MIDI Database and Representation Manager for Deep Learning

Jeff Ens, Philippe Pasquier
School of Interactive Arts and Sciences
Simon Fraser University
{jeffe,pasquier}@sfu.ca

Introduction

One of the most important decisions made when training neural networks, is how to represent the data. Despite the large number of possible representations, the piano roll dominates recent literature on modelling music [2, 3]. Furthermore, previous work suggests that modelling simpler conditional probability distributions such as $p(\text{pitch}|\text{rhythm})$, may be an advantageous approach to the complex task of modelling music [5]. However, many alternate representations are more difficult to implement than the piano roll. Motivated by these factors, we propose an accessible framework for symbolic musical data storage, and dataset construction, which supports a variety of representations.

Representations

Base Representation

Information extracted from a MIDI file is stored in a .tfrecords file, to be compatible with the Tensorflow [1] Dataset API. We store the start time, end time, pitch, velocity, channel number, and midi instrument of each note. In addition, we store the start time, and length of each measure, which is useful when a representation involves separating a piece into measures. All time-based attributes are stored in ticks, which allows the user to specify the degree of quantization at run-time, and preserves the original MIDI data without loss of information. Notablye, additional metadata such as the title, composer, and bpm is also stored.

Univariate Representations

A univariate representation is a sequence of one-hot vectors, encoding a sequence of ⟨state, value⟩ tuples. We represent each *state* and all valid *state-transitions* using a directed graph, as shown in Figure 1. Figure 1 shows a generalized version of the representation proposed by Liang and others, where the pitches sounding in each timestep are delimited by the step state [4]. Additionally, Liang and others specifies that the pitches in each timestep must be arranged in ascending order. To accomodate these types of representations, we provide a simple interface for specifying *states*, valid *state-transitions* and constraints on the value of a state dependant on the values of previous states. It is very straightforward to construct more complex representations. Once a directed graph has been specified, the conversion between a ⟨state, value⟩ tuple and a one-hot vector is provided via a single function call. Furthermore, sequences can be validated, ensuring that there are no invalid *state-transitions* and that no constraints are violated.

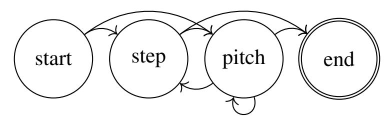

Figure 1: A univariate representation.

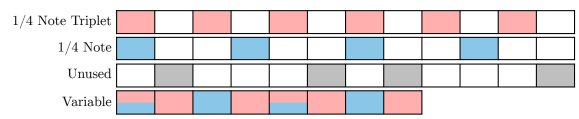

Figure 2: Fixed and variable resolution representation.

Multivariate Representations

A multivariate representation is a sequence of binary vectors (stored as a matrix). The most common multivariate representation is the piano roll. Typically, piano rolls are constructed using a fixed resolution (i.e. 16^{th} note resolution), however, we also support a variable resolution representation. The benefits of using a variable resolution are shown in Figure 2. Consider a rhythm consisting of $1/4$ note triplets and standard $1/4$ notes with the length of a single beat. A fixed resolution representation would require 12 bits to represent the rhythm, however, 4 of these bits would never be used. The variable representation would only require 8 bits.

References

[1] Abadi, M., et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

[2] Boulanger-Lewandowski, N., et al. 2012. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proc. of the 29th Int. Conf. on Machine Learning*, 1159–1166.

[3] Huang, C. A., et al. 2017. Counterpoint by convolution. In *Int. Conf. on Learning Representations (under review)*.

[4] Liang, F., et al. 2017. Automatic stylistic composition of bach chorales with deep lstm. In *Proc. of the 18th Int. Symposium for Music Information Retrieval*, 449–456.

[5] Walder, C. 2016. Modelling symbolic music: Beyond the piano roll. In *Proc. of the 8th Asian Conference on Machine Learning*, 174–189.