작곡하기

양승우

다음과 같은 오류가 발생하면

경로 문제와 코드를 수정해야 하므로 해결 방법을 다음과 같이 넣었습니다.

개인적으로 해결하기 어려웠던 부분입니다. 다음 링크를 참고해도 좋습니다.

https://github.com/cuthbertLab/music21/issues/348

```
---------------------------------------------------------------------------
OSError                                   Traceback (most recent call last)
<ipython-input-10-4545969421aa> in <module>
----> 1 original_score.show()

~₩anaconda3₩lib₩site-packages₩music21₩stream₩__init__.py in show(self, *args, **kwargs)
    257         if self.isSorted is False and self.autoSort:
    258             self.sort()
--> 259         return super().show(*args, **kwargs)
    260
    261     # ------------------------------------------------------------------------

~₩anaconda3₩lib₩site-packages₩music21₩base.py in show(self, fmt, app, **keywords)
   2628                                 app=app,
   2629                                 subformats=subformats,
-> 2630                                 **keywords)
   2631
   2632     # ------------------------------------------------------------------------

~₩anaconda3₩lib₩site-packages₩music21₩converter₩subConverters.py in show(self, obj, fmt, app, subformats, **keywords)
    358             for s in scores:
    359                 fp = helperSubConverter.write(s, helperFormat,
--> 360                                               subformats=helperSubformats, **keywords)
    361
    362                 if helperSubformats[0] == 'png':

~₩anaconda3₩lib₩site-packages₩music21₩converter₩subConverters.py in write(self, obj, fmt, fp, subformats, **keywords)
    936                 and ('png' in subformats or 'pdf' in subformats)
    937                 and not str(environLocal['musescoreDirectPNGPath']).startswith('/skip')):
--> 938             fp = self.runThroughMusescore(fp, subformats, **keywords)
    939
    940         return fp

~₩anaconda3₩lib₩site-packages₩music21₩converter₩subConverters.py in runThroughMusescore(self, fp, subformats, **keywords)
    865                 'To create PNG files directly from MusicXML you need to download MuseScore and ' +
    866                 'put a link to it in your .music21rc via Environment.')
--> 867         if not musescorePath.exists():
    868             raise SubConverterException(
    869                     "Cannot find a path to the 'mscore' file at " +

~₩anaconda3₩lib₩pathlib.py in exists(self)
   1354         """
   1355         try:
-> 1356             self.stat()
   1357         except OSError as e:
   1358             if not _ignore_error(e):

~₩anaconda3₩lib₩pathlib.py in stat(self)
   1176         os.stat() does.
   1177         """
-> 1178         return self._accessor.stat(self)
   1179
   1180     def owner(self):

OSError: [WinError 123] 파일 이름, 디렉터리 이름 또는 볼륨 레이블 구문이 잘못되었습니다: 'C:₩₩Users₩₩User₩₩GDL_code₩₩C:₩₩Program Files (x8
6)₩₩MuseScore 3₩₩MuseScore.exe'
```

In [1]:
```
from music21 import *
```

In [2]:
```
# get environment
env = environment.Environment()

# check the path
print('Environment settings:')
print('musicXML:  ', env['musicxmlPath'])
print('musescore: ', env['musescoreDirectPNGPath'])
```

```
Environment settings:
musicXML:   None
musescore:  C:₩Users₩User₩GDL_code₩C:₩Program Files (x86)₩MuseScore 3₩MuseScore.exe
```

모두 앱 문서 웹 더 보기

가장 정확

musicxml
파일 폴더
마지막 수정: 2020. 7. 31. 오전 12:40

웹 검색

musicxml - 웹 결과 보기

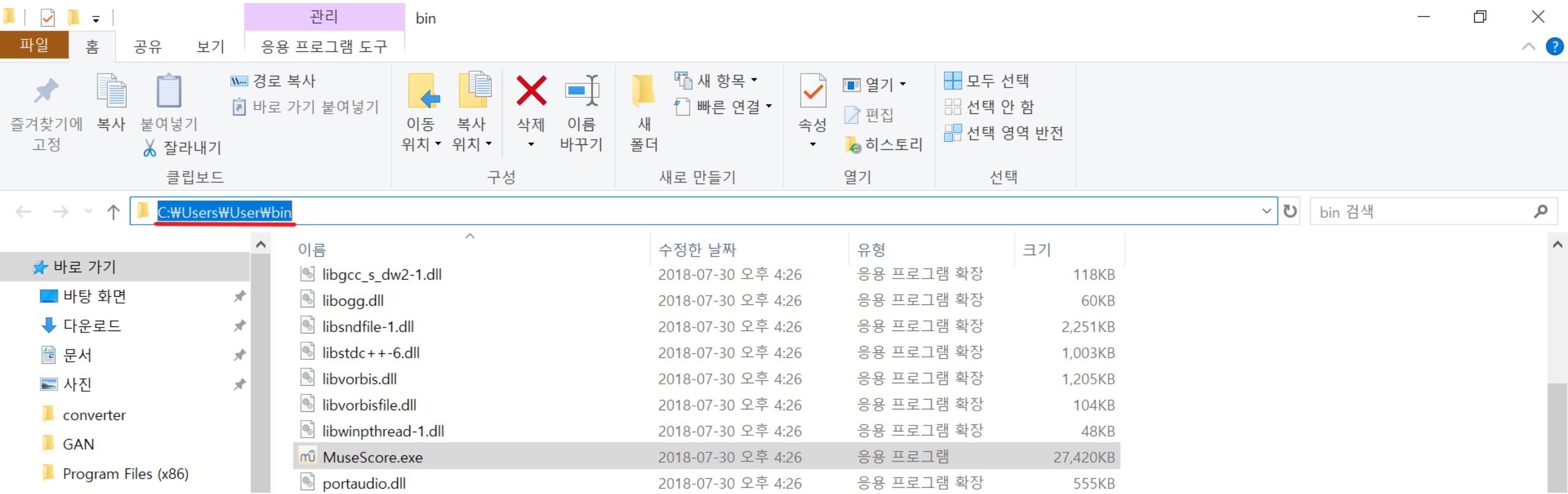
musicxml
파일 폴더

위치 C:₩Users₩User₩anaconda3₩envs₩testGAN₩lib₩site-packages₩music21

마지막 수정 2020. 7. 31. 오전 12:40

열기
파일 위치 열기
전체 경로 복사

musicxml


관리
bin
파일
홈
공유
보기
응용 프로그램 도구
즐겨찾기에 고정
복사
붙여넣기
잘라내기
경로 복사
바로 가기 붙여넣기
클립보드
이동 위치
복사 위치
삭제
이름 바꾸기
구성
새 폴더
새 항목
빠른 연결
새로 만들기
속성
열기
편집
히스토리
열기
모두 선택
선택 안 함
선택 영역 반전
선택
C:₩Users₩User₩bin
bin 검색
바로 가기
바탕 화면
다운로드
문서
사진
converter
GAN
Program Files (x86)
이름
수정한 날짜
유형
크기
libgcc_s_dw2-1.dll 2018-07-30 오후 4:26 응용 프로그램 확장 118KB
libogg.dll 2018-07-30 오후 4:26 응용 프로그램 확장 60KB
libsndfile-1.dll 2018-07-30 오후 4:26 응용 프로그램 확장 2,251KB
libstdc++-6.dll 2018-07-30 오후 4:26 응용 프로그램 확장 1,003KB
libvorbis.dll 2018-07-30 오후 4:26 응용 프로그램 확장 1,205KB
libvorbisfile.dll 2018-07-30 오후 4:26 응용 프로그램 확장 104KB
libwinpthread-1.dll 2018-07-30 오후 4:26 응용 프로그램 확장 48KB
MuseScore.exe 2018-07-30 오후 4:26 응용 프로그램 27,420KB
portaudio.dll 2018-07-30 오후 4:26 응용 프로그램 확장 555KB

In [ ]:
```
# set path
env['musicxmlPath'] = 'C:/Users/User/anaconda3/envs/testGAN/lib/site-packages/music21/musicxml'
env['musescoreDirectPNGPath'] = 'C:/Users/User/bin/MuseScore.exe'
```

jupyter

Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New

0 / anaconda3 / envs / testGAN / lib / site-packages / music21 / converter

	Name	Last Modified	File size
..		몇 초 전	
__init__.py		41분 전	72 kB
incorrectExtension.txt		41분 전	48 B
qmConverter.py		41분 전	4.42 kB
subConverters.py		41분 전	52.8 kB

```
        fpOut = fp[0:len(fp) - 3]
        fpOut += subformatExtension

        musescoreRun = '"' + str(musescorePath) + '" "' + fp + '" -o "' + fpOut + '" -T 0 '
        if 'dpi' in keywords:
            musescoreRun += ' -r ' + str(keywords['dpi'])

        if common.runningUnderIPython():
            musescoreRun += ' -r ' + str(defaults.ipythonImageDpi)

        storedStrErr = sys.stderr
        fileLikeOpen = io.StringIO()
        sys.stderr = fileLikeOpen
        os.system(musescoreRun)
        fileLikeOpen.close()
        sys.stderr = storedStrErr

        if subformatExtension == 'png':
            return self.findPNGfpFromXMLfp(fpOut)
        else:
            return fpOut
        # common.cropImageFromPath(fp)

    def writeDataStream(self, fp, dataBytes):  # pragma: no cover
        if fp is None:
            fp = self.getTemporaryFile()
        else:
            fp = common.cleanpath(fp)
```

```
877         fpOut = fp[0:len(fp) - 3]
878         fpOut += subformatExtension
879 
880         musescoreRun = '"' + str(musescorePath) + '" "' + fp + '" -o "' + fpOut + '" -T 0 '
881         if 'dpi' in keywords:
882             musescoreRun += ' -r ' + str(keywords['dpi'])
883 
884         if common.runningUnderIPython():
885             musescoreRun += ' -r ' + str(defaults.ipythonImageDpi)
886 
887         storedStrErr = sys.stderr
888         fileLikeOpen = io.StringIO()
889         sys.stderr = fileLikeOpen
890         subprocess.run(musescoreRun)
891         fileLikeOpen.close()
892         sys.stderr = storedStrErr
893 
894         if subformatExtension == 'png':
895             return self.findPNGfpFromXMLfp(fpOut)
896         else:
897             return fpOut
898         # common.cropImageFromPath(fp)
899 
900     def writeDataStream(self, fp, dataBytes):  # pragma: no cover
901         if fp is None:
902             fp = self.getTemporaryFile()
903         else:
904             fp = common.cleanpath(fp)
905 
```

jupyter subConverters.py 5분 전

Logout

File Edit View Language Python

```
1  # -*- coding: utf-8 -*-
2  # ------------------------------------------------------------------------------
3  # Name:         converter/__init__.py
4  # Purpose:      Specific subconverters for formats music21 should handle
5  #
6  # Authors:      Michael Scott Cuthbert
7  #               Christopher Ariza
8  #
9  # Copyright:    Copyright © 2009-2015 Michael Scott Cuthbert and the music21 Project
10 # License:      LGPL or BSD, see license.txt
11 # ------------------------------------------------------------------------------
12 '''
13 Subconverters parse or display a single format.
14
15 Each subconverter should inherit from the base SubConverter object and have at least a
16 parseData method that sets self.stream.
17 '''
18 # ------------------------------------------------------------------------------
19 # Converters are associated classes; they are not subclasses, but most define a pareData() method,
20 # a parseFile() method, and a .stream attribute or property.
21 import base64
22 import io
23 import os
24 import pathlib
25 import sys
26 import unittest
27 import subprocess
28
29 from music21 import common
```

설정 완료

음악 데이터 확인

In [1]:
```
from music21 import converter
```

데이터 다운로드

J.S. 바흐 첼로 모음집 미디 데이터를 다음 주소에서 다운로드 받을 수 있습니다. 다운로드할 파일은 cs1-2all.mid 입니다.

http://www.jsbach.net/midi/midi_solo_cello.html

이 파일을 './data/cello' 폴더 안에 저장하세요.

악보 소프트웨어

모델이 생성한 음악을 보거나 들으려면 Musescore 프로그램이 필요합니다. 다음 주소에서 다운로드하세요.

https://musescore.org/en

Note: music21 패키지는 Musescore 2.x 버전과 호환됩니다. Older versions 링크를 눌러서 2.3.2 버전을 다운로드하세요.

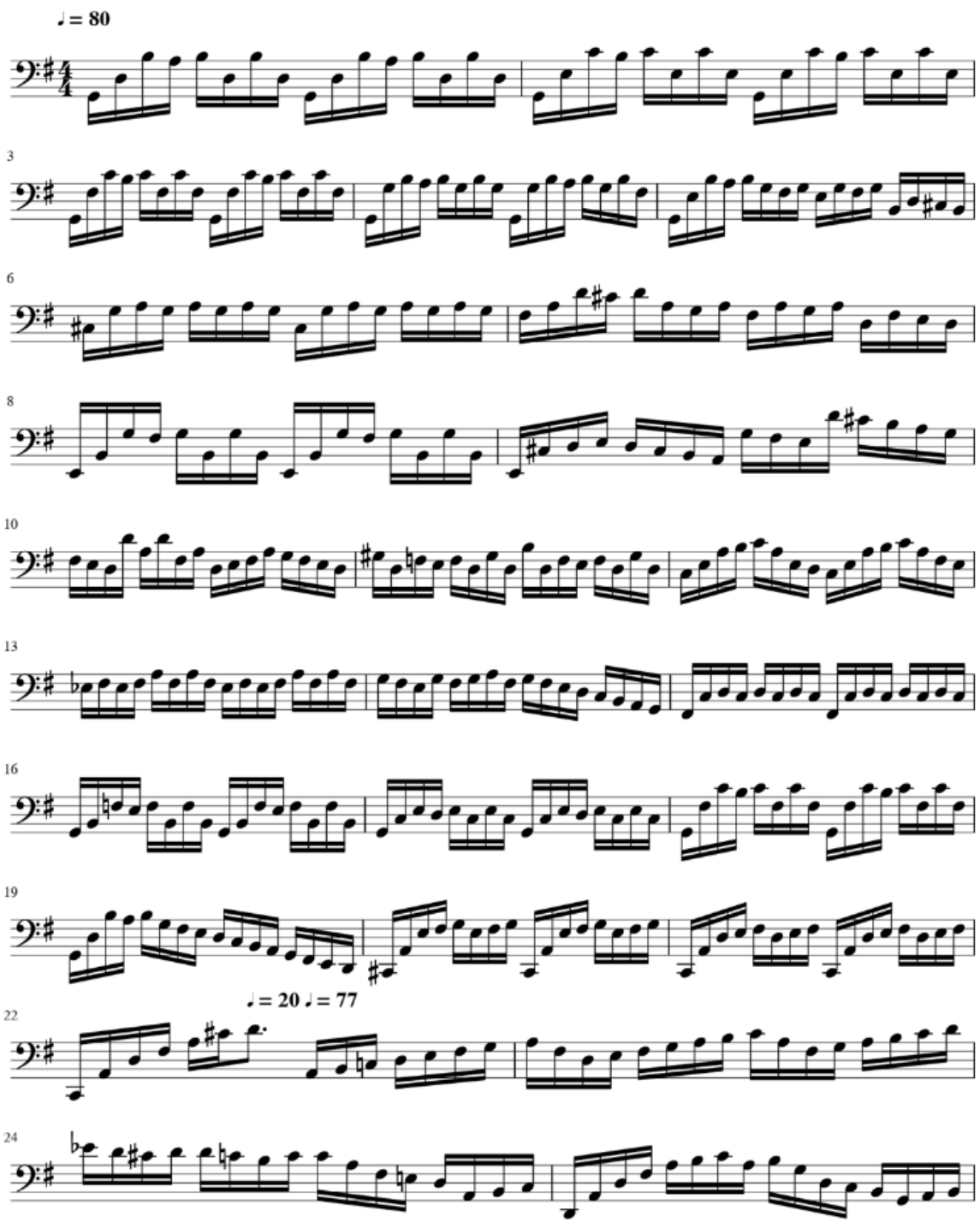
Note: 리눅스 버전의 Musescore는 지원이 불안정합니다. Windows나 macOS를 이용하세요. 이 노트북은 macOS에서 실행하였습니다.

데이터 확인

In [4]:
```
dataset_name = 'cello'
filename = 'cs1-1pre' # 유명한 바흐의 무반주 첼로 모음곡 1번 Prelude 입니다.
file = "./data/{}/{}.mid".format(dataset_name, filename)

original_score = converter.parse(file).chordify()
```

바흐의 무반주 첼로 모음곡 1번 (Prelude).

In [5]:
```
original_score.show()
```


♩ = 80
3
6
8
10
13
16
19
♩ = 20 ♩ = 77
22
24

In [6]:
```
original_score.show('text')
```

```
{0.0} <music21.instrument.Violoncello 'Violoncello'>
{0.0} <music21.tempo.MetronomeMark andantino Quarter=80.0>
{0.0} <music21.key.Key of G major>
{0.0} <music21.meter.TimeSignature 4/4>
{0.0} <music21.chord.Chord G2>
{0.25} <music21.chord.Chord D3>
{0.5} <music21.chord.Chord B3>
{0.75} <music21.chord.Chord A3>
{1.0} <music21.chord.Chord B3>
{1.25} <music21.chord.Chord D3>
{1.5} <music21.chord.Chord B3>
{1.75} <music21.chord.Chord D3>
{2.0} <music21.chord.Chord G2>
{2.25} <music21.chord.Chord D3>
{2.5} <music21.chord.Chord B3>
{2.75} <music21.chord.Chord A3>
{3.0} <music21.chord.Chord B3>
{3.25} <music21.chord.Chord D3>
{3.5} <music21.chord.Chord B3>
```

데이터 추출

In [7]:
```
from music21 import chord, note
```

In [8]:
```
notes = []
durations = []

for element in original_score.flat:

    if isinstance(element, chord.Chord):
        notes.append('.'.join(n.nameWithOctave for n in element.pitches))
        durations.append(element.duration.quarterLength)

    if isinstance(element, note.Note):
        if element.isRest:
            notes.append(str(element.name))
            durations.append(element.duration.quarterLength)
        else:
            notes.append(str(element.nameWithOctave))
            durations.append(element.duration.quarterLength)
```

In [9]:
```
print('₩nduration', 'pitch')
for n,d in zip(notes,durations):
    print(d, '₩t', n)
```

```
duration pitch
0.25     G2
0.25     D3
0.25     B3
0.25     A3
0.25     B3
0.25     D3
0.25     B3
0.25     D3
0.25     G2
0.25     D3
0.25     B3
0.25     A3
0.25     B3
0.25     D3
0.25     B3
0.25     D3
0.25     G2
```

작곡: 음악을 생성하는 모델을 훈련하기

In [1]:
```
import os
import pickle
import numpy
from music21 import note, chord

from keras.callbacks import ModelCheckpoint, EarlyStopping
from keras.utils import plot_model

from models.RNNAttention import get_distinct, create_lookups, prepare_sequences, get_music_list, create_network
```

```
Using TensorFlow backend.
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:526: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:527: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:528: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:529: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:530: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:535: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
```

파라미터 설정

In [2]:
```
# 실행 파라미터
section = 'compose'
run_id = '0006'
music_name = 'cello'

run_folder = 'run/{}/'.format(section)
run_folder += '_'.join([run_id, music_name])


store_folder = os.path.join(run_folder, 'store')
data_folder = os.path.join('data', music_name)

if not os.path.exists(run_folder):
    os.mkdir(run_folder)
    os.mkdir(os.path.join(run_folder, 'store'))
    os.mkdir(os.path.join(run_folder, 'output'))
    os.mkdir(os.path.join(run_folder, 'weights'))
    os.mkdir(os.path.join(run_folder, 'viz'))



mode = 'build' # 'load' #

# 데이터 파라미터
intervals = range(1)
seq_len = 32

# 하이퍼파라미터
embed_size = 100
rnn_units = 256
use_attention = True
```

악보 추출

In [3]:
```
if mode == 'build':

    music_list, parser = get_music_list(data_folder)
    print(len(music_list), 'files in total')

    notes = []
    durations = []

    for i, file in enumerate(music_list):
        print(i+1, "Parsing %s" % file)
        original_score = parser.parse(file).chordify()


        for interval in intervals:

            score = original_score.transpose(interval)

            notes.extend(['START'] * seq_len)
            durations.extend([0]* seq_len)

            for element in score.flat:

                if isinstance(element, note.Note):
                    if element.isRest:
                        notes.append(str(element.name))
                        durations.append(element.duration.quarterLength)
                    else:
                        notes.append(str(element.nameWithOctave))
                        durations.append(element.duration.quarterLength)

                if isinstance(element, chord.Chord):
                    notes.append('.'.join(n.nameWithOctave for n in element.pitches))
                    durations.append(element.duration.quarterLength)

    with open(os.path.join(store_folder, 'notes'), 'wb') as f:
        pickle.dump(notes, f) #['G2', 'D3', 'B3', 'A3', 'B3', 'D3', 'B3', 'D3', 'G2',...]
    with open(os.path.join(store_folder, 'durations'), 'wb') as f:
        pickle.dump(durations, f)
else:
    with open(os.path.join(store_folder, 'notes'), 'rb') as f:
        notes = pickle.load(f) #['G2', 'D3', 'B3', 'A3', 'B3', 'D3', 'B3', 'D3', 'G2',...]
    with open(os.path.join(store_folder, 'durations'), 'rb') as f:
        durations = pickle.load(f)
```

```
1 files in total
1 Parsing data/cello/cs1-1pre.mid
```

룩업 테이블 만들기

In [4]:
```
# 고유한 음표와 박자 얻어오기
note_names, n_notes = get_distinct(notes)
duration_names, n_durations = get_distinct(durations)
distincts = [note_names, n_notes, duration_names, n_durations]

with open(os.path.join(store_folder, 'distincts'), 'wb') as f:
    pickle.dump(distincts, f)

# 음표와 박자 룩업 딕셔너리 만들고 저장하기
note_to_int, int_to_note = create_lookups(note_names)
duration_to_int, int_to_duration = create_lookups(duration_names)
lookups = [note_to_int, int_to_note, duration_to_int, int_to_duration]

with open(os.path.join(store_folder, 'lookups'), 'wb') as f:
    pickle.dump(lookups, f)
```

In [5]:
```
print('₩nnote_to_int')
note_to_int
```

```
note_to_int
```

Out[5]:
```
{'A2': 0,
 'A3': 1,
 'B-3': 2,
 'B2': 3,
 'B3': 4,
 'C#2': 5,
 'C#3': 6,
 'C#4': 7,
 'C2': 8,
 'C3': 9,
 'C4': 10,
 'D2': 11,
 'D3': 12,
 'D4': 13,
 'E-3': 14,
 'E-4': 15,
 'E2': 16,
 'E3': 17,
 'E4': 18,
 'F#2': 19,
 'F#3': 20,
 'F#4': 21,
 'F3': 22,
 'F4': 23,
 'G#3': 24,
 'G2': 25,
 'G2.B3.G4': 26,
 'G3': 27,
 'G4': 28,
 'START': 29}
```

In [6]:
```
print('₩nduration_to_int')
duration_to_int
```

```
duration_to_int
```

Out[6]:
```
{0: 0, 0.25: 1, 0.5: 2, 0.75: 3, 4.0: 4}
```

신경망에 사용할 시퀀스 준비하기

In [7]:
```
network_input, network_output = prepare_sequences(notes, durations, lookups, distincts, seq_len)
```

In [8]:
```
print('pitch input')
print(network_input[0][0])
print('duration input')
print(network_input[1][0])
print('pitch output')
print(network_output[0][0])
print('duration output')
print(network_output[1][0])
```

```
pitch input
[29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29
 29 29 29 29 29 29 29 29]
duration input
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
pitch output
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 1. 0. 0. 0. 0.]
duration output
[0. 1. 0. 0. 0.]
```

신경망 만들기

In [9]:
```
model, att_model = create_network(n_notes, n_durations, embed_size, rnn_units, use_attention)
model.summary()
```

```
WARNING:tensorflow:From /home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/op_def_library.py:
263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, None)	0	
input_2 (InputLayer)	(None, None)	0	
embedding_1 (Embedding)	(None, None, 100)	3000	input_1[0][0]
embedding_2 (Embedding)	(None, None, 100)	500	input_2[0][0]
concatenate_1 (Concatenate)	(None, None, 200)	0	embedding_1[0][0] embedding_2[0][0]
lstm_1 (LSTM)	(None, None, 256)	467968	concatenate_1[0][0]
lstm_2 (LSTM)	(None, None, 256)	525312	lstm_1[0][0]
dense_1 (Dense)	(None, None, 1)	257	lstm_2[0][0]
reshape_1 (Reshape)	(None, None)	0	dense_1[0][0]
activation_1 (Activation)	(None, None)	0	reshape_1[0][0]
repeat_vector_1 (RepeatVector)	(None, 256, None)	0	activation_1[0][0]
permute_1 (Permute)	(None, None, 256)	0	repeat_vector_1[0][0]
multiply_1 (Multiply)	(None, None, 256)	0	lstm_2[0][0] permute_1[0][0]
lambda_1 (Lambda)	(None, 256)	0	multiply_1[0][0]
pitch (Dense)	(None, 30)	7710	lambda_1[0][0]
duration (Dense)	(None, 5)	1285	lambda_1[0][0]

```
Total params: 1,006,032
Trainable params: 1,006,032
Non-trainable params: 0
```

In [10]:
```
plot_model(model, to_file=os.path.join(run_folder ,'viz/model.png'), show_shapes = True, show_layer_names = True)
```

신경망 훈련하기

In [11]:
```
weights_folder = os.path.join(run_folder, 'weights')
# model.load_weights(os.path.join(weights_folder, "weights.h5"))
```

In [12]:
```
weights_folder = os.path.join(run_folder, 'weights')

checkpoint1 = ModelCheckpoint(
    os.path.join(weights_folder, "weights-improvement-{epoch:02d}-{loss:.4f}-bigger.h5"),
    monitor='loss',
    verbose=0,
    save_best_only=True,
    mode='min'
)

checkpoint2 = ModelCheckpoint(
    os.path.join(weights_folder, "weights.h5"),
    monitor='loss',
    verbose=0,
    save_best_only=True,
    mode='min'
)

early_stopping = EarlyStopping(
    monitor='loss'
    , restore_best_weights=True
    , patience = 10
)


callbacks_list = [
    checkpoint1
    , checkpoint2
    , early_stopping
 ]

model.save_weights(os.path.join(weights_folder, "weights.h5"))
model.fit(network_input, network_output
          , epochs=2000000, batch_size=32
          , validation_split = 0.2
          , callbacks=callbacks_list
          , shuffle=True
         )
```

```
601 - val_pitch_loss: 6.4970 - val_duration_loss: 0.1231
Epoch 128/2000000
523/523 [==============================] - 2s 3ms/step - loss: 0.1407 - pitch_loss: 0.1406 - duration_loss: 8.4723e-05 - val_loss: 6.6
166 - val_pitch_loss: 6.4934 - val_duration_loss: 0.1232
Epoch 129/2000000
523/523 [==============================] - 2s 3ms/step - loss: 0.1413 - pitch_loss: 0.1412 - duration_loss: 4.5041e-05 - val_loss: 6.7
771 - val_pitch_loss: 6.6540 - val_duration_loss: 0.1231
Epoch 130/2000000
523/523 [==============================] - 2s 3ms/step - loss: 0.2112 - pitch_loss: 0.2111 - duration_loss: 8.3203e-05 - val_loss: 6.9
468 - val_pitch_loss: 6.8238 - val_duration_loss: 0.1230
Epoch 131/2000000
523/523 [==============================] - 2s 3ms/step - loss: 0.1657 - pitch_loss: 0.1656 - duration_loss: 1.1927e-04 - val_loss: 6.8
294 - val_pitch_loss: 6.7063 - val_duration_loss: 0.1231
Epoch 132/2000000
523/523 [==============================] - 2s 3ms/step - loss: 0.1305 - pitch_loss: 0.1304 - duration_loss: 4.2781e-05 - val_loss: 6.9
117 - val_pitch_loss: 6.7887 - val_duration_loss: 0.1231
Epoch 133/2000000
523/523 [==============================] - 2s 3ms/step - loss: 0.1529 - pitch_loss: 0.1529 - duration_loss: 6.0797e-05 - val_loss: 6.7
041 - val_pitch_loss: 6.5809 - val_duration_loss: 0.1232
```

Out[12]: <keras.callbacks.History at 0x7fe304551ef0>

작곡: 음악을 생성하는 모델을 예측하기

라이브러리 임포트

In [1]:
```
import pickle as pkl
import time
import os
import numpy as np
import sys
from music21 import instrument, note, stream, chord, duration
from models.RNNAttention import create_network, sample_with_temp

import matplotlib.pyplot as plt
```

```
Using TensorFlow backend.
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:526: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:527: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:528: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:529: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:530: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:535: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
```

파라미터

In [2]:
```
# 실행 파라미터
section = 'compose'
run_id = '0006'
music_name = 'cello'
run_folder = 'run/{}/'.format(section)
run_folder += '_'.join([run_id, music_name])

# 하이퍼파라미터
embed_size = 100
rnn_units = 256
use_attention = True
```

룩업 테이블 적재

In [3]:
```
store_folder = os.path.join(run_folder, 'store')

with open(os.path.join(store_folder, 'distincts'), 'rb') as filepath:
    distincts = pkl.load(filepath)
    note_names, n_notes, duration_names, n_durations = distincts

with open(os.path.join(store_folder, 'lookups'), 'rb') as filepath:
    lookups = pkl.load(filepath)
    note_to_int, int_to_note, duration_to_int, int_to_duration = lookups
```

모델 만들기

In [4]:
```
weights_folder = os.path.join(run_folder, 'weights')
weights_file = 'weights.h5'

model, att_model = create_network(n_notes, n_durations, embed_size, rnn_units, use_attention)

# 각 노드에 가중치 적재하기
weight_source = os.path.join(weights_folder,weights_file)
model.load_weights(weight_source)
model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, None)]       0
__________________________________________________________________________________________________
input_2 (InputLayer)            [(None, None)]       0
__________________________________________________________________________________________________
embedding (Embedding)           (None, None, 100)    3000        input_1[0][0]
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, None, 100)    500         input_2[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, None, 200)    0           embedding[0][0]
                                                                 embedding_1[0][0]
__________________________________________________________________________________________________
lstm (LSTM)                     (None, None, 256)    467968      concatenate[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, None, 256)    525312      lstm[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, None, 1)      257         lstm_1[0][0]
__________________________________________________________________________________________________
reshape (Reshape)               (None, None)         0           dense[0][0]
__________________________________________________________________________________________________
activation (Activation)         (None, None)         0           reshape[0][0]
__________________________________________________________________________________________________
repeat_vector (RepeatVector)    (None, 256, None)    0           activation[0][0]
__________________________________________________________________________________________________
permute (Permute)               (None, None, 256)    0           repeat_vector[0][0]
__________________________________________________________________________________________________
multiply (Multiply)             (None, None, 256)    0           lstm_1[0][0]
                                                                 permute[0][0]
__________________________________________________________________________________________________
lambda (Lambda)                 (None, 256)          0           multiply[0][0]
__________________________________________________________________________________________________
pitch (Dense)                   (None, 30)           7710        lambda[0][0]
__________________________________________________________________________________________________
duration (Dense)                (None, 5)            1285        lambda[0][0]
==================================================================================================
Total params: 1,006,032
Trainable params: 1,006,032
Non-trainable params: 0
__________________________________________________________________________________________________
```

자신만의 악절 만들기

In [5]:
```
# 예측용 파라미터
notes_temp=0.5
duration_temp = 0.5
max_extra_notes = 50
max_seq_len = 32
seq_len = 32

# notes = ['START', 'D3', 'D3', 'E3', 'D3', 'G3', 'F#3','D3', 'D3', 'E3', 'D3', 'G3', 'F#3','D3', 'D3', 'E3', 'D3', 'G3', 'F#3','D3', 'D3
# durations = [0, 0.75, 0.25, 1, 1, 1, 2, 0.75, 0.25, 1, 1, 1, 2, 0.75, 0.25, 1, 1, 1, 2, 0.75, 0.25, 1, 1, 1, 2]


# notes = ['START', 'F#3', 'G#3', 'F#3', 'E3', 'F#3', 'G#3', 'F#3', 'E3', 'F#3', 'G#3', 'F#3', 'E3','F#3', 'G#3', 'F#3', 'E3', 'F#3', 'G#
# durations = [0, 0.75, 0.25, 1, 1, 1, 2, 0.75, 0.25, 1, 1, 1, 2, 0.75, 0.25, 1, 1, 1, 2, 0.75, 0.25, 1, 1, 1, 2]


notes = ['START']
durations = [0]

if seq_len is not None:
    notes = ['START'] * (seq_len - len(notes)) + notes
    durations = [0] * (seq_len - len(durations)) + durations


sequence_length = len(notes)
```

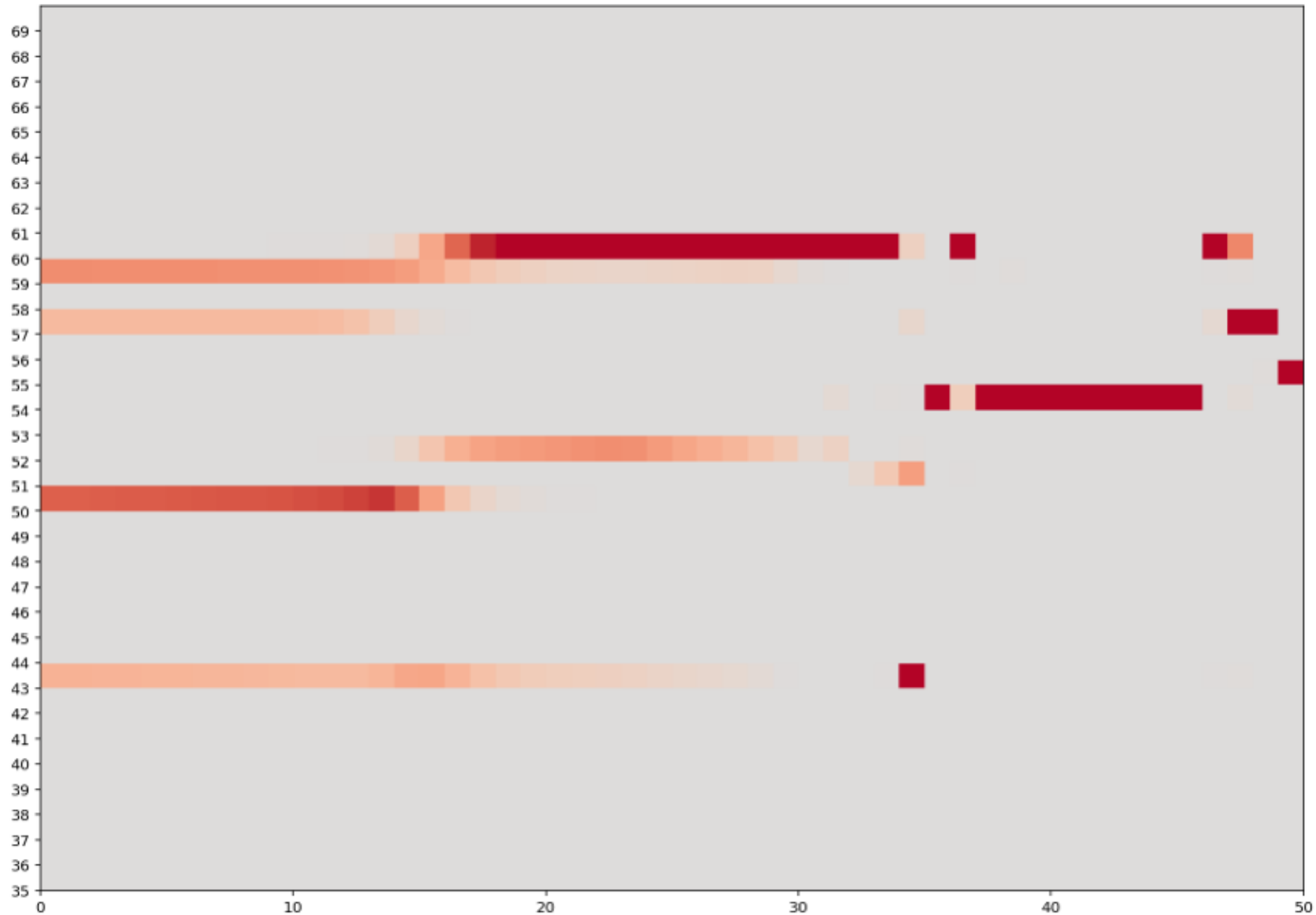
악보 시퀀스를 기반으로 신경망에서 악보 생성하기

In [6]:
```
prediction_output = []
notes_input_sequence = []
durations_input_sequence = []

overall_preds = []

for n, d in zip(notes,durations):
    note_int = note_to_int[n]
    duration_int = duration_to_int[d]

    notes_input_sequence.append(note_int)
    durations_input_sequence.append(duration_int)

    prediction_output.append([n, d])

    if n != 'START':
        midi_note = note.Note(n)

        new_note = np.zeros(128)
        new_note[midi_note.pitch.midi] = 1
        overall_preds.append(new_note)


att_matrix = np.zeros(shape = (max_extra_notes+sequence_length, max_extra_notes))

for note_index in range(max_extra_notes):

    prediction_input = [
        np.array([notes_input_sequence])
        , np.array([durations_input_sequence])
       ]

    notes_prediction, durations_prediction = model.predict(prediction_input, verbose=0)
    if use_attention:
        att_prediction = att_model.predict(prediction_input, verbose=0)[0]
        att_matrix[(note_index-len(att_prediction)+sequence_length):(note_index+sequence_length), note_index] = att_prediction

    new_note = np.zeros(128)

    for idx, n_i in enumerate(notes_prediction[0]):
        try:
            note_name = int_to_note[idx]
            midi_note = note.Note(note_name)
            new_note[midi_note.pitch.midi] = n_i

        except:
            pass

    overall_preds.append(new_note)


    i1 = sample_with_temp(notes_prediction[0], notes_temp)
    i2 = sample_with_temp(durations_prediction[0], duration_temp)


    note_result = int_to_note[i1]
    duration_result = int_to_duration[i2]

    prediction_output.append([note_result, duration_result])

    notes_input_sequence.append(i1)
    durations_input_sequence.append(i2)

    if len(notes_input_sequence) > max_seq_len:
        notes_input_sequence = notes_input_sequence[1:]
        durations_input_sequence = durations_input_sequence[1:]

#     print(note_result)
#     print(duration_result)

    if note_result == 'START':
        break

overall_preds = np.transpose(np.array(overall_preds))
print('Generated sequence of {} notes'.format(len(prediction_output)))
```

```
Generated sequence of 82 notes
```

In [7]:
```
fig, ax = plt.subplots(figsize=(15,15))
ax.set_yticks([int(j) for j in range(35,70)])

plt.imshow(overall_preds[35:70,:], origin="lower", cmap='coolwarm', vmin = -0.5, vmax = 0.5, extent=[0, max_extra_notes, 35,70])
```

Out[7]:
```
<matplotlib.image.AxesImage at 0x7f2ca81784a8>
```



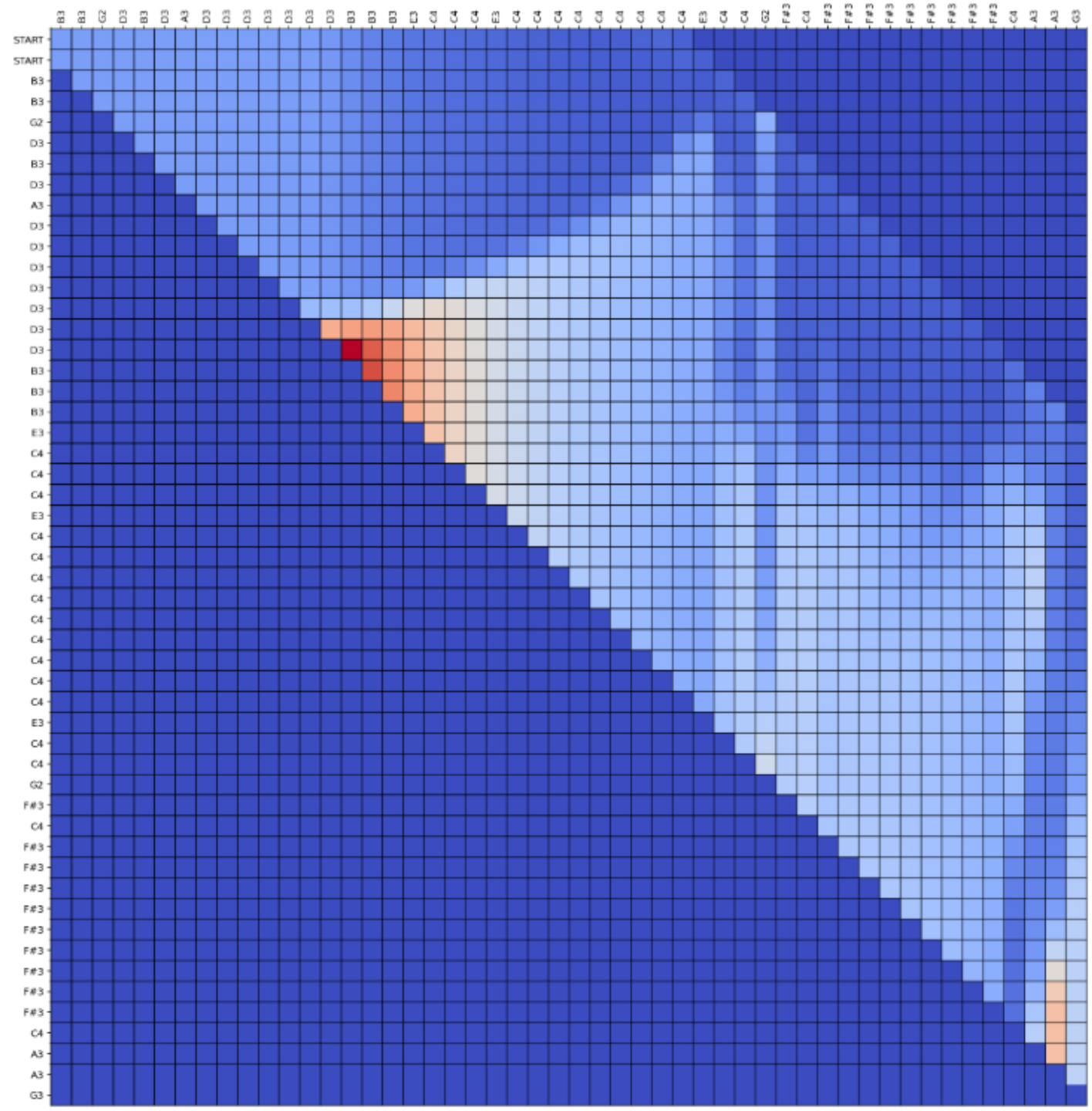
예측 출력을 악보로 변환하고 악보에서 미디 파일 만들기

In [8]:
```
output_folder = os.path.join(run_folder, 'output')

midi_stream = stream.Stream()

# 모델이 생성한 값을 기반으로 악보와 화음 객체 만들기
for pattern in prediction_output:
    note_pattern, duration_pattern = pattern
    # 패턴이 화음일 경우
    if ('.' in note_pattern):
        notes_in_chord = note_pattern.split('.')
        chord_notes = []
        for current_note in notes_in_chord:
            new_note = note.Note(current_note)
            new_note.duration = duration.Duration(duration_pattern)
            new_note.storedInstrument = instrument.Violoncello()
            chord_notes.append(new_note)
        new_chord = chord.Chord(chord_notes)
        midi_stream.append(new_chord)
    elif note_pattern == 'rest':
    # 패턴이 쉼표일 경우
        new_note = note.Rest()
        new_note.duration = duration.Duration(duration_pattern)
        new_note.storedInstrument = instrument.Violoncello()
        midi_stream.append(new_note)
    elif note_pattern != 'START':
    # 패턴이 하나의 음표일 경우
        new_note = note.Note(note_pattern)
        new_note.duration = duration.Duration(duration_pattern)
        new_note.storedInstrument = instrument.Violoncello()
        midi_stream.append(new_note)



midi_stream = midi_stream.chordify()
timestr = time.strftime("%Y%m%d-%H%M%S")
midi_stream.write('midi', fp=os.path.join(output_folder, 'output-' + timestr + '.mid'))
```

Out[8]:
```
'run/compose/0006_cello/output/output-20200728-180557.mid'
```

어텐션 그래프

In [9]:
```
if use_attention:
    fig, ax = plt.subplots(figsize=(20,20))

    im = ax.imshow(att_matrix[(seq_len-2):,], cmap='coolwarm', interpolation='nearest')




    # Minor ticks
    ax.set_xticks(np.arange(-.5, len(prediction_output)- seq_len, 1), minor=True);
    ax.set_yticks(np.arange(-.5, len(prediction_output)- seq_len, 1), minor=True);

    # Gridlines based on minor ticks
    ax.grid(which='minor', color='black', linestyle='-', linewidth=1)




    # We want to show all ticks...
    ax.set_xticks(np.arange(len(prediction_output) - seq_len))
    ax.set_yticks(np.arange(len(prediction_output)- seq_len+2))
    # ... and label them with the respective list entries
    ax.set_xticklabels([n[0] for n in prediction_output[(seq_len):]])
    ax.set_yticklabels([n[0] for n in prediction_output[(seq_len - 2):]])

    # ax.grid(color='black', linestyle='-', linewidth=1)

    ax.xaxis.tick_top()



    plt.setp(ax.get_xticklabels(), rotation=90, ha="left", va = "center",
             rotation_mode="anchor")

    plt.show()
```


B3 B3 G2 D3 B3 D3 A3 D3 D3 D3 D3 D3 D3 B3 B3 E3 C4 C4 C4 E3 C4 C4 C4 C4 C4 C4 C4 E3 C4 C4 G2 F#3 C4 F#3 F#3 F#3 F#3 F#3 F#3 F#3 C4 A3 A3 G3
START START B3 B3 G2 D3 B3 D3 A3 D3 D3 D3 D3 D3 D3 D3 D3 B3 B3 B3 E3 C4 C4 C4 E3 C4 C4 C4 C4 C4 C4 C4 C4 C4 E3 C4 C4 G2 F#3 C4 F#3 F#3 F#3 F#3 F#3 F#3 F#3 F#3 C4 A3 A3 G3

MuseGAN 훈련

라이브러리 임포트

In [1]:
```
import os
import matplotlib.pyplot as plt
import numpy as np
import types

from models.MuseGAN import MuseGAN
from utils.loaders import load_music


from music21 import midi
from music21 import note, stream, duration
```

```
Using TensorFlow backend.
```

In [2]:
```
# run params
SECTION = 'compose'
RUN_ID = '0017'
DATA_NAME = 'chorales'
FILENAME = 'Jsb16thSeparated.npz'
RUN_FOLDER = 'run/{}/'.format(SECTION)
RUN_FOLDER += '_'.join([RUN_ID, DATA_NAME])



if not os.path.exists(RUN_FOLDER):
    os.mkdir(RUN_FOLDER)
    os.mkdir(os.path.join(RUN_FOLDER, 'viz'))
    os.mkdir(os.path.join(RUN_FOLDER, 'images'))
    os.mkdir(os.path.join(RUN_FOLDER, 'weights'))
    os.mkdir(os.path.join(RUN_FOLDER, 'samples'))

mode =  'build' # ' 'load' #
```

데이터 적재

- Jsb16thSeparated.npz 파일이 없다면 아래의 주소에서 다운로드 한다.
  https://github.com/czhuang/JSB-Chorales-dataset/blob/master/Jsb16thSeparated.npz

- pickle 오류가 난다면 numpy==1.16.1 버전으로 설치한다.

In [3]:
```
BATCH_SIZE = 64
n_bars = 2
n_steps_per_bar = 16
n_pitches = 84
n_tracks = 4

data_binary, data_ints, raw_data = load_music(DATA_NAME, FILENAME, n_bars, n_steps_per_bar)
data_binary = np.squeeze(data_binary)
```

모델 만들기

In [4]:
```
gan = MuseGAN(input_dim = data_binary.shape[1:]
        , critic_learning_rate = 0.001
        , generator_learning_rate = 0.001
        , optimiser = 'adam'
        , grad_weight = 10
        , z_dim = 32
        , batch_size = BATCH_SIZE
        , n_tracks = n_tracks
        , n_bars = n_bars
        , n_steps_per_bar = n_steps_per_bar
        , n_pitches = n_pitches
        )

if mode == 'build':
    gan.save(RUN_FOLDER)
else:
    gan.load_weights(RUN_FOLDER)
```

```
WARNING:tensorflow:From /home/luxmaris16/anaconda3/envs/testGPU/lib/python3.6/site-packages/tensorflow/python/framework/op_def_library.
py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
```

In [5]:
```
gan.chords_tempNetwork.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
temporal_input (InputLayer)  (None, 32)                0
_________________________________________________________________
reshape_1 (Reshape)          (None, 1, 1, 32)          0
_________________________________________________________________
conv2d_transpose_1 (Conv2DTr (None, 2, 1, 1024)        66560
_________________________________________________________________
batch_normalization_1 (Batch (None, 2, 1, 1024)        4096
_________________________________________________________________
activation_1 (Activation)    (None, 2, 1, 1024)        0
_________________________________________________________________
conv2d_transpose_2 (Conv2DTr (None, 2, 1, 32)          32800
_________________________________________________________________
batch_normalization_2 (Batch (None, 2, 1, 32)          128
_________________________________________________________________
activation_2 (Activation)    (None, 2, 1, 32)          0
_________________________________________________________________
reshape_2 (Reshape)          (None, 2, 32)             0
=================================================================
Total params: 103,584
Trainable params: 101,472
Non-trainable params: 2,112
_________________________________________________________________
```

In [6]:
```
gan.barGen[0].summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bar_generator_input (InputLa (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 1024)              132096
_________________________________________________________________
batch_normalization_11 (Batc (None, 1024)              4096
_________________________________________________________________
activation_11 (Activation)   (None, 1024)              0
_________________________________________________________________
reshape_11 (Reshape)         (None, 2, 1, 512)         0
_________________________________________________________________
conv2d_transpose_11 (Conv2DT (None, 4, 1, 512)         524800
_________________________________________________________________
batch_normalization_12 (Batc (None, 4, 1, 512)         2048
_________________________________________________________________
activation_12 (Activation)   (None, 4, 1, 512)         0
_________________________________________________________________
conv2d_transpose_12 (Conv2DT (None, 8, 1, 256)         262400
_________________________________________________________________
batch_normalization_13 (Batc (None, 8, 1, 256)         1024
_________________________________________________________________
activation_13 (Activation)   (None, 8, 1, 256)         0
_________________________________________________________________
conv2d_transpose_13 (Conv2DT (None, 16, 1, 256)        131328
_________________________________________________________________
batch_normalization_14 (Batc (None, 16, 1, 256)        1024
_________________________________________________________________
activation_14 (Activation)   (None, 16, 1, 256)        0
_________________________________________________________________
conv2d_transpose_14 (Conv2DT (None, 16, 7, 256)        459008
_________________________________________________________________
batch_normalization_15 (Batc (None, 16, 7, 256)        1024
_________________________________________________________________
activation_15 (Activation)   (None, 16, 7, 256)        0
_________________________________________________________________
conv2d_transpose_15 (Conv2DT (None, 16, 84, 1)         3073
_________________________________________________________________
reshape_12 (Reshape)         (None, 1, 16, 84, 1)      0
=================================================================
Total params: 1,521,921
Trainable params: 1,517,313
Non-trainable params: 4,608
_________________________________________________________________
```

In [7]:
```
gan.generator.summary()
```

```
                                                                 total_input_bar_1_track_3[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 1, 16, 84, 4) 0           model_7[1][0]
                                                                 model_8[1][0]
                                                                 model_9[1][0]
                                                                 model_10[1][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 1, 16, 84, 4) 0           model_7[2][0]
                                                                 model_8[2][0]
                                                                 model_9[2][0]
                                                                 model_10[2][0]
__________________________________________________________________________________________________
concat_bars (Concatenate)       (None, 2, 16, 84, 4) 0           concatenate_1[0][0]
                                                                 concatenate_2[0][0]
==================================================================================================
Total params: 6,605,604
Trainable params: 6,576,612
Non-trainable params: 28,992
__________________________________________________________________________________________________
```

In [8]:
```
gan.critic.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
critic_input (InputLayer)    (None, 2, 16, 84, 4)      0
_________________________________________________________________
conv3d_1 (Conv3D)            (None, 1, 16, 84, 128)    1152
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 1, 16, 84, 128)    0
_________________________________________________________________
conv3d_2 (Conv3D)            (None, 1, 16, 84, 128)    16512
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 1, 16, 84, 128)    0
_________________________________________________________________
conv3d_3 (Conv3D)            (None, 1, 16, 7, 128)     196736
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 1, 16, 7, 128)     0
_________________________________________________________________
conv3d_4 (Conv3D)            (None, 1, 16, 1, 128)     114816
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 1, 16, 1, 128)     0
_________________________________________________________________
conv3d_5 (Conv3D)            (None, 1, 8, 1, 128)      32896
_________________________________________________________________
leaky_re_lu_5 (LeakyReLU)    (None, 1, 8, 1, 128)      0
_________________________________________________________________
conv3d_6 (Conv3D)            (None, 1, 4, 1, 128)      32896
_________________________________________________________________
leaky_re_lu_6 (LeakyReLU)    (None, 1, 4, 1, 128)      0
_________________________________________________________________
conv3d_7 (Conv3D)            (None, 1, 2, 1, 256)      131328
_________________________________________________________________
leaky_re_lu_7 (LeakyReLU)    (None, 1, 2, 1, 256)      0
_________________________________________________________________
conv3d_8 (Conv3D)            (None, 1, 1, 1, 512)      393728
_________________________________________________________________
leaky_re_lu_8 (LeakyReLU)    (None, 1, 1, 1, 512)      0
_________________________________________________________________
flatten_1 (Flatten)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              525312
_________________________________________________________________
leaky_re_lu_9 (LeakyReLU)    (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 1025
=================================================================
Total params: 1,446,401
Trainable params: 1,446,401
Non-trainable params: 0
_________________________________________________________________
```

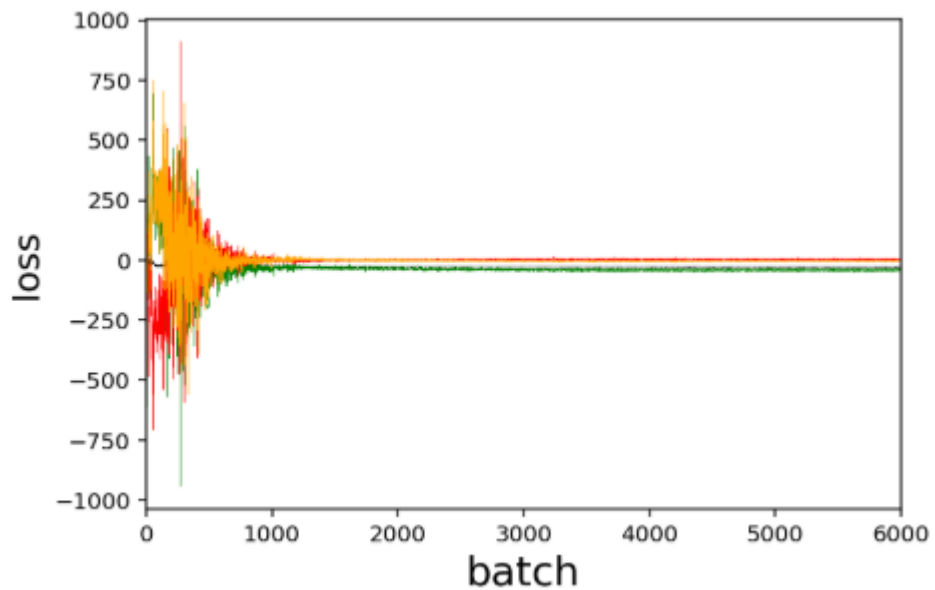
모델 훈련

In [9]:
```
EPOCHS = 6000
PRINT_EVERY_N_BATCHES = 10

gan.epoch = 0
```

In [10]:
```
gan.train(
    data_binary
    , batch_size = BATCH_SIZE
    , epochs = EPOCHS
    , run_folder = RUN_FOLDER
    , print_every_n_batches = PRINT_EVERY_N_BATCHES
)
```

```
5980 (5, 1) [D loss: (-27.5)(R -39.8, F -0.1, G 0.8)] [G loss: -3.1]
5981 (5, 1) [D loss: (-28.6)(R -39.7, F 2.6, G 0.9)] [G loss: -2.0]
5982 (5, 1) [D loss: (-27.8)(R -38.2, F 2.0, G 0.8)] [G loss: -2.1]
5983 (5, 1) [D loss: (-28.7)(R -37.2, F 0.7, G 0.8)] [G loss: -3.1]
5984 (5, 1) [D loss: (-27.7)(R -43.9, F 3.4, G 1.3)] [G loss: -4.8]
5985 (5, 1) [D loss: (-27.8)(R -34.6, F 0.5, G 0.6)] [G loss: -5.2]
5986 (5, 1) [D loss: (-28.7)(R -39.5, F 2.0, G 0.9)] [G loss: -3.2]
5987 (5, 1) [D loss: (-28.5)(R -43.6, F 2.0, G 1.3)] [G loss: -2.2]
5988 (5, 1) [D loss: (-29.7)(R -38.2, F 1.0, G 0.8)] [G loss: -3.3]
5989 (5, 1) [D loss: (-29.0)(R -41.0, F 2.1, G 1.0)] [G loss: -1.5]
5990 (5, 1) [D loss: (-28.8)(R -39.9, F 2.0, G 0.9)] [G loss: -2.0]
5991 (5, 1) [D loss: (-29.7)(R -40.5, F 2.9, G 0.8)] [G loss: -4.5]
5992 (5, 1) [D loss: (-29.3)(R -42.4, F 3.1, G 1.0)] [G loss: -3.5]
5993 (5, 1) [D loss: (-29.7)(R -42.3, F 3.1, G 1.0)] [G loss: -2.3]
5994 (5, 1) [D loss: (-28.9)(R -38.5, F 1.7, G 0.8)] [G loss: -2.8]
5995 (5, 1) [D loss: (-30.2)(R -43.4, F 4.1, G 0.9)] [G loss: -1.4]
5996 (5, 1) [D loss: (-27.1)(R -36.0, F -0.5, G 0.9)] [G loss: -3.8]
5997 (5, 1) [D loss: (-29.0)(R -41.6, F 3.0, G 1.0)] [G loss: -2.0]
5998 (5, 1) [D loss: (-28.0)(R -47.2, F 4.5, G 1.5)] [G loss: -4.9]
5999 (5, 1) [D loss: (-28.1)(R -43.7, F 4.0, G 1.2)] [G loss: -1.2]
```

In [11]:
```
fig = plt.figure()
plt.plot([x[0] for x in gan.d_losses], color='black', linewidth=0.25)

plt.plot([x[1] for x in gan.d_losses], color='green', linewidth=0.25)
plt.plot([x[2] for x in gan.d_losses], color='red', linewidth=0.25)
plt.plot(gan.g_losses, color='orange', linewidth=0.25)

plt.xlabel('batch', fontsize=18)
plt.ylabel('loss', fontsize=16)

plt.xlim(0, len(gan.d_losses))
# plt.ylim(0, 2)

plt.show()
```



MuseGAN 분석

In [1]:
```
import os
import matplotlib.pyplot as plt
import numpy as np

from music21 import *
from music21 import midi
from music21 import note, stream, duration
from music21 import converter

#from models.MuseGAN_old import MuseGAN
from models.MuseGAN import MuseGAN

from utils.loaders import load_music

from keras.models import load_model
```

```
Using TensorFlow backend.
```

In [2]:
```
# get environment
env = environment.Environment()

# set path
env['musicxmlPath'] = 'C:/Users/User/anaconda3/envs/testGAN/lib/site-packages/music21/musicxml'
env['musescoreDirectPNGPath'] = 'C:/Users/User/bin/MuseScore.exe'
```

In [3]:
```
# run params
SECTION = 'compose'
RUN_ID = '0017'
DATA_NAME = 'chorales'
FILENAME = 'Jsb16thSeparated.npz'
RUN_FOLDER = 'run/{}/'.format(SECTION)
RUN_FOLDER += '_'.join([RUN_ID, DATA_NAME])
```

데이터 적재

- Jsb16thSeparated.npz 파일이 없다면 아래의 주소에서 받을 수 있습니다.
  https://github.com/czhuang/JSB-Chorales-dataset/blob/master/Jsb16thSeparated.npz

- ValueError: Object arrays cannot be loaded when allow_pickle=False 가 발생하면 아래의 버전으로 바꾸어 줍니다.
  pip install numpy==1.16.1

In [4]:
```
BATCH_SIZE = 64
n_bars = 2
n_steps_per_bar = 16
n_pitches = 84
n_tracks = 4

data_binary, data_ints, raw_data = load_music(DATA_NAME, FILENAME, n_bars, n_steps_per_bar)
# data_binary = np.squeeze(data_binary)
```

In [5]:
```
gan = MuseGAN(input_dim = data_binary.shape[1:]
        , critic_learning_rate = 0.001
        , generator_learning_rate = 0.001
        , optimiser = 'adam'
        , grad_weight = 10
        , z_dim = 32
        , batch_size = BATCH_SIZE
        , n_tracks = n_tracks
        , n_bars = n_bars
        , n_steps_per_bar = n_steps_per_bar
        , n_pitches = n_pitches
        )
```

```
WARNING:tensorflow:From C:₩Users₩User₩anaconda3₩envs₩testGAN₩lib₩site-packages₩tensorflow₩python₩framework₩op_def_library.py:263: colocate_
with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
```

In [5]:
```
gan.load_weights(RUN_FOLDER, None)
```

In [6]:
```
gan.generator.summary()
```

```
model_10 (Model)                (None, 1, 16, 84, 4) 1321321     total_input_bar_0_track_3[0][0]
                                                                 total_input_bar_1_track_3[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 1, 16, 84, 4) 0           model_7[1][0]
                                                                 model_8[1][0]
                                                                 model_9[1][0]
                                                                 model_10[1][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 1, 16, 84, 4) 0           model_7[2][0]
                                                                 model_8[2][0]
                                                                 model_9[2][0]
                                                                 model_10[2][0]
__________________________________________________________________________________________________
concat_bars (Concatenate)       (None, 2, 16, 84, 4) 0           concatenate_1[0][0]
                                                                 concatenate_2[0][0]
==================================================================================================
Total params: 6,605,604
Trainable params: 6,576,612
Non-trainable params: 28,992
__________________________________________________________________________________________________
```

In [7]:
```
gan.critic.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
critic_input (InputLayer)    (None, 2, 16, 84, 4)      0
_________________________________________________________________
conv3d_1 (Conv3D)            (None, 1, 16, 84, 128)    1152
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 1, 16, 84, 128)    0
_________________________________________________________________
conv3d_2 (Conv3D)            (None, 1, 16, 84, 128)    16512
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 1, 16, 84, 128)    0
_________________________________________________________________
conv3d_3 (Conv3D)            (None, 1, 16, 7, 128)     196736
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 1, 16, 7, 128)     0
_________________________________________________________________
conv3d_4 (Conv3D)            (None, 1, 16, 1, 128)     114816
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 1, 16, 1, 128)     0
_________________________________________________________________
conv3d_5 (Conv3D)            (None, 1, 8, 1, 128)      32896
_________________________________________________________________
leaky_re_lu_5 (LeakyReLU)    (None, 1, 8, 1, 128)      0
_________________________________________________________________
conv3d_6 (Conv3D)            (None, 1, 4, 1, 128)      32896
_________________________________________________________________
leaky_re_lu_6 (LeakyReLU)    (None, 1, 4, 1, 128)      0
_________________________________________________________________
conv3d_7 (Conv3D)            (None, 1, 2, 1, 256)      131328
_________________________________________________________________
leaky_re_lu_7 (LeakyReLU)    (None, 1, 2, 1, 256)      0
_________________________________________________________________
conv3d_8 (Conv3D)            (None, 1, 1, 1, 512)      393728
_________________________________________________________________
leaky_re_lu_8 (LeakyReLU)    (None, 1, 1, 1, 512)      0
_________________________________________________________________
flatten_1 (Flatten)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              525312
_________________________________________________________________
leaky_re_lu_9 (LeakyReLU)    (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 1025
=================================================================
Total params: 1,446,401
Trainable params: 1,446,401
Non-trainable params: 0
_________________________________________________________________
```

샘플 악보 보기

In [9]:
```
chords_noise = np.random.normal(0, 1, (1, gan.z_dim))
style_noise = np.random.normal(0, 1, (1, gan.z_dim))
melody_noise = np.random.normal(0, 1, (1, gan.n_tracks, gan.z_dim))
groove_noise = np.random.normal(0, 1, (1, gan.n_tracks, gan.z_dim))
```

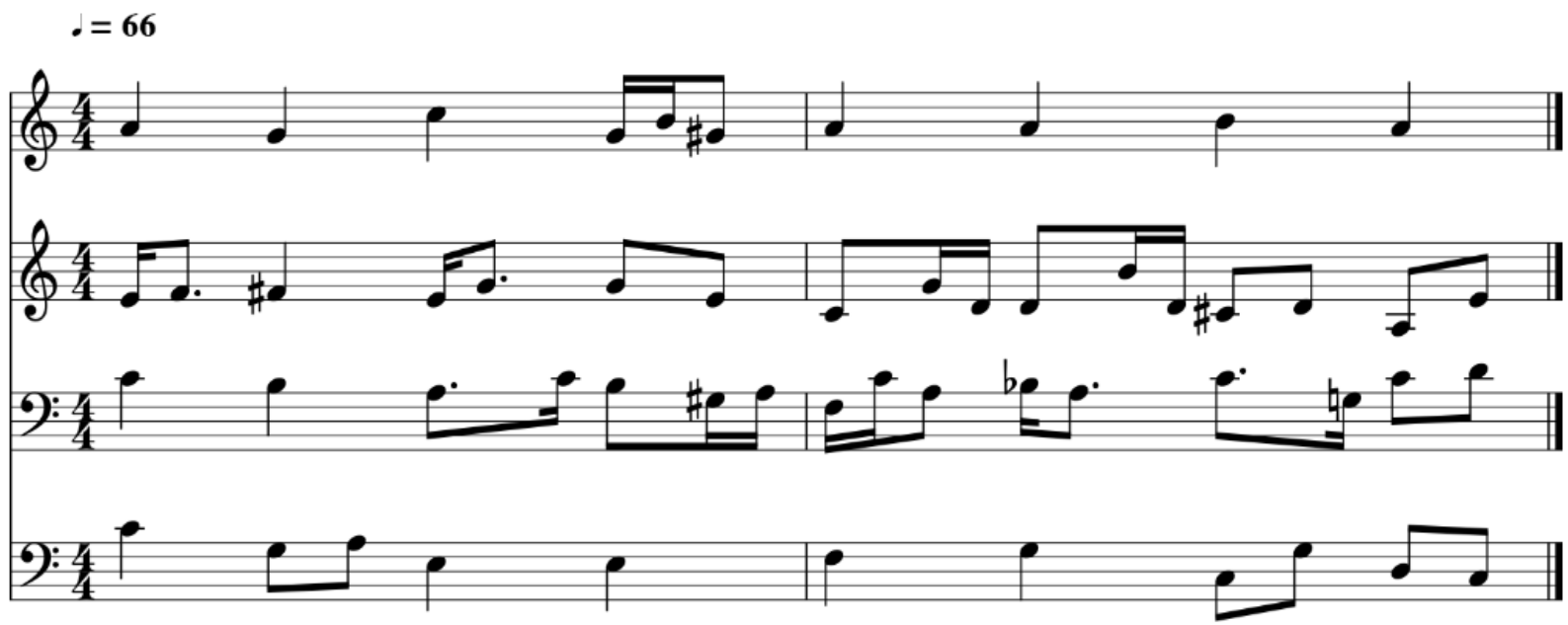
In [10]:
```
gen_scores = gan.generator.predict([chords_noise, style_noise, melody_noise, groove_noise])
```

In [11]:
```
np.argmax(gen_scores[0,0,0:4,:,3], axis = 1)
```

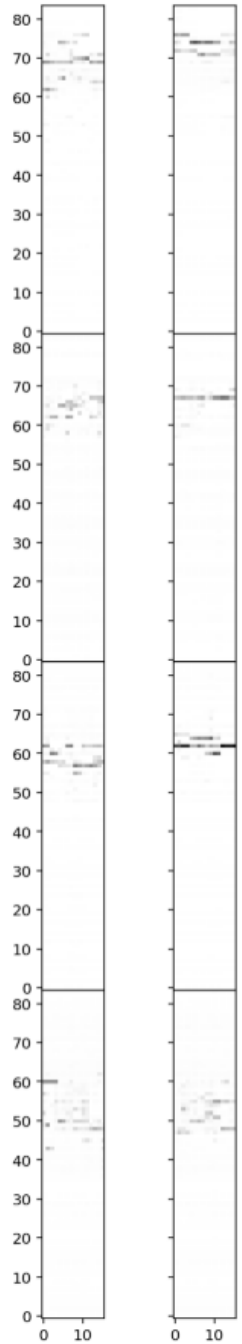
Out[11]:
```
array([57, 57, 57, 57], dtype=int64)
```

In [12]:
```
gen_scores[0,0,0:4,60,3] = 0.02347812
```

In [13]:
```
filename = 'example'
gan.notes_to_midi(RUN_FOLDER, gen_scores, filename)
gen_score = converter.parse(os.path.join(RUN_FOLDER, 'samples/{}.midi'.format(filename)))
gen_score.show()
```



In [13]:
```
gan.draw_score(gen_scores, 0)
```



가장 가까운 악보 찾기

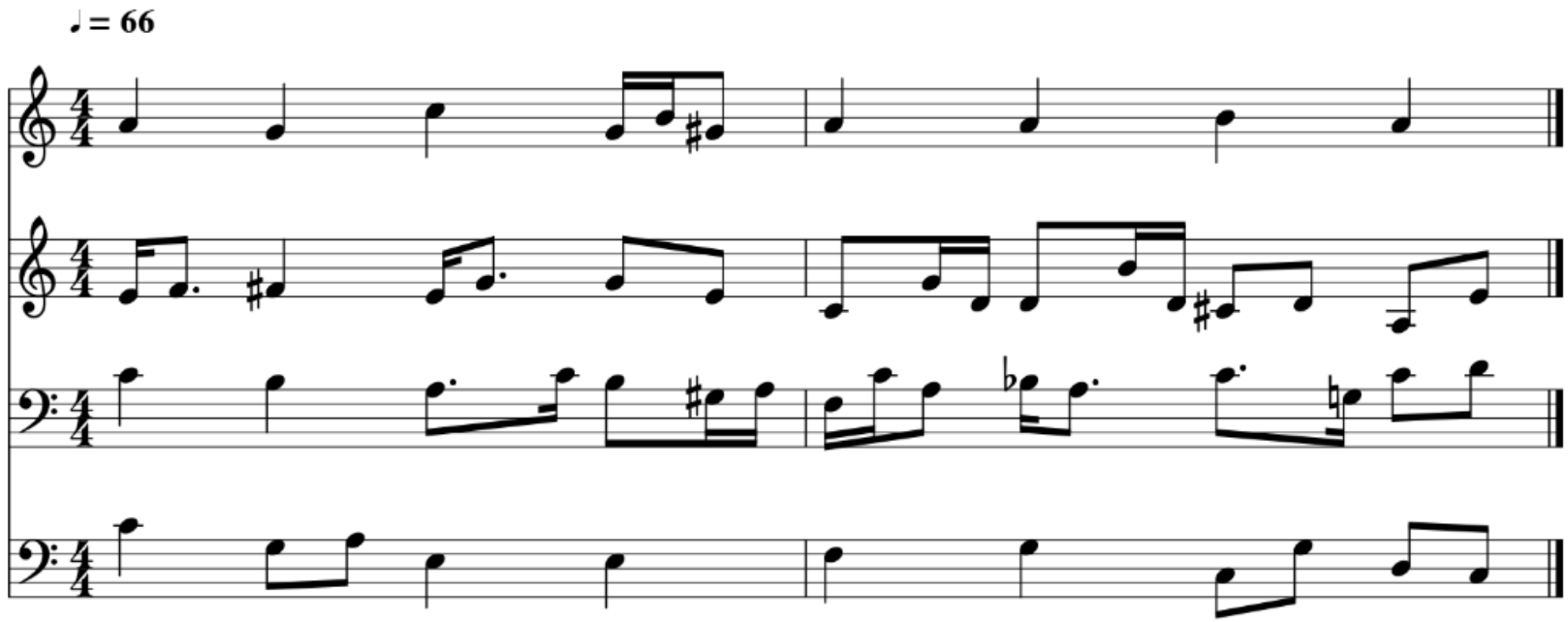
In [15]:
```
def find_closest(data_binary, score):
    current_dist = 99999999
    current_i = -1
    for i, d in enumerate(data_binary):
        dist = np.sqrt(np.sum(pow((d - score),2)))
        if dist < current_dist:
            current_i = i
            current_dist = dist

    return current_i
```

In [16]:
```
closest_idx = find_closest(data_binary, gen_scores[0])
closest_data = data_binary[[closest_idx]]
print(closest_idx)
```

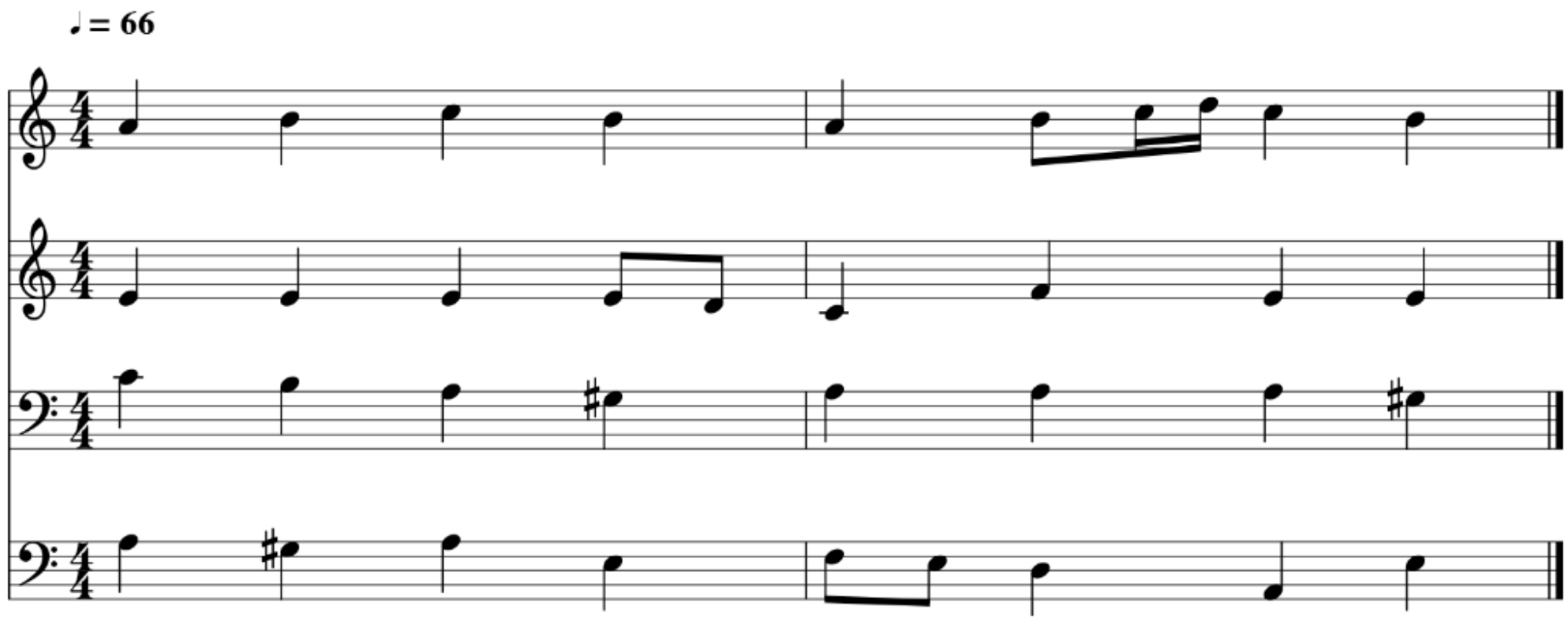
```
118
```

In [17]:
```
filename = 'closest'
gan.notes_to_midi(RUN_FOLDER, closest_data,filename)
closest_score = converter.parse(os.path.join(RUN_FOLDER, 'samples/{}.midi'.format(filename)))
print('생성된 악보')
gen_score.show()
print('가장 가까운 악보')
closest_score.show()
```

생성된 악보



가장 가까운 악보



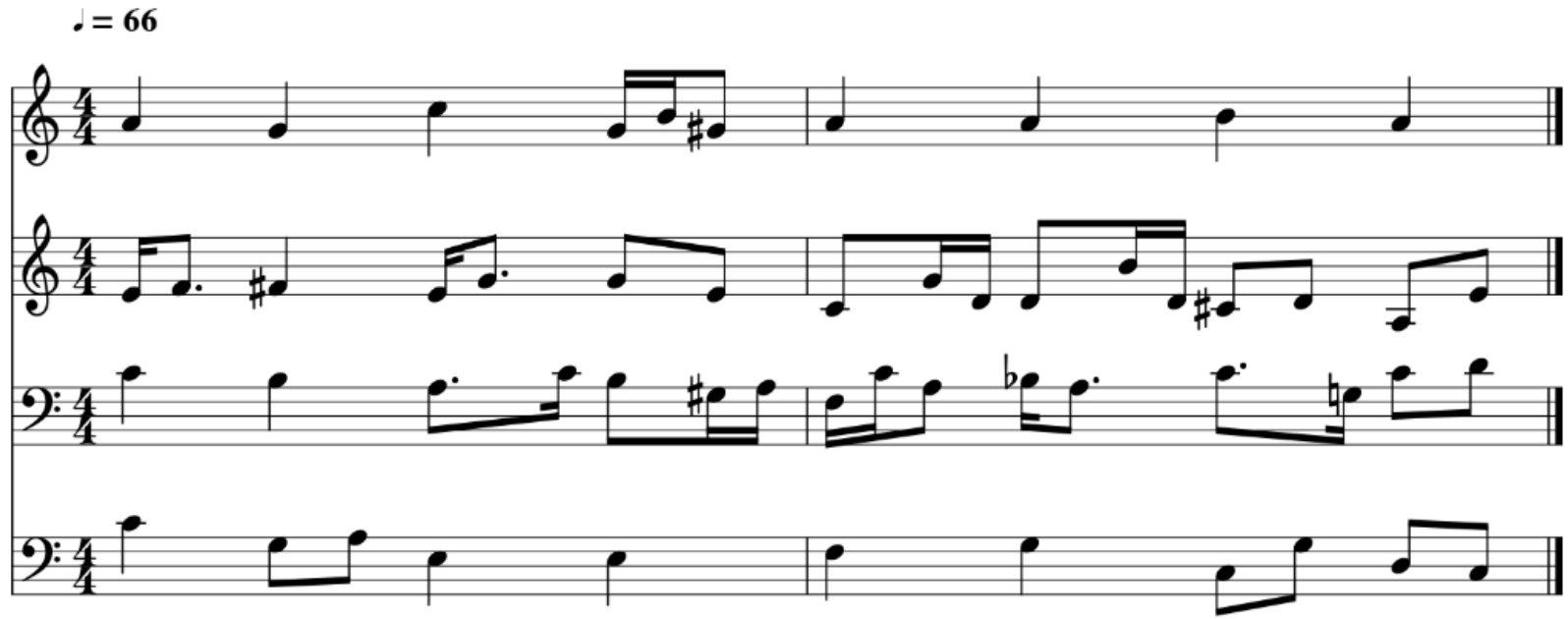
화음 잡음 바꾸기

In [18]:
```
chords_noise_2 = 5 * np.ones((1, gan.z_dim))
```

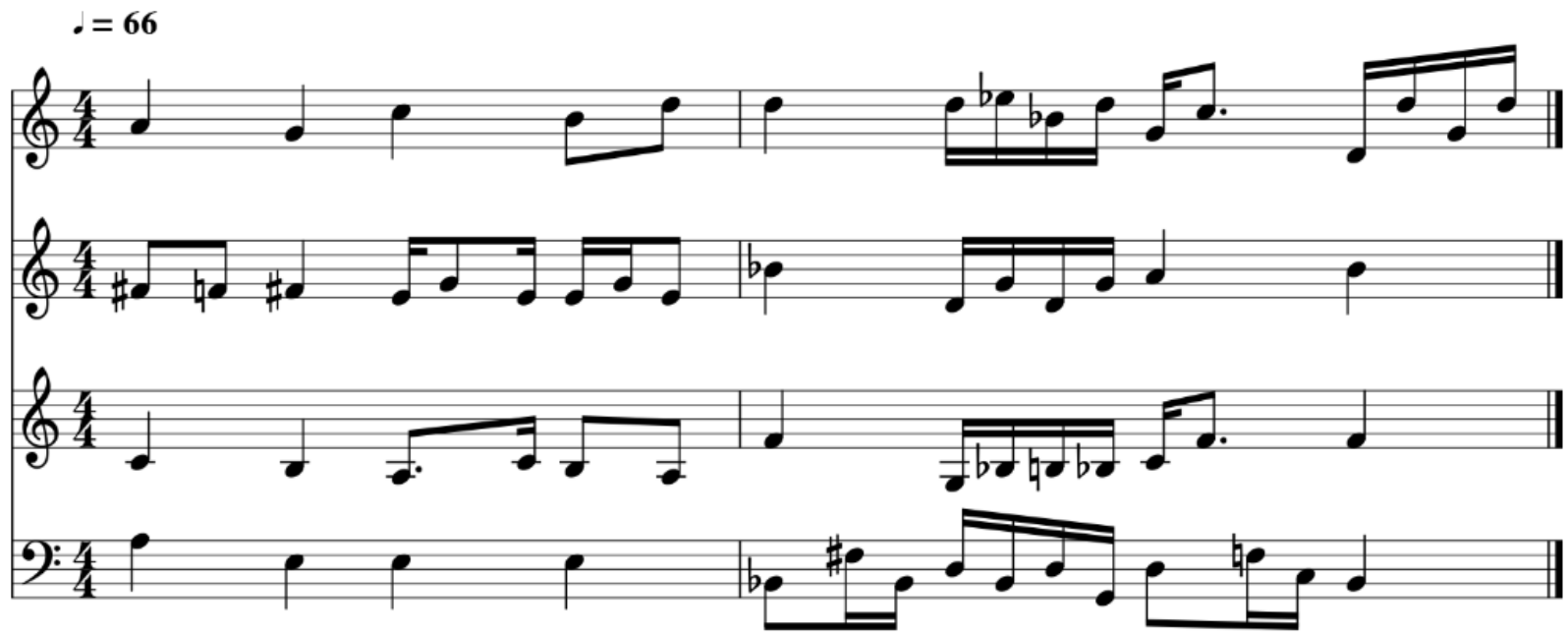
In [19]:
```
chords_scores = gan.generator.predict([chords_noise_2, style_noise, melody_noise, groove_noise])
```

In [20]:
```
filename = 'changing_chords'
gan.notes_to_midi(RUN_FOLDER, chords_scores, filename)
chords_score = converter.parse(os.path.join(RUN_FOLDER, 'samples/{}.midi'.format(filename)))
print('생성된 악보')
gen_score.show()
print('화음 잡음 변경')
chords_score.show()
```

생성된 악보



화음 잡음 변경



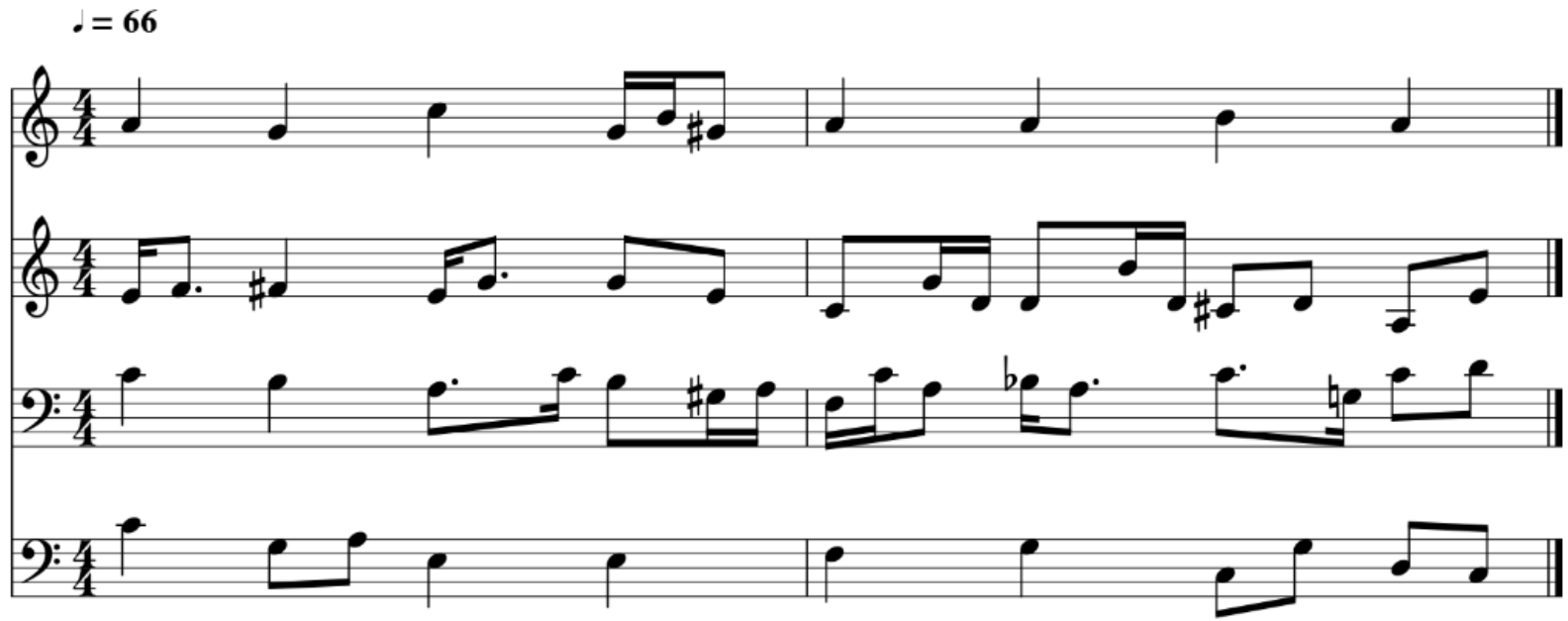
스타일 잡음 바꾸기

In [21]:
```
style_noise_2 = 5 * np.ones((1, gan.z_dim))
```

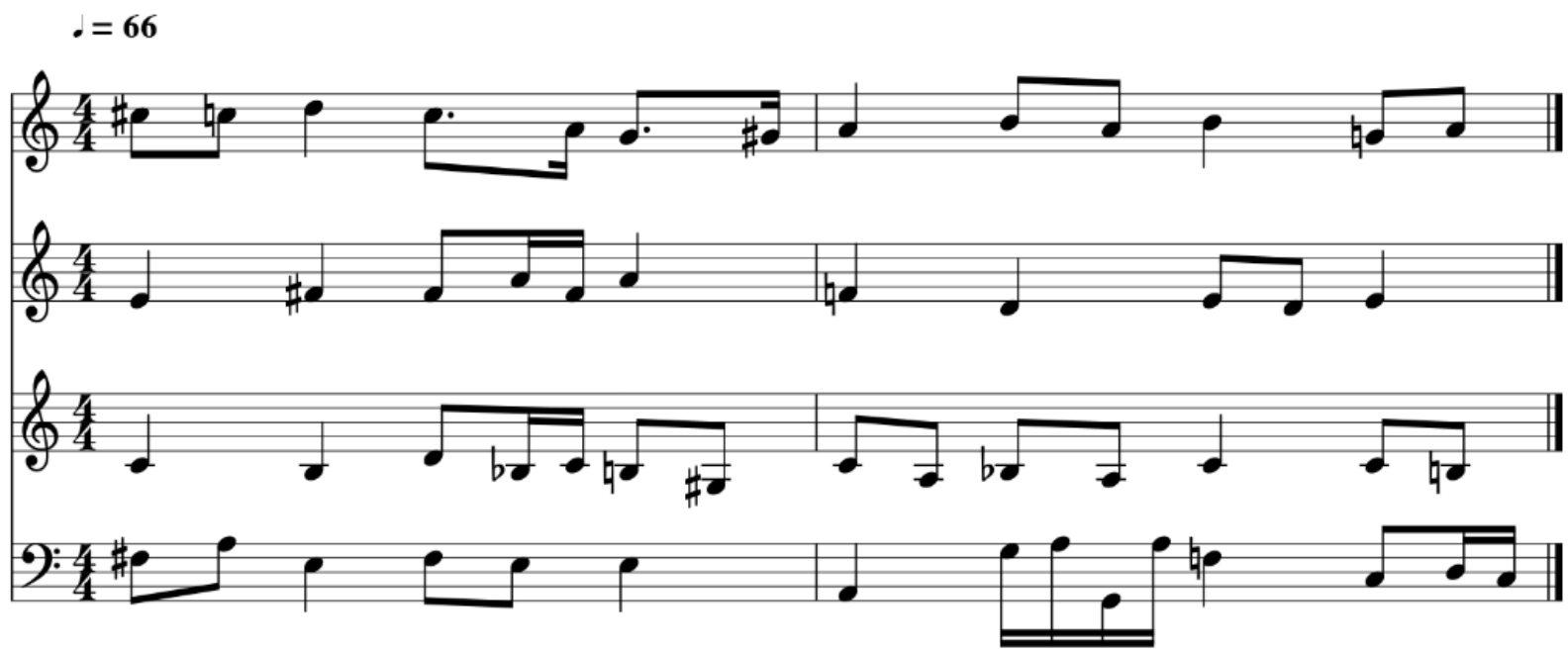
In [22]:
```
style_scores = gan.generator.predict([chords_noise, style_noise_2, melody_noise, groove_noise])
```

In [23]:
```
filename = 'changing_style'
gan.notes_to_midi(RUN_FOLDER, style_scores, filename)
style_score = converter.parse(os.path.join(RUN_FOLDER, 'samples/{}.midi'.format(filename)))
print('생성된 악보')
gen_score.show()
print('스타일 잡음 변경')
style_score.show()
```

생성된 악보



스타일 잡음 변경



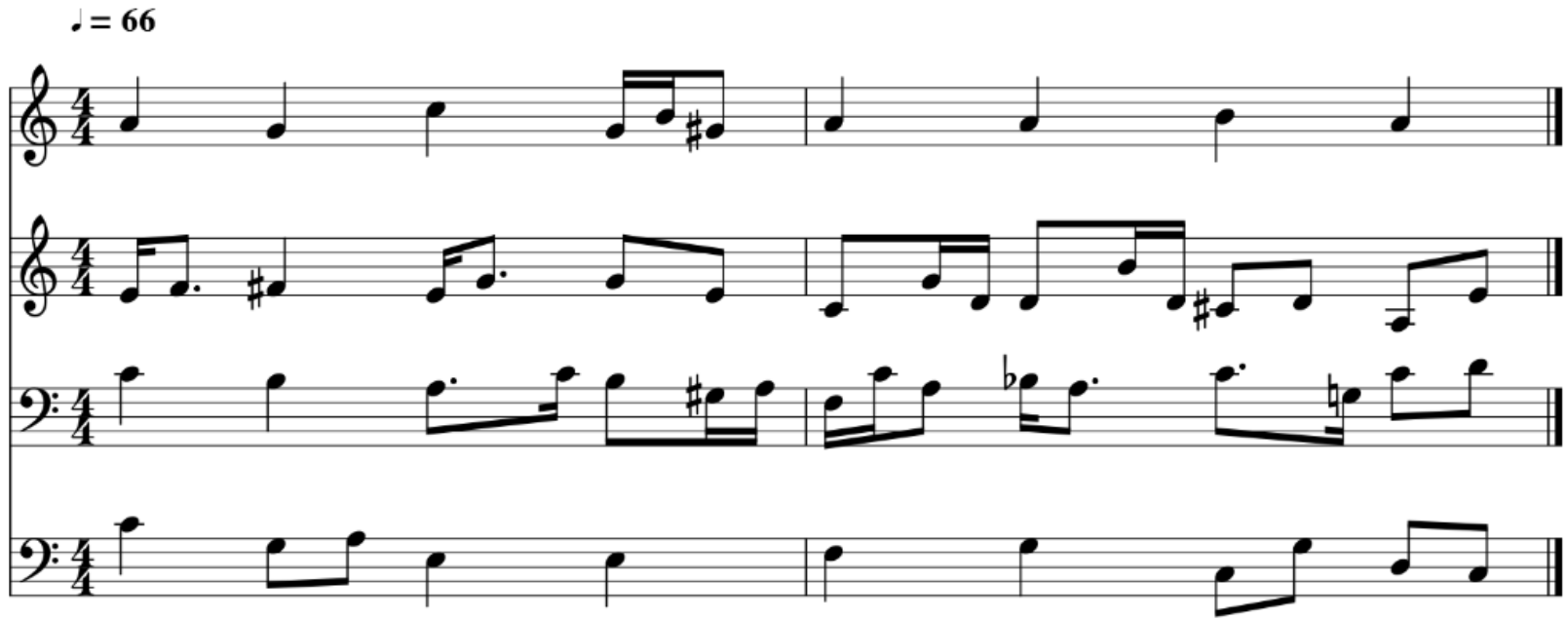
멜로디 잡음 바꾸기

In [24]:
```
melody_noise_2 = np.copy(melody_noise)
melody_noise_2[0,0,:] = 5 * np.ones(gan.z_dim)
```

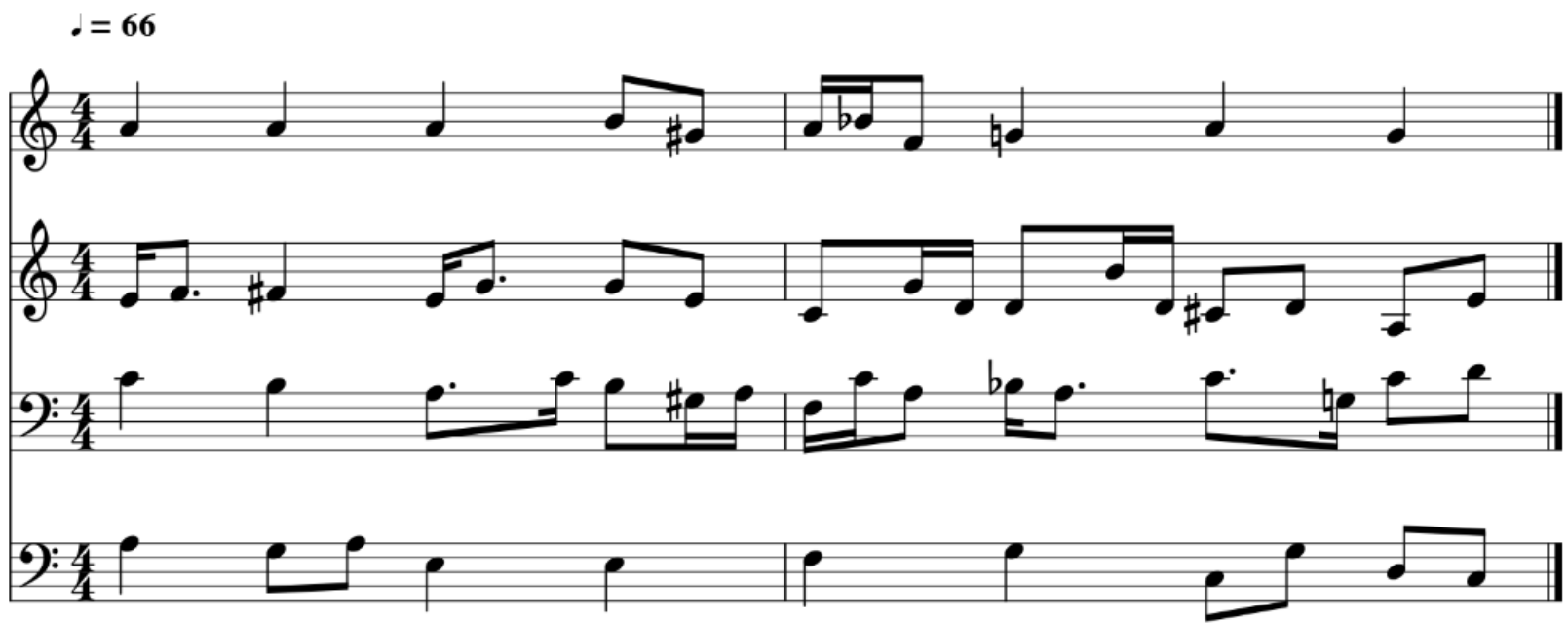
In [25]:
```
melody_scores = gan.generator.predict([chords_noise, style_noise, melody_noise_2, groove_noise])
```

In [26]:
```
filename = 'changing_melody'
gan.notes_to_midi(RUN_FOLDER, melody_scores, filename)
melody_score = converter.parse(os.path.join(RUN_FOLDER, 'samples/{}.midi'.format(filename)))
print('생성된 악보')
gen_score.show()
print('멜로디 잡음 변경')
melody_score.show()
```

생성된 악보



멜로디 잡음 변경



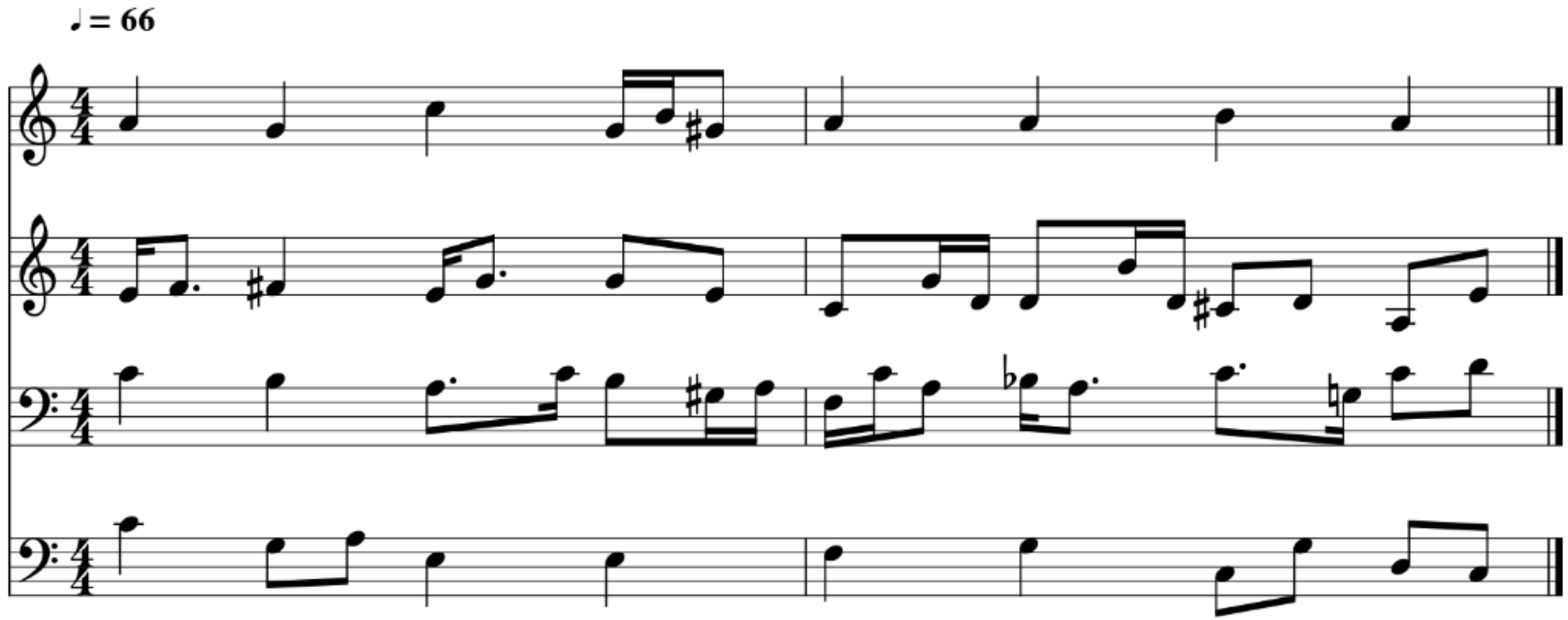
리듬(그루브) 잡음 바꾸기

In [27]:
```
groove_noise_2 = np.copy(groove_noise)
groove_noise_2[0,3,:] = 5 * np.ones(gan.z_dim)
```

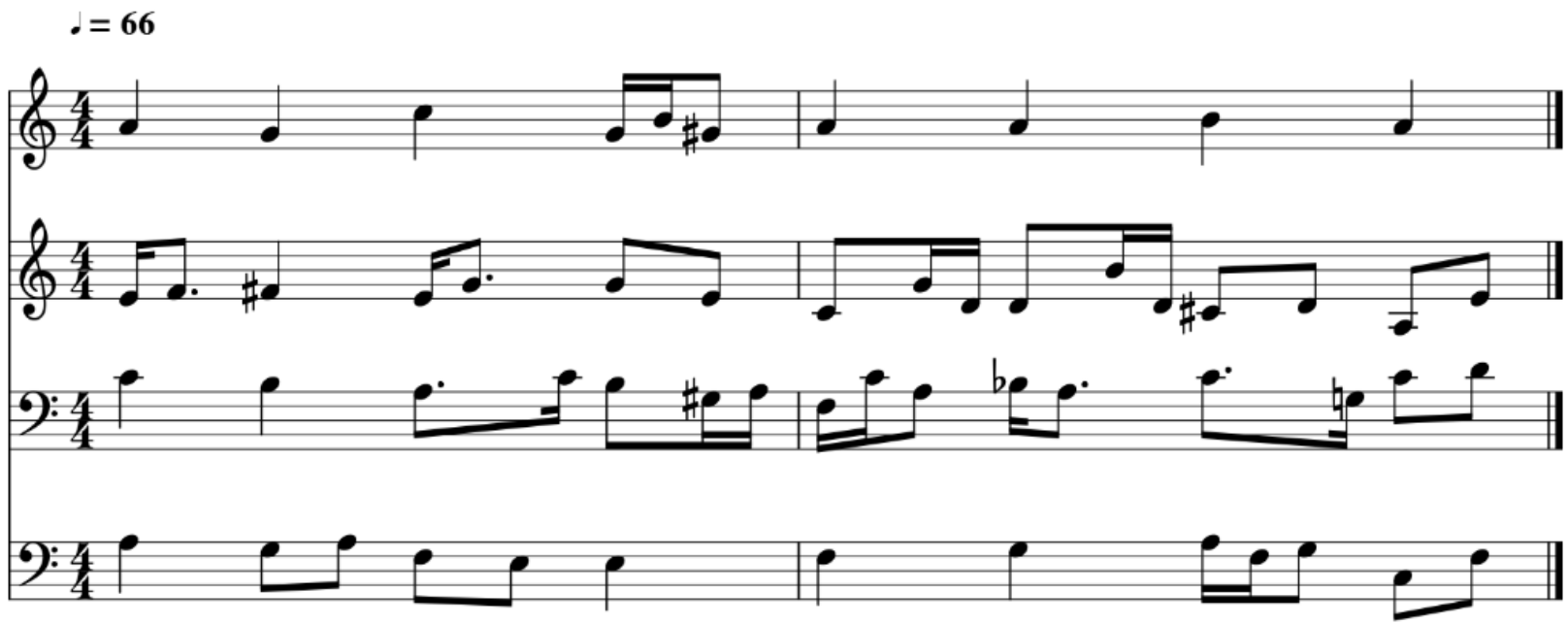
In [28]:
```
groove_scores = gan.generator.predict([chords_noise, style_noise, melody_noise, groove_noise_2])
```

In [29]:
```
filename = 'changing_groove'
gan.notes_to_midi(RUN_FOLDER, groove_scores, filename)
groove_score = converter.parse(os.path.join(RUN_FOLDER, 'samples/{}.midi'.format(filename)))
print('생성된 악보')
gen_score.show()
print('그루브 잡음 변경')
groove_score.show()
```

생성된 악보



그루브 잡음 변경



수고하셨습니다.