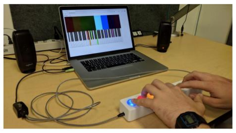
Harmonizing with Piano Genie

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



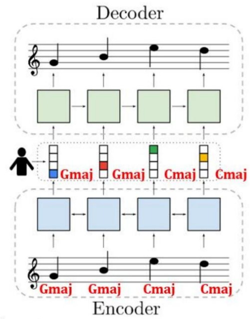
- Uses MIDI pitch and tempo features, yielding convincing melodic contours but lacking cohesive musical structure
- Project goal: extend Piano Genie capabilities by adding chord roots/types features to produce more melodic performances

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

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



- 2-layer LSTM RNNs with 128 units each
- $x_{feature} = [x_{pitch}, x_{\Delta T}, x_{chord\ root}, x_{chord\ quality}]$
  - $x_{\Delta T}$ represents start and end times of notes
  - $x_{chord\ root}, x_{chord\ quality}$ are one-hot encoded vectors
- Loss functions
  - L_{recons}: decoder reconstruction loss (average negative log likelihood of obtaining correct note sequence given encoding)
  - L_{margin}: discourages encoder from producing values outside an interval specified in discretization strategy.
  - $L_{contour}$: regularization term aligning direction of key, button contours

$$L = L_{recons} + L_{margin} + L_{contour}$$
$$L_{recons} = -\log P_{dec}(x \mid enc(x))$$
$$L_{margin} = \max(|enc_s(x)| - 1, 0)^2$$
$$L_{contour} = \max(1 - \Delta x \Delta enc_s(x), 0)^2$$

- Maintained many hyperparameter values from baseline Piano Genie for fair evaluation between baseline and modified models (used mini-batch size of 32 examples, note sequence length of 128 notes, and Adam optimizer with learning rate of 0.0003)

Results

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

Dataset	Model	PPL	CVR
Train	Baseline	2.445	2.4603E-03
	Modified	2.64	3.4449E-03
Test	Baseline	3.216	1.2303E-03
	Modified	3.385	4.9213E-04

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

Chris Donahue, Ian Simon, and Sander Dieleman. 2018. Piano Genie. arXiv:1810.05246v1.