Chord Progression and Music Estimation Using LSTM Networks

Senior Project Report

Emirhan Karagül (emirhan.karagul@boun.edu.tr)
Bogazici University

January 7, 2019

Contents

1 Introduction and Motivation

The pursuit of autonomous artistic creation has been a challenging and interesting problem for computer scientists and other disciplines. Especially autonomous music composition problem has been a lingering problem since the invention of modern automata [4]. Computer scientist, psychologists and musicologists have put a vast effort in unraveling musical cognition and the perception of pleasant sounding music and the automatic musical composition attempts peaked in the last years.

System for automatic music generation has a variety of use cases including:

- Helping novice musicians compose pleasant sounding musical pieces
- Generating program music for movies, games and real-life experiences based on the context of the scenes
- Accompanying musicians during jam sessions, enabling a dynamic musical environment for solo musicians
- Giving variable composition ideas for musicians to finish off their pieces, as it can be a very hard task sometimes(especially in classical era)
- Musical Education and practice purposes.

This project aims to implement an automatic musical composition system using tweaked Long short-term memory Network models with the help of insights gathered by the researches, which aims to unravel the processes behind musical perception, chord progressions and compositions[2, 3, 4].

2 State of The Art

2.1 Brief History

Automated music generation has been a computational challenge for a long time. Starting from the '80s various researchers tried to generate music with rule-based, grammar-based systems and with constraint satisfaction problems, CHORAL[6] and Steedman's system[22] are examples for these approaches. They tried to carry the problem to a clausal and grammatical form.

In the late '80s and early '90s people started using machine learning approaches to bring new aspects to the problem. Due to computational limits Brarucha and Todd[2] have tried using a very small neural network with recurrent links in the model. They have compiled early works of musicologists and psychologists to assess the neural networks behaviour. The authors were also able to draw conclusions from the neural networks behaviour, since the neural network was a relatively small network which consisted of 18 total nodes. They were able to train the network to produce chord sequences which were compatible with the Circle of Fifths. Mozer proposed a system working with neural networks, namely CONCERT[15], which had 40 hidden layer units. Mozer has tried to train the system to recognize melodies on specific domains (including random walks, pattern chord walks, and JS. Bach-like composition). The resulting model predicted 95% of the pitches and duration correctly after being trained with 10 relatively simple compositions of J.S Bach. The 95% prediction rate might be a result of relatively short and simple compositions(longest piece consisted of 198 notes in total).

After the start of 2000s many more modern techniques are used to generate Chord Progression sequences and musical compositions, with growing computation power came stronger models and higher accuracy. Many of the researchers used HMM to either generate or estimate Chord Progressions for music. Neural networks and extensions of the networks (Recurrent, convolutional, LSTM) are used to either generate chord progressions or melodic sequences. Midi, Chroma Vector, Piano roll, ABC and text-based input notations were used widely in most of the researches to represent musical data as input for the systems developed.

Matevž Pesek and France Msihelic[19] have used HMM as a layer of their system combined with Compositional hierarchical models. They have processed raw audio data using Q-transform. Raw audio processing methods like Q-transform and FFT are essential for real-time audio processing and melody estimation/recommendation system. These methods will not be

examined in this project, as the processing will be constructed over Midi representations. The authors tried to use Circle of Fifths as the provider of transition probabilities in their HMM implementation. The observation probability matrix was predefined on this project, based on normalizing the similarities of the current time window for the song with Chord Templates of major and minor chords. They have classified the chord progressions of 112 Beatles songs with an accuracy of 50%. In 2007 Helene Papadopoulos and Geoffroy Peeters propose a system[18] including raw audio processing to estimate chord progressions. They use 3 different methods to construct observation probabilities for the HMM. First is constructed by a trained Gaussian model on Chroma Vectors, the second method constructs the observation probabilities solely by music theory ad the third one is similar to the second one with normalized-correlation rather than Gaussian. In all the methods they construct covariance matrices to calculate the observation probabilities for 24 different states. In the third method they compare extracted Chroma Vectors in the audio data and calculate the correlation of the vector and the Chord Template, which are constructed considering the natural harmonicss. The correlations are then normalized to build an observation probability vector. For the state transition probabilities they used 4 different methods namely; Circle of Fifths approach, cognitive approach, Em algorithm and lastly training on musical transcription training sets. In the last method they only consider relative chord transitions rather than all the translations to get rid of the permutational bias.The results of this research show that the best performance in exact chord recognition rate is achieved through the cognitive approach in the state transformation and the method 3 in the construction of observation probabilities, nevertheless the observation probability construction methods did not have a significant impact on the performance. In the next year the authors competed in MIREX (Music Information Retrieval Evaluation eXchange) 2008 with their proposed models[11].

Another project carried out by 3 students from the Center for Computer Research in Music and Acoustics in Stanford[7] use HMM to recommend harmonic content. In the first sections the authors give information about music theory and harmonic background by including the circle of fifths. Circle of Fifths is a representation of musical chords in such an arrangement so that closer chords represent a consonance (pleasant link). The authors talk about their feature vectors in the next section, which includes tonal centroids which is an extended version of chroma vectors. Chroma Vector is a graphical representation of music in a given time interval, it consists of 12 bins for different notes and their pitches(emphasis). Tonal centroids are

another extended 6 dimensional representation of notes which consists of 3 different 2 dimensional features (Major thirds, minor thirds, fifths). They train a HMM with state set of 24 chords and tonal centroids as observation vectors (modeled with multivariate Gaussian distribution) with Viterbi algorithm. They use the HMM after extracting tonal centroids from audio data and they feed into the HMM to get a sequence of most likely chords as output. This project is not a generative one, rather it is a chord recognition classifier for songs without chord annotations. They test their algorithms performance on annotated Beatles' songs

In 2017 a project[17] conducted by EE and CS students of Stanford University have implemented and compared various time series data analysis techniques like Seq2Seq to compose music. They use ABC encoding for data representation with an alphabet of all the notes and pipes. They compare continuous Bag of Words, Character RNN, Sequence-to-Sequence, and Generative Adversarial Networks to compose musical structure. They have tried to train the models for generating sheet music that is syntactically correct. The authors demonstrate the hyperparameter tuning process for all the models and optimization methods they have used. CBOW implementation does not generate any kind of meaningful data, as it was mostly composed of meaningless characters. Although the character RNN outputs a meaningful musical data, the notes mostly have the same timestep in the composition, and it was unable to produce it syntactically correct, as the time bars were not places in the time signature. Seq2seq manages to produce syntactically correct and musical compositions, which sounds pleasant. In the assessment they made a small survey, where they asked the surveyees to classify human-composed songs. In one song the Seq2Seq was assessed to be more human likely than human-composed music. In all the cases Seq2Seq and human composition have familiar results, except the RNN has a significantly lower score.

LSTM networks and tweaked and extended versions of them are used widely for music composition after the 2000s especially in 2017 and 2018 [8, 10, 14, 12, 24, 9, 1, 13]. Douglas Eck and Jurgen Schmidhuber proposed a simple LSTM network model for music composition in 2001[5]. In 2017 a system that can generate polyphonic music using 2 LSTM networks have been propsed[8]; one for chord progression estimation and the second to generate polyphonic music with the given chords. They use the songs in Lakh dataset, which has over 100.000 Midi format songs. In the preprocessing, 48 different chord vectors were selected for the first LSTM's output, and 4 octaves of notes starting from C2 and ending in C6 were selected as the output of the second network. The notes in the song dataset were shifted

to C to obstruct any kind of symmetrical bias in music. Before training the LSTM, they extracted chords from the dataset with easy automation, which took the most occurring 3 notes from a bar to estimate a triad chord. They divide each bar in a song into 8 discrete timesteps to ease the task. They use a one-hot encoding for the chords in the chord LSTM's training process and they converted the one-hot matrix into a 10 dimensional embedding vector. The embedding is then fed into a 256-unit hidden LSTM layer to output each chords likelihood. They use a small seed to generate new chord progressions. In the polyphonic layer they feed the piano rolls and chord progression at the timesteps as input and also include the next chord played in the song to train the network for musical composition (melodies often lead to the next chord in the song, one lethal task is to find additional features to feed into the melody layer as it has a wide range of features to be influenced by). This system has a 512 unit hidden LSTM layer. Adam optimizer with cross-entropy loss is used in the training process. They ignore the same notes that are played consecutively. They also tried an upper and lower threshold for the number of notes to be played simultaneously. The results are published in their youtube channels to be evaluated by the viewers, as there is no way to come up with ground truth for composed music. This project has very interesting key points that can be extended and used in future work.

Hyungui Lim, Seungyeon Rhyu, Kyogu, LeeKyogu Lee built a model[10] that can generate chord progressions from melodies using BiLSTM Networks. All songs in the used dataset are major key and they transpose every song to C major to get rid of bias. They embed each note in a vector which consists of duration, note, octave, the chord of the bar in which the note lies. They then concatenate this rows to build up a matrix to represent a song. The authors mention the long-term context loss in RNN's as a result of vanishing gradients during Back Propagation Through Time. The authors also mention the melodies being dependant on the previous and the next chords, resulting in a symmetrical context in music, which makes BLSTMs useful. They used 12 inputs in the first layer, 2 hidden layers with 128 units in the hidden layer and 24 output units in the output layer. They used Wikifonia.org's song database to train the BLSTM and also 2 other models with HMM and DNN-HMM to compare their results with. The results were 50% performance in BLSTM model, 45% in DNN-HMM model and 40% in HMM models. Although quantitative analysis may seem like the results were not satisfactory, it would be convenient to have in mind that even the annotated chords of the songs can't show the ground truth for the musical chord progression solely, as some major and minor chords consist

of the same three notes in a different order. They then did a survey for 25 non-musicians to evaluate their results qualitatively and asked them to rate the chord progressions assonance or pleasantness on a scale from 0 to 5. The overall score for the BLSTM Network was 3.55 and the score for the original songs were 4.04. The qualitative results show that if the listeners recognized the original song, they tend to give a higher score to the chord progression BLSTM has created, but if the listener was unaware of the songs the score of the HMM were greater compared to the points given by aware listeners to the chord progression produced by HMM.

2.2 Modern Methods

Most of the modern software for composing melodic and harmonic music employ extensions of neural networks. Recurrent neural networks are used in state-of-the-art software to process time-series music data with short-term dependencies. But as musical pieces often have long-term dependencies and contexts LSTM networks showed significantly higher performance in composing and recognizing music. In the last 2 years various projects[13, 1, 9, 24, 12, 14] have implemented different kinds of systems using LSTM networks for this task.

2.2.1 Text-Based LSTM

Keunwoo Choi, George Fazekas, and Mark Sandler and proposed a system[13] to generate chord progressions using text-based LSTM and character-based LSTM. Char-RNN is an architecture, where the problem is reduced to generating character based outputs, which results in a very small input vocabulary but requiring a longer sequence to generate meaningful pieces. In this process Cmaj chord is generated after a sequence of length 4. On the contrary the word based approach only requires one timestep to generate Cmaj but resulting in a greater vocabulary domain. The results of both approaches were satisfying as they have learned some jazz chord progressions and they have generated syntactically correct progression starting with _Start_ and ending with _END_ words. They tried the same approaches for rhythm generation but the system failed to generate a meaningful sequence.

2.2.2 Biaxial LSTM

Deepj[9] is another attempt to compose style based music. The authors employ a biaxial LSTM, which consists of a note axis and a time axis module. before the LSTM network the authors used a 1D convolution for the note

inputs and used a context input for artists they also employed a dynamic embedding for the style of the songs (Baroque, Romantic, Classic). The time axis module only takes relatively higher 12 notes and 12 lower notes into consideration to overcome transposition invariance. They have tested their systems performance with 90 people. Overall the melodies composed by Deepj resulted in a higher human accuracy on classifying samples into genres. This demonstrates that Deepj can compose music with distinguishable styles.

2.2.3 Convolution-LSTM

In a more recent project[24] Convolution LSTM is employed to generate music. The system described by the authors use midi format as input features with a selected window of time. C-LSTM model has 2 convolutional layers, an LSTM layer and a fully connected layer after them. The authors do not take any other musical features into consideration as their model only relies on midi representation. In the assessment they have examined the time-domain, frequency-domain plots and sonograms of different implementations. The overall assessment shows that C-LSTM tends to generate musical compositions which are more human-likely than the sequences generated solely by LSTMs. The plots show that C-LSTM and human-composed music have more fluctuations in frequency and amplitude domain, whereas LSTM compositions show a lack of emotional expressions.

2.2.4 Assessment

Overall implementations including LSTM Networks show a significantly better performance than other methods. Most of the authors have tried extending LSTM Networks with additional layers, different training techniques, embeddings and model definitions with different input representations. Altogether the authors of the modern systems have implemented relatively successful pieces of music, which can be perceived as human-composed and pleasant music. [7]

3 Methods

I have split the project into 2 parts. The first task is to estimate chord progressions of songs, given the chroma representations for the pieces. The second task is to generate note sequences with a Long Short-Term Memory Network model, given audio features of a piece and notes played before the time point of generation. Keras with Tensorflow backend is used for the training and testing processes. Keras Generators were used to provide data when needed for the training and the input data were kept in pandas data frames. An Amazon p2.xlarge instance was used in some parts of the model training as it reduced training time dramatically.

3.1 Chord Progression Estimation

3.1.1 Dataset and Features

For this task I have used the chroma features and chord progression annotations provided by AudioLabs Cross-Composer Dataset[23]. There are 1100 classical songs by 11 different artists in the dataset. The chroma features provided are in 100 ms bins (10 Hz) and the chord progression annotations have the chord label and the start and end timestamps for the chords. The features extracted from this dataset are: key and mode of the songs, name and lifetime of the composers, chroma vectors and chord progression annotations.

3.1.2 Preprocessing

The preprocessing consisted of multiple phases. In the first phase I have written a regular expression to extract the keys and modes of the songs. The metadata about approximately 650 songs have been successfully extracted from the csv files. The lifetimes of the composers were present on the AudioLabs site, and extracted by hand. Extraction of chord progressions and chroma vectors are done by two csv readers. First the chord progression annotations are read with their starting and ending times from the chord progression csv file. The chord progression annotations consisted of Maj, Min, Aug and Dim chords, which then were reduced to just Maj and Min by labeling Aug chord with maj and dim chord by min Then the corresponding chroma vectors are read from the chroma label csv file and summed up to represent the total chroma features during that chord annotation. The chroma vectors are then normalized to sum up to 1. Also all the songs in the dataset is transposed to the key of C. All the features are stored in a

dataframe to be fed into the model by generators. An example summed chroma vector for a chord, which is starting at t=0s and ending at t'=0.4s would be as follows:

time	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0.4	0	0	0	0	0.4	0	0	0	0.1	0	0.4
0.2	0.8	0	0.5	0	0	2.1	0	0.4	0	0.7	0	0.9
0.3	0.7	0	0.4	0	0	1.0	0	0	0	0.6	0	0.7
sum	2.0	0	0.9	0	0	3.5	0	0.4	0	1.4	0	2.0
norm	0.2	0	0.1	0	0	0.4	0	0.0	0	0.1	0	0.2

Table 1: Example Input Vector For the Chord LSTM

3.1.3 Model Definition

Hidden Markov Models were first tried to estimate the chord progressions of the pieces based on chroma vectors. The parameters extracted in [16] were used to estimate the chord progressions of the songs. The hidden Markov model An accuracy of 32% was obtained using the hidden Markov model. The parameters of the Hmm were:

	Shapes	Explanation
Features	12	Chroma Vectors
Components	25	Chord Annotations
Covariance Type	full	Cov Used By GaussianHMM
Start Probability	25	Obtained in [16]
Covariance	25 x 12 x 12	Obtained in [16]
Transition Probability	25 x 25	Obtained in [16]

Table 2: Parameter shapes for Gaussian HMM

After the HMM an LSTM Network is constructed to obtain a model to estimate the chord progressions by the given last n normalized chroma vectors. The model starts with an input unit and then 2 LSTM hidden layers, preceded by two dropout layers in Keras. A time distributed layer with 25 classes and softmax activation is used in the output. Without using the time distributed features of LSTM the model was able to label 40% of the chords correctly. Using the time distributed features with step number of 4, the model was able to predict 82 % of the chords correctly.

Parameters	Values
Number of Steps	4
Hidden Layer 1 & 2	256 units each
Dropout 1 & 2	0.5 each
Epochs	20 (one epoch for all the dataset)
Input Size	14
Output Size	25

Table 3: Parameters used in the Chord LSTM Model

When the step number was changed to 6 during the model construction, the model was able to predict 84% of the chord annotations correctly. But the 2% improvement in the 6 steps model can be neglected for the sake of annotating chords with fewer chroma vectors in the pieces. The model was trained for 20 epochs, as there was no significant improvement after 10 epochs. I also tried to predict the next chord in the sequence using the same model, which was able to predict 24% of the chord annotations in the pieces correctly. Further improvement of this model may result in better accuracy in finding the next chord given the chroma features and this can be also used to generate chord progressions for musicians during live sessions.

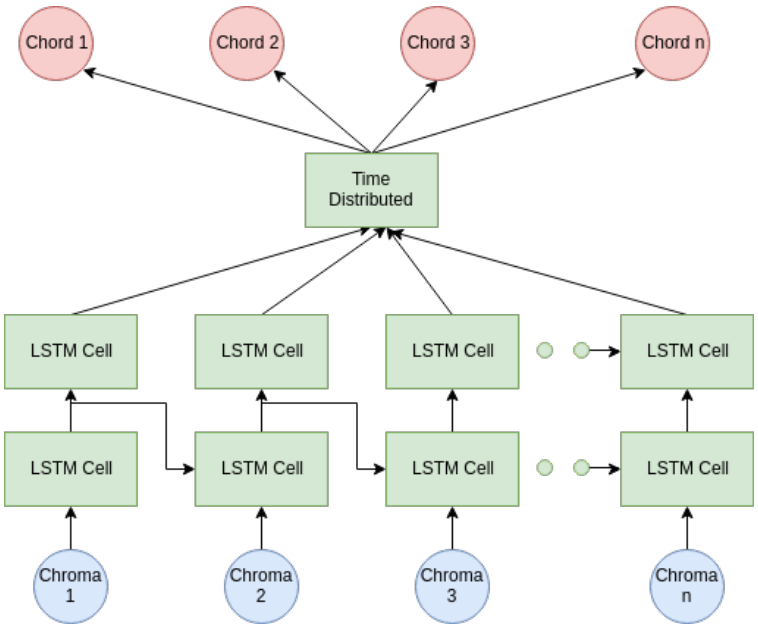

Figure 1: Architecture of the Chord Lstm

3.1.4 Evaluation

The model described in this section was able to label chords with an accuracy of 84% with only using the features: chroma vectors and mode of the songs. The categorical feature of composers and their lifetimes contributed too little on the results and the keys of the songs are discarded, because the songs were all transposed to the key of C. This model can be used to annotate chord labels of classical songs or can be used to generate chord sequences for a classical melody, which can make accompaniment for the music with various instruments easier and sound more pleasing. Comparing the results with additional chord progression estimation projects would be nonessential as they were trained and tested on Mirex chord estimation challenge dataset, which are not available at this time.

3.2 Note Generation

3.2.1 Dataset and Features

In this task songs in midi format were used. Lakh Dataset[20, 21] was used for the source of midi formatted songs. From the dataset a subset of the midi files, LMD-matched were used with the song and artist names annotated for the midi files. The subset consists of 45000 midi files. Additionally I used Spotify API to match the songs to the corresponding Spotify entries. I wrote a program to match the songs and extract their features from Spotify song search, artist search and audio features API. 6300 songs were matched from Spotify API after duplicates were dropped from the dataset. The features obtained from Spotify API are: key, mode, time signature, acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, genre, valence and tempo. Also the model described in the previous section is used to estimate the chord progressions of the songs.

3.2.2 Preprocessing

Music21 python library was used for the preprocessing of the data. Converter class of music21 reads the midi input and converts it to a custom Corpus object. After that I have taken the instrument partition in the corpus with the most notes. The music is transposed to the key C after being read and the notes are shifted so that they stay in the C1-C7 interval in the piano roll. Two additional models were tried in this task, so the three models are:

- Input notes and output notes only remained in C1-C7 interval (73)

- Input notes and output notes only remained in C3-C6 interval (37)
- Input notes and output notes only remained in X3-X6 interval (37)

The last one where the input notes and output notes remained in X3-X6 interval, where X is the chord annotations key corresponding to notes timestep. The last model tries to predict the notes relative to their chord progression key. So the input vectors consisted of number of steps times $\vec{input}$ =<notes[37], duration[17], Spotify features[9], chord annotations[25] > at the last model. At each timestep all the notes that are played are marked on a 2d array. Note features are binary inputs, Duration feature is a one hot vector representing a set of beats starting with 0 beats and ending with 4 beats, increasing by 0.25. Chord Annotations are also a one-hot vector representing all the minor and major chords for each key and one label for no chord. The timesteps are increasing with 0.25 beats each. So a bar of music takes 16 timesteps in the y-axis. For the last model preprocessing was done for a second time to get the positions of the input notes relative to their chord. The models proposed are similar to ones in [9, 24, 12], but Spotify features are introduced in this model and in the third model relative note positions were introduced.

T	C3	C#3	...	C6	Durations	Audio Features	Chords
0	0	1	0	0	010...00	key, mode, etc.	100....000
1	1	1	0	0	001...00	key, mode, etc.	000....001

Table 4: Example input vector for 2 timesteps

3.2.3 Model Definition

For the task of note generation the third model is an LSTM Network with 2 hidden layers. For the input it takes the previously stated input vector in table 4. Step count it selected to be 32 as it consisted of 2 bars of notes. The input layer is preceded with 2 LSTM layers with 256 units and preceded by dropout layers each. In the output there are 2 separate layers.

- **Note Output**: This time distributed layer is activated by sigmoid function as it is a binary matrix for played notes. This output is composed of 37 notes in axis 0 and 32 timesteps in axis 1. Binary cross entropy loss is used for this output.

- **Duration Output**: This time distributed layer is activated by softmax function as it is a one hot categorical output for each different duration. Categorical cross entropy loss is used for this output.

Parameters	Values	Explanation
Dropout	0.5	Dropout rate for regularization
Input	88	Input vector for the last model
LSTM1	256	Size of the hidden LSTM layer
LSTM2	256	Size of the hidden LSTM layer
Out1(sigmoid)	37	Binary vector of played notes
Out2(softmax)	17	One hot categorical durations
Step Count	32	Count of input vectors used to estimate notes
Batch Size	20	Number of input data vectors in 1 iteration
Epoch Count	20	Times to iterate over whole dataset

Table 5: Parameters for the Note LSTM

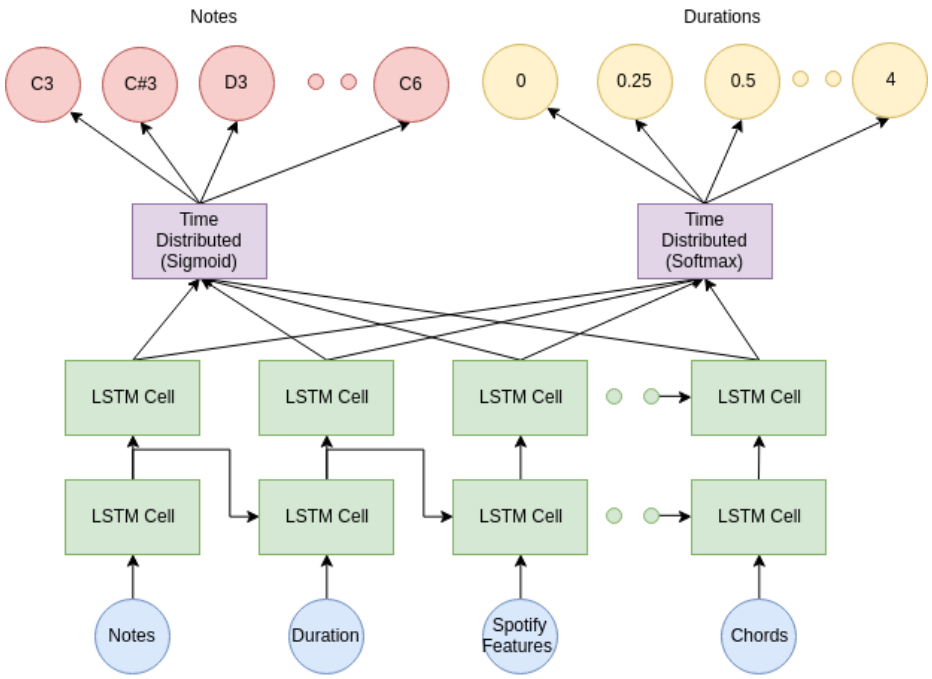

Figure 2: Architecture of the Note Lstm

The weighted loss is 30 times for the first output layer and 1.0 for the

second one. For this task I also used custom error functions where a false negative(missed note) contributed 3 times the loss compared to a false positive. After training all the first three methods failed to generate any kind of meaningful pieces. I have tried to reduce the input and output size to 25 notes by collapsing the piano roll to 2 octaves between C3 and C5.

3.2.4 Evaluation

Metrics used in validation of the model can be found in Table 6. The resulting model was trained with 30 pieces of Johann Sebastian Bach and tested with 8 pieces of him. The training generated a silent model in terms of composition. I tested the model by hand by giving random two bars of music from Bach and by appending the predicted note and duration outputs to the input in each timestep. The model is mostly silent and biased to playing first and 5th notes of the key. Manual thresholds were also used to generate a less silent output from the model, which resulted in generating most confident repetitive notes. So a meaningful and pleasant piece composition was not achieved with these models.

Metric	Values	Explanation
Notes Recall	0.995	Probability of silent notes predicted as silent
Notes Precision	0.041	Probability of active notes predicted as active
Durations	0.534	Categorical accuracy of the predicted durations
Notes	0.97	Binary accuracy of the predicted notes

Table 6: Results obtained from 2 Octave Model

4 Results

Evaluation of tasks are available in sections 3.2.4 and 3.1.4. As the results of the chord estimating Long Short-Term Memory Network suggests, it can be used to transcribe chord annotations for Chroma formatted musical data. The model was not tested on a different genre of musical dataset, it may fail to provide chord progression labels for pop or rock music. But the model can be trained to estimate chord progressions of pop and rock music as well.

Evaluation of the second task suggests that generation of pleasant sounding musical pieces is a challenging task. Although my model is partly similar to previously successful model like [9, 24, 12], it have failed to generate musical compositions. Further improvements of this model, bigger networks and more computational power may lead to a more successful musical composition. Few of the proposed improvements for this model will be discussed in section 6

5 Conclusion and Discussion

The problem of chord progression estimation and generation interesting and challenging, especially for musicians and computer scientists. Using statistical methods to estimate and generate chord progressions relies on structured musical data like midi files, chroma vectors and accurate chord annotations. Building a bigger and structured dataset for this task may improve the results of the proposed models dramatically. Even with a naive look and solution for the problem can result in pleasant sounding and accurate chord progressions. This can help novice musicians and also composers coming up with alternative chord progressions and help them during compositions. It can also be used to transcribe chord progressions in real time so that novice musicians can accompany other musicians with pleasant sounding chords and more harmonious improvisations.

The problem of autonomous musical composition is more challenging and complex compared to chord progression estimation. Even though models for such tasks can be evaluated on accuracy metrics, they are mostly uninformative, as one note sequence may lead to thousands of different harmonious musical pieces. In contrast to the result evaluated in this project, bigger and better models and systems can be constructed to compose meaningful pieces. One of the improvements can be treating notes, duration and chord progressions separately, constructing a more complex system including multiple generators. For instance a model for generating notes should only generate notes when the generated duration for that timestep is not 0.

6 Future Work

Further improvements can be made for the note LSTM model to generate more colorful, verbose and harmonious melodies. I have also found corrupted midi files in the dataset while testing them by hand. So a more structured and rich dataset, which consists of syntactically correct notes and true chord annotations would be really helpful for this task. Improvements for the proposed model can be listed as follows:

- Splitting the model so that the note prediction phase is only activated when there should be notes played (e.g when duration is not 0)
- Building a bigger model with more units in the hidden layers and using more computational power during training to reduce training duration resulting in more time for evaluating and tweaking the models.
- Using custom loss functions: E.g The loss function I used in the last model constructed
- Training a genre-specific model, i.e rock or pop music songs can be easier to learn with compact models
- Adding more regularisation to the model for repetition of notes and being silent may lead to more variant and consonant melodies.

7 References

[1] L. Z. Andrés E. Coca, Débora C. Corrêa, "Computer-aided music composition with lstm neural network and chaotic inspiration," in *The 2013 International Joint Conference on Neural Networks*, August 2013.

[2] J. J. Bharucha and P. M. Todd, "Modeling the perception of tonal structure with neural nets," *Computer Music Journal*, vol. 13, no. 4, pp. 44–53, Winter 1989.

[3] L. L. C. Carol L. Krumhansl, "A theory of tonal hierarchies in music," *Music Perception. Springer Handbook of Auditory Research*, vol. 36, pp. 51–87, June 2010.

[4] E. C. Dorien Herremans, Ching-Hua Chuan, "A functional taxonomy of music generation systems," *ACM Computing Surveys*, vol. 50, no. 5, pp. 1–30, September 2017.

[5] J. S. Douglas Eck, "A first look at music composition using lstm recurrent neural networks," 2001.

[6] K. Ebcioğlu, "An expert system for harmonizing chorales in the style of j. s. bach*," *J. Logic Programming*, vol. 8, pp. 145–185, 1990.

[7] Y. Y. Elie Adam, Elie Noune, "A system for music recommendation based on harmonic content," https://ccrma.stanford.edu/$\sim$noune/DSP{_}files/FYP{_}Final{_}Presentation.pdf, accessed: 05.10.2018.

[8] R. W. J. W. Gino Brunner, Yuyi Wang, "Jambot: Music theory aware chord based generation of polyphonic music with lstms," in *29th International Conference on Tools with Artificial Intelligence, ICTAI 2017, Boston, MA, USA*, November 2017.

[9] G. W. C. Huanru Henry Mao, Taylor Shin, "Deepj: Style-specific music generation," *arXiv:1801.00887v1*, January 2018.

[10] K. L. L. Hyungui Lim, Seungyeon Rhyu, "Chord generation from symbolic melody using blstm networks," in *18th International Society for Music Information Retrieval Conference 2017*, October 2017.

[11] G. P. Hélène Papadopoulos, "Chord estimation using chord templates and hmm," in *MIREX 2008*, September 2008.

[12] D. D. Johnson, "Generating polyphonic music using tied parallel networks," *Computational Intelligence in Music, Sound, Art and Design*, pp. 128–143, March 2017.

[13] M. B. S. Keunwoo Choi, George Fazekas, "Text-based lstm networks for automatic music composition," April 2016.

[14] P. L. A. K. Marko Docevski, Eftim Zdravevski, "Towards music generation with deep learning algorithms," in *CiiT 2018 - 15th International Conference on Informatics and Information Technologies*, April 2018.

[15] M. C. MOZER, "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing," *Connection Science*, vol. 6, no. 2-3, pp. 247–280, October 1994.

[16] E. Nichols, D. Morris, and S. Basu, "Data-driven exploration of musical chord sequences." ACM, February 2009, pp. 227–236. [Online]. Available: https://www.microsoft.com/en-us/research/publication/data-driven-exploration-musical-chord-sequences/

[17] Y. I. Nipun Agarwala and A. Sly, "Music composition using recurrent neural networks," https://web.stanford.edu/class/cs224n/reports/2762076.pdf, accessed: 20.10.2018.

[18] H. Papadopoulos and G. Peeters, "Large-scale study of chord estimation algorithms based on chroma representation and hmm," in *2007 International Workshop on Content-Based Multimedia Indexing*, June 2007.

[19] M. Pesek and F. Msihelic, "Hidden markov model for chord estimation using compositional hierarchical model features," in *Zbornik dvaindvajsete mednarodne Elektrotehnivske in ravcunalnivske konference*, January 2013.

[20] C. Raffel. The lakh midi dataset v0.1. [Online]. Available: https://colinraffel.com/projects/lmd/

[21] ——, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," Ph.D. dissertation, COLUMBIA UNIVERSITY, 2016.

[22] M. J. Steedman, "A generative grammar for jazz chord sequences," *Music Perception*, vol. 2, no. 1, pp. 145–185, October 1984.

[23] C. Weiß, "Computational methods for tonality-based style analysis of classical music audio recordings," Ph.D. dissertation, Ilmenau University of Technology, Ilmenau, Germany, 2017. [Online]. Available: http://www.db-thueringen.de/receive/dbt_mods_00032890

[24] Q. C. Yongjie Huang, Xiaofeng Huang, "Music generation based on convolution-lstm," June 2018.

Glossary

Back Propagation Through Time A gradient-based technique to train certain type of sequence based RNN's. 7

BLSTM Bidirectional LSTM Networks, with additional forward and backward activation capabilities.. 7, 8

chord progression Succession of musical chords, on which melody and harmony is built.. 7, 8

Chord Template Mathematics is what mathematicians do. 5

Chroma Vector A graphical representation of music in a given time or time interval, which consists of 12 bins for different notes and their pitches. 4, 5

Circle of Fifths A representation of musical chords in such an arrangement so that closer chords represent a consonance. 4, 5

HMM Hidden Markov Models, is a probabilistic representation of events and their observations. 4–6, 8

LSTM Long Short-Term Memory Networks, is an extension of Neural Networks with extensive memory capabilities. 6–9, 23

Midi A data format for music with the information of the notes and the durations of them. The instrument is not encoded in midi. . 4–6

natural harmonics A wave with a frequency that is a positive integer multiple of the original one. 5

polyphonic Music that consists of multiple simultaneous melodies.. 6

RNN Recurrend Neutal Networks is an extension of Neural Networks with dynamic capabilities on time series data.. 6, 7, 23

sonogram a two dimensional map representation with time and frequency axes and colors as the intensity. 9

triad A basic chord consisting of 1st 3rd and 5th note in a mode. The 3rd note differs in major and minor modes.. 7