Homework #3: Automatic Polyphonic Piano Transcription

1. Experiments and Results

1.1 Overall Experiment Results

Table 1 is the averaged results of baseline, Q1, Q2, and Q3 with default train configuration (batch size 32, training iterations 10K, learning rate 6e-4). All results are reported in **Appendix 1**. Overall, Baseline has highest precision, recall, and F1 score for frame, Note Onset, and Note with Offset condition.

Model	Frame			Note Onset			Note with Offset		
	precision	recall	F1	precision	recall	F1	precision	Recall	F1
CNN(Baseline)	**0.789**	**0.594**	**0.666**	**0.987**	**0.722**	**0.828**	**0.492**	**0.363**	**0.415**
Bi-LSTM(Q1)	0.706	0.405	0.508	0.945	0.342	0.485	0.315	0.118	0.167
CRNN(Q2)	0.657	0.512	0.571	0.968	0.689	0.799	0.425	0.304	0.352
ONF(Q3)	0.611	0.415	0.484	0.963	0.666	0.781	0.386	0.267	0.313
Uni-LSTM(Q4)	0.654	0.423	0.508	0.949	0.384	0.529	0.292	0.120	0.165

Table 1 Averaged Results of Models. All results are averaged with 3 independent runs.

1.2 Q4 : Uni-directional LSTM vs Bi-directional LSTM

Since the main task of this HW3 is experiencing and understanding the behavior of RNN models by reimplementing RNN-based transcription models, and I was wondering why we are using the Bi-directional LSTM rather than just using general Uni-directional LSTM for this task, and wanted to see whether there is a performance improvement, by using the Bi-directional information. Therefore, additional experiment was conducted with using unidirectional LSTM. To make fair comparison with 2-layer Bi-directional LSTM with 88 units, 2-layer Uni-directional LSTM with 176(2x88) unit was used, and compared with the result of RNN. The input size of final FC layer was adjusted for Uni-directional LSTM. For the result, as shown in **Table 1**, Bi-LSTM achieve 5% higher frame precision than Uni-LSTM.

2. Discussion

2.1 Limitation of this Study

Before moving onto discussing about the result, it will be needed to say it was hard to compare the performance of each model, because the model's result was not converged yet, and the size of our dataset was small, since we used the subset of the MAESTRO dataset (170 performances from original dataset with 1184 performances). Therefore, our discussion is mainly comparing the overall tendency of each model, considering the model's theoretical design, not only just using the final result.

2.2 CNN vs RNN(Bi-directional LSTM)

From the results from **Table 1**, we could see different result patterns of RNN and CNN. First observation is CNN resulted higher precision and recall value for all Frame, Note Onset, and Note with Offset than RNN. It is well known that training RNN is much slower than CNN, due to its native sequential structure so that it cannot fully utilize the powerful parallel computation of GPU. Therefore, we can interpret that the result shows training RNN is slower than training CNN.

Second observation is the onset precision value is bigger than Frame precision value for RNN. Since the RNN is designed to capture the temporal relationship between frames, abrupt temporal changes such as onset was more correctly captured by using RNN than CNN which we can infer from a high precision value of note onset in our experiment.

However, contrary to the precision value, the recall value of RNN is much lower than CNN. Although we cannot do fair comparison for these models in this HW3, one possible explanation for this situation is due to the nature of both models. while 2D CNN can capture both of temporal and harmonic features, RNN mostly focus on capturing the temporal feature. Since piano is polyphonic instrument, learning the harmonic relationship between notes can help models finding the target note of onset, which can affect the recall. To investigate this hypothesis, further study can be comparing the result of training CNN and RNN for transcribing polyphonic (e.g. considering total performance) vs. monophonic (e.g. considering only melody part) piano performance.

2.3 RNN vs CRNN

During training, CNN had better performance than question 1, 2, and 3, even though it was baseline. As discussed in *2.2*, CNN can capture both of the temporal and harmonic features, and RNN can capture mainly on temporal feature so that CNN can get better prediction than RNN. Therefore, when we view CRNN as added CNN to RNN, the performance of RNN was improved, by adding the good feature extractor CNN. However, when we view CRNN as added RNN to CNN, the accuracy was not significantly improved until 10k training, due to the slow training step of RNN.

2.4 CRNN vs ONF

By comparing the result of CRNN and ONF, we can investigate the effect of inter-connection of onset prediction and frame prediction. However, for our experiment, it was hard to find any advantage of using inter-connection between CRNN and ONF. The frame F1 score of ONF was even worse than the CRNN. We can interpret that the note onset information might not be helpful for frame prediction, or it is a matter of training time, since the model was not converged yet, or it is due to the small dataset size. To check this, further investigation can be training the transcription model with larger dataset and long training time.

2.5 *Q4 : Uni-directional LSTM vs Bi-directional LSTM*

The performance of Bi-LSTM and Uni-LSTM was similar, except Bi-LSTM achieve 5% higher frame precision than Uni-LSTM. From that, we can infer that the opposite direction information influenced on the Frame prediction accuracy, and not on the Onset prediction.

2.6 Frame Offset Prediction

While doing HW3, we could see that the onset prediction accuracies of models were higher than the frame prediction. As mentioned in *2.2*, whereas onset of a note is clear pattern that involves the abrupt temporal change, finding the offset of the note is more difficult, which we can see from **Fig 1**. Since even the Mel spectrogram, the input of the model, doesn't have clear offset of the note, it could be possible to add more spectral features such as delta MFCC or double delta MFCC to increase the offset prediction.

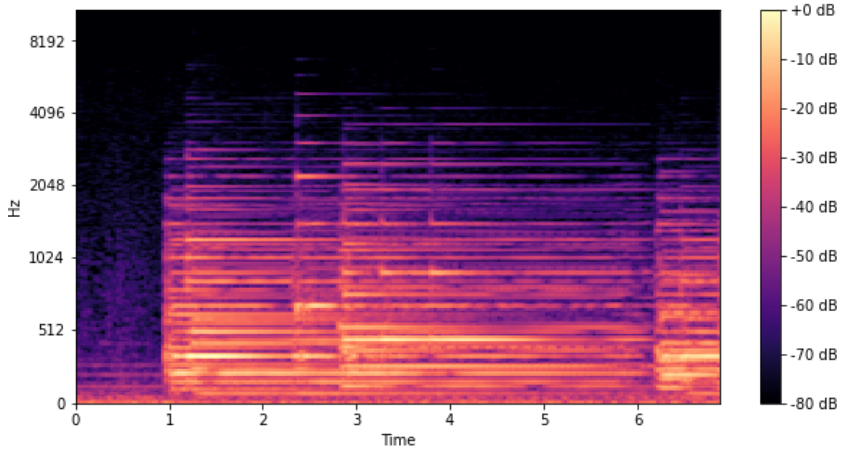

Fig 1 Mel-spectrogram of first 7 secs of random sample from dataset

(Same FFT, HOP, MEL bin size with training)

3. Conclusion

In this HW3, by reimplementing the RNN-based transcription models, we could more understand about the behavior of RNN. In summary, model with 3 stacked conv layers achieve the highest performance precision, recall and F1 for frame, Note Onset and Note Offset. Also, we could see that the frame prediction is more difficult than onset prediction.

4. Appendix

CNN(Baseline)	Frame			Note Onset			Note with Offset		
	precision	recall	F1	precision	recall	F1	precision	Recall	F1
1	0.753	0.649	0.692	0.984	0.743	0.842	0.501	0.380	0.430
2	0.764	0.647	0.695	0.991	0.694	0.810	0.518	0.366	0.425
3	0.851	0.486	0.612	0.985	0.729	0.833	0.457	0.343	0.389
Average	0.789	0.594	0.666	0.987	0.722	0.828	0.492	0.363	0.415

Bi-LSTM(Q1)	Frame			Note Onset			Note with Offset		
	precision	recall	F1	precision	recall	F1	precision	Recall	F1
1	0.730	0.370	0.484	0.952	0.328	0.471	0.324	0.117	0.167
2	0.690	0.413	0.511	0.933	0.373	0.517	0.300	0.123	0.170
3	0.697	0.431	0.529	0.950	0.325	0.468	0.322	0.115	0.164
Average	0.706	0.405	0.508	0.945	0.342	0.485	0.315	0.118	0.167

CRNN(Q2)	Frame			Note Onset			Note with Offset		
	precision	recall	F1	precision	recall	F1	precision	Recall	F1
1	0.630	0.536	0.576	0.972	0.691	0.802	0.429	0.305	0.354
2	0.689	0.497	0.573	0.969	0.678	0.792	0.431	0.303	0.353
3	0.651	0.504	0.565	0.962	0.698	0.803	0.416	0.303	0.348
Average	0.657	0.512	0.571	0.968	0.689	0.799	0.425	0.304	0.352

ONF(Q3)	Frame			Note Onset			Note with Offset		
	precision	recall	F1	precision	recall	F1	precision	Recall	F1
1	0.603	0.492	0.537	0.969	0.644	0.766	0.424	0.282	0.335
2	0.650	0.343	0.443	0.961	0.673	0.785	0.360	0.252	0.294
3	0.581	0.410	0.473	0.959	0.682	0.791	0.375	0.267	0.310
Average	0.611	0.415	0.484	0.963	0.666	0.781	0.386	0.267	0.313

Uni-LSTM(Q4)	Frame			Note Onset			Note with Offset		
	precision	recall	F1	precision	recall	F1	precision	Recall	F1
1	0.656	0.415	0.502	0.951	0.377	0.523	0.294	0.119	0.165
2	0.669	0.427	0.516	0.946	0.398	0.543	0.288	0.124	0.168
3	0.638	0.426	0.505	0.951	0.377	0.522	0.293	0.118	0.163
Average	0.654	0.423	0.508	0.949	0.384	0.529	0.292	0.120	0.165

Appendix 2 All Results of Models