Learning musical structure with Deep Neural Network

Abstract

Consistent automated identification of song structure is a process that would greatly improve computational analysis and generation of music, in terms of both scope and accuracy. Current solutions have employed hard-coded optimization algorithms to solve this problem to varying degrees of success, but little has been done to potentially solve this problem using deep learning. In this paper, we present a novel deep learning model to identify song structure. After exploring several architectures, we find that employing an optimized convolutional neural network yielded the most accurate results.

1 Introduction

Within the field of computational music generation and analysis, a significant challenge has been automated comprehension of song structure. Provided with song structure, computational algorithms are better reinforced to learn from musical data, primarily within the field of music generation. Furthermore, automated analysis of song structure allows us to artificially generate music with the structure as some input song. This is particularly useful within the field of Rhythmic Auditory Stimulation (RAS), a clinically proven but exceedingly costly form of neurological therapy. RAS is a novel field of rehabilitative studies that aids with gait function rehabilitation through neurological music therapy [Wan+22]. Effectively, if we were to develop an application that allows users/patients to input songs that maintains a beat and structure that is healthy for their rehabilitation, an understanding of the song's structure would allow generated music to maintain the input song's structure, while varying melodically [Ye21]. The framework for such an application has already been presented by [Dai+21], with the paper noting automated music structure analysis as a ever-present challenge. Effectively, automating music structural analysis would allow rehabilitative processes to be automated, lowering cost overheads and increasing treatment accessibility. Indeed, this was the initial source of inspiration for this very project.

Within the last year, the literature that has begun applying deep learning to structural musical analysis, but no paper has developed a neural network to precisely return a song's structure. As such, we sought to construct such a model. Neural networks were a natural framework to approach this problem from, given their strong ability to learn from the structure of (what in this case would be musical) data.

With all of this in mind, we step back to actually define "musical structure", so as to formally define our problem statement. Generally, musical structure refers to the segmentation of musical phrases into unique sections. For example, on a very high level, the verse, chorus, and bridge of a song represent different sections. Having said this, structure can and should be interpreted in a hierarchical way - larger structures have repeating patterns and substructures within them that we should be able to recognize, so as to better encompass the song. For example, if each chorus had two distinct musical phrases, we would want to be able to distinguish between them. As such, we can understand structure as an ordered, "quantized" (i.e. each phrase element has an associated length), sequence of phrases.

Having scaffolded a formalized definition of musical structure, we note long short term memory neural nets (LSTMs) and convolutional neural networks (CNNs) as promising network architectures to employ. CNNs in particular were theoretically most promising, given their ability to train deep feature embeddings.

2 Related Work

Generally, we can categorize existing adjacent literature into two classes. The first consists of works employing other methods to automate the music structure analysis process. These include [DZD20], which provides the theoretical music grounding for this exploration. In their paper, [DZD20] employ hard coded optimization algorithms to determine music structure from graphs. By converting musical data into a directed weighted graph and constructing thresholds for phrase similarity, they define a *Structure Description Length* formula, which they optimize to shorten [DZD20]. One other such paper is [Shi+19], which constructs a "homogeneity, repetitiveness, regularity-aware hierarchical hidden semi-Markov Model" that performs static music structure analysis. Another such paper is [HHT06], which develops an automatic time-span tree analyser (ATTA) based on the generative theory of tonal music (GTTM). They build on the musicology work done by [Mar05], which rigorously defines musical objects, symbols, and concepts to construct a hierarchical and generative representation scheme for musical structure.

The other class of related works consists of deep learning approaches to music segmentation. Over the last 2-3 years, there has been a notable surge in papers applying machine learning methods to meaningfully segment music in some way. Papers such as [All+19] employ unsupervised learning to segment 32 movements from Mozart's string quartet. More generally, [OBr16] is an "early" adopter of convolutions neural networks (CNNs) for music segmentation. Building on this work, [McC19] train CNNs to learn music deep feature embeddings, which they then apply to improve classical music segmentation algorithms. Choosing a slightly different direction, [SNY20] construct an LSTM-HSMM hybrid model for a segmentation statistical analysis method that classifies macro-structural musical parts such as intro, verse, or chorus.

Looking at the literature scope, we can make an observation. While deep learning experts have developed exciting models for music *segmentation*, they lack the music theory grounding so as to produce a generalizable *structure* analysis tool, which the first class of papers have attempted to do. Indeed, this is how these papers have in tandem shaped our approach to the problem - the first class of papers presented us with a theoretical framework for understanding and formalizing musical structure, while the second class of papers pointed us to deep learning architectures worth considering when looking at music segmentation (which structural analysis is a specialized, higher order instance of).

3 Data

This project made use of POP909, a Chinese Pop songs MIDI dataset published in 2020 [Wan+20]. It contains 909 songs each with multiple piano arrangements (MIDI), aligned to a lead melody, original audio, manually labelled tempo curves, as well as beat, key and cord labels generated with information retrieval algorithms. The MIDI format has the advantage of greater and more reliable rhythmic information along with the possibility of cleanly separating melody [DZD20]. We used the addition manual structure labelling provided in [DZD20] as benchmarks for our models.

While the dataset contains only 909 songs, each song is a rich feature set in it of itself, containing hundreds of beats. Combined with the fact that each data entry contains two human labels, we end up with a relatively rich data set containing more than 100000 seconds of music.

With regard to the output shape (i.e. what we are trying to produce), we call upon the output format presented in [DZD20], as not only is it very extensively and coherently defined, but it also allows us to make use of the manually annotated dataset referred to in the paper. We explain this output with an example:

```
i4A4B8A4A4b4B8A4A4b4b4A4A4b4A4o3
```

In this phrase, each letter is indicative of a phrase, with the accompanying number the amount of beats this phrase lasts for. Recurring letters imply the re-emergence of the phrase. Lower case letters indicate phrases without melody lines - for example, b4 refers to the same phrase as B4, but without the melody line (they could share the same chord sequence and percussive elements). The letters I and O refer to intro and outro respectively - in the structure output above, we have a 4 beat intro (without a melody), and a 3 beat outro (also without a melody). The letter X refers to any unique phrase that only appears once in the song - each x phrase is distinct (otherwise they would be categorized as a phrase by another letter).

(For CPSC 552, Nhi Nguyen is specialized in preprocessing the input data and ground truth labels to be compatible with the neural networks.)

3.0.1 Preprocess Input

From the chord and melody data extracted from the MIDI file, we represent each song using 2 dimensions: time and musical features. We time step represent 1 beat, whihc we suppose to represent a 16$^{\text{th}}$ note. In each time step, there are nine features: 5 features to represent the chord and 4 features to represent the melody in that time step. The 5 chord features include 1 value for the chord root note and up to 4 values for the chord tones. The 4 melody features include 4 values corresponding to the 4 melody notes played during that beat. Notes are represented by positive integers corresponding to their position on the piano keyboard. Rests, defined as beats when no notes are played, are also considered notes and represented by the value 0. Since the length of the songs varies, we padded all songs with 0s at the end. This is done so that the length of each song matched the maximum length among all songs, allowing us to use each as as an input for the network.

3.0.2 Preprocess human labels

We will represent the human labels using two binary vectors and one binary matrix. These are the phrase vector, melody vector, and pattern matrix. Let's consider the following hypothetical example for how we derived the labels. Assume our song has 30 beats. First, since a beat either has or doesn't have a melody, there are 30 values in the melody vector (one for each beat). Then, we want to understand whether the beat i is in the same phrase as beat $(i + 1)$ - so there are an addition of 30 values in the phrase vector (one per beat). Now, if we extend this comparison for beat i to another non-consecutive beat j, we have a total of $30 \times 30 = 900$ values. In sum, our output has $30 + 30 + 900 = 960$ different classes to determine the structure of the piece of music. We treat the structure analysis problem as a multi-class binary classification with 960 classes, where each song can be categorized into multiple classes. The ground truth human labels would be translated into a binary vector of length 960.

In generally, we divided a piece of music into small sections and represent its structure as two binary vector u, v and a binary matrix M, where: u_i represents whether section i has a melody, v_i represents whether section i and $(i + 1)$ are connected (i.e. in the same phrase), and M_{ij} represents whether section i and j mirror each other (i.e. re-emergence of phrase). u, v, and M capture both short-term and long-term structural relationship in a musical piece.

Like our pre-processed input, since some songs are shorter or longer than others, we padded the songs with 0s at the end so that the length of each song matched the maximum length among all songs.

In total, we had two humans labels per song, and we used this pre-processing pipeline to transform these labels as necessary for the training of the network.

4 Model

(For CPSC 552, Nhi Nguyen is specialized in implementing the LSTM model.)

4.1 Loss function and Metrics

We decided to use Binary Focal Cross Entropy as our loss function. This loss function generalizes binary cross-entropy by introducing a hyperparameter called the focusing parameter γ, that allows hard-to-classify examples to be penalized more heavily relative to easy-to-classify examples. Since we padded many of the songs, there will be many instances in which labels will be easy to predict (with '0' since they are padded), so this loss function is able to focus more on the '1' labels.

We used two accuracy metrics. The first was Binary Accuracy, which calculates how often predictions match binary labels ('1' or '0'). However, since the padding of input and output data artificially increase the binary accuracy (models can be very good at identifying that the a bunch of values at the end are 0 paddings, which is not important to our task), we decided on another metric better for comparison: Binary

Intersection-over-Union (IoU), which computes the Intersection-over-Union ratio

$$\frac{\#(\text{Predict: } 1,\ \text{True: } 1)}{\#(\text{Predict: } 1,\ \text{True: } 1) + \#(\text{Predict: } 0,\ \text{True: } 1) + \#(\text{Predict: } 1,\ \text{True: } 0)}.$$

Binary IoU would weigh the accurate prediction (1, 1) much more than (0,0), as it is very likely (0,0) is just correct prediction about padding.

We used Binary IoU to compare the two human labels per song in the original dataset. They achieved an 88% accuracy. This value acted as our human performance benchmark.

4.2 CNN

In our CNN model, we had an input layer, followed by 4 1D convolution layers, each of which had regularizations: L2 kernel regularization and a dropout parameter. Each of these layers additionally had a pooling operation. Note that we used the 1D version of convolution (and pooling) because our data is 1 dimensional, which can represent convolution in the temporal domain. We used multiple convolutional layers with increasing number of channels in the hope to extract as much important information about the piece of music as possible. The output of these four layers were flattened before being passed to a three-dense-layer multi-class classifier. A ReLu activation function was used in every layer apart from the input layer (7 layers in total), except the last layer used a Sigmoid activation function so that the output was between 0 and 1.

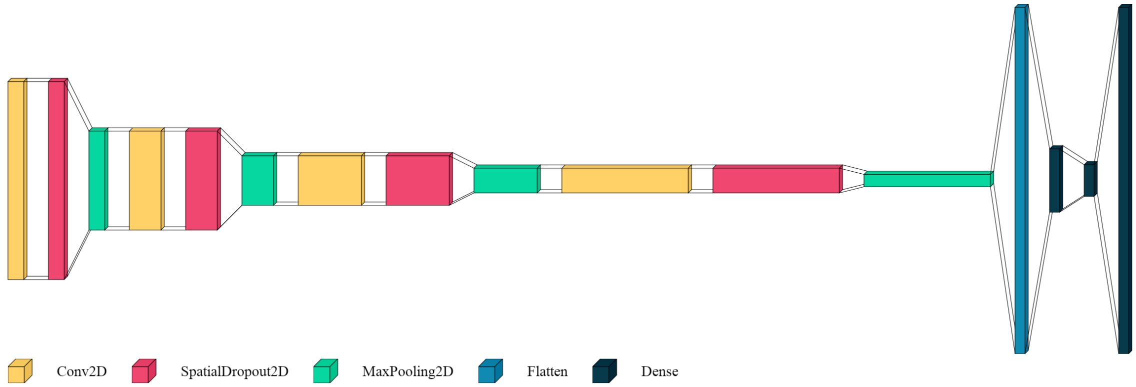

Figure 1: CNN model architecture

We tried a total of 48 combinations of hyper parameters in evaluating our model: `kernel size = [5, 10], kernel stride = [1, 4], dropout rate = [0, 0.2], l2 regularization = [0, 0.01, .05], pool size = [2], pool stride = [2], learning rate = [1e-4], focal gamma = [0, 5.0], epochs = [100], batch size = [50]`. We used the combination of hyperparameters which achieved the greatest Binary IoU metric as our final model.

4.3 LSTM

In our LSTM model, we had an input layer, followed by 3 stacked LSTM layers with L2 regularization. We used a stacked instead of a single LSTM layer in the hope of extracting more important information about the piece of music. Each of these three layers were followed by a dropout layer as a standard regularization. We used the Adams optimizer in this model. The output of these three layers were passed to a one-layer multi-class classifier with a Sigmoid activation function that outputs a binary vector.

We tried a total of 16 combinations of hyper parameters in evaluating our model: `units = [50, 100, 200, 400], dropout rate = [0, 0.2], learning rate = [1e-4], focal gamma = [0, 5.0], epochs = [100],` `batch size = [50]`. We used the combination of hyperparameters which achieved the greatest Binary IoU metric as our final model.

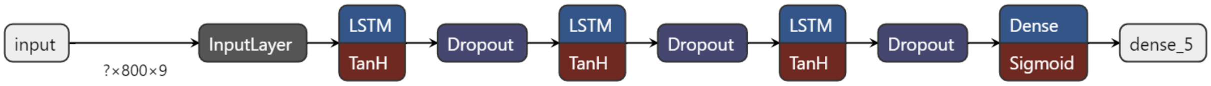

Figure 2: LSTM model architecture

5 Empirical Results

Our CNN model achieved 72.8% Binary IoU accuracy. Also, the model achieved a Binary Accuracy of 98%. The hyper parameters for this model are as follows: `kernel size = 5, kernel stride = 1, dropout rate = 0.0, L2 regularization = 0.0, pool size = 2, pool stride = 2, learning rate = 1e-4, focal gamma = 5.0, epochs = 100, batch size = 50`. Interestingly, we found that the best performing model had no regularization with `dropout rate = 0` and `L2 regularization = 0`. The best model's focal gamma parameter was greater than 0, indicating that it is beneficial for the model on each iteration to focus on learning the harder-to-classify classes that it has missed in previous iterations.

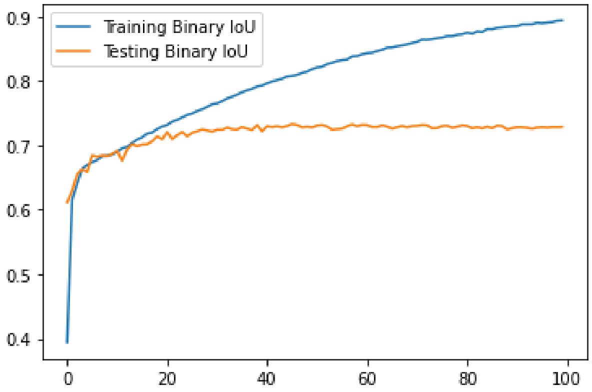

Figure 3: CNN training and testing accuracy

We plotted the training Binary IoU and the testing Binary IoU to assess the model generalization power. It seems that after around 40 epochs, the training accuracy continues to increase but the testing accuracy plateaus. It seems that we could have trained our model for a smaller number of epochs and still gotten a similar performance on test accuracy.

Our LSTM model achieved just under 61.4% Binary IoU accuracy. Also, the model achieved a Binary Accuracy of 97.8%. The hyper parameters for this model are as follows: `dropout rate = 0.2, pool size = 2, focal gamma = 5.0, epochs = 100, batch size = 50`. In contrast to our CNN model, we found the best performing model had regularizations: `dropout rate = 0.2`. The best model's focal gamma parameter was greater than 0, indicating again that it is beneficial for the model on each iteration to focus on learning the harder-to-classify classes that it has missed in previous iterations.

We also plotted the training and testing Binary IoU for our LSTM. It seems that the model doesn't learn much after the first 10 epochs, and there is some strange fluctuation in the testing accuracy at around 60 epochs. Similar to our CNN, we could have trained the model for a much smaller number of epochs and still gotten a similar test accuracy.

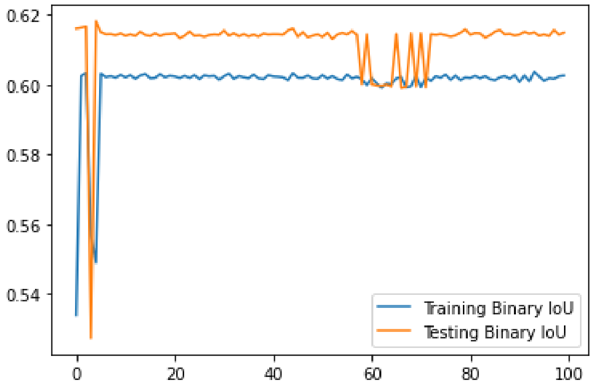

Figure 4: LSTM training and testing accuracy

Our baseline performance was the 88% Binary IoU accuracy achieved by human labelers. It is important to note that these human labelers are professional musicians, so for our CNN network to achieve 73% accuracy is impressive.

6 Conclusion

Evaluating our different architectures, we found that our CNN model was successful in identifying the song structure, achieving a 73% binary IoU accuracy relative to the two human input labels. To contextualize this, the human structure labels themselves, independently annotated by musicologists, only share an 88% similarity score. This is testament to the non-deterministic nature of music analysis - indeed, despite the formalized definition and evaluation of music structure, there will always be room for interpretation when evaluating artistic mediums such as music. Recentering ourselves, our aim for this project was to be able to produce musical structures that were similar enough such that when generating novel music with the same structure, there is a semblance of input song structure. With a similarity score of 73%, we feel as though we were able to tend closer to this goal.

Moving forward, there are several architectures still worth exploring that have the potential to yield interesting results. For example, recently, transformers have been shown to be incredibly powerful on specific natural language processing tasks, with the ability to retain entire phrases/sections as opposed to having a defined recursive attention window present in LSTMs. Therefore, a future direction would include the modelling of a Transformer that processes the music sequence all at once rather than note by note, to compare results to our CNN model.

Another architecture worth exploring are spectral graph convolutional networks. Over the last two decades, there has been notable work done by the likes of Princeton Professor Dmitri Tymoczko in defining musical compositions as graphs and/or topological spaces [see *A Geometry of Music, 2010*]. As we are working to understanding musical structure, an entirely distinct and novel way to approach this problem would have been to implement different methods for converting music to topological data before then running a spectral graph convolution network to attempt and learn musical structure.

7 Bibliography

References

[Mar05] Alan Marsden. “Generative structural representation of tonal music”. In: *Journal of New Music Research* 34.4 (Dec. 2005). Publisher: Routledge _eprint: https://doi.org/10.1080/09298210600578295, pp. 409–428. ISSN: 0929-8215. DOI: 10.1080/09298210600578295. URL: https://doi.org/10.1080/09298210600578295.

[HHT06] Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. "Implementing "A Generative Theory of Tonal Music"". In: *Journal of New Music Research* 35.4 (Dec. 2006). Publisher: Routledge _eprint: https://doi.org/10.1080/09298210701563238, pp. 249–277. ISSN: 0929-8215. DOI: 10.1080/09298210701563238. URL: https://doi.org/10.1080/09298210701563238.

[OBr16] Tim O'Brien. "MUSICAL STRUCTURE SEGMENTATION WITH CONVOLUTIONAL NEURAL NETWORKS". en. In: *International Society for Music Information Retrieval Conference* 17 (2016), p. 9.

[All+19] Pierre Allegraud et al. "Learning Sonata Form Structure on Mozart's String Quartets". en. In: *Transactions of the International Society for Music Information Retrieval* 2.1 (Dec. 2019). Number: 1 Publisher: Ubiquity Press, pp. 82–96. ISSN: 2514-3298. DOI: 10.5334/tismir.27. URL: http://transactions.ismir.net/articles/10.5334/tismir.27/.

[McC19] Matthew C. McCallum. "Unsupervised Learning of Deep Features for Music Segmentation". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (May 2019). arXiv: 2108.12955, pp. 346–350. DOI: 10.1109/ICASSP.2019.8683407. URL: http://arxiv.org/abs/2108.12955.

[Shi+19] Go Shibata et al. "Statistical Music Structure Analysis Based on a Homogeneity-, Repetitiveness-, and Regularity-Aware Hierarchical Hidden Semi-Markov Model". In: (Nov. 2019). Conference Name: International Society for Music Information Retrieval Conference (ISMIR 2019) Pages: 268-275 Publication Title: Proceedings of the 20th International Society for Music Information Retrieval Conference Publisher: ISMIR. DOI: 10.5281/zenodo.3527796. URL: https://zenodo.org/record/3527796.

[DZD20] Shuqi Dai, Huan Zhang, and Roger B. Dannenberg. "Automatic Analysis and Influence of Hierarchical Structure on Melody, Rhythm and Harmony in Popular Music". In: *arXiv:2010.07518 [cs, eess]* (Oct. 2020). arXiv: 2010.07518. URL: http://arxiv.org/abs/2010.07518.

[SNY20] Go Shibata, Ryo Nishikimi, and Kazuyoshi Yoshii. "MUSIC STRUCTURE ANALYSIS BASED ON AN LSTM-HSMM HYBRID MODEL". en. In: *Society for Music Information Retrieval Conference* 21 (2020), p. 8.

[Wan+20] Ziyu Wang et al. "POP909: A Pop-song Dataset for Music Arrangement Generation". In: *arXiv:2008.07142 [cs, eess]* (Aug. 2020). arXiv: 2008.07142. URL: http://arxiv.org/abs/2008.07142.

[Dai+21] Shuqi Dai et al. "Controllable deep melody generation via hierarchical music structure representation". In: *arXiv:2109.00663 [cs, eess]* (Sept. 2021). arXiv: 2109.00663. URL: http://arxiv.org/abs/2109.00663.

[Ye21] WANG Ye. *Music & Wearable Computing for Health and Learning:* en. National University of Singapore, 2021.

[Wan+22] Lei Wang et al. "Effects of Rhythmic Auditory Stimulation on Gait and Motor Function in Parkinson's Disease: A Systematic Review and Meta-Analysis of Clinical Randomized Controlled Studies". In: *Frontiers in Neurology* 13 (2022). ISSN: 1664-2295. URL: https://www.frontiersin.org/article/10.3389/fneur.2022.818559.