Generation and visualization of rhythmic latent spaces

Gabriel Vigliensoni[1], Louis McCallum[1,2], Esteban Maestre[3], and
Rebecca Fiebrink[1,2]

[1] Department of Computing, Goldsmiths University of London
[2] Creative Computing Institute, University of the Arts London
[3] Department of Music Research, McGill University
g.vigliensoni@gold.ac.uk

Abstract. In this paper we extend R-VAE, a system designed for the modeling and exploration of latent spaces of musical rhythms. R-VAE employs a data representation that encodes simple and compound meter rhythms, common in some contemporary popular music genres. It can be trained with small datasets, enabling rapid customization and exploration by individual users. To facilitate the exploration of the latent space, we provide R-VAE with a web-based visualizer designed for the dynamic representation of rhythmic latent spaces. 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.

Keywords: Latent space, music visualization, rhythm patterns, rhythm similarity, dimensionality reduction

1 Introduction

In this paper, we present research on customizing a variational autoencoder (VAE) neural network (Kingma & Welling, 2014) to play with musical rhythms encoded within a latent space. A number of publications, datasets, data structures, and network architectures to encode rhythmic patterns using VAEs have been recently released (e.g., Roberts et al., 2018, Gillick et al., 2019, Callender et al., 2020), however none of them can encode rhythms in compound meter, common in many traditional rhythms from Latin America or Africa, and in contemporary music genres, such as *footwork*, *trap*, *2-step*, *gqom*, or *dembow.* We observe that biases not only appear from the data we use to train models, but also from the representation we choose to encode the data. In addition, none of the previous approaches to create rhythmic latent spaces provides a dynamic way of visualizing the space as a whole, and so the performer is blind to how the rhythmic patterns are organized in the space and has to explore and play with them without visual cues.

The system we designed can generate a series of models using minimal training data, with as few as one dozen MIDI clips with rhythms. It uses a data structure that is capable of encoding rhythms in simple and compound meter.

To facilitate the exploration of the latent space, we provide our system with a web-based visualizer designed for the dynamic representation of rhythmic latent spaces that relies directly on the pulsing rhythmic patterns to trigger visual cues in the canvas of the browser.

To the best of our knowledge, this is the first time that a network architecture has been used to encode rhythms with simple and compound meter and the first time that a visualization has been implemented to observe dynamically a latent space learned from rhythmic patterns.

2 Implementation

We have implemented a variational autoencoder-based rhythm explorer called R-VAE (Vigliensoni, McCallum, & Fiebrink, 2020). It is built upon tfjs-vae[4] and M4L.RhythmVAE (Tokui, 2020). While the former contributes the Tensorflow backend for the VAE, the latter provides a data structure based on the one proposed by Gillick et al. (2019) that encodes the onsets of rhythms, their velocities, and microtimings, and comes conveniently packed as a Node for Max application that can be opened as a Max for Live device in Ableton Live.

2.1 Data representation

In R-VAE we extend the data representation of M4L.RhythmVAE to encode rhythms in simple and compound meter. Most previous approaches for encoding rhythms using VAEs used sixteen 16th notes per bar of 4/4 time, corresponding to a resolution of four ticks (i.e., subdivisions) per quarter note. However, the encoding of most contemporary music genres needs a much finer grid of up to a 32nd triplet note, which we consequently choose as the basic unit in our data representation. Therefore, the encoding of one bar of 4/4 time in R-VAE comprises three matrices (for onsets, velocities, and microtimings) of dimensions 96 × 3. These dimensions represent 24 ticks × 4 quarter notes × 3 drum instruments.

2.2 Network configuration

The network configuration for model training consists of a vanilla VAE architecture with 864 dimensions for the input, 512 for the intermediate layer, and 2 dimensions for the resulting latent space. The batch size is set to 64, the optimization algorithm to Adam, and the activation function to LeakyReLU. The favouring of fully connected feedforward layers by Tokui instead of Gillick et al.'s bidirectional LSTMs allows for faster training using CPUs. We compared the performance of this implementation with much larger and complex architectures such as MusicVAE (Roberts et al., 2018) and GrooVAE (Gillick et al., 2019) and found R-VAE required considerably less data and processing power to converge into a useful model.

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

2.3 Web-based model player and visualizer

R-VAE can be used for model training and playback as a Max for Live device. Using it as an Ableton device comes with the big advantage of being embedded in a DAW, offering fast and simple MIDI and audio routing for changing and processing drum sounds, and easy mapping using MIDI or OSC to explore the latent space using a gestural controller at performance time, or to automatize the device for fine control during recording or mixdown.

In addition to the Ableton device, we also released R-VAE as a web-based application that can be used as a rhythm model player, enabling people to explore rhythmic latent spaces and make music directly in the browser. The webapp has a GUI that provides that reveals all rhythmic patterns in the latent space at the same time. 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 (e.g., Toussaint et al., 2004), using dimensionality reduction techniques such as self-organizing maps (Kohonen, 1990), or adapting metrics to try to represent how *rich* or *smooth* (Berthelot, Raffel, Roy, & Goodfellow, 2019) is the rhythmic latent space, we implemented a visualization that relies on the temporal nature of the musical events to generate a dynamic representation of the latent space. The interface allows the performer to dynamically visualize the whole space at once, providing a helpful visual feedback during improvisation and musical performance.

The implementation of the visualization is based on the mapping of the onset probability values of the instruments in the latent space to the brightness of their representation in the browser canvas. To achieve this, we sample the original, continuous two-dimensional latent space at discrete points over time and retrieve the onset probabilities for each drum instrument (e.g., kick, snare, and hi-hat). Then, we scale the instruments' probability values to the range $[0, 255]$ and fill square matrices of order 2 or 3 with these scaled probability values. Cells within each matrix correspond to a specific drum instrument. These instruments are rendered in the browser using a single color per instrument and 8-bit RGBA values. Their onset probability is mapped to their brightness.

A graphic representation of how the instruments per latent space point are mapped to the visualization canvas is shown in Figure 1. In the figure, we see four matrices of order 3 corresponding to four discrete points (i.e., four rhythms) in the latent space. 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 instrument in a rhythmic pattern will trigger a specific matrix cell 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 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 the colors within each of the matrices change according to the onset probabilities retrieved from the latent space for the corresponding instruments, so that each

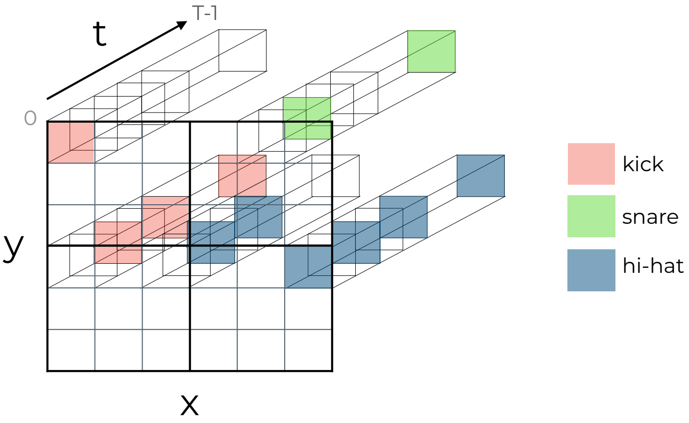

Fig. 1. A diagram illustrating how the dynamically changing instrument patterns in the latent space are mapped to the canvas of the browser. Four discrete points (i.e., four rhythms) of the latent space are sampled, each represented in the figure as a 3-by-3 matrix. Each instrument in a rhythmic pattern will trigger a specific matrix cell with a single color.

pixel of the visualization canvas is mapped to the latent space and updated accordingly over time.

Although the data representation of R-VAE encodes up to nine drum instruments, in order to limit the amount of information displayed in the visualizer we opted to display only the three main drum instruments in contemporary music genres (i.e., kick, snare, and hi-hat). Additional knobs for *threshold* and *noise* help the performer control how the latent space is sampled. Mute buttons enable the performer to silence individual instruments.

A video demonstrating the capabilities of R-VAE and snippets of renditions performed with it can be accessed at `https://vimeo.com/433780684`. Both implementations of R-VAE, for Ableton Live[5] and the R-VAE-JS browser-based model player,[6] are available. A series of models trained on rhythms with simple and compound meter are also available.[7]

3 Conclusions and Future Work

The main contributions of this research are twofold: (i) a data representation that broadens the type of rhythms that can be encoded using VAE towards contemporary music genres, and (ii) a dynamic visualizer that displays all rhythmic patterns in the latent space at once. Although the first contribution could be seen as a modest technical extension to previous implementations, we think it

[5] `https://github.com/vigliensoni/R-VAE`

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

[7] `https://github.com/vigliensoni/R-VAE-models`

entails a great musical contribution since, for the first time in the modelling of rhythms using VAEs, we can now encode rhythms in compound meter, thus reducing the biases (e.g., toward particular genres and cultures) produced by data representations that only encode certain types of rhythms. In regards to the second contribution, our visualization advances research in latent spaces for music by moving beyond conventional mapping towards a more integrated visual expression of the sonic material.

When experimenting with data selection and model training of rhythms in simple and compound meter, we found that R-VAE was able to learn useful and playable models with as low as one dozen MIDI clips. However, when we increased the number of clips to a few dozen, the learned latent spaces exhibited a more even topology, therefore broadening the boundary zones, improving the interpolations, and creating smoother and richer spaces, which is good for performance.

We investigated the viability of R-VAE in live performance contexts at the MUTEK International Festival of Digital Creativity and Electronic Music and the Network Music Festival 2020, and 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 cue to discriminate these zones and their boundaries. The threshold knob was a also very important parameter in live contexts because it helped the performer to control the complexity of the rhythmic patterns by limiting the number of onsets retrieved from the latent space at any given time. Sometimes it was interesting to stay in certain point of the latent space and only play with the threshold to obtain interesting rhythmic variations. Currently, the thresholding parameter only updates latent space that the audio is generated from without changing what is being represented by the visualiser. We plan to rectify this in future.

Our experience with R-VAE has reinforced the idea that a system for the exploration of latent spaces of musical rhythms is worth pursuing further. For example, systems like this could be also used for browsing through libraries of rhythms, common in contemporary music production, or for the generation of non-linear music soundtracks, common in videogames. Finally, the visualizer presented in this work as part of R-VAE was implemented for models created with a VAE network, however its design is easily generalizable and has the potential to be used with other network architectures.

4 Acknowledgments

This research has been supported by Fonds de recherche du Québec – Société et culture (FRQSC) through a research creation grant (Ref. 2020-B5-272998).

References

Berthelot, D., Raffel, C., Roy, A., & Goodfellow, I. (2019). Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *Proceedings of the 7th International Conference on Learning Representations.*

Callender, L., Hawthorne, C., & Engel, J. (2020). Improving perceptual quality of drum transcription with the expanded groove MIDI dataset. *arXiv:2004.00188*.

Gillick, J., Roberts, A., Engel, J., Eck, D., & Bamman, D. (2019). Learning to groove with inverse sequence transformations. In *Proceedings of the 36th International Conference on Machine Learning.* Retrieved from `https://magenta.tensorflow.org/datasets/groove`

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations.*

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464–1480.

Roberts, A., Engel, J., Raffel, C., Hawthorne, C., & Eck, D. (2018). A hierarchical latent vector model for learning long-term structure in music. In *Proceedings of the 35th International Conference on Machine Learning.*

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. T., et al. (2004). A comparison of rhythmic similarity measures. In *Proceedings of the 5th International Conference on Music Information Retrieval.*

Vigliensoni, G., McCallum, L., & 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.*