Binaural Scene Classification with Time-Frequency Scattering and Deep Convolutional Networks

Vincent Lostanlen
École normale supérieure
45 rue d'Ulm, 75005 Paris, France

***Abstract*—The abstract goes here.**

I. Introduction

Fine-grained classification of acoustic scenes is only made possible by integrating signal information over a long temporal context. Whereas a few seconds are often sufficient to recognize a speaker, a musical instrument, or a genre, it may require up to 30 seconds to disambiguate closely related acoustic scenes.

II. Time-frequency scattering

Let $\boldsymbol{\psi}[t]$ an analytic band-pass filter of dimensionless frequency 1 and bandwidth $1/Q$. A filter bank of wavelets is built by dilating $\boldsymbol{\psi}$ according to a geometric sequence of scales $2^{-k_1/Q}$, where the log-frequency index k_1 takes integer values. We denote by $\boldsymbol{\psi_{k_1}}[t]$ the resulting wavelets. In all subsequent experiments, $\boldsymbol{\psi}$ is designed as a Gammatone wavelet of quality factor $Q = 4$, so as to approximate the properties of the human cochlea. The wavelet transform of an audio signal $\boldsymbol{x}[t]$ is obtained by convolution with all wavelets: $\boldsymbol{y_1}[t,k_1] = (\boldsymbol{x} \overset{t}{*} \boldsymbol{\psi_{k_1}})[t]$. Applying pointwise complex modulus to $\boldsymbol{y_1}$ yields the wavelet scalogram $\boldsymbol{x_1}[t,k_1] = |\boldsymbol{y_1}[t,k_1]|$, also called constant-$Q$ transform (CQT), indexed by time t and log-frequency k_1.

Time-frequency scattering consists in convolving $\boldsymbol{x_2}[t,k_1]$ with a family of two-dimensional wavelets $\boldsymbol{\Psi_{k_2}}[t,k_1]$, where the index k_2 encapsulates two values, i.e. a temporal scale α and a log-frequential scale β. The temporal scale α takes 5 values between $23\,\mathrm{ms}$ and $370\,\mathrm{ms}$ according to a geometric sequence. The log-frequential scale β takes 5 values between $1/4^{\mathrm{th}}$ of an octave and 4 octaves, as well as their 5 "mirror" frequencies, according to a geometric sequence. In addition, the edge case $\beta = \infty$ corresponds to a log-frequential low-pass filter along 4 octaves according to all 5 temporal scales α. Finally, the edge case $(\alpha,\beta) = (\infty, 0)$ coresponds to a temporal moving average $\boldsymbol{\phi}[t]$ along $370\,\mathrm{ms}$ without any transformation of the log-frequency axis. In sum, there are $5 \times 5 \times 2 + 5 + 1 = 56$ time-frequency scattering features for every time t and log-frequency k_1.

The time-frequency scattering coefficients are thus defined as

$$\begin{aligned}\boldsymbol{y_2}[t,k_1,k_2] &= \left(\boldsymbol{x_1} \overset{t,k_1}{*} \boldsymbol{\Psi_{k_2}}\right)[t,k_1] \\ &= \sum_{\tau,\kappa_1} \boldsymbol{x_1}[t-\tau, k_1-\kappa_1]\,\boldsymbol{\Psi_{k_2}}[\tau,\kappa_1]. \end{aligned} \tag{1}$$

For some index k_2 involving a temporal scale α, applying pointwise complex modulus to $\boldsymbol{y_2}[t,k_1,k_2]$ provides translation-invariant coefficients as long as the amount of translation does not exceed α. To increase the amount of invariance to translation, we apply the low-pass filter $\boldsymbol{\phi}[t]$ to the moduli, hence bringing all coefficients to the same sample rate.

$$\boldsymbol{x_2}[t,k_1,k_2] = |\boldsymbol{y_2}[t,k_1,k_2]| \overset{t}{*} \boldsymbol{\phi}[t] \tag{2}$$

III. Deep Convolutional Networks

Each layer in a convolutional network typically consists in the composition of three operations: two-dimensional convolutions, application of a pointwise nonlinearity, and local pooling.

$$\begin{aligned}&\boldsymbol{y_3}[t,k_1,k_3] \\ &= \sum_{k_2} \boldsymbol{b_3}[k_2,k_3] + \mathbf{W_3}[t,k_1,k_2,k_3] \overset{t,k_1}{*} \boldsymbol{x_2}[t,k_1,k_2]. \end{aligned} \tag{3}$$

We apply the rectified linear unit (ReLU) nonlinearity, with a rectifying slope of $\nu = 0.3$ for negative inputs.

$$\boldsymbol{y_3^+}[t,k_1,k_3] = \begin{cases} \nu\, \boldsymbol{y_3}[t,k_1,k_3] & \text{if } \boldsymbol{y_3}[t,k_1,k_3] < 0 \\ \boldsymbol{y_3}[t,k_1,k_3] & \text{if } \boldsymbol{y_3}[t,k_1,k_3] \geq 0 \end{cases} \tag{4}$$

At the pooling step, we retain the maximal activation among neighboring units in the time-frequency domain (t,k_1) over non-overlapping rectangles of width Δt and height Δk_1.

$$\boldsymbol{x_3}[t,k_1,k_3] = \max_{\substack{0 \leq \tau < \Delta t \\ 0 \leq \kappa_1 < \Delta k_1}} \left\{ \boldsymbol{y_3^+}[t-\tau, k_1-\kappa_1, k_3] \right\} \tag{5}$$

The hidden units in $\boldsymbol{x_3}$ are in turn fed to a second layer of convolutions, ReLU, and pooling.

Data augmentation is a simple, yet effective, way to enforce invariance in feature learning is to

$$\boldsymbol{x}[t] = r \times \boldsymbol{x}^{\mathsf{L}}[t] + (1-r) \times \boldsymbol{x}^{\mathsf{R}}[t], \tag{6}$$

where r is drawn uniformly at random in the interval $[0,1]$.

This work is supported by the ERC InvariantClass grant 320959. The source code to reproduce figures and experiments is freely available at http://www.github.com/lostanlen/dcase2016.

IV. Conclusion

The conclusion goes here.