LEARNING INVARIANTS FOR POLYPHONIC INSTRUMENT RECOGNITION

First author
Affiliation1
author1@ismir.edu

Second author
Retain these fake authors in submission to preserve the formatting

Third author
Affiliation3
author3@ismir.edu

ABSTRACT

The abstract should be placed at the top left column and should contain about 150-200 words.

1. INTRODUCTION

2. DEEP CONVOLUTIONAL NETWORKS

2.1 Time-frequency representation

We used the implementation from the librosa package [3] with $Q = 12$ filters per octave, center frequencies ranging from 55 Hz to 14 kHz (8 octaves from A1 to A9), and a hop size of 23 ms. Furthermore, we applied perceptual weighting of loudness in order to reduce the dynamic range between the fundamental partial and its upper harmonics. A 3-second sound excerpt $x(t)$ is represented by a time-frequency matrix $\boldsymbol{x_1}(t, k_1)$ of width $T = 128$ samples and height $K_1 = 96$ MIDI indices.

2.2 Architecture

First of all, we apply a family $\boldsymbol{W_2}(t, k_1, k_2)$ of $K_2 = 50$ learned time-frequency convolutional operators. Furthermore, element-wise biases $\boldsymbol{b_2}(t, k_1, k_2)$ are added to the convolutions, resulting in the tensor

$$\boldsymbol{y_2}(t, k_1, k_2) = \boldsymbol{b_2} + (\boldsymbol{x_1} \overset{t,k_1}{*} \boldsymbol{W_2}). \quad (1)$$

The second step in the network is the application of a pointwise nonlinearity. We have chosen the *rectified linear unit* (ReLU) because of its popularity in computer vision and its computational efficiency.

$$\boldsymbol{y_2^+}(t, k_1, k_2) = \max\left(\boldsymbol{y_2}(t, k_1, k_2), 0\right) \quad (2)$$

$$\boldsymbol{x_2}(t, k_1, k_2) = \max_{\substack{|\tau| \leq \Delta t \\ |\kappa_1| \leq \Delta k_1}} \left\{ \boldsymbol{y_2^+}(t + \tau, k_1 + \kappa_1, k_2) \right\} \quad (3)$$

$$\boldsymbol{y_3}(t, k_1, k_3) = \sum_{k_2} (\boldsymbol{x_2} \overset{t,k_1}{*} \boldsymbol{W_3}) \quad (4)$$

$$\boldsymbol{x_4}(k_4) = \left(\sum_{v_3} \boldsymbol{W_4}(k_4, v_3) \boldsymbol{x_3}(v_3) \right)^+ \quad (5)$$

$$\boldsymbol{x_5}(k_5) = \left(\sum_{k_4} \boldsymbol{W_5}(k_5, k_4) \boldsymbol{x_4}(k_4) \right)^+ \quad (6)$$

$$\boldsymbol{x_6}(k_6) = \sigma \left(\sum_{k_6} \boldsymbol{W_6}(k_6, k_5) \boldsymbol{x_5}(k_5) \right) \quad (7)$$

2.3 Training

The network is trained on categorical cross-entropy with *Adam* [2], a state-of-the-art stochastic optimizer for gradient-based learning.

3. DEEP SUPERVISION OF MELODIC CONTOUR

3.1 Disentangling pitch from timbre

3.2 Extraneous supervision

3.3 Joint supervision

4. SINGLE-INSTRUMENT CLASSIFICATION

4.1 Experimental design

In order to evaluate the proposed algorithms, we used MedleyDB [1], a dataset of 122 multitracks annotated with instrument activations as well as melodic f_0 curves when present.

4.2 Results

5. POLYPHONIC CLASSIFICATION

5.1 Experimental design

5.2 Results

6. CONCLUSIONS

7. REFERENCES

[1] Rachel Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Bello. Medleydb: a multitrack dataset for annotation-intensive mir research. *International Society for Music Information Retrieval Conference*, 2014.

[2] Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations*, pages 1–13, 2015.

[3] Brian McFee, Matt McVicar, Colin Raffel, Dawen Liang, Oriol Nieto, Eric Battenberg, Josh Moore, Dan Ellis, Ryuichi Yamamoto, Rachel Bittner, Douglas Repetto, Petr Viktorin, Joo Felipe Santos, and Adrian Holovaty. librosa: 0.4.1. zenodo. 10.5281/zenodo.18369, October 2015.