Inside the Spectrogram: Convolutional Neural Networks in Audio Processing.

Monika Dörfler, Roswitha Bammer
NuHAG, Faculty of Mathematics
Oskar-Morgenstern-Platz 1, 1090 Vienna

Thomas Grill
Austrian Research Institute for Artificial Intelligence
Freyung 6, 1010 Vienna

***Abstract*—Convolutional Neural Networks have established a new standard in many machine learning applications not only in image but also in audio processing. In this contribution we investigate the interplay between the primary representation mapping a raw audio signal to some kind of image (feature) and the convolutional layers of an ensuing neural network. We introduce a new notion of equivalence of feature-network pairs and show the relation of feature and networks for the example of mel-spectrogram input on the one hand and varying analysis windows on the other hand.**

I. Introduction

Convolutional Neural Networks (CNNs) have gained a lot of attention due to recent impressive successes in areas such as image processing, speech recognition and artificial intelligence/ human-machine interaction (e.g. Go playing). Initially introduced in image processing, CNNs have recently achieved similarly convincing results in audio processing and in particular in music information retrieval (MIR). This fact is by no means obvious, since audio data are quite differently structured than natural image data. In particular, the input to the CNNs in learning tasks such as classification is hardly ever raw audio, but the given data sequences undergo some pre-processing before being fed into the first convolutional layer.

While deep convolutional networks based on fixed wavelet [1], [2] or general families of semi-discrete frames [3] have recently been studied in the context of so-called scattering transforms, the interplay between a chosen representation and standard CNNs is the focus of the present contribution.

More precisely, we are interested in the influence of the chosen primary representation on the over-all expressivity of the CNN, in particular, we study, when the combination of a particular primary representation with a certain network architecture leads to the same family of functions if weight parameters may vary. We are motivated by the fact that rather specific representations such as the mel spectrogram are commonly used as primary data representation and this choice has hardly been discussed. For this special choice we show, that equivalent results can be achieved by frequency-adaptive filters, as e.g. provided by constant-Q filter banks, if a time-averaging layer is added. More generally, we show that the choice of different primary analysis windows can basically be compensated for by specific choice of kernels in subsequent convolutional layers. This does not preclude, however, the empirical observation that the choice of an appropriate signal representation alleviates the learning process.

We will next introduce the concepts of classical and non-stationary Gabor frames, as well as a formal description of CNNs. Starting from a data-based notion of network-equivalence, we then state some results on equivalent expressivity of representation/network families.

A. Notation

The Fourier transform of a function f will be denoted by $\mathcal{F}(f)$ and we use the normalisation

$$\mathcal{F}(f)(\omega) = \int_t f(t) e^{-2\pi i \omega t} dt.$$

Denoting a time-shift by b by T_b, we use the following notation for the short-time Fourier transform of a function f with respect to a window g:

$$\mathcal{V}_g f(b,k) = \mathcal{F}(f \cdot T_b g)(k) \tag{1}$$

and obtain the spectrogram as $S_0(b,k) = |\mathcal{V}_g f(b,k)|^2$.

We consider annotated data sets $\mathcal{D}$, which are defined as sets of vectors

$$\mathcal{D} = \{(f_i, c_i) \in \mathbb{R}^L \times \mathbb{R}, i \in \mathcal{I}\},$$

where L is the input dimension, e.g. the length of a music excerpt in samples, and c_i are the class labels.

II. Spectrogram and Gabor frames

In practice, subsampled versions of the highly redundant spectrogram are used. In this case, we can speak of the samples of the spectrogram (1) as coefficients of f with respect to a *Gabor frame*. Denoting a frequency-shift by ν by M_ν, the elements of a Gabor frame are given by

$$\{g_{k,l} = M_{k\nu_0} T_{lb_0} g : k, l \in \mathbb{Z}\}.$$

Separating time- and frequency dimension, we obtain a matrix $S_0 \in \mathbb{R}^{M\times N}$, i.e. M time-samples and N frequency channels, where $S_0(l,k) = |\langle f, g_{k,l}\rangle|^2$.

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

$$\{h_{\nu,l} = T_{lb_k} h_{\nu_k} : l \in \mathbb{Z}, k \in \mathcal{J}\}$$

with time-shift parameters b_k chosen separately for each band in some index set $\mathcal{J}$. For details on frequency adaptive non-stationary Gabor frames, cf. [4]. Note that the different time-shift parameters lead to varying lengths in the output of each channel. Since a detailed treatment of the resulting numerical and implementation problems is beyond the scope of this contribution, we assume that $b_k = 1$ for all channels. Thus we obtain, as in the Gabor frame case a matrix S_a of size $M \times N$ containing the coefficients of f with respect to the non-stationary Gabor frame, i.e.

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

A. Feature Extractor

The spectrogram as defined above can be understood as a *feature extractor*, since it maps raw (audio) data to a more structured representation, which expresses essential signal properties much more clearly. In general, we call a mapping $\Phi : \mathbb{R}^L \mapsto \mathbb{R}^{M_1\times\ldots\times M_d}$ a feature extractor, if the feature array $\Phi(f)$ is, in some sense, more useful for the machine learning problem at hand.

EXAMPLE II.1 (Mel spectrogram). In most MIR tasks, the inputs are derived from rather short snips, that is, about 2 to 4 seconds of sound. Considering a sampling rate of 22050 Hz, a window size of 2048 samples and a time shift parameter of 512 samples, i.e., $23ms$, the resulting spectrogram (containing positive frequencies only) is of size $M \times N = 1024 \times 130$, where the latter is the time dimension. It is immediately obvious, that the frequency dimension is, in some sense, over-sampled. In particular, the higher frequency regions contain hardly any information. The mel spectrogram is derived from S_0 by taking weighted averages over frequency channels defined by the mel-scale [5]:

$$\mathrm{MS}_g(f)(l,\nu) = \sum_k S_0(l,k)\cdot\Lambda_\nu(k). \tag{2}$$

III. CNN WITH SPECTROGRAM INPUT

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

$$x_{n+1} = \sigma(A_n x_n + b_n) \tag{3}$$

where x_n is the data vector, or array, in the n-th layer, A_n represents a linear operator, b_n is a vector of biases in the n-th layer and the nonlinearity σ is applied component wise. Note that in each layer the array x_n may have a different dimension. Now, in the case of convolutional layers of CNNs, the matrix A has a particular structure for the convolutional layers, namely, it is a block-Toeplitz matrix, or, depending on the implementation of the filters, a concatenation of circular matrices, each representing one convolution kernel. There may be an arbitrarily high number of convolutional layers, followed by a certain number of so-called dense layers, for which A_n is again an arbitrary linear operator.

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: For $1 \le p \le \infty$, we define $K \times L$ pooling as the operator mapping an $M \times N$ matrix S_0 to a $M/K \times N/L$ matrix $S_1$by
$$S_1(m,n) = P_p^{K,L} S_0(m,n) = \|v_{S_0}^{m,n}\|_p \tag{4}$$
where $v_{S_0}^{m,n}$ is the vector consisting of the matrix entries $S_0((m-1)\cdot K+1,\ldots,m\cdot K; n\cdot(n-1)\cdot L+1,\ldots,n\cdot L)$ for $m = 1,\ldots,M/K$, $n = 1,\ldots,N/L$. Max-pooling, i.e. $p = \infty$, has been the most successful choice.
- A nonlinearity $\sigma : \mathbb{R} \mapsto \mathbb{R}$, whose action is always to be understood component-wise.

Given the above definitions we can now write the output of a convolutional layer , given an input matrix S_n, as follows:

$$\underbrace{S_{n+1} = P_\infty^{K_n,L_n}\sigma\Big(\sum_{k_{n-1}=1}^{K_{n-1}} \underbrace{S_n * w_{k_n}^{k_{n-1}}}_{M_n\times N_n\times K_{n-1}\times K_n} + b^{k_n}\cdot\mathbf{1}\Big)}_{M_n/K_n\times N_n/L_n\times K_n}$$

where S_n, the output of layer $n-1$, is an $M_n \times N_n \times K_{n-1}$ array, and $\mathbf{1}$ is an all-ones matrix of size $M_n/K_n \times N_n/L_n$.

The convolutional layers are followed by several (often two) dense layers as described in (3) and a final dense layer providing the single output. Observe, that in the second convolutional layer $K_1 \cdot K_2$ convolutional kernels, that is, K_2 kernels for each of the K_1 channels in the first layer are learned, followed by a summation over the first layer channels.

IV. EQUIVALENCE OF FEATURE-NETWORK PAIRS

A CNN $\mathcal{N}$ is understood as the architecture defined by the input dimension, the number of convolutional layers D_c, the number of dense layers D_d, the number and size of convolutional kernels in each convolutional layer and the type of non-linearity and pooling in each layer. We call the parameter vector comprising all the weights occurring in the network by $\theta \in \mathbb{R}^p$ and thus a concrete realisation given a parameter vector θ is denoted by $\mathcal{N}(\theta)$. We then define the following notion of equivalence.

Definition IV.1 (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

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

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

A. Varying the analysis window

In signal processing, adaptive signal representations have been the object of investigations in the past decades, in particular in the context of sparse representations, which rely on a certain adaptation of the underlying dictionary to the class of data of interest. In audio processing, where data are inherently highly non-stationary, using adaptive representations has led to improvement in various processing tasks, cf. [6]. In particular, various window-shapes are associated to better resolution of either frequency- or time-concentrated signal components, respectively, both of which may have significant impact on the analysis and subsequent classification of audio signals. Simultaneously using spectrograms based on different window shapes has also led to improvement in MIR tasks [7]. In the next proposition we use a similar technique as in Proposition IV.5 in order to understand to which extent an appropriate choice of filter shapes in convolutional layers can compensate different resolution quality of the primary feature extractor. In order to distinguish spectrograms obtained from varying analysis windows, we now write $S_0^g(l,k) = |\langle f, g_{k,l}\rangle|^2$. In the sequel $\mathcal{F}_s$ denotes the *symplectic* Fourier transform, defined in (7) below.

Proposition IV.2. *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 $S_0^g * w_g = S_0^h * w_h$.

Proof: The proof is based on the observation that the two-dimensional convolutions applied to a spectrogram S_0 can be written via the operation of a so-called *Gabor multiplier*, cf. [8], [9] on any given function in the sense of a bilinear form. Given a window function g and a function $\mathbf{m} : \mathbb{Z}\times\mathbb{Z} \mapsto \mathbb{C}$, the corresponding Gabor multipler $G_{g,\mathbf{m}}$ is defined as

$$G_{g,\mathbf{m}} f = \sum_k \sum_l \mathbf{m}(k,l)\langle f, g_{k,l}\rangle g_{k,l}. \tag{5}$$

We next derive the expression of $S_0^g * w_g$ by an appropriately chosen Gabor multiplier:

$$\begin{aligned} S_0^g * w_g(l,k) &= \sum_{k',l'} |\langle f, g_{k',l'}\rangle|^2 w_g(l-l', k-k') \\ &= \langle \sum_{k'} \sum_{l'} \mathbf{m}^\lambda(l',k')\langle f, g_{k',l'}\rangle g_{k',l'}, f\rangle \end{aligned}$$

where we set

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

for $\lambda = (l,k)$, i.e. $S_0^g * w_g(\lambda) = \langle G_{g,\mathbf{m}^\lambda} f, f\rangle$.

We now denote a time-frequency shift by $\pi(x,\xi) = M_\xi T_x$ and recall that an operator H can equally be written by means of its *spreading function* η_H as

$$Hf = \int_x \int_\xi \eta_H(x,\xi)\pi(x,\xi) f \, d\xi dx$$

and two operators H_1, H_2 are equal if and only if their spreading functions coincide, see [9], [8]. A Gabor multiplier's spreading function $\eta_{g,\mathbf{m}}$ is given by ([9])

$$\eta_{G_{g,\mathbf{m}}}(x,\xi) = \mathcal{F}_s(\mathbf{m})(x,\xi)\mathcal{V}_g g(x,\xi), \tag{6}$$

where

$$\mathcal{F}_s(\mathbf{m})(x,\xi) = \sum_{k,l} \mathbf{m}(k,l) e^{-2\pi i(l\xi - kx)} \tag{7}$$

is the symplectic Fourier transform of $\mathbf{m}$. Writing $z = (x,\xi)$ we obtain by straight-forward calculation:

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

Since the extra phase factor $e^{-2\pi i(l\xi - kx)}$ is the same if the same sampling points are used for sampling both $\mathcal{V}_g g$ and $\mathcal{V}_h h$ and since the symplectic Fourier transform is self-inverse, the claim of the proposition follows. ∎

An illustration of the above proposition is given in Figure1, where the spectrograms of a synthetic signal, generated with two windows of different lengths, namely 2048 (long) and 512 (short) samples are shown, along with the output of a convolution ad max-pooling step.

REMARK IV.3. Note that the condition in Proposition IV.2 is related to the well-known fact that the spectrogram, a member of Cohen's class of bilinear time-frequency distributions, can be written as two-dimensional convolution of the Wigner transforms of the window g and the input signal f, cf. [10].

From Proposition IV.2 we can now deduce a statement about two different network architectures.

Theorem IV.4. *Consider two convolutional neural networks $\mathcal{N}_j$, $j = 1, 2$ 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 $(S_0^g, \mathcal{N}_1)$ and $(S_0^h, \mathcal{N}_2)$ are equivalent.

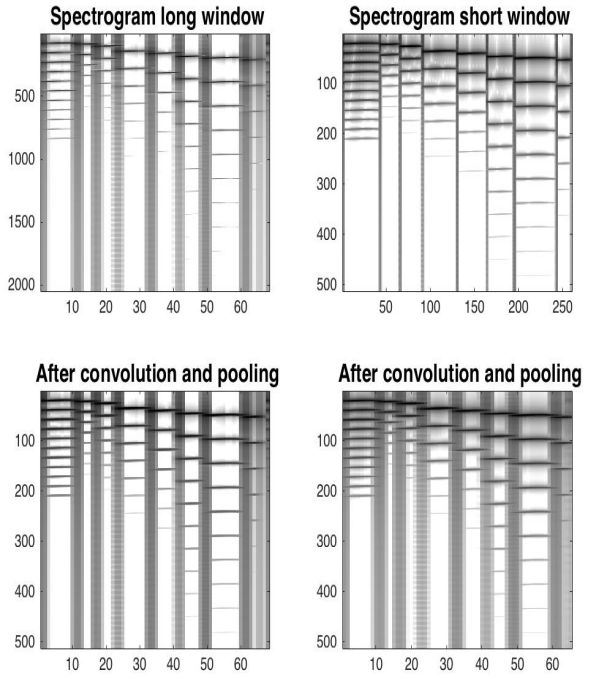

Fig. 1. Synthetic signal consisting of several damped notes. Left upper plot shows spectrogram obtained with long window g, upper right plot shows spectrogram obtained with short window. Lower plot show output of convolution and subsequent max-pooling chosen to result in equivalent matrix size. It is visible, that the convolution leads to similar results from previously different spectrograms.

B. The Mel-spectrogram and adaptive filter banks

Anden and Mallat showed in [2], that the mel-spectrogram can be approximated by time-averaging the absolute values-squared of a wavelet transform. Here, we make their considerations precise by showing that for a particular choice of windows, mel-coefficients can be obtained by using adaptive filters. We define the time-averaging operator with respect to a family of averaging functions ϖ_{ν_k}, $k \in \mathcal{I}$ and a non-stationary Gabor frame as

$$\mathrm{TA}^h_{\varpi}(f)(l,\nu_k) := \sum_{l'} S_a(l',\nu_k)\cdot \varpi_{\nu_k}(l'-l). \qquad (8)$$

Proposition IV.5. *Let an analysis window g and mel-filters Λ_ν be given, for $k \in \mathcal{I}$. If for each k, the windows h_{ν_k} and time-averaging functions ϖ_{ν_k} are chosen such that*

$$V_{h_{\nu_k}} h_{\nu_k}(x,\xi)\cdot \mathcal{F}(\varpi_{\nu_k})(\xi) = \\ = V_g g(x,\xi)\cdot \mathcal{F}^{-1}(\Lambda_{\nu_k})(x), \qquad (9)$$

then the mel-spectrogram coefficients $\mathrm{MS}_g(f)(l,\nu_k)$ can be obtained by time-averaging the filtered signal's absolute value squared i.e. $\mathrm{MS}_g(f) = \mathrm{TA}^h_{\varpi}(f)$.

The proof is based on a similar idea as that of Proposition IV.2 and will be given in detail in [11]. The following statement is an immediate consequence.

Theorem IV.6. *Consider $\mathcal{N}_1$ as a convolutional network with D_c convolutional layers and a network $\mathcal{N}_2$ which is the same as $\mathcal{N}_1$, except that it has 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 in $\mathcal{N}_1$. If the windows g, h_{ν_k} and the mel-filters Λ_{ν_k} are chosen such that (9) holds, then $(\mathrm{MS}_g, \mathcal{N}_1)$ is subordinate to $(S_a, \mathcal{N}_2)$.*

V. Conclusion and Perspectives

We introduced a new notion of equivalence between feature-network pairs and gave two examples for which (partial) equivalence can be derived. Future work will study more complex connections by considering deeper network layers; furthermore, in parallel to Theorem IV.6, the impact of more general adaptive frames and the connection to explicit sparsity priors on the coefficients, in particular for data sets $\mathcal{D}$ with a known special structure will be investigated.

Acknowledgment

This work was supported by the Vienna Science and Technology Fund (WWTF) project SALSA (MA14-018) and the Uni:docs Fellowship Programme for Doctoral Candidates in Vienna.

References

[1] S. Mallat, "Group Invariant Scattering," *Comm. Pure Appl. Math.*, vol. 65, no. 10, pp. 1331–1398, 2012.
[2] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, 2014.
[3] T. Wiatowski and H. Bölcskei, "A mathematical theory of deep convolutional neural networks for feature extraction," *CoRR*, vol. abs/1512.06293, 2016.
[4] 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.
[5] S. S. Stevens, "A scale for the measurement of the psychological magnitude pitch," *Acoustical Society of America Journal*, vol. 8, 1937.
[6] P. Balazs, M. Dörfler, M. Kowalski, and B. Torrésani, "Adapted and adaptive linear time-frequency representations: a synthesis point of view," *IEEE Signal Processing Magazine*, vol. 30, no. 6, pp. 20–31, 2013.
[7] T. Grill and J. Schlüter, "Music boundary detection using neural networks on combined features and two-level annotations," in *ISMIR 2015 conference*, 2015.
[8] H. G. Feichtinger and W. Kozek, "Quantization of TF lattice-invariant operators on elementary LCA groups," in *Gabor analysis and algorithms*, ser. Appl. Numer. Harmon. Anal., H. G. Feichtinger and T. Strohmer, Eds. Birkhäuser Boston, 1998, pp. 233–266.
[9] M. Dörfler and B. Torrésani, "Representation of operators in the time-frequency domain and generalized Gabor multipliers," *J. Fourier Anal. Appl.*, vol. 16, no. 2, pp. 261–293, 2010.
[10] L. Cohen, *Time-Frequency Analysis: Theory and Applications.*, ser. Prentice Hall Signal Processing Series. Prentice Hall, 1995.
[11] M.Dörfler, T. G. adn R. Bammer, and A. Flexer, "Basic filters for convolutional neural networks: Training or design?" *in preparation*, 2017.