Harmonizing with Piano Genie

Craig Chan, Matthew Stein, Divya Gupta
{ckc1, klezmer, dg524} @ stanford.edu

Introduction
- Piano Genie: Google Magenta project trying to make music composition more accessible via a simple interface translating high-level musical gestures to musical notes.
- User improvises sequences on an 8-button input device which is decoded into realistic 88-key piano music in real time.
- Uses MIDI pitch and tempo features, yielding convincing melodic contours but lacking cohesive musical structure.
- Project goal: extend Piano Genie capabilities by adding chord roots/types features to produce more melodic performances.

Model
- Autoencoder setup for unsupervised learning task (learn mappings between button contours and melodies, ground truth pairings do not exist).
- Bidirectional LSTM encoder learns the mapping of 88-key piano sequences to 8-button sequences using integer quantization autoencoding (IQAE).
- Unidirectional LSTM decoder learns to map button contours back to note sequence melodies.

Dataset
- MAESTRO Dataset: > 172 hours of e-piano performance MIDI recordings of 17th - 20th century classical music.
- Real human performance captures nuanced timing, leading to generative output which more closely mimics humans.
- Chords represented by (root, quality), where root = [C-0, C#: 1, ..., B: 11] and quality = [0: major, 1: minor, 2: aug. 3: dim].
- Root and quality are separate features.
- More complex or unknown chords passed over.
- Chord annotation algorithm: infer chords based on what notes are playing at a given time, accept only "simple" chords, otherwise use last seen valid chord.
- Chords quality tuned manually with two parameters:
  - min_notes_per_chord: min # of simultaneous notes required to attempt to infer a chord.
  - max_repeated_chords: how many consecutive times an invalid/unknown chord can take on the value of the last seen valid chord.
- Data augmentation:
  - Time stretch augmentation: scale all note durations by a factor of 0.95 to 1.05.
  - Random subsequence sampling.

Results
- Perplexity: cross-entropy measurement between the original sequence and model's predicted sequence (PPL = \( e^{\text{trained loss}} \)).
- Contour violation ratio (CVR): proportion of time steps where sign of note interval does not match sign of button interval.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>PPL</th>
<th>CVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>Baseline</td>
<td>2.445</td>
<td>2.4603E-03</td>
</tr>
<tr>
<td></td>
<td>Modified</td>
<td>2.64</td>
<td>3.4449E-03</td>
</tr>
<tr>
<td>Test</td>
<td>Baseline</td>
<td>3.216</td>
<td>1.2303E-03</td>
</tr>
<tr>
<td></td>
<td>Modified</td>
<td>3.385</td>
<td>4.9213E-04</td>
</tr>
</tbody>
</table>

Conclusions
- Hypothesis: chord model would lower perplexity, neutral effect on contour violation ratio.
- Result: no significant difference in contour violation or perplexity scores between our baseline and modified models.
- Perplexity results for both the baseline and modified model were higher on the test set on train set, which could suggest a variance problem (overfitting) – use larger dataset + data augmentation to help reduce variance.

Future Work
- Use chord model in Piano Genie interactive web demo.
- Pitch augmentation: transpose each piece into every key.
- Improve chord annotation: human annotated, ML signal processing, beat annotation (where to place chords).
- Use user-generated chord model output as training data for other models, such as trading 4’s over a chord progression.

References