Towards Custom Dilated Convolutions on Pitch Spaces

Rony Abecidan **Mathieu Giraud** **Gianluca Micchi**
Université de Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
{rony,mathieu,gianluca}@algomus.fr

ABSTRACT

We benchmark several convolution kernels, in particular custom *dilated convolutions*. We test whether convolutions inspired by known pitch spaces like the Tonnetz may help to achieve better results on the task of key detection.

1. INTRODUCTION

Many studies on music analysis and generation use Convolutional Neural Networks (CNNs) [1], which have obtained amazing results in image analysis. A key advantage of CNNs is their inherent capacity to detect the same feature in different locations as convolution kernels process neighboring pixels in the same way across all the image.

Music, too, can be regarded as bidimensional: Scores are usually symbolically encoded using a horizontal time axis and a vertical pitch axis (cf. pianoroll notation). Most music features are invariant if considered at different bars (time invariance) or transposed to different keys (pitch invariance). Therefore, one can easily see the appeal of using CNNs to analyse music.

However, time and pitch dimensions have their own unique properties which differ from the spatial dimensions of an image. In this short paper we focus on the *pitch* domain, trying to define what metric space would best highlight the properties of music and, therefore, simplify the task of a machine learning model.

A first approach is to consider as neighboring pitches those associated with neighboring keys on the piano (semitone distance). However, this is not convenient in the majority of cases. Take harmonic analysis, for example. It is generally accepted that the sequence of the harmonics is a better approximation to the theoretical proximity of pitches than the semitone distance, meaning that C is closer to G than to C♯. This suggests the use of other representations such as the circle of fifths or, more generally, Tonnetz [2].

The correct pitch metric and space could be learned by deep models in the presence of a sufficiently large amount of data [3], but making them first-order citizens of music learning could improve the way the network learns music. To our knowledge, as of today there is no systematic study on the effect of the choice of different pitch representation for convolutional layers. We propose here the first steps towards such a study, especially focusing on *dilated convolutions*. Our goal is not to improve the state of the art, but only to explore whether new convolution types inspired by expert knowledge in music theory can improve the results on a given musical task.

2. MODEL

As a musical task we choose key detection, which is relatively simple to define (given a music score, identify the local key at regular time intervals) but constitutes an important first step for more complicated tasks in music analysis and generation.

2.1 Architecture

The proof-of-concept model we use is made of 3 convolutional layers, each followed by a batch normalization, then, after the last normalization, a bidirectional GRU and a fully connected layer (Figure 1). Intuitively, the convolutional layers function as feature detectors, the recurrent layers study the progression of chords, and the fully-connected layer adapts the size of the analysis to the size of the desired outputs.

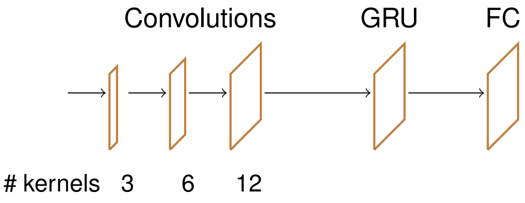

Figure 1. Model for benchmarking convolution layers.

On the time axis, everything is quantised to the eighth note and we perform standard convolutions with kernel size 9 and no dilation. This means that every note is provided a context of 4 eighth notes before and 4 after. The pitch input of the model is encoded with 12 pitch classes. The output, instead, has dimension 24 (12 pitch classes, major or minor).

Attribution: R. Abecidan, M. Giraud, G. Micchi, "Towards Custom Dilated Convolutions on Pitch Spaces", *Extended Abstracts for the Late-Breaking Demo Session of the 21st Int. Society for Music Information Retrieval Conf.,* Montréal, Canada, 2020.

2.2 Pitch Space Analysis

We tested several models for the structure of the pitch space. These models are implemented in practice by different kernels on the pitch space in the convolutional layers.

Baseline. A convolution with kernel size 1 ignores all interactions between notes, effectively considering them all at infinite distance.

Semitone distance. Each note is coupled to the ones immediately next to it. Within a kernel of size 3, C is coupled with B, C, and C♯.

Circle of fifths. A coupling following the circle of fifths gives a dilation on the pitch axis of 7 (for usual spaces with 12 pitches) or 23 (for Base 40 pitches [6]). C is connected to F, C, and G.

Tonnetz. An arbitrary graph of relationships is drawn, notated with a set of integers (Figure 2). For example, the graph $\{0, 3, 4\}$ means that we connect C with C, D♯/E♭, and E, i.e., with its minor and major third.

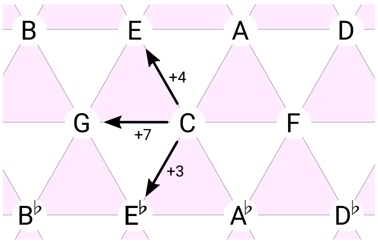

Figure 2. Graphical representation of a Tonnetz. A kernel $\{0, 3, 4\}$ corresponds to a self-connection plus the SW- and NW-bound arrows.

The Tonnetz generalizes the other cases: On spaces with 12 pitches, the baseline can be represented as $\{0\}$, the semitone distance as $\{-1, 0, 1\} = \{0, 1, 11\}$, and the circle of fifths as $\{-7, 0, 7\} = \{0, 5, 7\}$.

2.3 Corpus and implementation

The corpus is made of 201 MusicXML scores gathered in [4] from baroque, classical and romantic repertoire (mainly Bach, Mozart, Beethoven), with Roman Numeral annotations for each beat. The corpus was sliced into 5808 frames, each of length 200 eighth notes. The dataset was randomly cut into a training (40%), a validation (30%), and a test set (30%). In a second experiment, each subpart in the training set was transposed twelve times to balance the keys.

Music parsing was done through music21 [5] and the network was implemented with PyTorch v1.2.0 [1] . Custom dilated convolution were implemented through masking of the kernels.

without transposition of the training set

Kernel	Accuracy	Kernel	Accuracy
$\{0, 6\}^{12}$	45.3%	$\{0, 4, 8\}^{8}$	42.3%
$\{0, 8\}^{12}$	46.4%	$\{0, 3, 10\}^{8}$	43.9%
...		...	
$\{0, 10\}^{12}$	50.7%	$\{0, 8, 11\}^{8}$	53.1%
$\{0, 7\}^{12}$	51.0%	$\{0, 5, 7\}^{8}$	54.0%

with transposition of the training set

Kernel	Accuracy	Kernel	Accuracy
$\{0, 4\}^{12}$	68.1%	$\{0, 2, 7\}^{8}$	66.4%
$\{0, 8\}^{12}$	68.3%	$\{0, 2, 4\}^{8}$	67.0%
...		...	
$\{0, 10\}^{12}$	70.4%	$\{0, 3, 10\}^{8}$	71.3%
$\{0, 3\}^{12}$	71.0%	$\{0, 5, 8\}^{8}$	71.8%

Table 1. Accuracy in ascending order on the test dataset for the task of key detection on 2-note kernels (left) and 3-note kernels (right). The baseline is $\{0\}^{24} : 51.1\%$ without transposition, 68.3% with. In the third convolutional layer, all kernels have 24×9 weights.

3. RESULTS

Since they can learn transposition-independant features in a compact way, convolutions are a good candidate as a constituent part of models for musical tasks. The first results show that the choice of convolution kernel influences the training and that promising results are achieved with dilated convolutions that follow usual music theory pitch spaces such as Tonnetz.

For example, Table 1 shows that, in the two-note kernels, kernels $\{0, 10\}$ (C-B♭, minor seventh), $\{0, 3\}$ (C-E♭, minor third), and $\{0, 7\}$ (C-G, fifth) provide the best results. Of course, fifths are essential elements of a key, but they are also ubiquitous. Therefore, we expected intervals such as minor sevenths or even the tritone (C-F♯) to give more information on the local key.

Another consideration to make is that each convolutional layer after the first increases the complexity of the interacting music entities. For example, using 2-note kernels the second layer connects intervals, while with 3-note kernels it connects triads. Having such higher-order features is necessary for the study of functional harmony – but the task of key detection is probably too simple to observe any such effect.

These results should be further studied to understand some symmetry-breaking we observe – kernels $\{0, 2\}$ and $\{0, 10\}$ should give similar results, provided that bounding conditions are properly implemented, but they don't. Perspectives also include testing with more data, including more key-balanced data, tests on Base 40 pitches [6], benchmarking more kernel combinations, more options on the rhythm domain, as well as well as on other toy tasks.

Acknowledgements. We thank Mark Gotham for inspiring discussions.

[1] www.pytorch.org

4. REFERENCES

[1] G. H. Jean-Pierre Briot and F.-D. Pachet, *Deep Learning Techniques for Music Generation*, 2019.

[2] D. Tymoczko, *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice*. Oxford University Press, 2011.

[3] S. Lattner, M. Grachten, and G. Widmer, “Learning Transposition-Invariant Interval Features from Symbolic Music and Audio,” in *International Society for Music Information Retrieval Conference 2018*, 2018, p. 8, arXiv: 1806.08236. [Online]. Available: http://arxiv.org/abs/1806.08236

[4] G. Micchi, M. Gotham, and M. Giraud, “Not all roads lead to Rome: Pitch representation and model architecture for automatic harmonic analysis,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 42–54, 2020.

[5] M. S. Cuthbert and C. Ariza, “music21: A toolkit for computer-aided musicology and symbolic music data,” in *International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010, pp. 637–642.

[6] W. B. Hewlett, *A Base-40 Number-line Representation of Musical Pitch Notation*, 1986, pp. 1–14.