Exploring Musical Structure using Tonnetz Lattice Geometry and LSTMs *

Manuchehr Aminian[1 [0000−0001−5970−9709]], Eric Kehoe[1 [0000−0001−5595−9452]], Xiaofeng Ma[1 [0000−0002−3967−5975]], Amy Peterson[1 [0000−0001−8273−1365]], and Michael Kirby[1]

Colorado State University, Fort Collins, CO 80523 USA
{manuchehr.aminian, eric.kehoe, xiaofeng.ma, amy.peterson, michael.kirby}@colostate.edu

Abstract. We study the use of Long Short-Term Memory neural networks to the modeling and prediction of music. Approaches to applying machine learning in modeling and prediction of music often apply little, if any, music theory as part of their algorithms. In contrast, we propose an approach which employs minimal music theory to embed the relationships between notes and chord structure explicitly. We extend the Tonnetz lattice, originally developed by Euler to introduce a metric between notes, in order to induce a metric between chords. Multidimensional scaling is employed to embed chords in twenty dimensions while best preserving this music-theoretic metric. We then demonstrate the utility of this embedding in the prediction of the next chord in a musical piece, having observed a short sequence of previous chords. Applying a standard training, test, and validation methodology to a dataset of Bach chorales, we achieve an accuracy rate of 50.4% on validation data, compared to an expected rate of 0.2% when guessing the chord randomly. This suggests that using Euler's Tonnetz for embedding provides a framework in which machine learning tools can excel in classification and prediction tasks with musical data.

Keywords: Long Short-Term Memory · LSTM · Music Theory · Multidimensional Scaling · Euler's Tonnetz

1 Introduction

Long Short-Term Memory (LSTM) neural networks are well known and well utilized neural networks for prediction and classification tasks [1,4,3]. In particular they have been used to create polyphonic music models— models that can predict or create music by learning from existing musical pieces. Usually one classifies or predicts on collections of notes called chords. Such a predictive

* This research is partially supported by the National Science Foundation under Grant No. DMS-1322508. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_31

model can aid in composing musical scores which capture the harmonic stylings of a particular musical artist or genre. Learning the large scale harmonic structure, such as harmonic cadences, from the level of individual to an entire musical tradition is of great interest to the musical community. In particular, we will use music theory to construct an embedding of chords as a foundation for an LSTM neural network to predict in a musical piece the chord following its previous measure.

Prior work has been done in applying machine learning to the prediction, classification, and composition of music in various contexts, and in particular in the utility of LSTMs to learn long sequences of patterns in ordered data. There is a focus on music composition, generation, and improvisation using LSTMs [5,4,6,8] which involve future prediction on different time scales and styles using many approaches. Some work has been done on an even broader scale; trying to incorporate large scale musical structure [7] and classification of musical genre [16].

Similarly, there have been approaches to incorporating music theory; see [13] as an example which relates information about major and their relative minor keys, among other things. Analysis has been done on networks which explicitly encode information beyond harmonic structure, including pitch and meter [14], which suffers (as they suggest in their paper title) from an extremely large state and hyperparameter spaces.

Towards the analysis of LSTM network structure in music prediction, the authors in [1] use the JSB chorales dataset to predict the next set of notes in a given chorale. They use this to test various LSTM architectures to quantify the use of gates and peephole connections in an LSTM structure.

Finally, approaches to embedding chords via a "chord2vec" (in analogy with the famous word2vec in natural language processing) has been explored in [3] which embeds the data via the word2vec methodology then studies LSTM performance, among a variety of null models.

Taking a different approach, we choose to train our LSTMs on chord types, so collections of notes modulo an octave. Chords take into account the information of pitch class, e.g. there is a difference between the notes $C4$ and $C5$. In our setting the chord $(C4, E5, G4)$ is identified with $(C5, E3, G5)$, both of which are called a C-major triad. We then can measure harmonic distance between chord types by measuring how far apart they are from one another on the Tonnetz lattice— a lattice structure originally developed by Euler to represent musical harmony, see Figure 1 for Eulers original picture. Multidimensional scaling can then be used to embed chords as vectors in a relatively low dimensional space using the Tonnetz distance. This embedding uses musical information to create the feature space of the chords rather than trying to train a neural network directly on one-hot encoded chords or the use of other embedding methods such as `chord2vec` [3].

A description of the training and the setup of the model are in the Section 2 below along with an explanation of the dataset on which we train our model. We

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_31

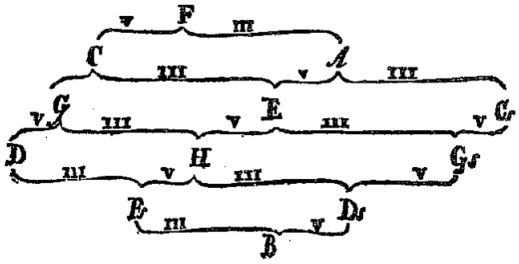

Fig. 1. The original Tonnetz depicted first in Euler's 1739 *Tentamen novae theoriae musicae ex certissismis harmoniae principiis dilucide expositae.*

also reference our visualization illustrating the process of a song being classified by chords [17].

In the following sections, we investigate the size of the hidden layers in their ability to predict chords. The details of this model are also discussed in Section 2. We look at the results of two different types of prediction and various hidden layer sizes in Section 3.

2 Evaluation

For the setup of our experiments, that will be discussed in this section, we focus on two important aspects: the embedding method and the structure of the LSTMs. First, we describe the embedding method of chords which takes into account the structure of chords utilizing Euler's Tonnetz lattice and the dataset that will be used. Second, we give the details of the LSTM structure and how it is applied to the embedded chords.

2.1 Data Set and Embedding

We train, test, and validate our model on a set of 382 Bach chorales, `JSB Chorales` [9]. This collection of Bach chorales is pre-partitioned into sub-collections of training, testing, and validation songs. The format of the songs is given in MIDI or Musical Instrument Digital Interface. Unlike other music formats such as .wav or .mp3, MIDI allows one to access individual notes and beats within a song. Musical software can then be used to play the music similar to how a player piano plays a piano roll. The advantage of the data being represented in this format is that one can work with music on the level of individual notes and beats to create discretizations of songs which preserve their harmonic or melodic structure.

We preprocess the MIDI files using the python package `music21` [10]. In particular, we process MIDI files by partitioning their time axis into 8th note bins and snapping all the notes in a particular bin to that bin's left time point. This process is called quantization. Once the music has been quantized all the

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_31

musical notes at each eighth note are grouped into a collection of notes called chords. Of course in some time bins there may be no notes present, this is a musical rest. To preserve the time structure of the song we replace the rest with the previous non-empty chord. A chord $\mathbf{C}$ in a song $\mathbf{S}$ can then be represented as a subset of musicals pitches e.g. $\mathbf{C} = \{C_3, E_4, G_3\}$. Here the subscript 3 indicates the specific pitch of the note type C. For instance, middle C on a piano is given by C_3 while C_4 is an octave above. We then quotient the set of chords by their pitch class, thereby identifying C_3 with C_4 and so on. By identifying the pitch classes $\mathbf{N} = \{C, C\sharp/D\flat, D, D\sharp, \ldots, A\sharp/B\flat, B\}$ with $\mathbb{Z}_{12}$, we define the set of all chord types as the power set

$$\mathbf{Ch} = 2^{\mathbb{Z}_{12}}.$$

This set of chord types is quite large, so we restrict ourselves to only those chord types which are subsets of 7-chords which are common amongst many generations of music. We build 7-chords in the following manner. First define the sets:

1. Set of fifths $\underline{\mathbf{5}} = \{6, 7, 8\}$ (diminished, perfect, augmented)
2. Set of thirds $\underline{\mathbf{3}} = \{2, 3, 4, 5\}$ (diminished, minor, major, augmented)
3. Set of sevenths $\underline{\mathbf{7}} = \{9, 10, 11\}$ (diminished, minor, major)

We then build the set of C 7-chords as

$$\mathbf{Ch}_7(C) = \{\{0, x, y, z\} \mid x \in \underline{\mathbf{5}}, y \in \underline{\mathbf{3}}, z \in \underline{\mathbf{7}}\}.$$

Now the set of all 7-chords is the union over all translates of $\mathbf{Ch}_7(C)$

$$\mathbf{Ch}_7 = \bigcup_{k \in \mathbb{Z}_{12}} (\mathbf{Ch}_7(C) + k).$$

Hence the collection of all sub-chords of 7-chords is the union of powersets

$$\mathbf{Ch}_{\leq 7} = \bigcup_{\mathbf{C} \in \mathbf{Ch}_7} 2^{\mathbf{C}}.$$

The collection $\mathbf{Ch}_{\leq 7}$ has exactly 529 chord types consisting of single notes, dyads, triads, and tetrachords subsetted from 7-chord types.

Given a chord $\mathbf{C}$ in the song $\mathbf{S}$ we use `music21`'s root, fifth, third, and seventh functions to project $\mathbf{C}$ to a subchord $\tilde{\mathbf{C}}$ lying in $\mathbf{Ch}_{\leq 7}$. From this point of view we project a song to a lower dimensional harmonic space where prediction becomes a more feasible task. We then embed these projected chords into a weighted version of the well-known Tonnetz lattice $\mathbf{T}$ originally developed by Euler, see Figure 2 for a modern unrolled display. The Tonnetz lattice is built using the following construction:

1. Let the underlying node set of $\mathbf{T}$ be $\mathbb{Z}_{12}$
2. For each node $k \in \mathbf{T}$:
   (a) Join node k to the node $k + 7$ by an edge of weight 3
   (b) Join node k to the nodes $k + 3$ and $k + 4$ by edges of weight 5

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_31

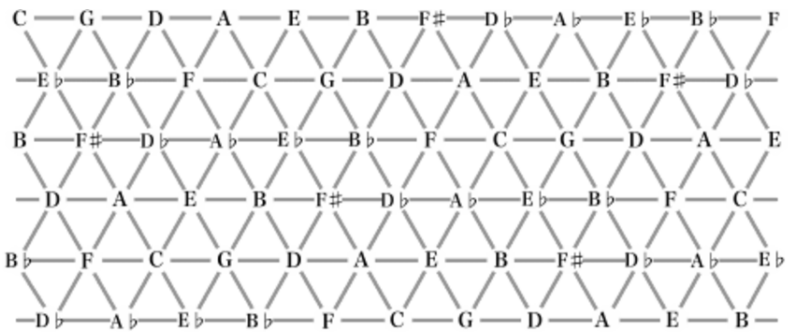

Fig. 2. The unrolled Tonnetz lattice formed by connecting notes to their neighboring fifth, major third, and minor third. Image from [11]

The significance of the weights comes from the harmonic distance between notes. For example, if we strike a key on the piano we would produce a sound wave whose displacement can be expanded into a Fourier series given by

$$\phi(t) = \sum_{n=1}^{\infty} a_n \cos(n\omega t) + b_n \sin(n\omega t).$$

Here the frequency term ω is called the *fundamental* and for $n \geq 2$ the terms $n\omega$ indicate its *overtones*. The first three overtones are the octave, fifth, and a major third respectively. A fifth occurs by tripling the frequency of the fundamental, this is represented in the Tonnetz with an edge of weight 3. Similarly a major third occurs by quintupling the fundamental frequency— so an edge weight of 5 in the Tonnetz. In order to have symmetry across major and minor modes we treat a minor third the same as a major third in harmonic distance. We then define the weighted Tonnetz metric $d_{\mathbf{T}}$ on $\mathbf{T}$ as the shortest-path-distance in $\mathbf{T}$. We can view $d_{\mathbf{T}}$ as a 2nd order approximation of the true harmonic distance between notes. For those who wish to explore the structure of the Tonnetz further, we refer are readers to [11].

We use the Tonnetz metric $d_{\mathbf{T}}$ to induce a metric $d_{\mathbf{T}}^{\text{total}}$ on $\mathbf{Ch}_{\leq 7}$ given by its *total-average*. Explicitly for $\mathbf{C}, \mathbf{D} \in \mathbf{Ch}_{\leq 7}$ we define

$$d_{\mathbf{T}}^{\text{total}}(\mathbf{C}, \mathbf{D}) = \frac{1}{|\mathbf{C}||\mathbf{D}|} \sum_{\substack{x \in \mathbf{C} \\ y \in \mathbf{D}}} d_{\mathbf{T}}(x, y).$$

It can be verified that this defines a metric. One can also use the well-known Hausdorff metric between subsets of a metric space but there will less variability in distances between chords. The total-average distance intuitively measures how every note in the first chord changes with respect to every note in the latter. We then use metric multidimensional scaling (mMDS) via python's `sklearn` package to approximately embed the metric space $\mathbf{Ch}_{\leq 7}$ into $\mathbb{R}^n$ as chord-vectors. If we

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_31

order the space $\mathbf{Ch}_{\leq 7}$ we can represent $d_{\mathbf{T}}^{\text{total}}$ as a matrix D with entries δ_{ij} giving the distance $d_{\mathbf{T}}^{\text{total}}(x_i, x_j)$ between the ith and jth chord respectively. If we let $m = 529$ denote the number of chords in $\mathbf{Ch}_{\leq 7}$ then for a given integer n mMDS solves the convex minimization problem

$$\min_{X \in \mathbb{R}^{m \times n}} \sum_{i,j} \left(\delta_{ij} - d_{ij}(X)\right)^2$$

where $d_{ij}(X)$ denotes the Euclidean distance between the rows $X^{(i)}$ and $X^{(j)}$ of X. The above objective function is called the *stress* of the embedding X. In particular, the package finds a solution by applying an iterative algorithm SMACOF to find the fixed point to the so-called Guttman transform— providing the embedding. See [15] for in-depth analysis. To estimate the optimal embedding dimension for $\mathbf{Ch}_{\leq 7}$ we plot the stress of the mMDS embedding as a function of the embedding dimension and locate where the stress first stabilizes, see Figure 3. We estimate an embedding dimension of $n = 20$ using this method. We conjecture that $d_{\mathbf{T}}^{\text{total}}$ cannot be exactly embedded into Euclidean space. This is supported by the non-zero leveling off of the stress with dimension. Essentially points in $\mathbf{Ch}_{\leq 7}$ which are far apart will always have an error in their embedded distance due to the "curvature" of the space.

We now have a map $\texttt{Tonnetz2vec} : \mathbf{Chords} \to \mathbb{R}^{20}$. Using this map we can convert a song in MIDI format to a time-series of vectors in $\mathbb{R}^{20}$. We then train an LSTM to predict harmonic changes in music and test the validity of the Tonnetz lattice as a fundamental space for learning musical harmony.

2.2 Model Architecture and Training

We use a network with 2 uni-directional LSTM hidden layers and a linear output layer to train the embedded Bach chorales data set to predict the next chord in a song given the previous measure of chords. This is implemented in Python using the Pytorch library [18]. For each element(chord) in the input sequence (chords series) each layer computes the following functions:

$$\begin{aligned}
i_t &= \sigma(W^{ix}x[t] + W^{ia}a[t-1] + b_i) && \textit{Input gate}\\
f_t &= \sigma(W^{fx}x[t] + W^{fa}a[t-1] + b_f) && \textit{Forget gate}\\
o_t &= \sigma(W^{ox}x[t] + W^{oa}a[t-1] + b_o) && \textit{Output gate}\\
\tilde{c}[t] &= \tanh(W^{gx}x[t] + W^{ga}a[t-1] + b_g) && \textit{Candidate cell state}\\
a[t] &= o_t \odot \tanh(c[t]) && \textit{Hidden state update}\\
c[t] &= f_t \odot c[t-1] + i_t \odot \tilde{c}[t] && \textit{Cell state update}
\end{aligned}$$

where $a[t]$ is the hidden state at time t, $c[t]$ is the cell state at time t and $x[t]$ is input at time t for $t = 1, 2, \cdots, s$. The output of the model is an affine transformation of the hidden state in the last time step $t = s$, i.e.,

$$y[s] = W^{ya}a[s] + b_y.$$

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_31

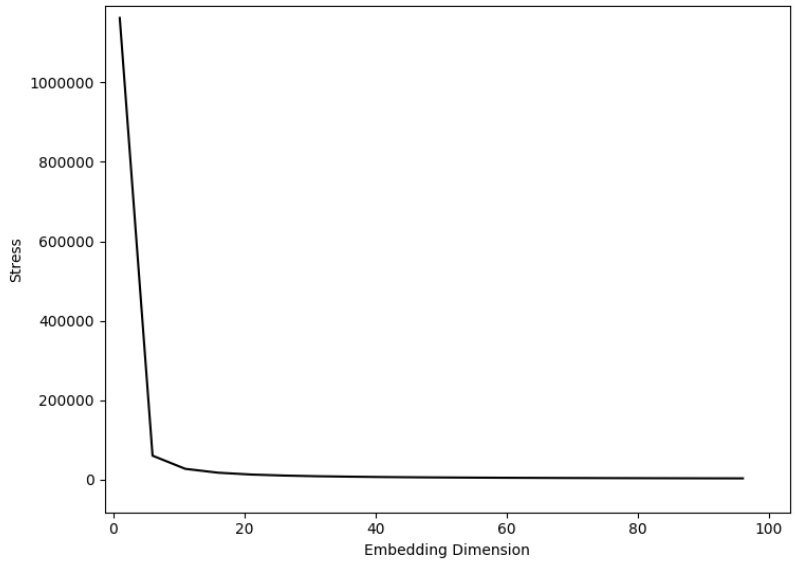

Fig. 3. mMDS stress of the Tonnetz embedding as a function of embedding dimension.

Let n be the input dimension and m be the hidden layer dimension. Now the weight matrices are $W^{ix}, W^{fx}, W^{ox}, W^{gx} \in \mathbb{R}^{m \times n}$, and $W^{ia}, W^{fa}, W^{oa}, W^{ga} \in \mathbb{R}^{m \times m}$. The bias terms are $b_i, b_f, b_o, b_g \in \mathbb{R}^m$ and for the output layer we have the weight matrix $W^{ya} \in \mathbb{R}^{n \times m}$ and the bias vector $b_y \in \mathbb{R}^n$. We parse the embedded Bach chorales data set into input-output pairs such that the input time series consists of the previous 8 chords $\{x[t-7], x[t-6], \cdots, x[t]\}$ and the output being the next chord $x[t+1]$. Each song is parsed using moving window of size 9 (8 input chords and 1 output chord). For the training phase, we train all the input-output pairs occuring in 229 Bach chorales in one batch and backpropagate with Adam optimization algorithm [12] using for the loss function the mean squared error (MSE) in the chord embedding space.

3 Results

In this section we apply the methodology described above to the JSB Chorales dataset. We first study how performance on test data varies depending on the use of simple accuracy (exact chord matches) compared to a relaxed notion of a match (at least 3 notes agree with the true chord). We find that performance is improved with the relaxation, but qualitative behavior and location of local maxima/minima is not noticeably different. Hence, model selection to avoid overfitting is not impacted by the use of different metrics.

In the second part of this study, we vary the size of the hidden layers in our LSTM model and study its effect on prediction accuracy. We see a monotone

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_31

increase in accuracy on the validation data set with a maximum of 0.504 with a hidden layer size of 512 (the maximum value studied), though the values begin to plateau with a hidden layer size of 128.

3.1 Bach Chorales

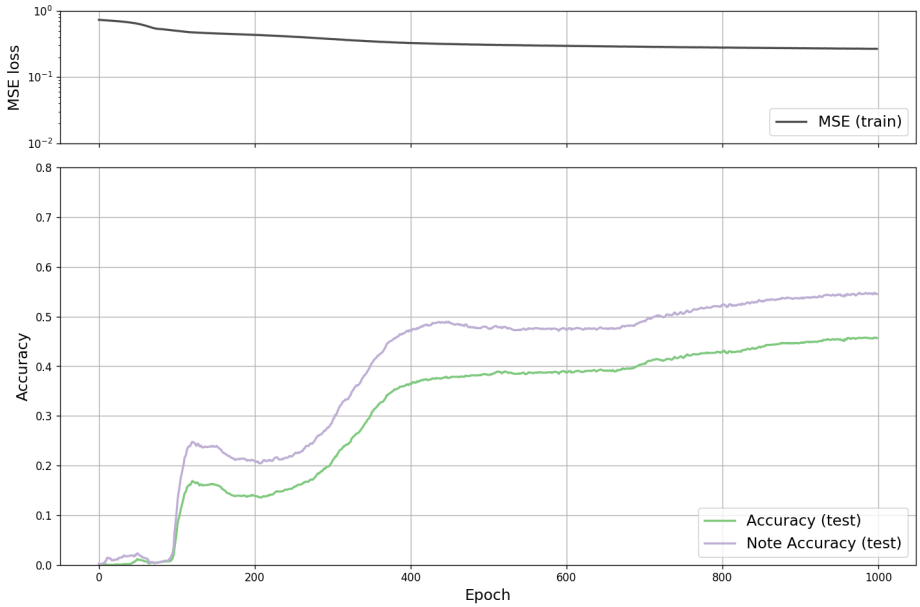

Fig. 4. Value of loss function on training data (top) and two types of accuracy on test data (bottom) for an LSTM with a hidden layer size of 128 over 1000 epochs of training. Similar qualitative behavior is observed using the two types of accuracy, making decisions for model selection the same for both.

Table 1. Training and test set performance for the JSB chorale dataset for various choices of the size of the LSTM hidden layer.

Hidden layer size	4	8	16	32	64	128	256	512
Validation accuracy	0.000	0.000	0.201	0.321	0.412	0.477	0.495	0.504
Validation accuracy (3-note)	0.100	0.073	0.247	0.376	0.467	0.550	0.596	0.614

The publicly available Bach chorale data set comes preprocessed and divided into training, testing, and validation sets; we use this subdivision here. Chords

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_31

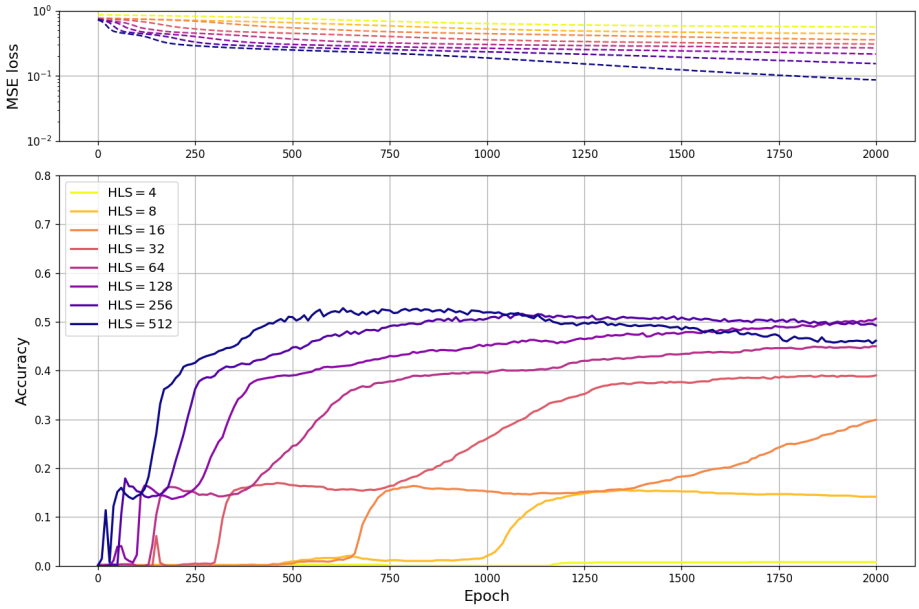

Fig. 5. Value of loss functions (top) and simple accuracy on test data (bottom) for a range of hidden layer sizes over 2000 epochs of training. For hidden layer sizes above 128, maximum accuracies are observed at different points in training.

are mapped to 20 dimensions as described in subsection 2.2. Mean squared error is used as a loss function for training, and test set performance is evaluated either as an exact match "Accuracy" or as "3-note accuracy" which is a relaxation which counts a prediction as correct if at least three notes in the true and predicted chord agree. From the perspective of live music, two chords agreeing on three notes often gives a similar harmonic impression. From a machine learning perspective, this relaxes the classification task to achieving 3-note accuracy. In Figure 4, we visualize an experiment with an LSTM trained on 200 chorales and the accuracy on test data evaluated on 77 chorales. We observe nearly identical qualitative trends in both accuracy and note accuracy, with approximately a difference of ten percentage points in reported. While a 3-note accuracy may be of further interest as more fine-tuned models are developed, for the purposes of model selection we find it sufficient to restrict to simple accuracy (exact matches).

We now study the influence of the size of the hidden layer of the LSTM in an otherwise fixed model training scenario. Decay of the loss function on training data as well as performance on test data are visualized in Figure 5 for various hidden layer sizes. A model is chosen based on optimal performance on test data during training to reduce the likelihood of overfitting. While there is non-monotonic trends in qualitative behavior as the size of the hidden layer increases, we see a general increase in performance on test data. An optimal

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_31

model is selected based on the epoch achieving maximum accuracy, then the performance is evaluated on the validation data set.

The resulting validation accuracy is in Table 1. For each hidden layer size, the performance of the optimal model on the validation data tracks that of the optimal test data performance quite closely. When the hidden layers are significant bottlenecks, of sizes 4 or 8, training is quite slow and the respective performance on validation is extremely poor. For sizes 16 upwards, we see a monotonic improvement in prediction which decelerates at a hidden layer size of 128. This gives strong evidence that successful chord prediction can be done without an extremely large neural network. Such small networks are useful for the purposes of interpretability and detailed analysis, especially if one seeks to later extend such a model for harmonic structure to also incorporate, for example, pitch and tempo.

4 Conclusion

We have developed a framework for machine learning on music which incorporates music theory via the Tonnetz lattice. This is distinguished from a one-hot encoding of notes in the chord which provides little intuition and has many challenges associated with it in the translation from output layer to identifying predicted notes or chord(s). By contrast, this embedding allows one to work directly with chords, which are often more intuitive for musicians in understanding broader musical structure. There are a broad range of directions to investigate in optimizing performance and predictions made by an LSTM model using this embedding. However, the prediction accuracy results of this paper are promising for learning the harmonic structure within a musical genre. This suggests a path forward to predict music by learning its harmonic, melodic, and rhythmic structures separately. From the machine learning perspective, understanding the performance and behavior of predictions using this embedding is of great interest. Analysis of the model in terms of well-known chord transitions and its behavior on different genres is also of note. Admittedly, our study in this paper focuses on the Western musical tradition in a genre of music with rigorous rules; towards this we are very interested in exploring how such models behave when trained in other musical traditions, and what such a model can do when asked to make predictions across culture.

References

1. Greff, K. et al.: LSTM: A Search Space Odyssesy. IEEE Transactions on Neural Networks and Learning Systems **28**, 2222–2232 (2017)
2. Hochreiter, S. et al: Long Short-term Memory. Neural computation. **9** (8)., 1735–80. (1997)
3. Madjiheurem, S. et. al. Chord2Vec: Learning Musical Chord Embeddings. https://doi.org/10.13140/RG.2.2.15031.93608 (2016)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_31

4. Eck, D., Schmidhuber J.: A First Look at Music Composition using LSTM Recurrent Neural Networks. Technical Report. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale. (2002).
5. Eck, D., Schmidhuber, J. Finding temporal structure in music: blues improvisation with LSTM recurrent networks. Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing, 747-756. (2002).
6. Lyu, Q., Wu, Z. et al: Modelling high-dimensional sequences with Lstm-Rtrbm: Application to polyphonic music generation: Twenty-Fourth International Joint Conference on Artificial Intelligence. (2015).
7. Eck, D., Lapalme, J.: Learning musical structure directly from sequences of music. University of Montreal, Department of Computer Science, CP **6128** 48. (2008).
8. Coca, A., Coorêa et al: Computer-aided music composition with LSTM neural network and chaotic inspiration. The 2013 International Joint Conference on Neural Networks (IJCNN), 1-7 (2013).
9. Boulanger-Lewandowski, N. et al. Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription. (2012). `http://www-etud.iro.umontreal.ca/~boulanni/icml2012`
10. Cuthbert, M., Ariza, C.: music21: A toolkit for computer-aided musicology and symbolic music data. International Society for Music Information Retrieval. (2010). `https://web.mit.edu/music21/`
11. Bergstrom, T., Karahalios, K., Hart, J: Isochords: visualizing structure in music. Proceedings of the Graphics Interface 2007 Conference, 297-304. 10.1145/1268517.1268565. (2007)
12. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. 3rd International Conference for Learning Representations, San Diego. (2015).
13. Brunner, G., Wang, Y., et al: JamBot: Music theory aware chord based generation of polyphonic music with LSTMs. IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI). 519–526. (2017).
14. Sturm, B.: What do these 5,599,881 parameters mean?: An analysis of a specific LSTM music transcription model, starting with the 70,281 parameters of its softmax layer. International Conference on Computational Creativity. (2018)
15. De Leeuw, J.: Applications of Convex Analysis to Multidimensional Scaling. In Recent Developments in Statistics, edited by J.R. Barra, F. Brodeau, G. Romier, and B. Van Cutsem, 133–45. Amsterdam, The Netherlands: North Holland Publishing Company. (1977) `http://deleeuwpdx.net/janspubs/1977/chapters/deleeuw_C_77.pdf`.
16. Tang, C. and Chui, K. et al: Music genre classification using a hierarchical long short term memory (LSTM) model. Third International Workshop on Pattern Recognition. **10828**. (2018).
17. `Tonnetz2vec`. `https://github.com/ekehoe32/Tonnetz` Last accessed on Feb. 7 2020
18. Paszke, A., Gross, S. ,et al.:PyTorch: An Imperative Style, High-PerformanceDeep Learning Library. Advances in Neural Information Processing Systems, 8024-8035 (2019) `https://pytorch.org/`

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_31