Creating Latent Spaces for Modern Music Genre Rhythms Using Minimal Training Data

Gabriel Vigliensoni,[1] **Louis McCallum,**[1] **and Rebecca Fiebrink**[1,2]
[1]Department of Computing, Goldsmiths University of London, UK
[2]Creative Computing Institute, University of the Arts London, UK
g.vigliensoni@gold.ac.uk

Abstract

In this paper we present R-VAE, a system designed for the exploration of latent spaces of musical rhythms. Unlike most previous work in rhythm modeling, R-VAE can be trained with small datasets, enabling rapid customization and exploration by individual users. R-VAE employs a data representation that encodes simple and compound meter rhythms. To the best of our knowledge, this is the first time that a network architecture has been used to encode rhythms with these characteristics, which are common in some modern popular music genres.

Introduction

In this paper, we present research on customizing a variational autoencoder (VAE) neural network to play with musical rhythms encoded within a latent space. To enable customization and personalization, the network can be trained with as few as one dozen MIDI clips with rhythms.

Additionally, our approach employs a data structure that is capable of encoding rhythms with binary, ternary, or a combined metrical grid. A metrical grid can be explained as the main ratio by which the onsets of notes are placed in a measure. Music where the beats are split in two—a binary grid—is in *simple meter*. Music where the beats are split in three—a ternary grid—is in *compound meter*. Many modern music genres, such as *footwork*, *gqom*, *dembow*, or *trap*, can be characterized by rhythmic elements using a compound meter. In these rhythms, the main meter is usually simple but there are elements that are placed on an overlaid ternary grid. To the best of our knowledge, this work is the first time that a network architecture has been used to encode rhythms that exhibit a combined binary and ternary grid.

Related Work

Several recent projects have aimed to use machine learning to encode the regularities in rhythmic patterns present in user-provided examples into a model, so that this model can afterwards be used to sample rhythms from the original distribution or generate new, unheard rhythms. For instance, Choi, Fazekas, and Sandler (2016) created a model trained on drum patterns from songs by Metallica to generate new rhythmic sequences. Their approach was based on a text-based LSTM (Long Short Term Memory) network, so that they had to adapt and encode the rhythm onsets to fit the data representation. They limited the number of events in a bar to 16 by quantizing the drum tracks to 16^{th} notes in a simple meter binary grid. They used 60 MIDI files of drum tracks as training data and created drum sequences that were "reasonable rock drum patterns." However, since the data representation they used did not encode the deviations of onsets from the grid—known as *microtiming*—nor how hard the onsets were struck—known as *velocity* in the MIDI domain—the resulting patterns were very rigid.

Instead of focusing on a specific musical artist or style, Nikolov (2016) trained another LSTM network on drum patterns from a wide range of music genres with the goal of creating a general model for rhythm generation. This work used professionally produced MIDI loops, increased the quantization grid to a 32^{nd} note, but used only rhythms with a 4/4 time signature and simple meter. Nikolov described some of the resulting examples as "musically coherent patterns" but only in a short timescale.

With the goal of learning longer-term musical structure, researchers of the Google Magenta team experimented with language modeling and LSTMs to encode and generate melodies and drum patterns. They released network architectures designed to learn representations of melodies[1] and rhythms[2] encoded in a symbolic format, as well as models pre-trained on large datasets containing "thousands of MIDI files" of undisclosed origin. The data representation did not encode microtimings or velocities.

Using the data representations implemented in the Magenta projects, and with the goal of modeling sequences with even longer term structure, the Magenta team released MusicVAE (Roberts et al. 2018). This network used a hierarchical variational autoencoder (VAE) architecture to encode and generate melodies, drums, and "trios" consisting of a drum part, a bass line, and a melodic line. For these three categories, the Magenta team also released models that were trained on a very large dataset of more than 1.5 million MIDI files collected from the web. Roberts et al. reported that MusicVAE was able to generate 2-bar drum sequences reliably,

[1] https://github.com/magenta/magenta/tree/master/magenta/models/melody_rnn

[2] https://github.com/magenta/magenta/tree/master/magenta/models/drums_rnn

but failed when trying to reconstruct 16-bar sequences.

In the aforementioned approaches to drum pattern modeling, only the position of the drum onsets was encoded, not their microtiming or velocity. These two characteristics are important for giving the drum loops a human feel, or groove. In order to overcome those flaws, the Magenta team released GrooVAE (Gillick et al. 2019), a data representation and a set of models trained on real drum performances. Again, this work used quantization to a 16th note grid in both the data structure and models, resulting in an inability to encode rhythms such as footwork or trap, which commonly have rhythmic elements in compound meter. As a result, these elements are quantized incorrectly into the binary grid, probably loosing their rhythmic signature or "feel."

A number of applications have been released based on the Magenta data representation and their pre-trained models. For example, the Neural Drum Machine[3] is a web-based application in which the user seeds the system with a short rhythmic sequence, and the model "imagines" the continuation. Beat Blender[4] packages MusicVAE pre-trained models in a web application in which the user plays back patterns and create paths in a latent space filled with rhythms. The Drum Beats Latent Space Explorer[5] is another web application that uses a VAE architecture trained on 33K MIDI drum files to learn a bi-dimensional latent space representation that can be explored in a browser. All the rhythms in these applications are quantized to 16th notes and so their data structure is only able to decode rhythms based on a binary grid of simple meter.

R-VAE

Motivation

Our goal is to design a system that can generate a series of models using minimal training data, to better enable artists without extensive computational resources to build and explore bespoke rhythm models. Further, we would like this system to encode the onsets, velocities, and microtimings of rhythms, and to allow the encoding of simple or compound meter rhythms, or their combination. Once a model is trained, performers using the system should be able to explore the latent space of the model and retrieve rhythmic patterns as if they were moving a playback head on it.

Most of the architectures, models, and applications we have reviewed have been trained using very large datasets of rhythms. For example, MusicVAE models were trained on about 1.5 million unique MIDI files (Roberts et al. 2018), and the GrooVAE and Expanded Groove MIDI Dataset models were trained on more than 13 and 444 hours of music, respectively, performed by professional drummers and recorded in both MIDI and audio formats (Gillick et al. 2019; Callender, Hawthorne, and Engel 2020). The goal of these representations was to learn the groove—the human feel—in drum performances. To achieve this, the authors encoded in a VAE network the onsets, velocities, and microtimings of the drum hits.

These approaches entail practical challenges, both in that they require large-scale datasets and in that they are trying to learn commonalities from disparate data. Creating generic models from large and diverse data can be interesting and reasonable from a computational point of view, but this can also hinder customization. Individual creative practitioners usually do not have access to large datasets or the processing power to train such large models. On the other hand, the ability to train models from smaller datasets can enable the modeling of niche genres or even a personal style, while requiring fewer resources for training.

Some prior work has aimed to empower individual expression and personalization of generative models by enabling training from smaller datasets. Dinculescu, Engel, and Roberts (2019) introduced MidiMe, an approach to quickly train a small model to control a much larger and generic latent variable model. Their system learns a compressed representation of the already encoded latent vectors of MusicVAE and generates musical melodies from only portions of its latent space based on MIDI files with melodies provided by users by means of a web-based app. Other work has enabled musicians to create bespoke supervised learning systems with small training sets; for instance, Wekinator (Fiebrink, Trueman, and Cook 2009) uses shallow multi-layer perceptron neural networks to learn bespoke mapping functions (e.g., from a performer's gestures to sound synthesis parameters) using small datasets generated by musicians in realtime.

Implementation

Autoencoders (Kingma and Welling 2014) can learn a compact representation of the training data that captures important factors of variation in the dataset. Points in the latent space map to realistic datapoints, and nearby points map to semantically similar (or here, musically similar) examples. Variational autoencoders (VAEs), in particular, assume the training data has an underlying probability distribution and attempt to find the parameters of the distribution. Once those parameters are found, we can sample the space to generate data that will follow the original distribution. In other words, in VAEs the generated data will be related to (but not necessarily the same as) the source data. This make VAEs a good network topology for creating generative models of rhythms.

We have implemented a variational autoencoder-based system, called R-VAE, which, for the first time, encodes simple and compound meter rhythms. It also encodes the onsets, velocities, and microtimings of rhythms, and can be trained using small datasets. We released 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. We based the implementation of our rhythm explorer on the Tensorflow.js VAE implementation called tfjs-vae[6] and the M4L.RhythmVAE rhythm generator device (Tokui 2020). While the former provides the Tensorflow backend for the VAE, the latter provides a data

[3] https://bit.ly/nikolov-neuralbeats

[4] https://experiments.withgoogle.com/ai/beat-blender/view/

[5] https://towardsdatascience.com/drum-patterns-from-latent-space-23d59dd9d827

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

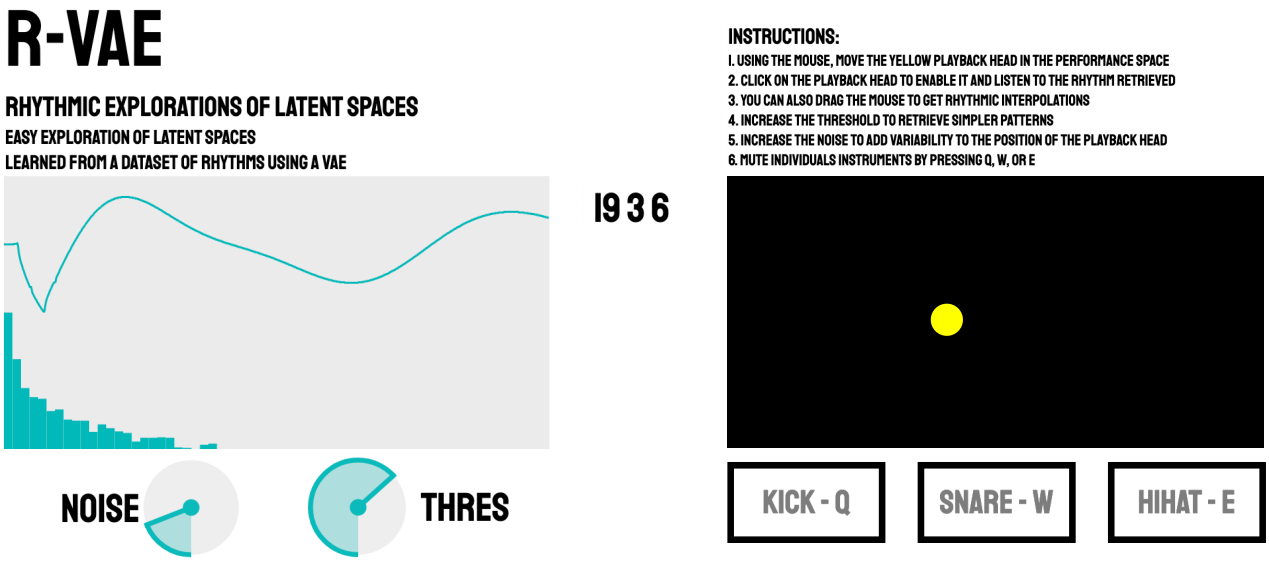

Figure 1: User interface of R-VAE-JS web application. The latent space can be explored by using a playback head represented by the yellow circle, in the black performance space on the right of the figure. Additional knobs for threshold and noise help the performer to control how to the latent space is sampled. Mute buttons enable the performer to silent individual instruments. User interface elements on the left side provide additional visual feedback.

structure based on the one by Gillick et al. (2019) that encodes the onsets of rhythms, their velocities, and microtimings. M4L.RhythmVAE also comes conveniently packed as a Node for Max application that can be opened as a Max for Live device in Ableton Live, a popular digital audio workstation.

The configuration for training our model consists of a vanilla VAE architecture with 864 dimensions for the input, 512 for the intermediate layer, and 2 dimensions for the 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 and we compared the performance of this implementation with much larger and complex architectures such as MusicVAE and GrooVAE. We found it required considerably less data and processing power to converge into a useful model.

Binary and ternary representations In R-VAE we extended the internal data representation of M4L.RhythmVAE to encode simple and compound meter rhythms, as well as their combination. Most previous approaches used only sixteen 16^{th} notes in one bar of 4/4 time, corresponding to a resolution of four ticks (i.e., subdivisions) per quarter note. However, the encoding of most modern music genres needs a much finer grid of up to a 32^{nd} triplet note, which we then chose as the basic unit in our data representation. Then, 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 $\times$ 4 quarter notes $\times$ 3 drum instruments. Although GrooVAE and M4L.RhythmVAE work with nine canonical drum categories, for this project we only work with the three main drum instruments in modern popular music: kick, snare, and hi-hat.

User interface The chosen VAE topology projects the input matrices to a two-dimensional space. As can be seen in Fig. 1, this rhythmic latent space is presented as a two-dimensional plane that the practitioner can play using the analogy of controlling a playback head. Clicking on any given point of the latent space will retrieve and decode a rhythm sequence. As expected, the patterns mimic closely those ones in the training data with the additional benefit of being able to interpolate between them by dragging the playback head, or to extrapolate to new ones when moving to a new zone in the space.

The user interface exposes two variables to control and add variability to the network decoding. *Threshold* controls the complexity of the patterns decoded from the latent space, and *noise* rules the precision of the mapping between the performance space to the latent space. Mute buttons per instrument allow the performer to silence instruments at any given moment. These parameters can be seen on Fig. 1 as part of the web-based player application. This browser-based version also features MIDI output so that performers can integrate R-VAE with external standalone devices and software-base sound engines.

A video demonstrating the capabilities of R-VAE and snippets of renditions performed with it can be accessed at `https://vimeo.com/422294058`. Both implementations of R-VAE, for Ableton Live[7] and the R-VAE-JS browser-based model player,[8] are available.

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

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

Discussion

We have investigated the viability of our system by encoding rhythms from music genres such as footwork, gqom, and trap, and using the models in musical improvisation and performance. When experimenting with data selection and model training, we found that the system 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 were richer and exhibited a more even topology, helping the performer to create smoother interpolations when moving the playback head between rhythms in different zones of the performance space. The use of R-VAE in creative practice demonstrates that its data representation is able to model accurately simple and compound meter rhythms, opening possibilities for using this tool with other rhythms exhibiting these characteristics.

Even if not perfect, individually crafted models from smaller datasets could prove to be useful and inspiring to the creative practitioner. The amount of data needed to generate creatively interesting models will vary with context and intention, but small data may be more suited to generate custom models that are good for exploring a specific idea or creative concept (Fiebrink, Trueman, and Cook 2009).

Interacting and performing with latent spaces encoding rhythms pose excellent questions, from both a computational and musical point of view. For example, how can we characterize the latent space in terms the *smoothness* of the interpolations? What is a good metric to measure the *richness* of the encoded space? Musical performance contains some pertinent differences to other fields unaddressed by current technical literature. For example, when measuring rhythmic similarity, the apparently small change of moving a few onsets from a 16^{th} grid to its closest triplets is small in terms of edit distance, but it can have a large perceptual impact. Furthermore, there are some instrumental hierarchies when working with rhythms. For example, changing a few hi-hats in a rhythmic sequence is likely to have a subtler perceptual influence than changing a drum kick pattern.

Conclusions and Future Work

We have presented R-VAE, a system designed for the exploration of latent spaces of simple and compound meter rhythms, a common combination in modern musical genres. The ability to use it with small datasets can enable the modeling of niche genres, while requiring fewer resources for training. We used R-VAE to learn models of a few modern music genre rhythms and used them in creative music production and performance. Some of the insights we learned are: (i) VAEs are capable of encoding compound meter rhythms; (ii) small training data is enough to create a useful and playable rhythmic latent space; but (iii) if the data is too small, the space can be perceived as a discrete collections of zones, instead of a contiguous space.

Research is needed to overcome the issues found and improve the system. For example, visual feedback may help performers to visualize the topology of the latent space, so that they know if they are in zones with specific rhythms or in zones of transition. Along the same lines, and in particular for small datasets, displaying where the original training data points are encoded in the latent space may provide visual guidance to explore the space. As a result, there is a pressing question about how to best incorporate this visual feedback into the system. Additional improvements are useful and needed. The web-based application may benefit from making the training process available directly in the browser, so that there is no need for an additional application. Extra playability and variability of the rhythmic patterns can be achieved by extending the number of bars encoded in the space and the number of drum instruments.

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. Systems like this could be also used for browsing through libraries of rhythms, common in contemporary music production.

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

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

Choi, K.; Fazekas, G.; and Sandler, M. 2016. Text-based LSTM networks for automatic music composition. In *Proceedings of the 1st Conference on Computer Simulation of Musical Creativity*.

Dinculescu, M.; Engel, J.; and Roberts, A. 2019. MidiMe: Personalizing a MusicVAE model with user data. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*.

Fiebrink, R.; Trueman, D.; and Cook, P. R. 2009. A meta-instrument for interactive, on-the-fly machine learning. In *Proceedings of the 9th International Conference on New Interfaces for Musical Expression*, 280–285.

Gillick, J.; Roberts, A.; Engel, J.; Eck, D.; and Bamman, D. 2019. Learning to groove with inverse sequence transformations. In *Proceedings of the 36th International Conference on Machine Learning*.

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

Nikolov, S. 2016. NeuralBeats: Generative techno with recurrent neural networks. `https://bit.ly/nikolov-neuralbeats`. Accessed: 2020-05-01.

Roberts, A.; Engel, J.; Raffel, C.; Hawthorne, C.; and 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*.