Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Deep Music: Towards Musical Dialogue

Mason Bretan
Georgia Institute of Technology
Atlanta, GA

Sageev Oore, Jesse Engel
Douglas Eck, Larry Heck
Google Research
Mountain View, CA

Abstract

Computer dialogue systems are designed with the intention of supporting meaningful interactions with humans. Common modes of communication include speech, text, and physical gestures. In this work we explore a communication paradigm in which the input and output channels consist of music. Specifically, we examine the musical interaction scenario of call and response. We present a system that utilizes a deep autoencoder to learn semantic embeddings of musical input. The system learns to transform these embeddings in a manner such that reconstructing from these transformation vectors produces appropriate musical responses. In order to generate a response the system employs a combination of generation and unit selection. Selection is based on a nearest neighbor search within the embedding space and for real-time application the search space is pruned using vector quantization. The live demo consists of a person playing a midi keyboard and the computer generating a response that is played through a loudspeaker.

Introduction

We present an interactive music system in which a person plays music with a computer system in a turn-taking paradigm. Unlike previous n-gram based interactive systems (Pachet 2003; Weinberg, Raman, and Mallikarjuna 2009), this system generates its responses based on a deep neural architecture that can be used to generate music at the note level or perform unit selection on a database comprised of human composed music where each unit consists of four beats worth of music.

The main premise for the interaction comes from musical "call and response" in which, in its basic form, one musician plays a phrase and a second musician plays a phrase in response. Continued interaction in this fashion creates a type of dialogue between the two musicians. In our system, the computer generated responses are derived from a weighted combination of the person's immediate input and its own previous outputs.

There are three important attributes a computational system must possess in order to achieve a successful interaction:

1. The ability to interpret and understand the higher level semantics of the humans input.
2. The knowledge to derive an appropriate response (i.e. describe the relationship between call and response pairs in the musical dialogue) and generate the response in real-time.
3. The ability to accommodate for the timing deviations inherent to human performance (as opposed to pure musical score representations).

Learning a Deep Music Model

In order for the dialogue to be successful the computer generated responses must be semantically relevant to what the human is playing. To determine semantic relevance we use a method similar to those used in language (word2vec, skip-gram, DSSM) and learn to predict the surrounding context of the music (Mikolov et al. 2013; Huang et al. 2013). The model consists of stacking encoders beginning at the level of a single "beat" and ending with an encoding of four beats (see Figure 1).

Each network in the stack is trained using cosine similarity.

$$sim(\vec{X}, \vec{Y}) = \frac{\vec{X}^T \cdot \vec{Y}}{|\vec{X}||\vec{Y}|} \quad (1)$$

Negative examples are included in a softmax function to compute $P(\vec{R}|\vec{Q})$ where $\vec{R}$ is the reconstructed vector and $\vec{Q}$ is the input vector.

$$P(\vec{R}|\vec{Q}) = \frac{\exp(sim(\vec{Q}, \vec{R}))}{\sum_{\vec{d}\epsilon D} \exp(sim(\vec{Q}, \vec{d}))} \quad (2)$$

The network learns the parameters by minimizing the following loss function using gradient descent:

$$-log \prod_{(Q,R)} P(\vec{R}|\vec{Q}). \quad (3)$$

Measure level (four beat) units are chosen by selecting the unit in the library with the minimum distance to the four-beat encoded vector. Unit selection is a powerful method because the model does not need to learn how to re-create all of the musical structure (in both the rhythmic and pitch space) and the information within each unit is guaranteed to be valid.

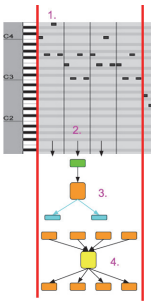

Figure 1: Deep Architecture – 1) The input to the network is based on a standard MIDI piano roll representation. In this system only the onset time of the note (finely quantized) is considered; duration is disregarded. 2) The grid corresponding to each beat is flattened to a one dimensional "beat-info" vector of length 1440 (24 ticks per beat and 60 possible pitches) and normalized to sum to unity. 3) Embeddings of these beat-info vectors are learned using a network that predicts the surrounding beat-info vectors. 4) For each measure, the four embedded beat-info vectors are concatenated and the semantic features describing all four beats are learned using an autoencoder.

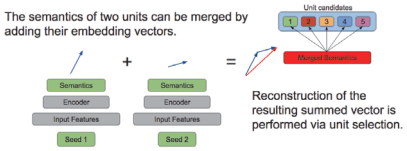

Figure 2: Merging of two input seeds.

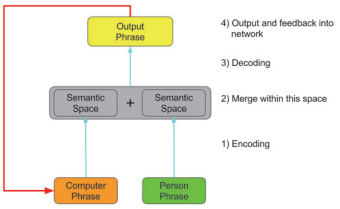

Figure 3: Musical dialogue system – A person plays within a defined duration of music which is then encoded and reconstructed. Reconstruction is performed at the note level using a decoder or through unit selection by searching for the closest unit (within the embedding space) in a music database. The reconstruction is then played by the computer and fed back into the network to influence its next output.

However, the flexibility to generate music encompassing a wide range of styles can be lost in varying degrees depending on the duration of each unit and the size of the database. With this architecture, unit selection can also be performed at the beat level allowing for greater flexibility in what music can be reconstructed.

Additionally, maximum flexibility can be achieved by using the model to generate sequences at the note level (though not without some degradation in quality). To this a decoder at the measure level is used to reconstruct the beat-level embeddings. Then a separate decoder, trained to predict the original information within a single beat from the embedding space, is used to generate individual notes.

Interaction

Adapting for performance The model is trained on a corpus of publicly available MIDI data that includes classical and jazz music. We also include original performance data of two musicians playing in a turn taking paradigm. The majority of the available data are highly quantized representations of music and not typically what would be seen in live performance. One method of accommodating for this is to quantize the human input before providing it to the network. However, quantization algorithms tend to remove desirable rhythmic sophistication.

Instead, we augment the dataset by including temporally "humanized" representations of the music. The humanization process entails shifting the positions of the quantized notes. While synthetic, these temporal shifts help the model learn to respond to more than just quantized data. We also pitch shift the data so that each piece of music is covered in all twelve keys. Thus, the system is key agnostic and will adapt to the user.

Creating a dialogue One good method of responding in a call and response interaction is to play back an almost identical version of the call in which only one or two notes are different. An artifact (or feature) of the unit selection process is that the unit that exactly replicates the call may not exist in the database. However, the most similar unit typically serves as a reasonable response.

This method of interaction is expanded upon by creating outputs that are constructed from a combination of multiple inputs. The merging of multiple input seeds is performed in the embedding space (see Figure 2). Such merging (or any manipulation in the embedding space) requires that the embeddings encode information that is perceptually meaningful and musically relevant. The final output of the system is a reconstruction of a merged representation of the person's most immediate input and the computer's previous output (see Figure 3).

References

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2333–2338. ACM.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Pachet, F. 2003. The continuator: Musical interaction with style. *Journal of New Music Research* 32(3):333–341.

Weinberg, G.; Raman, A.; and Mallikarjuna, T. 2009. Interactive jamming with shimon: a social robotic musician. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, 233–234. ACM.