JAZZ BASS ACCOMPANIMENT AND SOLO GENERATION

Machine Learning Final Project

Yu-Ting, Tsai   
Department of Computer Science  
National Tsing Hua UniversityHsinchu City, Taiwan(ROC)  
A38050787@gmail.com

Yi-Chieh, Chiu  
Department of Computer Science  
National Tsing Hua UniversityHsinchu City, Taiwan(ROC)  
shes202@gmail.com

Kevin Richardson  
Department of Computer Science  
National Tsing Hua UniversityHsinchu City, Taiwan(ROC)  
k.richardsonhalim2002@gmail.com

*Abstract*—Machine Learning has been very popular recently. However, instead of the well-known application, such as picture recognition, and category prediction, we try to implement a model that can generate bass automatically for jazz music. Even though it may not seem to be the most lucrative application of Machine Learning, compared to Some of the most prominent fields of Natural Language Processing (NLP,) Computer Vision, and Quantitative trading, we still see a bright future in the field of introducing machine learning to music. In this paper, we use bidirectional RNN to deal with this task. We also put some effort into data preprocessing and experiments.

Keywords—jazz, bass, bidirectional, Recursive Neuron Network (RNN)

# Introduction

Jazz has become a very popular music genre after its birth in the late 19th and early 20th centuries, in the African-American communities of New Orleans, Louisiana. Jazz is characterized by swing and blue notes, complex chords, call-and-response vocals, polyrhythms, and improvisation. Jazz music is known for its intricate and improvisational bass lines, which add depth and complexity to the music. However, creating original and fitting bass lines for jazz can be challenging, even for experienced musicians.

The reason we choose jazz music is that it isn’t too complex to understand, and its musical structure is simpler compared to classic music. Moreover, the chords of jazz are analyzable, by decomposing every note, it’s not too difficult to find the certain pattern that the composer wants to construct.

There are a lot of jazz music resources available on the internet, and thanks to the existence of the midi (Musical Instrument Digital Interface) file, we can translate music to figures based on several models that are built for midi files.

In this project, we aim to use machine learning techniques to automatically generate bass lines for jazz music. By training a model on a dataset of existing jazz bass performances, we hope to create a tool that can generate new and original bass lines that fit the harmony and structure of jazz music. For instance, Let a piece of jazz music be A + B, B = music of double bass, and A = music played by other instruments (piano, guitar, drums.... etc). We want the model to learn the relationship between A and B. If we input A, the output will be B', We want B' to be as similar to B as possible. We then combine B' and A to get the complete song. In our project, we tested several methodologies to achieve this goal. In the following sections, we’ll be introducing Recursive Neuron Network(RNN), and some augmentation on the data set and model.

# Methods

## Input format

Our input format contains the following elements:

* chromagram 12x3
* speed (integer)
* number of instruments (integer)
* time signature (two integers)
* beat position (int)
* bar position (int)

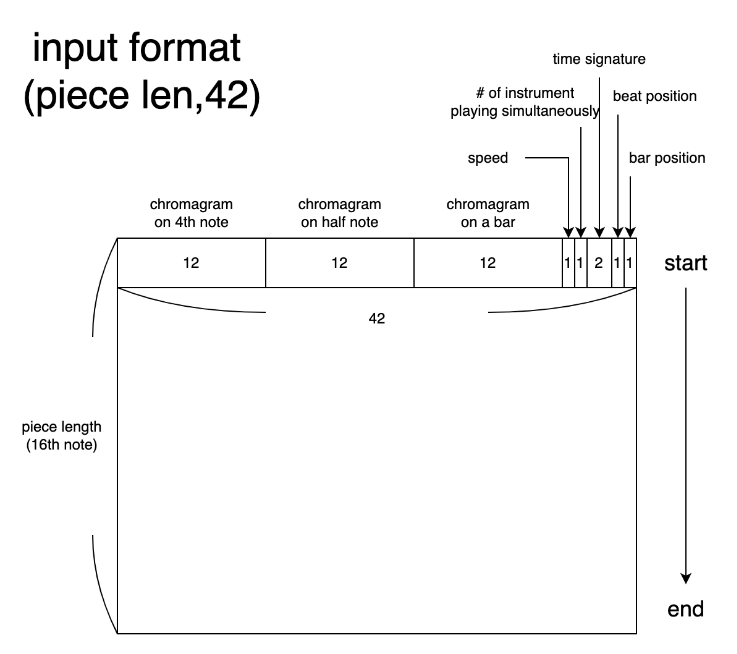


Fig. 1. Input format figure

From Fig. 1, we can see that each row contains the elements which represent the information of a song. Chromagram can be seen as a feature that gives a unique index to every pitch. With the knowledge that a set of notes {C, C♯, D, D♯, E, F, F♯, G, G♯, A, A♯, B} can be viewed as twelve-pitch spelling attributes, we can assign chroma C with index 0, and chroma C# with index 1, for example. We record the chromogram on the 4th note, the half note, and the bar because this is usually the pattern and basic unit of a chord change. By recording these notes, we can catch the trend of the music.

For the rest of the features, we wish the model can make use of them and improve the training result, since they are all important features that represent a song, such as speed, time signature, beat, and bar position. For beat and bar position, beat position records the location of the music this row element is representing. For example, for a time signature which is Four-Four Time, since each row element is recorded on the frequency of 16th notes, then the 16th row element will have beat position 15 (starting from 0), and the 17th row element will have beat position 0 since it is the start of another bar. For bar position, it’s just which bar the row element resides in the music, so for the previous example, the 15th and 16th row elements both have bar position 0, while the 17th row element has bar position 1. Beat position and bar position can help the model catch the trend of the music as well. For example, at the beginning of most jazz music, the music is smooth, and it will become exciting or exaggerated in the middle since this is usually the theme of the music. Then, by recording the bar position, the model can try to learn the relationship between notes and bar position. Similarly, the beat position has the same feature, but with a smaller scale.

Each column represents a certain period of music, the sampling period is 16th note-long. For the leftmost chromogram, it will change every four rows, since one 4th note is as long as 4 16th notes. Similarly, for the chromogram on half note, it will change every eight rows, since one half note is as long as 8 16th notes. Of course, we can sample the music with the unit 32nd note-long, but it’s not practical. From the musical perspective, the 16th note is more often to be viewed as the basic unit rather than the 32nd note, and if we adopt the latter period, the input and output size of our data will be extremely huge.

Another concern is why we only need 12 pitches to represent a song. The reason is that humans perceive two musical pitches as similar if they differ by octave. Hence, From the view of chords, they are not any different. Since the number of pitches of a song may be huge (piano has 88 keys that represent 88 pitches!), by reducing it to only 12 pitches, we can largely reduce the size of the input format.

## Output format

Since our task is to predict the bass part of jazz music, we simply use one-hot encoding to represent the pitches that may be played by the bass as shown below.

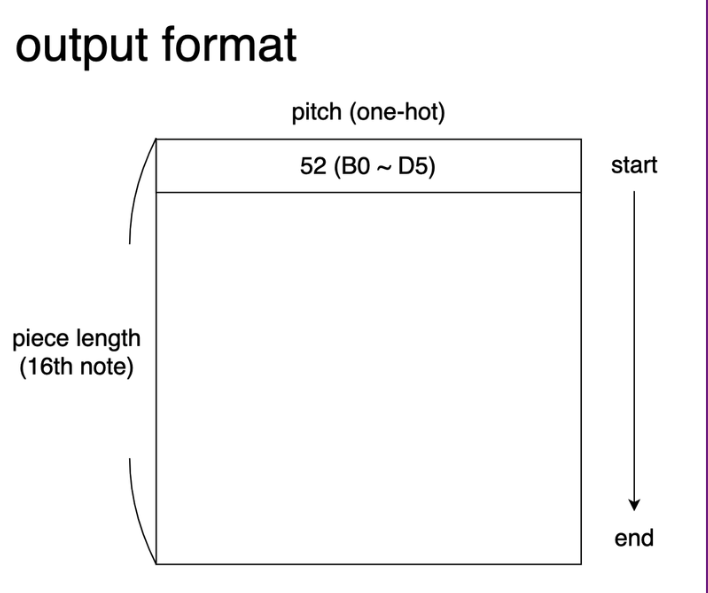


Fig. 2. Output format figure

## Data Set

As mentioned before, we use the midi file as our raw input data. This is because the midi (Musical Instrument Digital Interface) file is a well-defined technical standard that can illustrate and record music perfectly. Not to mention there are a lot of resources and python packages that can support our implementation. We tried three data sets listed below, and I’ll explain their advantages and disadvantages.

* minor9 jazz standards + Jazz ML ready MIDI

These data sets are all midi files, and since there’s no position for the rest note, we use the position of D5(the highest note) instead.

* JAAH Dataset

Midi file is great, but we think that it will be greater if we have more information about a song, so we turn to the JAAH dataset. JAAH stands for Jazz Audio-Aligned Harmony Dataset, it is a special data structure that contains audio, and a lot of annotations including chord beat structures. The only problem with this dataset is that it’s hard to separate the bass part from the original music compared to midi files. We need the bass part seriously since we need it to be the ground truth of our training data. Our solution is to take advantage of another machine learning-based model BassUNet, which uses U-Net Fully Convolutional Networks to separate the bass from the original music. Unfortunately, the result of this model is not good enough to be the ground truth, we ended up terminating the use of this dataset.

* iReal Pro dataset

After the failure of the JAAH dataset, we return to the midi files again. However, since the result trained by the original midi files was poor, we tried another midi dataset which is the iReal Pro dataset. All the midi files of this dataset are computer generated, their purpose is for music practicing, so the structure is simpler compared to our original midi files. We have to emphasize that even though the dataset is computer generated, it has nothing to do with machine learning, it just follows some certain pattern to generate a large amount of canned music.

* midikar dataset

For validation purposes.

After the completion of our model, we perform transfer learning on the model with the iReal Pro dataset. The result is solid, so we trained the model with other midi files, and the model still works fine. Thus, for the final dataset, we combine minor9 jazz standards, Jazz ML ready MIDI, and iReal Pro. We also transpose these midi files to 12 keys to enlarge our dataset.

## Data Augmentation

Jay, please fill in this part.

## Data Preprocessing

Jay, please fill in this part.

## Model

We use a bidirectional Recursive Neuron Network(RNN) to implement our model. The detail of the model is as follows.

一張含有 文字 的圖片

自動產生的描述

Fig. 3. Model description

# Results

Our model reaches an accuracy of 0.8873 with 500 epochs and 3 hours of training. The example predicted result can be found here in the form of an mp3. Note that the predicted result generated by the model is only the bass part, we then combine the bass with other instruments to form a complete song. The combination is done by ourselves. (maybe a further explanation is required by Jay.) However, there are some interesting things: some unnatural short notes are jumping up and down, and there are many long notes because we cannot make pitch repetition in the output.

Since there is still time before the deadline of this project, we think it’s a good idea to raise the difficulty of our implementation. Recall that we use chromograms to represent jazz music. Jazz music is nothing special from other music, they are all composed of chores. A chore can be viewed as a structure, and the most critical component of the structure is called the root. In music theory, the concept of root is the idea that a chord can be represented and named by one of its notes. It is linked to harmonic thinking—the idea that vertical aggregates of notes can form a single unit, a chord [1]. We suspect that the root may play the role of a hint for the model to predict the bass part, so we want to explore what will happen if we remove the root. Will the model still perform perfectly? If not, what method can be done to enhance the model? This will be covered in the Experiment section.

# Experiment

After removing the root of the chores from all the chromograms, we have done some experiments listed below.

## Experiment 01

We use Adam (adaptive moment estimation) as an optimization algorithm for our bidirectional RNN model. The result is bad, and we discover there might be overfitting and gradient exploding issues. We figured out two ways to improve this model. First, enhance the quality of our input dataset. Second, improve the capabilities of our LSTM model. We decided to focus on the second improvement, and the corresponding experiments are as follows.

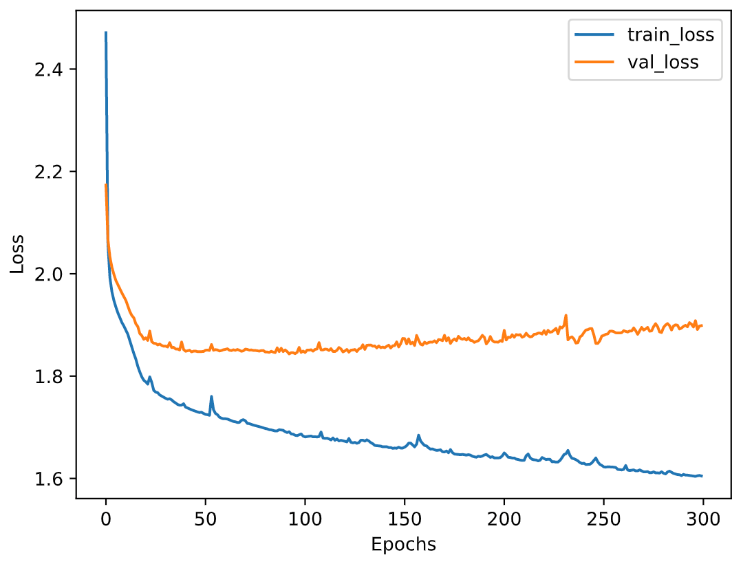


Fig. 4. The lost diagram of experiment 01

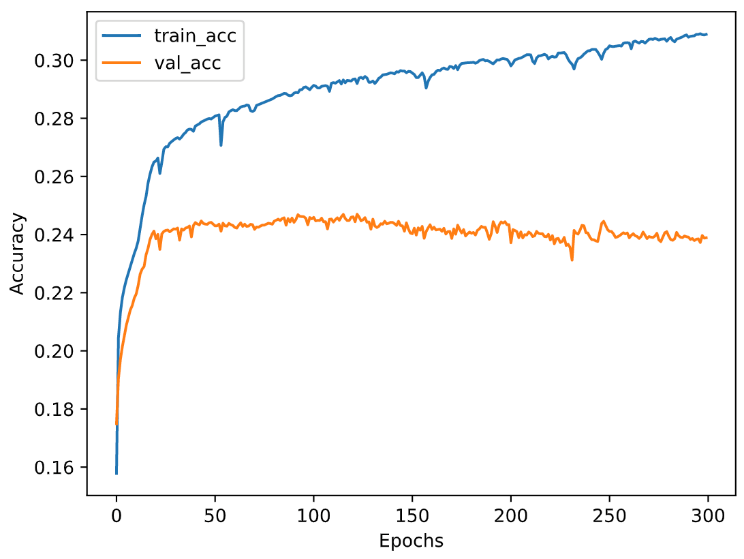


Fig. 5. The accuracy diagram of experiment 01

## Experiment 02

In this experiment, there are two deep layers, and one wide layer, the units remain the same. The improvement and corresponding reasons are as follows.

* Add a dense layer to input

The original chromogram lacks the chord root and is not complete, so it would be great if the input dense layer can play a complementary role

* Add a dense layer to output

We hope LSTM can focus more on memorizing information

* Regularization

The previous experiment has a serious overfitting issue, so l2 was added to the dense layers, and the LSTM kernel also has it, but it is relatively small (0.0001).

* Gradient clipping

After running the previous experiment, the problem of gradient explosion is too serious to ignore.

Note that all dense layers have batch normalization. The result of the above experiment shows that regularization did do its job, however, gradient explosion is getting worse.

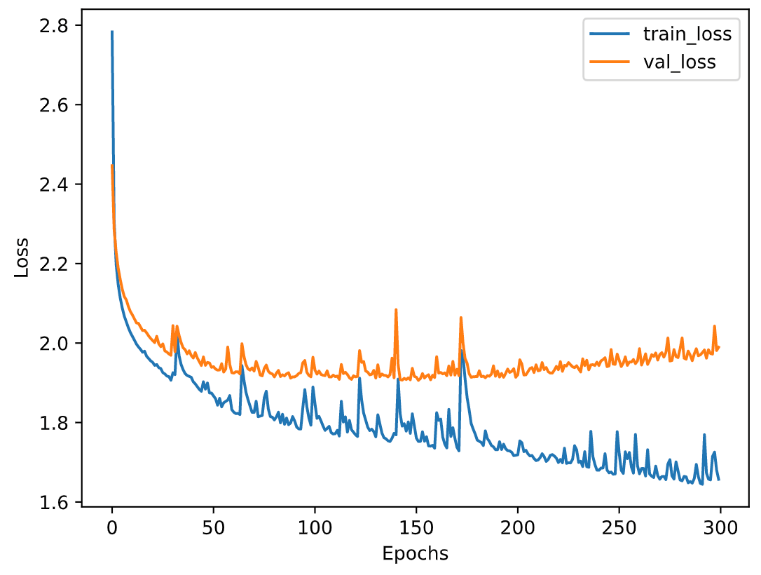


Fig. 6. The lost diagram of experiment 02

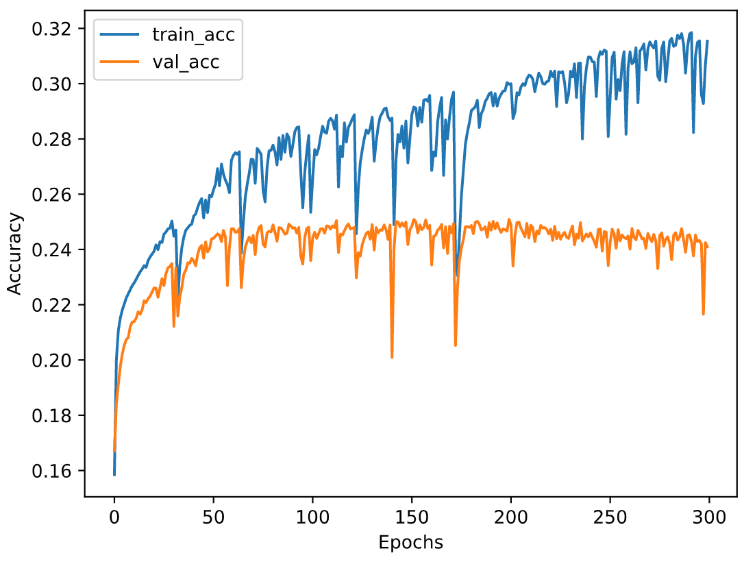


Fig. 7. The accuracy diagram of experiment 02

## Experiment 03, Experiment 04

In experiment 03, we tried to add regularization to adam, and in experiment 04, we added a thicker FeedForward before and after LSTM, and add regularization. Both experiments ended up a failure.

## Experiment 05

In this experiment, we simplify experiment 04, leaving only one layer before LSTM, and adding a bunch of regularization, the performance is ordinary.

## Experiment 06 - Experiment 08

In this experiment, we found out that currently, it seems that the smaller the piece\_length, the better, and more LSTM\_units will be better, but it will also cost a relatively high time cost.

## Conclusion

This will be covered in the Discussion/Conclusion section below.

# Discussion/Conclusion

In this paper, we introduced a bidirectional Recursive Neuron Network(RNN) to build a model which can generate jazz bass accompaniment and solo automatically. The model performs well with an accuracy of 0.8873, and we can conclude that we have achieved our goal. For further study in the Experiment section, we can conclude that it’s extremely hard for a model to learn the relationship between the music of bass and other instruments without the root of the chores. This makes sense since even for humans this is also mission impossible. Given a pitch, it has no absolute answer for a person to decide which other pitch should be combined with this given pitch without the root of the chore. The reason is that there may be multiple answers. This can be introduced by another concept, “ consonance and dissonance”. In music, consonance and dissonance are categorizations of simultaneous or successful sounds. Within the Western tradition, some listeners associate consonance with sweetness, pleasure, and acceptability, and dissonance with harshness, unpleasantness, or unacceptability, although roads led to this on familiarity and musical expertise [2]. Hence, for anyone to guess the correspondent pitch, the goal for him or her is to choose another pitch that would be consonance with the given pitch. Unfortunately, there may be multiple choices at the same time since there is more than one consonance chord, and the answer to this question only depends on the preference of the composer when composing the song. This is exactly what we wish the model to do, a task that is very difficult even for human beings.

At the end of the day, we believe that machine learning is suitable for tasks that can be done by a human, but it’s too costly and inefficient. A human can recognize whether the food on the plate is pizza or not, but if there are 1000 plates, we train a model to let the computer do it instead of doing it by ourselves. Our experiment has shown that some tasks that are difficult for a human to accomplish, may be hard for a machine as well.

However, in conclusion, the results of this project demonstrate the potential for using machine learning to generate high-quality bass lines for jazz music. Our model is beneficial to humans, especially to musicians and the music industry. The model can assist musicians with tedious work, so they can put their focus on high-value tasks. It can Evaluate/score a piece of music before any further production, and it can also be viewed as an Inspirational and innovative tool for the musician. Overall, this project provides a proof of concept for using machine learning to generate jazz music and opens the door to future research in this area.

# Future work

## Some Direction of Better Input/output Format

* Add drums information into the input, since drums play an important role in jazz music.
* Add velocity as output since now the velocity is set by ourselves.
* add onset/offset timing correction into the output
* Incorporating more information about the context of the music, such as the melody or the style of the song, to generate bass lines that are more fitting and coherent with the music.

## A Newly-designed Model

* Train encoder-decoder RNN with attention since it’s a great idea to treat it as a kind of sequence to sequence translation problem
* A transformer is suitable for this project as well.

# Author Contribution Statements

* Yu-Ting, Tsai (33.33%)

Data collecting, model training, programming of the model, final presentation, and writing the final report.

* Yi-Chieh, Chiu (33.33%)

Data collecting, data preprocessing, model training, programming of the model, and writing the final report.

* Kevin Richardson (33.33%)

Model training, and programming of the model.

# SOURCE CODE

##### References

1. Root (Chord). (n.d.). Wikipedia. https://en.wikipedia.org/wiki/Root\_(chord)
2. Lahdelma, Imre, and Tuomas Eerola (2020). ["Cultural Familiarity and Musical Expertise Impact the Pleasantness of Consonance/Dissonance but Not Its Perceived Tension."](https://www.nature.com/articles/s41598-020-65615-8) Scientific Reports 10, no. 8693 (26 May)