Data Augmentation for Drum Transcription with Convolutional Neural Networks

Céline Jacques

Analysis-Synthesis Team
STMS-UMR 9912, IRCAM, Sorbonne University, CNRS
Paris, France
celine.jacques@ircam.fr

Axel Roebel

Analysis-Synthesis Team
STMS-UMR 9912, IRCAM, Sorbonne University, CNRS
Paris, France
axel.roebel@ircam.fr

Abstract—A recurrent issue in deep learning is the scarcity of data, in particular precisely annotated data. Few publicly available databases are correctly annotated and generating correct labels is very time consuming. The present article investigates into data augmentation strategies for Neural Networks training, particularly for tasks related to drum transcription. These tasks need very precise annotations. This article investigates state-of-the-art sound transformation algorithms for remixing noise and sinusoidal parts, remixing attacks, transposing with and without time compensation and compares them to basic regularization methods such as using dropout and additive Gaussian noise. And it shows how a drum transcription algorithm based on CNN benefits from the proposed data augmentation strategy.

Index Terms—data augmentation, deep learning, drum transcription, convolutional neural network CNN

I. INTRODUCTION

The present paper deals with data augmentation strategies and their application to automatic transcription of drum events with deep neural networks given a limited set of training examples.

Automatic drum transcription in music recordings consists in annotating the time position at which a drum event occurred and specifying which drum instrument was played.

In [1], they noticed that the onset detector based on Convolutional Neural Network (CNN) learned different features for harmonic onsets and percussive onsets. So the problem of drum transcription can be seen as a detection of qualified onsets and has therefore close relationship to the more general problem of finding onsets in music. For the onset detection task, deep learning reached the best results in 2017 at MIREX onset detection task over all the MIREX evaluation campaigns[1].

On the other hand, drum transcription can also be considered as a problem in itself. While some methods present very good results, e.g., [2] with Hidden Markov Model (HMM), methods based on deep learning seem to be more and more common. In [3], a Recurrent Neural Network (RNN) is trained to calculate the activation functions of different drum instruments. The first study of CNN for drum transcription was performed in [4]. Now CNN are widely used for drum detection [5], [6] and got the best results during the MIREX2018

drum transcription challenge[2]. A review of automatic drum transcription challenge is available in [7].

A recurrent issue for deep learning is the scarcity of annotated data. In image processing, for example, if an object has always been presented to the network with the same orientation, it may encounter difficulties to detect it when the object is moved in other direction. For music similar issues appear when the available databases do not cover sufficient possibilities. The way an instrument sounds varies greatly from one music track to another. In fact, because getting precise annotation of data is very difficult and generating them time consuming, the amount of publicly available data with precise annotations will hardly contain as much variety as is present in the real world. For training a Deep Neural Network (DNN) nearly all the information comes from annotated data. Therefore the more data with relevant variability one can provide, the better algorithm will work.

Besides other strategies such as student/teacher paradigm [8], data augmentation can be a solution to bridge the lack of data. It consists in applying different transformations on annotated sounds to create new data labeled with similar precision. However it is necessary that the augmented data keeps the features of the label. As all information is encoded in the data, high quality sound transformation is essential to avoid training the network on irrelevant artifacts that are introduced by means of the transformation algorithm. In extreme cases training on such artifacts may lead to algorithms which work only with transformed data. Because human perception is the ultimate target when playing music instruments, it appears a reasonable first approach to validate augmented data by means of listening.

Some transformation methods with preservation of label meanings have been proposed in [9], [10] such as pitch shifting or time stretching. [11] investigates into transcribing 18 classes of drum instruments. Because the database does not present a balanced distribution of the percussive onsets over the different classes, it is re-synthesized to obtain a better distribution of the 18 classes. In [6], training data are created by synthesizing sounds from midi files using three different soundfonts. In [10] data augmentation is achieved

---

[1]https://nema.lis.illinois.edu/nema_out/mirex2017/results/aod/

[2]https://www.music-ir.org/mirex/wiki/2018:Drum_Transcription_Results

by transforming the magnitude spectrogram of the input by scaling the spectrogram images. In [12], the MUDA toolbox [9] is used to transform environmental signals for the task of environmental sound classification.

In this article, the use of 4 different transformations of sound signals will be evaluated in the context of a drum transcription task. The transformations are particularly selected to transform features that seem relevant for detecting drum events: remixing attacks, remixing noise and sinusoidal parts, as well as transposition with and without time compensation and with varying degrees of transformation of the spectral envelope. It will be compared to two other methods often used for regularisation of DNN: the use of dropout and adding a white noise to the input spectrograms.

In this article, we also focus on the choice of augmentation strategies and their parameters. The recall is always expected to increase if some variability is introduced in the database by data augmentation while precision might decrease if the parameter of data augmentation are not carefully chosen.

The article is organized as follows: Section II presents the model on which the studied networks are based and the different transformations applied to the database. Section III presents results with those augmented databases. We finally draw conclusions in section IV.

II. METHODOLOGY

A. Pre-processing

The networks are fed with a Multi-Channel Mel Spectrograms (MCMS) representation [6] initially proposed in [1]. MCMS represents the data in three log-magnitude mel band spectrograms. The short-time Fourier transform (STFT) is created with a hop size of 10 ms and three window sizes: 23, 46 and 93m s. The spectrograms are then filtered with an 80 mel-band filter covering the bands from 27.5Hz to 16kHz. Thus the input data are three mel-band spectrograms with 80 mel-bands and different time resolutions.

B. CNN topology

The networks used in this article are all based on the CNN topology introduced in [1]. The network topology shown in Fig. 1 alternates two stacks of convolutional layers with ReLU activation followed by max-pooling layers and ends with a fully connected hidden layer with ReLU units and the output layer containing either a sigmoid unit or a linear unit.

Each layer has different hyper-parameter such as filter sizes, number of filters, etc. The parameterization will be specified by means of a single character designating the type of layer operation (C: convolutional, M: max-pooling, F: fully-connected) followed by two or three numbers. In case of convolutional layers, the numbers specify filter dimension in time and frequency followed by the number of filters, for max-pooling, the pooling strides in time and frequency dimensions are specified, and for the fully connected layer, simply the number of hidden units. For the network in figure 1, the topology can be specified as

$$C_1: C1_t \times C1_f | N_{C1}, M_1: M1_t \times M1_f,$$
$$C_2: C2_t \times C2_f | N_{C2}, M_2: M2_t \times M2_f, \qquad (1)$$
$$F: N_F$$

That is a convolutional layer with N_{C1} filters of size $C1_t \times C1_f$ followed by a max-pooling layer with strides $M1_t \times M1_f$, then a convolutional layer with N_{C2} filters of size $C2_t \times C2_f$ followed by a max-pooling layer with strides $M2_t \times M2_f$ and in the last hidden layer F, N_F nodes fully-connected to the previous feature maps.

C. Basic regularization strategies

The strategies of data augmentation investigated in this article are compared to two regularization strategies: dropout and additive Gaussian noise. Because these methods are used similarly for image and sound processing, they are considered as data-independent methods in [10].

The use of dropout during the training consists in setting some of outputs of a layer to zero with a given probability. Because the dense layers connect all neurons from previous layer to all neurons of the dense layer, it potentially creates extremely specialized hidden units which in turn may lead to an over-fitting of the training database. Randomly cancelling some hidden units increases redundancy and avoids over-specialization of the network which is expected to learn more robust features. We note P_d the probability of dropout.

An other way to introduce variability in data and to strengthen the network to variations is to add Gaussian noise to the input spectrogram. The Gaussian noise of mean zero is defined by its standard deviation which we note σ.

Both data-independent method of regulation are applied on-the-fly during the training and do not intervene during evaluation and testing steps.

D. Data augmentation for audio

To provide the networks with augmented data, we experiment four transformations: remixing the noise part, remixing attacks, transposing with or without time compensation including various degrees of transposition of the spectral envelope. All transformations are applied directly to the audio signal before pre-processing to convert it into the MCMS representation. In addition, all transformations conserve ground truth and adjust it when the playback speed changes.

The four transformations described below are all performed with the signal transformation kernel available in version 3 of the AudioSculpt program [13] that can be scripted and controlled via the Unix command line:

- **Remix noise**: After classifying sinusoidal and noise peaks in the signal audio [14], they can be separated by means of spectral masking and remixed with selected mixing ratio. This leads to a modification of the balance between sinusoidal and noise components. Attacks, or onsets in general and percussive instruments in particular, contain
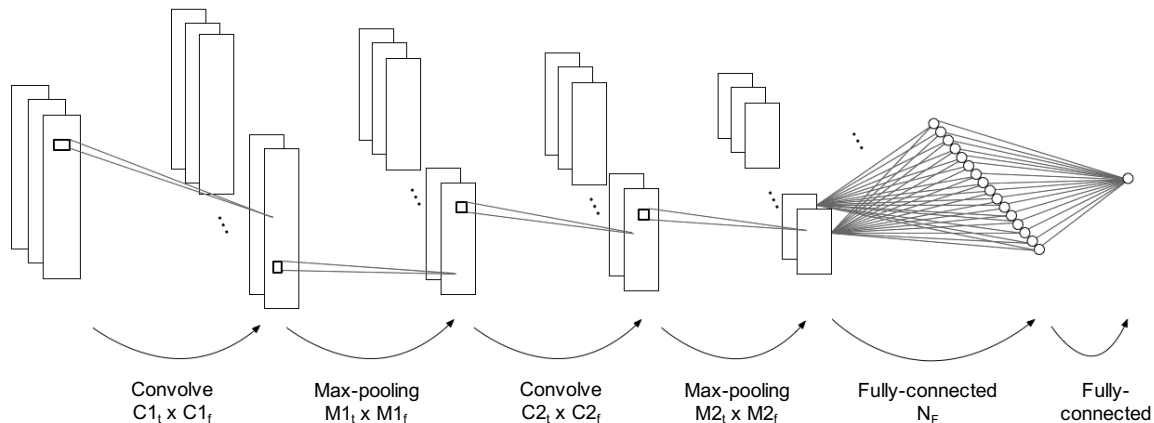
Fig. 1: *Network topology for onset detection and drum transcription.*

different amounts of noise and transient sinusoids (reso-nances). Remixing them changes the energy distribution between these two components and can therefore be seen as a transformation of onset properties. Fast onsets will neither be detected as sinusoid nor as noise and will be covered by the following effect.
Parameter: r_n designates the remixing factor applied to the spectral peaks detected as noise.

- **Remix attacks**: After detecting the time-frequency locations of transient bins [15], these bins are scaled by applying a linear factor to either soften or increase the transient strength. 100 ms after each transient, the factor fades to one. This transformation aims to modulate the attack properties, notably in comparison to the subsequent stationary or release phase of the sound event.
Parameter: r_a designates the remixing factor applied to the spectral peaks detected as transients.

- **Transposition with and without time compensation**: Transposition is achieved by means of resampling the signal to a new sample rate without changing the sample rate used to play the sound. The anti-aliasing filter used has 3 dB bandwidth up to 90% of the respective sample rate, and aliasing attenuation over -70dB. The resampling will scale all temporal relations in the signal leading notably to tempo changes, and faster evolution of attack transitions. These time scale changes can be compensated by applying a phase vocoder approximately following [16] additionally making use of the transient preservation described in [15].
Parameter: t transposition in cents that is achieved by means of the re-sampling operation and $t\,nc$ for transposition without time compensation.

- **Spectral envelope transposition**: For the transposition with and without time compensation, an additional transformation is added. The spectral envelope is estimated [17] and trans-posed while the pitch remains unchanged. This last effect

leads to time varying filters and sound color changes. It has the potential to further increase data variability.
Parameter: t_e transposition in cents of the spectral envelope.

Data augmentation will therefore be performed with 5 different setups: noise remixing only, attack remixing only, transposition with time compensation and envelope transpo-sition, transposition without time compensation and envelope transposition and finally all the transformation combined. The parameterization of the resampling operation will be used equivalently for transposition with and without time compen-sation and the combination with envelope transposition will always be performed covering all possible combinations.

E. Evaluation strategy

The evaluation of the data augmentation strategies is per-formed independently for each type of transformation by means of cross database validation experiments [18]. We train the CNN models on all but one databases and evaluate it on the remaining database. The training databases are augmented by one of the augmentation strategies.

The selection of the result of a training cycle is selected by means of early stopping. For this a few tracks of the original training databases are selected as validation data and all of the augmented sounds based on this validation subset are removed from the augmented versions. Evaluation with these examples serves to determine the best network from all the networks that have been obtained during the training phase. The validation set contains only original non-augmented data.

The experiments are realised with 3 different seeds for the initialization of the networks. The results are the average over those three initializations.

III. APPLICATION TO DRUM TRANSCRIPTION

For automatic drum transcription, we compare the different transformations on the training data of MIREX 2018 drum transcription challenge with a cross-fold validation. As what

is usually done in literature, we focus on the three main drum instruments: hi-hat (hh), bass-drum (bd) and snare-drum (sd). Those three instruments are responsible of most of the basic rhythms in western music.

A. CNN model

To perform the transcription of the three main drum instruments, three individual networks are trained independently. The three independent networks are based on the architecture of the network shown in Fig. 1. Each network aims to detect one and only one drum instrument. They all have the same parameters given by the topology following Eq. 1.

$$C_1: 7 \times 3 | 10, \ M_1: 1 \times 3$$
$$C_2: 3 \times 3 | 20, \ M_2: 1 \times 3 \qquad (2)$$
$$F: 256$$

To learn the weights, we use cross-entropy as loss function and Adam optimizer.

B. Augmented databases

Transformation parameters

Each transformation is controlled by a parameter. This parameter is chosen so that the augmented version of an event is still recognizable as the same event. Because there is no measure which can determine if an instrument still sounds as the same instrument after the transformation, the parameters were chosen by listening the transformed sounds. They are given in Table I and explained in Sec. II-D. For each value of the parameter t, the data are processed with all values of $t_e \in [-300, -200, -100, 100, 0, 100, 200, 300]$.

The probability of dropout on the third layer and the standard deviation of the Gaussian noise added to the input are also given in Table I.

In the following, the reference is the networks trained without any kind of data augmentation, i.e. no dropout, no additive Gaussian noise and trained only on original data.

Factor	Values
P_d	0.25, 0.5, 0.75
σ	0.05, 0.1, 0.2
r_a	0.1, 0.3, 0.6, 1.5, 2, 3
r_n	0.6, 1.5, 2, 3
t	-300,-200,-100,100,200,300
$t \ nc$	-300,-200,-100,100,200,300

TABLE I: *Transformation parameters for MIREX 2018 database.*

MIREX2018: Drum transcription training database

This dataset was provided for MIREX 2018 drum transcription challenge to train models. It is detailed on MIREX 2018 website[3]. It is made up of four subsets from different databases: 2005, GEN, MEDLEY and RBMA. They all contain annotated polyphonic pieces of music of different genres as well as drum only tracks.

From 3 hours of recordings, we obtain after transformations: about 22 hours for remix attacks, about 15,5 hours for

remix noise and about 116 hours for transposition with time compensation and 116 hours without.

C. Experiments

We give the results for each augmented version in the Table II. The results are given as the mean of the Recall, Precision and F-measures over the four cross validation steps and over the three different initialization. We ran five experiments.

1) **Original** We train the three CNNs detailed in section III-A on the original data without dropout or additive Gaussian noise.
2) **Dropout** We change the value of the probability of the dropout
3) **Gaussian noise** Three values of standard deviation are tested to generate Gaussian noise which are added to the input spectrograms.
4) **Audio transformation** The fourth experiment investigates the influence of the four data augmentation strategies explained at the section II-D.
5) **Transformations combined** The networks are trained on original data, the four augmented databases.

The drums are considered as correctly detected if they are closer than 50 ms from the ground-truth onsets.

D. Results

		BD			SD			HH		
		R	P	F	R	P	F	R	P	F
Orig.		87.4	73.8	80.2	67.5	54.0	59.0	78.6	64.3	69.8
Drop. P_d	0.25	88.2	74.2	80.6	66.5	54.3	58.9	78.6	64.0	69.6
	0.50	88.2	74.4	80.5	65.7	53.8	58.3	77.4	64.8	69.5
	0.75	88.4	73.9	80.2	67.3	53.3	58.7	78.6	64.5	69.8
Gaus. noise σ	0.05	87.8	73.7	79.9	67.3	54.1	59.2	79.8	63.8	70.0
	0.1	87.8	74.3	80.3	67.0	52.8	58.3	79.9	63.5	70.0
	0.2	88.1	73.8	80.1	67.2	52.7	58.2	**79.8**	**64.1**	**70.2**
Trans.	rn	90.2	73.5	81.6	**65.8**	**55.6**	**59.8**	78.4	64.3	69.7
	ra	90.2	75.0	81.7	67.0	53.7	59.0	78.7	63.9	69.5
	t	91.2	70.0	80.7	69.4	53.5	59.5	80.4	61.1	68.4
	t nc	**90.0**	**75.7**	**82.1**	68.3	53.6	59.4	80.4	61.7	68.8
All		90.6	74.0	81.2	67.7	54.3	59.6	79.6	62.3	68.9

TABLE II: *Results (F-measure) of drum transcription on MIREX 2018 database with dropout regularisation, with adding Gaussian noise, with the four strategies separately and the four strategies combined.*
ra: remix attacks, rn: remix noise, t: transposition, t noc: transposition without time compensation

First we describe results for the traditional regularization strategies. Using dropout of probability greater than 0 does not seem to improve the training at least for snare-drum and hi-hat. Although the results are better with probabilities of 0.25 and 0.5 for bass-drum than without dropout, they are still below or equal for the other instruments. Moreover the results are still below the ones obtained with remixing noise and sinusoidal parts or remixing attacks except for hi-hat.

Regularization by means of adding Gaussian noise to the input spectrograms is beneficial for all instruments, but does not achieve best results for bass-drum and snare-drum. On the other hand, adding noise is the most effective augmentation

[3]https://www.music-ir.org/mirex/wiki/2018:Drum_Transcription#Training_Data

strategy for hi-hat outperforming the results with all other augmentation methods for all noise levels. The best result for hi-hat is in fact achieved by means of adding Gaussian noise of standard deviation of 0.2.

Concerning data augmentation with the four signal transformation strategies introduced above it becomes clear that results are different for bass-drum and snare-drum (vibrating membranes with tunable resonances) and hi-hat (no resonance).

We first note that augmentation by means of transposition with time compensation always considerably increases recall, but at the same time decreases precision, to an extend that for the hi-hat the F-measure overall even decreases. This effect might indicate that the transposition up to 300 cents is to strong, even if perceptually the transformed sounds did still sound acceptable. The worst effect is observed for the hi-hat. It might be explained by the fact that the hi-hat is the drum instrument with the highest frequency range. So the same transposition leads to significantly increased frequency offset. It seems necessary to run instrument dependent adaptation of the transposition extents to get the best effect of the transposition for all instruments.

On the other hand, the transposition without time compensation does not improve only the recall for the bass-drum but it improves considerably the precision too. That leads to the best results we get over all augmentation strategies. In fact, without time compensation, the transposition does not introduce artifacts.

Augmentation by means of remixing the attack strength has a positive effect for all instruments except for hi-hat for which it degrades slightly the F-measure.

For the snare drum, remixing noise and sinusoidal parts leads to best results. Here the F-measure increases by 0.6 points in comparison to F-measure obtained with the model trained without any augmentation. We note that for snare-drum and hi-hat, it improves precision but degrades the recall.

It is interesting to note that augmentation with a combination of all signal transformation strategies is generally worse than the best strategy. This comforts the idea that careful adjustment of data augmentation needs to be performed taking into account the properties of the individual instruments.

IV. CONCLUSIONS

This article aimed to analyze the influence of adding augmented data to the training set. Moreover it compared several transformations between them as well as training with only original data and two data-independent methods which are dropout and additive Gaussian noise.

The different strategies were applied to a MIREX 2018 database for automatic drum transcription task. The first important result was the improvement of results of all experiment with at least one of the data augmentation strategies.

In addition, for some strategies, adding augmented data did not only improve the recall as expected but also the precision. That seems to indicate that the network learns more features but chooses them more precisely too.

However, the results with networks trained on the four strategies data combined do not reach the best results. This article shows that the data augmentation strategy and their parameters have to be chosen carefully. Moreover, the results seem to show that different strategy have to be considered for the different instruments.

REFERENCES

[1] Jan Schlüter and Sebastian Böck, "Improved musical onset detection with convolutional neural networks," *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[2] Jouni Paulus and Anssi Klapuri, "Drum sound detection in polyphonic music with hidden markov models," *EURASIP Journal on Audio, Speech, and Music Processing*, 2009.

[3] Richard Vogl, Matthias Dorfer, and Peter Knees, "Recurrent neural networks for drum transcription," *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 730–736, 2016.

[4] Carl Southall, Ryan Stables, and Jason Hockman, "Automatic drum transcription for polyphonic recordings using soft attention mechanisms and convolutional neural networks," *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 606–612, 2017.

[5] Richard Vogl, Matthias Dorfer, Gerhard Widmer, and Paul Knees, "Drum transcription via joint beat and drum modelling using convolutional recurrent neural networks," *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[6] Céline Jacques and Axel Röbel, "Automatic drum transcription with convolutional neural networks," *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2018.

[7] Chih-Wei Wu, Christian Dittmar, Carl Southall, Richard Vogl, Gerhard Widmer, Jason Hockman, Meinard Müller, and Alexander Lerch, "A review of automatic drum transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, 2017.

[8] Chih-Wei Wu and Alexander Lerch, "Automatic drum transcription using the student-teacher learning paradigm with unlabeled music data," *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[9] Brian McFee, Eric Humphrey, and Juan Pablo Bello, "A software framework for musical data augmentation," *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[10] Jan Schlüter and Grill Thomas, "Exploring data augmentation for improved singing voice detection with neural networks," *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[11] Richard Vogl, Gerhard Widmer, and Peter Knee, "Towards multi-instrument drum transcription," *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2018.

[12] Justin Salamon and Juan Pablo Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. abs/1608.04363, 2017.

[13] Niels Bogaards, Axel Röbel, and Xavier Rodet, "Sound analysis and processing with AudioSculpt 2," in *Proc. Int. Computer Music Conference (ICMC)*, 2004.

[14] Miroslav Zivanovic, Axel Röbel, and Xavier Rodet, "A new approach to spectral peak classification," in *Proceedings of the 12th European Signal Processing Conference (EUSIPCO)*, Vienna, Austria, Sept. 2004, pp. 1277–1280.

[15] Axel Röbel, "A new approach to transient processing in the phase vocoder," in *Proc. of the 6th Int. Conf. on Digital Audio Effects (DAFx03)*, 2003, pp. 344–349.

[16] Jean Laroche and Mike Dolson, "Improved phase vocoder time-scale modification of audio," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 323–332, 1999.

[17] Axel Röbel and Xavier Rodet, "Efficient Spectral Envelope Estimation and its application to pitch shifting and envelope preservation," in *International Conference on Digital Audio Effects*, Madrid, Spain, Sept. 2005, pp. 30–35.

[18] Arie Livshin and Xavier Rodet, "The Importance of Cross Database Evaluation in Sound Classification," in *Proc. ISMIR*, Oct. 2003.