Carnegie Mellon University

Long Short-Tunes Memory

Hüsni Almoubayyed, Yizhou He
McWilliams Center for Cosmology

Peter Schaldenbrand
Human-Computer Interaction Institute

Introduction

- Previous work in generating music usually places large assumptions into the music, such as key, rhythm, or genre [1, 2], or generates music that is easily distinguishable from that made by a human.
- Representations of music regularly have extremely high dimensionality, posing a challenge for learning algorithms. Pianos have 88 keys, and raw music files are often on the order of megabytes.
- Our training dataset consisted of 100,000 MIDI files that feature wide varieties of genres, keys, key signatures, and instrumentation.
- Google Magenta [3] is a tool that performs music generation with a similar architecture to ours, with several more assumptions about the data. Our goal is to create a music generative model that learns these features: key signature, recurrent structure, tempo, chords, key/notes, and rhythm.

Method

- A note in a song has a high dependency on the history of the song. To capture this history in prediction, we chose to implement an Long Short-Term Memory (LSTM) network for music generation, with a look-back of 3 messages.
- Recent research suggests generative adversarial networks (GANs) have success in generating music, [4, 5] even using raw music files [6], however, the output is of fixed size and we wanted to produce arbitrarily long streams of music.

- We used MIDI files as our training data due to their compact representation of music. The files were converted from streams of messages into a three valued feature vectors.
- A note corresponds to a pitch on a piano, the velocity is how hard the note is struck, and Δ time is the time between the current note and the previous one. A "Δ time" of 0 would indicate that the current and previous notes are played simultaneously, i.e. a chord.

MIDI

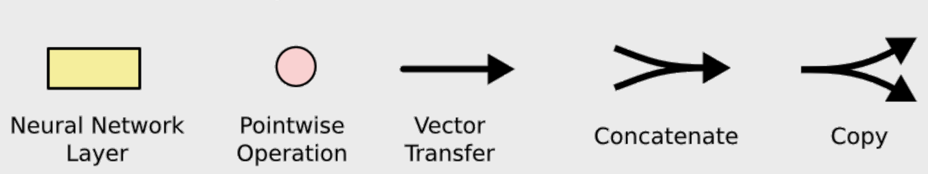

```
<message note_on channel=1 note=31 velocity=127 time=0>
<message note_on channel=0 note=67 velocity=50 time=0.13>
<message note_off channel=1 note=31 velocity=64 time=0.7>
<message note_off channel=0 note=67 velocity=100 time=0.6>
<message note_on channel=0 note=70 velocity=50 time=0.13>
<message note_on channel=0 note=65 velocity=123 time=0>
<message note_on channel=1 note=31 velocity=64 time=0>
<message note_off channel=0 note=70 velocity=64 time=0.52>
<message note_off channel=0 note=65 velocity=64 time=0.01>
<message note_off channel=1 note=31 velocity=64 time=0.03>
```


Feature Vector

	Note	Velocity	Δ Time
x_1	31	127	0
x_2	67	50	0.13
x_3	70	50	0.26
x_4	65	123	0
x_5	31	115	0

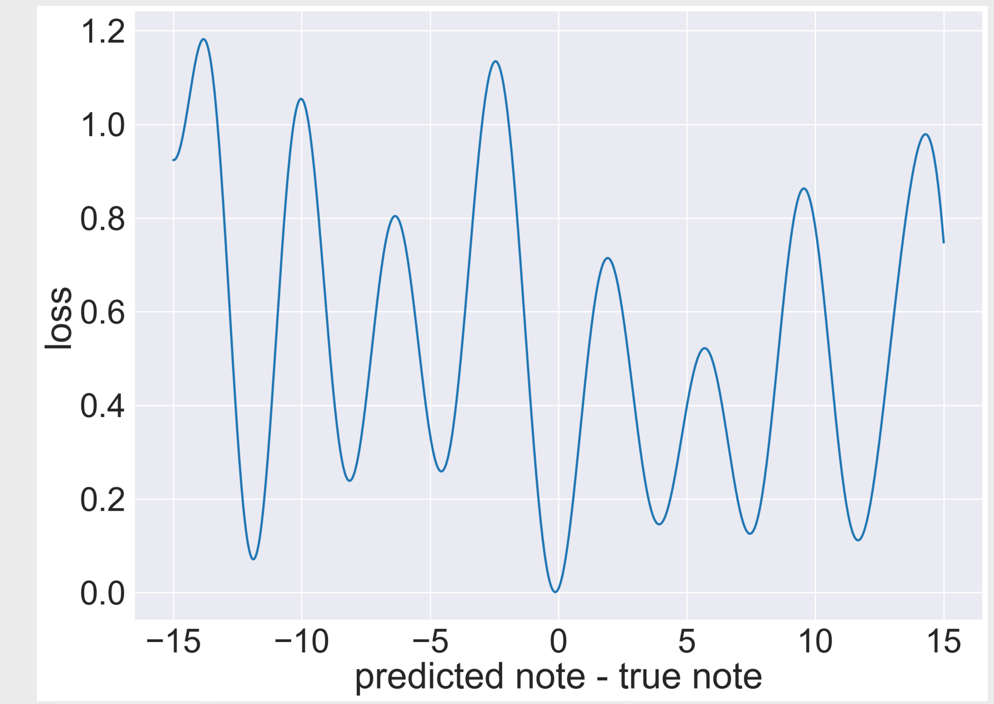
- Notes and velocity are both discrete values, however, we decided to use regression and rounding to predict the note. The training data was scaled before being used in the model.
- Our LSTM network was implemented in Keras.

Anatomy of an LSTM Unit



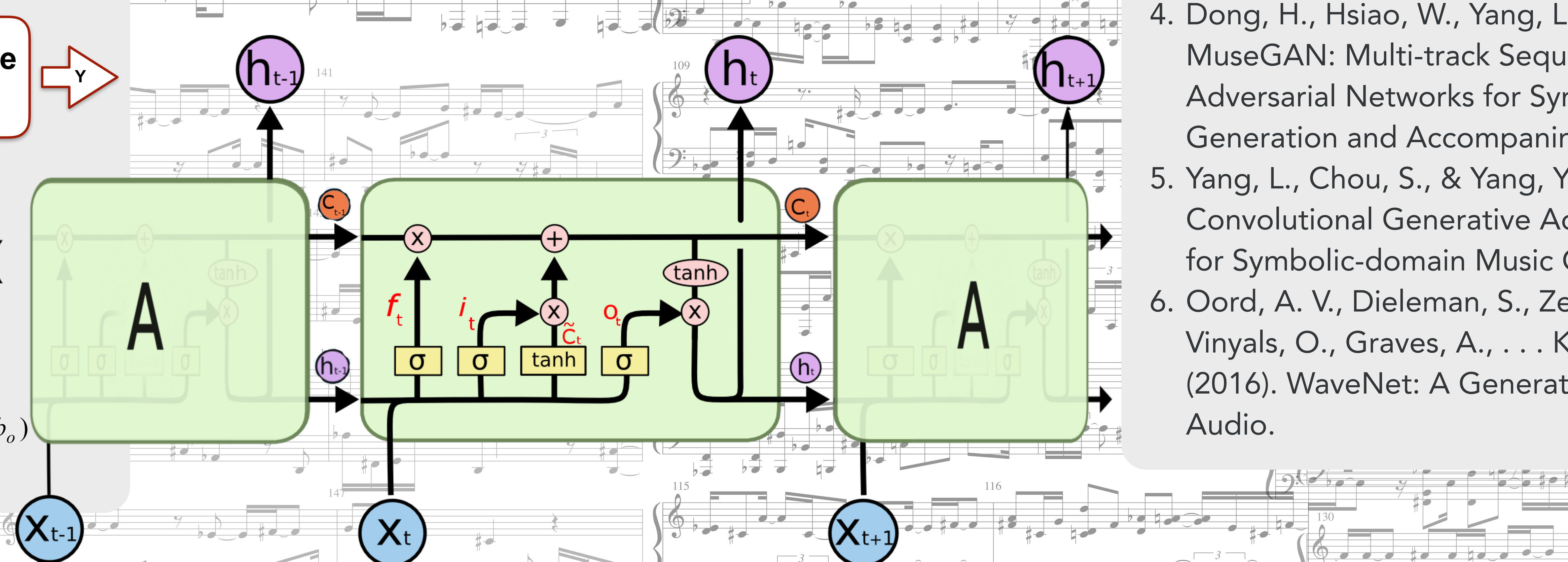
$$f_t = \sigma (W_f \bullet [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma (W_C \bullet [h_{t-1}, x_t] + b_C)$$
$$\check{C}_t = \tanh (W_C \bullet [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \check{C}_t$$
$$o_t = \sigma (W_o \bullet [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh (C_t)$$

- We made a loss function to represent the errors in our model's prediction of notes that would preserve the nature of music. Predicted values near the true value, 4 notes away, or 7 notes away would have low loss since these match the distribution of notes in a song.



Analysis

Feature	Evaluation
Key Signature	Consistently outputs the most common signature 4/4
Recurrent structure	Shows recurrent melodies and rhythms, that evolve over time
Tempo/Rhythm	Often finds steady tempos, and familiar rhythms, but does deviate from both sometimes
Chords	Learns the correct structure of chords, often played in 3s, while keeping a good balance with single notes
Key/Notes	Often fails to stay on a consistent key



Future Work

- Our model predicts which note should be played and when. It would be interesting if the model could predict the instrumentation of the note or note duration.
- To enhance learning a key, we could (a) increase the look-back, and (b) restrict our model to use one key at a time.
- With more computational power and a large dataset, it would be interesting to repeat this process with raw music files.

References

1. Eck, D., & Schmidhuber, J. (2002). A First Look at Music Composition using LSTM Recurrent Neural Networks. USI-SUPSI Instituto Dalle Molle(Tech. No. IDSIA-07-02).
2. Liu, T. I., & Ramakrishnan, B. (2014). Bach in 2014: Music Composition with Recurrent Neural Network.
3. Sidor, S. (2016). Magenta: Make Music and Art Using Machine Learning.
4. Dong, H., Hsiao, W., Yang, L., & Yang, Y. (2017). MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment.
5. Yang, L., Chou, S., & Yang, Y. (2017). MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation.
6. Oord, A. V., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., . . . Kavukcuoglu, K. (2016). WaveNet: A Generative Model For Raw Audio.