Homework #3: Automatic Polyphonic Piano Transcription

The Robotics Program
20195595
Dooyoung Hong

1. Introduction

Automatic music transcription (AMT) refers to an automated process that converts musical signals into a piano roll. Polyphonic piano transcription is a specific AMT task for piano music. Because of the sequential aspect of piano transcription, the recurrent neural network (RNN) module is commonly used for the task. However, the problem of AMT is already notoriously difficult even for humans because of the polyphonic nature of piano music. The transcription problem is made more harder by the fact that there is an enormous number of possible outputs. HW#3 deals with Onsets, Frames for data inputs, and various types of neural networks to perform AMT.

2. Algorithm description, Experiments, and results

LSTM (Long Short-Term Memory) is a special type of recurrent neural network (RNN) that aims to solve the forgetting of neural network. A common RNN contains loops that allow the network to better train sequential data. However, it contains problem that arises when the location gap between the sequential data. An LSTM solves this by essentially allowing the neural network to learn what to keep and throw away. Namely, LSTM plays a role in ensuring that the network retains the necessary information for a long time by appropriately adjusting the information to be discarded and the information to be transmitted.

Layer	Specification	Output shape
LogMel	model.LogMelSpectrogram	`(Time, 229)`
LSTM	2 layer Bi-directional LSTM. 88 unit for each direction.	`(Time, 88*2)`
Output FC	88 unit, linear	`(Time, 88)`

Figure 1. Structure of LSTM-based model.

Question 1 requires implementing LSTM-based model for performing the automatic polyphonic piano transcription. The model uses mel spectrogram normalized through log function and then it uses BiLSTM that runs in both directions, meaning the algorithm is fed the data both front-to-back and back-to-front, which decrease training time of model. The input data replace loss function of Frame and Onset by the fully connected layer after LSTM layer. Figure.2 shows the performance of LSTM-based model. It uses the performance matric of F1-Score.

F1 Score is a ratio average of Precision@K and Recall@K. It is mainly used when data imbalance between classification classes is severe. The machine learning performance cannot be properly represented when if the data classification class is not uniform. So, F1 Score are derived by the harmonic average of precision and recall. The model has the higher value, the better the model. LSTM-based model has about 0.46 F1 score. I will compare it with other models.

```
100%|██████████| 10000/10000 [1:27:07<00:00,  1.91it/s, loss: 1.208e-01]
Loading 1 group(s) of MAESTRO_small at data
Loading group test: 100%|██████████| 50/50 [00:12<00:00,  4.10it/s]

metric/loss/frame_loss : 0.12271907180547714
metric/loss/onset_loss : 0.07273221760988235
metric/frame/frame_f1 : 0.4714975367603482
metric/frame/onset_f1 : 0.4265495107875212
metric/note/f1 : 0.49629181354077956
metric/note-with-offsets/f1 : 0.1603866108404027
                            loss frame_loss                 : 0.123 +- 0.057
                            loss onset_loss                 : 0.073 +- 0.029
                           frame frame_precision            : 0.735 +- 0.085
                           frame frame_recall               : 0.352 +- 0.069
                           frame frame_f1                   : 0.471 +- 0.072
                           frame onset_precision            : 0.730 +- 0.034
                           frame onset_recall               : 0.320 +- 0.147
                           frame onset_f1                   : 0.427 +- 0.135
                            note precision                  : 0.930 +- 0.024
                            note recall                     : 0.355 +- 0.155
                            note f1                         : 0.496 +- 0.154
                            note overlap                    : 0.422 +- 0.057
               note-with-offsets precision                  : 0.294 +- 0.122
               note-with-offsets recall                     : 0.115 +- 0.077
               note-with-offsets f1                         : 0.160 +- 0.094
               note-with-offsets overlap                    : 0.813 +- 0.085
```

Figure 2. Result of LSTM-based model.

Question 2 requires sequential model with Convolutional neural networks (CNN). CNN is a special type of neural network that uses alternating convolutional and polling layers to gain compress information for recognition, and they are primarily used for image classification.

Layer	Specification	Output shape
LogMel	model.LogMelSpectrogram	`(Time, 229)`
ConvStack	model.ConvStack	`(Time, fc_unit)`
LSTM	2 layer Bi-directional LSTM. 88 unit for each direction.	`(Time, 88*2)`
Output FC	88 unit, linear	`(Time, 88)`

Figure 3. Structure of CRNN model.

CNN commonly takes input data and analyze certain sections of that at a time. It is essentially to make multi-dimensional data like images or spectrograms easier to handle while still preserving the important features. CNN works by alternating convolution and pooling layers. Convolution layers analyze little parts of whole data at a time and combine the feature of data. The pooling layers aim to further reduce the spatial size of the data by only keeping dominant feature.

The model also uses mel spectrogram normalized through log function and BiLSTM. Instead, it uses convolutional layer before LSTM layer. Convolutional filters have a positive effect on the model by consolidating information between frames in the spectrogram. Figure.4 shows the performance of CRNN model. It shows a significant performance improvement compared with the LSTM-based model.

```
100%|██████████| 10000/10000 [16:42:24<00:00,  6.01s/it, loss: 1.058e-01]
Loading 1 group(s) of MAESTRO_small at data
Loading group test: 100%|██████████| 50/50 [00:12<00:00,  4.04it/s]

metric/loss/frame_loss : 0.11014062911272049
metric/loss/onset_loss : 0.10949159413576126
metric/frame/frame_f1 : 0.5853393938393112
metric/frame/onset_f1 : 0.6804186048498932
metric/note/f1 : 0.7726592543807013
metric/note-with-offsets/f1 : 0.35514066011837786
                            loss frame_loss                : 0.110 +- 0.045
                            loss onset_loss                : 0.109 +- 0.056
                           frame frame_precision           : 0.671 +- 0.058
                           frame frame_recall              : 0.523 +- 0.081
                           frame frame_f1                  : 0.585 +- 0.065
                           frame onset_precision           : 0.811 +- 0.037
                           frame onset_recall              : 0.595 +- 0.129
                           frame onset_f1                  : 0.680 +- 0.096
                            note precision                 : 0.966 +- 0.016
                            note recall                    : 0.653 +- 0.131
                            note f1                        : 0.773 +- 0.098
                            note overlap                   : 0.517 +- 0.053
               note-with-offsets precision                 : 0.444 +- 0.104
               note-with-offsets recall                    : 0.300 +- 0.096
               note-with-offsets f1                        : 0.355 +- 0.098
               note-with-offsets overlap                   : 0.856 +- 0.072
```

Figure 4. Result of CRNN model.

Question 3 is implementing Onsets-and-Frames model. This model uses an inter-connection between the onsets and frames. This is a method of using the logit information of Onsets before going through the sigmoid function as basic information to derive the frame logits.

Separating the task of transcription into note onset and framewise activation prioritizes more important musical moments, ensuring a musically useful transcription. This model directly links the derived relationship of onset and frame to give certainty to the model's putative relationship.

```
100%|██████████| 10000/10000 [17:07:35<00:00,  6.17s/it, loss: 9.716e-02]
Loading group test:   4%|▍         | 2/50 [00:00<00:05,  8.81it/s]
metric/loss/frame_loss      : 0.0873
metric/loss/onset_loss      : 0.0819
metric/frame/frame_f1       : 0.5265
metric/frame/onset_f1       : 0.7189
metric/note/f1              : 0.8319
metric/note-with-offsets/f1 : 0.2844
Loading 1 group(s) of MAESTRO_small at data
Loading group test: 100%|██████████| 50/50 [00:11<00:00,  4.47it/s]

metric/loss/frame_loss : 0.13314303755760193
metric/loss/onset_loss : 0.12594786286354065
metric/frame/frame_f1 : 0.5006160852850676
metric/frame/onset_f1 : 0.7016586762366096
metric/note/f1 : 0.8004433032357353
metric/note-with-offsets/f1 : 0.3204597846678233
                         loss frame_loss                  : 0.133 +- 0.052
                         loss onset_loss                  : 0.126 +- 0.067
                        frame frame_precision             : 0.547 +- 0.073
                        frame frame_recall                : 0.471 +- 0.102
                        frame frame_f1                    : 0.501 +- 0.079
                        frame onset_precision             : 0.806 +- 0.035
                        frame onset_recall                : 0.629 +- 0.123
                        frame onset_f1                    : 0.702 +- 0.088
                         note precision                   : 0.970 +- 0.016
                         note recall                      : 0.690 +- 0.124
                         note f1                          : 0.800 +- 0.090
                         note overlap                     : 0.458 +- 0.076
            note-with-offsets precision                   : 0.389 +- 0.122
            note-with-offsets recall                      : 0.276 +- 0.104
            note-with-offsets f1                          : 0.320 +- 0.111
            note-with-offsets overlap                     : 0.845 +- 0.076
```

Figure 5. Result of Onsets-and-Frames model.

Figure 5 shows the performance of Onsets-and-Frames model. This model does show some performance improvement over the CRNN model although there is no such dramatic performance improvement between the LSTM-based model and CRNN model.

Question 4 requires the visualization of model prediction results as the piano roll format. Figure.6 is the sample dataset of MAESTRO (it presented dataset.py). It shows audio sample and its frames and onsets. Finally, figure.7 shows the model prediction results of AMT.

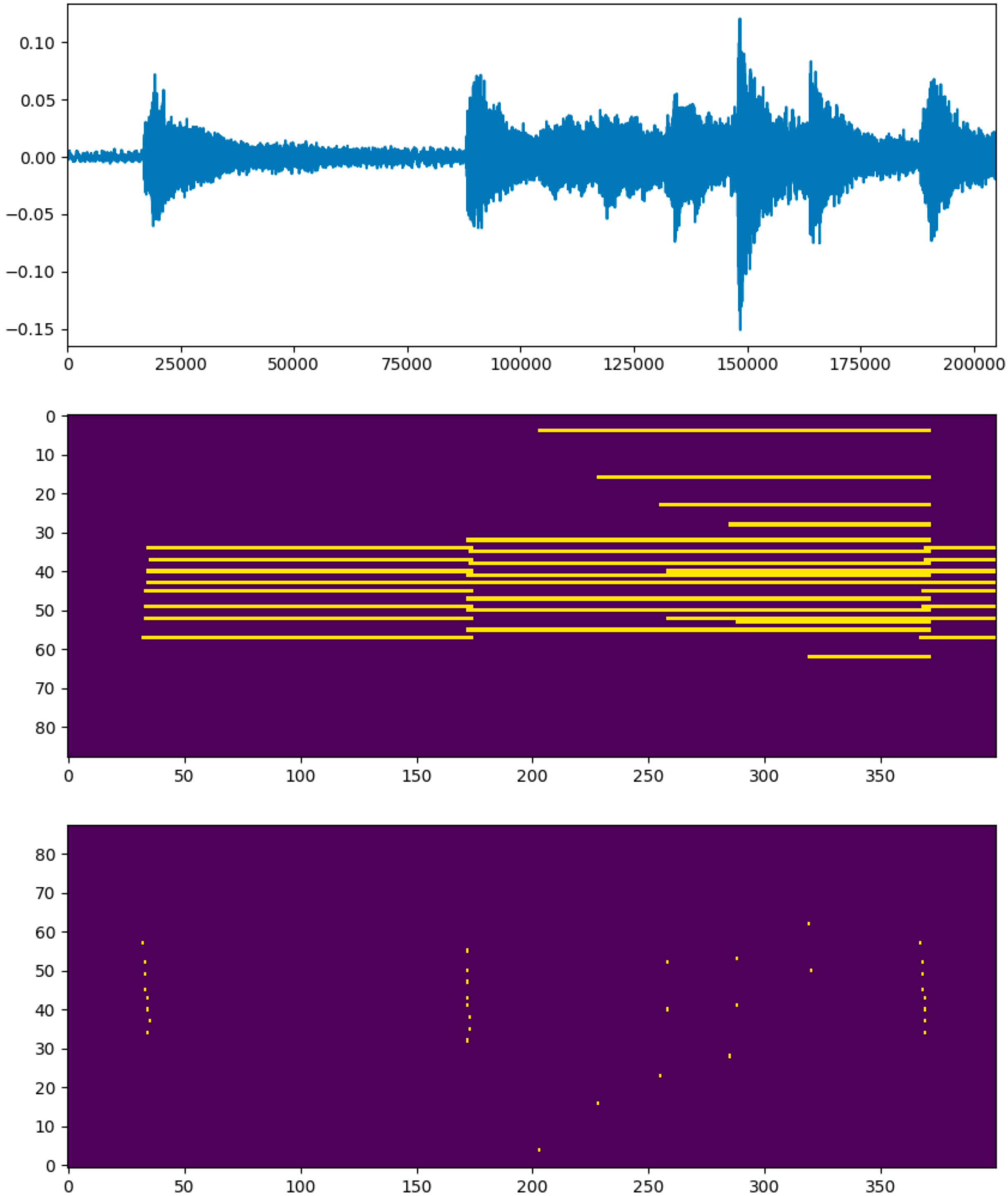

Figure 6. Sample audio and its frames and onsets.

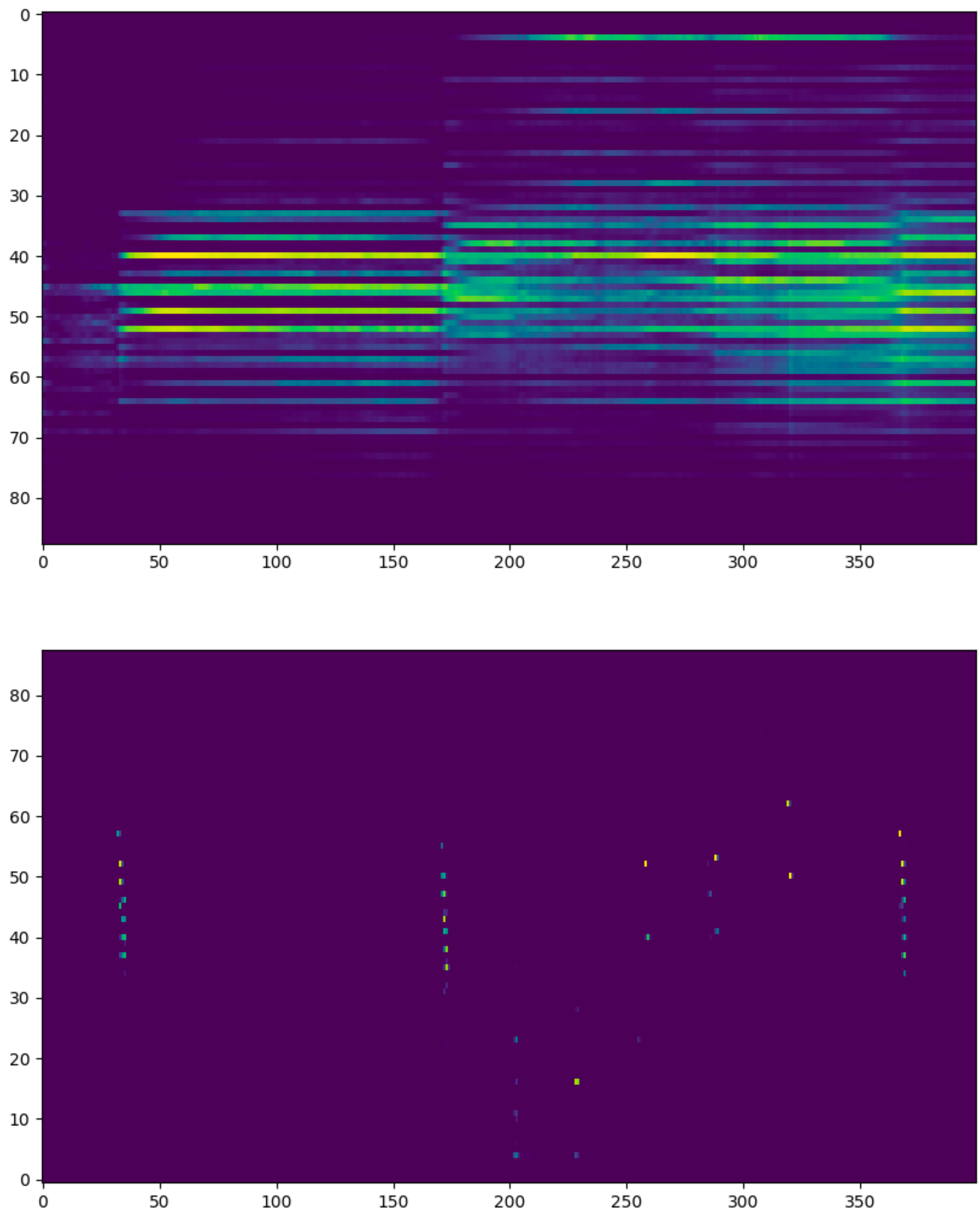

Figure 7. Prediction of frames and onsets by ONF model.

Both frame and onset show visibility comparable to the naked eye compared with the basic data. However, in the frame the noise that occurs between predictions of each tone is clearly recognized. It through that approximation of the model affects the accuracy when the model predicts the variability of the input signal for the computer evaluate all sounds.

3. Conclusion

Homework #3 performs the automatic music transcription using machine learning techniques. It contains convolutional filter and sequential network for analyzing the overlap of harmonics in the acoustic signal. AMT is already notoriously difficult even for humans due to the polyphonic sound of piano. However, Onsets and Frames can convert piano recordings into a MIDI sequence. It uses a system similar to speech recognition in that it uses acoustic models in conjunction with a music language model and show the higher classification performance.