Convolutional Recurrent Music Sequence Classification and Generation

Shen (Sean) Chen
MIT ORC/Sloan
seanchen@mit.edu

Yujie Wang
MIT SA+P
yujiew@mit.edu

Abstract

We investigate music sequence classification by leveraging deep learning methods including Recurrent Neural Network (RNN) and one-dimensional convolutional neural network (Conv1D), and classical machine learning classifiers. We find that deep learning methods have a better performance in terms of classification accuracy than classical machine learning classification models, such as logistic regression, SVM, KNN and tree-based models. We also explore the architecture of deep learning models and realize that stacking multiple Conv1D networks has a more significant improvement in capturing the patterns of music sequence compared to stacking multiple RNNs, specifically LSTMs. Besides, we also explore Music generation by LSTMs for the purpose of creating music content that belong to different types of genres so that we can increase the richness of the therapeutic music database. Our code can be found at this link.

1 Introduction

Among the 40 million adults or 18% of the total population in the USA who suffer from anxiety, only 37% receive medical treatment. Anxiety disorders are the most common mental illness in the US. According to the World Health Organization 2019 Report, music is clinically proven to be effective in treating, managing and coping with anxiety disorders. In a society with pharmaceutical products as major treatments, music therapy offers wellness options that are therapeutic, non-invasive, and promote independent coping.

However, current music therapy is observation-based and lacks quantitative tools to quantify, manage, and improve real-time effectiveness. Thus, music therapists lack data to optimize treatments for patients. We see a huge opportunity in the medical application of wearable-enhanced music therapy in psychological treatment and maintenance.

In this project, our focus will be on developing and implementing an algorithm that is able to learn sequential patterns in music to predict whether a song is suitable for therapeutics, as well as establishing a mechanism that enables deeper quantitative understanding about what music features matter most to accurately identify its own kind. We will be selecting and training our classification model by leveraging well-categorized music playlists in Epidemic Sound music database, and then be testing it on self-labeled data of music types.

2 Background

Sequence Classification: There has been work on using the combination of convolutional and recurrent neural networks for sequence classification, especially with implementation in the field of text classification (Siwei Lai, 2015). It has been proven that using convolutional layers, which plays the role of representation learning to capture latent semantic patterns, is going to improve the performance of recurrent models. Such method has been implemented in music classification (Keunwoo Choi, 2016), where CNN is leveraged to extract local music features as a better representation.

However, it was not clear which RNN model was used in the music classification work. Hence, we specifically compared stacks of LSTMs and its bidirectional version in our work.

Sequence Generation: Sequence generation has been widely researched in the fields of NLP and music. The most common methods include Variational AutoEncoders (VAE) (Adam Roberts, 2019) and Generative Adverserial Networks (GAN) (Sang-gil Lee and Yoon, 2018), which turn out to have high quality of sequence reconstruction and generation. These methods, however, also suffer from the issue of computational inefficiency and mode collapsing, especially for GAN.

3 Methodology

3.1 Classifiers

Conv1D and RNN: We propose to use 1-Dimensional Convolutional Neural Network (Conv1D) to learn the latent features of the music and feed the extracted sequence information into Recurrent Neural Networks (RNN) which is good at processing data in the format of time series. We use Long short-term memory (LSTM) Networks and its bidirectional version (BiLSTM) to model the latent representations from Conv1D and eventually output the results with a 5-dimensional softmax activation function after a few feedforward dense layers. We have experimented the idea using three main types of neural network architecture, which include 1) purely LSTM-based, 2) stacks of alternating layers of Conv1Ds and LSTMs, 3) heavily Conv1D based followed by one layer of LSTM. We have also compared the performance using LSTM and BiLSTM. The results are discussed in Section 4.

The following figures have demonstrated the structure of the three neural networks we have implemented with Bidirectional LSTMs.

Layer (type)	Output Shape	Param #
bidirectional_24 (Bidirectio	(None, 800, 512)	528384
dropout_24 (Dropout)	(None, 800, 512)	0
bidirectional_25 (Bidirectio	(None, 800, 256)	656384
dropout_25 (Dropout)	(None, 800, 256)	0
bidirectional_26 (Bidirectio	(None, 128)	164352
dropout_26 (Dropout)	(None, 128)	0
dense_24 (Dense)	(None, 128)	16512
dense_25 (Dense)	(None, 64)	8256
dense_26 (Dense)	(None, 5)	325

Total params: 1,374,213
Trainable params: 1,374,213
Non-trainable params: 0

Figure 1: Bidirectional Neural Network with stacks of pure LSTMs.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 14993, 256)        2304
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 3748, 256)         0
_________________________________________________________________
bidirectional_27 (Bidirectio (None, 3748, 512)         1050624
_________________________________________________________________
dropout_27 (Dropout)         (None, 3748, 512)         0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 3741, 128)         524416
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 935, 128)          0
_________________________________________________________________
bidirectional_28 (Bidirectio (None, 935, 256)          263168
_________________________________________________________________
dropout_28 (Dropout)         (None, 935, 256)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 928, 64)           131136
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 232, 64)           0
_________________________________________________________________
bidirectional_29 (Bidirectio (None, 128)               66048
_________________________________________________________________
dropout_29 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_27 (Dense)             (None, 32)                4128
_________________________________________________________________
dense_28 (Dense)             (None, 16)                528
_________________________________________________________________
dense_29 (Dense)             (None, 5)                 85
=================================================================
Total params: 2,042,437
Trainable params: 2,042,437
Non-trainable params: 0
```

Figure 2: Stacks of alternating layers of Conv1Ds and Bidirectional LSTMs.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_3 (Conv1D)            (None, 14993, 128)        1152
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 3748, 128)         0
_________________________________________________________________
dropout_30 (Dropout)         (None, 3748, 128)         0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 3741, 64)          65600
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 935, 64)           0
_________________________________________________________________
dropout_31 (Dropout)         (None, 935, 64)           0
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 928, 32)           16416
_________________________________________________________________
max_pooling1d_5 (MaxPooling1 (None, 232, 32)           0
_________________________________________________________________
dropout_32 (Dropout)         (None, 232, 32)           0
_________________________________________________________________
bidirectional_30 (Bidirectio (None, 32)                6272
_________________________________________________________________
dropout_33 (Dropout)         (None, 32)                0
_________________________________________________________________
dense_30 (Dense)             (None, 16)                528
_________________________________________________________________
dense_31 (Dense)             (None, 5)                 85
=================================================================
Total params: 90,053
Trainable params: 90,053
Non-trainable params: 0
```

Figure 3: Bidirectional Neural Network with stacks of pure LSTMs.

ML Classifiers: To serve as baselines, we also propose to use classical classification models in machine learning, including Random Forests, Support Vector Machine (SVM), K Nearest Neighbors (KNN), Multivariate Logistic Regression, and Gradient Boosting. We have trained our models using Gridsearch Cross Validation for hyperparameter tuning and have compared our outputs with neural networks, which will be demonstrated in Section 4.2.

3.2 Generation

Within this work, inspired by modelling in sequence classification, we propose to use one layer of Conv1D and one layer of LSTM for music generation. The model fitting procedure is conducted by splitting a whole music sequence into training sequence and testing sequence, so that we can predict the testing notes based on the previous sequences. Some example plots have been illustrated as below. From the generation outputs, it can be seen that Conv1D and LSTM are able to capture the general trend of the music sequence, though it is difficult to capture the exact fluctuality of the sequences.

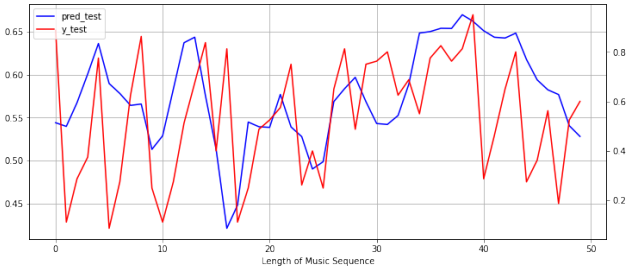

Figure 4: Music Sequence Generated - 1

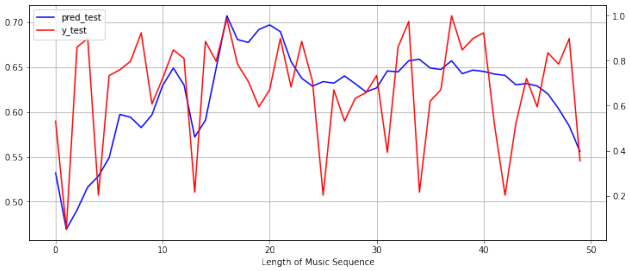

Figure 5: Music Sequence Generated - 2

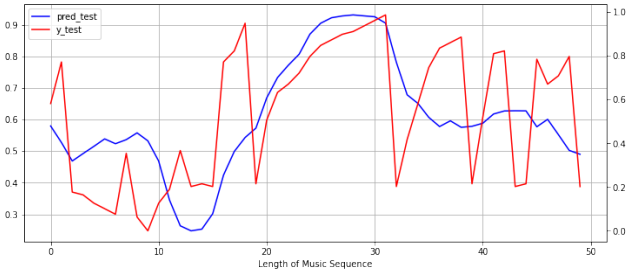

Figure 6: Music Sequence Generated - 3

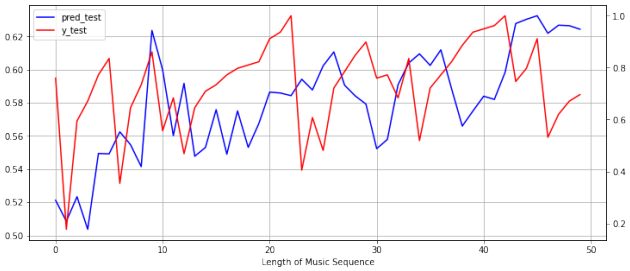

Figure 7: Music Sequence Generated - 4

Future work needs to be done to discover better sequence generation models, such as VAE and SeqGANs, as mentioned in Section 2.

4 Experiments and Results

We have selected a music list of 50 songs in Epidemic Sound music website, with musical feature tags of relatively distinct emotional and contextual characteristics. These musical pieces are also shortlisted for having similar length of sequence — around three minutes, therefore, they can be fairly used in our model training and testing processes with similar number of notes. Our collection of music sequences were labelled by humans with five different categories: Calming, Sports, Mysterious, Nostalgia, and Happy. Due to the fact that humans' preferences for certain types of music could trace back to their childhood environments, cultures and personal experiences (Thoma MV, 2013), it is essential that our model can capture the pattern for the intrinsic bias in music for an individual. That is, to accurately identify the music type a human prefers to listen to in a certain scenario, e.g. when they want to calm down or feel like doing sports.

4.1 Experiments

We have converted the 50 songs into MiDi files and have used MiDi notes as our sequence data. Due to the limitation in the total number of songs we have, we augmented the richness of the dataset by splitting each song into segments of length 800-note series, with the label for each music segment being identical to the original song it belongs to. By doing so, we end up obtaining a collection of 835 music segments with 800 time steps for each segment. We have also converted the labels by one-hot encoding. We then implement the classifiers mentioned in Section 3 onto the enriched dataset. The training and testing ratio is 0.8.

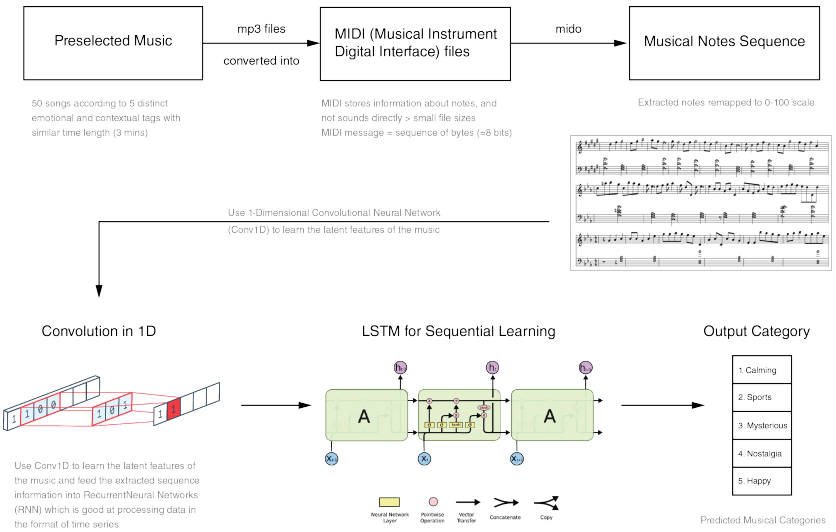

Figure 8: Music Sequence Classification Pipeline

4.2 Results

The experiment results for the classifiers are the following:

	NN1	NN2	NN3	Bi-NN1	Bi-NN2	Bi-NN3
Train	0.3593	0.3593	**0.6991**	0.3593	**0.6640**	**0.6976**
Test	0.3772	0.3772	**0.7126**	0.3772	**0.7305**	**0.6674**

Table 1: NN1, NN2 and NN3 represent the following three neural networks mentioned in Section 3: 1) purely LSTM-based, 2) stacks of alternating layers of Conv1Ds and LSTMs, 3) heavily Conv1D based followed by one layer of LSTM. Bi-NN models are using Bidirectional LSTMs.

	RF	SVM	KNN	Logit	GradBoost
Train	0.6302	0.6362	0.7515	0.5300	0.9
Test	**0.5210**	0.5030	0.2635	0.5090	**0.5689**

Table 2: Other classifier were all trained using GridSearchCV to find the best hyperparameters.

From Table 1, it can be seen that models with convolutional networks generally have a better performance in terms of class prediction accuracy. In particular, models heavily using Conv1D layers with one LSTM layer tend to perform generally well, with or without the LSTMs being bidirectional. The best model for the predictive task is Bi-NN2, which uses alternating layers of Conv1D and Bidirectional LSTM. The out-of-sample testing accuracy has peaked at 0.7305, which is much higher than its non-neural network counterparts.

The results have convinced us that the convolutional recurrent neural network framework is good at capturing the patterns in music sequence data, which contains information about human's bias and preferences for different music types.

5 Conclusion

In this paper, we have specifically focused on using music notes data for sequence classification and generation and end up having promising results as proof of concept. We have demonstrated that sequential patterns in music can be reasonably well captured by stacks of convolutional and recurrent layers of neural networks. Also, LSTMs can be used for music sequence generation, though we cannot guarantee the granularity of the generated music to be highly consistent with the original ones.

In the future, we would like to incorporate more musical features, such as tempo, key, beat, and tatum, to enrich the dimensions of feature space. In addition, we would also conduct lyrical and sentimental analysis by leveraging NLP techniques in order to achieve more comprehensive music analytical results. For music generation, VAE and Sequence GAN are great methods to explore, using the LSTM results in this work as baseline to compare with. Also, since music by nature can be classified into multiple categories, which is not the approach we take in this paper, ideally we will be able to achieve more granular results by predicting distributional outputs for music classification, rather than deterministic ones.

References

C. R. C. H. D. E. Adam Roberts, Jesse Engel. A hierarchical latent vector model for learning long-term structure in music, 2019. 2

M. S. K. C. Keunwoo Choi, Gyorgy Fazekas. Convolutional recurrent neural networks for music classification, 2016. 1

S. M. Sang-gil Lee, Uiwon Hwang and S. Yoon. Polyphonic music generation with sequence generative adversarial networks, 2018. 2

K. L. J. Z. Siwei Lai, Liheng Xu. Recurrent convolutional neural networks for text classification, 2015. 1

B. R. F. L. E. U. N. U. Thoma MV, La Marca R. The effect of music on the human stress response, 2013. 5