Infilling Piano Performances

Daphne Ippolito dei@google.com
Anna Huang annahuang@google.com
Curtis Hawthorne fjord@google.com
Douglas Eck deck@google.com

Abstract

Existing systems for music generation have generated music in a left-to-right direction or have used a fill-in-the-blank approach on a quantized piano-roll musical representation. In this work, we show that it is possible to train a self-attention based Transformer to infill deleted sections of MIDI transcriptions of performed piano music. This infilling technique can be used collaboratively by composers to select contiguous sections of their work to be 'rewritten' by a neural net. It can also be used to gradually morph one musical piece into a different one.

1 Introduction

Generative models for music usually focus on predicting the future from the past (Oore et al., 2018). However, if both the future and past are known, prediction becomes much easier. CoCoNet tackles this problem by training a neural network to make predictions for masked positions on a piano roll (Huang et al., 2017). Inspired by CoCoNet, we extend the music infilling task to virtuosic piano performances by taking advantage of the recently-introduced NoteTuple representation (Hawthorne et al., 2018). We introduce a novel approach to collaborative music generation, where musicians can select a contiguous section of their performance to be 'rewritten' by a neural network. We also show how repeated iterations of Gibbs sampling can gradually transform a MIDI piano performance into a new one with similar phrasal structure to the original. Audio samples can be found at `https://goo.gl/magenta/performance-infilling-examples`.

2 Approach

Data Representation In CoCoNet's infilling task, music is represented as a binary matrix with the x-axis corresponding to quantized timesteps and the y-axis corresponding to pitches (Huang et al., 2017). Multiple monophonic instruments are supported by adding a 'depth' dimension to the matrix. This representation led to impressive infilling results on quantized sequences, such as excerpts from Bach chorales, but it does not scale to longer, unquantized piano performances. In contrast, language models for music have been trained on unquantized, polyphonic piano music by using a MIDI-like Performance representation (Simon & Oore, 2017). Time shifts, note on/offs, and other events are flattened into a single sequence of tokens, resulting in information on a note's start time and duration potentially spanning many positions in the encoded sequence. It is not possible to mask adjacent music notes by masking contiguous sub-sequences of the encoded representation.

To solve the problems with existing representations, we turn to the NoteTuple representation introduced by Hawthorne et al. (2018) where each note is encoded in a single six-element tuple. The tuples are ordered temporally, and chords are ordered lowest pitch to highest pitch.

Model and Training To train a model that takes in a musical sequence with a random window of notes missing, and then predicts the missing note sequence given the context sequence, we use the Transformer introduced in Vaswani et al. (2017) and also Music Transformer with relative attention from Huang et al. (2018). For each sequence of note tuples $X = x_1, ..., x_n$, we predict

$$p_\theta(x_{r+1,...,r+k} | x_{r-c+1,...,r}, x_{r+k+1,...,r+k+c})$$

Preprint. Work in progress.

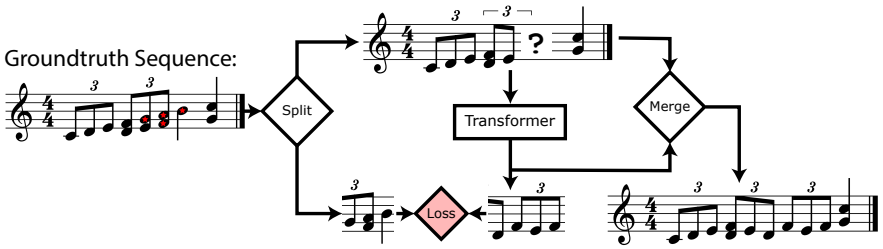

Figure 1: In the infilling task, a temporally contiguous set of notes (shown with red stars) is randomly selected to be 'rewritten' by the neural net. Although in this visual the model predicts a replacement with the same temporal length as the original, in our actual model, predictions can vary in duration.

where k is the size of the window to infill, and $r \sim \text{Uniform}[0, n-k]$. The input sequence is formed by concatenating the left context, a special separator token, then the right context. For windows on the far left ($r < c$) or far right ($r > n - c - k$), the left context is left-padded and the right context is right-padded, such that the separator token is always at the same position in the sequence. In the default Tranformer implementation, position in the sequence is encoded by adding sine and cosine functions of different frequencies to the input sequence embeddings. In our implementation, we ensure that these remain continuous across the entire sequence. That is, the positional encoding for the input sequence uses the indices $[0, ..., c, c+k+1, ..., 2c+k]$, and the encoding for the target sequence uses the indices $[c, ..., c+k]$. Experiments were conducted on two deletion window sizes: $k = 16$ and $k = 32$, with either relative or absolute attention, and with a context window of $c = 512$ notes. All models were trained on the MAESTRO dataset using the Tensor2Tensor framework with six hidden layers, hidden size 288, and filter size 2048 (Anonymous, 2018; Vaswani et al., 2018).

3 Results

Single Step Inference We conducted a listening test where participants were asked to listen to two audio snippets and then rate which was more musically pleasing. The snippets contained 96 notes, either entirely groundtruth or with the middle 32 coming from a model. For each $k = 32$ model (relative attention and absolute attention), eight participants were asked to compare 12 pairs of music snippets. The table below gives the raw counts of how many times each was rated more musical. The relative attention model was rated to be equivalent to or better than the groundtruth 66% of the time.

# Preferred	Model	Groundtruth	No preference	Validation NLL
Rel Att	36	42	30	1.45
Abs Att	25	64	19	1.42

Multi Step Inference Using Gibbs sampling, it is also possible to convert a full MIDI performance into a completely new one (Resnik & Hardisty, 2010). Given an input MIDI sequence, at every iteration, a random window of notes is selected to be replaced by predictions from the trained model. After several hundred iterations, nearly every note in the original sequence has been modified by the network, leading to a music performance that has structural resemblances to the original performance, but sounds quite different. The level of difference can be controlled through the number of iterations of Gibbs sampling that are performed. Example sequences after 512 iterations can be found in the supplemental.

4 Discussion

At $k = 16$, the model is very good at replicating the missing notes. As the window size is increased, the model deviates more and more from the original notes. Some preliminary work has been done on models trained with variable window size; more effort is necessary to work out a balance between an 'adventurous' model able to generate new material, and a 'safe' one that excels at staying true to the input. It would also be beneficial to train with extra conditioning signals, such as tempo and key signature so that the stylistic direction of the composition can be more easily controlled by the user.

References

Anonymous. Enabling factorized piano music modeling and generation with the maestro dataset. *In submission*, 2018.

Curtis Hawthorne, Cheng-Zhi Anna Huang, Daphne Ippolito, and Doug Eck. Transformer-NADE for piano performances. In *submission, NIPS Second Workshop on Machine Learning for Creativity and Design*, 2018.

Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Doug Eck. Counterpoint by convolution. In *International Conference on Music Information Retrieval*, 2017.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer, 2018.

Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. This time with feeling: Learning expressive musical performance. *arXiv preprint arXiv:1808.03715*, 2018.

Philip Resnik and Eric Hardisty. Gibbs sampling for the uninitiated. Technical report, University of Maryland, College Park, Institute for Advance Computer Studies, 2010.

Ian Simon and Sageev Oore. Performance RNN: Generating music with expressive timing and dynamics. `https://magenta.tensorflow.org/performance-rnn`, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416, 2018. URL `http://arxiv.org/abs/1803.07416`.

5 Supplemental

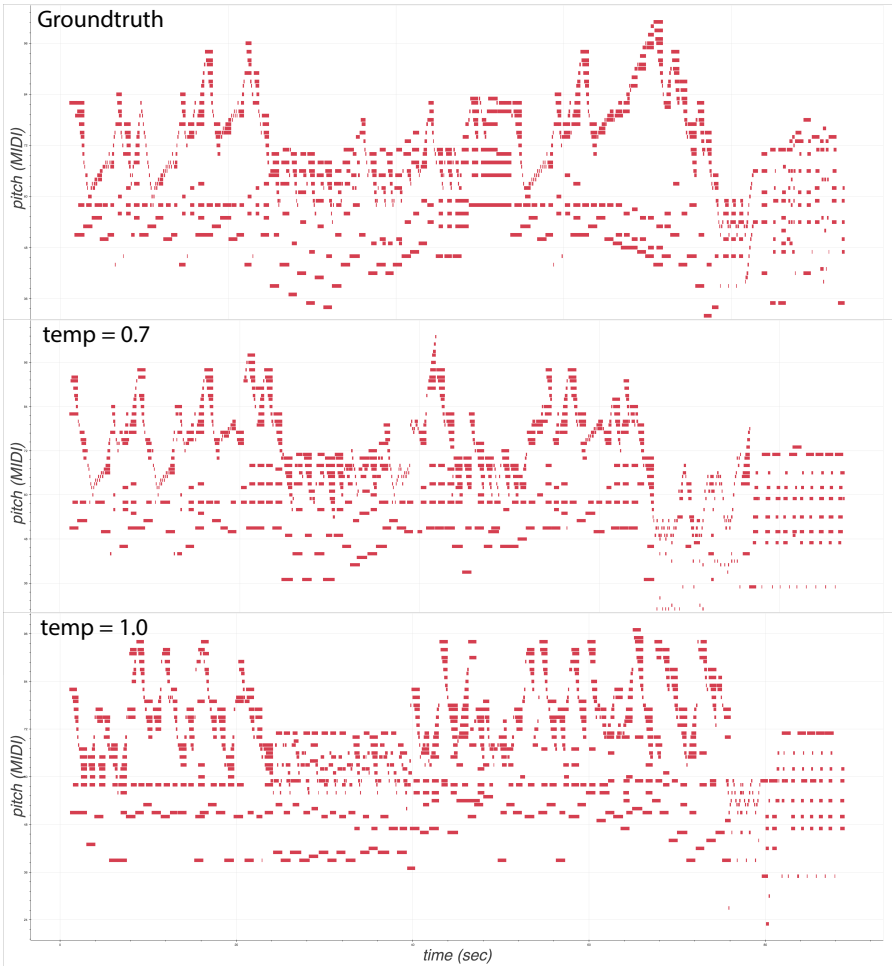

Figure 2: A groundtruth music sequence and the results of running Gibbs Sampling for 512 iterations using the relative-attention $k = 32$ model. Even after every note has been "rewritten" with high probability, the generated sequences preserve the musical structure of the original, i.e. fast runs up and down the scale, followed by a more melodic period, followed by another fast scalar period.