Transformer-NADE for Piano Performances

Curtis Hawthorne
fjord@google.com

Anna Huang
annahuang@google.com

Daphne Ippolito
dei@google.com

Douglas Eck
deck@google.com

Abstract

Self-attention based Transformers are compelling sequence models because they can capture relatively long-term dependencies by having direct access to the past. However, their memory requirements grow quadratically with sequence length, making it prohibitive to model long sequences with global attention. In contrast to previous representations of music that "flatten" a note's performance attributes such as velocity, note on, and note off into a single sequence, we reduce sequence length by proposing a new representation, NoteTuple, which groups a note's attributes as one event. This makes it natural to factorize a musical performance into a sequence of notes and model a note as a NADE on its attributes. The resulting models require fewer parameters and have faster generation. NoteTuples is a promising extension to Music Transformer (Huang et al., 2018b), enabling rich downstream tasks such as infilling of piano performances, e.g., Ippolito et al. (2018). Samples can be heard at https://goo.gl/magenta/notetuple-examples.

1 Introduction

In contrast to musical scores, MIDI recordings of piano performances include timing and dynamics. To capture this expressiveness, Oore et al. (2018) proposed a Performance representation that serializes all notes and their attributes into a sequence of events. This causes already lengthy musical pieces to be on average 3-4x longer, thus requiring >10x memory when using a Transformer model (Vaswani et al., 2017). Furthermore, a note's attributes, such as note on and note off, may be far apart in the list of events. To address these challenges, we propose a new representation, NoteTuple, which groups a note's attributes as one event. Similar motivations are seen in image modeling. For example, PixelCNN++ (Salimans et al., 2017) conditions on whole pixels rather than R/G/B sub-pixels to simplify the causal structure of the model. Our proposed representation makes it natural to factorize a piano performance into a sequence of notes and then model a note as a NADE (Larochelle & Murray, 2011) on its attributes. Our approach is inspired by Boulanger-Lewandowski et al. (2012)’s RNN-NADE where 2D pianorolls are factorized as a sequence of discretized timesteps (columns) and at each timestep any number of the notes (rows) can be turned on, with its joint probability modeled by a NADE. This formulation enables rich downstream tasks such as chunk-wise infilling of piano performances (Ippolito et al., 2018), and potentially OrderlessNADE (Uria et al., 2014, 2016) style inpainting such as in Coconet (Huang et al., 2017, 2018a).

2 NoteTuple Representation

Encoding virtuosic piano performances at a temporal resolution of 10ms, a pianoroll requires a sequence length of 100 to represent a second regardless of how many events happen, while the Performance representation only uses a token when an event occurs. The latter uses four event types to represent a single note: time shift to when it is played, velocity, pitched note on, and pitched note off. It does not explicitly represent duration and instead relies on time shifts to a note-off event, which can be far away in the sequence due to interleaving notes, as shown by the circled events in Figure 1.

*Equal contribution.
To address the challenges in existing representations, we created the NoteTuple representation, where a note's attributes are represented by a single tuple that contains the pitch, velocity, and duration of the note, as well as a time offset from the previous note in the sequence. Rather than one large shared vocabulary, each element in the note tuple has its own vocabulary. Since time shifts and duration can both be potentially very long, we avoid large vocabulary sizes for these fields by separating them into major and minor tick fields. Time shift has 13 major ticks and 77 minor ticks, representing 0 through 10 seconds. Duration has 25 major tick values and 40 minor ticks. Therefore, each note tuple contains six elements: time shift major ticks, time shift minor ticks, pitch, velocity, duration major ticks, and duration minor ticks. Each element has a vocabulary size of 100 or less. In cases where multiple notes start on the same tick (e.g., chords), the note tuples are ordered from lowest pitch to highest pitch. This can be seen in the first and second tuples in Figure 1, where the tuple for the D (pitch 62) whole note precedes the tuple for the G (pitch 67) quarter note.

3 Transformer-NADE

To save space, we refer the reader to Vaswani et al. (2017) on the Transformer architecture. We replace their output softmax with a NADE to autoregressively predict a note’s attributes $x_i$, each with a smaller softmax. $W_i$ and $c$ correspond to the hidden embedding weights and biases that project previous $x_{<i}$ into the hiddens $h_i$, and $V_i$ and $b_i$ the attribute-specific output weights and biases.

$$p(x_i|x_{<i}) = \text{softmax}(V_i h_i + b_i)$$

(1)

$$h_i = \text{sign}(W_i x_{<i} + c)$$

(2)

4 Experiments

We compare Transformer-NADE (using NoteTuple representation) to prior work on Music Transformer (using Performance representation), on the Piano-e-Competition dataset. As the representations are different, losses (in NLL) are not directly comparable, 1.43 (ours) and 1.84 (prior work). Comparing model sizes, ours need only 4 layers and 512 filters, in contrast to 6 layers and 2048 filters in prior work. For the latter, the attention mechanism operates directly on the performance representation which is a lower level sub-note level representation, which may explain why the model requires more filters. In our small listening test of 5 participants and a total of 25 pairwise comparisons, our model was rated as more preferable in 5 pairwise comparisons and the prior work as more preferable in the remaining 20. However, the differences were not always very noticeable, as indicated by several comments from participants, including, “Very hard time comparing these two, and it could go either way.”

In preliminary analysis, our model seems to struggle with note durations, causing consecutive notes to overlap and resulting in undesirable dissonances. It performs better with relatively equal durations, and is therefore better at generating pieces that are more fugue-like rather than virtuosic. Note duration is challenging to model explicitly because it involves modeling the effects of the sustain pedal, which extends the duration of all notes under the pedal until the release of the pedal. This requires learning to compute an arithmetic so that all note offs align. In the previous Performance representation this is circumvented by not explicitly representing duration and using consecutive note-off events to turn off a sustained chord at the same time. For future work, we propose using a discretized logistic mixture likelihood (Salimans et al., 2017) to smooth over the note-off alignments. We invite readers to listen to our current samples at [https://goo.gl/magenta/notetuple-examples](https://goo.gl/magenta/notetuple-examples).

---

References


