MIREX 2017: FREQUENCY DOMAIN CONVOLUTIONS FOR MULTIPLE F0 ESTIMATION

John Thickstun[⋆] **Zaid Harchaoui**[⋆] **Dean Foster**[†] **Sham M. Kakade**[⋆]
[⋆] University of Washington, [†] Amazon
{thickstn,sham}@cs.washington.edu, zaid@uw.edu, dean@foster.net

ABSTRACT

This document describes the THK1 submission in the 2017 MIREX Multi-F0 competition. The model is a convolutional neural network, trained using the MusicNet labels. Its input is a bank of logarithmically-spaced frequency filters. These filters exhibit translation invariance along the log-frequency axis, which is captured in this model by one-dimensional convolutions along the frequency axis. The model fully connects across the time axis to capture temporal dependencies. The training data is augmented by pitch-shifting the original data by up to 5 semitones and applying small (up to 1/10 semitone) continuous pitch jitter to the input.

1. ARCHITECTURE

The MusicNet dataset and training methodology used for this model are described in [1]. In contrast to the raw-audio training described in [1], this model is trained on a bank of 512 cosine-windowed sinusoidal filters with logarithmically spaced frequency ranging from 50Hz and 6kHz. Constructing a hand-crafted filterbank is advantageous because its output exhibits a topological structure that can be exploited by the convolutional architecture.

The model accepts input audio frames of $16,384$ samples (at 44.1kHz; approximately $1/3$ of a second). Each audio frame vector is normalized to unit magnitude and filterbank features are constructed at a 512-sample stride (approximately 10ms). Each filter is a cosine-windowed sinusoid of $4,096$ samples, which results in $25 = (16,348 - 4,096)/512$ filterbank features per audio frame; the input to the learned portion of the network is a 512×25 dimensional volume.

The learned network has two layers, plus a linear classifier layer on top. The first layer of the model convolves along the frequency axis of the input volume. Using 128 hidden nodes of size 128×1 with a stride of 2 gives us an output from level 1 of $193 \times 25 \times 128$ features. The second layer of the network fully connects across the time axis using $4,096$ hidden nodes of size $1 \times 25 \times 128$, producing an output from level 2 of $193 \times 4,096$ features. This representation is fed to a linear classifier for each pitch.

2. DATA AUGMENTATION

To avoid overfitting, we augment the MusicNet dataset by randomly stretching each input audio frame. We can speed up or slow down the audio by linear interpolation and thus control its pitch. Too much stretching distorts the audio and limits its effectiveness as training data; we limit stretching of to a maximum of 5 semitones. In addition to full-semitone pitch-shifts we also train with continuous jitter, speeding up or slowing down the audio by a uniformly random multiplicative factor in (-1/10,1/10).

3. OPTIMIZATION

The model was trained with minibatch SGD using momentum and iterate averaging. The optimizer processed 150 datapoints per minibatch with a learning rate of 10^{-6} and $.95\%$ momentum. The final network weights are computed from a moving average of iterates with a decay factor of 2×10^{-4}. We implemented this optimization in Tensorflow, and accelerated training with an NVIDIA 1080Ti GPU.

4. RESULTS

We achieve 78.8% average precision on a representative subset of the MusicNet dataset and 83.1% average precision on the MIREX MultiF0 development set. Using mir_eval [2] and a threshold of $.4$ for predicted output, we find the following for this model on the MIREX dev set:

P	R	Acc	Etot	Esub	Emiss	Efa
79.76	73.75	0.62	0.35	0.10	0.17	0.09

5. REFERENCES

[1] John Thickstun, Zaid Harchaoui, and Sham M. Kakade. "Learning Features of Music from Scratch," *ICLR*, 2017.

[2] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel P. W. Ellis. "mir_eval: A Transparent Implementation of Common MIR Metrics," *ISMIR*, 2014.