**일일보고서**

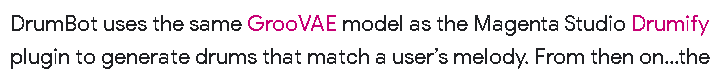
**컴퓨터공학부**

**2017156037 정수경**

**2020-08-12**

**▶ Drumbot 데모에 사용된 학습 모델 확인**

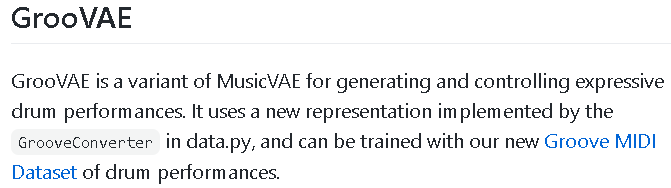
→ 출처: <https://magenta.tensorflow.org/groovae>

****

**▶ Drumbot 데모에 사용한 GrooVAE 모델 분석**

**(1) GrooVAE 모델**

→ 출처: <https://magenta.tensorflow.org/groovae>

****

GrooVAE는 표현력 있는 드럼 연주를 만들어내고 제어하기 위한 뮤직베이의 변종이다. data.py에서 GrowConverter에 의해 구현된 새로운 표현을 사용하며, 드럼 공연의 새로운 Groove MIDI Dataset으로 훈련을 받을 수 있다.

**(2) GrooVAE.ipynb구동 (Google Colab 사용)**

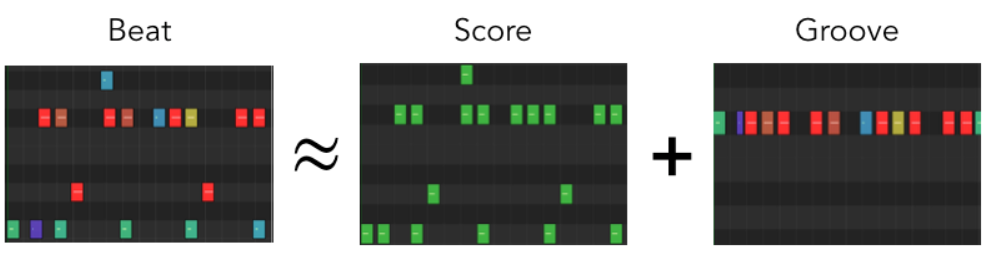
→ 출처:<https://colab.research.google.com/github/tensorflow/magenta-demos/blob/master/colab-notebooks/GrooVAE.ipynb>

이 notebook은 비트와 드럼 연주를 만들고 조작하기 위한 기계 학습의 응용을 보여준다. 우리는 몇몇 재능 있는 전문 드러머들을 고용하여 전자 드럼 키트에 녹음하고 (자세한 내용은 그루브 MIDI 데이터세트를 참조), 그리고 나서 이 데이터에 대해 "GrooVAE" 모델을 훈련시켰다.

MIDI 드럼 비트(박자)가 라이브로 재생되든 전자 시퀀싱이 되든 간에 MIDI 드럼 비트에 대해 생각하는 한 가지 방법은 드럼 비트를 두 가지 주요 구성 요소로 분해하는 것이다.

* 스코어(서양의 음악 표기법으로 쓰인 드럼): 악보
* 그루브(드럼 재생 방법, 즉 역동성과 타이밍): 리듬

드럼 비트(박자)의 한 가지(간단한) 관점은 그것이 스코어와 그루브의 결합이라는 것이다. 어느 쪽이든, 훌륭한 드러머는 완전한 비트를 만들기 위해 나머지를 채우는 방법을 알고 있다 - 이 프로젝트에서, 우리는 모델에게 이 완성을 가르치기 위해 노력한다.

****

**박자 ≈ 악보 + 리듬**

**(1) Environment Setup (colab 환경 구축)**

**(2) Generate New Beats (새로운 박자 생성)**

좀 더 구체적으로 얘기하기 전에 처음부터 박자를 만들어 봅시다. Variational Autoencoder 모델의 강력한 능력 중 하나는 그들이 훈련받은 것과 유사한 새로운 데이타포인트를 생성하는 것이다. MusicVAE와 마찬가지로 우리의 잠재된 공간에서 새로운 비트를 얼마든지 샘플링할 수 있지만 GrooVAE를 통해 우리의 잠재된 공간에서 드럼 패턴뿐만 아니라 그것을 연주한 드러머들의 연주 특성도 암호화할 수 있다. 우리는 또한 잠재된 공간에서 다른 박자 사이를 부드럽게 보간할 수 있다.

**- Load checkpoint**

**- Generate Beats**

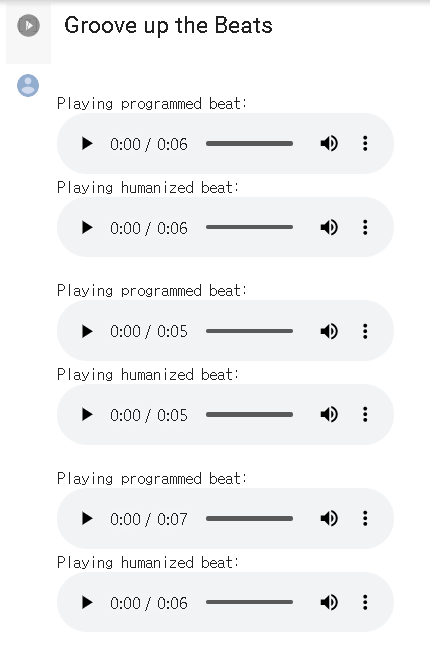
**- Interpolate Between Beats**

**(3) Groove: Add some groove to a programmed beat (그루브: 프로그래밍된 박자에 그루브를 더하기)**

이제 정량화된 비트(박자)에 그루브(리듬)을 더하는 것이 어떤 것인지 알아보자. 이것은 음악 제작에서 드럼에 더 많은 특징를 주기 위해 자주 사용되는 기능이다. 과거에는 일반적으로 음표의 타이밍과 속도를 임의로 변경하거나 모든 타이밍과 속도를 "swing" 설정 또는 템플릿에 의해 정의된 특정 값에 고정하는 방식으로 수행되었다. 대신 우리는 모델이 리듬과 속도를 비트가 무엇인지에 따라 조정하면서 그루브 특성이 어떻게 되어야 하는지 예측하게 한다.

**- Load checkpoint**

**- Groove up the Beats**

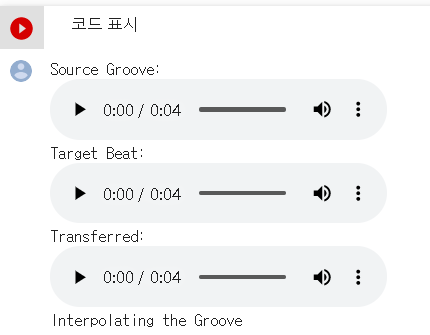
****

**(4) Tap2Drum: Generate a beat from any rhythm (리듬으로부터 박자 생성)**

Groove 모델은 마이크로타이밍과 속도정보를 없애고 드럼 패턴만으로 예측하는 학습을 하는 방식으로 작동하지만, 반대 방향으로 할 수 있다. 여기서 Groove를 입력(정확한 타이밍은 가질 수 있지만 드럼 카테고리는 무시되는 리듬의 형태)으로 표현한 다음 이 리듬이 내포한 그루브와 일치하는 드럼 비트를 생성한다. 우리는 이 모델을 훈련 데이터의 각 박자의 모든 드럼 히트를 단 하나의 "탭" 리듬으로 접고 그 리듬으로부터 완전한 박자를 해독하는 법을 배우면서 훈련시켰다. 이것은 우리가 "탭"에 정확한 시작 시간을 통해 좋아하는 리듬을 입력할 수 있게 하고 모델이 우리의 리듬을 박자로 해독하게 한다. 우리는 단지 탭들을 오디오로 녹음할 수도 있고, 미디 컨트롤러가 필요하기 보다는 다른 악기의 녹음에서 추출할 수도 있다.

**(5) Transfer a Groove from one Beat to another (리듬을 하나의 박자에서 또다른 박자로 변환하기)**

GrooVAE 모델의 또 다른 재미있는 용도는 "Groove Transfer"이다. 데이터셋에서 무작위 비트를 두 개 로드하여 한 박자의 "그루브"와 다른 박자의 드럼 패턴을 어떻게 조합하는지 봅시다. 그리고 나서 우리는 가능한 그루브의 "공간"을 통해 부드럽게 움직이거나 보간하는 것이 어떤 것인지 살펴볼 것이다. 쉽게 제어할 수 있는 또 다른 흥미로운 가능성은 "탭"을 소스 그루브로 사용하는 방법을 배우는 것이다.



**▶ Google Colab 런타임 연결 (추가 공부)**

출처<https://colab.research.google.com/github/jsstar522/facebook_MLClass/blob/master/11_Colab_Basics.ipynb>

<https://theorydb.github.io/dev/2019/08/23/dev-ml-colab/>

① 호스팅된 런타임에 연결(Connect to Hosted Runtime)

→ Google Cloud의 새로운 머신 인스턴스에 연결

→ 라이브러리를 다시 설치해야할 수도 있음

→ Colab의 연결이 끊길 경우 해당 버튼을 클릭해서 다시 연결할 수 있음

② 로컬 런타임에 연결(Connect to Local Runtime)

→ 내 PC(로컬)을 사용

**▶ Drumbot 활용 방안 및 구현 계획**

- 구현 방향: Magenta js 기반으로 구현

- Magenta에서 지원하는 모델을 분석하고 구동해 봄으로써 기존 멜로디 생성 방식을 개선

**▶ Melody RNN 구동 방법 (08/11 진행사항)**

→ 출처: <https://github.com/magenta/magenta/tree/master/magenta/models/melody_rnn>

① 미리 학습된 모델 사용



해야 할 것: 멜로디 생성하기

② 직접 학습시키기

1. Dataset 구축하기

→ 출처: <https://github.com/magenta/magenta/blob/master/magenta/scripts/README.md>

MIDI가 없다면 MidiWorld(<https://www.midiworld.com/files/142/>)나 [Lakh MIDI Dataset](http://colinraffel.com/projects/lmd/)

(<https://colinraffel.com/projects/lmd/>)에서 다운로드 가능

1. NoteSequences 생성하기

MIDI 파일들을 NoteSequences로 변환하기

NoteSequences 과물은 /tmp/notesequences.tfrecord 에 저장됨

1. Sequence 예 생성하기
2. 모델 훈련시키고 평가하기
3. 멜로디 생성하기