Learning Sparse Analytic Filters for Piano Transcription

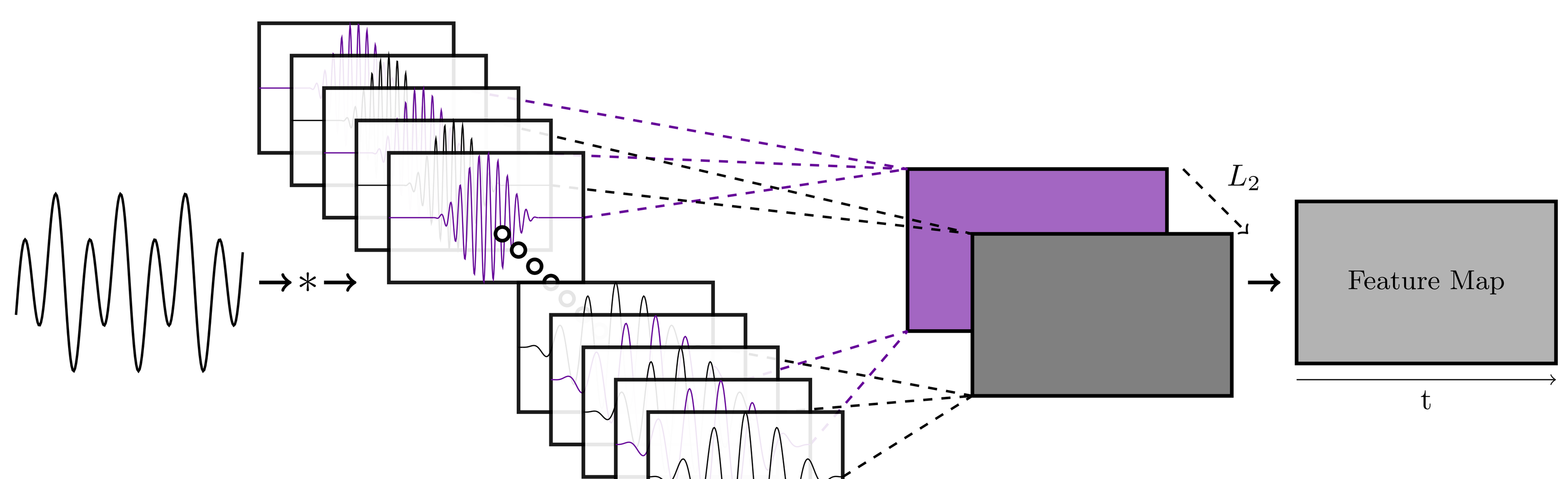
Frank Cwitkowitz, Mojtaba Heydari, and Zhiyao Duan

Audio Information Research Lab, University of Rochester

Overview

- Several variations of a complex filterbank learning module are investigated as a frontend for a baseline piano transcription model [1, 2].
- Techniques are introduced to learn analytic filters and to enforce sparsity among the weights of each filter.

Filterbank Learning Module



The complex filterbank learning module is formulated such that an inner product is taken between a time-domain signal x and M filters, indexed by μ, of respective lengths l_μ with weights θ_μ at discrete hops k spaced l_h samples apart.

$$X[k, \mu] = \sum_{n=0}^{l_\mu - 1} x[kl_h + n]\theta_\mu[n] \quad (1)$$

The real and imaginary response of complex filter μ are taken separately, and are combined using L_2 pooling, a simple mechanism for computing the magnitude.

$$|X[k, \mu]| = \sqrt{(x * \mathcal{R}(\theta_\mu))^2 + (x * \mathcal{I}(\theta_\mu))^2} \quad (2)$$

Forcing Analyticity

The module can be re-formulated to learn only the real part of each filter and to infer the imaginary part using the Hilbert Transform $H(\cdot)$. The resulting filters are analytic [3] and shift invariant.

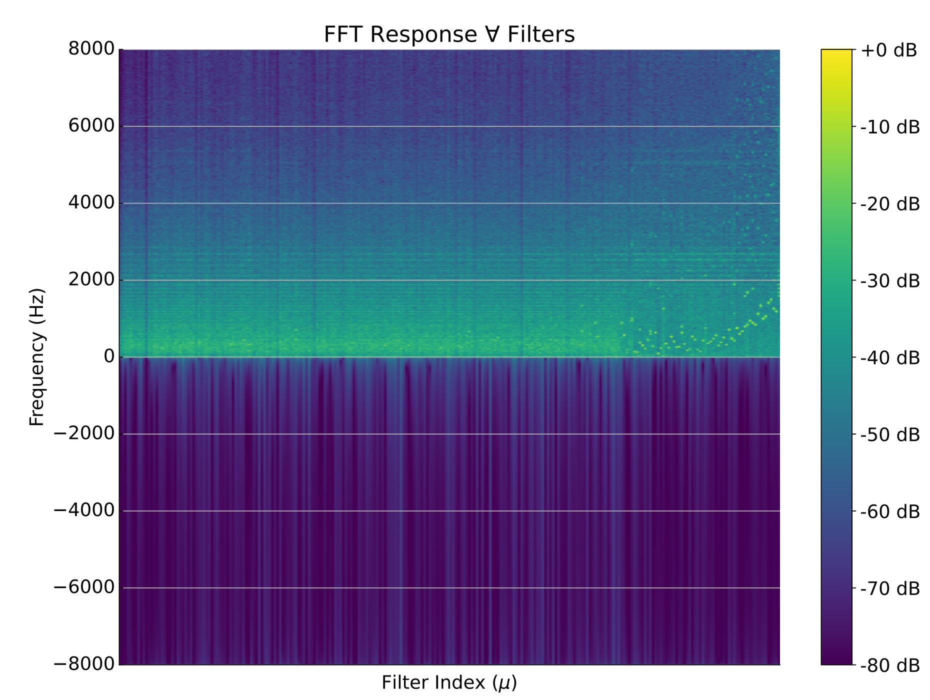
$$|X[k, \mu]| = \sqrt{(x * \mathcal{R}(\theta_\mu))^2 + (x * H(\mathcal{R}(\theta_\mu)))^2} \quad (3)$$

Inducing Sparsity

Sparse filters are learned by applying variational dropout [4], treating weights as random variables with learned variance σ_μ^2.

$$X[k, \mu] \sim \mathcal{N}(x * \theta_\mu, x^2 * \sigma_\mu^2). \quad (4)$$

Frequency Response



Experimental Results

	MAESTRO			MAPS		
Experiment	Frame F_1	Note-On F_1	Note-Off F_1	Frame F_1	Note-On F_1	Note-Off F_1
mel	91.80*	95.95*	83.44*	81.40*	81.42*	59.15*
mel	**90.91**	**95.82**	**83.14**	**81.26**	83.86	**59.07**
cqt	90.79	95.29	82.30	77.46	82.35	52.18
vqt	90.18	94.74	80.51	80.26	83.42	55.34
fixed comb	87.59	92.19	75.09	80.30	**84.64**	57.22
cl+rnd	86.70	91.22	73.72	70.03	78.77	43.92
cl+vqt	87.53	92.24	75.64	72.90	80.90	48.06
hb+rnd	86.50	91.30	73.19	76.13	80.00	51.88
hb+vqt	**87.81**	**92.63**	**76.01**	75.11	80.71	50.66
hb+rnd+brn	85.23	90.19	70.75	74.81	79.48	50.57
hb+rnd+gau	85.63	90.41	71.81	75.58	80.44	51.28
hb+rnd+var	86.04	90.61	72.27	73.59	80.09	47.99
hb+vqt+var	87.45	92.39	74.92	75.62	80.80	50.55
hb+comb+var	87.52	92.27	75.69	**76.74**	**80.98**	**52.57**

- Learned filterbanks underperform standard time-frequency features.
- Randomly-initialized filterbanks perform on-par with VQT counterparts.

Acknowledgements & References

This work has been funded by the National Science Foundation grants IIS-1846184 and DGE-1922591.

All of the code is available at `https://github.com/cwitkowitz/sparse-analytic-filters`.

[1] Curtis Hawthorne et al. "Onsets and Frames: Dual-Objective Piano Transcription". In: *Proceedings of ISMIR*. 2018.

[2] Curtis Hawthorne et al. "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset". In: *Proceedings of ICLR*. 2019.

[3] Manuel Pariente et al. "Filterbank Design for End-to-End Speech Separation". In: *Proceedings of ICASSP*. 2020.

[4] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. "Variational Dropout Sparsifies Deep Neural Networks". In: *Proceedings of ICML*. 2017.

Filter Examples