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 $\mu$, of respective lengths $l_\mu$, with weights $\theta_\mu$ at discrete hops $k$ spaced $l_\mu$ samples apart.

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

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

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

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.

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

Inducing Sparsity

Sparse filters are learned by applying variational dropout [4], treating weights as random variables with learned variance $\sigma^2_\mu$.

$$X[k, \mu] \sim N(x * \theta_\mu, x^2 * \sigma^2_\mu)$$

Frequency Response

Experimental Results

Filter Examples

Acknowledgements & References

This work has been funded by the National Science Foundation grants IIS-1846384 and DGE-1922591. All of the code is available at https://github.com/cwitkowitz/sparse-analytic-filters.


