Spline Filters For End-to-End Deep Learning

Randall Balestriero [*1] **Romain Cosentino** [*1] **Hervé Glotin** [2] **Richard Baraniuk** [1]

Abstract

We propose to tackle the problem of end-to-end learning for raw waveforms signals by introducing learnable continuous time-frequency atoms. The derivation of these filters is achieved by first, defining a functional space with a given smoothness order and boundary conditions. From this space, we derive the parametric analytical filters. Their differentiability property allows gradient-based optimization. As such, one can equip any Deep Neural Networks (DNNs) with these filters. This enables us to tackle in a front-end fashion a large scale bird detection task based on the freefield1010 dataset known to contain key challenges, such as high dimensional inputs (> 100000) and the presence of multiple sources and soundscapes.

1. Introduction

Most current learning tasks take the form of pattern recognition. This includes cases dealing with speech, bioacoustic, healthcare, economic time series containing different kinds of nuisances (colored noises, multiple sources, measurements errors, ...). Recently, deep neural networks (DNNs) have offered an end-to-end learnable pipeline (from raw input data to the final prediction) particularly suited for pattern recognition. In particular, convolutional-based DNNs are state-of-the-art in computer vision and other areas (LeCun et al., 2015; He et al., 2016; Leung et al., 2014). This approach reduces the need of hand-crafted pre-processing steps which involves expert knowledge and a tedious search over the set of all possible features. Such paradigm shift opens the door to novel algorithms that encapsulate the learning of both, the featurizing and the decision.

While providing a fully automated approach, DNNs performances depend on the number of perturbations such as noise and inherent nuisances contained in the dataset. This is mainly due to the use of greedy optimization schemes applied on a very high-dimensional parametric model as well as the lack of explicit perturbation modeling in DNNs (Cohen & Welling, 2016). This effect is amplified with the dimensionality of the input and the dimensionality of the filters. It is thus particularly detrimental for time-series, specially for bio-accoustic signals. In fact, those signals are sampled at high-frequency rate (ranging from 44kHz to 500kHz), for long recording duration and are paved with non-stationary nuisances including different sources, soundscapes, and inherent hardware noise (Ramli & Jaafar, 2016; Glotin et al., 2017; Trone et al., 2015). In addition, features of interest can lie at many different frequencies and in small time window adding complexity to the learning task.

Therefore, current solutions to tackle bio-acoustic signal still rely on hand-crafted features providing representations onto which DNNs are applied. Considered representations are often based on time-frequency framework as they stretch and reveal crucial information lying in the time-amplitude domain (Jaffard et al., 2001). Moreover, decomposing signals in the time-frequency plane leverage the capability of Convolutional Neural Networks (CNN) to capture meaningful features in this "image"(Krizhevsky et al., 2012). Withal, the design and selection of the filter enabling the time-frequency representation of the signal is directed by the prior knowledge on the feature of interest. For instance, in the case of wavelet transform, one selects the most suitable wavelet family (i.e: Seismic data: Morlet wavelet, Speech: Gammatone wavelet,...(Lostanlen, 2017; Serizel et al., 2018)). Since the capability of generalization of handcrafted features is only proportional to the amount of data witnessed by the designer, in (Cosentino et al., 2016; Megahed et al., 2008), they developed algorithms automatizing the search for the optimal filter. However, these pre-processing techniques were derived for goals not necessarily aligned with the current tasks at hand (reconstruction, compression, classification) and thus do not provide a universal solution. In this work, we propose to alleviate the limitation of DNNs by proposing a universal learnable time-frequency representation that can be trained with respect to the application at hand.

[*]Equal contribution [1]ECE Department, Rice University, Houston, TX [2]Univ. Toulon, AMU, CNRS, LIS, DYNI, Marseille, France. Correspondence to: Romain Cosentino <rom.cosentino@gmail.com>, Randall Balestriero <randallbalestriero@gmail.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

Related Work: To provide flexible time-frequency representations and avoid the selection of hand-crafted filters, (Cakir et al., 2016) proposed to learn the Mel-scale filters leading to Mel-Frequency Spectral Coefficients (MFSC). It boils down to learning the linear combination of the spectrogram frequency filters instead of using triangular windows. In this case, the underlying representation still relies on Fourier basis and thus inherits the problem of a pre-imposed basis. On the other hand, (Zeghidour et al., 2017) proposed the use of a complex 1D Convolutional layer followed by complex modulus and local averaging. This was motivated by stating that a Gabor scalogram followed by complex modulus and local averaging approximates MFSC coefficients (Andén & Mallat, 2014). Finally, (Sainath et al., 2015; Dai et al., 2017; Trigeorgis et al., 2016) apply directly DNNs on the raw waveforms and demonstrated that, with careful model design, one could reach results on parity with MFSC. Yet, the previously described work was applied onto datasets that are obtained from controlled experiments containing negligible noise and low-frequency sampling (leading to small length signals). As such, their results do not reflect the reliability and robustness of their methods for general real world-tasks.

Our Contributions: Our solution learns the optimal time-frequency representation for the task and data at hand. This is done via learning time-frequency atoms with respect to the loss function (which can be of reconstruction, compression, anomaly detection, classification,...). The expression of these atoms corresponds to continuous filters analytically derived via spline functions. The filters can be constrained to inherit some pre-imposed properties such as smoothness and boundary conditions. Since the unique analytical expression of the filters is differentiable with respect to their parameters, they can be optimized via first-order derivative methods such as gradient descent. As such, they can be cast in a DNN layer and learned via backpropagation. In summary, our contributions are as follows:

1. Leverage spline interpolation methods to provide explicit expression of learnable continuous filters, in Sec. 2.

2. Derivation of learnable time-frequency representations removing the need for a priori knowledge, in Sec. 2.

3. Provide a novel robust and interpretable Convolutional Neural Network embedding, in Sec. 3.

4. Application of the spline filters in challenging bird detection task, in Sec. 3.

2. Continuous Filter Learning via Subspace Restriction

In this work, we propose to build continuous filters that can be extended to render time-frequency representation and specifically constant-Q transform (Brown, 1991). This transformation render the signal into a time-frequency plane where the frequency resolution decreases as the frequency increases. This transformation is directly related to the mapping performed by the human cochlea (Shera et al., 2002). Our approach is general enough to produce any continuous filter as soon as a functional space to which they belong exist. For sake of clarity, we will present the development of smooth locally supported oscillating filters, namely wavelet filters. As such, we provide the theoretical building blocks enabling one to build its own continuous filters depending on the application at hand.

2.1. Overall Approach: Deriving Filter Analytical Formula from Functional Spaces

As we will show for the specific case of wavelet filters, our method is based on the definition of a functional space highlighting the properties of the wished filters. Given the latter, we will first perform its discretization in the same manner as finite element methods for the variational problem of partial differential equations (Clough, 1990). We build a discretization of the functional space such that as the number of knots grows, any continuous filters from the original functional space can be approximated arbitrarily closely. The filters are based on the linear combination of atoms of basis elements of the discrete space, Hermite cubic splines in our case. Those atoms, being parametrized by the coefficients of the linear combination of the basis functions, they can easily be optimized. It results in a filter that approximates a particular function in the infinite dimensional space. This filter, learned with respect to the data and the task at hand, will describe a physical process underlying the signal while holding the properties of the functional space that it approximates. We thus create a framework enabling one to have theoretical guarantees based on the original functional space while being data and task driven.

2.2. Wavelets

Wavelets are square integrable localized wave functions (Mallat, 1999). Their ability to extract subtle patterns within non-stationary signals is inherited from their compact support (Xu et al., 2016). In fact, wavelets are known to provide a robust time-frequency representation for non-stationary signals as it is localized both in time and frequency, and close to optimal from an uncertainty principle perspective with constant bandwidth to center frequency ratio (Meyer, 1993). In fact, the higher the frequency is, the higher the wavelet is precise in time. Per contra, for low-frequency

contents, wavelets are highly localized in frequency but wide in time. Besides, given the nature of the time-series (e.g.: non-stationary biological time-series), this embedding will encode the signal with only a few activated wavelet atoms resulting in a sparse representation (Cosentino et al., 2017). While we will leverage spline interpolation techniques to sample the filters from the functional space, our approach is independent of the spline wavelets setting. As a matter of fact, spline wavelets, well developed by (Unser, 1997b) are constructed upon multiresolution analysis. These wavelets have an explicit expression in both the time and frequency domain hence facilitating their computation. Besides, they span a wide range of filter's smoothness order (Unser, 1997a). Despite the detachment between our framework and the one of spline wavelet, we can make an analogy between them. The ability of spline wavelet to provide an analytical formula for discrete wavelets is analogous to our proposal to provide the analytical continuous formula for the discrete filter-banks of convolutional networks.

2.3. Wavelet Ambient Space Definition

In our case, we provide a theoretical framework enabling one to build through a data-driven process a continuous filter-bank spanning wavelet filters. Let define the space of wavelets be

$$\mathcal{V}_{L_c^2} = \left\{ \psi \in L_c^2(\mathbb{R}), \int \psi(t)dt = 0 \right\}, \tag{1}$$

where $L_c^2(\mathbb{R})$ defines the space of square integrable functions with compact support.

2.4. Discretization of the Ambient Space

We direct the reader to a complete review of spline operators in (Schoenberg, 1964). In order to control the smoothness of the wavelets and thus of the sampled filters, we propose to restrict our study to the space of zero-mean functions with compact support belonging to $C_c^n(\mathbb{R})$

$$\mathcal{V}_{C_c^n} = \left\{ \psi \in C_c^n(\mathbb{R}), \int \psi(t)dt = 0 \right\}. \tag{2}$$

Since continuous and differentiable functions with compact support are square integrable, and a fortiori they belong to L_c^∞, it is clear that $\mathcal{V}_{C_c^n} \subset \mathcal{V}_{L_c^2}$. Therefore, $\mathcal{V}_{C_c^n}$ is a space of function with compact support where the smoothness is described by the order n. In this work, we will restrain our study to the space $\mathcal{V}_{C_c^1}$ which will provide an efficient trade-off between smoothness characterization and tractability. In order to build the discrete space denoted by V, we first proceed with the partition of the support of the function, denoted by the segment $[a, b]$, in $N+1$ intervals of length $h = \frac{b-a}{N+1}$, we thus defined as $t_i = a + ih, \forall i \in \{0, ..., N+1\}$ the $N+2$ points on the mesh, where in particular $t_0 = a$ and $t_{N+1} = b$. We define the discretization of the functional space $\mathcal{V}_{C_c^1}$ as

$$V = \left\{ \psi_h \in \bar{V}, \int \psi_h(t)dt = 0 \right\} \tag{3}$$

where

$$\bar{V} = \left\{ \psi_h \in S_{C_c^1}, \psi_h(a) = \psi_h(b) = \frac{\mathrm{d}\psi_h}{\mathrm{d}t}(a) = \frac{\mathrm{d}\psi_h}{\mathrm{d}t}(b) = 0 \right\} \tag{4}$$

and

$$S_{C_c^1} = \left\{ \psi_h \in C_c^1([a, b]), \psi_{h_{|[t_i, t_{i+1}]}} \in \mathcal{P}^3, i = 1, \ldots, N \right\}, \tag{5}$$

where $\mathcal{P}^3$ defines the space of order 3 polynomials and $S_{C_c^1}$ the space of cubic and smooth splines.

2.5. Analytical Filter Formula via Spline interpolation

We now derive a basis of the space $\bar{V}$ such that we can provide explicit formulation of the functions belonging to such space.

Lemma 1. *Any function in $S_{C_c^1}$ is entirely and uniquely defined by its values and its first order derivative values on each point of the mesh $t_i, \forall i \in \{0, ..., N+1\}$.*

Proof. Let $\psi_h \in S_{C_c^1}$, without loss of generality we focus on $\psi_{h_{|[t_i, t_{i+1}]}}$. It is clear that given the fact that it is a polynomial of degree 3 on the interval $[t_i, t_{i+1}]$ it can be expressed as

$$\psi_{h_{|[t_i, t_{i+1}]}} = a(t - t_i)^3 + b(t - t_i)^2 + c(t - t_i) + d. \tag{6}$$

Let show that the coefficients a, b, c, d of the polynom are uniquely determined by $\theta_{t_i}, \theta_{t_{i+1}}, \theta'_{t_i}, \theta'_{t_{i+1}}$. Naturally, $d = \theta_{t_i}$ and $\theta'_{t_i} = c$, then, the coefficient a, b are defined by the solution of the following problem

$$\begin{pmatrix} h^3 & h^2 \\ 3h^2 & 2h \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \theta_{t_{i+1}} - \theta'_{t_i} h - \theta_{t_i} \\ \theta'_{t_{i+1}} - \theta'_{t_i} \end{pmatrix}, \tag{7}$$

since $\det \begin{pmatrix} h^3 & h^2 \\ 3h^2 & 2h \end{pmatrix} = -h^4$, the system has a unique solution. □

Theorem 1. *Let define $u^{(i)}$ and $v^{(i)}$ as functions belonging to $S_{C_c^1}$ such as $\forall i \in \{0, ..., N+1\}$*

$$u^{(i)}(t_j) = \delta_{ij},\ {u^{(i)}}'(t_j) = 0, \tag{8}$$

$$v^{(i)}(t_j) = 0,\ {v^{(i)}}'(t_j) = \delta_{ij}. \tag{9}$$

These functions form a basis of $S_{C_c^1}$, and for all $\psi_h \in S_{C_c^1}$, we have,

$$\psi_h = \sum_{i=0}^{N+1} (\theta_{t_i} u^{(i)} + \theta'_{t_i} v^{(i)}). \tag{10}$$

Proof. We first show that the space $S_{C_c^1}$ is spanned by such functions. Let ψ_h any function belonging to $S_{C_c^1}$, et let z defined such as

$$z = \sum_{i=0}^{N+1} (\theta_{t_i} u^{(i)} + \theta'_{t_i} v^{(i)}), \tag{11}$$

it is clear that z belongs to $S_{C_c^1}$ as a linear combination of functions belonging to $S_{C_c^1}$. Then, for all $j \in \{0, ...N+1\}$, we have $z(t_j) = \theta_{t_j}$ and $\frac{\mathrm{d}z}{\mathrm{d}t}(t_j) = \theta'_{t_j}$. Thus z coincides with the function ψ_h on all the points of the mesh. From Lemma 1, we know that $z = \psi_h$, thus $u^{(i)}$ and $v^{(i)}$ span the space $S_{C_c^1}$. Let's now prove that this family is linearly independent. Let's assume $\psi_h = \sum_{i=0}^{N+1} (\lambda_i u^{(i)} + \mu_i v^{(i)}) = 0$, where λ_i, μ_i are scalar coefficients. Then, for all $j \in \{0, ...N+1\}$ we have $\theta_{t_j} = \lambda_j = 0$ and $\theta'_{t_j} = \mu_j = 0$. □

Notice that the parameters $\theta_{t_i}, \theta'_{t_i}$, correspond respectively to the value of the function ψ_h and the derivative of the function ψ_h at the knot t_i.

Corollary 1. *The dimension of the space $S_{C_c^1}$ is $2(N+2)$.*

The proof is immediate given that its basis forms a $2(N+2)$ functions as defined in the previous theorem. We have built a basis for the space $S_{C_c^1}$, it is simple to analyze the basis of its subspaces, namely $\bar{V}$ and V, where we have $V \subset \bar{V} \subset S_{C_c^1}$. From the space $S_{C_c^1}$ to $\bar{V}$ we add Dirichlet and Neumann boundary conditions. These conditions imply directly that any function in $\bar{V}$ is $C^1(\mathbb{R})$ as the function in $S_{C_c^1}$ has a compact support, it is null out of its support, then imposing that both the derivative and the value on the boundary of the support is zeros implies the continuity and differentiability on $\mathbb{R}$.

Corollary 2. *The dimension of the space $\bar{V}$ is $2N$.*

Proof. Imposing the boundaries conditions remove 4 degrees of freedom from the space $S_{C_c^1}$ as we only consider the internal part of the mesh. □

Then, $\forall \psi_h \in \bar{V}$, we have

$$\psi_h = \sum_{i=1}^{N} \theta_{t_i} u^{(i)} + \sum_{i=1}^{N} \theta'_{t_i} v^{(i)}. \tag{12}$$

One can easily explicitly derived this basis via the following reference functions

$$u_0(t) = (1+2t)(1-t)^2, u_1(t) = (2-2t)t^2, \tag{13}$$
$$v_0(t) = t(1-t)^2, v_1(t) = -(1-t)t^2, \tag{14}$$

then $\forall i \in \{1, ..., N\}$ we have the following functions defined on their supports

$$u^{(i)}(t) = u_0(\frac{t-t_{i-1}}{h}),\ \forall t \in [t_{i-1}, t_i] \tag{15}$$
$$= u_1(\frac{t-t_i}{h}),\ \ \forall t \in [t_i, t_{i+1}], \tag{16}$$

and

$$v^{(i)}(t) = v_0(\frac{t-t_{i-1}}{h})h,\ \forall t \in [t_{i-1}, t_i] \tag{17}$$
$$= v_1(\frac{t-t_i}{h})h,\ \ \forall t \in [t_i, t_{i+1}]. \tag{18}$$

Finally, from $\bar{V}$ to V, we require that the integral of the polynomial is null over the whole domain, which implies the following corollary

Corollary 3.

$$V = \left\{ \psi_h \in \bar{V}, \exists j,\ \theta_{t_j} = -\sum_{i \neq j} \theta_{t_i} \right\}, \tag{19}$$

and the dimension of V is $2N-1$.

Proof. While integrating $\psi_h \in \bar{V}$ and using Chasles' relation to split the integral over the mesh's segments, the C^1 property implies that the coefficients θ'_{t_i} cancel each other. Then the equality of the integral to zeros is equivalent to the condition following condition $\exists j \in \{1, ..., N\}, \theta_{t_j} = -\sum_{i \neq j} \theta_{t_i}$, which proves the first part of the corollary. The dimension of the space is the dimension of $\bar{V}$ minus one degree of freedom, which completes the proof. □

Furthermore, the error of the approximation involved by the discretization of the space by mean of cubic Hermite splines is of the order $\mathcal{O}(h^4)$(Hall & Meyer, 1976). As a matter of fact, the smaller the segment of the mesh is, the closer the approximant will be to the associated function in the functional space.

2.6. From Primitive Filter to Overcomplete Dictionary

Another advantage of analytical filters resides in the possibility to apply standard continuous operators such as time-dilation and frequency-shift. Applying such operators to the primitive filter yields the creation of the filter-bank. From Lemma 1, it is clear that the set of parameters $\theta = \{(\theta_{t_i}, \theta'_{t_i}), \forall i \in \{1, ..., N\}\}$ defines uniquely the spline filter. We now denote our discretized filter ψ_h by ψ_θ. For our experiments, we will consider the use of our filter formulation to derive a filter-bank. This is done by only learning a mother filter which is then dilated to build the collection of filters. Hence they all rely upon the same analytical form but are dilated versions of each other. Let's suppose we have a mother wavelet, $\psi_\theta \in \mathcal{V}_{L_c^2}$, we propose an operation, a dilation, that will provide the analytic expression of our redundant frame.

Dilation Operator Let D_λ, a dilation operator defined by

$$\mathcal{D}_\lambda[\psi_\theta](t) := \frac{1}{\sqrt{\lambda}}\psi_\theta\Big(\frac{t}{\lambda}\Big). \tag{20}$$

The scale parameter $\lambda \in \mathbb{R}^+$ allows for time dilation and frequency-shift and follows a geometric progression for the case of wavelets. It is defined as $\lambda_i = 2^{\frac{i-1}{Q}}, i = 1, \dots, JQ$ where $J \in \mathbb{N}, Q \in \mathbb{N}$ define respectively the number of octave and the number of wavelets per octave. Taking $Q > 1$ yields a redundant frame, which can be more powerful for representation analysis (Olshausen & Field, 1996). We now denote this collection of scales as $\Lambda := \{\lambda_i, i = 1, \dots, JQ\}$. Note that, in this work, this parameter will not be learned but will be specified given a priori knowledge on the data.

2.7. Gradient based Learning Rule

Note that since our filters can be used as part of a DNN or as a stand-alone for representation learning, we remind bellow the generic gradient-based learning rule leveraging the chain rule. In order to learn the collection of filters, since we know that the filters are entirely and uniquely defined by their parameters θ, we propose to learn the internal parameters θ via iterative first order optimization method such as gradient descent. Therefore, given a differentiable loss function $\mathcal{L}$ for the task at hand, such as classification, regression, detection, one can learn the filters that will produce the representation that is the most suitable. We have via the chain rule

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{\lambda,t} \frac{\partial \mathcal{L}}{\partial \mathcal{W}_{\psi_\theta}[x](\lambda,t)} \frac{\partial \mathcal{W}_{\psi_\theta}[x](\lambda,t)}{\partial \theta}, \tag{21}$$

where $\mathcal{W}_{\psi_\theta}[x](\lambda, t)$ defines the wavelet transform

$$\mathcal{W}_{\psi_\theta}[x](\lambda,t) = (x \star \mathcal{D}_\lambda[\psi_\theta])(t), \forall \lambda \in \Lambda. \tag{22}$$

2.8. Implementation

In order to implement such filters, we leverage the Hermite cubic spline interpolation formula (12) between each of the knots of a specified domain to obtain the sampled filter's chunk per region (between two knots). This takes the following form for a given filters

$$\begin{aligned}\psi_i(t) = (2t^3 - 3t^2 + 1)\theta_{t_i} + (t^3 - 2t^2 + t)\theta'_{t_i} \\ + (-2t^3 + 3t^2)\theta_{t_{i+1}} + (t^3 - t^2)\theta'_{t_{i+1}}\end{aligned} \tag{23}$$

$$\psi_\theta(t) = \sum_{i=0}^{N} \psi_i\left(\frac{t - t_i}{t_{i+1} - t_i}\right) 1_{\{t \in [t_i, t_{i+1}]\}}. \tag{24}$$

Then, one derives the filter bank by using the above equation with different time sampling according to the dilation from Λ. For each scale λ_i the time sample is refined as $t = \{t_0, t_0 + \frac{h}{\lambda_i}, \dots, t_N\}$. These latter are then concatenated to produce the filter bank. This process can be done independently for the real and the imaginary part as the real and imaginary coefficients are independent. For the time-dilation operation, it suffices to repeat this process with a finer or larger sampling grid on which the Hermite cubic spline interpolation occurs. The code is provided as an open-source implementation in Theano [1]. This code is embedded as a special Convolutional layer class of the Lasagne library for ease of use for any interested parties looking to integrate this Spline Convolutional layer as part of their DNN pipeline.

3. Validation with a Bird Detection Problem

In order to validate the proposed method in a supervised setting problem, we provide experiments on a large scale bird detection task. The data set is composed of 7000 field recording signals of 10 sec. sampled at 44kHz from the Freesound (Stowell & Plumbley, 2013) audio archive representing slightly less than 20 hours of audio signals. The audio waveforms are extracted from diverse scenes such as city, nature, people, train, voice, water..., some of which include bird songs. In this paper, we will focus on the supervised bird detection task consisting of assigning the label 1 if the sound contains a bird song and 0 otherwise. The labels regarding the bird detection task can be found in freefield1010[2]. Due to the unbalanced distribution of the classes (3 for 1), the evaluation is achieved via the Area Under Curve (AUC) metric on a test set consisting of 33% of the complete dataset.

3.1. Architecture Comparison

To compare our method we propose different training settings. For all the trained methods, the signals are subsampled by 2, leading to a sampling rate of $\approx$ 22kHz. The learning was achieved for 120 epochs with batch size equals to 10. The learning rate for each method has been cross-validated with respect to the following learning rate grid: $[0.0001, 0.005, 0.01, 0.05]$. We did not perform data augmentation. We provide average and standard deviation for the AUC evaluation score over 10 independent runs. For each run, all the topologies are trained and tested on the same training and testing set leading to a comparison of the different algorithms on the same data. The different methods we will apply correspond to variants of the state-of-the-art method proposed in (Grill & Schlüter, 2017). The difference will lie in the first layer of the topology which corresponds to either an MFSC transform, an unconstrained complex $1D$ convolutional layer and finally the complex

[1] https://github.com/RandallBalestriero/SplineWavelet

[2] http://machine-listening.eecs.qmul.ac.uk/bird-audio-detection-challenge/

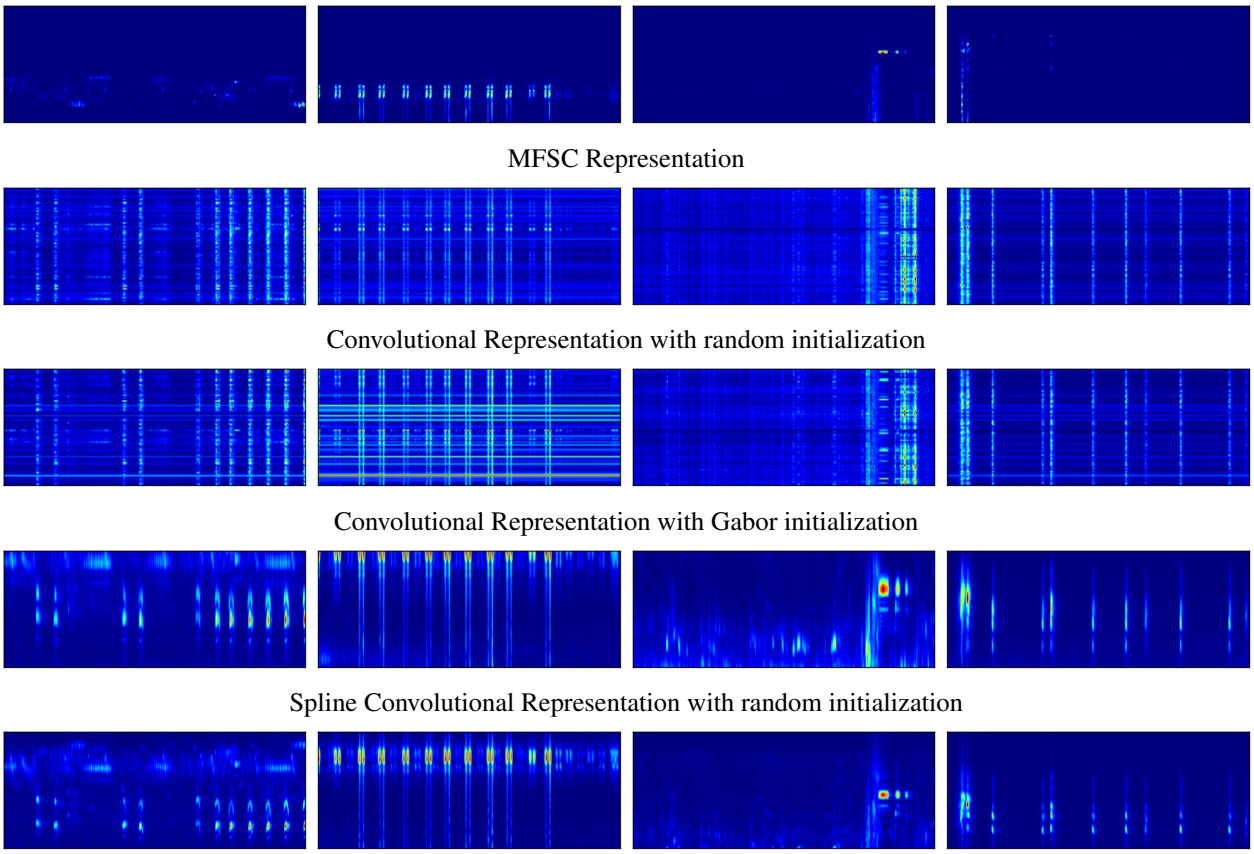

Figure 1. First layer representations (*time-frequency plane*). Signals of class 1 (a bird is present). Each column depicts a different signal. Firstly, the amount of sparsity (the $L1$-norm of the representation) often considered as a quality criterion can be seen to be conserved with the spline Convolutional. In addition, the events are well localized in frequency as opposed to the convolutional representations depicting events covering the whole axis and/or time dimension. The detected events seem to be in accordance with all representations.

spline filters cast into the complex $1D$ convolutional layer. For all cases, the number and sizes of the filters are identical. Everything else in the DN is kept identical between the methods. Also, both the Spline Convolutional layer and the Convolutional Layer were tested with two filter initialization settings: random and Gabor. Finally, due to the induced extra representation to store on GPU (namely $\mathcal{W}_{\psi_\theta}[x](\lambda, t)$) prior applying the mean-pooling, the required memory for the Spline Convolutional and Convolutional topologies is higher than the baseline which computes the MFSC on CPU a priori. As a result, the mean-pooling applied to these cases is chosen twice bigger for those topologies as opposed to the MFSC baseline, leading to a first layer representation twice smaller. We briefly describe the different methods and choice of parameters.

State-of-the-art method MFSC + ConvNet: The baseline and state-of-the-art method (Grill & Schlüter, 2017) is based on MFSC: spectrogram with window size of 1024 and 30% overlap, then mapped to the mel-scale by mean of 80 triangular filters from 50Hz to 11kHz. The MFSC are computed by applying a logarithm. This time-frequency representation is then fed to the following network: Conv2D. layer (16 filters 3×3), Pooling (3×3), Conv2D. layer (16 filters 3×3), Pooling (3×3), Conv2D. layer (16 filters 3×1), Pooling (3×1), Conv2D. layer (16 filters 3×1), Pooling (3×1), Dense layer (256), Dense layer (32), Dense layer (1 sigmoid). At each layer a leaky ReLU is applied following a batch-normalization. For the last three layers a 50% dropout is applied.

ConvNet: In this method, we keep the architecture of state-of-the-art solution, while replacing the deterministic MFSC by a regular complex Convolutional neural network layer, followed by a complex modulus, a logarithm and an average pooling, providing as stated in (Zeghidour et al., 2017) a learnable MFSC representation. The number of complex filters for the first layer is 80 leading to a representation at the first layer equivalent to the MFSC. We propose two initialization settings for the first layer discrete filters: random and

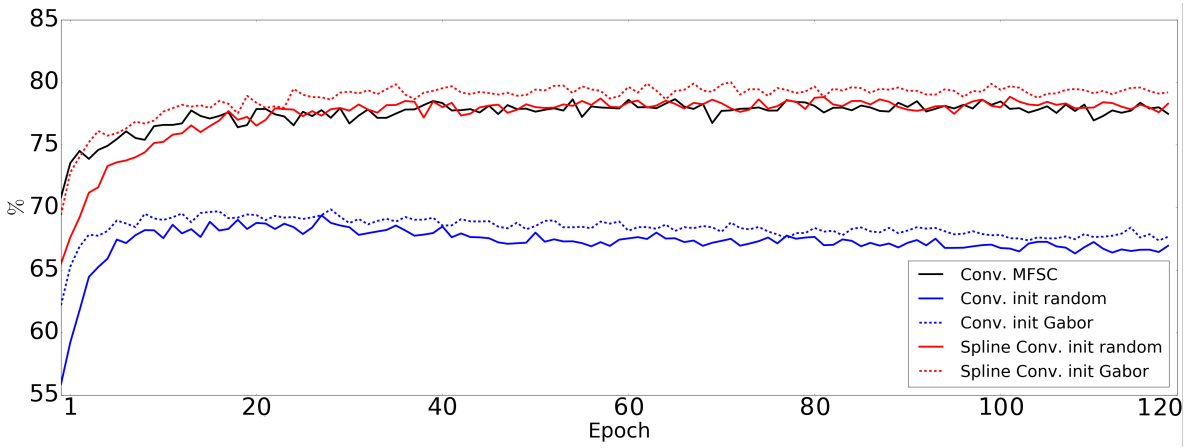

Figure 2. Final Results on FreeField data set. Initializing the CNN filters with a Gabor filter-bank leads to increased performances as opposed to random initialization. Yet, the final performances remain around 10 percentage point below the other methods. The spline-based Convolutional layer with random initialization is able to reach similar performances with the MFSC features after only 20 epochs. Finally, the Gabor initialized spline filter-bank starts on pair with the MFSC features as can be seen for the first couple of epochs and is then able to overcome the MFSC feature to rapidly obtain about 2 point of percentage increased performances. Hence we can see the MFSC representation to be a satisfactory initializer yet not optimal.

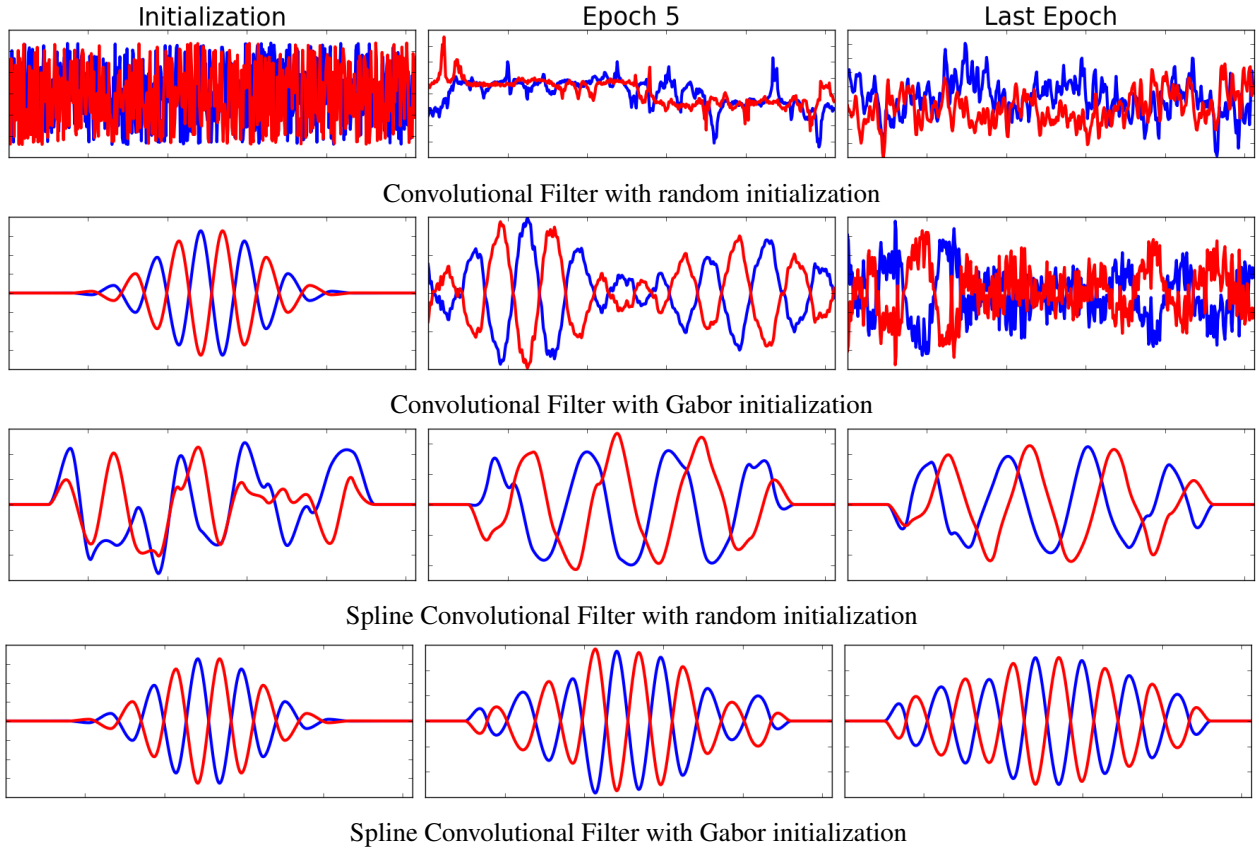

Figure 3. Filters extracted from the Convolutional Layer and Spline Convolutional Layer. The red and blue lines correspond to the complex and real part respectively. Filters are presented in the left, middle, and right column respectively corresponding to the initialization, during learning, and after learning. As can be witnessed in the third row, even with random initialization, the smoothness and boundary conditions are able to prevent too erratic filters. Our Spline configuration initialized Gabor (the bottom row) through learning tends to a modified Gabor. In fact, while a Gabor is roughly a complex sine localized via a Gaussian window, the learned filter seems closer to a complex sine localized with a Welch window(Harris, 1978). For the Discrete Convolutional filters, even with Gabor initialization (second row), the nuisances (noise, and other nonstationary class independent perturbations) are absorbed during learning even at early stages (middle column).

Table 1. Classification Results - Bird Detection - Area Under Curve metric (AUC)

Model (learning rate)	*AUC* (mean±std)
Conv. MFSC (0.01)	77.83 ± 1.34
Conv. init. random (0.01)	66.77 ± 1.04
Conv. init. Gabor (0.01)	67.67 ± 0.98
Spline Conv. init. random (0.005)	78.17 ± 1.48
Spline Conv. init. Gabor (0.01)	**79.32** ± 1.53

Gabor. The complex convolution is simply implemented as a two channel convolution corresponding to the real and imaginary part.

Spline Continuous Filter ConvNet: As for the Conv. Net model, we keep the same architecture but replace the first layer with the proposed method. In particular, the first layer is a complex Convolutional Layer with filters computed via our method. Given the dataset context, we naturally impose as functional space for the filters the wavelet space. We use 80 filters based on the dilation operator developed in 2.6 with $J = 5$, $Q = 16$. This layer is followed by a complex modulus, logarithm and an average pooling. We propose two initializations as for the previous method: random, and Gabor. For each filter, the number of cubic Hermite polynomials respective the boundary condition is 15 as 16 knots are used. Since the set of filters are derived by dilation of one mother filter, the number of parameters for this layer is only of 56 (14×4).

Speed of Computation and Number of Parameters: The number of parameters for the spline Convolutional DNN is of 145073. The computation time for one batch made of 10 examples is 0.44 ± 0.009 sec. For the Convolutional DNN, the number of parameters is 227089 and the computation time for one batch is of 0.42 ± 0.01 sec. In fact, given our current implementation, the Spline Convolutional layer first has to interpolate and generate the filter-bank based on the parameters of the Hermite cubic spline and this filter-bank (for real and complex parts) is then used in a Convolutional layer. This extra computation time of interpolation and filter-bank derivation thus takes an additional 0.02 sec. per batch on average. Finally, for the state-of-the-art method, the number of parameters is 374385 and the computation time for one batch is 0.01 ± 0.0004 sec. This comes from the input being directly the MFSC representation as opposed to the raw waveform. The increased number of degrees of freedom comes from having a time-frequency representation longer in time as opposed to the other two topologies having larger time-pooling for memory constraints.

3.2. Results

Table 1 displays the average over the last 20 epochs of the 10 runs for each methods as shown in 2. We see that using classical discrete filters on the raw waveforms fails to generalize and is seen to overfit starting at epoch 50. However, performing MFSC representation drastically increases the accuracy. Finally, we see that our approach is capable of performing equivalent results than the state-of-the-art in the case of random initialization and has an increase of ≈ 2 points of percentage when initialized with Gabor filters.

4. Conclusions and Future Work

In this work, we proposed a novel way to tackle end-to-end deep learning for waveform analysis. To do so, we proposed to highlight the need for designing new learnable filters that can be learned with any differentiable loss function and architecture. This approach showed its potential and robustness on a challenging audio scene dataset reaching significantly better results as opposed to using pre-imposed MFSC representation or unconstrained DNs. For future work, one can extend the filter learning to jointly learn the dilation operator. In fact, as we have shown in 2.6 this operator leverages the parameter pre-imposed parameters λ to follow a geometric progression. We can instead learn it as it is differentiable. This would include the case of learning the correct geometric progression but also learning arbitrary dilation with different types of relationships between themselves.

Acknowledgements

Richard Baraniuk and Randall Balestriero were supported by DOD Vannevar Bush Faculty Fellowship grant N00014-18-1-2047, NSF grant CCF-1527501, ARO grant W911NF-15-1-0316, AFOSR grant FA9550-14-1-0088, ONR grant N00014-17-1-2551, DARPA REVEAL grant HR0011-16-C-0028, and an ONR BRC grant for Randomized Numerical Linear Algebra. This work was partially supported by EADM MADICS and SABIOD.org.

References

Andén, J. and Mallat, S. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014.

Brown, J. C. Calculation of a constant q spectral transform. *Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

Cakir, E., Ozan, E. C., and Virtanen, T. Filterbank learning for deep neural network based polyphonic sound event detection. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp. 3399–3406. IEEE, 2016.

Clough, R. W. Original formulation of the finite element method. *Finite Elements in Analysis and Design*, 7(2): 89–101, 1990.

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pp. 2990–2999, 2016.

Cosentino, R., Balestriero, R., and Aazhang, B. Best basis selection using sparsity driven multi-family wavelet transform. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 252–256, Dec. 2016. doi: 10.1109/GlobalSIP.2016.7905842.

Cosentino, R., Balestriero, R., Baraniuk, R., and Patel, A. Overcomplete frame thresholding for acoustic scene analysis. *arXiv preprint arXiv:1712.09117*, 2017.

Dai, W., Dai, C., Qu, S., Li, J., and Das, S. Very deep convolutional neural networks for raw waveforms. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pp. 421–425. IEEE, 2017.

Glotin, H., Ricard, J., and Balestriero, R. Fast chirplet transform injects priors in deep learning of animal calls and speech. In *International Conference on Learning Representations (ICLR), Workshop*, 2017.

Grill, T. and Schlüter, J. Two convolutional neural networks for bird detection in audio signals. In *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, Kos Island, Greece, August 2017. URL http://ofai.at/~jan.schlueter/pubs/2017_eusipco.pdf.

Hall, C. A. and Meyer, W. W. Optimal error bounds for cubic spline interpolation. *Journal of Approximation Theory*, 16(2):105–122, 1976.

Harris, F. J. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1):51–83, 1978.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Jaffard, S., Meyer, Y., and Ryan, R. *Wavelets: Tools for Science and Technology*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, 2001. ISBN 9780898714487. URL https://books.google.com/books?id=hAwhJ0mLKaMC.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

Leung, M. K., Xiong, H. Y., Lee, L. J., and Frey, B. J. Deep learning of the tissue-regulated splicing code. *Bioinformatics*, 30(12):i121–i129, 2014.

Lostanlen, V. *Opérateurs convolutionnels dans le plan temps-fréquence*. PhD thesis, Paris Sciences et Lettres, 2017.

Mallat, S. *A Wavelet Tour of Signal Processing*. Academic press, 1999.

Megahed, A., Moussa, A. M., Elrefaie, H., and Marghany, Y. Selection of a suitable mother wavelet for analyzing power system fault transients. In *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, pp. 1–7. IEEE, 2008.

Meyer, Y. Wavelets-Algorithms and Applications. *Wavelets-Algorithms and applications Society for Industrial and Applied Mathematics Translation., 142 p.*, 1, 1993.

Olshausen, B. A. and Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.

Ramli, D. A. and Jaafar, H. Peak finding algorithm to improve syllable segmentation for noisy bioacoustic sound signals. *Procedia Computer Science*, 96:100–109, 2016.

Sainath, T. N., Weiss, R. J., Senior, A., Wilson, K. W., and Vinyals, O. Learning the speech front-end with raw waveform cldnns. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

Schoenberg, I. J. On interpolation by spline functions and its minimal properties. In *On Approximation Theory/Über Approximationstheorie*, pp. 109–129. Springer, 1964.

Serizel, R., Bisot, V., Essid, S., and Richard, G. Acoustic features for environmental sound analysis. In *Computational Analysis of Sound Scenes and Events*, pp. 71–101. Springer, 2018.

Shera, C. A., Guinan, J. J., and Oxenham, A. J. Revised estimates of human cochlear tuning from otoacoustic and behavioral measurements. *Proceedings of the National Academy of Sciences*, 99(5):3318–3323, 2002.

Stowell, D. and Plumbley, M. D. An open dataset for research on audio field recording archives: freefield1010. *CoRR*, abs/1309.5275, 2013. URL http://arxiv.org/abs/1309.5275.

Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M. A., Schuller, B., and Zafeiriou, S. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 5200–5204. IEEE, 2016.

Trone, M., Glotin, H., Balestriero, R., and Bonnett, D. E. Enhanced feature extraction using the morlet transform on 1 mhz recordings reveals the complex nature of amazon river dolphin (inia geoffrensis) clicks. *Journal of the Acoustical Society of America*, 138(3):1904–1904, 2015.

Unser, M. A. Ten good reasons for using spline wavelets, 1997a. URL http://dx.doi.org/10.1117/12.292801.

Unser, M. A. Ten good reasons for using spline wavelets. In *Wavelet Applications in Signal and Image Processing V*, volume 3169, pp. 422–432. International Society for Optics and Photonics, 1997b.

Xu, C., Wang, C., and Liu, W. Nonstationary vibration signal analysis using wavelet-based time–frequency filter and Wigner-Ville distribution. *Journal of Vibration and Acoustics*, 138(5):051009, 2016.

Zeghidour, N., Usunier, N., Kokkinos, I., Schatz, T., Synnaeve, G., and Dupoux, E. Learning filterbanks from raw speech for phone recognition. *CoRR*, abs/1711.01161, 2017. URL http://arxiv.org/abs/1711.01161.