Drum Fills Detection and Generation

Frederic Tamagnan and Yi-Hsuan Yang *

Academia Sinica, Taiwan
frederic.tamagnan@gmail.com
yang@citi.sinica.edu.tw

Abstract. Drum fills are essential in the drummer's playing. They regularly restore energy and announce the transition to a new part of the song. This aspect of the drums has not been explored much in the field of MIR because of the lack of datasets with drum fills labels. In this paper, we propose two methods to detect drum fills along a song, to obtain drum fills context information. The first method is a logistic regression which uses velocity-related handcrafted data and features from the latent space of a variational autoencoder. We give an analysis of the classifier performance regarding each features group. The second method, rule-based, considers a bar as a fill when a sufficient difference of notes is detected with respect to the adjacent bars. We use these two methods to extract regular pattern/ drum fill couples in a big dataset and examine the extraction result with plots and statistical test. In a second part, we propose a RNN model for generating drum fills, conditioned by the previous bar. Then, we propose objective metrics to evaluate the quality of our generated drum fills, and the results of a user study we conducted. Please go to https://frederictamagnan.github.io/drumfills/ for details and audio examples.

Keywords: Drum fills detection, Drum fills generation

1 Introduction

Percussions and drums are a fundamental core aspect of music. One important part of long-term drums generation is the drum fills issue. In recent works on music generation using generative deep learning models, drum fills have often been treated implicitly. The main challenge of drum fills generation comes from the lack of labelled data. So that, drum fills detection is an important preliminary task. The second tricky issue that comes in mind when dealing with drum fills is the lack of rules that defined them. Nevertheless, our empirical observations can lead to these properties: 1) a greater use of toms, snares or cymbals, than in the regular drum pattern; 2) a difference of played notes between the regular pattern and the drum fills; 3) an appearance in general at the end of a cycle of 4 or 8 bars.The task of detecting and generating drum fills explicitly has at least the following two use cases : first, segmenting the parts of a music piece,

* This work was done when FT was a visiting student at Academia Sinica.

as important drum fills are often located as a transition between two parts of a song, from the verse to the chorus for example; second, allowing the generation of long music sequences, in order to be able to create drum patterns with real evolution and ruptures.
In this paper, we present an initial attempt towards generating drum fills. Our goal is first to address drum fills detection and to build-up a dataset of regular pattern/drum fills couples (Figure 1). Secondly, we use this dataset to train a model able to generate a drum fill based on a regular pattern. In particular, this work allows us to answer three research questions: (1) Can we train a fill detector from isolated fills? or is it mandatory to take into account the context? (2) Is a rule-based method effective enough to detect fills? (3) How objectively a human can rate a drum fill? In sections 4–5, we develop two methods to detect and classify drum fills. The first is a logistic regression based on two different group of features: velocity-related handcrafted features and variables from a variational auto-encoder latent space. The classifier has been trained on drums kits from Native Instruments and OddGrooves.com with regular pattern and drum fills labels. The second method is a rule-based method that reflects the interpretation of a drum fill as a variation. Then, in Section 6 using these two classifiers, we extract regular pattern/ drum fills couples in the Lakh pianoroll dataset to build-up two generation datasets. After cleaning these extracted datasets to provide clean and balanced enough datasets to our further generation model, we evaluate the extraction. Our generation model, whose architecture is precisely described in Section 7, is able to generate a drum fill based on the regular pattern given as a input. We use a many-to-many RNN with 2 layers of GRU units, followed by fully-connected and batch-normalization layers. Section 8 shows the results of the user-study we have conducted with musicians and non musicians where our model trained on our two different datasets is confronted with a rule-based method to generate drum fills.

2 Related Works

Lopez-Serrano *et al.*. [5] 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.* [3], wrote a paper about learning and generating long term structure music. Recurrent Variationnal Auto-Encoder (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 series 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.

3 Preliminaries

In this paper, we do not care about the precise boundaries of a drum fills. To simplify the problem, we decide to detect and generate bars containing drum fills. We also reduce the problem by working with only 9 different drums instruments as [3]: kick (abbreviated BD for bass drum), snare (SD), low, mid and high tom (LT,MT and HT), closed and open hi-hat (CHH and OHH), crash and ride cymbals (CC and RC). We work only with bars having a 4/4 time signature. We decide to work with a precision of 4 time steps for each beat. This gives us a tensor with a 9×16 dimensions filled with the velocity of each note. In the next sections, we use the term "reduced pianoroll" to call a pianoroll transformed to a 9×16 tensor and "binarized pianoroll" to call a pianoroll filled with 0 and 1 instead of velocity.

3.1 Datasets

Labelled Datasets The Native Instruments' Battery Kits and the oddgrooves website's fill pack are composed by loops with different time signatures and length. We decided to crop and add paddings to these loops to form bars with a 4/4 signature. The concatenation of these bars from the two datasets gives us a dataset composed of 5,317 regular patterns and 1,412 drum fills.

Unlabelled Dataset The dataset we would like to label is the Lakh Pianoroll Dataset [2], a derivative of Lakh Midi Dataset [1], which contains 21,425 songs with their related drums pianorolls.

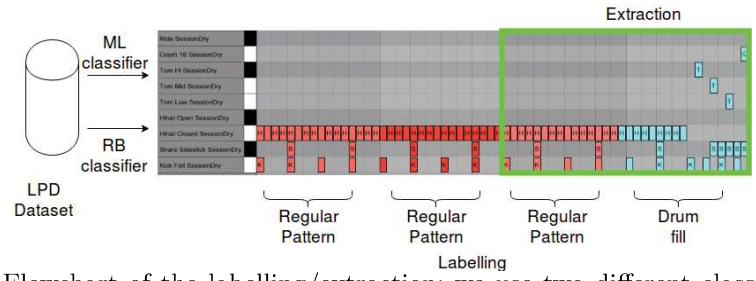


Fig. 1: Flowchart of the labelling/extraction: we use two different classifiers to label the LPD dataset. Then, we extract regular pattern/drum fill couples to constitute a drum fills dataset for generation

4 Machine Learning Classifier

4.1 Features Used and Model

We use two groups of features to train our model. 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. It give us a 27 dimensions-vector. This vector represents the use, the amount of notes and the dynamics of playing of each drum class for each bar. We trained a VAE over thousands of bars of the Lakh pianoroll dataset [2], to obtain features that capture a good compressed representation of the drums patterns. Then, we use the encoder of this VAE to encode the data of our labelled dataset and to obtain the latent space features. It gives us a 32-dimensions vector.

We train a logistic classifier with regularization on our whole labelled dataset using *LogisticRegressionCV* from the Sklearn API [7]. We use standardization as pre-processing of our data and automatic-cross validation to tune the regularization hyperparameter.

4.2 Validation

Using the $L2$ regularization that performs better in our case, we obtain the result shown in Table 1.

Feature set	Precision	Recall	F1 Score
HD	0.80	0.79	0.79
LS	0.58	0.06	0.10
HD+LS	0.89	0.81	0.85

Table 1: Validation metrics of our classifier. HD: Handcrafted features, LS: VAE's latent space features

The results for the VAE's latent space features are low because there were few drum fills compared to regular patterns in the VAE's training dataset. So that, latent space features badly capture the essence of drum fills. Although the training with $L1$ regularization has worse performance results, it is interesting to have a look on the weights, to see which features are the most correlated with the purpose of detecting fills. The three most correlated LS features are the 18th, 20th and 1st latent space variables ; they are associated with the regression coefficients 2.06,1.92 and 1.61. The three most correlated HD features are the max velocity of high tom, the standard deviation of mid tom and the max velocity of low tom ; they are associated with the regression coefficients 1.26, 1.26 and 1.26. That confirms our intuition that fills are related with toms and cymbals and that gives us a better comprehension of our VAE. The drawback of this approach is that we characterize a drum fill with absolute rules, and not with the relative difference between bars.

5 Rule-based Classifier

Fills can be seen as a variation regarding the regular pattern of the song they belong to. In order to answer to research question 2, we build another approach, rule-based. Let A, B be two binarized pianorolls (tensors) of dimension $t \times n$ (time steps $\times$ number of instruments), we define the difference of notes DN between A and B as:

$$DN(A,B) = \sum_{\substack{0 \leq i < t \\ 0 \leq j < n}} max(0, A_{i,j} - B_{i,j}) \tag{1}$$

Iterating over the binarized and reduced bars of our unlabelled dataset, we decide to consider the current bar as a drum fill if the difference of notes between the current bar and the two adjacents bars respectively is above a threshold. We use a threshold of 7 notes for the extraction part.

6 Extraction of Fills

We apply our machine learning classifier and our rule-based classifier to our unlabelled dataset. Then, we extract 2-bars sequences composed by a drum fill following a regular pattern. So, we obtain two datasets from our two labelling methods that we will call ML dataset (extracted with the machine learning classifier) and RB dataset (extracted with the Rule-based classifier).

6.1 Data Cleaning

In order to have a good enough datasets for the generation we apply the three following rules (Table 2) to clean our datasets: removing duplicated rows (Rule 1), removing all the couples where the regular pattern or the drum fill have fewer than 7 notes (Rule 2), removing all the couple where the drum fill has a too high density of snare notes, above 8 (Rule 3) .

	#ML dataset	#RB dataset
Raw	13,476	97,023
After Rule 1	6,324	45,723
After Rule 2	5,271	39,108
After Rule 3	3,283	32,130

Table 2: Influence of the cleaning process on our datasets size. The RB dataset is less sensitive to our filtering rules.

6.2 Analysis of the Extracted Datasets

Total of Notes by Instrument A Pearson's chi-squared test between the total of notes by instrument of the regular patterns and the drum fills certifies us that the distributions are significantly different for the two extracted datasets. The drum fills of the two datasets contains more toms and cymbals notes than the regular patterns, as we can see for example in the Figure 2.

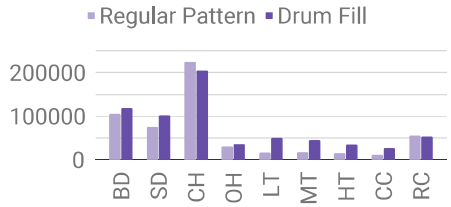


Fig. 2: Amount of notes by instrument for the ML dataset

Proximity of Drum Fills We want our extracted drum fills not too close to each other, as we consider that fills only appear every 4, 8, or 16 bars. Thus, we compute the length of the longest serie of adjacent fills in every song of our dataset. The average lenght is 0.68 and 1.90 for the ML dataset and the RB dataset respectively. The perfect result would be 1, so it is close to what we expect.

Distribution of Genres We compute the average amount of fills extracted over genres. We expect to find more fills in the following genres: Metal and Jazz. Our RB dataset follows well this intuition but this is not the case for our ML dataset.

7 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. We decide to use an architecture often found in the Natural Language Processing state-of-the-art, many-to-many Recurrent Neural Networks (RNN), whose architecture is described in Figure 3.

7.1 Training

We train our model over 300 epochs with a batch size of 4096 for the RB dataset and 256 for the ML dataset. We use Adam [4] as optimization algorithm with a learning rate of 0.001 and binary cross-entropy as loss function. We remove from each dataset the intersection of the two datasets which we use later as a test dataset, in order to evaluate the model trained on different datasets. We use a split of 80/20 for the training/validation datasets.

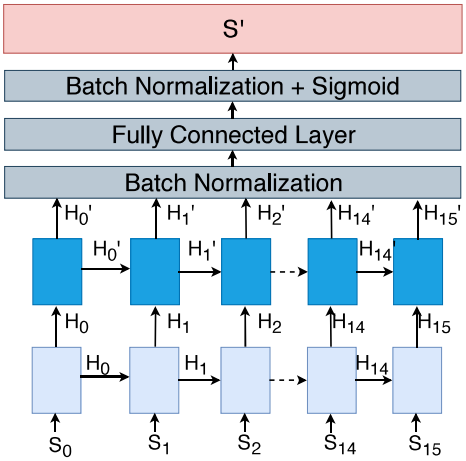


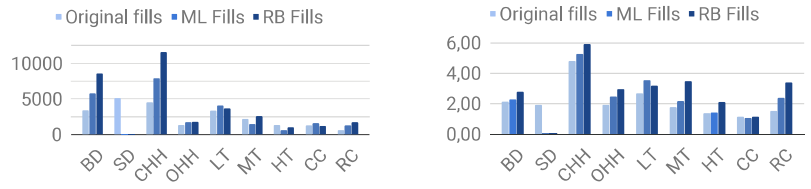
Fig. 3: Our model architecture is composed by two stacked GRU layers followed by batch norm and fully connected layers

7.2 Evaluation

We use the test dataset to generate two set of fills with our model trained on the ML dataset and the RB dataset. Then, we compare the original fills (ground truth fills) from our test dataset with the two other sets of generated fills (ML fills and RB fills)

Total of Notes by Instrument Applying the Pearson's chi-squared test between the total of notes by instrument of the three datasets (pair-wise), the p-value is less than 0.01, that shows that the fills are different in the three sets.

As main differences, we can see that the RB fills includes more bass drum and closed hit-hat than the other sets of fills. The ML fills include more low tom notes than the other set of fills as well. Unfortunately, our datasets, substracted from their high-density snares fills, do not allow us to create drum fills with snare rolls.



(a) Total (b) Standard-deviation

Fig. 4: Total and standard-deviation of the amount of notes in the generated/original fills by instrument

Diversity of Fills We take each set of fills, and we encoded them in the VAE's latent space of the Section 4.1. So, for each set of fills, we compute the sum of

the Euclidean distance between fills (pair-wise) as a measure of diversity for each set of fills. And give us 93012, 93844 and 102135 for the ML fills, the RB fills and the original fills respectively. We also compute the standard deviation of the amount of notes by instrument . From these two perspectives, we can see that the RB fills are more diverse than the ML fills but less diverse than the original fills.

8 User Study

Finally, we conduct a user study involving 51 participants (66% of musicians) recruited from the Internet. After a small test to know if participants are able to recognize a fill from a regular pattern (two people did not pass the test), people were asked to listen to 4 pieces of drums including a regular pattern repeated three times and then a drum fill. Two pieces of drums come from the ML fills and the RB fills respectively, the two other are the original fill and a "Rule based composed fill" (RC fill). The RC fill is the original regular pattern with the same toms/crash pattern added each time. We report in Table 3 the results. We can see that our methods do not beat a human for the task of composing a drum fill, even when the fill is composed with the same rule. Nevertheless, the RB fills are getting closer from the original fills. The original fill is not rated with a good grade in our experiment. In other words, human listeners do not think the human composed fills are good enough. Additionally, the RC fill has almost the same grade as the original fill. This indicates the subjective nature of the task and answers to the research question 3.

	ML	RB	Original	RC
Overall grade	2.61	2.90	**3.13**	3.10
Most coherent	17%	18%	29%	**36%**
Less coherent	**30%**	**30%**	23%	18%
Best groove	13%	25%	**34%**	28%
Worst groove	**35%**	30%	18%	17%

Table 3: Results of the user study, averaged over 49 subjects. The mean of the five-point scale grade is given in the first line. For the rest of the lines, the ratio of vote is given.

9 Conclusion

We have presented several axes to research in the field of drum fills detection and generation. We have shown the importance of considering a fill as a variation rather than through an absolute view. The results of our generation pipeline (detect drum fills with a rule-based method and then generate them with an RNN) are getting closer from the human-composed fills. In future work, we will explore fusion-method that combines machine-learning and rule-based method to improve the results of the drum fills detection with the help of more hand labelled data.

References

1. Raffel,C.: Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching. PhD Thesis (2016)
2. Dong, H.W.,Hsiao, W.Y., Yang, L.C., Yang, Y.H.: MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In:Thirty-Second AAAI Conference on Artificial Intelligence (2018)
3. Roberts, A., Raffel,C. , Engel,J., Hawthorne, C., Eck, D.: A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. In: ICML 2018 (2018).
4. P. Kingma, D., Ba,J.: Adam: A method for stochastic optimization. In: ICLR 2015 (2015).
5. López-Serrano, P., Dittmar, C., Müller, M.: Finding drum breaks in digital music recordings. In: International Symposium on Computer Music Multidisciplinary Research (2017)
6. Lehner, B., Widmer, G., Sonnleitner, R.: On the reduction of false positives in singing voice detection. In: (ICASSP) (pp. 7480-7484). IEEE. (2014)
7. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Layton, R. (2013). API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning ,pp 108–122 (2013)
8. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: The International Conference on Learning Representations (2014)
9. Dong, H. W., Yang, Y. H.:. Convolutional generative adversarial networks with binary neurons for polyphonic music generation. In: ISMIR (2018)
10. Schroedl, S.: Play Drums Today. Hal Leonard. ISBN 0-634-02185-0 (2001)