Extraction of Temporal Features with the Scattering Transform

Daniel Haider, Peter Balazs

Acoustics Research Institute

Abstract. In this paper we will motivate the scattering transform in the context of convolutional neural networks (CNNs) to gain deeper insights of how these networks learn multi-scale audio features. This transform computes a layered structure, similar to a CNN but with no learning involved. Applied to audio it is able to capture temporal dependencies beyond those, possible for common time-frequency representations. This has been demonstrated by experiments for modulations of single tones. We will here provide a setup, which extends the temporal range to the scale where rhythm and tempo live, allowing very intuitive explanations of the observations. This should emphasize the power of this deterministic network to enable a better understanding of what's going on inside a neural network when "listening".

Keywords: Time-Frequency Representation, Audio Features, Deep Learning, Scattering Transform

1 Introduction

One of the main goals in Computer Science these days seems to be anthropomorphizing computers and in particular, make them able to perceive. In the case of audio, a fundamental thing to archieve would be to teach a computer to "listen" like humans do. When we talk about "teaching" a computer, we get into the field of machine learning where a specific model has been on the rise, outperforming many of the common techniques so far and archieving impressive results in various learning tasks, the *Convolutional Neural Networks* (CNNs). Although they are a very powerful tool, their layered structure inherits intransparence of what is going on inside, i.e. it is somehow unclear und hard to control what the networks actually learns and especially why it learns certain things. When it wants to learn from audio signals, it clearly needs to capture the information at all different time scales. We will try to understand how this can be archieved. We will start with a small discussion on representations of audio and their lacks in the context of machine learning and ilucidate their connection to CNNs. The insights we get will motivate to introduce a special transform of audio signals which mimics the computational structure of a CNN. It is called *Scattering Transform* and is computed by a cascade of time-frequency transforms setting up a layered network, [1]. We will emphasize on its ability to capture wide range dependencies in the signal and demonstrate this by extracting the rhytmical structure in

terms of tempo of a musical piece in the last chapter. This setup will allow to illustrate the mechanics of a CNN in a very intuitive way and brings us closer to the initial motivation of understanding how a computer can learn to "listen". Moreover, the experiments will enhance that the scattering transform is very powerful tool, yielding interesting representations of audio.

2 Computers Listening

When we listen to music, we identify and interpret patterns on several temporal scales. We perceive pitch and timbre within the scale of milliseconds, capture tempo and rhythmical structures over periods of seconds and understand general progressions as minutes and hours pass. The soundwaves that carry all this information to our ears are pre-processed by our auditory system and transferred to our brain via electrical stimuli which encode the information. Our brain interprets the patterns of these stimuli. So, if we want to make a computer "listen" like humans do, we might have to present it the essential features of the audio signal that are representative of its composition in advance and incorporate specific invariances which are important for abstraction and generalization. This is actually what people do for audio-related machine learning tasks. The data is pre-processed into time-frequency representations, which provide important basic features and used as input for the trainable machines. It has been shown that this pre-processing works particularly well when only small datasets are available, [2]. However, time-frequency representations only show small-scale information of a signal, whereby long-term dependencies inherit necessary information to capture the signal at large, [3]. E.g., how does a CNN extract the rhythmical structure of a song?
In the following we will introduce CNNs and investigate on their ability to extract dependencies over longer scales and ilucidate their intrinsic connection to time-frequency representations.

2.1 Convolutional Neural Networks

Deep Neural Networks (DNNs) build the cornerstone of a relatively new branch of Machine Learning called *Deep Learning* and have established themselves as a very successful tool for various learning tasks, [4]. The idea is to setup a function, that can theoretically approximate any other function via an optimization procedure ("learning"). A DNN has a layered structure, each consisting of single neurons which filter the importance of the information arriving and a non-linear function, called *activation function* that amplifies the "significance" of the neuron to the network. The classical architecture has fully connected layers, i.e. every output neuron interacts with every input neuron. This is computationally expensive and in some sense unnecessarly much information processed. *Convolutional Neural Networks* are a specialized form of DNNs to deal with grid-like data, originally introduced for image processing problems. The idea is to use filters that are much smaller than the input dimension and convolve it with the

input, which can be interpreted as localization of certain properties of the data. This architecture has sparsely connected layers and allows to model a receptive field which captures more focussed. The filters are initially set with random parameters and are adapted during a learning procedure to extract the essential features for the task.
To reduce redundancy and increase invariance, there are several techniques, among them, *pooling*. The operation of pooling computes a "summary" of nearby elements, so it decreases the dimensionality and generates invariances to specific deformations and variations in the data. This also expands the range of the filters in the subsequent layers and allows to capture wider dependencies in the original data. So, a typical layer has three stages: Filtering by several different filters in parallel, applying a non-linear activation function and then dimensionality reduction.
This construction has shown to work very well and led to an immense progress in image-related learning tasks. Clearly it makes totally sense to apply it also on the images obtained by time-frequency decompositions of audio signals, [4], [5]. We will investigate a bit deeper on that setup by looking at the construction of time-frequency representations of audio signals itself.

3 Computers Listening - Revisited

Most of the used time-frequency representations are set up via a filterbank construction, i.e. a bunch of filters, that decompose the signal into different frequency bins. The filtering can be realized via the convolution operation and usually, a modulus or a modulus squared is applied on the computed coefficient. Furthermore, some transformations also inherit dimensionality reduction in time, like scaling or subsampling. It is obvious that this procedure can be seen as computing one layer of a CNN with manually set, instead of learned filters, which points out the fundamental connection between filterbank decompositions and the principle structure of a CNN. In other words, CNNs basically have already all the tools to learn their own "auditory system"!
This observation gives us the chance of understanding something about the learning behaviour of a CNN with respect to time-frequency features. Frequencies in the audible region are referred to as pitch, that means a common time-frequency representation gives fine-scale information. If we assume that a CNN in a training setup might indeed learn something similar in its first layer, the construction of a CNN indicates that wider scales are captured in the subsequent layers (e.g. via pooling).
To see what's going on down there in particular, why not simply perform another time-frequency decomposition.

3.1 The Scattering Transform

The scattering transform is computed by a cascade of time-frequency transforms based on filterbank constructions with subsequent modulus and time averaging

operations applied. In other words, time-frequency decompositions are performed on the single filter outputs of the previously decomposed filter outputs, see Figure 1. This mimics the computational structure of a CNN and allows to get some insights of how *deeper* audio features might be learned by a CNN.
Originally, the transform was introduced by Mallat in [1] and was based on the wavelet transform. It came with a rigorous mathematical analysis, enabling to show translation invariance and stability w.r.t. time-deformations. Later the approach was generalized to semi-discrete frames, which include common filterbank constructions, [6]. In [7] it was used as a feature extractor for audio and it turned out that it is able to represent temporal features beyond those, common time-frequency representations can capture. Events such as note attacks, amplitude and frequency modulation, as well as chord structures can be embraced by this transform, [8]. The examples there show that indeed, as we look deeper into the scattering network, wider structures of the signal are represented. Applied on an amplitude modulated tone it is the envelope that appeared, i.e. the timbre of the sound and in a vibrato tone it is the frequency of the vibrato modulation. We will go beyond the modulation of single tones and bother with temporal features, that live in a larger scale, namely rhythm and tempo.

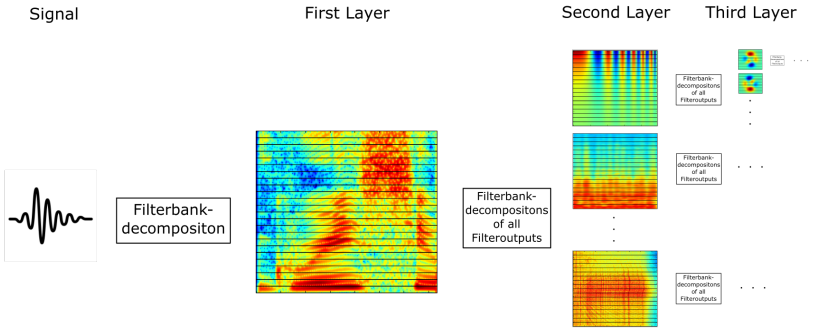

Fig. 1. The scattering procedure of a signal. (The images are just representative and don't depict how the single layers shall look like)

4 Temporal Feature Analysis

In principle, every feature of an audio signal can be interpreted as a temporal feature since a digital audio signal is a time series. In that sense we can find all the information at different time scales; pitch and timbre live within the scale of milliseconds, tempo and rhythm is spread over periods of seconds and general progressions, like the composition of a song pass minutes or even hours. As the

first layer of the scattering transform provides small-scale information, we want to get the scale of rhythm in the second layer, just similar to [8].
Rhythm refers to the timing of events within a musical piece and has different levels of periodicity. In the notation of western music, a hierachial metrical structure is used to distinguish between different time scales. The *Tatum* level (temporal atom) is the smallest and is related to the shortest durational value encountered between two events within the musical piece. The *Tactus* level (beat) is the perceptually most prominent and refers to the rate, most people would "tap" their feet to. Finally the widest, the *Bar* (measure) is related to the length of a rhythmical pattern, [9]. In a classical drumbeat in 4/4 with a kick on the one, a snare on the three and a quarter note hihatpattern, Tatum would correspond to the hihat, Tactum to the snare and the Bar to the whole pattern.
We will define the *tempo* of a musical piece as the speed of the played pattern with respect to the Tactus level. As it is introduced here, tempo is clearly not well-defined since some people might tap differently than others but in most cases it is rather clear or differs only within a power of two. Usually, tempo is measured in bpm (beats per minute) and reaches among different genres and styles of music from 30-300bpm. Embodying the tempo as a periodic pattern of events in time we could also assign a frequency to it; we play here in the range of 0.5-5Hz. As a frequency, this is clearly not audible, but indeed perceivable as a rhythmical pattern. We will see that the scattering transform is also capable of "perceiveing" a rhythmical pattern by depicting its (subsampled) frequency in the second layer.
For many musicians the embodiment of tempo is known as *metronome*. It usually *ticks* in the Tactus tempo or once every Bar and *tacks* according to certain subdivisions. We can model a metronome simply by an impulse train with a stepsize T and consider what we will get when we apply the scattering procedure on it, see Figure 2. We will use a scattering transform based on a sampled Short-Time Fourier Transform (STFT) with a time-hopsize parameter α and a subsequent modulus operation applied, aka. *Gabor scattering*, [10]. As first step, the STFT of the impulse train is computed using a window function ϕ. To avoid overlapping, we choose the length of ϕ to be smaller than T. Since we are moving modulated windows with a stepwidth of α across single peaks of ones and take absolute values afterwards, what we get in every filter output is a smoothed and subsampled version of the metronome, *ticking* now with periodicity αT, instead of T. Now let's compute the second layer. If we chose a large α, we have a higher frequency present than before, so it can be captured and depicted by another STFT, the "conventional" way. To amplify time-shift invariance, we average in time by applying a low-pass filter. Figure 2 illustrates the latter explanation. Note that choosing a non-trivial time step parameter α is crucial for the desired effect and is exactely referred to the principle of pooling we mentioned earlier.
If we want to consider more complex musical signals, we may see such as consisting of tonal, transient and stochastic components. Truly, the transient parts of a signal indicate its rhytmical structure, i.e. percussion, onsets of instruments, etc. In that manner, the scattering transform will detect periodic patterns among

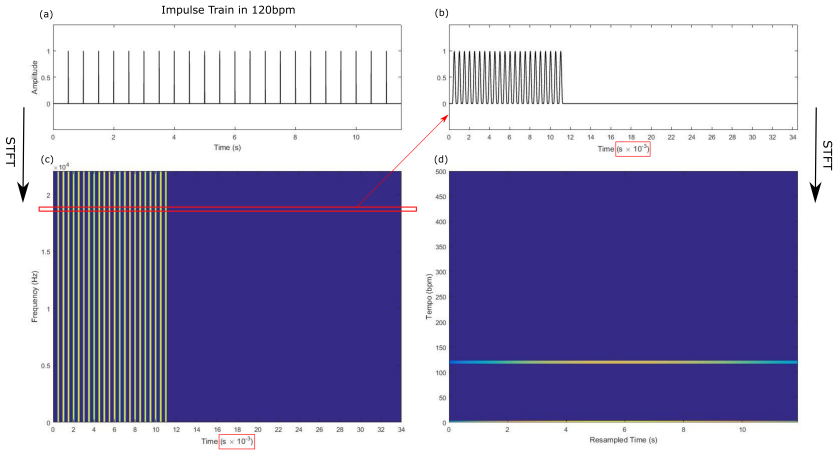

Fig. 2. (a) shows an impulse train of 12 seconds in the tempo of 120bpm, i.e. 2Hz. (c) is the STFT of (a) using a time-hop size of $\alpha = 1000$, it is the first layer of the scattering network (without time-averaging). (b) displays how the single filter outputs look like: the impulse train is smoothed by the window and scaled in time by the factor α, i.e. it is shorter and has a higher frequency, 2000Hz. (d) is then the STFT of (b) with $\alpha = 1$ on a resampled timescale and a "tempo"scale measured in bpm. This is the second layer

the transient arrangement on several scales and extract the temporal information with respect to those. Tweaking the time hop size α emphasizes different scales. In the case of the metronome, the second layer is clearly independent of the choice of the filter since all outputs are exactely the same. However, this is not the case in a more complex musical signal. Choosing different filter bins for the construction of the second layer can indeed amplify or soften the presence of certain components; transients that origin from percussion tend to live alone in the high frequency regions, whereas onset transients of played instruments can rather be found in the lower frequency regions.
The examples in Figure 3 show how the scattering transform captures tempo levels of different musical signals. The measured level depends on the most present transient patterns in the signal. Since only periodic patterns can be detected, irregular rhythmpatterns may only be captured on wider scales where they repeat. In that sense, the scattering transform is not a stand-alone "rhythm detector" as such, but however, it is remarkable how good and clear the tempo levels are depicted.

5 Conclusion

We presented an intuitive motivation of the scattering transform by considering it as deterministic analogue of a CNN. This made it possible to gain insights

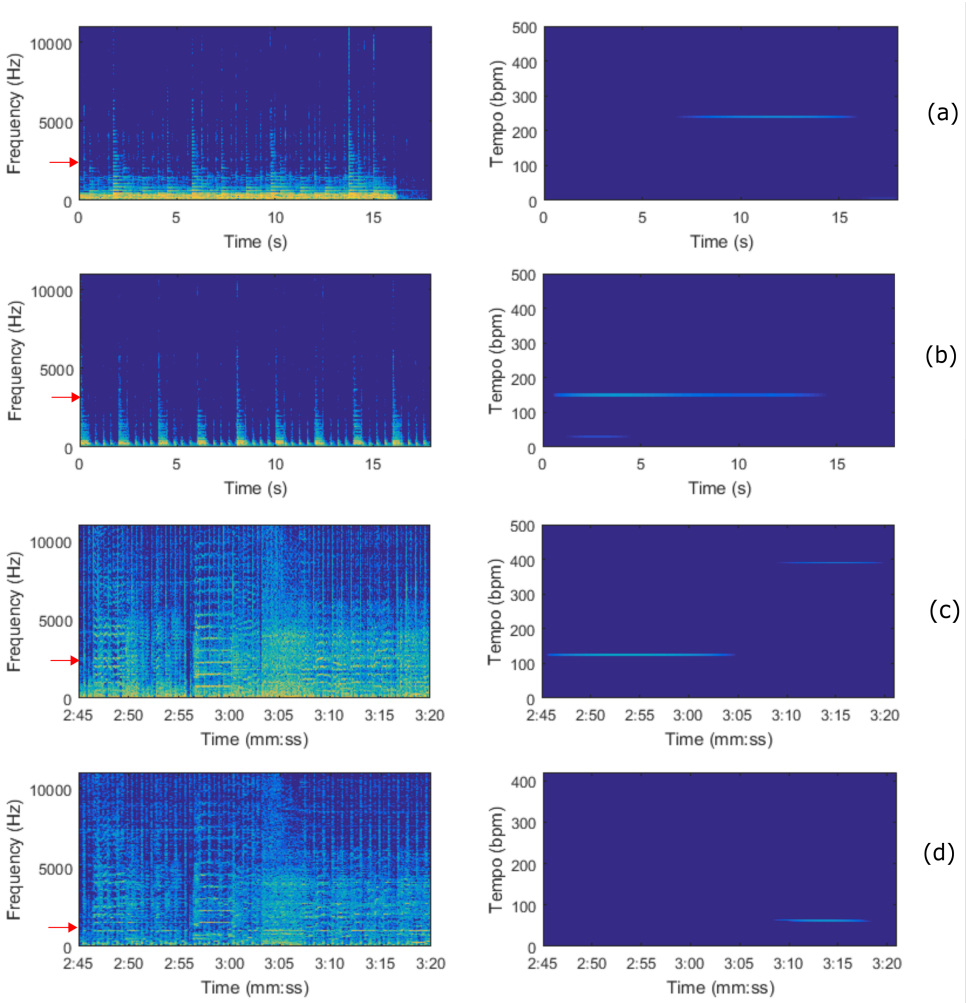

Fig. 3. Left (a)-(d): subsampled STFTs of the signals with modulus (first layer without time averaging) plotted in seconds and Hz. The red arrow indicates the filteroutputs that are used to compute the second layer. Right (a)-(d): STFTs of the chosen filter outputs with modulus and averaging in time (second layer), plotted in seconds (referring to the original signal) and bpm.
(a) uses a recording of a guitar, which plays a melody in fingerpicking style. The Tactus level of the melody is 120bpm, the melody itself is played rather monotonically and co5nsists of eighth notes i.e. the Tatum is 240bpm. This is depicted in the second layer.
(b) uses a recording of a guitar, playing a strokepattern of five with a strong accentuation on every first in 150bpm. Here both levels are captured, the Tatum level ref. to all strokes (150bpm) and the Tactus level ref. to the ones (30bpm).
(c),(d) use an excerpt of the song "Money" by Pink Floyd covering the transition from the saxophone solo (2:45-3:03) into David Gilmour's guitar solo (3:07-3:20). The song is approximately in 126bpm. In the saxophone solo, the hihat pattern consists of quarter notes over a 7/4 meter, changing to shuffled triplets over a 4/4 in the guitar solo. This indicates Tatum levels of 126, resp. 378bpm. The Tactus level is usually indicated by the kick-snare pattern. Here the transform struggles to detect the irregular pattern in the 7/4, but gets it in the 4/4 when it becomes regular.

into how a CNN learns audio-features on several temporal scales. Considering the tempo of transient patterns in terms of frequency made it particularly good understandable, how wide-term dependencies are embraced. In perspective, one could suspect the network to contain even wider spread features in its depth. Furthermore, the experiments on tempo extraction with different musical pieces emphasized that the scattering procedure provides interesting feature representations itself, depicting the temporal rhythm information in an intuitive and simple way. Hence, this makes it a very powerful tool for several signal processing applications and might also be used incorporated in rhythm-related learning tasks to improve learning performance.

In the end, we emphasize that the scattering transform is a very powerful tool because of its accessible simplicity and its intuitive structure, which extends the concept of time-frequency decompositions in a natural way, parallel to a CNN. The rigorous mathematical theory behind it reinforces its potential even more, both as deterministic analogue of a CNN, as well as a multi-scale feature extractor itself.

References

1. S. Mallat. Group invariant scattering. *Comm. Pure Appl. Math.*, vol. 65, no. 10, pp. 1331-1398, 2012.
2. J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann and X. Serra. End-To-End Learning for Music Audio Tagging at Scale. *Proc. of the 19th ISMIR Conference*, Paris, France, September 23-27, 2018.
3. H. Schreiber and M. Müller. A Single-Step Approach to Musical Tempo Estimation Using a Convolutional Neural Network. *Proc. of the 19th ISMIR Conference*, Paris, France, September 23-27, 2018.
4. I. Goodfellow, Y. Bengio and A. Courville. *Deep Learning*, MIT Press, `http://www.deeplearningbook.org`, 2016.
5. L. Wyse. Audio Spectogram Representations for Processing with Convolutional Neural Networks. *Proc. of the First International Workshop an Deep Learning and Music joint with IJCNN*, Anchorage, US. May, 2017. 1(1). pp 37-41.
6. T. Wiatowski and H. Bölcskei. Deep Neural Networks Based on Semi-Discrete Frames. *Proc. of IEEE International Symposium on Information Theory(ISIT)*, pp. 1212-1216, 2015.
7. J. Andén and S. Mallat. Deep Scattering Spectrum. *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4114-4128, 2014.
8. J. Andén and S. Mallat. Scattering Representation of Modulated Sounds. *Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-12)*, York, UK, September17-21, 2012.
9. A. Klapuri, M. Davy. *Signal Processing Methods for Music Transcription*, Springer, 2007.
10. R. Bammer and M. Dörfler. Invariance and Stability of Gabor Scattering for Music Signals. *Proc. of Sampling Theory and Applications (Sampta)*, 2017.