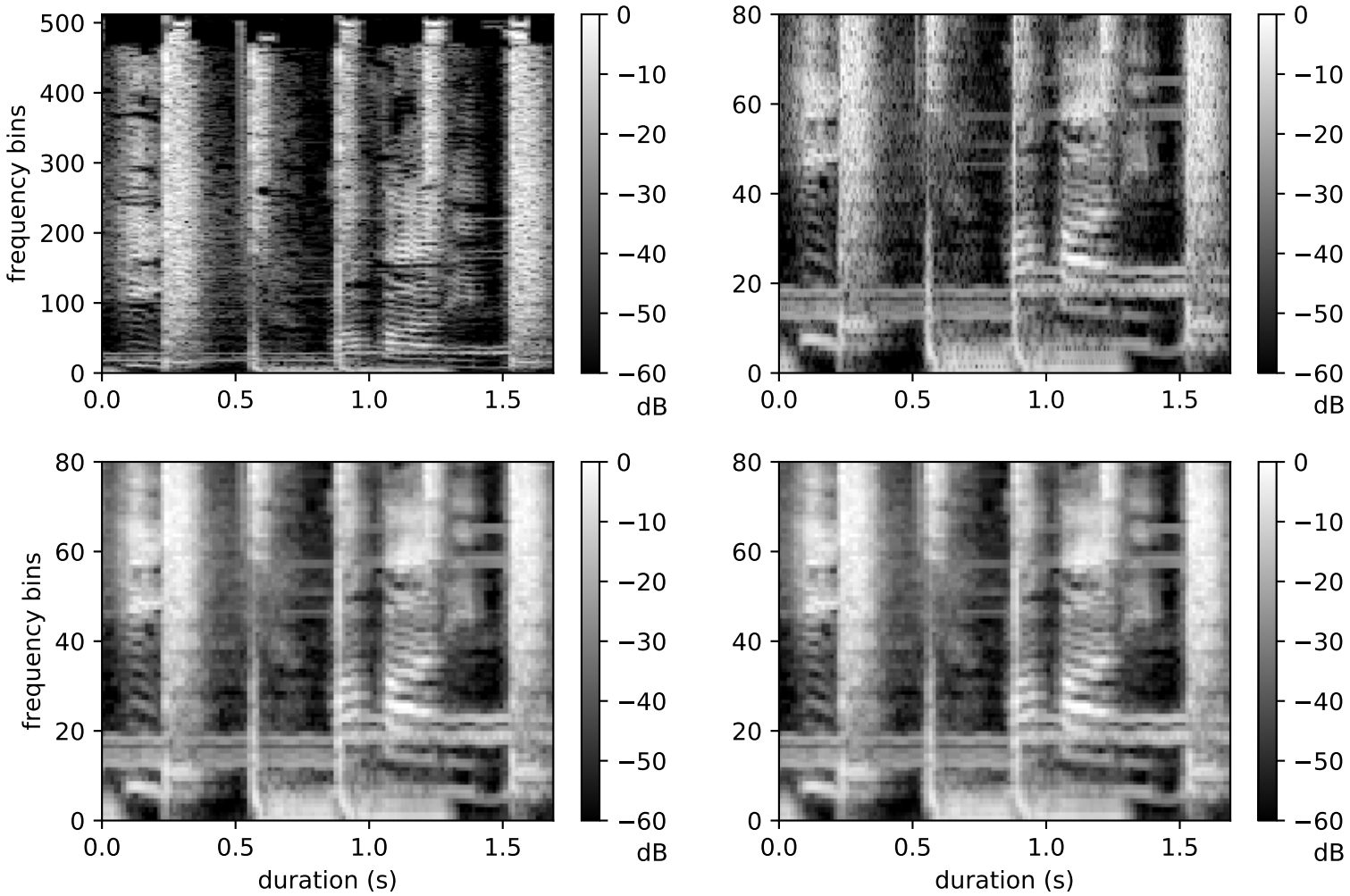
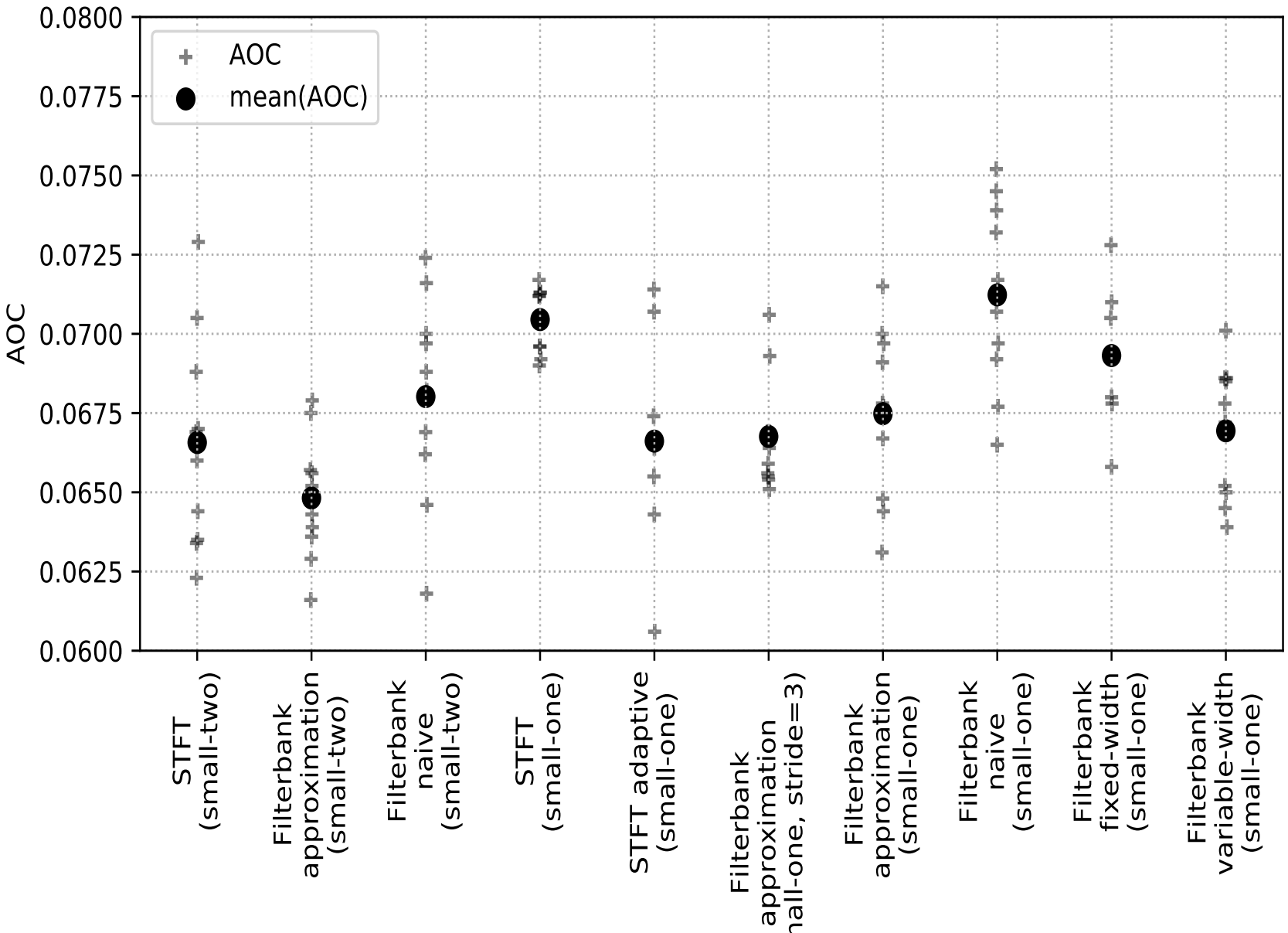


Figure: Time-frequency representations for the problem of singing voice detection. Spectrogram(upper left), STFT-based mel spectrogram (bottom left), filter bank computed (top right), and filter bank with time-averaging (bottom right).

Equivalence of feature-network pairs - empirical results



Equivalence of feature-network pairs

What is the influence of choice of analysis window?

Proposition

Let $S_0^g(l,k) = |\langle f, g_{k,l}\rangle|^2$ and let analysis windows g and h be given. If convolutional kernels w_g and w_h are chosen such that

$$\mathcal{F}_s(w_g) \cdot \mathcal{V}_g g = \mathcal{F}_s(w_h) \cdot \mathcal{V}_h h,$$

then $S_0^g * w_g = S_0^h * w_h$.

$\mathcal{F}_s \cdots$ symplectic Fourier transform.

Equivalence of feature-network pairs

IDEA of Proof:

- Write two-dimensional convolution as Gabor multiplier: $S_0^g * w_g(\lambda) = \langle G_{g,\mathbf{m}^\lambda} f, f\rangle$, where $\lambda = (l,k)$ and

$$\mathbf{m}^\lambda(l',k') = w_g(l-l',k-k') =: \tilde{w}_g^\lambda(l',k')$$

- Resulting operators can be written by means of their spreading functions:

$$\eta_{G_{g,\mathbf{m}}}(x,\xi) = \mathcal{F}_s(\mathbf{m})(x,\xi)\mathcal{V}_g g(x,\xi),$$

-

$$\langle G_{g,\mathbf{m}^\lambda} f, f\rangle = \int_z \mathcal{F}_s(\tilde{w}_g^\lambda)(z) e^{-2\pi i(l\xi - kx)} \mathcal{V}_g g(z) \overline{\mathcal{V}_f f}(z) dz$$

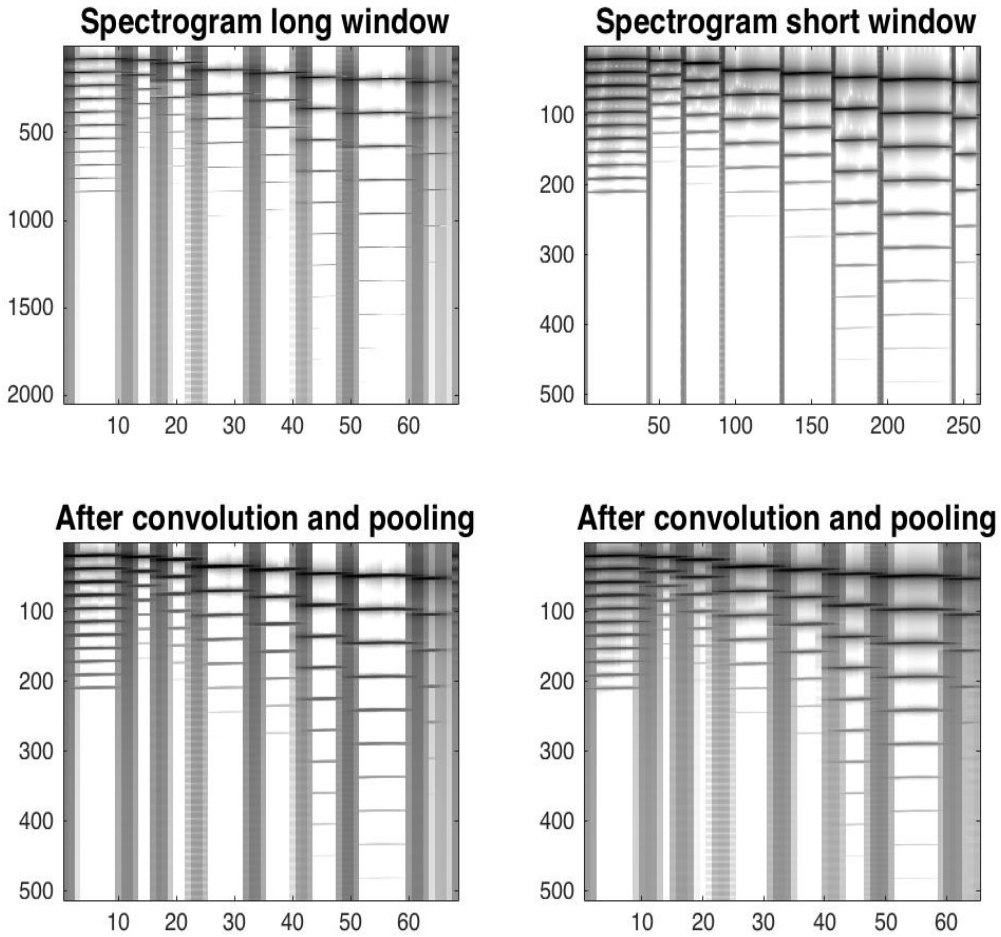


Figure: Synthetic signal consisting of several damped notes: convolution leads to similar results from previously different spectrograms.

Equivalence of feature-network pairs

Theorem

Consider two convolutional neural networks Π_j, $j = 1, 2$ with D_c convolutional layers and the same number of convolutional kernels $w_{k_1}^{k_0}$ for $j = 1, 2$.

For two windows g, h, the size of $w_{k_1}^{k_0}$ for $j = 1, 2$ can be chosen such that (S_0^g, Π_1) and (S_0^h, Π_2) are equivalent (under appropriate support conditions on g and h).

Invariance

- Recall: CNNs can learn ”almost anything”
- Recent example on music signals (Oriol Nieto, Pandora, private communication): deep CNN learns semantic music content with more than 98% accuracy (!?).
- However..
- Pandora owns 1.5 millions of manually annotated music tracks.
- for training data of up to 500.000 hours of music, learning on raw audio cannot beat learning on pre-processed data.

...so it can be helpful to provide known invariance by means of an appropriate feature extractor!

Proposition

Let $(\Phi_1, \mathcal{N}_1)$ be subordinate to $(\Phi_2, \mathcal{N}_2)$ with respect to $\mathcal{D}$ and let $\mathcal{A}(\mathcal{D})$ denote an augmented data-set.
If $\mathcal{N}_1(\Phi_1(\mathcal{A}(x))) = \mathcal{N}_1(\Phi_1(x))$ for all $x \in \mathcal{D}$, and Φ_2 is invariant to $\mathcal{A}$, then $(\Phi_1, \mathcal{N}_1)$ is also subordinate to $(\Phi_2, \mathcal{N}_2)$ with respect to $\mathcal{A}(\mathcal{D})$.

Example: Let $(Id, \mathcal{N}_1)$ be subordinate to $(S_0, \mathcal{N}_2)$ with respect to $\mathcal{D}$; let $M(\mathcal{D})$ denote the augmented data-set achieved by multiplication with a phase factor. If $\mathcal{N}_1$ is invariant to M, then $(Id, \mathcal{N}_1)$ is also subordinate to $(S_0, \mathcal{N}_2)$ with respect to $M(\mathcal{D})$.

S. Mallat.
Understanding deep convolutional networks.
Philos Trans A Math Phys Eng Sci., 374(2065), 2016.

J. Sokolic et al
Generalization Error of Invariant Classifiers
Preprint, 2017

Invariance

Sokolic, Giryes, Sapiro and Rodrigues recently defined the notion of an invariant learning algorithm as follows.

Definition

A learning algorithm $\mathcal{T}$ is invariant with respect to an augmentation $\mathcal{A}$, if for any learned model f it holds that $f(\mathcal{A}(x)) = f(x)$ for all $x \in \mathcal{D}$.

J. Sokolic et al
Generalization Error of Invariant Classifiers
Preprint, 2017

It can then be shown that the generalization error reduces by a factor proportional to $\mathcal{N}(\mathcal{D})/\mathcal{N}(\mathcal{A}(\mathcal{D}))$ (Recall that $\mathcal{N}(\mathcal{D})$, the covering number of $\mathcal{D}$ characterizes the complexity of $\mathcal{D}$, in this case the signal class.
Here: invariant feature extractor leads naturally to invariant learning algorithm and thus reduces the generalization gap!

Conclusions - Perspectives

- CNNs learn signal representations similar to standard representations such as mel-spectrogram; there can be small, but significant performance differences in dependence on structure of input representation.
- Allowing for mild variants of adaptivity can lead to small but significant performance improvement; full adaptivity ("true end-to-end learning") requires huge amount of data.
- Invariance of feature extractors may lead to smaller generalization gap and sampling error.
- Equivalence of feature/network pairs will be studied including deeper layers and taking signal class properties into account; this will provide estimates for generalization error and give guidelines for architecture choice.

Thanks for your attention!
Questions? Remarks?