Transcription-driven spectral basis learning for audio source separation combining Convolutional Neural Networks and Non-negative Matrix Factorization

Alejandro Koretzky
Splice
ale@splice.com

Naveen Rajashekharappa
Splice
naveen@splice.com

ABSTRACT

In the context of Audio Source Separation, one of the main limitations of supervised Non-negative Matrix Factorization (NMF) solutions is the difficulty in designing optimal spectral bases that generalize to any input mix. This lack of generalization has been one of the main reasons why most of the current solutions rely on artificial neural networks (ANN). In this contribution we present a hybrid, transcription-driven template-learning approach that combines the power of ANN with the simplicity and performance of NMF, achieving high-quality, real-time, low interference separation of drums & percussion components. Where in most implementations, NMF-based solutions try to estimate the Activations Matrix (H) given an input mix and a static, manually-defined set of spectral bases (W matrix), here we adapt the transcription output from an ANN to instantiate the H matrix, and have NMF estimate the W matrix instead, resulting in optimal spectral templates that adapt to the input mix. Because the task of transcription is, in general, less complex compared to the task of audio source separation, we still end up with a highly efficient, fast-inference, low-memory footprint pipeline that can run on CPU, making it particularly suitable for client-side implementations as part of creator tools for music production.

1. INTRODUCTION

The problem that we are trying to solve can be described as follows: given an input mix made of drums and percussion content, we want to separate the mix into 5 component layers corresponding to the following classes: *kick, snare/clap, open hi-hat, closed hi-hat, toms/cowbells.*

2. DESIGN

2.1 Transcription task

We designed a regression Convolutional Neural Network (CNN) that accepts a segment of the input mix represented as a Mel spectrogram and outputs the onset profile for each of the predefined classes.

The model consists of three blocks of 2 convolutional layers followed by a max-pooling layer, where N_MELS = 512 and TIME_STEPS = 344.

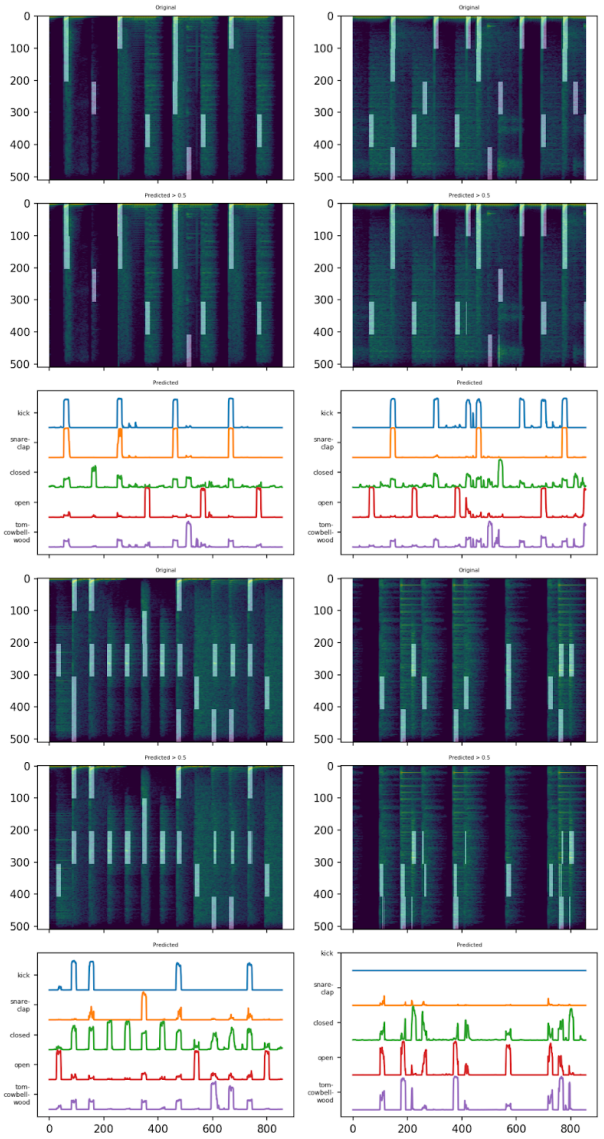

Figure 1. Example outputs from our transcription network. For each block, the top image represents transcription ground truth aligned with the underlying Mel spectrogram, the middle image, the resulting transcription after

thresholding of the transcription output and the bottom, the raw output from the network.

```
model = Sequential()
model.add(InputLayer(input_shape=(N_MELS, TIME_STEPS, 1)))
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(5, 5)))

model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(5, 5)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(5, 5)))
model.add(Dropout(0.5))

model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(N_CLASSES, activation='sigmoid'))
```

Figure 2. CNN model architecture using the Keras API.

The model was trained on a proprietary dataset of synthetically-generated drum patterns with their annotated transcription information and we relied on several training augmentation strategies such as random-slicing the input mix and time-stretching. For the learning process we used Stochastic Gradient Descent (SGD) with a learning rate of 0.0001.

2.2 Spectral bases learning and source separation

Given an input mix and the generated transcription from our CNN model, we use this transcription information in NMF by instantiating the Activations Matrix (H) using a smoothed version of the transcription profile of each layer. In this context the number of columns in the basis matrix (W) equals the number of rows in H, which corresponds to the number of transcription layers by design (5).

We perform NMF by minimizing the generalized Kullback-Liebler (KL) divergence cost function, implemented by the ubiquitous multiplicative updates approach. Upon convergence and because of the coherence that we enforce by setting the Activations Matrix H to follow the transcription information or onset function associated with each layer, our NMF optimization successfully learns spectral bases that are highly relevant to the input mix, while minimizing interference between sources. With both W and H matrices populated, we now reconstruct each of the sources using the well-known Wiener filtering inspired approach, and relying on the input mix's original phase information for inverting the resulting estimated spectrograms back into the time domain via Inverse Short Time Fourier Transform. (iSTFT).

3. EXTENSIONS

3.1 Similarity search

An interesting extension of the solution is the use of the separated drums and percussion components for the purpose of similarity search. In the context of music production, it is not uncommon to find a drum loop and want to either use a specific isolated component or find similar high-quality one-shot sounds available as part of a library. Using simple heuristics on the separated component layers we were able to identify the highest quality, lowest interference instance of a certain component (e.g. a kick sound) to use it for the purpose of content-based retrieval. The results were highly positive, validating the overall approach for content-based retrieval use cases.

3.2 Sample replacement

Related to 3.1 and by generating a midi representation from the transcription profiles, we recomposed the original input loops using the closest one-shot replacements using similarity search and performed perceptual A/B sessions comparing the original drum loop and the recomposed one using the closest sample replacements, and a video demo is available as part of this submission. This use case is highly promising in the context of music production tools at the intersection of search, discovery and creation.

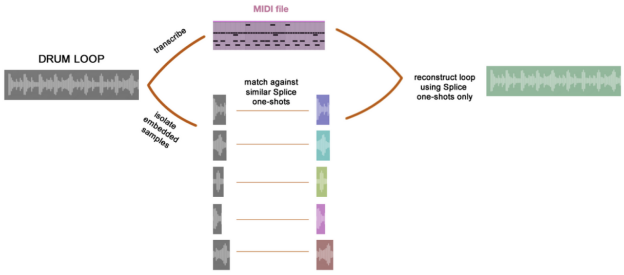

Figure 3. Diagram representing the process and operations for the sample replacement use case.

4. REFERENCES

[1] N. Bryan, D. Sun, E. Cho. Single-Channel Source Separation Tutorial Mini-Series. CCRMA. *https://ccrma.stanford.edu/~njb/teaching/sstutorial/*

[2] P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'03), pages 177–180*, October 2003.

[3] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, Nov 2004.