Rubato

Music Generation : Make Music with MIDI

소준섭
모두의연구소 Rubato Lab.

소개

- 소준섭
- 모두의 연구소 Rubato Lab.
- 고양이

Making Music ?


Eko


KRK
Roland


Pioneer Dj
SEARCH
BROWSE
TAG LIST
LINK INFO
INFO
UTILITY
MENU
01:27.368
0.00
150

Rubato

우리가 듣는 음원, 어떻게 제작 될까?

음악을 만드는 과정

Rubato

- 보편적으로 음악을 만드는 과정을 알아봅시다.

음악을 만드는 과정

Rubato

- 보편적으로 음악을 만드는 과정을 알아봅시다.
- 멜로디 / 코드 등을 떠올린다.

음악을 만드는 과정

Rubato

- 보편적으로 음악을 만드는 과정을 알아봅시다.
- 멜로디 / 코드 등을 떠올린다.
- 떠올린 멜로디를 구체화 / 기록을 한다.

음악을 만드는 과정

Rubato

- 보편적으로 음악을 만드는 과정을 알아봅시다.
- 멜로디 / 코드 등을 떠올린다.
- 떠올린 멜로디를 구체화 / 기록을 한다.
- 악보를 작성한다.

음악을 만드는 과정

Rubato

- 보편적으로 음악을 만드는 과정을 알아봅시다.
- 멜로디 / 코드 등을 떠올린다.
- 떠올린 멜로디를 구체화 / 기록을 한다.
- 악보를 작성한다.
- 컴퓨터로 할 순 없을까?

Rubato

How to record my music?

DAW

Rubato

- Digital Audio Workstation 을 사용한다.

DAW

Rubato

- Digital Audio Workstation 을 사용한다.
- 우리가 IDE (Integrated Development Environment)를 사용하는 것처럼
  곡을 만들때 사용하는 툴이다.

DAW

Rubato

CUBASE PRO 10

Ableton

AVID

Au

Studio One

BITWIG

FL STUDIO

LMMS

DAW

Rubato

- Digital Audio Workstation 을 사용한다.
- 우리가 IDE (Integrated Development Environment)를 사용하는 것처럼
  곡을 만들때 사용하는 툴이다.
- 그럼 DAW의 역할은 무엇일까?

DAW

Rubato

- Digital Audio Workstation 을 사용한다.
- 우리가 IDE (Integrated Development Environment)를 사용하는 것처럼
  곡을 만들때 사용하는 툴이다.
- 그럼 DAW의 역할은 무엇일까?
- Audio (Wave 파일의 편집) / MIDI data를 이용해 VST 사용

Rubato

What does VST stand for?

VST

Rubato

- Virtual Studio Technology (VST)

VST

Rubato

NATIVE INSTRUMENTS

iZotope

Reason

SPECTRASONICS

SERUM

WAVES

VST

- Virtual Studio Technology (VST)
- 가상 악기 (VSTi), 가상 이펙터 (VSTfx) 를 통합해서 부르는 명칭


GRAVITY
STUDIOS


CLA CLASSIC COMPRESSORS
A: Rock Dry Room
A→B
Load
Save
ATTACK
RATIO
METER
GR
IN
OUT
INPUT
OUTPUT
RELEASE
ALL
PluginChild 670
A: Hogarth Mix Slam
api
VU
METER
COMPRESSOR
THRESH
ATTACK
RATIO
RELEASE


WAVES HYBRID LINE
A: Pumping Drums
IN
GR
OUT
LIMITER
ANALOG
RATIO
OUTPUT
PUNCH
CLA CLASSIC COMPRESSORS
A: Start Me Up
WAVES AUDIO LTD.
COMPRESS
LIMIT
GAIN
PEAK REDUCTION
VU DISPLAY
CLA-2A
ANALOG
MIPREG
A: Overheads
ATTACK
RATIO
METER
INPUT
OUTPUT
RELEASE
CLA-76
REVISION
BLUEY
BLACKY
COMP OFF


A: OurTinLev
Master
Release
Bypass
Output
A: LAYERS VINTAGE MASTERING SUITE
THRESHOLD
TIME CONSTANT
OUTPUT GAIN
PluginChild 660
A: Pumper II
Manual
Electro
Warm
Attack
18.0
Release
72.4
Thresh
-22.0
Ratio
1.91
-2.3
-1.1
-10.1
Input
Attenuation
Renaissance

VST

Rubato

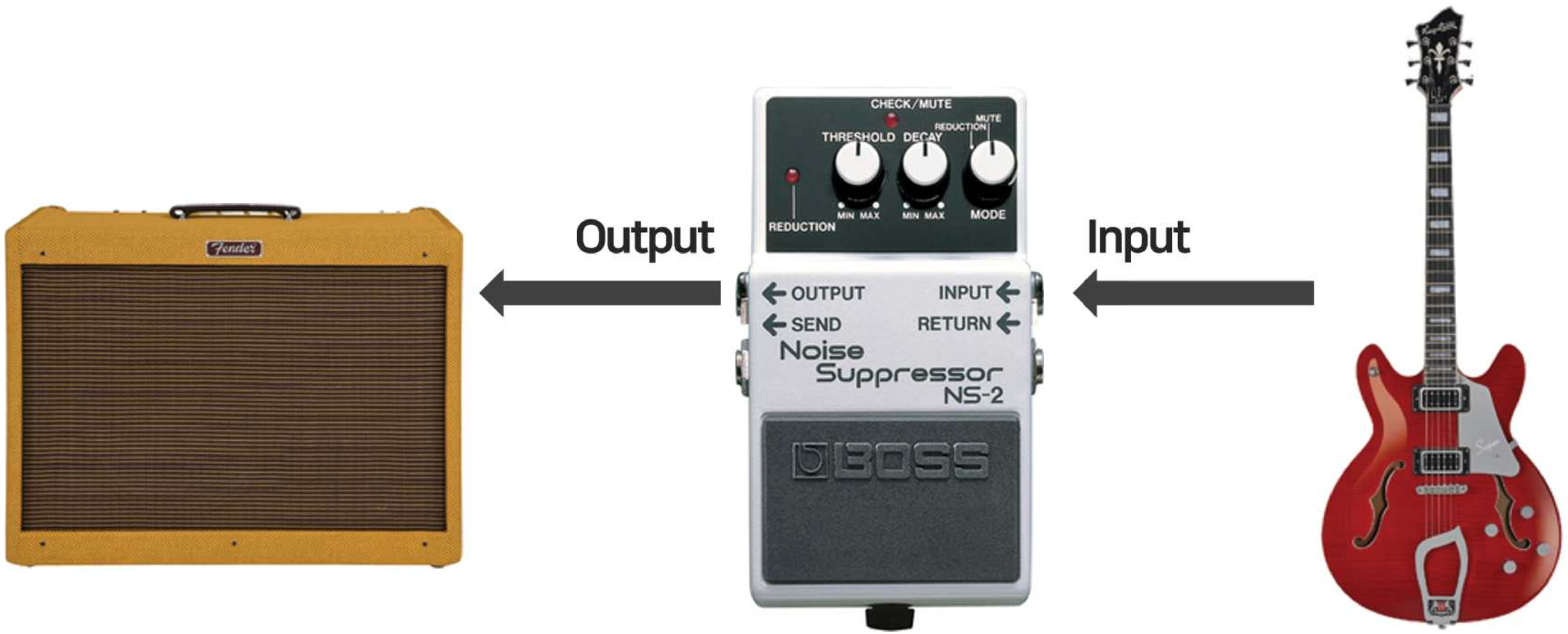
- Virtual Studio Technology (VST)
- 가상 악기 (VSTi), 가상 이펙터 (VSTfx) 를 통합해서 부르는 명칭
- VSTi는 MIDI 가 가진 data를 기반으로 소리(wave)를 만들어준다.
- VSTfx는 만들어지거나 주어진 wave data에 효과를 주는 역할을 한다.

VST

Rubato

- Virtual Studio Technology (VST)
- 가상 악기 (VSTi), 가상 이펙터 (VSTfx) 를 통합해서 부르는 명칭
- VSTi는 MIDI 가 가진 data를 기반으로 소리(wave)를 만들어준다.
- VSTfx는 만들어지거나 주어진 wave data에 효과를 주는 역할을 한다.
- AU / AAX 는 일단 여기에선 Pass!

VST

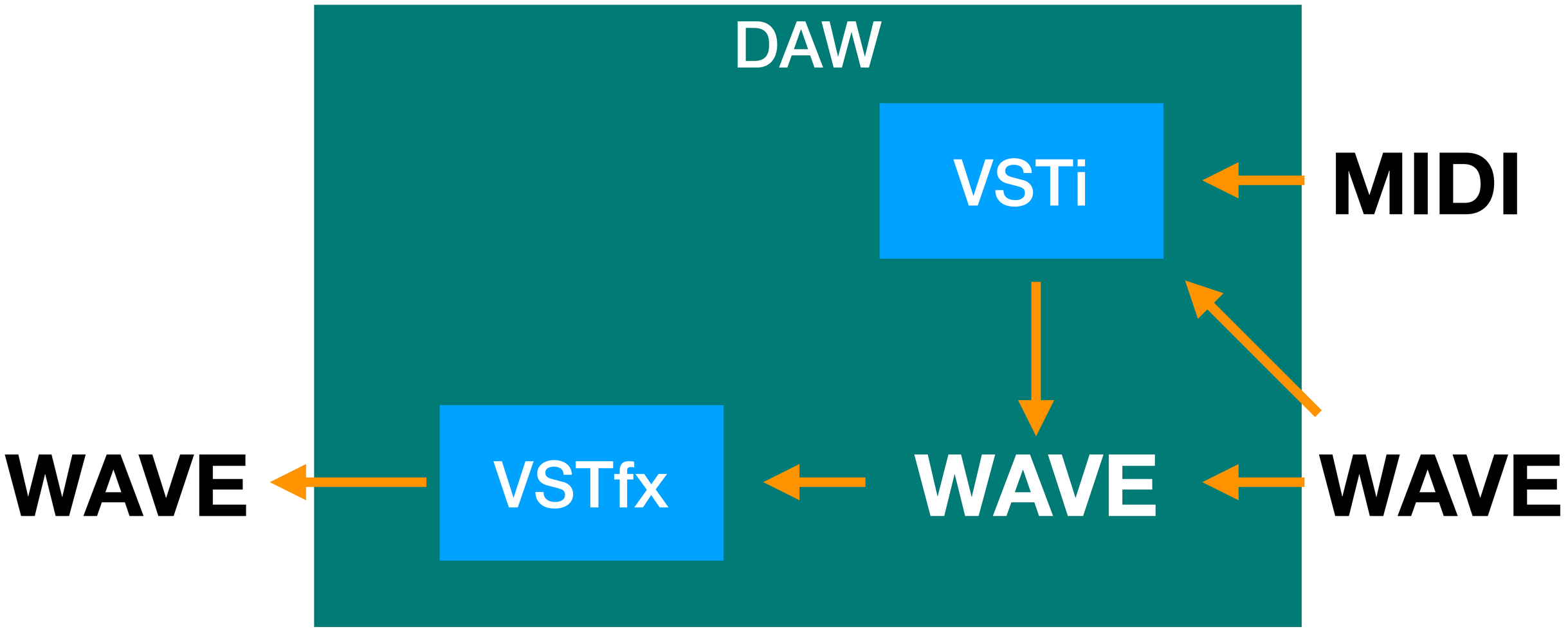


https://blog.naver.com/PostView.nhn?blogId=yogoho210&logNo=221401793033&parentCategoryNo=&categoryNo=6&viewDate=&isShowPopularPosts=false&from=postView

VST

https://blog.naver.com/PostView.nhn?blogId=yogoho210&logNo=221401793033&parentCategoryNo=&categoryNo=6&viewDate=&isShowPopularPosts=false&from=postView

VST

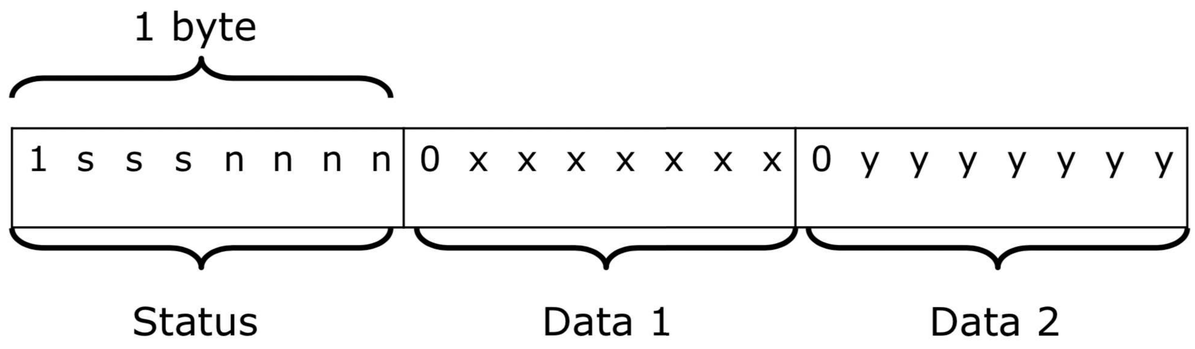


Rubato

MIDI와 WAV 구분

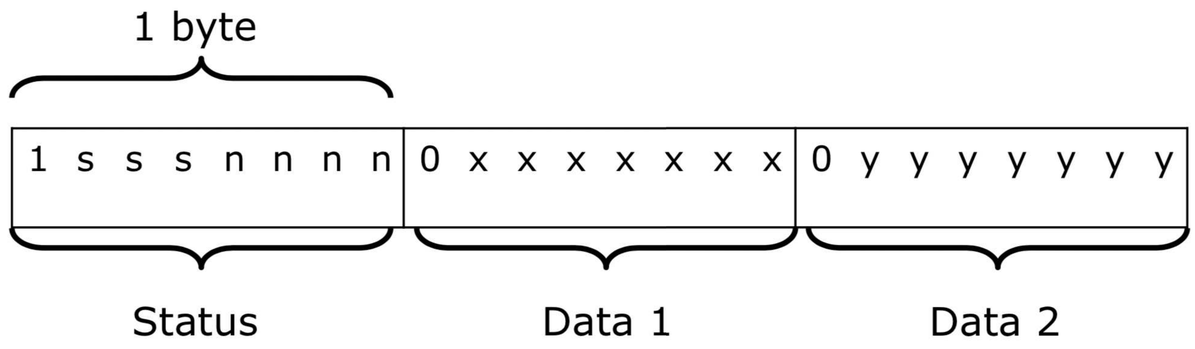
MIDI

- Musical Instrument Digital Interface 의 약자



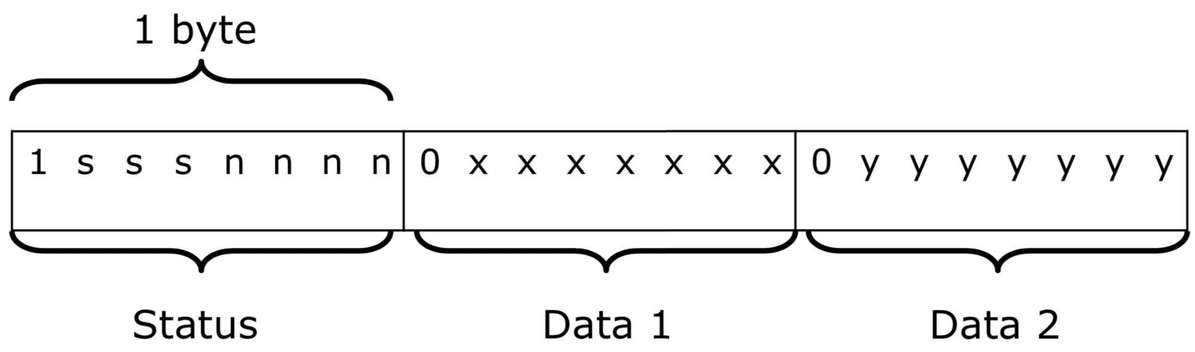
MIDI

- Musical Instrument Digital Interface 의 약자
- 전자 악기끼리 디지털 신호를 주고 받기 위해 각 신호를 규칙화한 일종의 규약



MIDI

- Musical Instrument Digital Interface 의 약자
- 전자 악기끼리 디지털 신호를 주고 받기 위해 각 신호를 규칙화한 일종의 규약
- MIDI 패킷은 Status, Data1, Data2 등 8 bytes 세 개 패킷으로 나눠져있다.
- 패킷 데이터 따라 동작하는 테이블이 정해져 있다.



MIDI Reference Tables

Our reference tables are the quick and easy way to look up the meaning of a particular MIDI message number, find a Manufacturer ID number, find international standards that incorporate MIDI, and more.

Summary of MIDI Messages

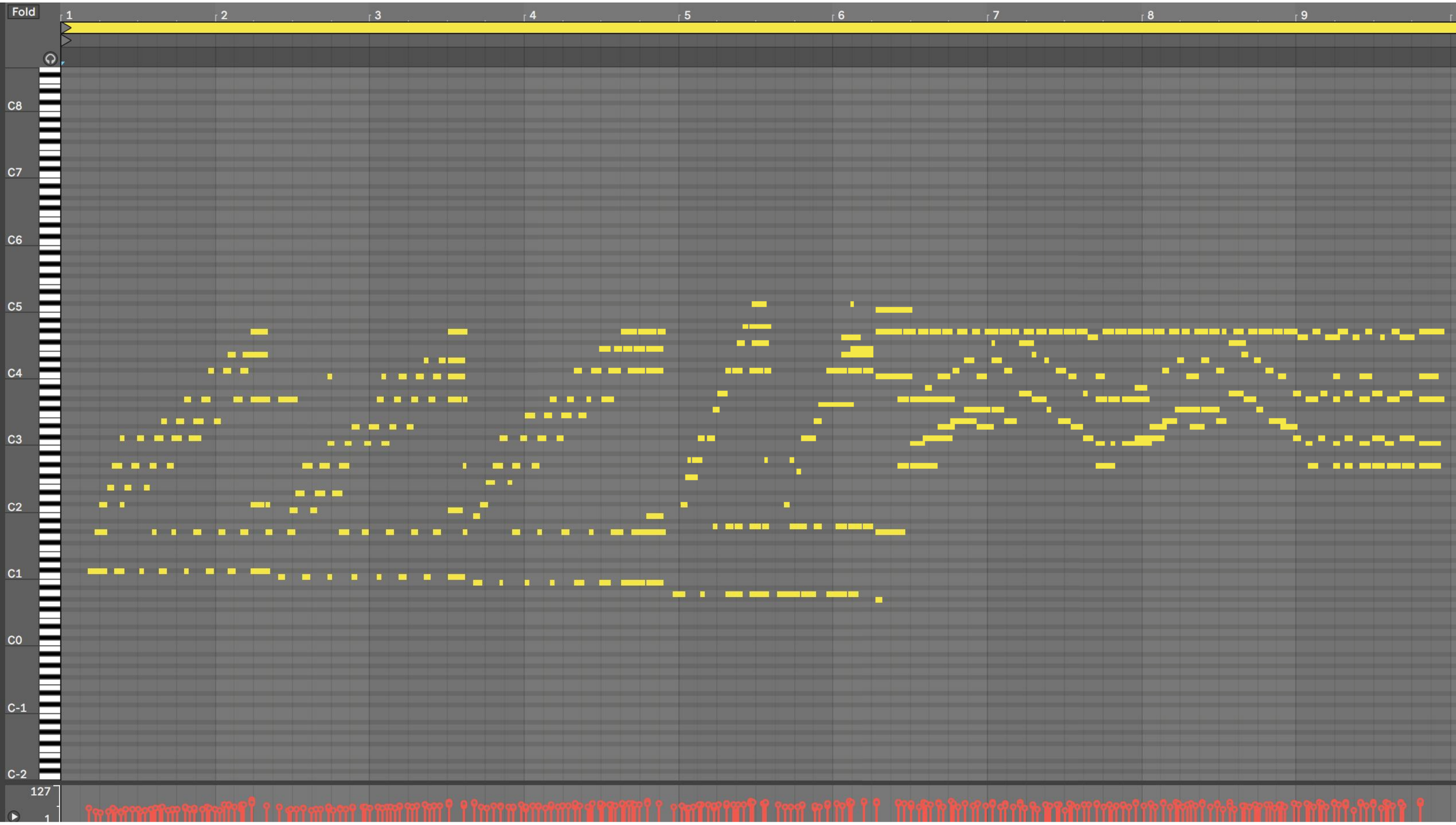
The following table lists many of the major MIDI messages in numerical (binary) order.

SHOW DETAILS

Expanded Messages List (Status Bytes)

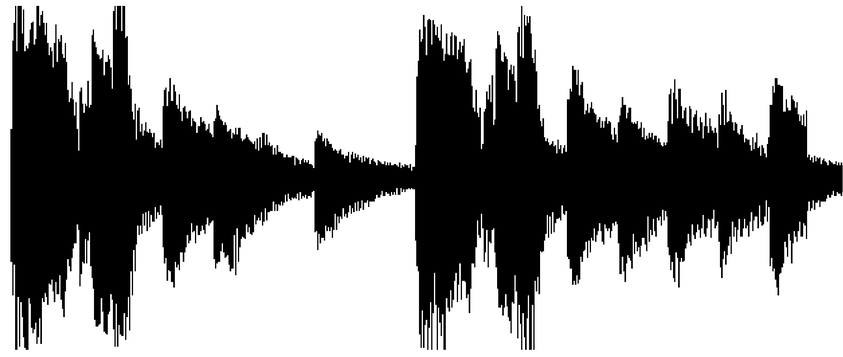
The following table lists all of the Status Bytes in binary numerical order.

SHOW DETAILS


Fold
1
2
3
4
5
6
7
8
9
C8
C7
C6
C5
C4
C3
C2
C1
C0
C-1
C-2
127
1

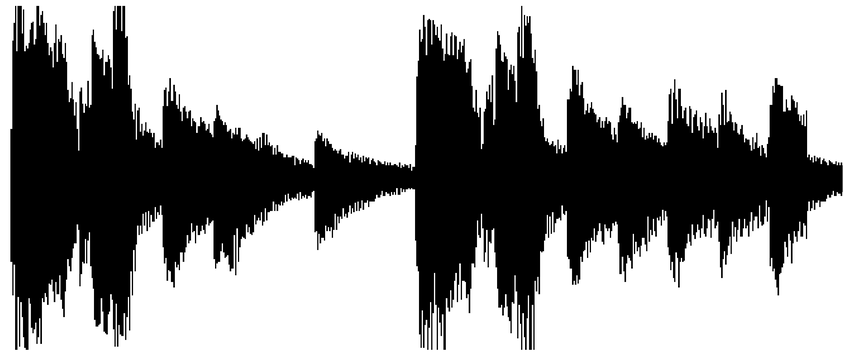
WAVE

- wav 또는 wave는 웨이브폼 오디오 포맷(waveform audio format)



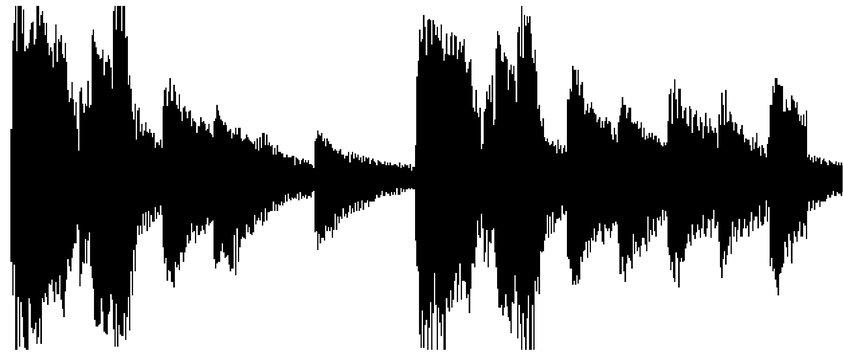
WAVE

- wav 또는 wave는 웨이브폼 오디오 포맷(waveform audio format)
- 개인용 컴퓨터에서 오디오를 재생하는 마이크로소프트와 IBM 오디오 파일 포맷 표준



WAVE

- wav 또는 wave는 웨이브폼 오디오 포맷(waveform audio format)
- 개인용 컴퓨터에서 오디오를 재생하는 마이크로소프트와 IBM 오디오 파일 포맷 표준
- 우리가 흔히 알고 있는 소리 신호를 -1. ~ 1. 사이의 값으로 저장한다.

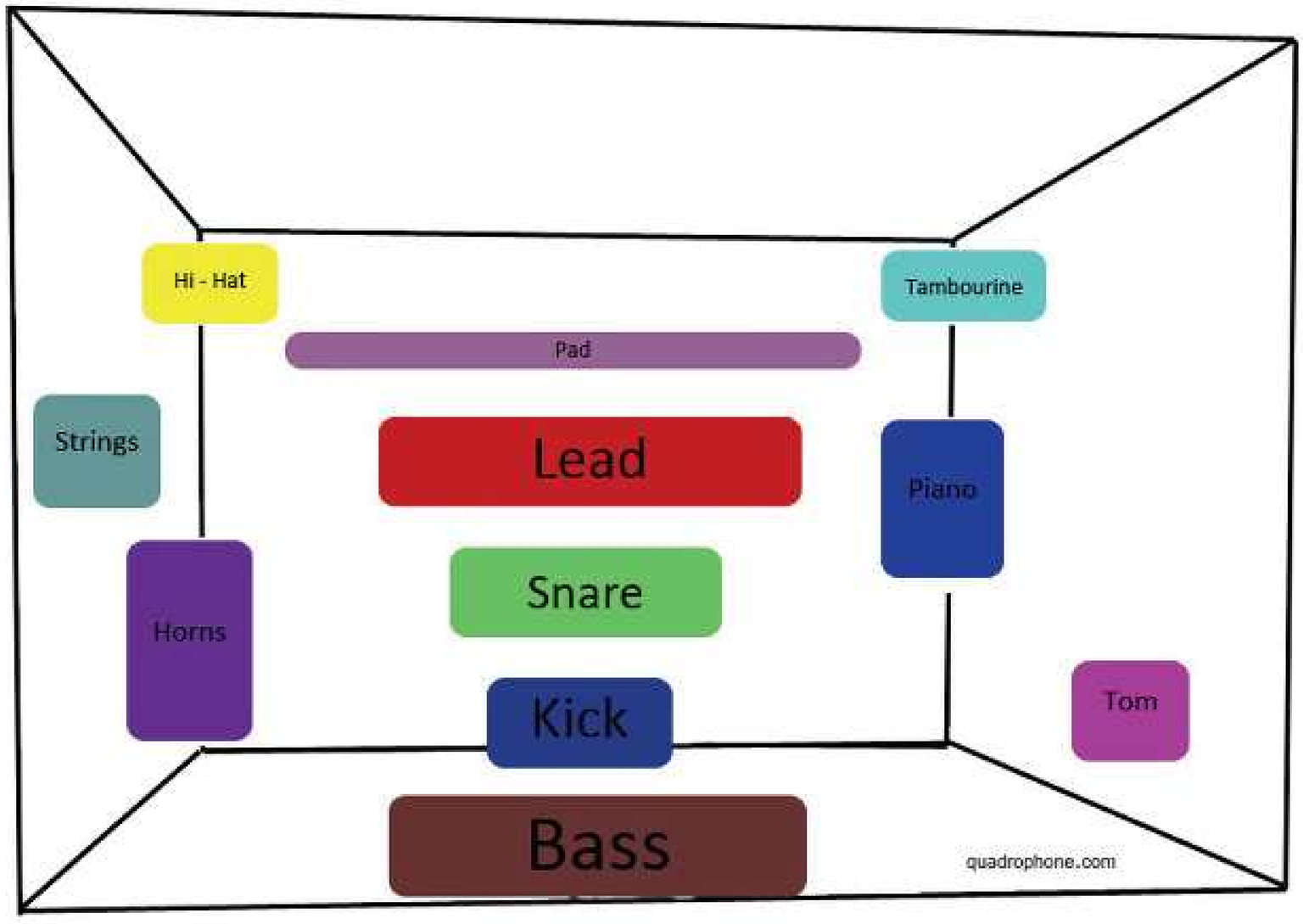



1
1.2
1.3
1.4
2
2.2
2.3
2.4
3
3.2
3.3
3.4
4
4.2
4.3
4.4
5
5.2
5.3
5.4
6
6.2
6.3
6.4
7
7.2
7.3
7.4
8
8.2
8.3
8.4

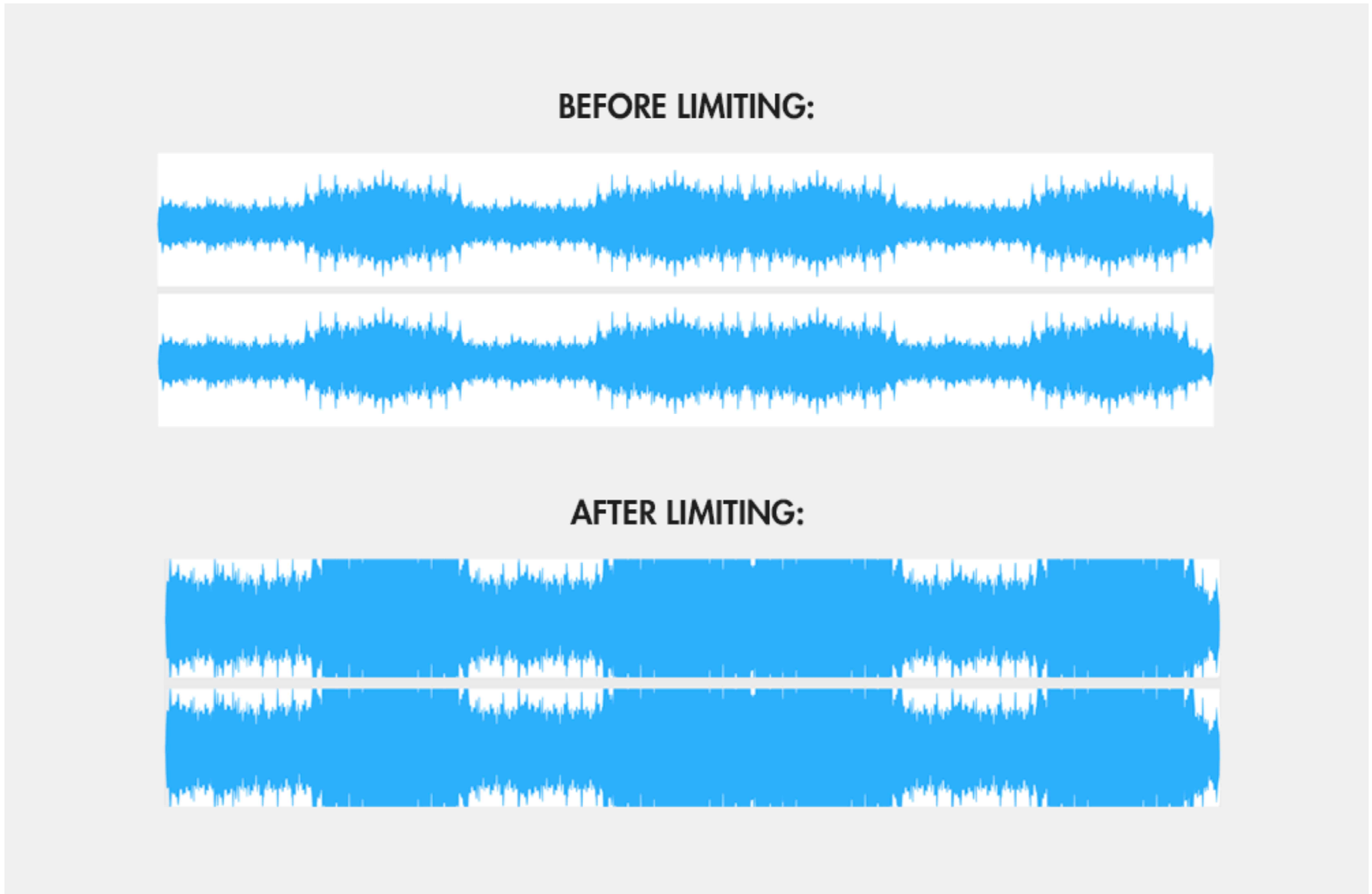

Ableton


MIX & MASTER

MIX / Master


Hi - Hat
Tambourine
Pad
Strings
Lead
Piano
Horns
Snare
Kick
Tom
Bass
quadrophone.com

MIX / Master



Rubato

이 과정에 AI 가 적용될 수 있을까?

AI가 세계를 지배할 거라는
AI 알못들:

내가 만든 AI:

어디에 적용해 볼 수 있을까?

Rubato

MIDI

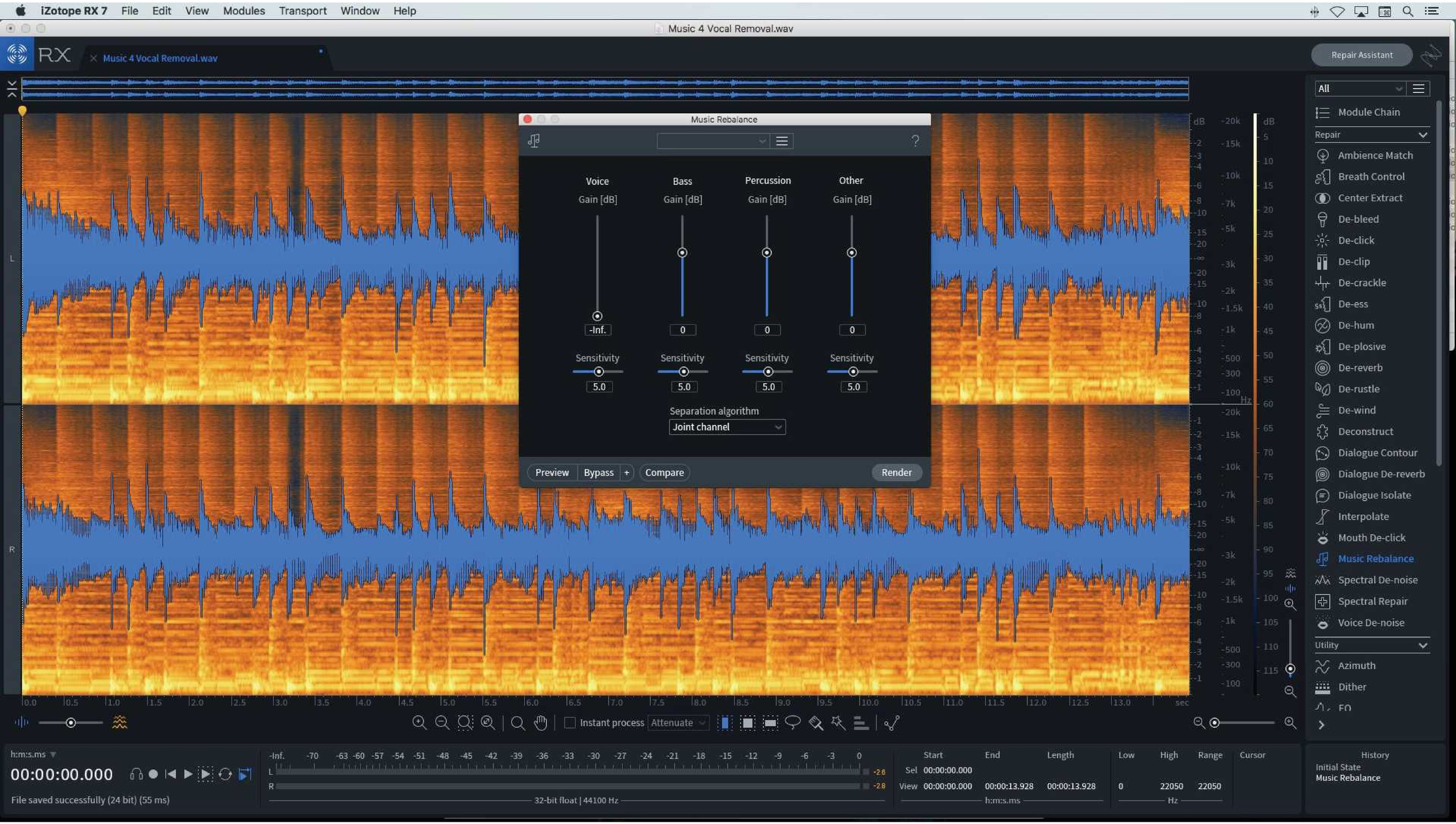
- Live performance
- Generate Melody
- Generate Accompaniments
- …

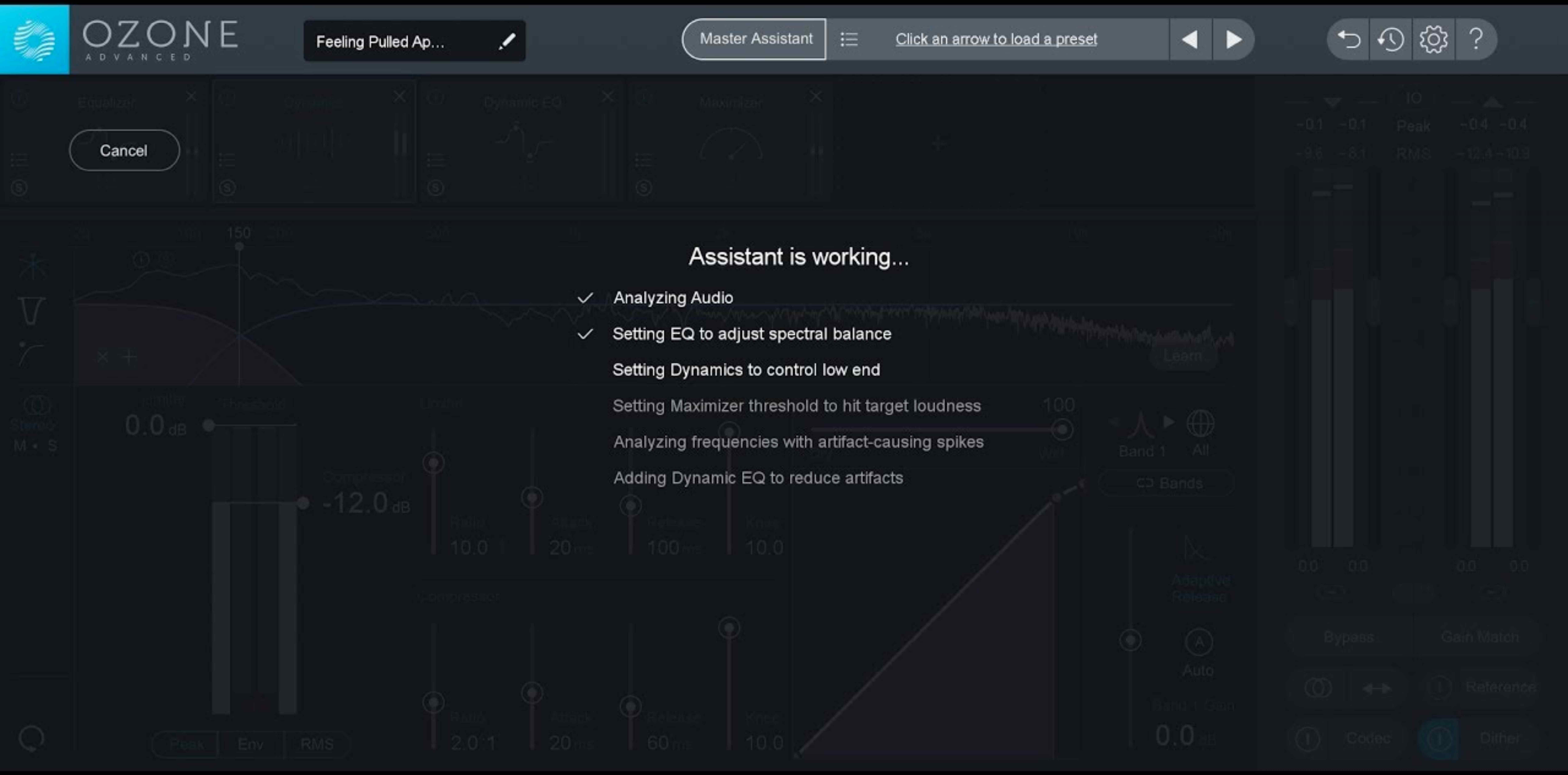
WAVE

- Sound design
- Mix / Master
- Source Separation
- Music Analysis
- …

magenta


Sony CSL


iZotope RX 7
File
Edit
View
Modules
Transport
Window
Help
Music 4 Vocal Removal.wav
RX
Music 4 Vocal Removal.wav
Repair Assistant
Music Rebalance
Voice
Gain [dB]
-Inf.
Bass
Gain [dB]
0
Percussion
Gain [dB]
0
Other
Gain [dB]
0
Sensitivity
5.0
Separation algorithm
Joint channel
Preview
Bypass
Compare
Render
All
Module Chain
Repair
Ambience Match
Breath Control
Center Extract
De-bleed
De-click
De-clip
De-crackle
De-ess
De-hum
De-plosive
De-reverb
De-rustle
De-wind
Deconstruct
Dialogue Contour
Dialogue De-reverb
Dialogue Isolate
Interpolate
Mouth De-click
Music Rebalance
Spectral De-noise
Spectral Repair
Voice De-noise
Utility
Azimuth
Dither
Instant process
Attenuate
h:m:s.ms
00:00:00.000
File saved successfully (24 bit) (55 ms)
Start
End
Length
Low
High
Range
Cursor
Sel 00:00:00.000
View 00:00:00.000
00:00:13.928
00:00:13.928
0
22050
22050
32-bit float | 44100 Hz
History
Initial State
Music Rebalance


OZONE
ADVANCED
Feeling Pulled Ap...
Master Assistant
Click an arrow to load a preset
Cancel
Assistant is working...
Analyzing Audio
Setting EQ to adjust spectral balance
Setting Dynamics to control low end
Setting Maximizer threshold to hit target loudness
Analyzing frequencies with artifact-causing spikes
Adding Dynamic EQ to reduce artifacts

Rubato

Music Information Retrieval (MIR)

Rubato

그럼 적용해 봅시다.

Rubato

MIDI data preprocessing

MIDI preprocessing

Rubato

- 그럼 어떻게 이런 데이터를 정제해야할까?

MIDI preprocessing

- 그럼 어떻게 이런 데이터를 정제해야할까?
- 다른 분야와 마찬가지로 MIDI 데이터를 다룰 수 있는 패키지들이 존재

MIDI preprocessing

- 그럼 어떻게 이런 데이터를 정제해야할까?
- 다른 분야와 마찬가지로 MIDI 데이터를 다룰 수 있는 패키지들이 존재
- mido, python-midi, pretty-midi, music21 …

MIDI Library

- mido
  - 파이썬에서 midi 데이터를 다루는 패키지이다.
  - Mido는 rtmidi를 기반으로 하기 때문에 미디 포트를 직접 열거나 닫을 수 있다.

MIDI Library

- mido
  - 파이썬에서 midi 데이터를 다루는 패키지이다.
  - Mido는 rtmidi를 기반으로 하기 때문에 미디 포트를 직접 열거나 닫을 수 있다.

- python-midi
  - High level API로 MIDI 데이터를 쉽게 write 할 수 있다. Linux를 쓴다면 ALSA의 시퀀서의 기능을 쓸 수 있다.

MIDI Library

- mido
  - 파이썬에서 midi 데이터를 다루는 패키지이다.
  - Mido는 rtmidi를 기반으로 하기 때문에 미디 포트를 직접 열거나 닫을 수 있다.

- python-midi
  - High level API로 MIDI 데이터를 쉽게 write 할 수 있다. Linux를 쓴다면 ALSA의 시퀀서의 기능을 쓸 수 있다.

- Pretty-midi
  - mido를 기반(?)으로 MIDI 데이터를 쉽게 읽고 쓸 수 있게 해준다.

MIDI Library

Rubato

- Music 21
  - Simple하게 구현했다는 midi library이다. 꾸준히 업데이트 되고 있으며, 가이드나 doc이 상세한 편이다.

MIDI Library

Rubato

- Music 21
  - Simple하게 구현했다는 midi library이다. 꾸준히 업데이트 되고 있으며, 가이드나 doc이 상세한 편이다.

- 아쉽게도 오디오 쪽의 Librosa 와 다르게 통합된 좋은 라이브러리가 아직 없기때문에 여러 라이브러리를 사용하기도 한다.

MIDI preprocessing

Rubato

- 그럼 어떻게 MIDI 데이터들을 가져올 수 있을까?

MIDI preprocessing

Rubato

- 그럼 어떻게 MIDI 데이터들을 가져올 수 있을까?
- 먼저 우리에게 필요한 데이터가 무엇인지 알아보자

MIDI preprocessing

Rubato

- 그럼 어떻게 MIDI 데이터들을 가져올 수 있을까?
- 먼저 우리에게 필요한 데이터가 무엇인지 알아보자
- 어떤 음이 나와야하는지?

MIDI preprocessing

Rubato

- 그럼 어떻게 MIDI 데이터들을 가져올 수 있을까?
- 먼저 우리에게 필요한 데이터가 무엇인지 알아보자
- 어떤 음이 나와야하는지?
- 음이 얼마나 나와야하는지?

MIDI preprocessing

Rubato

- 그럼 어떻게 MIDI 데이터들을 가져올 수 있을까?
- 먼저 우리에게 필요한 데이터가 무엇인지 알아보자
- 어떤 음이 나와야하는지?
- 음이 얼마나 나와야하는지?
- 얼마나 세게 음이 나와야하는지?

MIDI preprocessing

Rubato

- 그럼 어떻게 MIDI 데이터들을 가져올 수 있을까?
- 먼저 우리에게 필요한 데이터가 무엇인지 알아보자
- 어떤 음이 나와야하는지?
- 음이 얼마나 나와야하는지?
- 얼마나 세게 음이 나와야하는지?
- Note on / off, velocity 를 알아야 기본적인 음악의 종류를 파악할 수 있다.

```python
def get_eventlist(data_file):

    '''
    event : [Time, Type(ON, OFF, CC), Value1, Value2]
    '''

    ON = 1
    OFF = 0
    CC = 2

    midi = mido.MidiFile(data_file)

    current_time = 0
    eventlist = []
    cc = False
    for msg in midi:
        #print(msg)
        current_time += msg.time

         # NOTE ON CASE
        if msg.type is 'note_on' and msg.velocity > 0:
            event = [current_time, ON, msg.note, msg.velocity]
            eventlist.append(event)

     # NOTE OFF CASE
    elif msg.type is 'note_off' or (msg.type is 'note_on' and msg.velocity == 0):
        event = [current_time, OFF, msg.note, msg.velocity]
        eventlist.append(event)

    if msg.type is 'control_change':

        # 64 sustain pedal
        if msg.control != 64:
            continue

        if cc == False and msg.value > 0:
            cc = True
            event = [current_time, CC, 0, 1]
            eventlist.append(event)

        elif cc == True and msg.value == 0:
            cc = False
            event = [current_time, CC, 0, 0]
            eventlist.append(event)

eventlist = np.array(eventlist)
return eventlist
```

- Event 형식으로 데이터 파싱 - mido 사용
  - MIDI 데이터 타입을 이용하여 각 노트를 파싱
  - Note on : 1
  - Note off : 0
  - CC (control change) : 2

- 파싱하는 이벤트는 Note on, Note off (Velocity == 0), control_change

- Note off 와 Velocity를 같이 파싱하는 이유는 음이 끝나는 부분을 Note off, 혹은 Velocity 를 0로 설정하기 때문이다.

- Control Change 부분 주로 CC로 나타내며 CC64는 피아노의 페달 데이터를 나타낸다.

- 위 데이터를 모두 파싱하게 되면 모델에 넣을 수 있는 numpy array로 표현 할 수 있다.

  Event ：[Time, Data Type, Value1, Value2]

Rubato

Rubato

MIDI dataset

Dataset

- 구글 마젠타에서 제공한 Maestro 데이터셋 사용하였다.

Dataset

- 구글 마젠타에서 제공한 Maestro 데이터셋 사용하였다.
- 실제 연주한 데이터를 MIDI와 WAVE 형태로 200여 시간 저장되어 있다.

Dataset

Rubato

- 구글 마젠타에서 제공한 Maestro 데이터셋 사용하였다.
- 실제 연주한 데이터를 MIDI와 WAVE 형태로 200여 시간 저장되어 있다.
- 위 데이터를 Music Information Retrieval을 위한 용도로 사용 가능하다. 현재 자동 작곡, Wave2MIDI 등 다양한 분야에서 사용되고 있는 데이터셋이다.

Dataset

Rubato

- 구글 마젠타에서 제공한 Maestro 데이터셋 사용하였다.
- 실제 연주한 데이터를 MIDI와 WAVE 형태로 200여 시간 저장되어 있다.
- 위 데이터를 Music Information Retrieval을 위한 용도로 사용 가능하다. 현재 자동 작곡, Wave2MIDI 등 다양한 분야에서 사용되고 있는 데이터셋이다.
- 야마하 사와 International Piano-e-Competition을 10여년에 걸쳐 개최
- 야마하 디스클라비어를 사용하여 MIDI 데이터와 WAVE 데이터를 수집하였다.

Maestro Dataset v2

Field	Description
canonical_composer	작곡가 이름
canonical_title	곡의 제목
split	train/validation/test 을 나타냄
year	공연 연도
midi_filename	MIDI 파일 이름
audio_filename	WAV 파일 이름
duration	미디 파일 재생시간 (초)

- Meta Data

  - Maestro dataset은 meta 정보를 포함하기 때문에 위 정보를 이용하여 데이터를 편하게 처리할 수 있다.
  - meta data를 이용하여 한 작곡가의 데이터만 뽑거나 조절하는 것이 가능하다.

Maestro Dataset v2

Split	Performances	Duration (hours)	Size (GB)	Notes (millions)
Train	967	161.3	97.7	5.73
Validation	137	19.4	11.8	0.64
Test	178	20.5	12.4	0.76
Total	1282	201.2	121.8	7.13

- Maestro dataset은 총 1282개의 퍼포먼스, train 161.3시간, valid 19.4시간, test 20.5시간으로 이루어져있으며, MIDI와 WAV 가 같이 제공된다.

- 여기에 표시된 Size는 WAV 파일의 Size이며, 실제 MIDI 파일의 크기는 훨씬 작다.

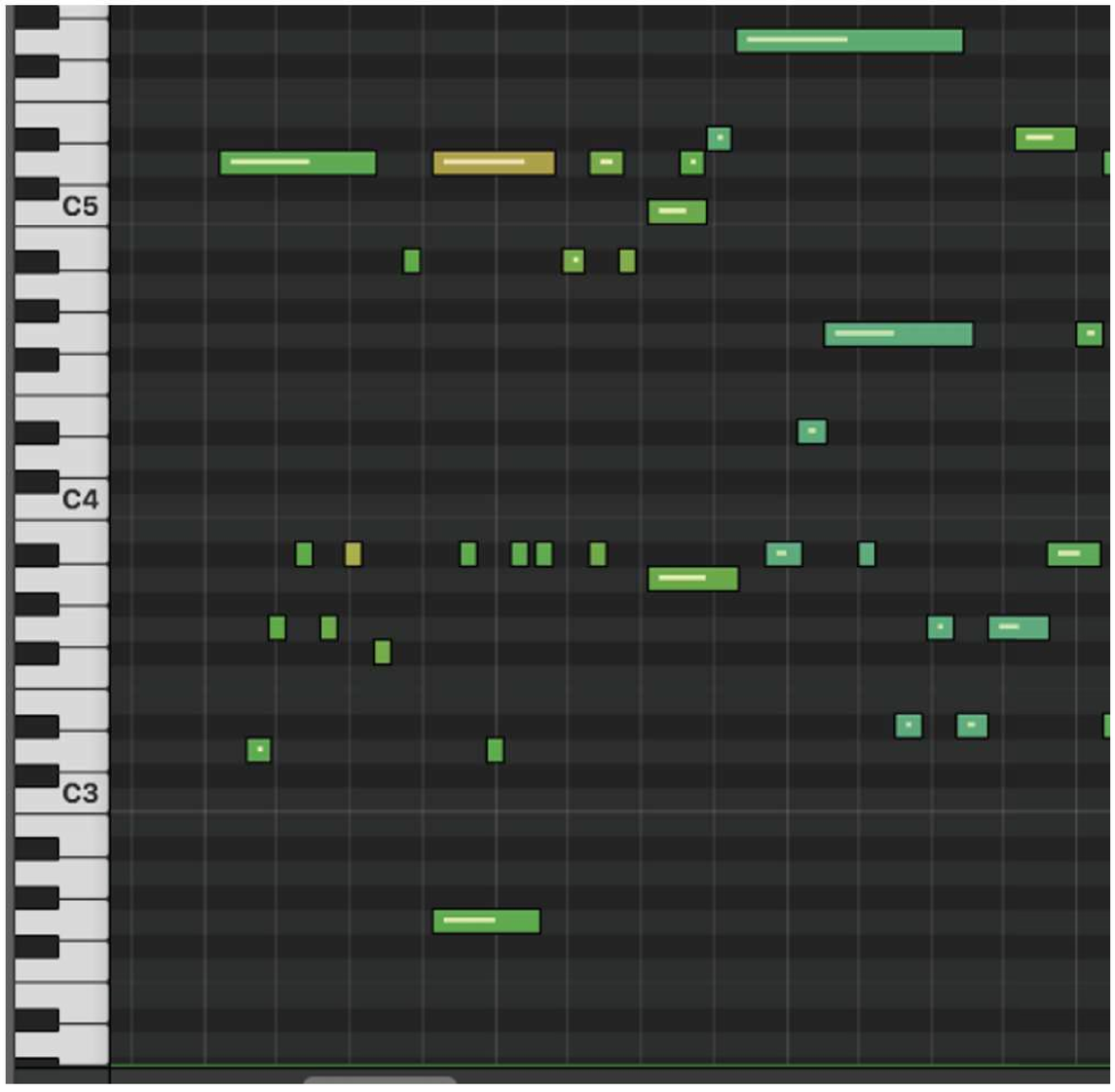
- Notes는 음의 갯수를 나타내며 단위는 음하나하나가 한 데이터셋에 몇개가 들어가 있는지를 표시한다.

Rubato

DL model for Music Generation

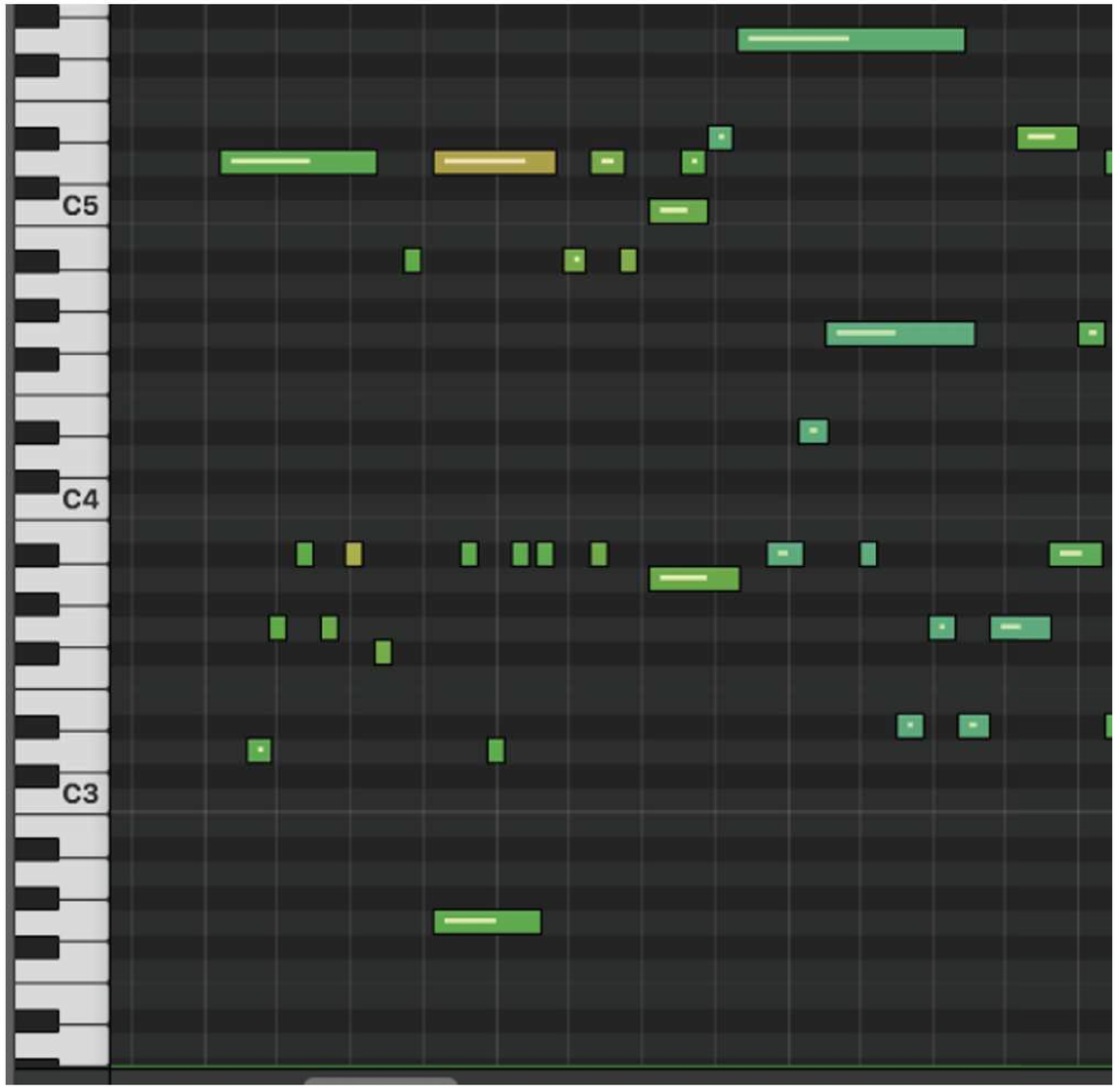
Performance RNN

- Magenta에서 2017년에 공개한 RNN 기반 모델



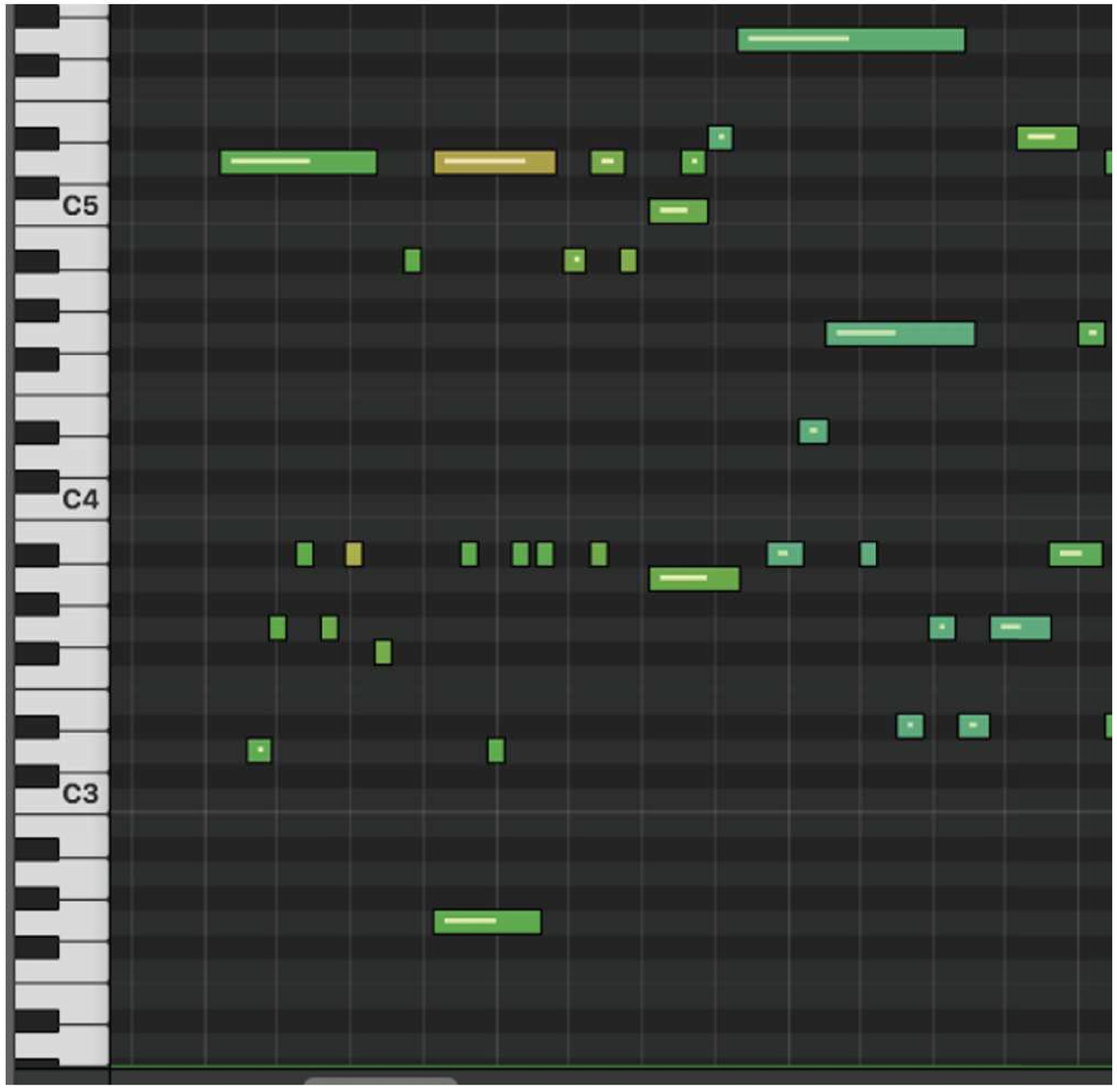
Performance RNN

- Magenta에서 2017년에 공개한 RNN 기반 모델
- MIDI note에 대한 embedding 방법 제시
  - 128 note-on
  - 128 note-off
  - 100 time-shift
  - 32 velocity (32 bins)



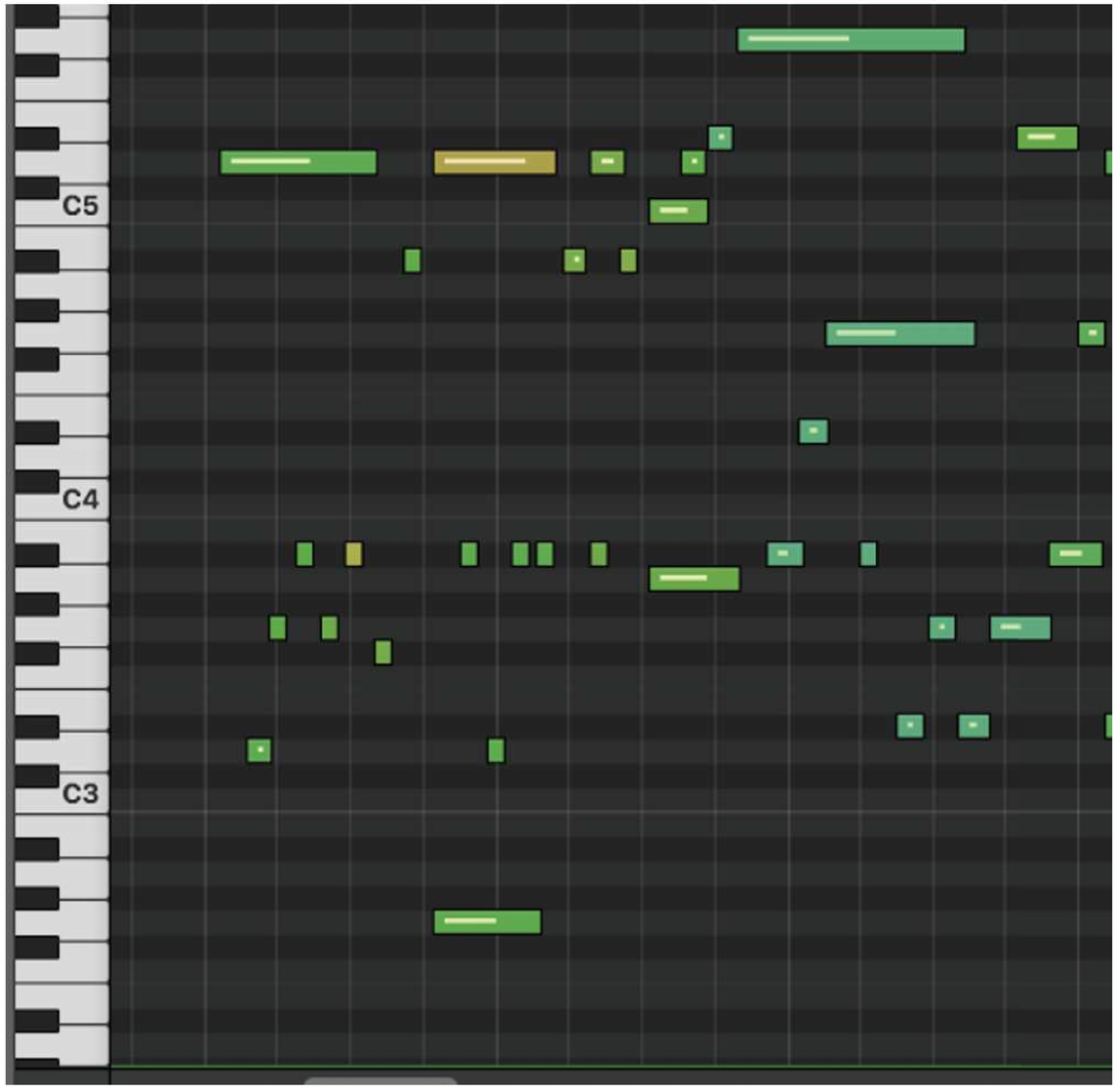
Performance RNN

- Magenta에서 2017년에 공개한 RNN 기반 모델
- MIDI note에 대한 embedding 방법 제시
  - 128 note-on
  - 128 note-off
  - 100 time-shift
  - 32 velocity (32 bins)
- 총 388개의 events를 갖는다.



Performance RNN

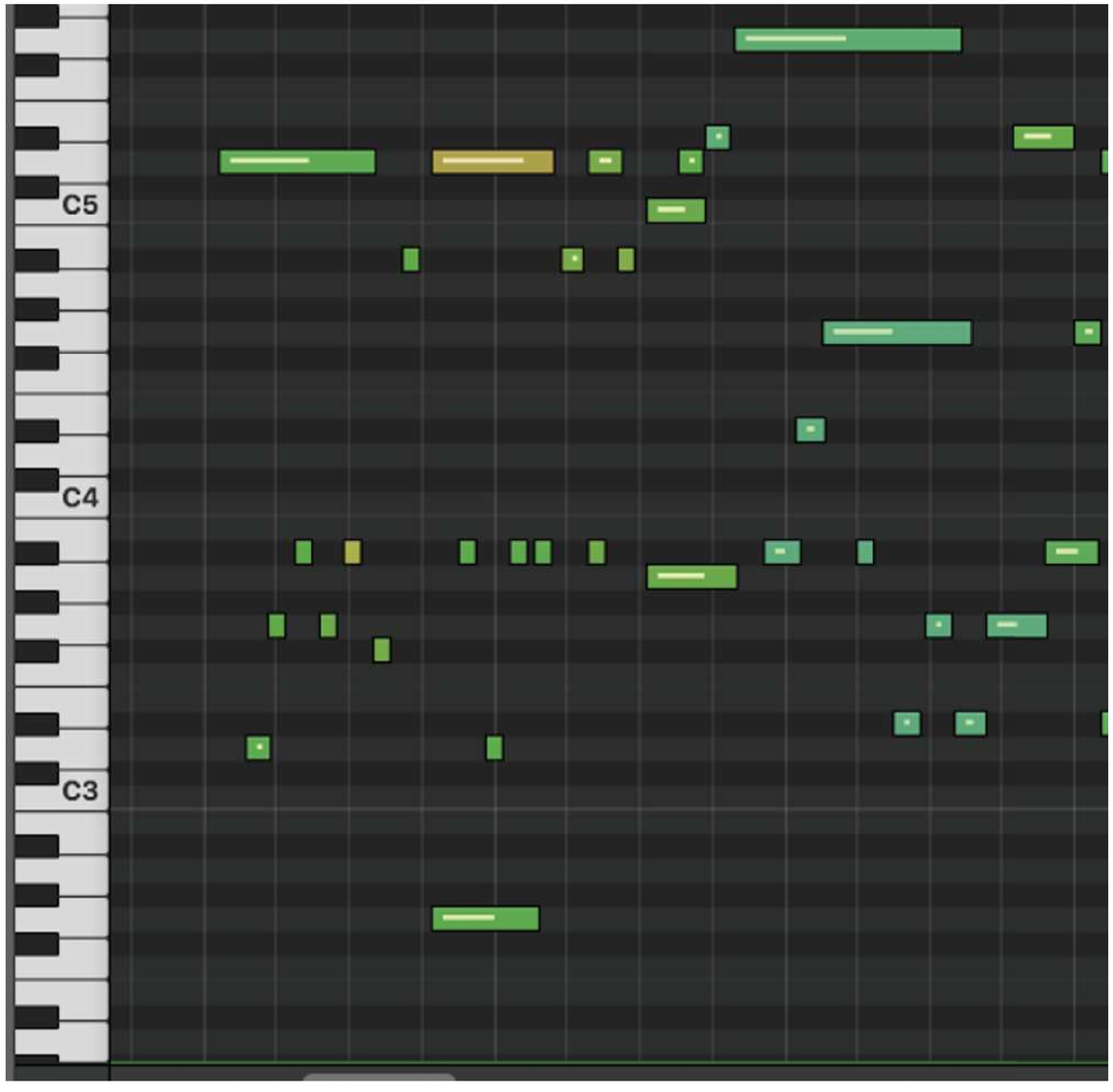
- Magenta에서 2017년에 공개한 RNN 기반 모델
- MIDI note에 대한 embedding 방법 제시
  - 128 note-on
  - 128 note-off
  - 100 time-shift
  - 32 velocity (32 bins)
- 총 388개의 events를 갖는다.
- 30초에 1200개 정도의 events vector를 가진다.



Performance RNN

Rubato

- RNN의 한계로 음악의 주제가 길게 이어지지 못함
- 생성 될때 마다 다른 음악같은 한계점 존재



Transformer

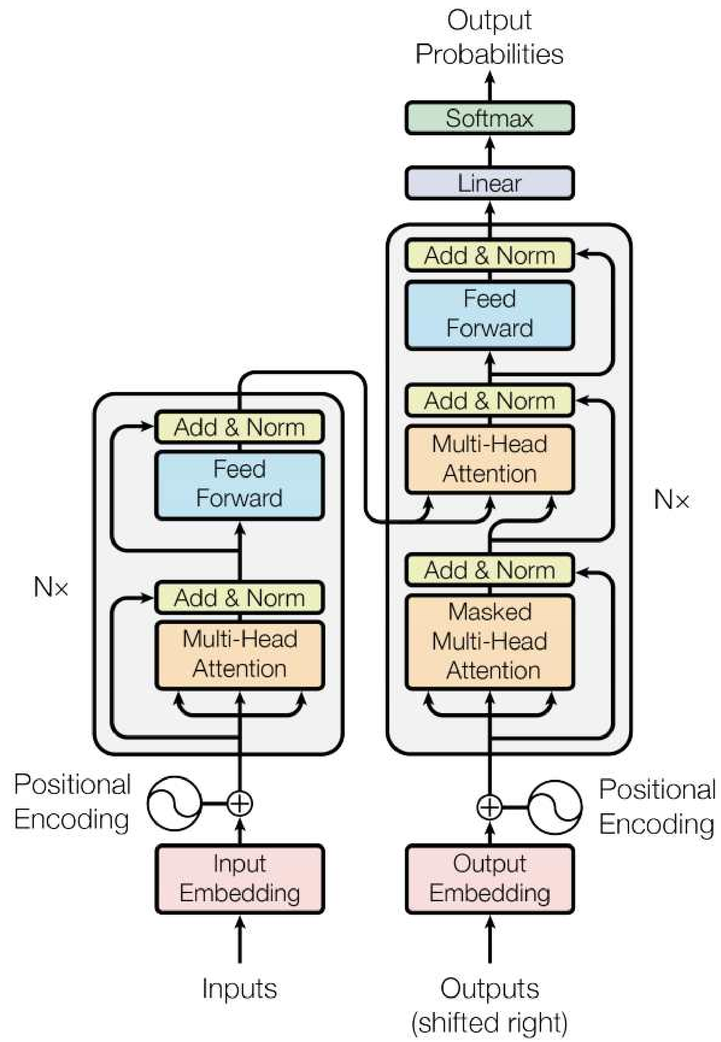
- Transformer는 크게 Inputs 데이터를 인코딩하는 Encoder와, 인코딩 된 데이터를 기반으로 Outputs을 디코딩하여 Outputs의 확률분포 파라메터를 출력하는 Decoder로 구성된다.

- 이러한 구조는 seq2seq 모델이라고 불리며 원래 언어의 문장을 대상 언어의 문장으로 변환시키는 번역 작업이나, 질의 문장을 토대로 응 답 문장을 생성해내는 질의 응답 작업 등에 사용되었다.

- Encoder는 원래 언어의 문장의 토큰을 입력으로 받아 임베딩 테이블 을 이용해 벡터로 변환하고 여러 인코딩 블록을 거쳐 최종적으로 Decoder에 들어갈 형태로 출력한다.

- Decoder도 마찬가지로 대상 언어의 문장의 토큰을 입력으로 받아 임 베딩 테이블을 이용해 벡터로 변환하고 여러 디코딩 블록을 거쳐 다 음 토큰에 해당하는 확률 분포의 파라메터를 출력한다.

- 이 때, 디코딩 블록 내에서 어텐션 메카니즘을 통해 Encoder에서 인 코딩된 정보를 가져온다. 어텐션 시 여러 헤드로 분리하여 각각 다양 한 부분의 정보를 가져올 수 있도록 하며, 가져온 정보들은 concatenation하여 다음 레이어로 전해진다.



Music Transformer

- 기존의 Transformer에서는 Absolute Positional Embedding을 통해 문자열의 각 토큰이 어디에 존재하는지를 절대적인 위치정보를 통해서 나타내었으나, 이러한 방법은 문자들 간의 상대적인 위치관계를 담아내기에 부족한 점이 있다.

- 이러한 점을 극복하고자 고안된 방법이 Relative Positional Embedding이다. 토큰의 절대 위치 정보를 입력 레이어에서부터 임베딩 시키지 않고, Attention Score를 구할 때 상대적인 위치를 측정해 Weight를 부여하고 계산한다.

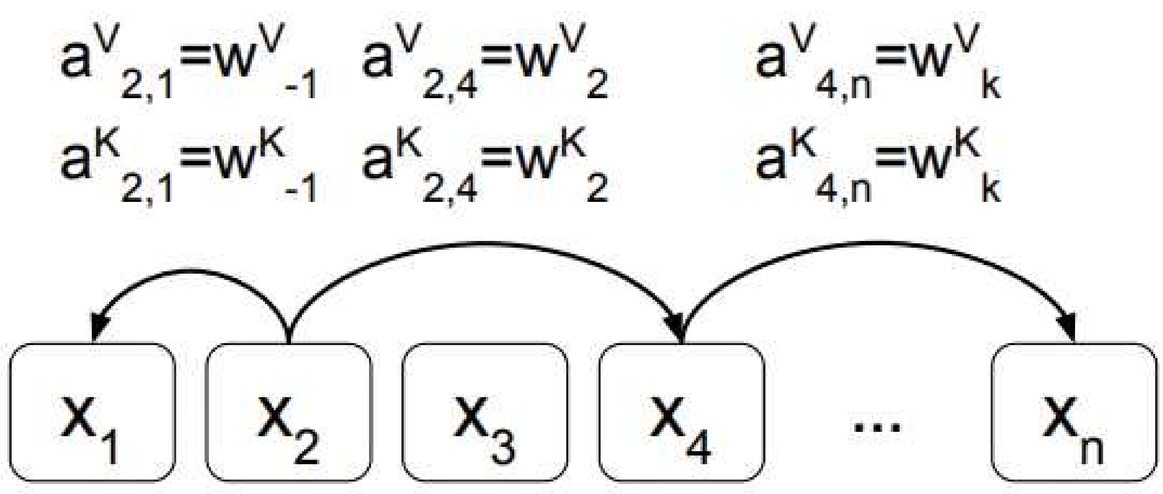
$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$RelativeAttention(Q, K, V) = \text{softmax}\left(\frac{QK^\top + QR^T}{\sqrt{d_k}}\right)V$$

Music Transformer

- 아래 그림에서 x_2벡터가 x_1의 정보를 참조하기 위해 사용될 Positional Embedding은 x_1의 첫번째 위치에 대한 값이 아니라 1-2=-1 즉, 왼쪽으로 첫번째 순서에 있는 위치에 대한 값이 사용됨을 표현하고 있다.

- 같은 식으로 x_2벡터에서 x_4벡터의 정보를 참조하기 위해서는 네번째 위치에 대한 값이 아니라 4-2=2 즉, 오른쪽으로 두번째 순서에 있는 위치 에 대한 값을 사용한다.



Music Transformer

Rubato

- Absolute Positional Embedding에서는 입력 벡터 한 개에 대해 하나의 임베딩 벡터만 대응되었다. 이와 달리 Relative Positional Embedding에서는 가능한 모든 임베딩 벡터가 대응될 수 있다.

Music Transformer

- Absolute Positional Embedding에서는 입력 벡터 한 개에 대해 하나의 임베딩 벡터만 대응되었다. 이와 달리 Relative Positional Embedding에서는 가능한 모든 임베딩 벡터가 대응될 수 있다.

- 이를 위해 모든 입력 벡터와 모든 임베딩 벡터 간의 조합을 고려해야 하는데, 이는 매우 메모리가 많이 소모되는 작업이다. 이를 위해 Music Transformer에서 효율적인 임베딩 방법을 고안해내었다.

Music Transformer

Rubato

- Absolute Positional Embedding에서는 입력 벡터 한 개에 대해 하나의 임베딩 벡터만 대응되었다. 이와 달리 Relative Positional Embedding에서는 가능한 모든 임베딩 벡터가 대응될 수 있다.

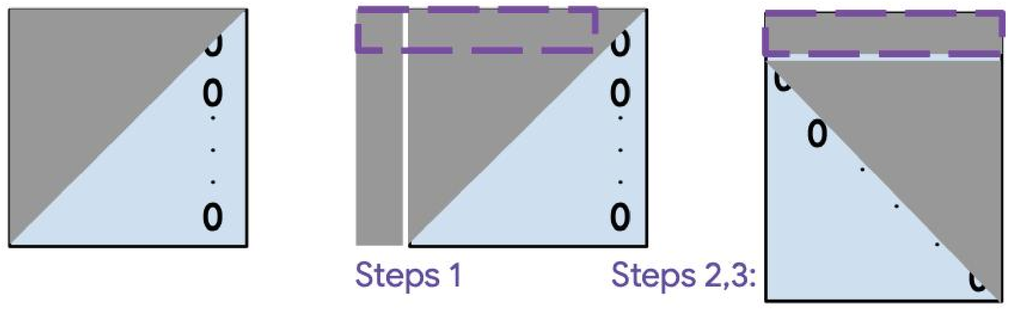
- 이를 위해 모든 입력 벡터와 모든 임베딩 벡터 간의 조합을 고려해야 하는데, 이는 매우 메모리가 많이 소모되는 작업이다. 이를 위해 Music Transformer에서 효율적인 임베딩 방법을 고안해내었다.

- Attention Score를 계산하는 단계에서 Query Q와 Embedding Table E^r과 간에 행렬곱 (QE)^r.T을 얻고서 다음과 같은 과정을 수행한다.

Music Transformer

- 가장 왼쪽의 column에 length L 만큼 dummy column을 패딩한다.
- (L+1, L) shape을 가지도록 matrix를 reshape 해준다.
- Reshape 한 (L+1, L) Matrix의 첫번째 dummy row 제거해준다.
- 결과적으로 다시 (L, L) Matrix 가 되면, 우리가 구하는 S^rel이 된다.

PREVIOUS FIGURES FOR THE "SKEWING" PROCEDURE

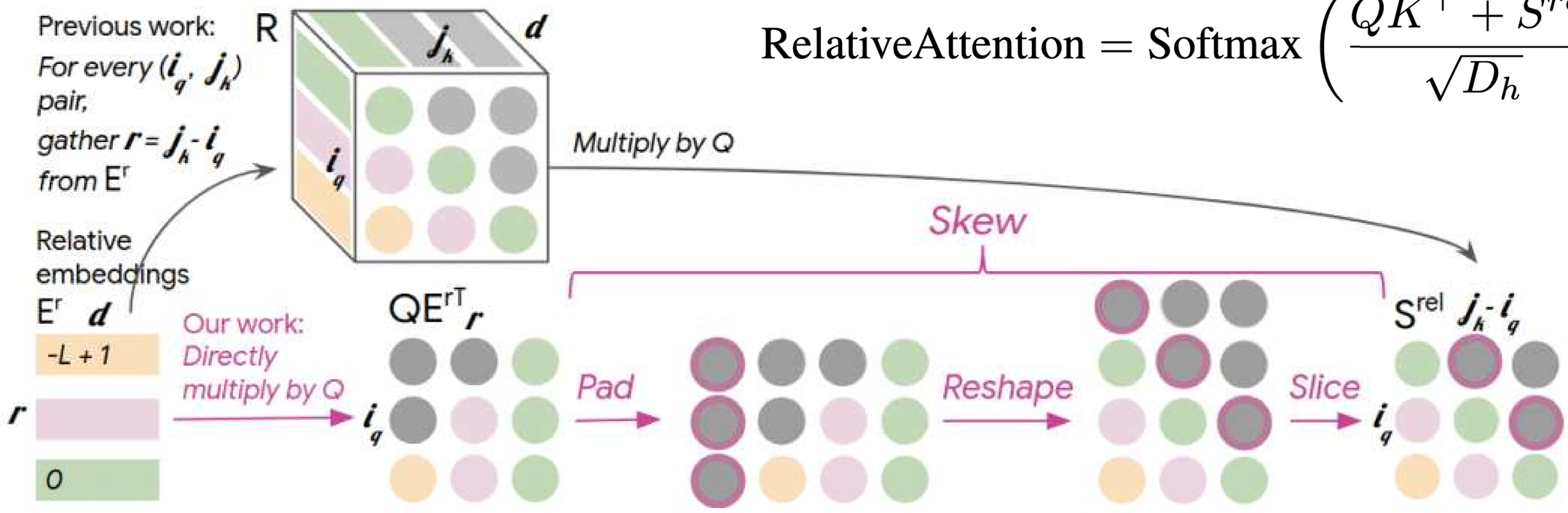


Music Transformer

Rubato

- 아래 그림은 위 과정을 도식화한 그림이며 이를 통해 행렬곱 (QE)^r.T 가 비스듬히 한칸씩 옆으로 밀려나 원하는 모습인 S^rel으로 변환됨을 알 수 있다.

$$\begin{matrix} QE^{r\top} \\ \| \\ \text{RelativeAttention} = \text{Softmax}\left(\frac{QK^\top + S^{rel}}{\sqrt{D_h}}\right)V. \end{matrix}$$



GPT-2

- GPT-2는 Transformer를 기반으로 openAI에서 만든 모델이다. 자연어를 비롯한 시계열 데이터의 생성에 있어서 좋은 성능을 내고 있다.

GPT-2

- GPT-2는 Transformer를 기반으로 openAI에서 만든 모델이다. 자연어를 비롯한 시계열 데이터의 생성에 있어서 좋은 성능을 내고 있다.

- 기본적으로 Transformer와 비슷한 구조를 가지고 있으나 다음과 같은 두가지 차이점이 있다.

GPT-2

- 마지막 Softmax 이전 Linear 레이어의 Weight를 Word Embedding Table로 둔다.

```
169         # Language model loss.  Do tokens <n predict token n?
170         h_flat = tf.reshape(h, [batch*sequence, hparams.n_embd])
171         logits = tf.matmul(h_flat, wte, transpose_b=True)
172         logits = tf.reshape(logits, [batch, sequence, hparams.n_vocab])
173         results['logits'] = logits
174         return results
```

GPT-2

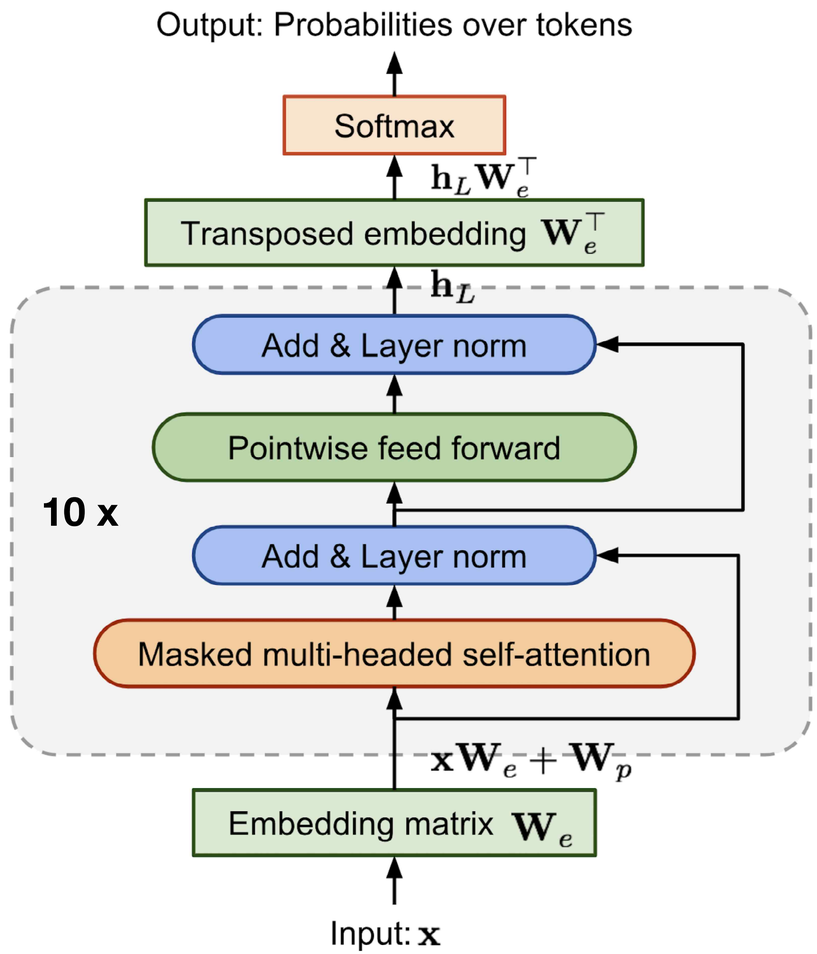
- transformer_block에 사용되는 RELU Activation을 GELU Activation으로 대체하였다.

```
25  def gelu(x):
26      return 0.5*x*(1+tf.tanh(np.sqrt(2/np.pi)*(x+0.044715*tf.pow(x, 3))))
```

```
115 def mlp(x, scope, n_state, *, hparams):
116     with tf.variable_scope(scope):
117         nx = x.shape[-1].value
118         h = gelu(conv1d(x, 'c_fc', n_state))
119         h2 = conv1d(h, 'c_proj', nx)
120         return h2
```

Music GPT-2

- Music GPT-2는 공개된 GPT-2 소스를 기반으로 Music Transformer에서 사용된 Relative Positional Embedding을 추가하여 만들었다.
- Music GPT-2의 경우엔 인코딩할 소스가 존재 하지 않는다.
- Encoder부분을 떼어내고 Decoder부분만을 이용해 Autoregressive 모델로 작동하도록 만들었다.



Music GPT-2

Rubato

모델 하이퍼 파라메터 비교

	GPT-2	Music Transformer	Music GPT-2
Embedding Dimension	768	256, 512	512
Number of Heads	12	8	16
Number of Layers	12	4, 5, 6	8
Length of Attention	N/A	3500	2000

Music GPT-2

- loss는 autoregressive models에서 사용되는 예측 분포와 실제 데이터 간의 cross entropy에 의해 계산 된다. 예측 분포는 categorical distribution으로 나타내고, 실제 데이터는 one-hot vector로 구성된다.

- categorical distribution의 클래스 갯수가 C개이고, 실제 데이터 one-hot vector의 c번 째 값이 t_c, 예측 분포에서 c번째 클래스의 확률이 y_c라 할 때 cross entropy는 다음과 같이 계산된 다.

$$CE = -\sum_{c}^{C} t_c log(y_c)$$

Music GPT-2

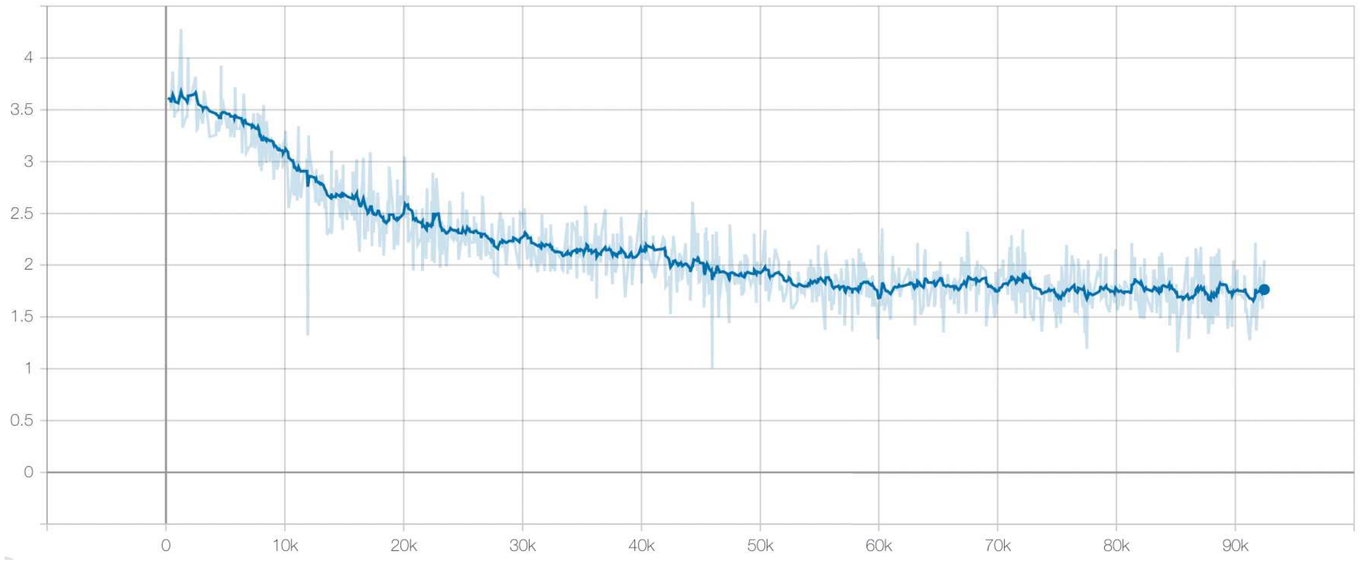
- 트레이닝 시에 데이터는 배치 사이즈와 하이퍼 파라메터 상 정해진 attention 길이를 가지므로 shape이 [batch, attention_length]로 구성된다.

- batch를 B, attention_length를 L이라 할 때, batch와 attention_length 축으로 평균값을 취한 cross entropy는 다음과 같이 계산된다.

$$CE = -\frac{1}{BL}\sum_{b}^{B}\sum_{l}^{L}\sum_{c}^{C} t_c log(y_c)$$

Music GPT-2

- 트레이닝은 2080ti 4대를 이용 배치 사이즈 4로 9만 step 이상 트레이닝 하였다.
- Optimizer 는 Adam을 사용했으며, 학습률은 1e-4로 유지하였다.
- Loss 는 1.5 ~ 2 사이로 수렴했다.
- Music Transformer 논문의 Validation NLL 1.835, 1.840와 근접한다.




2019 ARTIST NEST
즉흥춤 프로젝트
인공지능 작곡/연주
Three Musical Moments by GPT-2
모두의연구소 Rubato Lab
AI Performance by VirtuosoNet
카이스트 문화기술대학원 MAC Lab
AiFrenz
KAIST
모두의연구소
YAMAHA


YAMAHA

Reference

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, 2017.

- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In Proc. of NAACL.

- Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In Seventh International Conference on Learning Representations, 2018a

- Ian Simon and Sageev Oore. "Performance RNN: Generating Music with Expressive, Timing and Dynamics." Magenta Blog, 2017. , https://magenta.tensorflow.org/performance-rnn