Web-based Dynamic Visualization of Rhythmic Latent Spaces

Gabriel Vigliensoni

Goldsmiths University of London

g.vigliensoni@gold.ac.uk

Esteban Maestre

McGill University

esteban.maestre@mcgill.ca

Rebecca Fiebrink

University of the Arts London

r.fiebrink@arts.ac.uk

ABSTRACT

We present a web-based visualizer designed for the dynamic representation of rhythmic latent spaces. These spaces are learned from user-provided MIDI clips with rhythms. Instead of computing a specific set of hand-coded features to describe the rhythms and using them for visualizing the space, we rely directly on the rhythmic patterns and their pulses to trigger visual cues in the canvas of the browser. To the best of our knowledge, this is the first time that a dynamic visualization has been implemented to observe a latent space learned from rhythmic patterns.

INTRODUCTION

Several recent projects have aimed at creating a model from a dataset of rhythmic patterns. The high-dimensional characteristics of the rhythms are encoded and mapped into a lower dimensional latent space, where points in this space will resemble datapoints from the the original distribution. Once such as a space is created, it can be used to interpolate between rhythms by means of travelling in the learned latent space.

Currently, variational autoencoder neural networks (VAEs) (Kingma and Welling, 2014) are a popular technique to learn a compact representation of the training data that retains its structure. Researchers from the Google Magenta team have released a number of datasets, data structures, and network architectures that are used to create representations of rhythmic patterns using VAEs. These include MusicVAE (Roberts et al, 2018), GrooVAE (Gillick et al, 2019), and E-GMD (Callender, Hawthorne, and Engel, 2020). These networks are becoming popular and a number of applications have used them and related models (e.g., DrumsRNN (Google Magenta, 2016)) to create instruments for exploring or playing with rhythmic patterns in the browser. Among these are *Latent Loops*,[1] *Beat Blender*,[2] *Neural drum machine*,[3] and *Drumify*.[4]

All the aforementioned web-based applications provide some form of user interaction that permits selecting, sequencing, or creating paths between points in the latent space. These actions permit the user the retrieval and decoding of rhythmic patterns from the latent space into some form of data structure that can be played back. However, the latent space is often hidden (e.g., in Drumify or Neural Drum Machine), or else only a static representation of the available patterns is provided (e.g., Latent Loops or Beat Blender). The lack of meaningful

---

[1] https://teampieshop.github.io/latent-loops

[2] https://g.co/beatblender

[3] https://codepen.io/teropa/full/RMGxOQ/

[4] https://magenta.tensorflow.org/studio/ableton-live#drumify

visual cues makes it very hard to repeat a musical gesture that maps to a specific musical response, leaving the performer without useful feedback to perform with the latent space.

RESEARCH QUESTIONS

Since the lack of visual feedback can hinder the process of learning how the rhythmic patterns are distributed in the latent space, there are a number of questions whose answers may help performers to understand how they might explore, play, and perform with rhythmic patterns encoded in a model. For example: how can a visualization enable a performer to understand how rhythmic patterns are spatially distributed? How does such a visualization support performance-time interaction?

IMPLEMENTATION

We have implemented a variational autoencoder-based rhythm explorer called R-VAE (Vigliensoni et al. 2020). It is built upon tfjs-vae[5] and M4L.RhythmVAE (Tokui, 2020). The former contributes the backend for the VAE, and the latter encodes the onsets of rhythms, their velocities, and microtimings, using a fully connected feedforward layers for fast training and inference using CPUs. We extended the data representation of this architecture to encode rhythms in simple and compound meter, common in modern music genres such as *gqom*, *dubstep*, *trap*, and *footwork*.

We released R-VAE as a web-based app that can be used as a rhythm model player, enabling people to explore rhythmic latent spaces and make music directly in the browser. From a human-computer interaction point of view this poses challenges because the latent space is a continuous function where salient characteristics of the original distribution are encoded but the space axes have no clear labels. Therefore, instead of implementing metrics for characterizing rhythms and their similarities (Toussaint et al, 2004), or hand-crafting metrics to try to represent how *rich* or *smooth* the rhythmic latent space is, we designed a tool to visualize the latent space that relies on the dynamic nature of the musical events.

We sample the original, continuous two-dimensional latent space at discrete points, retrieving the onset probabilities for the drum instruments (e.g., kick, snare, and hi-hat) for a specific point in the latent space over time t = [0: T-1]. Then, we scale the normalized, continuous probability values of each instrument to discrete, 8-bit RGBA values for each zone in the performance space, and we map these values into square matrices of order 2 or 3 representing the different zones of the space.

A representation of how the instruments per pattern in the latent space are mapped to the visualization canvas is shown in Figure 1.

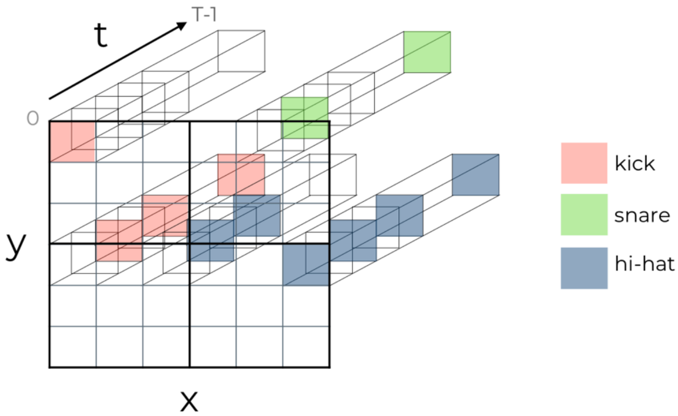

Figure 1: A diagram illustrating how the dynamically changing instrument patterns in the latent space are mapped to the canvas of the browser. Rhythmic patterns are retrieved from discrete points of the latent space. The onsets of instruments are then mapped to individual pixels in the visualization canvas over time.

In the figure, we show four points (i.e., four rhythms) in the latent space---one per quadrant, each represented as a 3-by-3 matrix. Each instrument in a rhythmic pattern will trigger a specific point in the matrix with a single color. For example, a kick will only trigger *red* pixels in the position [0,0] of the matrices, a snare will only

[5] https://github.com/songer1993/tfjs-vae

trigger *green* pixels in position [0,1], and hi-hats will only highlight pixels in *blue* in position [0,2]. The time dimension t shows how each point in the latent space changes according to the onsets of the corresponding instrument over time for the given rhythm. Using a clocking system sync to a specific tempo, an imaginary playback head traverses all the matrices from time `t = 0` to `t = T - 1`; each pixel of the visualization canvas is updated accordingly over time.

The code for the visualizer[6] as well as a live demo[7] are available.

DISCUSSION

We have used the visualization tools to play with latent spaces trained on datasets of rhythms of different genres such as footwork, trap, gqom, and 2-step. We have observed that the visualizer captures nicely how the different patterns are distributed in the latent space and provides a much needed visual feedback when interacting with the model. Since patterns are synced in time, similar, neighbouring zones flash synchronously, exposing previously hidden rhythmic clusters in the space. On the contrary, adjacent zones with elements in different meter flash asynchronously, giving the performer a natural visual to discriminate these zones and their boundaries. The use of the three basic colors enhances the visualization of the edges of zones with similar patterns and also help to identify the transition zones between them. These zones of transition are particularly interesting because this is where the rhythmic morphing of different zones happen. The visualizations have also been helpful to show that the dimensionality reduction process of the VAE produces unexpected neighbouring clusters. For example, when dealing with rhythms with simple and compound meter, it can be seen that groups of perceptually different patterns are neighbours in the space. We have not yet implemented in the visualizer a routine to limit the complexity of the retrieved patterns. This thresholding technique is implemented in the audio part of the application and allows the performer to increase or reduce the number of onsets retrieved. The visualization tool should exhibit the same behaviour. Finally, the visualizer presented in this work was implemented for models created with a VAE network, however its design is easily generalizable to other architectures.

REFERENCES

Callender, L., Hawthorne, C. and Engel, J. (2020) ‘Improving perceptual quality of drum transcription with the Expanded Groove MIDI Dataset’, arXiv:2004.00188.

Gillick, J. *et al*. (2019) ‘Learning to Groove with Inverse Sequence Transformations’, in *Proceedings of the 36th International Conference on Machine Learning*.

Google Magenta. (2016) ‘Drums RNN’. Available at: https://github.com/tensorflow/magenta/tree/master/magenta/models/drums_rnn (Accessed: 1 August 2020)

Kingma, D. and Welling M. (2014) ‘Auto-encoding Variational Bayes’, in *Proceedings of the 2nd International Conference on Learning Representations*.

Roberts, A. *et al.* (2018) ‘A Hierarchical Latent Vector Model for Learning Long-term Structure in Music’, in *Proceedings of the 35th International Conference on Machine Learning*. arXiv preprint arXiv:1803.05428.

Tokui, N. (2020). ‘Towards democratizing music production with AI-design of variational autoencoder-based rhythm generator as a DAW plugin’. arXiv preprint arXiv:2004.01525.

Toussaint, G. et al (2004) ‘A Comparison of Rhythmic Similarity Measures’, in *Proceedings of the 5th International Conference on Music Information Retrieval.*

Vigliensoni, G., McCallum L. and Fiebrink R. (2020) ‘Creating Latent Spaces for Modern Music Genre Rhythms Using Minimal Training Data’, in *Proceedings of the 11th International Conference on Computational Creativity*.

[6] https://github.com/vigliensoni/R-VAE-JS/

[7] https://vigliensoni.github.io/R-VAE-JS-dev/