SEQUENCE TO SEQUENCE DRUM FILLS GENERATION AND DETECTION IN SYMBOLIC DOMAIN.

Frederic Tamagnan and Yi-Hsuan Yang
Research, Center for IT Innovation, Academia Sinica, Taipei, Taiwan
frederic.tamagnan@gmail.com, yang@citi.sinica.edu.tw

ABSTRACT

Percussions and drums are a fundamental core aspect of music, they are responsible for the rhythm and groove of song. It is what makes people want to dance to music as well. Drum fills are essential in the drummer's playing, they regularly restore energy and dynamic to a song. Moreover, they can prepare and announce the transition to a new part of the song. They build-up the tension before releasing it. These drums fills must answer to the main regular pattern by proposing a break, which has to be coherent. This aspect of the drums has not been explored much in the field of music information retrieval because of the lack of datasets with drum fills labels. Some datasets are made up of independent loops of regular patterns or drums fills. But there is no datasets with drum fills labeled along the entire drum track of a song. Moreover, drum kits dataset are relatively small . In addition, as drum fills creation don't follow any precise rule, it is hard to identify them. In this paper, we propose two methods to detect drums fills - one rule-based and one machine-leaning based- along a song in a big dataset, to obtain drum fills context information . In a second part, we propose a model for generating drum fills, given the regular pattern that took place previously.

1. INTRODUCTION

Music generation could not be considered without focusing on drums. One important part of drums generation that appears when one deals with long-term music generation is the drum fills question. In recent work on music generation using generative deep learning models, the generation of drum fills has often been treated implicitly. This might be explained because there are few datasets with labelled drum fills. In addition, the majority of them consists of battery kits for music producers. The regular patterns and drum fills are therefore separated into independent loops with no link between them. In addition, in the large datasets usually used for music generation, drum fills are a minority share of drum tracks. This justifies why they can be forgotten by the models.

So the main challenge of drum fills generation comes from the lack of labelled data as explained above. There is therefore an important preliminary task to be accomplished, which is the detection of drum fills, and in our case, in symbolic music datasets

The second tricky issue that comes in mind when dealing with drum fills is the lack of rules that defined them. A drum fill extracted from a quiet folk song could perfectly as a regular pattern in a jazz track with dense and complex rhythms.

According to Scott Schroedl [9] in his drumming method, a drum fill is "short break in the groove–a lick that 'fills in the gaps' of the music and/or signals the end of a phrase. It's kind of like a mini-solo". There are few theoretical elements or composition rules giving a framework to the definition of drums fills. That constitutes the main difficulty concerning prediction or generation of those. Nevertheless, our empirical observations can lead to these properties:

- a use of toms, snares or cymbals, more important than on the regular drum pattern
- a difference of played notes between the regular pattern and the drum fills
- an appearance in general at the end of a cycle of 4 or 8 bars

The task of detecting and generating drum fills explicitly, which can be useful for several reasons:

- segment the parts of a music piece, as important drum fills are often located as a transition between two parts of a song, from the verse to the chorus for example.
- allow the generation of long music sequences, in order to be able to create drum patterns with real evolution and ruptures.

2. RELATED WORKS

2.1 Drums detection and generation

Lopez-Serrano et al. [7] have proposed a method to detect drum breaks in the audio domain. In this paper it is not a question of detecting short drum fills but rather percussion-only passages . The authors address this problem inspired by a baseline method initially designed for

singing voice detection. In order to detect frames that contain percussion-only passages , they use features in the audio domain as included in [6], and a random forest model to define a median filtered decision function over the frames, and then apply a decision threshold.
Roberts et al. [8], writed a paper about learning and generating long term structure music. Recurrent VAE having difficulties to model a piece of music made up of several bars, they use a VAE including a hierarchical decoder. The VAE encoder produces a latent vector from n bars using a bi-directional recurrent layer .The first level of the decoder, the conductor, generates a serie of embedding vectors from the latent vector, each corresponding to a bar. A second level of recurrent decoder decodes these embeddings into notes sequences. The most interesting thing in this paper related to our topic, is that their model is able to produce drum fills implicitly as we can hear in their demo files.

A closer work from our topic is the PhD thesis of Vogl [10], focusing on automatic drum transcription (ADT), and generation in the symbolic domain. Vogl deals with recurrent, convolutional and recurrent convolutional architecture to solve the ADT problem, and provides methods able to detect drums intruments until a range of 18. Then Vogl tackles drum generation problem focusing first on drums patterns variations with Restricted Boltzmann Machines, using Gibbs sampling to generate variations of patterns. In the last part of the thesis, Vogl use CNN-based GAN coupled with RNN cells to generate long term sequences of drums.

2.2 Sequence-to-sequence RNN with semantic VAE

Most of drums generation papers use generative models based on sampling from gaussian noise for the inference part. Related to our sequence to sequence topic, one interesting approach is the introduction of sequence-to-sequence RNN model combined with a VAE. It allows to feed a compressed view of the global input sequence at each time step to the RNN, helping the reconstruction. The strength of these models is the use of two different representations of the data as input : the original raw data and a latent vector, guaranteeing that the RNN cell will keep an overview of the input data and strengthening the ability of advanced RNN architecture such as GRU or LSTM to remember important information. This will also reinforce the consistency between input and output data.

Ha and Eck [3], for the purpose of unconditional and conditional generation of sketch drawings, use this kind of architecture. They use a sequence representation of a sketch : each timestep includes x and y coordinates and a binary one-hot vector of three possible states that indicates if the the pen is beginning the sketch, touching the paper and joining two points or leaving the paper. After an unconditional training over sketch inputs, they are able to complete a sketch, reconstructing a sequence S' from a human-input sequence S.

2.3 Data representation

We decide to work at the bar level, our goal is not to predict drums fills with precise boundaries, but to predict if a bar contains a drums fill. As [8] for drum patterns, we mapped the 61 drum classes defined by the General MIDI standard to 9 canonical classes. We work only with bars having a 4/4 time signature. We decide to work with a precision of 4 timesteps for each beat. So that, it gives us a tensor with a 9*16 dimensions.

2.4 Dataset

2.5 Labelled Datasets

The datasets we have at our disposal don't have bars labeled along a song, they only provide isolated loops. The two datasets we use, are extracted from the Native Instruments Battery Kits and from the oddgrooves.com website and have the following distribution :

Dataset name	Regular patterns bars	Drum fills bars
NI	5,000	300
OG	0	900

The concatenation of these two datasets gives us a dataset with 5000 regular patterns and 1200 drums fills.

2.6 Unlabelled Dataset

The datasets we would like to label is the Lakh Pianoroll Dataset, which contains 21,425 songs with their related drums pianorolls

3. CLASSIFICATION OF DRUMS FILLS

We decide to use our labelled dataset to train a logistic classifier in order to predict drums fills.

3.1 Features used

3.1.1 Handcrafted features

For each bar of our labelled dataset, we decide to compute the maximum, the standard deviation and the mean of the velocity for each instrument of the reduced drums classes along the time axis. We use also a vector that represents if a reduced drum class is used or not during the bar with a 0 or a 1. It give us a 36 dimensions-vector. This vector represents the use,the amount of notes and the dynamic of playing of each drum class for each bar.

3.1.2 VAE's Latent Space features

We trained a Variational Auto-Encoder [5] on the lakh pianoroll dataset [2], to obtain features that captures a good representation of the drums patterns. Then, we use the encoder of this VAE to encode the data of our labelled dataset. It gives us a 32 dimensions-vector. We train a logistic classifier with both L1 and L2 regularization on our whole labelled dataset. We use standardization as preprocessing of our data and automatic-cross validation to tune the regularization hyperparameter as it is provided in sklearn.

3.2 Validation

Features Used	Precision	Recall	F1 Score
HD	0.90	0.90	0.90
LS	0.90	0.90	0.90
HD+LS	.90	0.90	0/90

Table 1. HD : Handcrafted features, LS : VAE's latent space features

We can have also have a look on the most correlated coefficients of the lasso-regression. TABLE 2

Most correlated feature	value of regression coef
6th,7th latent space variable	5
max velocity of mid tom	1.514
std velocity of low tom	1.514
mean velocity of open hi-hat	1.514

Table 2. Most correlated features with drums fills in logistic regression with lasso reg

These most correlated features confirms our empiric intuitions that drum fills are related with the use of toms and cymbals.

3.3 Discussion

We use our labelled dataset to predict drums fills. As the Native Intruments dataset and the oddgrooves dataset contain ideal drums fills, and not only regular pattern with the simple add of crash notes, we are certain to extract highly enriched and complex fills. The drawback of this approach is that we use a global approach, not a intrinsic one. Fills can be seen as a variation regarding the regular pattern of the song they belongs to.

4. LABELLING AND EXTRACTION

We apply the supervised method to our unlabelled dataset.

4.1 evaluation of labelling

In order to evaluate our labelling, we decided to compute several metrics for each track :

- the minimum, the maximum and the mean length of sets of adjacent fills divided by the amount of fills
- the global count of fills divided by the amount of fills
- the proportion of fills that are located on a "fourth bar" (end bar of 4 bars)

4.2 Extraction

Then, from this labels, we extract a dataset to allow us to proceed to drums fills generation. We decide to extract all the couple regular pattern, drums fills pattern in our labelled LPD dataset.

It give us for each approach

	Supervised approach	Clustering approach
Size	30,000	85,000

Table 3. Size of drums fills generation dataset with two differents methods

5. GENERATION OF DRUM FILLS

Our main goal is to generate a bar containing a drum fill, conditioned by a previous bar containing a regular pattern. As pianoroll representation of drum patterns is very sparse data structure. The models we tried to train with binary cross entropy as the only loss, always failed, generally leading to the prediction of a drum fill pianoroll filled with zero. One of our experiments was also to work in a latent space of a trained Variational Auto-encoder, by predicting the drum fills latent vector from a regular pattern latent vector and then decoding it. This second case led to generated drum fills that were inconsistent with the regular pattern given as input.

5.1 Model Architecture

Our model is a sequence to sequence variational auto-encoder, whose architecture is similar to [1,3,4] , the architecture predicts a new sequence S' from a input sequence S with a RNN network, conditioned at each time step by latent vector representing a bar view-level of the input sequence thanks to a semantic variational encoder. The latent space is not learned separately but during the training, that allows our model to learn the latent representation the most useful to help the sequence-to-sequence part.

5.1.1 Encoder

We use a birectional GRU layer to encode our input sequence in the forward and in the backward direction. Rather than use only the last hidden state of each directional GRU, we concatenate all the hidden state to obtain a vector h, in a attention mechanism way, as described in [4]

We then process the hidden state through batch normalization layers and fully connected layers to get two vectors μ and, after an exponential operation, σ of size d_z (in our architecture $d_z = 32$). In the same way as in the VAE, we make a random vector $z \in \mathbb{R}^{d_z}$:

$$z = \mu + \sigma \odot \mathcal{N}(0, 1)$$

5.1.2 Sequence 2 sequence RNN

Inspired by [3], we initialize the hidden states with the output of of a single network $h_{-1} = tanh(W_z \times z + b_z)$. We feed the GRU with a concatenated input of S_i and z at each step.

Then we concatenate all the hidden states into a $2 \times 16 \times 32$ tensor and feed batch normalization layers along the sequence axis and a fully connected layer to retrieve a $2 \times 16 \times 9$ tensor.

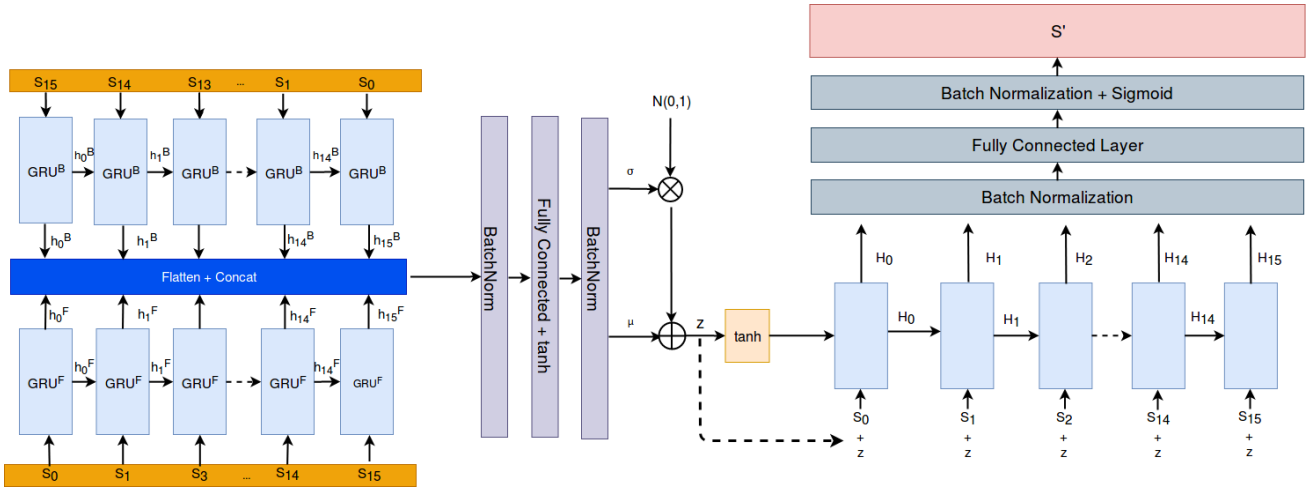

Figure 1. Schematic architecture of our model

5.2 Training

6. LEARNING FROM FAILS

7. CITATIONS

[**?**], or for a range [?, **?**, **?**].
[**?**]

8. REFERENCES

[1] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. *CoRR*, abs/1511.06349, 2015.

[2] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and yi-hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment, 09 2017.

[3] David Ha and Douglas Eck. A neural representation of sketch drawings. *CoRR*, abs/1704.03477, 2017.

[4] Myeongjun Jang, Seungwan Seo, and Pilsung Kang. Recurrent neural network-based semantic variational autoencoder for sequence-to-sequence learning. *CoRR*, abs/1802.03238, 2018.

[5] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[6] B. Lehner, G. Widmer, and R. Sonnleitner. On the reduction of false positives in singing voice detection. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7480–7484, May 2014.

[7] Patricio López-Serrano, Christian Dittmar, and Meinard Müller. Finding drum breaks in digital music recordings. In Mitsuko Aramaki, Matthew E. P. Davies, Richard Kronland-Martinet, and Sølvi Ystad, editors, *Music Technology with Swing*, pages 111–122, Cham, 2018. Springer International Publishing.

[8] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. *CoRR*, abs/1803.05428, 2018.

[9] Scott Schroedl. *Play Drums Today!* Hal Leonard, 2001.

[10] Richard Vogl. *Deep Learning Methods for Drum Transcription and Drum Pattern Generation*. PhD thesis, 11 2018.