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 numpys 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)

\[
\text{Loss}_{\text{CE}} = \frac{1}{N} \sum_{i=1}^{N} (-y_i \log y_i - (1 - y_i) \log(1 - y_i))
\]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Loss Function</th>
<th>Parameter Searching*</th>
<th>F-measure</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>MSE</td>
<td></td>
<td>0.6363</td>
<td>0.4968</td>
</tr>
<tr>
<td>Baseline</td>
<td>MSE</td>
<td>dropout 0.3</td>
<td>0.6398</td>
<td>0.4703</td>
</tr>
<tr>
<td>Baseline</td>
<td>MSE</td>
<td>hidden units 125</td>
<td>0.6243</td>
<td>0.4538</td>
</tr>
<tr>
<td>CNN</td>
<td>MSE</td>
<td>dropout 0.2</td>
<td>0.3812</td>
<td>0.2355</td>
</tr>
<tr>
<td>CNN</td>
<td>MSE</td>
<td>learning rate 0.0001</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>CNN</td>
<td>MSE</td>
<td>L2 = 0.00005</td>
<td>0.6368</td>
<td>0.4671</td>
</tr>
<tr>
<td>CNN</td>
<td>MSE</td>
<td>L2 = 0</td>
<td>0.4172</td>
<td>0.2636</td>
</tr>
<tr>
<td>CNN</td>
<td>MSE</td>
<td>window size = 9</td>
<td>0.5964</td>
<td>0.4249</td>
</tr>
<tr>
<td>CNN</td>
<td>CE</td>
<td></td>
<td>0.6326</td>
<td>0.4628</td>
</tr>
<tr>
<td>CNN</td>
<td>CE</td>
<td>L2 = 0.00005</td>
<td>0.5653</td>
<td>0.3940</td>
</tr>
<tr>
<td>CNN</td>
<td>CE</td>
<td>L2 = 0</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

*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