Piano Music Transcription Using Neural Networks

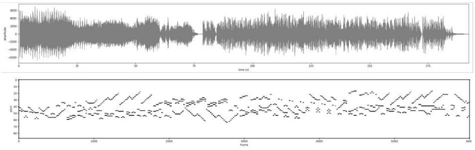
Zehui Wang (wzehui@Stanford.edu)

Stanford University | Stanford ENGINEERING Computer Science

Introduction

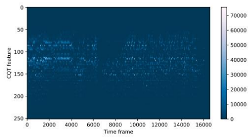
We aim to solve piano music transcription problem. Our model takes an input of a WAV file and translates into a MIDI file that contains information of the duration and the pitch of each note. We preprocess the audio file with CQT transform and then use a CNN architecture to predict the music notes.

Dataset



Audio files: WAV files of piano music
Label files: the aligned MIDI files (duration and pitch)
Training vs. dev vs. test: 19.37h, 5.01h, 4.04h

Data preprocessing: Constant Q transform (7 octaves, 36 bins, hope size of 512, 252 dim features)



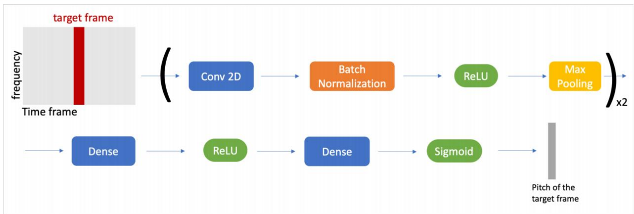
Output: multi-hot binary vector of length 88 (88 keys on keyboard)
Postprocessing: convert numpy arrays to MIDI

Method

Baseline: a deep neural network proposed in [1]. Input (252) – [dense (256) – ReLU] * 3 – dense (88) – sigmoid

CNN: takes an input of a context window of frames, of which the center is the target frame. Zero paddings are used in the beginning and the end of the input.

Model



Loss function: mean squared error (MSE) & binary cross entropy (average over all classes)

$$Loss_{CE} = \frac{1}{n}\sum_{i=1}^{n}(-y_i \log \hat{\boldsymbol{y}}_i - (1-y_i)\log(1-\hat{\boldsymbol{y}}_i))$$

Model parameters: Context window size: 7; Conv2D kernel size: (25, 5) and (5,3); max pooling size: (3,1); dropout 0.5 in all layers; L2 0.0001 in dense layers; Adam optimizer learning rate 0.0001

Experimental Results

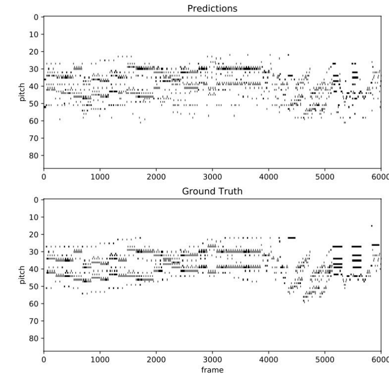
Model	Loss Function	Parameter Searching*	F-measure	Accuracy
Baseline	MSE		0.6393	0.4698
Baseline	MSE	dropout 0.3	0.6398	0.4703
Baseline	MSE	Hidden units 125	0.6243	0.4538
CNN	**MSE**		**0.6527**	**0.4850**
CNN	MSE	Dropout 0.2	0.3812	0.2355
CNN	MSE	Learning rate 0.001	0	-
CNN	MSE	L2 = 0.00005	0.6368	0.4671
CNN	MSE	L2 = 0	0.4172	0.2636
CNN	MSE	Window size = 9	0.5964	0.4249
CNN	CE		0.6328	0.4628
CNN	CE	L2 = 0.00005	0.5653	0.3940
CNN	CE	L2 = 0	0	-

*CNN Parameters listed in this column are compared to those described in Section Model; baseline parameters are compared to the model in [1].

Evaluation Metrics: Precision(P) = TP/(TP+FP), Recall(R)=TP/(TP+FN), F-measure(F)=2PR/(P+R). TP,FP,FN are computed at each time frame. The average is then computed across the entire dataset. Evaluation methods are adopted from [2].

Analysis

Regularization: a small value of L2 regularization can greatly reduce overfitting; cross entropy loss is more sensible than MSE in terms of L2 parameters.
Error analysis: higher error rate when predicting notes that last a long time. The problem might be that long notes appear less often.
Data mismatch: overfitting issue (F-measure of baseline on training data is 0.8656). Use real piano music as dev set and test set, and synthesizers as training set.



Future Work

- Fine tuning CNN architectures and parameters
- Explore RNN structures and compare with CNN
- Diving into data mismatch problem

Reference

[1] Diego Gonzalez Morin. Deep neural networks for piano music transcription. https://github.com/diegomorin8/Deep-Neural-Networks-for-Piano-Music-Transcription

[2] Sigtia, Siddharth, Emmanouil Benetos, and Simon Dixon. "An end-to-end neural network for polyphonic piano music transcription." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24.5 (2016): 927-939.

Stanford CS230, Final Project Poster, Winter 2019