MIREX 2016 SUBMISSION: LARGE VOCABULARY AUTOMATIC CHORD ESTIMATION WITH DEEP NEURAL NETWORKS

Junqi Deng, Yu-Kwong Kwok
Department of Electrical and Electronic Engineering
The University of Hong Kong
{jqdeng,ykwok}@eee.hku.hk

ABSTRACT

This extended abstract presents two types of systems. These systems take a traditional Gaussian-HMM ACE approach as a motherboard and replace certain key components by different deep neural networks.

1. ACRONYMS

Here are some acronyms or pre-knowledges in order to follow this article:

- ACE: Automatic Chord Estimation (or Audio Chord Estimation)
- DBN: Deep Belief Network
- RNN: Recurrent Neural Network
- LSTM: Long-short-term-memory
- BLSTM: Bidirectional Long-short-term-memory
- MajMin: The "Major and minor" vocabulary considered in MIREX ACE.
- SeventhsBass: The "Seventh chords with inversions" vocabulary considered in MIREX ACE.
- Chordino: An ACE system implemented by M. Mauch [1] .

2. INTRODUCTION

There are two kinds of systems submitted. The first kind considers a hybrid use of a Chordino-like segmentation engine and deep neural networks (System I); The second kind considers using a Chordino-like feature extraction process with an end-to-end BLSTM-RNN sequence transducer (System II).

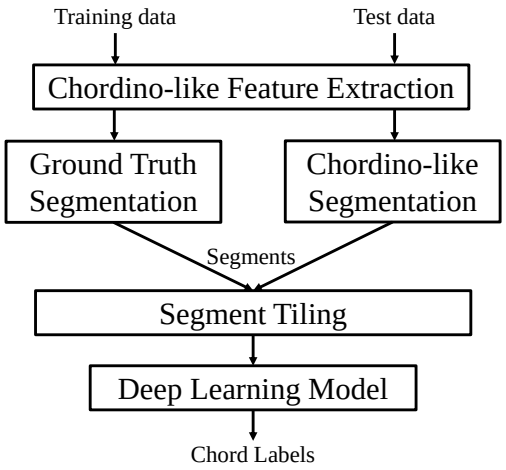

Figure 1. System overview. The audio input (test data) goes through a Chordino-like process for segmentation, then the segments are classified into chord labels.

3. SYSTEM I

3.1 Overview

Figure 1 shows the ACE system framework. As for training, each input audio track is first feature extracted using a Chordino-like process (Section 3.2). The feature sequence is then segmented (in terms of chords) by ground truth annotations. Then each feature segment goes through a process (Section 3.4) that tiles it into a fixed number of sub-segments. Each set of sub-segments and its corresponding ground truth chord label together form one training case to train a deep learning model (Section 3.5) that learns the input-output relationship. As for testing, except for the ground-truth segmentation process is replaced by a Chordino-like segmentation process (Section 3.3), and the deep learning model used for prediction, the work flow remains the same.

3.2 Chordino-like Feature Extraction Process

This process is similar to the one described in [2]. It takes the raw input and resamples it at 11025 Hz, and transforms it using short-time-Fourier-transform (STFT) with 4096-point Hamming window and 512-point hop size . To

[1] http://www.isophonics.net/nnls-chroma

Table 1. Chordino-like segmentation process parameters

	μ	σ^2
Bass - chord bass	1	0.1
Bass - not chord bass but chord note	1	0.5
Bass - not bass	0	0.1
Treble - chord note	1	0.2
Treble - not chord note	0	0.2
No Chord (for all notes)	1	0.2

make more musical sense, it then transforms the linear-frequency-scale spectrogram (2049-bin) to log-frequency-scale (252-bin, 3 bins per semitone ranging from MIDI note 21 to 104), and tunes the spectrogram to standard tuning as indicated in Chapter 3 of [2]. To enhance harmonic content and attenuate background noise, it further performs a standardization process. Then it uses the NNLS method to extract note activation patterns (84-bin, 1 bin per semitone). This matrix is further reduced to a 24-bin per column bass-treble chromagram weighted by the bass and treble profiles.

3.3 Chordino-like Segmentation Process

The Chordino-like segmentation process [2] decodes the bass-treble chromagram feature sequence using a probabilistic model. It applies a Gaussian-HMM as the decoding/smoothing engine. The HMM's hidden node models discrete states corresponding to all chord classes, and the observable node models continuous states corresponding to a chroma. The emission probabilities are designed as Gaussian with parameters shown in Table 1. The prior matrix is uniform and the transition matrix is set with extremely high uniform self-transition weight (more than 99 times of non-self-transition weight) and extremely low uniform non-self-transition weight.

3.4 Segment Tiling Process

Segment tiling refers to a computing procedure that first divides a segment into a N equal-sized sub-segments, then takes the average feature of each sub-segment, yielding a N-frame segment. If the number of frames of the original segment is not divisible by N, then the last frame is extended several times to make it divisible.

3.5 Deep Learning Models

Each N-frame segment will be classified as a chord label through a deep learning model. Two types of model are considered: DBN and BLSTM-RNN.

3.5.1 Deep Belief Network

The DBN, as shown in Figure 2, is built upon a fully connected feedforward neural network. It has an input layer of N-frame of features, and has multiple hidden layers. At the pre-training phrase, every pair of layers (except for the output layer), are trained one at a time as restricted Boltzmann machine (RBM). The RBM formed by the input layer and the first hidden layer is a Gaussian-Bernoulli RBM, since the input layer has feature in continuous value space. The

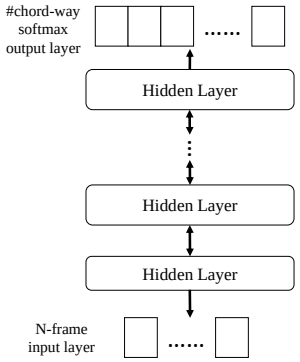

Figure 2. The deep belief network architecture used in our system. All layers are fully connected. The input is N-frame of features.

RBMs formed by neighboring hidden layers are Bernoulli-Bernoulli RBMs, since each neuron is stochastic binary. At the fine-tuning phrase, the network is regarded as a deep feedforward neural network and trained via stochastic gradient descent with backpropagation.

3.5.2 Recurrent Neural Network

The RNN shown in Figure 3 is a bidirectional one with long-short-term-memory (LSTM) hidden units. Thus it is a BLSTM-RNN. LSTM is introduced in order to relief gradient vanishing/exploding problem for long sequence training. For a fixed length input, the RNN is expanded to N frames, each taking one frame of input features. A mean pooling operation is inserted before the output layer to summarize the outputs of N frames.

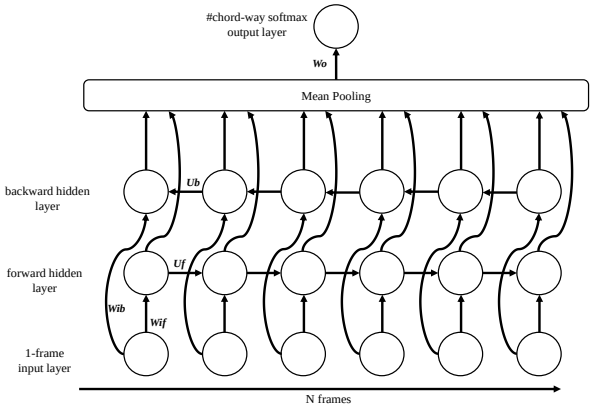

Figure 3. The bidirectional recurrent neural network architecture used in our system. Both hidden layers employ LSTM units in place of normal logistic units. The RNN is expanded to N frames, with mean pooling to summarize results.

3.6 Training

A DBN model is pre-trained using persistent-contrastive-divergence (PCD-20), for 100 epochs with learning rate

0.001. It is fine-tuned using mini-batch stochastic gradient descent regularized with dropout and early-stopping. During mini-batch stochastic gradient descent, both MLP and DBN use a learning rate of 0.01 and batch size of 100. A BLSTM-RNN model is trained using an Adadelta optimizer, regularized with dropout and early-stopping. All dropout probabilities are set to 0.5. All early-stopping criteria is monitored by a validation set, which is a random choice of 20% of the training dataset. The other %80 are used for computing gradients and updating the network. For reproducible research, all implementation details are available online [2] .

4. SYSTEM II

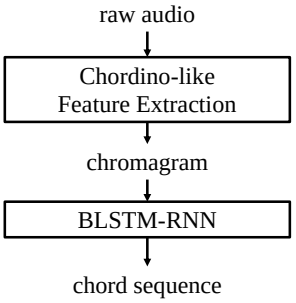

Figure 4. System Overview. The raw audio is transformed by a chordino-like feature extraction frontend into a piece of chromagram, and then decoded by a BLSTM-RNN into chord sequence.

System II considers a Chordino-like feature extraction process with an end-to-end BLSTM-RNN as sequence classifier. Figure 4 shows the system overview, which mainly contains a chordino-like feature extraction frontend and a BLSTM-RNN sequence decoding backend.

The Chordino-like feature extraction process is the same as described in the previous section. Figure 5 shows the graphical model of the BLSTM-RNN, which has a forward and a backward hidden layer both with 800 LSTM units. The output layer is a 277-way softmax layer, corresponding to a normalized probabilities of 277 labels in SeventhsBass. The input layer has 24 nodes with continuous values, corresponding to an input chroma.

4.1 Training

To generate training data, all raw audios are transformed to chromagrams. The original segment-wise ground truth annotations are upsampled to become frame-wise annotations that have 1-to-1 mappings to their input representations. Due to the absence of phase information in chromagram, all data can be circularly transposed to all 12 keys for 12 times data augmentation. A random split of 80% of them are used as training set, and the other split as validation set. During each iteration, one random training case is fed into an Adadelta optimizer to update the network

[2] github.com/tangkk/tangkkace

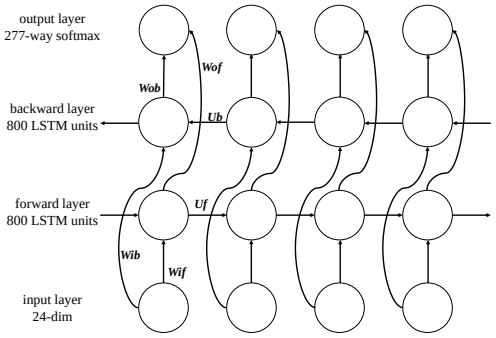

Figure 5. The BLSTM-RNN. Both the forward and backward hidden layers contain 800 LSTM units

connections. Each training case contains 500 frames of audio content with ground truth labels. The training is regularized by dropout with 0.5 probability, and further regularized by early stopping when there is no improvement after 10 validations, with 1000 iterations per validation. The BLSTM-RNN is implemented and trained under the framework of Theano [3] . The model with the best validation score will be saved for testing.

5. DATASETS

For training/validation, we use five datasets of 366 tracks in total. They contain both eastern and western pop/rock songs. They are: 1, JayChou29 (J) dataset [1]; 2, a Chinese pop song dataset (CNPop20, or C) [4] ; 3, Carole King + Queen dataset (KingQueen26, or K) [5] ; 4, 191 songs from USPop dataset (U) [6] . 5, 100 songs from RWC (R) dataset [7] .

Each track is extracted as tuned notegram and chromagram, which will be transposed to all 12 keys by pitch shifting (and zero padding in tuned notegram's case). Adjusting the ground truth labels accordingly, this results in a 12-time augmentation to the training data, which helps to avoid over-fitting.

[3] http://deeplearning.net/software/theano/

[4] containing 20 songs from both male and female singer-songwriters from Chinese cultural backgrounds

[5] http://isophonics.net/datasets

[6] https://github.com/tmc323/Chord-Annotations

[7] https://staff.aist.go.jp/m.goto/RWC-MDB/

6. SUBMISSIONS

There are totally 4 submissions:

- DK1 - System I - A hybrid Chordino-DBN approach, with two hidden layers, each of 800 units, and 6-frame segment tiling. The system is trained with UR, supporting SeventhsBass
- DK2 - System I - A hybrid Chordino-BLSTM-RNN approach, with a forward and a backward LSTM hidden layers, each with 800 LSTM units, and 6-frame segment tiling. The system is trained with UR, supporting SeventhsBass
- DK3 - System I - A hybrid Chordino-DBN approach, with two hidden layers, each of 800 units, and 6-frame segment tiling. The system is trained with UR (mapped to MajMin), supporting MajMin.
- DK4 - System II - A BLSTM-RNN end-to-end approach. The system is trained with CJKU, supporting SeventhsBass.

7. REFERENCES

[1] Junqi Deng and Yu-Kwong Kwok. Automatic chord estimation on seventhsbass chord vocabulary using deep neural network. In *Proceedings of the 41th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016). Shanghai, China*, 2016.

[2] Matthias Mauch. *Automatic chord transcription from audio using computational models of musical context*. PhD thesis, School of Electronic Engineering and Computer Science Queen Mary, University of London, 2010.