Pre-processing Data for Deep Learning? The Balance Between Discriminability and Invariance

Monika Dörfler

NuHAG, Faculty of Mathematics, University of Vienna

SPARS19

July 4th, 2019

Contents

Projects:

- SALSA (Semantic Annotation by Learned Structured and Adaptive Signal Representations) (WWTF, Mathematics+)
- aMoby (Acoustic Monitoring of Biodiversity) (WWTF, NEXT - New Exciting Transfer Projects)
- People involved:
  - Roswitha Bammer, Pavol Harar (NuHAG, University of Vienna)
  - Arthur Flexer, Thomas Grill, Jan Schlüter (OFAI)
  - Stefan Lattner (Sony Computer Science Laboratories, Paris, France)

Structure in Data Sets

Learning is generalization ..

- Learning language
- Learning categories
- Learning mathematics, how to play instruments, how to build furniture
- And how can this be formalized?

Structure in Data Sets

Generalization depends on structure

- *"It is impossible to justify a correlation between reproduction of a training set and generalization error off of the training set using only a priori reasoning. As a result, the use in the real world of any generalizer that fits a hypothesis function to a training set (e.g., the use of back-propagation) is implicitly predicated on an assumption about the physical universe."*

D. H. Wolpert,
On the connection between in-sample testing and generalization error; Complex Systems, Vol.6/1, 1992

- Learning without considering structure is memorization.
- Structure can be found in data and in learning tasks.

Generalization depends on structure

- *"It is impossible to justify a correlation between reproduction of a training set and generalization error off of the training set using only a priori reasoning. As a result, the use in the real world of any generalizer that fits a hypothesis function to a training set (e.g., the use of back-propagation) is implicitly predicated on an assumption about the physical universe."*

D. H. Wolpert,
On the connection between in-sample testing and generalization error; Complex Systems, Vol.6/1, 1992

- Learning without considering structure is memorization.
- Structure can be found in data and in learning tasks.
- Formally the (assumed) structure in learning tasks is described by the chosen *hypothesis space* from which the input-output mapping is eventually chosen.

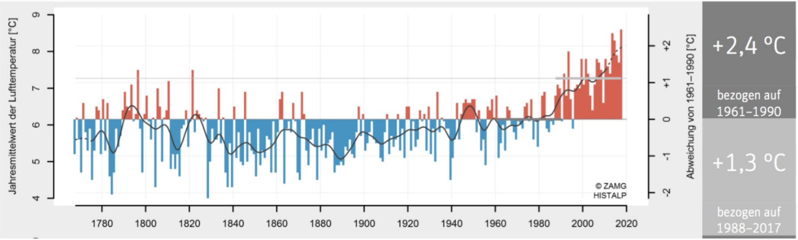
Structure in Data Sets

- Relevant structures in image data are relatively straight-forward to understand
- (Deep) convolutional neural networks designed to extract local structures in images
- Equivalently, some basic invariances in images are easily understood, such as (depending on problem)
  - rotation
  - illumination
  - small deformations
- Structures due to these invariances often imposed by augmentation

Structure in Data Sets

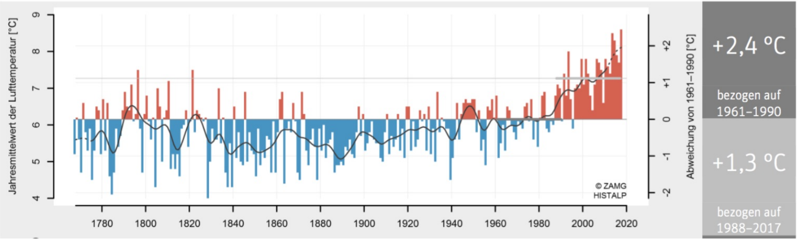
- Relevant structures in image data are relatively straight-forward to understand
- (Deep) convolutional neural networks designed to extract local structures in images
- Equivalently, some basic invariances in images are easily understood, such as (depending on problem)
  - rotation
  - illumination
  - small deformations
- Structures due to these invariances often imposed by augmentation
- What about time series?

- What about time series? The most critical...



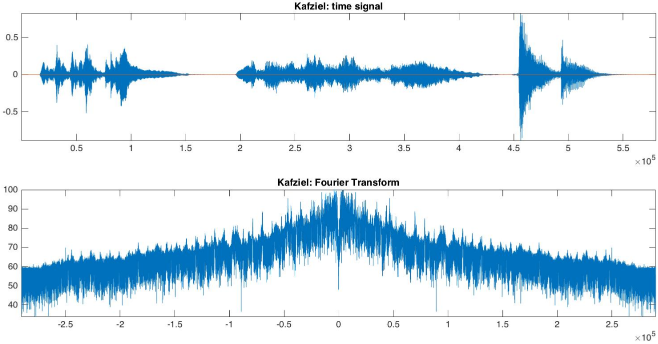
Structure in Data Sets - Excursus 1

- What about time series? The most critical...



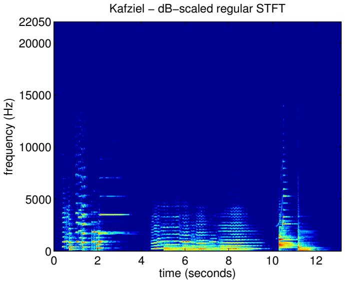
- Avoid using airplanes whenever possible
- Ask for video conferences
- Join `https://www.scientists4future.org`
- Ask your institution to promote *Climate-Friendly Research* and to join alliances of climate-friendly universities

- Our favorite time series: music, speech [1]

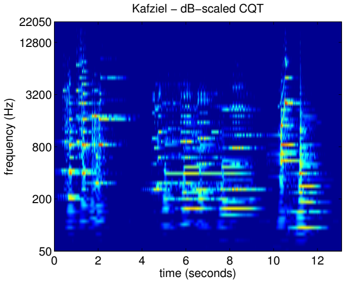


[1] Mark Feldman, Sylvie Courvoisier: KAFZIEL, from: Book of Angels: music of John Zorn

- Our favorite time series: music, speech [2]



(a) Standard Spectrogram of music excerpt



(b) CQ-Spectrogram of music excerpt

[2] Mark Feldman, Sylvie Courvoisier: KAFZIEL, from: Book of Angels: music of John Zorn

- (Applied) harmonic analysis studies representation of functions (signals) as superposition of basic waves which reflect the expected structure of a signal class under inspection.
- A sequence $\{g_j : j \in J\} \subseteq \mathcal{H}$ is called *frame*, if there exist $A, B > 0$ such that $\forall f \in \mathcal{H}$

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

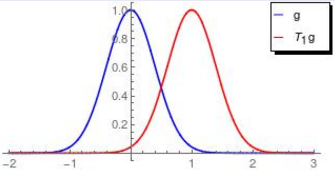
- Also, for a so-called dual frame $\tilde{g}_j$ and $\forall f \in \mathcal{H}$

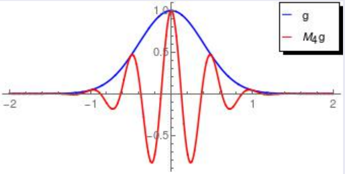
$$f = \sum_{j \in J} \langle f, g_j \rangle \tilde{g}_j.$$

Example: 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 $M_{\beta j} g(t) = g(t) \cdot e^{2\pi i \beta j}$.





- Note: audio signals are *almost* always turned into images before being further processed in deep learning.
- Recent example on music signals: deep CNN learns semantic music content from raw audio data with more than 90% accuracy (!?).

J. Pons, O. Nieto et al,
End-to-end learning for music audio tagging at scale
http://arxiv.org/abs/1711.02520, ISMIR 2018, Paris

- However..

- Note: audio signals are *almost* always turned into images before being further processed in deep learning.
- Recent example on music signals: deep CNN learns semantic music content from raw audio data with more than 90% accuracy (!?).

  J. Pons, O. Nieto et al,
  End-to-end learning for music audio tagging at scale
  http://arxiv.org/abs/1711.02520, ISMIR 2018, Paris

- 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.
- Training time around 4 weeks.

We are therefore facing several questions when learning from audio:

- Which representation would a Neural Network learn?
- To which extent can end-to-end learning improve performance if sufficient amount of data is available?
- Can a representation which encodes beneficial invariances reduce necessary network size, amount of data and training time?

1. Introduction and Motivation
   - Learning is generalization
   - Time Series and Excursus 1

2. Pre-Processing Audio for Deep Learning
   - Spectrogram, Mel-Spectrogram and Gabor Frames
   - Convolutional Neural Networks, Invariance and Gabor Multipliers (Excursus 2)
   - Example: Performance on Singing Voice Detection

3. Designing invariant representations for audio
   - Gabor scattering
   - Complex Autoencoder

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: $\mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{c_1, \dots, c_n\}$, $c_j \in \mathbb{R}$

- **Features** are supposed to make life for learners easier ...
- 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

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

Spectrogram and Gabor Frames

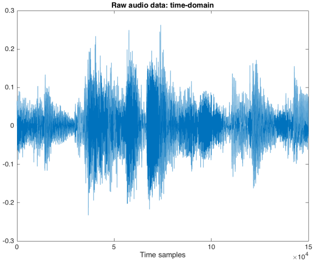
STFT of f with respect to a time-localized window g (e.g. Gaussian):

$$\mathcal{V}_g f(b,k) = \mathcal{F}(f \cdot T_b g)(k) = \int_t f(t) g(t-b) e^{-2\pi i k t} dt$$

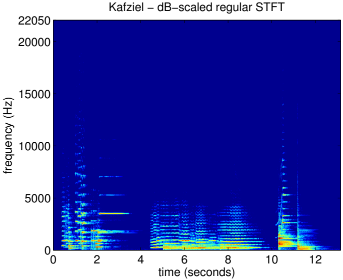
Spectrogram: $S_0(lb_0, k\nu_0) = |\mathcal{V}_g f(lb_0, k\nu_0)|^2 = |\langle f, g_{k,l} \rangle|^2$ where

$$\{g_{k,l} = M_{k\nu_0} T_{lb_0} g : k, l \in \mathbb{Z}\} \ldots \text{ Gabor frame}$$

- Spectrogram expresses essential signal properties much more clearly, or sparsely, than raw audio data.



(c) Raw audio data: time-domain



(d) Spectrogram of music excerpt

Feature Extractor and Mel Spectrogram

- Spectrogram expresses essential signal properties more clearly, or sparsely, than raw audio data.
- ...and induces invariance e.g. to phase shift and local changes (by subsampling)
- Further invariances can by introduced by averaging over computed coefficients.

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.

Definition

Given an augmentation $\mathcal{A}$, that is, a set of bounded operators acting on $\mathcal{X}$: $\mathcal{A} = \{T_p : \mathcal{X} \to \mathcal{X}\}$, then f is said to be invariant to $\mathcal{A}$ with respect to $\mathcal{D} \subset \mathcal{X}$, if $f(T_p(x)) = f(x)$ for all $x \in \mathcal{D}$. If $\mathcal{A}$ is parametrised by a set $\mathcal{P}$, on which a metric $|\cdot|_{\mathcal{P}}$ is defined, then we say that f is locally stable to $\mathcal{A}$, if $\|f(T_p(x)) - f(x)\| \leq C \cdot |p|_{\mathcal{P}} \cdot \|x\|$ for all $x \in \mathcal{D}$, all $p \in \mathcal{P}$ and some constant C.

Note that for categorical problems local stability actually implies local invariance.

- Time-frequency representations can introduce approximate invariance to small, local time-frequency modifications.
- Convolutional Neural Networks adaptively extract local invariances
- Can the extent of desirable invariance be learned by tuning the representation parameters?

Parameters defining layer n in a neural network:

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

- $x_n \in \mathbb{R}^{d(n)}$ – data vector (array) in the n-th layer
  A_n – matrix of weights in n-th layer
  b_n – vector of biases in n-th layer
- nonlinearity σ (applied component wise, e.g. sigmoid, ReLU (Thresholding), modulus)

Structure of Convolutional Neural Networks

Parameters defining layer n in a neural network:

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

- $x_n \in \mathbb{R}^{d(n)}$ – data vector (array) in the n-th layer
  A_n – matrix of weights in n-th layer
  b_n – vector of biases in n-th layer
- nonlinearity σ (applied component wise, e.g. sigmoid, ReLU (Thresholding), modulus)
- Convolutional layers of CNNs: A_n are block-Toeplitz. (Front-end, Feature-Extraction)
- Dense layers: general A_n. (Back-end, Classification stage)
- Parameters $\theta = (A_n, b_n)_{n=1}^{N_p}$ are learned by gradient descent algorithms.

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!

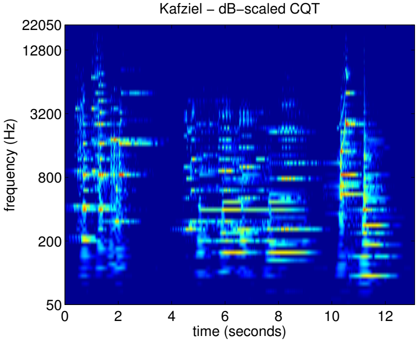
`http://ofai.at/~jan.schlueter/pubs/2016_ismir/alexanderross/index.html`

The architecture has a total number of 1.41 million weights (91% for the dense layers), but far less data points for learning, and leads to an error rate of less than 7% (on unseen data).

Linear sampling in frequency → most energy accumulated in lower frequency channels.

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

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



(e) CQ-Spectrogram of music excerpt

Non-stationary Gabor frames:

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

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 $M \approx N$.

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.

Idea: learn the parameters of the adaptive filter bank.
Expectation: results should out-perform mel-spectrogram

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

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:

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

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*

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

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

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

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*

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

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

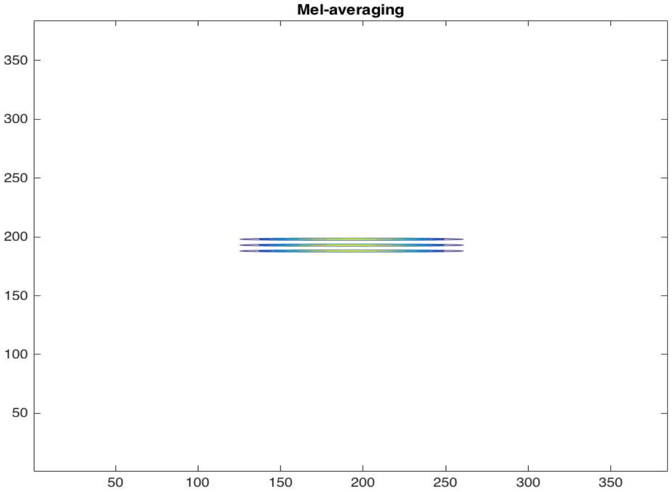
$$|\,\mathrm{MS}_g(f)(b,\nu) - \mathrm{FB}_{h_\nu}(f)(b,\nu)| \leq \|\mathcal{V}_g g \cdot \mathcal{M}^\nu - \mathcal{V}_{h_\nu} h_\nu \cdot \mathcal{M}^\nu_F\|_2 \|f\|_2^2$$

In particular, if

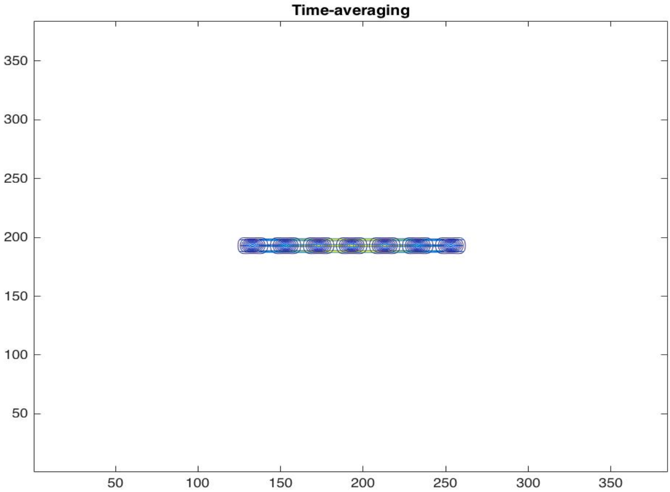
$$V_{h_{\nu_k}} h_{\nu_k}(x,\xi) \cdot \mathcal{F}(\varpi_{\nu_k})(\xi) = V_g g(x,\xi) \cdot \mathcal{F}^{-1}(\Lambda_{\nu_k})(x),$$

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* $\mathbb{Z}$ $(\alpha = 1)$*.*

IDEA of Proof:



IDEA of Proof:



- Start from $S_0(\alpha l, \beta k) = |\mathcal{V}_g f(\alpha l, \beta k)|^2 = |\mathcal{F}(f \cdot T_{\alpha l} g)(\beta k)|^2$.

- Start from $S_0(\alpha l, \beta k) = |\mathcal{V}_g f(\alpha l, \beta k)|^2 = |\mathcal{F}(f \cdot T_{\alpha l} g)(\beta k)|^2$.
- Then, with $\mathbf{m}(k, l) = \delta(\alpha l - b)\Lambda_\nu(\beta k)$:

$$\begin{aligned} \mathrm{MS}_g(f)(b, \nu) &= \sum_k |\mathcal{F}(f \cdot T_b g)(\beta k)|^2 \cdot \Lambda_\nu(\beta k) \\ &= \langle \sum_k \sum_l \mathbf{m}(k, l) \langle f, M_{\beta k} T_{\alpha l} g \rangle M_{\beta k} T_{\alpha l} g, f \rangle \end{aligned}$$

- Start from $S_0(\alpha l, \beta k) = |\mathcal{V}_g f(\alpha l, \beta k)|^2 = |\mathcal{F}(f \cdot T_{\alpha l} g)(\beta k)|^2$.
- Then, with $\mathbf{m}(k,l) = \delta(\alpha l - b)\Lambda_\nu(\beta k)$:

$$\begin{aligned} \mathrm{MS}_g(f)(b,\nu) &= \sum_k |\mathcal{F}(f \cdot T_b g)(\beta k)|^2 \cdot \Lambda_\nu(\beta k) \\ &= \langle \sum_k \sum_l \mathbf{m}(k,l) \langle f, M_{\beta k} T_{\alpha l} g \rangle M_{\beta k} T_{\alpha l} g, f \rangle \end{aligned}$$

- Mel-coefficients can thus be interpreted via a Gabor multiplier: $\mathrm{MS}_g(f)(b,\nu) = \langle G^{\alpha,\beta}_{g,\mathbf{m}} f, f \rangle$.

- Start from $S_0(\alpha l, \beta k) = |\mathcal{V}_g f(\alpha l, \beta k)|^2 = |\mathcal{F}(f \cdot T_{\alpha l} g)(\beta k)|^2$.
- Then, with $\mathbf{m}(k,l) = \delta(\alpha l - b)\Lambda_\nu(\beta k)$:

$$\begin{aligned} \mathrm{MS}_g(f)(b,\nu) &= \sum_k |\mathcal{F}(f \cdot T_b g)(\beta k)|^2 \cdot \Lambda_\nu(\beta k) \\ &= \langle \sum_k \sum_l \mathbf{m}(k,l) \langle f, M_{\beta k} T_{\alpha l} g \rangle M_{\beta k} T_{\alpha l} g, f \rangle \end{aligned}$$

- Mel-coefficients can thus be interpreted via a Gabor multiplier: $\mathrm{MS}_g(f)(b,\nu) = \langle G^{\alpha,\beta}_{g,\mathbf{m}} f, f\rangle$.
- Alternative operator representation (*spreading function* η_H):

$$Hf(t) = \int_x \int_\xi \eta_H(x,\xi) f(t-x) e^{2\pi i t\xi} d\xi dx.$$

- Gabor multiplier's spreading function
  $\eta_{g,\mathbf{m}}^{\alpha,\beta}(x,\xi) = \mathcal{M}(x,\xi)\mathcal{V}_g g(x,\xi)$ where
  $\mathcal{M}(x,\xi) = \mathcal{F}_s(\mathbf{m})(x,\xi) = \sum_k \sum_l \mathbf{m}(k,l)e^{-2\pi i(\alpha l\xi - \beta kx)}$.

M. Dörfler, T. Grill, et al: "Basic Filters for Convolutional Neural Networks Applied to Music: Training or Design?" *Neural Computing and Applications, 2018*, https://arxiv.org/abs/1709.02291, 2017.

- Gabor multiplier's spreading function $\eta^{\alpha,\beta}_{g,\mathbf{m}}(x,\xi) = \mathcal{M}(x,\xi)\mathcal{V}_g g(x,\xi)$ where $\mathcal{M}(x,\xi) = \mathcal{F}_s(\mathbf{m})(x,\xi) = \sum_k \sum_l \mathbf{m}(k,l)e^{-2\pi i(\alpha l\xi - \beta k x)}$.
- Equally rewrite the time-averaging operation as Gabor multiplier:

$$\mathrm{FB}_{h_\nu}(f)(b,\nu) = \langle G^{\alpha,\beta}_{\check{h}_\nu,\mathbf{m}_F} f, f\rangle.$$

with $\mathbf{m}_F(k,l) = T_b\varpi_\nu(l)\delta(\beta k)$ and spreading function $\eta^{\alpha,\beta}_{h_\nu,\mathbf{m_F}}(x,\xi) = \mathcal{M}_{\mathcal{F}}(x,\xi)\mathcal{V}_{h_\nu}h_\nu(x,\xi)$.

M. Dörfler, T. Grill, et al: "Basic Filters for Convolutional Neural Networks Applied to Music: Training or Design?" *Neural Computing and Applications, 2018*, https://arxiv.org/abs/1709.02291, 2017.

- Gabor multiplier's spreading function $\eta^{\alpha,\beta}_{g,\mathbf{m}}(x,\xi) = \mathcal{M}(x,\xi)\mathcal{V}_g g(x,\xi)$ where $\mathcal{M}(x,\xi) = \mathcal{F}_s(\mathbf{m})(x,\xi) = \sum_k \sum_l \mathbf{m}(k,l)e^{-2\pi i(\alpha l\xi - \beta kx)}$.
- Equally rewrite the time-averaging operation as Gabor multiplier:

$$\mathrm{FB}_{h_\nu}(f)(b,\nu) = \langle G^{\alpha,\beta}_{\check{h}_\nu,\mathbf{m}_F} f, f\rangle.$$

  with $\mathbf{m}_F(k,l) = T_b\varpi_\nu(l)\delta(\beta k)$ and spreading function $\eta^{\alpha,\beta}_{h_\nu,\mathbf{m_F}}(x,\xi) = \mathcal{M}_{\mathcal{F}}(x,\xi)\mathcal{V}_{h_\nu}h_\nu(x,\xi)$.
- Comparing the spreading functions leads to claimed result.

M. Dörfler, T. Grill, et al: "Basic Filters for Convolutional Neural Networks Applied to Music: Training or Design?' *Neural Computing and Applications, 2018*, https://arxiv.org/abs/1709.02291, 2017.

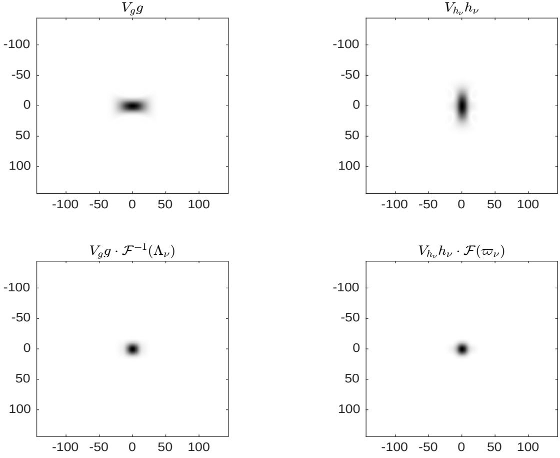


Figure: Spreading functions of operators defining different feature extractors.

Therefore, adaptive filter bank with subsequent time-averaging over learned intervals yields a more expressive feature- network pair than using classical Mel-coefficients.

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.

Therefore, adaptive filter bank with subsequent time-averaging over learned intervals yields a more expressive feature- network pair than using classical Mel-coefficients.

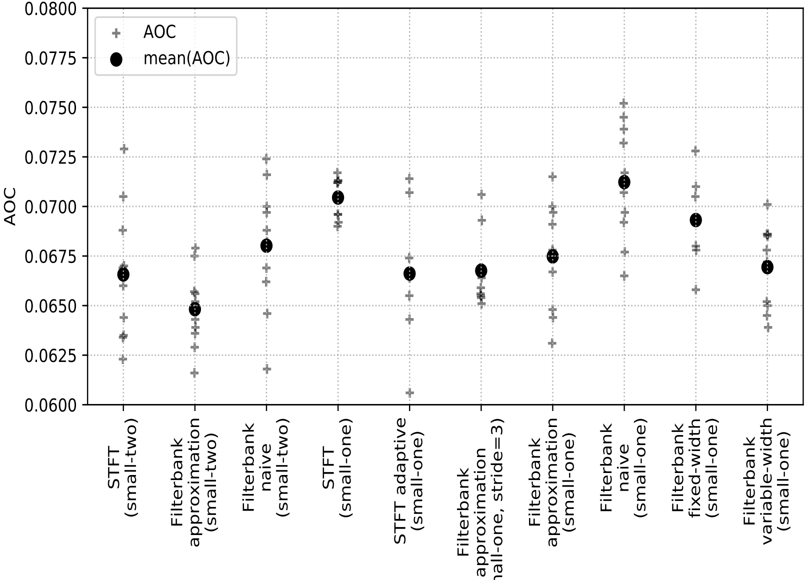
Theorem

Consider CNNs $\mathcal{N}_1, \mathcal{N}_2$ *with* D_c *convolutional layers.*
$\mathcal{N}_2$ *has an additional convolutional layer, preceding the* D_c *convolutional layers and comprising a finite number of convolutional kernels with sufficient length in time-direction and length* 1 *in frequency direction.*
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* $\mathrm{MS}_g = \mathrm{FB}_{h_\nu}$.

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.

Equivalence of feature-network pairs - empirical results



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

Proposition

Introducing invariance to augmentation $\mathcal{A}$ in a stable learning algorithm leads to a reduction of the generalization error by a factor proportional to $\mathcal{N}(\mathcal{D})/\mathcal{N}(\mathcal{A}(\mathcal{D}))$. Here, $\mathcal{N}(\mathcal{D})$ is the covering number of a metric space.

(Example: rotation invariance in images).
Hence: invariant feature extractor leads naturally to invariant learning algorithm and thus reduces the generalization gap!

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

Proposition

Introducing invariance to augmentation $\mathcal{A}$ in a stable learning algorithm leads to a reduction of the generalization error by a factor proportional to $\mathcal{N}(\mathcal{D})/\mathcal{N}(\mathcal{A}(\mathcal{D}))$. Here, $\mathcal{N}(\mathcal{D})$ is the covering number of a metric space.

Observation: Invariance in CNNs is obtained by concatenating learned filter-bank representations with non-linearities.

💡 May look for representations which directly provide desired invariances.

Inspired by Mallat's wavelet-based scattering transform, we introduced Gabor Scattering: iteratively applies Gabor transforms with different subsampling schemes, a non-linearity and subsequent time-averaging.

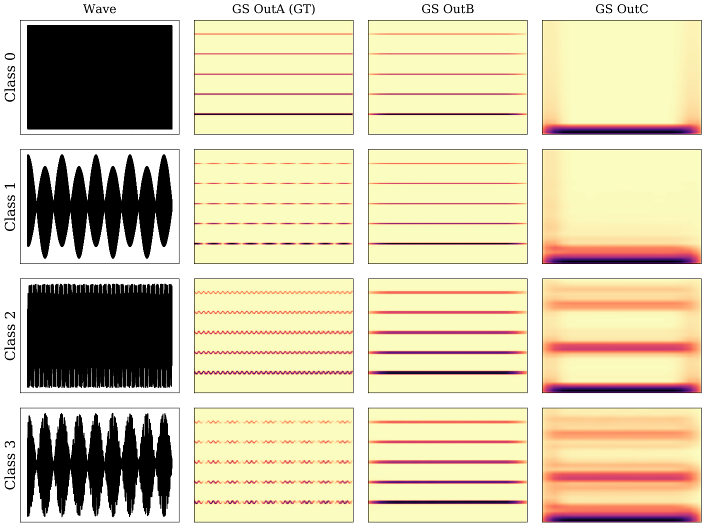
Definition (Gabor Scattering, schematic)

For given Gabor frames $\{M_{\beta_\ell j} T_{\alpha_\ell k} g_\ell\}$, and non-linearities σ_l, $\ell = 1, \ldots, N$, the j-th component in the ℓ-th layer of Gabor scattering defined by

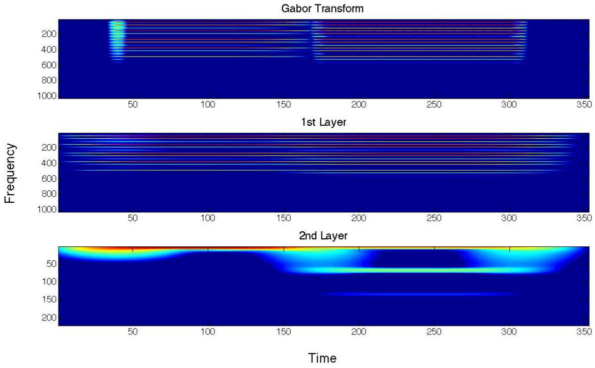
$$f_\ell^j(k) = \sigma_\ell(\langle f_{\ell-1}, M_{\beta_\ell j} T_{\alpha_\ell k} g_\ell \rangle_{\mathcal{H}_{\ell-1}}),$$

where f_0 is the input signal and $f_{\ell-1}$ is an output-vector from the previous layer. Time-averaging with ϕ_ℓ yields ***Feature Extractor*** :

$$\Phi(f) := \bigcup_{\ell=0}^{N} \bigcup_j \{f_\ell^j * \phi_\ell\}.$$

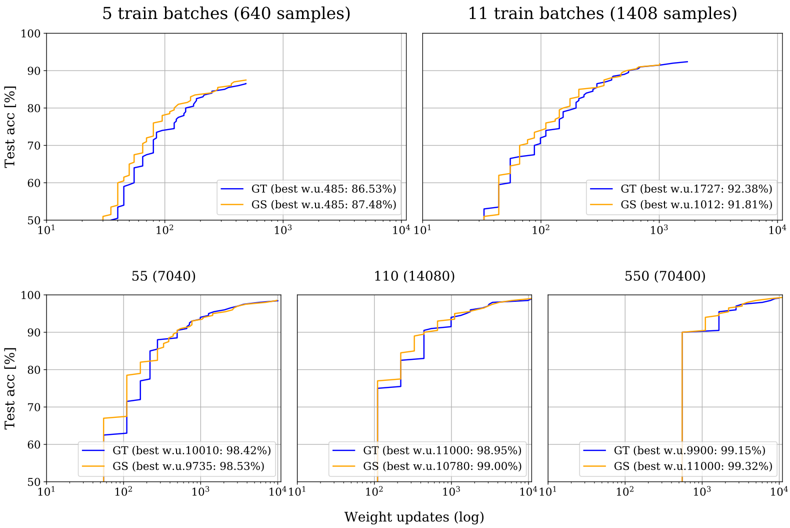


Layers of Gabor Scattering on Synthetic Data

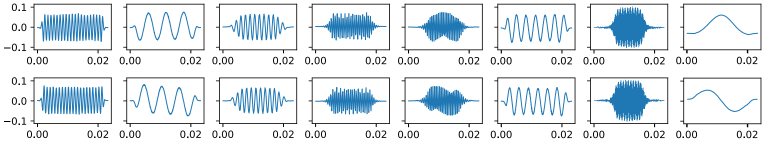


- 1st layer in Gabor scattering locally invariant to amplitude variations.
- 2nd layer locally invariant to frequency variations.

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



Comparison of Performance between Spectrogram and Gabor Scattering on GoodSounds Data

- Propose an architecture called Complex Autoencoder (CAE): learns features invariant to orthogonal transformations.
- Mapping signals onto complex basis functions learned by the CAE results in a transformation-invariant "magnitude space" and a transformation-variant "phase space".
- 

- Some examples of real (top) and imaginary (bottom) basis vectors learned from audio signals by imposing shift-invariance.

S.Lattner, MD, A. Arzt: "Learning Complex Basis Functions for Invariant Representations of Audio ," *ISMIR 19*, 2019.

Invariance: Complex Autoencoder

Principal Idea: aim at learning orthogonal transformations encoding invariances of a class of signals assumed to be useful for learning task at hand.

Proposition

If an orthogonal transformation $\psi : \mathbb{R}^N \to \mathbb{R}^N$ is diagonalised by a unitary matrix $\mathbf{W}$, then the feature vector given by $|\mathbf{W}\mathbf{x}|$ for all $\mathbf{x} \in \mathbb{R}^N$ is invariant to ψ. In other words, we have $|\mathbf{W}\mathbf{x}| = |\mathbf{W}\psi(\mathbf{x})|$ for all $\mathbf{x} \in \mathbb{R}^N$.

Invariance-property of the magnitude space leads to state-of-the-art results in audio-to-score alignment and repeated section discovery for audio.

Commuting operators possess simultaneous diagonalization.

Summary

- Deep Learning has reached most areas of relevance, both in research and everyday life
- For complex problems, satisfactory results require huge amount of data and solving them consumes a lot of energy.
- Designing smart feature extractors can lead to smaller generalization gap and sampling error with less data/computation time.
- Encoding known invariances plays an important role in reducing generalization error and thus improving performance on unseen (validation) data.

Thanks for your attention! Questions? Remarks?