Shift-Invariant Dictionary Learning using Temporal CONV-WTA Autoencoders for Discovering Music Relations

Anonymous ACL submission

Abstract

The temporal structure of music is full of shift-invariant patterns (e.g. motifs, ostinatos, loops, etc.). We propose using a Temporal Convolutional Winner-Take-All (CONV-WTA) autoencoder to find a shift-invariant dictionary to represent symbolic, multivariate, musical signals. The model learns to represent fixed length drum beats and variable length piano music. We discuss applications of this sparse representation such as de-noising musical ideas, unsupervised learning of composer styles, and music generation. To assist in related work we include interactive code along with the trained models

1 Introduction

The dictionary learning framework aims at finding a sparse representation of the input data (sparse coding) in the form of a linear combination of basic elements called atoms. In doing so, sparse coding enables faster inference and easier interpretability thanks to its lightweight stored memory, encoding of prior knowledge in the sparsity patterns, and discerning patterns in an informed and principled manner.

Sparse dictionary learning has led to state-of-art results in various tasks including image and video processing, texture synthesis (Peyré, 2009), and unsupervised clustering (Ramírez et al., 2010). In evaluations with the Bag-of-Words model (Koniusz et al., 2017), sparse coding was found empirically to outperform other coding approaches on category recognition tasks.

When applied to music, the ability to distil complex data structures down to sets of dictionaries—salient features of a specific performer or music, has a multitude of applications. Music transcription and classification tasks have seen a strong usage of sparse dictionary learning in the past (Grosse et al., 2007) (Costantini et al., 2013), (Blumensath and Davies, 2006), (Srinivas M et al., 2014), (Srinivas et al., 2014), (Cogliati et al., 2016). Nonetheless, we have yet to see a study that harnesses the advantages of sparse representation for the purpose of music creation. Instead, the popular methods for discovering music relations and achieving music generation have been a transformer with some sort of attention mechanism or other recurrent architectures. For instance, (Jiang Junyan et al., 2020) uses an attention module that is tailored to the discovery of sequence level relations in music, while studies like (Roberts et al., 2018) uses the recurrent variational autoencoder and a hierarchical decoder in order to model long-term musical structures. In our study, we explore applications of sparse representation such as denoising musical ideas, unsupervised learning of composer styles, and music generation.

2 Preliminaries

2.1 Dictionary learning

Given the data: $X = [x_1, \ldots, x_K], x_i \in \mathbb{R}^d$. We want a dictionary $\mathbf{D} \in \mathbb{R}^{d \times n} : D = [d_1, \ldots, d_n]$, and a representation $R = [r_1, \ldots, r_K], r_i \in \mathbb{R}^n$ such that the reconstruction $\|X - \mathbf{D}R\|_F^2$ is minimized and r_i are sparsed. The optimization problem can be formulated as:

$$\underset{\mathbf{D} \in \mathcal{C}, r_i \in \mathbb{R}^n, \lambda > 0}{\operatorname{argmin}} \sum_{i=1}^{K} \|x_i - \mathbf{D}r_i\|_2^2 + \lambda \|r_i\|_0$$
$$\mathcal{C} \equiv \left\{ \mathbf{D} \in \mathbb{R}^{d \times n} : \|d_i\|_2 \leq 1 \forall i = 1, \ldots, n \right\}$$

There are various methods to solve this problem, however this formulation does not look for shift-invariant features. The dictionary components are the same size as the original signal we are seeking to reconstruct.

2.2 Shift-invariant Dictionary Learning (SIDL)

Shift-invariant dictionary learning (SIDL) refers to the problem of discovering a latent basis that captures local patterns at different locations of input signal, and a sparse coding for each sequence as a linear combination of these elements (Zheng et al., 2016) In previous works, various shift-invariant dictionary learning (SIDL) methods have been employed to discover local patterns that are embedded across a longer time series in sequential data such as audio signals (Grosse et al., 2007)

This has a similar formulation as DL except that in order to reconstruct the signal, we need to stride along the input signal:
$\mathbf{D}r_i \longrightarrow \sum_{k=1}^{K} (\boldsymbol{r}_i)_k T\left(\mathbf{d}_k, t_{ik}\right)$
where

$$T(\mathbf{d}, t) = \begin{cases} \mathbf{d}_{i-t} & \text{if } 1 \leq i-t \leq q \\ 0 & \text{otherwise} \end{cases}$$

here t_{ik} corresponds to the first location where d_k matches our signal. Therefore, $t_{ik} = 0$ indicating that $\mathbf{d}_k$ is aligned to the beginning of $\mathbf{x}_i$ and that $t_{ik} = p - q$ indicating the largest shift $\mathbf{d}_k$ can be aligned to $\mathbf{x}_i$ without running beyond (Zheng et al., 2016).

2.3 Temporal Convolutional Networks (TCN)

Recent results suggest that TCN outperform baseline recurrent architectures across a broad range of sequence modeling tasks. The attributes of TCN are: the convolutions in the architecture have no information leakage from future to past; and the architecture can take a sequence of any length and map it to an output sequence of the same length, similar to an RNN. In summary: TCN = 1D FCN + causal convolutions (Shaojie Bai et al., 2018).

This architecture is designed according to recent convolutional architectures for sequential data (van den Oord et al., 2016), (Kalchbrenner et al., 2017) ; (Dauphin et al., 2016); (Zheng et al., 2016). TCN have several advantages: they have no skip connections across layers, conditioning, context stacking, or gated activations.

2.4 SIDL by CONV-WTA Autoencoders

To learn the shift-invariant dictionaries, we use a Temporal CONV-WTA Autoenconder (Makhzani and Frey, 2014). This is a standard convolutional autoencoder except after training the encoder, the single largest hidden activity of each feature map is kept and the rest (as well as their derivatives) are set to zero. Next, the decoder reconstructs the output from the sparse feature maps. This results in a sparse representation where the sparsity level is the number of non-zero feature maps. If a shallow decoder (1 layer) is used, the kernel weights of the decoder are the atoms of the dictionary used to reconstruct the signal.

In theory, all that is required to find a dictionary is a 1D-Conv Encoder-Decoder layer. However, if the input signal is dense, learning can improve by adding a TCN Layer to extract features. This TCN can have variable depth depending on the task.

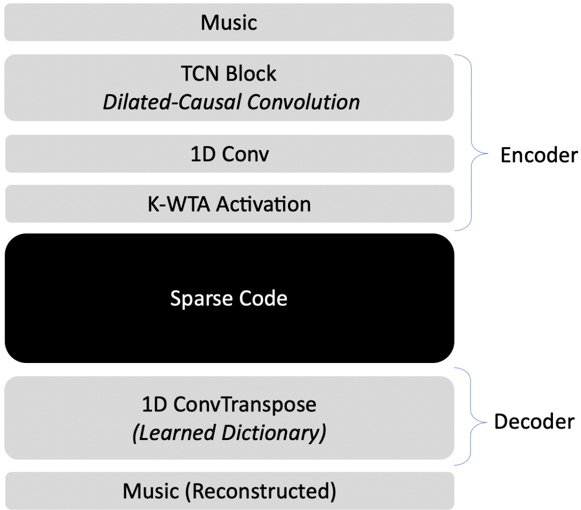

Figure 1: Diagram depicting the Temporal CONV-WTA Autoencoder. The TCN Block can have arbitrary depth, and is not required to construct a dictionary but useful to extract features

3 Experiments

We show a few applications of our CONV-WTA model, de-noising musical ideas, unsupervised learning of composer styles, and music generation. We do this for two distinct datasets with different MIDI encodings.

3.1 Datasets

Two distinct datasets are used: MAESTRO (Hawthorne et al., 2019), and Groove (Gillick et al., 2019) with two distinct MIDI representations (see Table 1 for more details).

3.2 Model Implementation

Both models were implemented in Pytorch, with MSE loss on reconstruction, and AdamW optimizer. The model implementation differs slightly for the two datasets.

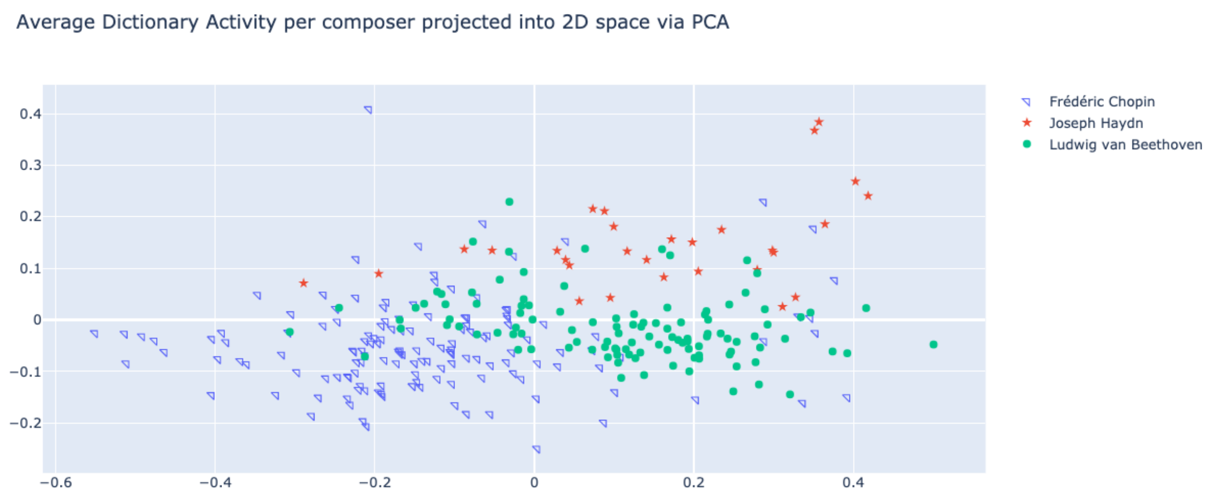

Figure 2: After training the model on the MAESTRO dataset we can encode music of arbitrary length, average atom activity (rows of sparse code), and perform PCA for visualization

Dataset	Size	Instrument	MIDI Representation
MAESTRO	1020 (Hrs)	Piano	One-hot encoding over 388 different MIDI events. Music has an arbitrary length (Oore et al., 2018)
Groove	3.6 (Hrs)	Drum	T timesteps (one per 16th note) and 27 MIDI events. We use fixed length 64 time step sections (Gillick et al., 2019)

Table 1: Details for the datasets used to train the Fully Convolutional Temporal Autoencoder (FCTA)

MAESTRO Model: The architecture is illustrated in Figure 1. The TCN layer uses a [1,8,16,24] feature map, and a dictionary size of 1000 along with a decaying k-WTA[1]. The training begins with $k = 100$ and decay to $k = 75$ over 60 epochs. The convolutions are non-overlapping strides, meaning every kernel-length time step is only made up of one column in our sparse code. This helps to reduce memory requirements and find repeating sections over a fixed kernel length. A batch size of 1 is used, this allows us to train on variable length music since our autoencoder is fully convolutional. However our sparse representation can also be variable length.

Groove Model: The architecture is illustrated in Figure 1, however since the Groove drum representation is not very dense, we do not use a TCN layer. The dictionary size is 100 along with a decaying k-WTA with $k = 4$ and decay to $k = 1$ over 1000 epochs. The convolutions are also non-overlapping strides. The full dataset is used with no batching. The dataset is processed to match 120 bpm, 4/4 time signature and only train on 1 bar of music. For this dataset we have a good understanding of repeating time intervals and structure, unlike the MAESTRO dataset.

[1]In the original paper k-WTA is applied after training the encoder. We use the k-WTA activation during training for lower GPU memory requirements, and the decaying k showed to achieve a higher accuracy

3.3 Music Reconstruction

De-noising Musical Ideas: Dictionary learning has successfully been shown to de-noise images (Beckouche et al., 2013). By simply reconstructing our musical signal with a limited dictionary, our signal should dismiss "noisy" features that are not shift-invariant such as inconstant musical ideas; moreso recognizing where music introduces shift-variant patterns can be useful for music analysis.

Keep N Active Atoms: The sparse coding's ability to recognize the most used atoms in the dictionary enables a low dimensional feature extraction for machine learning tasks. We believe that this could be applied to music reduction wherein the complexity of the arrangement is reduced to a simpler transcription and parts in an unsupervised way.

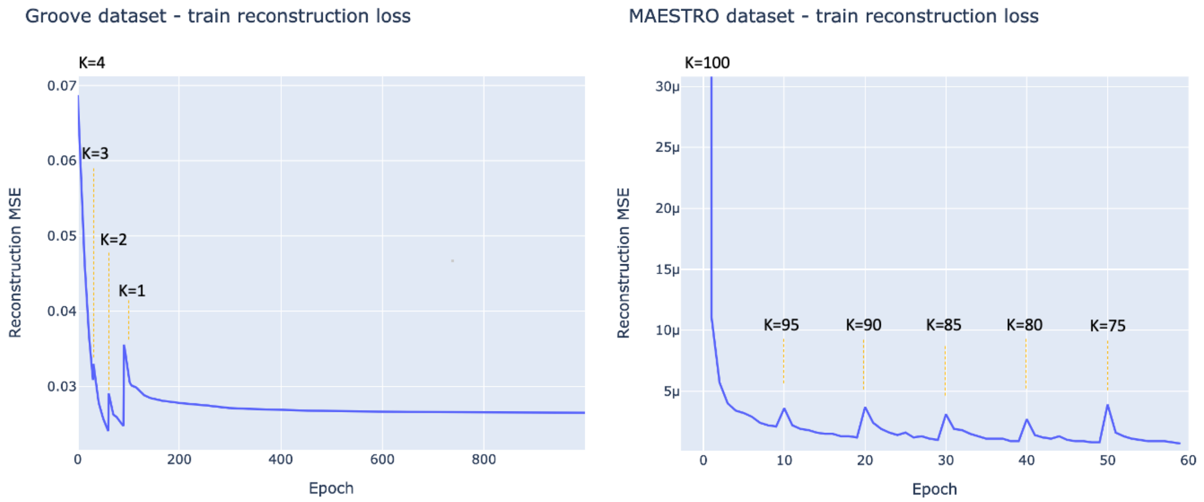

Figure 3: Reconstruction loss of the Temporal Convolutional Winner-Take-All (CONV-WTA) autoencode for MAESTRO and Groove training data. The value of k is decaying as we progress training[1]

3.4 Unsupervised Learning of Composer Styles

We apply our model to a dataset that includes multiple composers to see if different composers utilize different shift-invariant patterns. After training, we encode music of arbitrary length, average atom activity (average rows of sparse code), and perform PCA for visualization. The result, as shown in Figure 2, illustrates separation and clustering of three distinct composer styles, and agrees with our assumption, for instance, that Haydn is stylistically closer to Beethoven than he is to Chopin. This sparse representation can also be used to measure similarity between composers by comparing distance of centroids between clusters, and we plan to utilize such measurement to aid music analysis and creation.

3.5 Generating New Music

Interpolating Sparse Code: To generate new music, we encode existing music and interpolate between sparse codes[2]. For drum generation, all of the sections are fixed length whereas for the piano generation a variable length sparse code is used. To interpolate variable length code we can fix the input size, or interpolate specific time intervals of the sparse code.

Swapping Atoms: Another method to generate new music is to measure the most active atoms in the dictionary from a musical section (row in the sparse code), and replace them with the most active atoms from a different musical section. This can be used as a tool for artists to freely re-mix and merge features and can offer a unique way to re-compose their own version of a given music.

[2] The interpolation result is better if the sections have similar musical ideas. We can use Cos similarity between the encodings to find similar musical sections to interpolate.

4 Conclusion

We have shown that a Temporal CONV-WTA Autoencoder can learn a sparse representation of arbitrary length symbolic musical signal. This shift-invariant, sparse representation can be used to analyze features, de-noise, extract style, and to generate musical content in a structured or unstructured way. The reconstruction and generation for the drum (Groove) dataset was significantly better than the piano (MAESTRO) dataset. This is in part because the drum dataset was preprocessed to match with the kernel size, all drum sections were the same length, and had lower dimensionality in comparison.

In the future, we hope to use a larger and more diverse dataset, improve reconstruction performance, and apply similar preprocessing to the piano data as done for the drum data. We also plan to further develop applications of this technology and build tools for artists.

References

S. Beckouche, J. L. Starck, and J. Fadili. 2013. Astronomical image denoising using dictionary learning. *Astronomy and Astrophysics*, 556:A132.

Thomas Blumensath and Mike Davies. 2006. Sparse and shift-invariant representations of music. In *IEEE Transactions on Audio, Speech and Language Processing*, volume 14.

Andrea Cogliati, Zhiyao Duan, and Brendt Wohlberg. 2016. Context-Dependent Piano Music Transcription with Convolutional Sparse Coding. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 24(12).

Giovanni Costantini, Massimiliano Todisco, and Renzo Perfetti. 2013. NMF based dictionary learning for automatic transcription of polyphonic piano music. *WSEAS Transactions on Signal Processing*, 9(3).

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2016. Language modeling with gated convolutional networks. *CoRR*, abs/1612.08083.

Jon Gillick, Adam Roberts, Jesse Engel, Douglas Eck, and David Bamman. 2019. Learning to groove with inverse sequence transformations. In *International Conference on Machine Learning (ICML)*.

Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Y. Ng. 2007. Shift-invariant sparse coding for audio classification. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence, UAI 2007*.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. 2019. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *International Conference on Learning Representations*.

Jiang Junyan, Gus Xia, and Taylor Berg-Kirkpatrick. 2020. Discovering Music Relations with Sequential Attention. In *Proceedings of the 1st workshop on nlp for music and audio (nlp4musa)*, pages 1–5.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2017. Neural machine translation in linear time.

Piotr Koniusz, Fei Yan, Philippe-Henri Gosselin, and Krystian Mikolajczyk. 2017. Higher-order occurrence pooling for bags-of-words: Visual concept detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):313–326.

Alireza Makhzani and Brendan J. Frey. 2014. A winner-take-all method for training sparse convolutional autoencoders. *CoRR*, abs/1409.2752.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio.

Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. 2018. This time with feeling: Learning expressive musical performance. *CoRR*, abs/1808.03715.

Gabriel Peyré. 2009. Sparse modeling of textures. 34(1):17–31.

I. Ramírez, P. Sprechmann, and G. Sapiro. 2010. Classification and clustering via dictionary learning with structured incoherence and shared features. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3501–3508.

Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2018. A hierarchical latent vector model for learning long-term structure in music. In *35th International Conference on Machine Learning, ICML 2018*, volume 10.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

M. Srinivas, Debaditya Roy, and C. Krishna Mohan. 2014. Learning sparse dictionaries for music and speech classification. In *International Conference on Digital Signal Processing, DSP*, volume 2014-January.

Srinivas M, Roy D, and Mohan CK. 2014. Music genre classification using on-line dictionary learning. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1937–1941.

Guoqing Zheng, Yiming Yang, and Jaime Carbonell. 2016. Efficient shift-invariant dictionary learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 13-17-August-2016.