**<보고서>**

|  |
| --- |
|  |
| **Magenta Project** |
|  |

|  |  |  |
| --- | --- | --- |
|  | 팀원: | 고하은 |
|  | **팀원:** | **장선호** |
|  | **팀원:** | **정종윤** |

**목차**

[1 개요 3](#_Toc26715521)

[1.1 프로젝트 개요 3](#_Toc26715522)

[1.2 순환신경망(RNN) 3](#_Toc26715523)

[2 코드 분석 4](#_Toc26715524)

[2.1 RNN 신경망 4](#_Toc26715525)

[2.1.1 make\_rnn\_cell 4](#_Toc26715526)

[2.1.2 Make\_cudnn 5](#_Toc26715527)

[2.1.3 Build 5](#_Toc26715528)

[2.2 학습모델 6](#_Toc26715529)

[2.3 모델 훈련 7](#_Toc26715530)

[3 실행 8](#_Toc26715531)

[3.1 모듈 세팅 8](#_Toc26715532)

[3.2 NoteSequences 입력 9](#_Toc26715533)

[3.3 새로운 음악 요소의 생성 10](#_Toc26715534)

[3.4 최종 결과 11](#_Toc26715535)

[4 문제점과 해결방안 11](#_Toc26715536)

[4.1 문제점 11](#_Toc26715537)

[4.2 해결방안 12](#_Toc26715538)

[5 참고 문헌 12](#_Toc26715539)

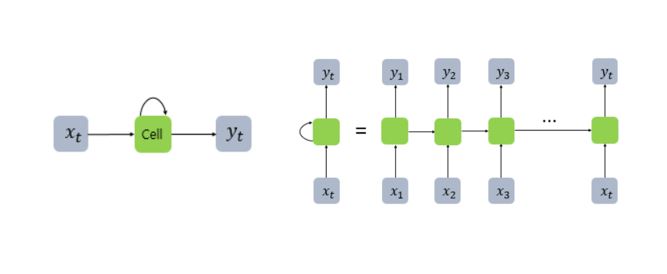
# 개요

## 프로젝트 개요

마젠타 프로젝트는 미술과 음악을 창조하는 인공지능 알고리즘을 설계하는 프로젝트이다. 재미있고 다양한 데모와 함께 인공지능의 창작 가능성을 볼 수 있다. 이 프로젝트의 궁극적인 목적은 사람들이 좋아하는 새로운 악기를 만들어내는 것이라고 한다.

## 순환신경망(RNN)

Recurrent Neural Network, 시간의 흐름대로 입력되는 데이터 패턴을 통해 특정 도메인에 대한 예측 모델을 만들기 위해 사용되는 알고리즘으로, 시간적 흐름에 따른 데이터의 패턴을 분석해 특정 시점의 결과 예측에 이전 시점의 데이터들이 관여한다는 특징을 가진다. 가령, 은닉층 노드에서 활성 함수를 통해 나온 결과 값을 출력층 방향으로도 보내면서 다시 은닉층 노드의 다음 계산의 입력으로 보낸다.

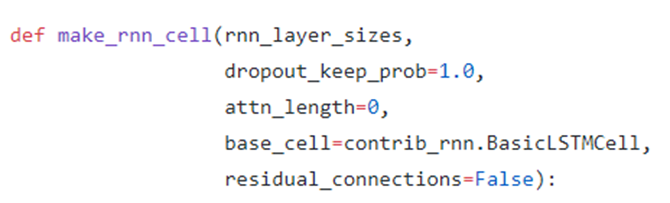
<그림 1> RNN 구조의 도식화

<그림 1>에서 x는 입력층의 입력 벡터, y는 출력층의 출력 벡터이다. RNN에서 은닉층에서 활성 함수를 통해 결과를 내보내는 역할을 하는 노드를 셀이라고 하는 데 앞서 언급한 바와 같이 이 셀은 다음 계산의 입력으로 보내기 위해 이전의 값을 기억하려고 하는 일종의 메모리 역할을 수행하므로 메모리 셀 또는 RNN 셀이라고 표현하는데, 이러한 형태의 신경망 구조가 Magenta 프로젝트에도 나타나 있다.

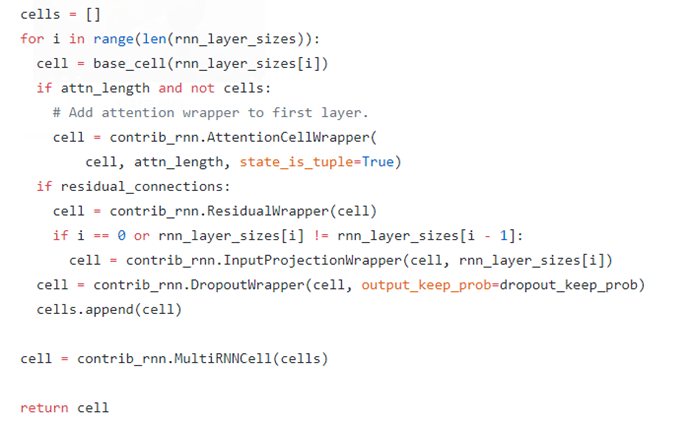
# 코드 분석

## RNN 신경망

### make\_rnn\_cell



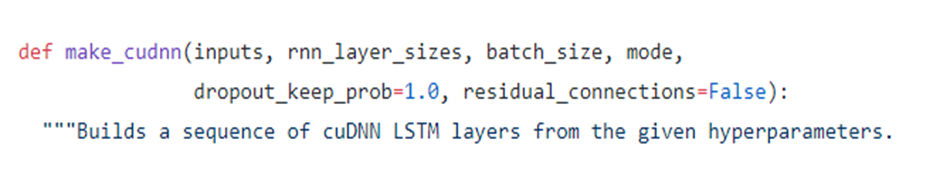
<그림 2> make\_rnn\_cell



<그림 3> add attention wrapper

위의 <그림 2>를 보면 make\_rnn\_cell이라는 함수로 RNN 신경망의 셀을 만든다. 입력값으로 각 rnn 층의 크기를 나타내는 배열인 rnn\_layer\_size, 주어진 sub-cell의 출력을 유지하기 위한 실수형 확률값인 dropout\_keep\_prob, attention vector의 크기를 나타내는 attn\_length, rnn cell을 만들기 위한 base cell과 추가 연결을 할지 말지 결정하는 residual\_connection을 입력 받아 최종적으로 multi RNN cell을 출력한다. 이 과정은 cell이라는 빈 리스트를 만들고 조건에 따라 리스트에 cell을 추가하는 방식으로 진행되는데 이 때 첫 번째 층에 attention wrapper를 추가해준다. attention은 출력 시퀀스의 정확도를 높이기 위해 사용하는 것으로 해당 시점에서 예측해야 할 것과 연관이 있는 입력 부분을 좀 더 집중해서 보게 만든다. 이는 <그림 3>에 주석 #add attention wrapper to first layer에 나타난다.

### Make\_cudnn



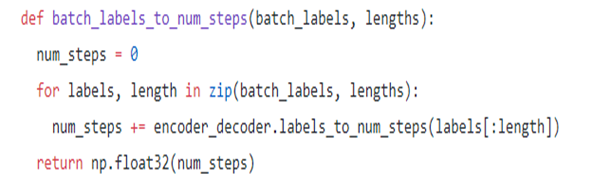
<그림 4>make\_cudnn

다음으로 make\_cudnn 함수는 주어진 하이퍼 매개변수를 통해 cuDNN lSTM 층을 조직한다. cudnn은 CUDA의 딥러닝 신경망 라이브러리이며 LSTM은 RNN이 불충분한 기억력을 가지고 있을 때의 문제점을 보완하기 위한 장단기 메모리이다. LSTM은 은닉층의 메모리 셀에 입력 게이트, 망각 게이트, 출력 게이트를 추가해 구성되며 필요한 기억을 지우고 기억해야 할 것을 정하는 역할을 수행한다.

### Build



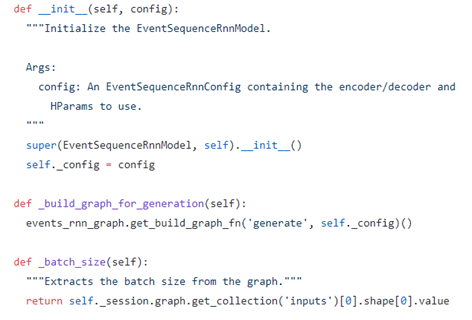
<그림 5> build



<그림 6> batch labels to num step

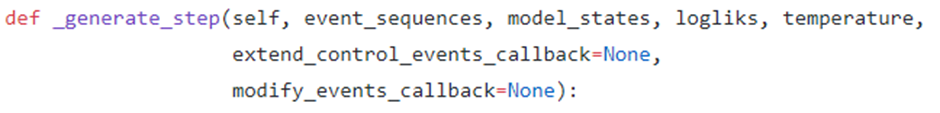
마지막으로 Build 함수는 Tensorflow 그래프를 만들고 배치의 전 과정에서의 time steps를 계산할 수 있도록 함수 batch\_labels\_to\_num\_steps를 만든다. 여기서 배치는 한 번에 처리하는 데이터 양을 뜻한다.

## 학습모델



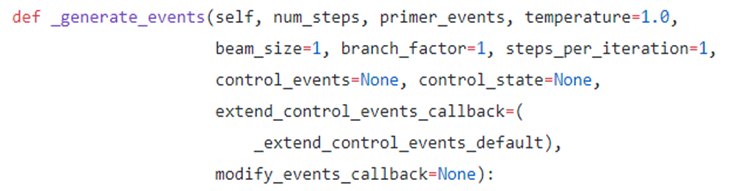
<그림 7> event sequence RNN model 클래스

event sequence RNN model이라는 클래스를 통해 학습 모델을 만든다. 초기화, 그리고 배치의 크기를 결정하는 함수들을 정의하고 있다.



<그림 8> generate\_step

이어서 generate\_step에서는 시행 중인 event sequence를 수정할 수 있도록 한다.



<그림 9> generate\_events

generate\_event는 이전의 sequence에서 event sequence를 생성하는 함수로 이는 마젠타의 주요 목적인 음악의 생성, 그림 그리기와 밀접한 연관이 있는 함수로 은닉층의 메모리 셀에서 나온 값을 자신의 입력으로 사용하는 RNN의 재귀적 활동을 반영하는 함수이다.

## 모델 훈련



<그림 10> 훈련

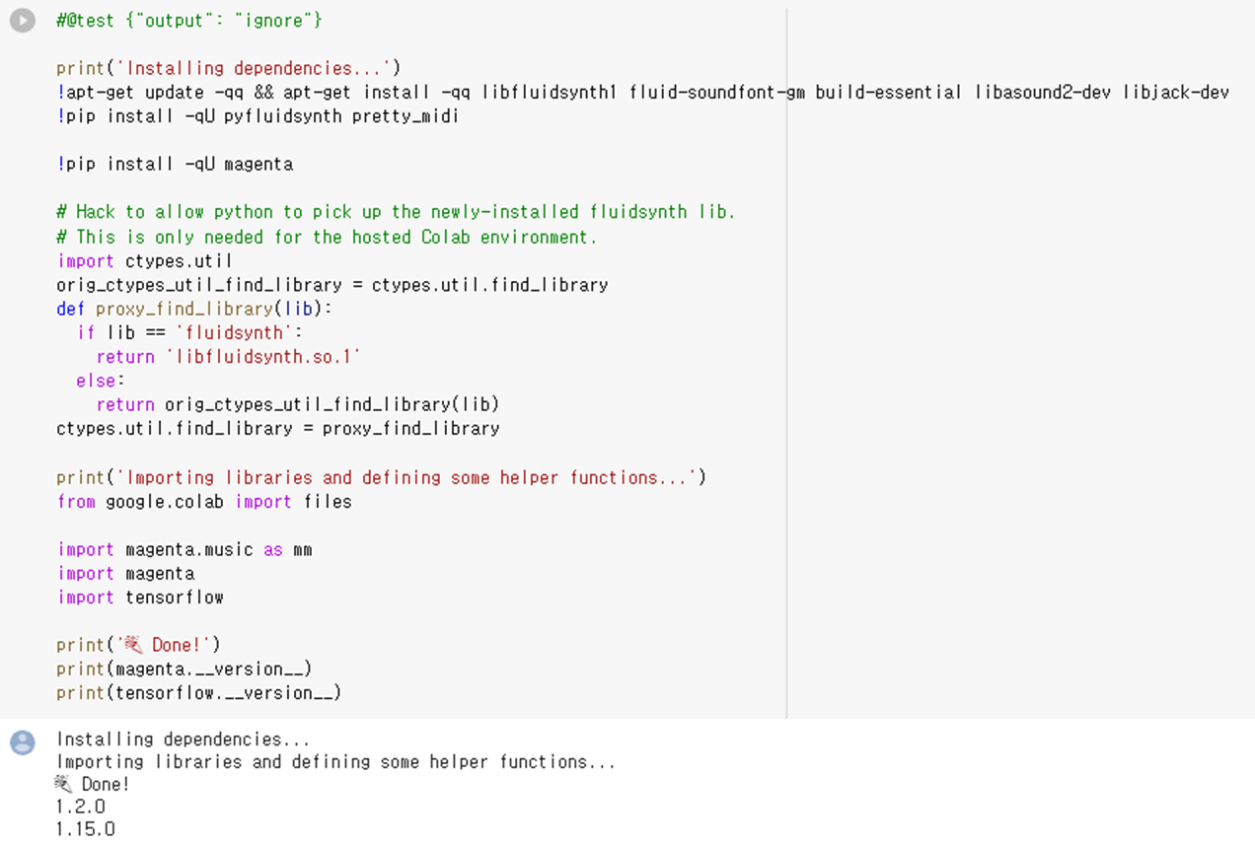
훈련 부분에서는 run training 함수를 통해 앞의 event sequence rnn model을 훈련시킨다. with tf.Graph().as\_default() 부분은 지금까지 만들었던 모든 그래프 구성 요소(operation과 텐서)들을 하나의 전역 graph 안에서 사용하겠다는 뜻이다.

global\_step은 학습의 step 카운트를 자동으로 관리해주는 tensor로써 사용자가 별도로 step을 카운트할 필요 없이 이 global step을 이용할 수 있도록 한 것이다. 해당 코드는 딕셔너리 변수를 통해 훈련과 관련된 변수를 관리한다. num\_training\_step은 훈련에 필요한 step 수로, 해당 변수만큼 훈련이 진행된다. 훈련은 텐서 플로우 컨트립 트레이닝 모듈의 train 함수를 통해 이루어진다.

# 실행

## 모듈 세팅

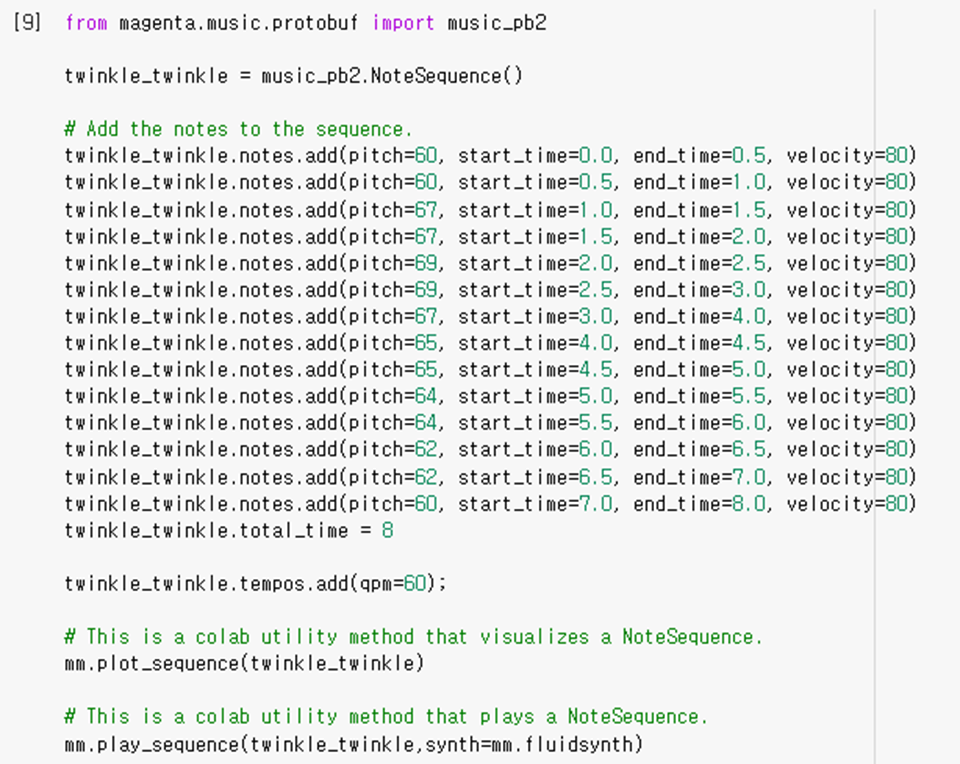
프로그램 실행에 필요한 모듈을 세팅해줍니다.



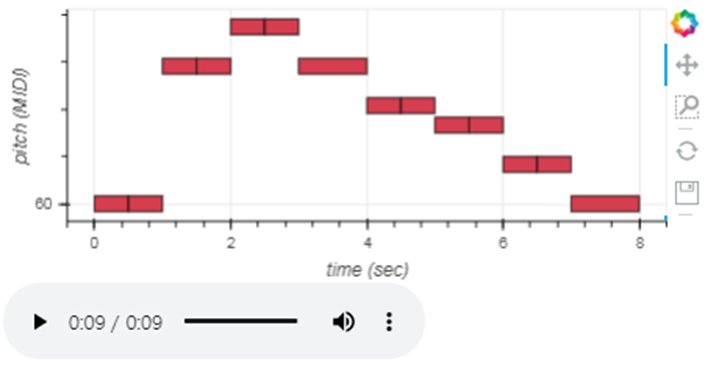
<그림 11> 모듈 세팅

## NoteSequences 입력

입력하고자 하는 음악의 NoteSequences를 입력해주는데, 우리는 노래 ‘반짝반짝 작은별’의 NoteSequences를 입력해주었다.



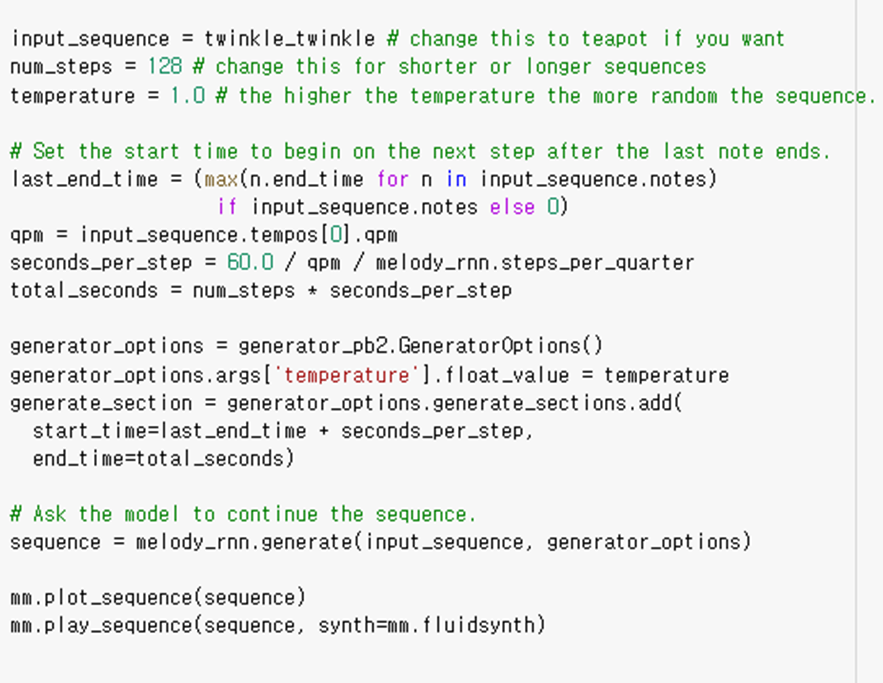
<그림 12> NoteSequences 입력



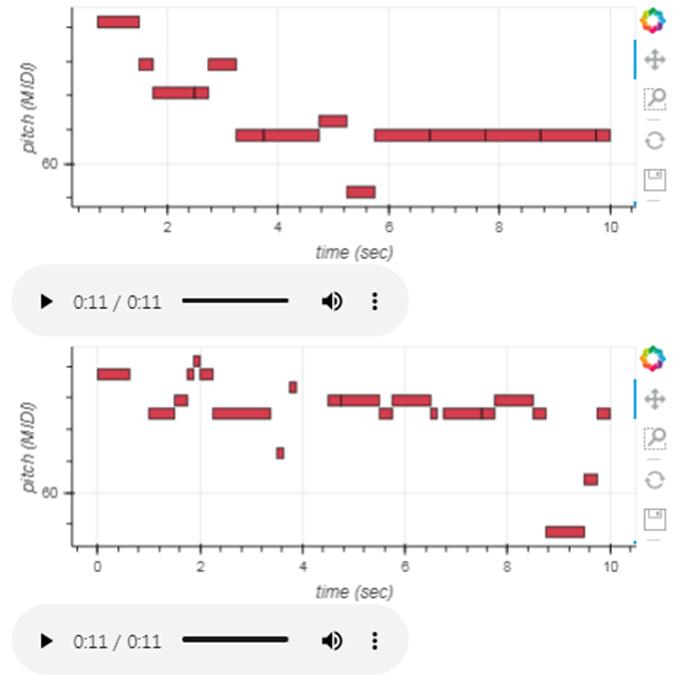
<그림 13> NoteSequences 입력결과

## 새로운 음악 요소의 생성

Melody RNN모델을 기반으로 입력된 음악과 어울리는 새로운 음악 요소를 생성한다.



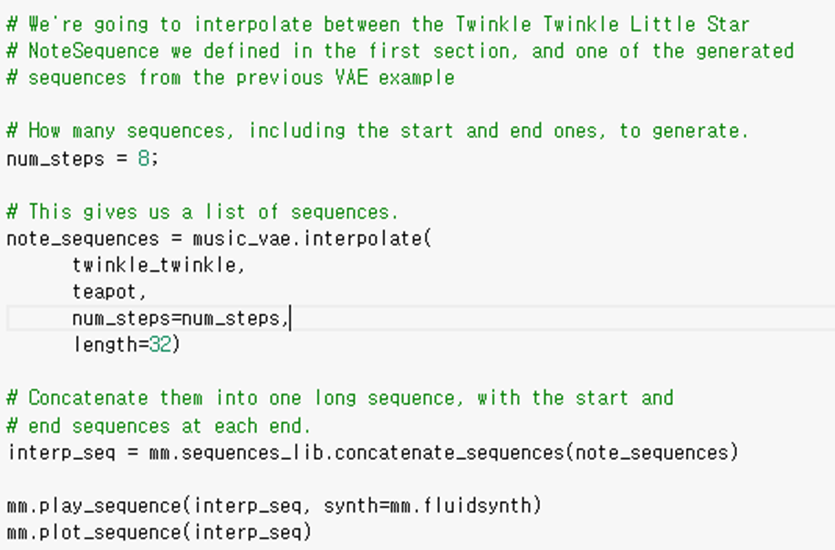
<그림 14> 음악 요소 생성



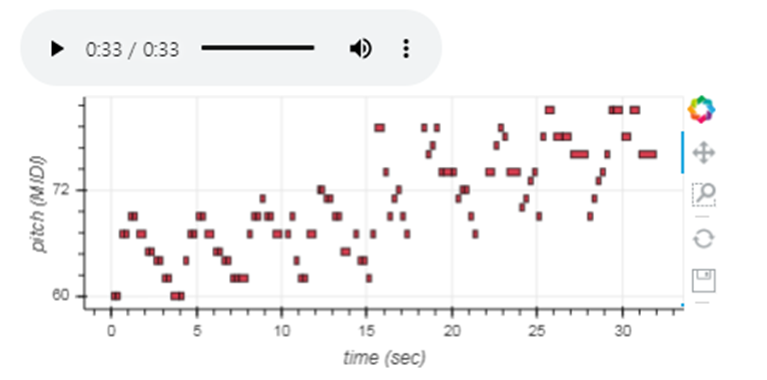
<그림 15> 음악 요소 생성 결과

## 최종 결과

입력된 음악과 생성한 음악 요소를 시켜준다.



<그림 16> 입력한 음악과 생성한 음악 요소들을 결합



<그림 17> 최종 결과

# 문제점과 해결방안

## 문제점

구글 프로젝트팀 마젠타 프로젝트의 목표가 “머신 러닝을 통해 예술과 음악을 창조할 수 있는지 알아보는 것”이라고 밝혔다. 작곡하는 인공지능 마젠타의 최종 목표는 인간의 마음까지 움직일 수 있는 ‘서사’를 만들어내는 것이다.

하지만 ‘기쁘다, 슬프다, 무섭다’ 와 같은 감정을 느낄 수 없는 기계가 인간을 웃거나, 울게 만드는 예술 작품을 ‘창작’할 수 있을지는 예상할 수 없다. 2016년 구글이 공개한 마젠타 프로젝트의 80초짜리 피아노곡은 인간이 일반적으로 화음이라고 느낄 수 없는 기초적인 음계를 보여주는데, 단순하고 뻔한 선율이기에 아직 갈 길이 먼 것처럼 느껴진다.

또한 음반업계에서는 어쩔 수 없는 ‘유행’이라는 것이 존재한다. 하지만 인공지능이 그러한 유행까지 자동적으로 습득하고 유행에 맞춰 장조를 선택할지, 단조를 선택할지, 어떤 리듬을 선택할지 자연스럽게 따라가기는 쉽지 않다.

## 해결방안

기계의 음악성이나 창의성에 대한 연구는 우리의 뇌만큼 복잡하고 알 수 없는 것들로 가득 차 있다. 이러한 문제를 해결하기 위해 화성학에 관련된 정보를 학습시키거나 화음에 대한 유형을 학습시키면 조금 더 화음의 문제가 해결되지 않을까 생각해본다.

# 참고 문헌

**소스코드**

https://github.com/tensorflow/magenta?utm\_source=mybridge&utm\_medium=blog&utm\_campaign=read\_more

https://github.com/tensorflow/magenta/commit/db5d35611c95b1d3ad0a9e324dbe2a192d35c4d4#diff-29ca8eb85c15b46a4fa6b69a16b51050L444

https://github.com/tensorflow/magenta/tree/master/magenta/models/shared

https://colab.research.google.com/notebooks/magenta/hello\_magenta/hello\_magenta.ipynb