Shift-Invariant Dictionary Learning using TCN-WTA Autoencoders for Discovering Musical Relations

Anonymous ACL submission

Abstract

Music temporal structure is full of shift-invariant patterns (e.g. motifs, ostinatos, loops, samples, etc.). In machine learning, the standard methods to encode a generic sequence is usually achieved by recurrent architectures such as LSTMs or Transformers. However, RNN architectures do not take advantage of this repetitive structure . We propose using a fully convolutional Temporal Convolutional Autoencoder to find a shift-invariant dictionary that can recreate symbolic multivariate musical signals. To find a dictionary we utilize a k-Winner Takes All (k-WTA) activation function to promote a sparse representation. In addition to gaining insight of shift-invariant patterns, some results indicate that CNN architectures can outperform recurrent networks on specific tasks and provide advantages while also demonstrating longer effective memory. We show few applications of this sparse representation such as de-noising musical ideas, unsupervised stylistic segmentation, and music generation. To assist related work, we have made interactive code available along with the trained models

1 Introduction

Deep learning practitioners commonly regard recurrent architectures as the default starting point for sequence modeling tasks. The sequence modeling chapter in the canonical textbook on deep learning is titled “Sequence Modeling: Recurrent and Recursive Nets” (Goodfellow et al., 2016), capturing the common association of sequence modeling and recurrent architectures. A well-regarded recent online course on “Sequence Models” focuses exclusively on recurrent architectures (Ng, 2018).

In addition, recent research indicates that certain convolutional architectures can reach state-of-the-art accuracy in audio synthesis, word-level language modeling, and machine translation (van den Oord et al., 2016; Kalchbrenner et al., 2016; Dauphin et al., 2017; Gehring et al., 2017a;b).This raises the question of whether these successes of convolutional sequence modeling should be explored.

One possible way to use convolutional architectures for sequence modeling is to employ a Dictionary Learning (DL) framework. DL requires using a few basis elements, atoms, learned from data itself. This appraoch has led to state-of-art results in various image and video processing tasks. Sparse dictionary learning has been successfully applied to various image, video and audio processing tasks as well as to texture synthesis[16] and unsupervised clustering (Ramírez et al., 2010) In evaluations with the Bag-of-Words model,(Vogl and Knees, 2017) (Koniusz et al., 2017) sparse coding was found empirically to outperform other coding approaches on the object category recognition tasks. Furthermore, a sparse representation can be used to encode prior knowledge in the sparsity patterns. Second, they are lightweight— requiring less memory to store and allowing faster inference and easier interpretability. Sparsity provides a way to discern patterns in an informed and principled manner, resulting in smaller model size.

The ability to distill complex data structures such as music down to sets of dictionaries—salient features of a specific performer or music, has a multitude of applications in music, we will show a few

2 Related Work

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 original signal we are seeking to recustruct.

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)

This has a similar formulation as eq.1 except to reconstruct we have to stride along our signal. We can rewrite

$$\mathbf{D}r_i \longrightarrow \sum_{k=1}^{K} \boldsymbol{r}_{ik} T\left(\mathbf{d}_k, t_{ik}\right)$$

where

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

here t_{ik} corresponds 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

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. While (Grosse et al., 2007) employs shift- invariant sparse coding with a convolutional optimization and gradient descent method for an audio classification task, (Zheng et al., 2016) demonstrates an efficient algorithm with the ability to combine shift-invariant patterns in a sparse coding of the original data for audio reconstruction and classification tasks. Such unsupervised learning methods have shown to be powerful in discovering shift-invariant patterns and a handful of studies have implemented SIDL for the purpose of music. Although 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), 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 the recurrent architectures. (Jiang Junyan et al., 2020) for instance, 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.

2.3 Temporal Convolutional Networks (TCN)

The distinguishing characteristics of TCNs are: 1) the convolutions in the architecture are causal, meaning that there is no information “leakage” from future to past; 2) the architecture can take a sequence of any length and map it to an output sequence of the same length, just as with an RNN. Beyond this, we emphasize how to build very long effective history sizes (i.e., the ability for the networks to look very far into the past to make a prediction) using a combination of very deep networks (augmented with residual layers) and dilated convolutions. Our architecture is informed by recent convolutional architectures for sequential data (van den Oord et al., 2016; Kalchbrenner et al., 2016; Dauphin et al., 2017; Gehring et al., 2017a;b), but is distinct from all of them and was designed from first principles to combine simplicity, autoregressive prediction, and very long memory. For example, the TCN is much simpler than WaveNet (van den Oord et al., 2016) (no skip connections across layers, conditioning, context stacking, or gated activations). Compared to the language modeling architecture of Dauphin et al. (2017), TCNs do not use gating mechanisms and have much longer memory

2.4 SIDL by TCN K-WTA Autoencoders

In our study, we use a fully convolutional temporal autoencoder to find shift-invariant dictionaries while ensuring sparsity via the K-WTA activation function [Winner-Take-All Autoencoders]. The use of K-WTA in conjunction with dictionary learning was inspired by [Towards Contrastive Learning for Time-Series] where the K-WTA's ability to achieve sparse representations is explored in the context of constructive learning for time-series data. The use of Temporal Convolutional Nets was encouraged by various advantages that convolutional architectures bring for sequence modeling over recurrent networks as illustrated in (Shaojie Bai et al., 2018).

3 Experiments

We show a few applications of our TCN k-WTA model

- Cleaning musical sections
- Unsupervised feature extraction
- Generating new music

3.1 Datasets

We use two distinct datasets: MAESTRO (Hawthorne et al., 2019), and grove Groove (Gillick et al., 2019) . See table 2 for more details on the datasets used. We also use distinct MIDI representations for each dataset.

3.2 Model Implementation

Our model implementation differs slightly for the different datasets used.

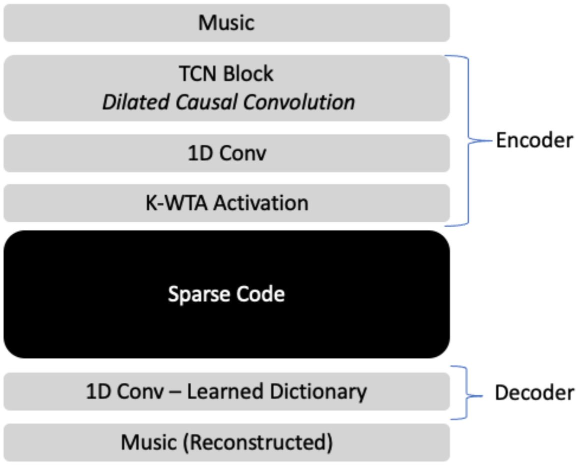

Figure 1: After training the model we can use it to encode datapoints of arbitrary length unsupervised stylistic segmentation. We use PCA on the average sparse code for each piece. We project into 2 dimensional sparse to visualize

MAESTRO

Our TCN-KWTA Autoencoder is designed with [1, 8, 16, 32, 1000, 1] layers. The sparse representation is the layer before the last. Our WTA activation function is in the layer before the last. We also use a decaying WTA activation fucntions

GROOVE

Our TCN-KWTA Autoencoder is designed with [1, 8, 16, 32, 1000, 1] layers. The sparse representation is the layer before the last. Our WTA activation function is in the layer before the last. We also use a decaying WTA activation fucntions

3.3 Music Reconstructions

Center the title, author's name(s) and affiliation(s) across both columns. Do not use footnotes for affiliations. Do not include the paper ID number assigned during the submission process. Use the two-column format only when you begin the abstract.

The title, author names and addresses should be completely identical to those entered to the electronical paper submission website in order to maintain the consistency of author information among all publications of the conference. If they are different, the publication chairs may resolve the difference without consulting with you; so it is in your own interest to double-check that the information is consistent.

3.4 Keep Top N Dictionary words

Another application of having a sprase code, is the ability to recognize the most used words in the dictionary. Given a sparse code we can limit a piece to only be made up of the top N words. Some example applications of keeping to top N words are for example, low dimensionality feature extraction for machine learning taks; music reduction–reduction wherein the complexity of the arrangement is reduced to a simpler transcription and parts. And Music Segmentation (Discretization) wherein various musical ideas used in a piece of music are isolated from the piece itself. Such segmented ideas could be used in analysis or creatively repurposed to generate new music.

3.5 Unsupervised Stylistic Segmentation

If we train with a dataset that includes multiple composers we should expect to find that different composers utilize different shift invariant patters. To visualize kernel usage we average out the rows of our sparse code. This should provide us with the average dictionary word value. If we do PCA on the average dictionary vector and reduce dimensionality to 2D. We obtain the plot in Figure 2

It is also possible to use this average kernel usage for each composer and make comparisons for

Dataset	Size	Instrument	MIDI Representation
MAESTRO	1020 (Hrs)	Piano	One-hot encoding over 388 different MIDI events. Every datapoint here has an arbitrary length
Groove	3.6 (Hrs)	Drum	T timesteps (one per 16th note) and 27 MIDI events. We use fixed length 64 time step sections

Table 1: Datasets used to experiment with fully convolutional temporal autoencoder model. All datasets used are MIDI format

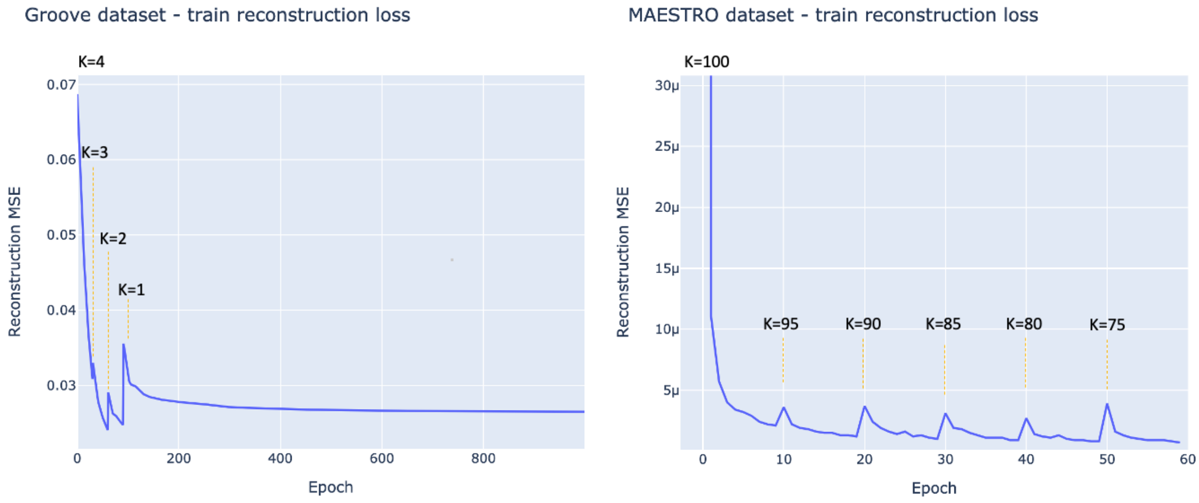

Figure 2: After training the model we can use it to encode datapoints of arbitrary length unsupervised stylistic segmentation. We use PCA on the average sparse code for each piece. We project into 2 dimensional sparse to visualize

composers, such as most similar or disimilar between styles or comparisons.

3.6 Genrating Structured Drums

If we have a specific musical structure we would like to follow. For example, 4/4 120bpm. We can train inject prior information into our 1D convolutions. For example, we can make each kernel the length of 1/4 beat. This allows for better dictionary learning since repetition is expected to happen at this time steps. We can stitch together musical sections, to create bars of music. and desing a specfic structure

3.7 Genrating Piano Music

There are multiple benefits to this sequence learning methodology. The first is such as simplicity and felxibilty, the TCN autoencocoder is a simple to implemnt arquitecture and requires any abitrary size combinations of multivariate musical signals.

4 Discussion

There are multiple benefits to this sequence learning methodology. The first is such as simplicity and felxibilty, the TCN autoencocoder is a simple to implemnt arquitecture and requires any abitrary size combinations of multivariate musical signals.Also the size of the model for both Magenta and Groove are 877 KB and 750 KB respectively. In comparison, googles Performance RNN–LSTM-based recurrent neural network–is 25MB, and other transformer based models can be GBs in size.

In addition, our method allows for incorporating known structural information into a model prior to trainng. and finally we have the most imporant benifit is the sparse representatoin and learned dictionaries, as we have shown this can be used for mutiple applications and analysis.

The performance on the recustruction and generation on for the drum Grove dataset was significantly better than the piano (MAESTRO dataset). This is in part because the dataset was pre-

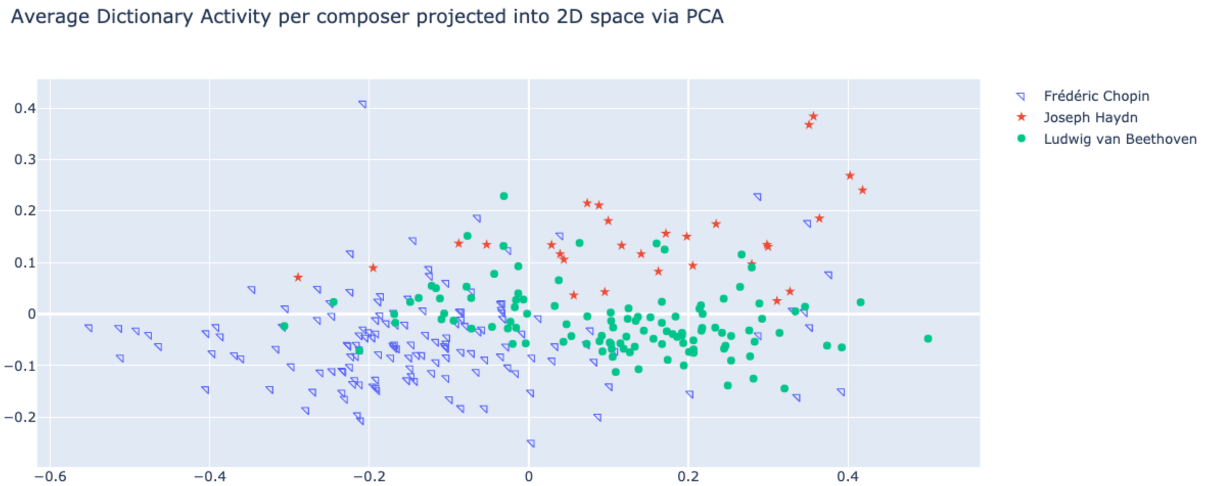

Figure 3: After training the model we can use it to encode datapoints of arbitrary length unsupervised stylistic segmentation. We use PCA on the average sparse code for each piece. We project into 2 dimensional sparse to visualize

proccessed to match with kernel size, and the drum sections were the same length and have lower dimensionality.

5 Conclusion

We have shown that TCN-kWTA autoencoder can learn a sparse representation of abitrary length musical signal. This shift invariant, sparse representation can be used to analyze, preprocess, or generate musical content in a structured, or unsctuctured way

It is also advised to supplement non-English characters and terms with appropriate transliterations and/or translations since not all readers understand all such characters and terms. Inline transliteration or translation can be represented in the order of: original-form transliteration “translation”.

Acknowledgments

The acknowledgments should go immediately before the references. Do not number the acknowledgments section. Do not include this section when submitting your paper for review.

Preparing References:

Include your own bib file like this:
`\bibliographystyle{nlp4MusA_natbib}`
`\bibliography{nlp4MusA}`
where `nlp4MusA` corresponds to a nlp4MusA.bib file.

References

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).

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.

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.

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.

Richard Vogl and Peter Knees. 2017. An Intelligent Drum Machine for Electronic Dance Music Production and Performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*.

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.

A Appendices

Appendices are material that can be read, and include lemmas, formulas, proofs, and tables that are not critical to the reading and understanding of the paper. Appendices should be **uploaded as supplementary material** when submitting the paper for review. Upon acceptance, the appendices come after the references, as shown here. Use `\appendix` before any appendix section to switch the section numbering over to letters.

B Supplemental Material

Submissions may include non-readable supplementary material used in the work and described in the paper. Any accompanying software and/or data should include licenses and documentation of research review as appropriate. Supplementary material may report preprocessing decisions, model parameters, and other details necessary for the replication of the experiments reported in the paper. Seemingly small preprocessing decisions can sometimes make a large difference in performance, so it is crucial to record such decisions to precisely characterize state-of-the-art methods.