Deep Learning Guitar Player

Thomas Laure

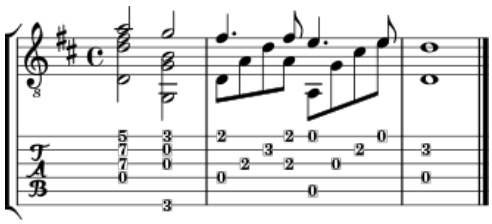
November, 2018

Resume :

The aim of this work is to develop a deep learning model to write guitar music. The structure employed here may be interesting as it seems different from the examples dealing with the same objective online. For this problem we will use a LSTM model as it already prove its value for this kind of exercises using TensorFlow & Keras.

1. Introduction

The aim of this work is to generate guitar tracks using tablatures as an input. Tablature is a common format for fretted string instrument and is used by almost all guitar learners. While standard notation represents the rhythm, duration and pitch of each note a tablature is instead operationally based. It indicate where and when a finger should be placed to generate a note, so pitch is denoted implicitly rather than explicitly.



Standard notation on top with its tablature translation

The underlying idea is to train the model just like an aspiring guitar player would do.

2. Data

For this work we will feed the model with all tablatures from an artist, as an example we will use the Australian hard-rock legends AC/DC. All the tablatures are downloaded using some web scraping technics and converted to a format which can be translated as an input for the neural network, using the algorithm :

```
For tablature in all_tablatures:
◦◦time_temp=0
◦◦for i_time in tablature.length
◦◦◦◦Note=[]
◦◦◦◦For string in all_strings
◦◦◦◦◦◦If is_note(string,i_time)==True
◦◦◦◦◦◦◦◦Note.append(convert_format(string,i_time))
◦◦If Note=[]
◦◦◦◦time_temp+=1
◦◦Else
◦◦◦◦record(note)
◦◦◦◦record(time_interval)
```

As described in the pseudo-code we take two informations : a variable « time » which is the length of the tablature between two notes, and a variable « note » which can be a single or a group of strings played.

3. Models

To understand LTSM networks the article linked here from Colah's blog is very well written and accessible. For this work we use two models, one to predict the note, one for the time that preceded the note. The two models use the same input :

$$\begin{pmatrix} note_{t-seq} & time_{t-seq} \\ note_{t-seq+1} & time_{t-seq+1} \\ \cdots & \\ note_{t} & time_{t} \end{pmatrix}$$

Where seq is chosen by the user and represent the number of previous data used to predict the next one.

Each model is composed of several LTSM and fully connected layers. The loss function used for both models is categorical crossentropy.

4. Predictions

After fine-tuning and training both models the weights are saved so they can be reused. To start generating guitar, a part of the data is randomly selected to start with. In a loop, both models predicts at each step the time to wait and the note played, the input data for the next step is then shifted to include the new note and time.

5. Export and conclusion

Using the music21 package on python the guitar tracks can be exported to a midi format the results are rather good, some examples can be found here.