MIREX 2017 : Separately training Convolutional Neural Nets for Ensemble category estimation and for Multiple F0 estimation

Shinjiro Mita, Gaku Hatanaka, Alexis Meneses, Nattapong Thammasan, Daiki Miura

Osaka University
mita@fbs.osaka-u.ac.jp

ABSTRACT

In this submission for MIREX 2017, our primary purpose is to test how well Convolutional Neural Network (CNN) performs on F0 estimation tasks of "multi-instrumental" music. We tried End-to-End training CNN by using more than 100,000 "multi-instrumental" tunes. Another purpose in this submission is to compare which approach – (1) training CNN End-to-End approach (another submission), or (2) separately training between ensemble category estimation CNN and f0 estimation CNN (this submission) – is better for F0 estimation tasks of multi-instrumental music.

1. INTRODUCTION

Automatic music transcription (AMT), especially multiple fundamental frequency (F0) estimation, is one of the fundamental tasks in music information retrieval (MIR).

Compared to F0 estimation from the monophonic music, multiple F0 estimation from polyphonic music is more challenging in that harmonics of multiple tones overlapping one another prevent machines from recognizing each tone with high accuracy.

Recently, data-driven deep learning approaches have been applied to variety of recognition tasks and achieved outstanding successes [1] : image category classification tasks, language sound recognition tasks, and so on. This approach has also been applied to multiple F0 estimation from polyphonic music [2,3] outperforming previous approaches such as NMF (non-negative matrix factorization) [4], and PLCA (probabilistic latent component analysis) [5].

However, the researches mentioned above are using dataset consisting of only single instrumental music (especially piano music). Therefore, how well data-driven deep learning approaches perform on "multi-instrumental" polyphonic music is still challenging cutting-edge problem in AMT.

In this MIREX 2017 submission, we try to apply CNN for F0 estimation from multi-instrumental polyphonic music, and test how well CNN performs after End-to-End training by using more than 100,000 tunes (synthesized audio from midi files).

2. METHOD

2.1 Dataset preparation

We used more than 130,000 midi file collection [a] consisting of various types of music – Pop, Rock, Jazz, Classic, and others. After eliminating broken or empty midi files, we synthesized 105,579 audio files from the midi files. In order to synthesize audio files, we used soundfont files – "Yamaha Disklavier Pro Grand Piano" [b] for piano part, and "Fluid R3 GM" [c] for other instruments parts.

2.2 Pre-processing

We applied constant Q transform (CQT) for each audio to get spectrogram matrix by using the library librosa [6]. We tried 3 kinds of filter scales = {0.8, 1.0, 1.2} exploring the optimal filter size of CQT. The sampling rate of the spectrogram and ground truth piano-roll were 50 Hz. As for frequency axis of the spectrogram, we divided 36 bins per octave, so the data size of frequency axis was 264 dimensions. After CQT, for each tune, we normalized spectrogram so that mean is 0 and standard deviation is 1, then applied logarithmic magnitude for the spectrogram. In terms of psychophysics, the Weber-Fechner law, perceived intensity is proportional to logarithm of the actual intensity measured with an accurate nonhuman instrument, roughly holds in any kinds of perception. Spectrogram with logarithmic magnitude scale may more approximate to human perception of sound intensity than that with linear scale.

2.3 Model architecture

Our model architecture is shown in Figure 1. In Figure 1, the CNN on the left is a classifier to predict 4 ensemble categories – (1) Piano solo, (2) Including other instruments than piano except drums, (3) Ensemble including drums, and (4) Blank. The model structure of the classifier is shown in Table 1. The classifier predicts the output ensemble category label binary vector from the 0.9 sec (45 frames) CQT spectrogram input. The right part of

Figure 1 is composed of 3 CNNs. The CNNs are F0 estimators trained by using the spectrogram input for each ensemble category. The architectures of the F0 estimators are based on ResNet [7]. The number of layers is 18. The model input and output structures are changed to fit in the size of 264 frequency bins and 0.1 sec time window (5 frames) in the spectrogram matrix and output 88 pitches * 1 frame, respectively.

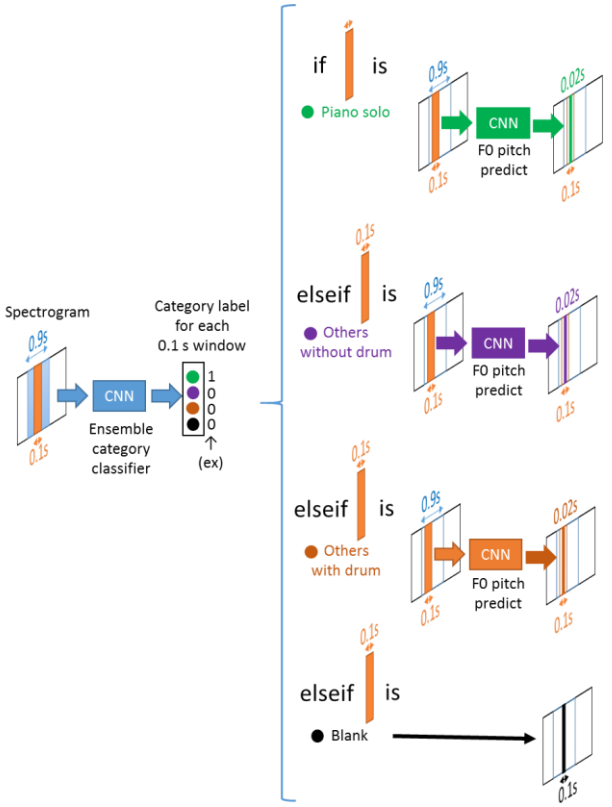

Figure 1 : The workflow of our model connecting ensemble category classifier and F0 estimators

2.4 Model training

We split the 105,579 tunes into 90,000 tunes for training models, 10,000 tunes for validation, and 5579 tunes for evaluation and exploring the best binary threshold value for post-processing.

We used the library - TensorFlow & Keras API [d] - for training models. We used Adam optimizer [8] and categorical cross entropy as the error function for training models. After going full one circle of 100,000 tunes, we finished training.

Layer name	Value
Input	(Frequency axis, Time axis) = (264, 45)
Convolution	(N of filters, Filter size time axis, Filter size frequency axis) = (32, 3, 3)
Activation function	Hyperbolic tangent
Max pooling	(Time axis, Frequency axis) = (1,2)
Dropout	0.25
Convolution	(64, 3, 3)
Activation function	Hyperbolic tangent
Max pooling	(1,2)
Dropout	0.25
Fully connected	64
Activation function	Relu
Dropout	0.5
Fully connected	4
Output	Binary category vector (4, 1)

Table 1 : The architecture of ensemble category classifier

2.5 Post-processing

We chose the binary threshold (between 0 and 1) by evaluating the accuracy by using 5579 tunes after model training. Then, we connected 4 trained CNNs into 1 system as the workflow shown in Figure 1. Overall, the system results in the output of the predicted binary piano-roll matrix whose structure is 88 pitch * number of frames.

3. EVALUATION

The formal evaluation result is going to be published in the MIREX 2017 results.

4. REFERENCES

[1] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.

[2] S. Sigtia, E. Benetos, and S. Dixon, “An end-to-end neural network for polyphonic piano music transcription,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):927–939, May 2016.

[3] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, “On the Potential of Simple Framewise Approaches To Piano Transcription.”, *ISMIR 2016*, 475–481

[4] E. Vincent, N. Bertin, and R. Badeau, “Adaptive harmonic spectral decomposition for multiple pitch estimation,” *IEEE Transactions on Audio, Speech, and Language Processing*., vol. 18, no. 3, pp. 528–537, 2010.

[5] E. Benetos and S. Dixon, “A shift-invariant latent variable model for automatic music transcription,” *Computer Music Journal*, vol. 36, no. 4, pp. 81–94, 2012.

[6] B. Mcfee, C. Raffel, D. Liang, D. P. W. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, “librosa: Audio and Music Signal Analysis in Python,” *Proceedings of the 14th Python in science conference*, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Arxiv.Org*, vol. 7, no. 3, pp. 171–180, 2015.

[8] D. Kingma and J. Ba, “ADAM: A Method for Stochastic Optimization,” Proc. 3rd Int. *Conf. Learn. Represent.*, pp. 1–15, 2015.

5. OTHER MATERIALS

URLs below are as of August, 2017

[a] 130,000 midi file collection

https://mega.nz/#!Elg1TA7T!MXEZPzq9s9YObiUcMCoNQJmCbawZqzAkHzY4Ym6Gs_Q

[b] Yamaha Disklavier Pro Grand Piano SoundFont

http://freepats.zenvoid.org/Piano/acoustic-grand-piano.html

[c] Fluid R3 GM SoundFont

https://sourceforge.net/p/fluidsynth/wiki/SoundFont/

[d] TensorFlow r1.2 / Keras 2 API

https://www.tensorflow.org/versions/r1.2/

https://faroit.github.io/keras-docs/1.1.1/