

AIを用いた楽曲制作に関する検討

1532117 秋場 翼

1532151 松元 孝樹

指導教員：中村 直人 教授

平成31年度

目次

第1章	序論	1
1.1	研究の背景と目的	1
1.2	本論文の構成	2
第2章	理論	3
2.1	AIを用いた楽曲作成	3
2.1.1	MIDI	3
2.1.2	Magenta	3
2.2	開発環境の構築	4
第3章	実験内容	5
3.1	モデルによる違い	5
3.1.1	MelodyRNN	5
3.1.2	PolyphonyRNN	6
3.2	学習回数による違い	6
3.3	ノード数による違い	6
第4章	楽曲制作	7
4.1	NoteSequenceについて	7
4.2	NoteSequenceの作成	7
4.2.1	学習の開始	8
4.2.2	音楽データの作成	9
第5章	結論	10
5.1	今後の課題	11
	謝辞	12
	参考文献	13

目 次

1.1	magenta による MIDI 音楽データ生成までのプロセス	1
2.1	magenta による MIDI 音楽データ生成までのプロセス	3
3.1	NoteSequence の作成	5
4.1	NoteSequence の作成	7
4.2	NoteSequence を学習用と評価表に分割	8
4.3	BasicRNN を使用した学習の開始	8
4.4	学習モデルを使用し,10 曲を作成	9

表 目 次

2.1 開発環境	4
--------------------	---

第1章 序論

1.1 研究の背景と目的

近年，AI 分野は急速な発展を続けている．スマートスピーカなどの対話型の AI が Google や Amazon, IBM によって商品化され，現在ではスマートフォンにも Siri という AI が搭載されるなどその存在は非常に身近になっており，その種類も非常に多岐にわたる．



図 1.1: magenta による MIDI 音楽データ生成までのプロセス

また囲碁や将棋, チェスなどの競技においても，プロに勝利するなどその精度は以前から高いが，その AI は一つの競技でしか使用できない特化型人工知能 (AGI) であった．しかし，英 DeepMind が発表した AlphaZero という様々なボードゲームに対応できる汎用性を持った AI が発表され，汎用人工知能 (GAI) の成長も著しい．芸術の分野ではまだ発展途上ではあるが，絵画や音楽に関しても AI を用いて新しい作品を作るものが出回っている．

このように AI の発展は様々な分野においてその成果を上げており，今後は業務の効率化や補助だけにとどまらず，自動車の自動運転や医療の現場でも人間の手よりも高精度なものとして活躍することが期待されている．

本研究では AI による楽曲生成についての実証実験を行う。Googole brain によって公開されている Magenta を用いて学習データやノード数による楽曲の生成結果の違いを比較，検証し，AI による楽曲制作が有用なものか調査する。

1.2 本論文の構成

本論文の構成は以下の通りである。

第 1 章では本論文の背景と目的について述べている。

第 2 章では本論文で利用する理論について述べている。

第 3 章では実験内容について述べている。

第 4 章では楽曲制作について述べている。

第 5 章では AI を用いた楽曲制作についての本研究の結論について述べている。

第2章 理論

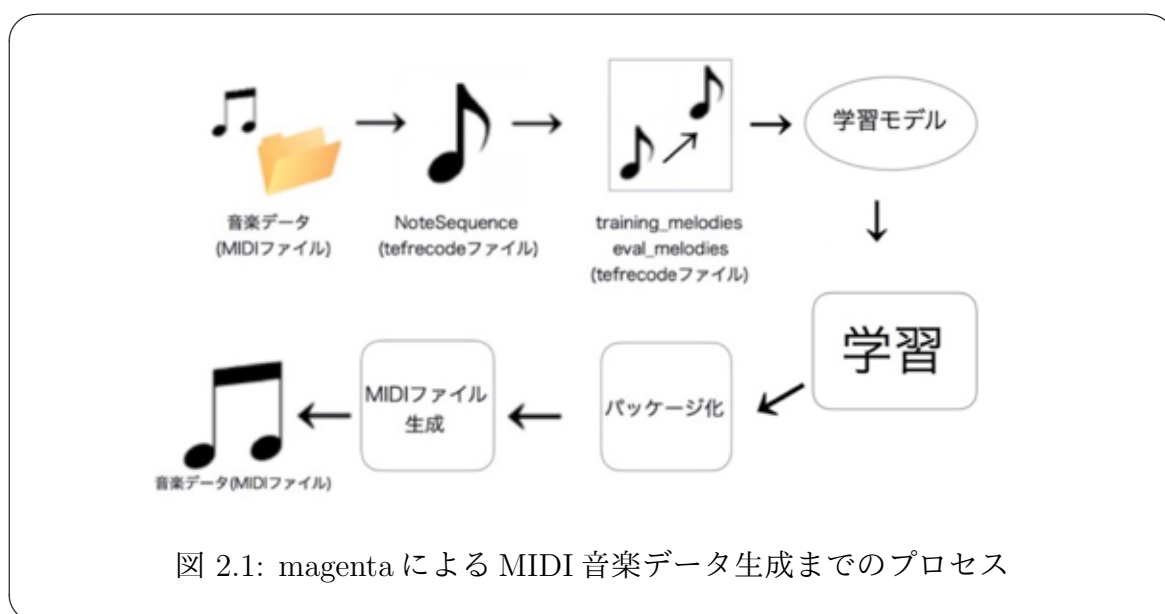
2.1 AIを用いた楽曲作成

2.1.1 MIDI

AIによる曲制作では主に MIDI ファイルの音楽データを使用する。MIDI ファイルには実際の音ではなく音楽の演奏情報（音の高さや長さなど）である。本研究で用いる AI はこの MIDI ファイルの情報を元に学習をする。また入出力の際もこの規格を用いる。

2.1.2 Magenta

本研究では Magenta[1] を使用する。これは音楽などを TensorFlow を使って機械学習するライブラリであり、Google Brain が GitHub 上に公開している。Magenta ではまず学習させたい音楽の MIDI データを NoteSequence（magenta が扱うファイル形式）とよばれるデータフォーマットに変更する。それを学習用データセットと評価用データセットに変換したあと学習を行う。このとき、一度に学習させるデータの数、学習を行う回数、ノード数を設定する。これをパッケージ化し、MIDI ファイルとして新たに楽曲を生成するという流れである。これを図 2.1 に示す。



2.2 開発環境の構築

開発環境の構築には二つ方法があり, 一つは Docker というコンテナ型の仮想化環境を構築できるオープンソースソフトウェアを用いる方法と, ローカル環境に Python のパッケージ管理システムである pip を用いて構築する方法の 2 つがある.

Docker を用いることで DockerHub という仮想化環境をクラウド上で共有できるサービスを使用できるため, 短時間でパッケージのインストールをおこなわずに環境を構築できる. しかし, Docker に GPU を割り当てることは Docker に精通していないと難しいため, 本研究ではローカル環境に開発環境を構築した. 本システムの開発環境を表 2.1 に示す.

表 2.1: 開発環境

OS	OS X Yosemite
CPU	Intel Core i5
メモリ	8GB
使用ライブラリ	TensorFlow , magenta

本システムは Macbook Pro を使用し OS は OS X Yosemite を使用した. 使用するライブラリとして, ニューラルネットワークを構築できる Tensorflow を利用した.

第3章 実験内容

3.1 モデルによる違い

本研究では Magenta で用意されている 2 つの学習モデルを用いた。以下に示すモデルを用いて楽曲制作をそれぞれ行い，比較，検証する。

3.1.1 MelodyRNN

MelodeRNN は楽曲のメロディを制作するモデルである。MelodyRNN である 3 つのモデルを以下に示す。

(1) basic_rnn

前の状態を保持し，これを記憶または忘却する。時系列を学習することにより，次の音の予測を可能にしている。Lookback_rnn と Attention_rnn はこれを基に機能を追加したものである。

```
root@2e6500d0c5d2:/magenta-data# convert_dir_to_note_sequences \
> --input_dir=/magenta-data/basemelody \
> --output_file=/tmp/notesequences.tfrecord \
> --recursive
```

図 3.1: NoteSequence の作成

(2) lookback_rnn

Basic_rnn を基に，1 小節前と 2 小節前の音，拍数，前の小節の繰り返しかどうかの情報を与え，音楽の流れを掴もうとするもの。

(3) attention_rnn

basic_rnn を基に，過去の情報を予測結果に加えてこれによる繰り返しを捉えるもの。

3.1.2 PolyphonyRNN

複数の同時音のモデリングが可能になっており，複数音の響きを1つのかたまりとして捉えて学習しているモデルである．このモデルを使用することで，伴奏も含めた楽曲の生成が可能である。

上記のモデルを用いて制作を行い，それぞれの違いと有用性について検証する．

3.2 学習回数による違い

学習回数を変更して楽曲制作を行い，それぞれの違いと有用性について検証する．

3.3 ノード数による違い

ノード数を変更して楽曲制作を行い，それぞれの違いと有用性について検証する．

第4章 楽曲制作

4.1 NoteSequence について

NoteSequence とは MIDI データから作成されるプロトコルバッファである。プロトコルバッファとは Google が 2008 年にオープンソース化したバイナリベースのデータフォーマットである。既存の技術としては XML や JSON などのテキストベースのデータフォーマットがあるが、プロトコルバッファはバイナリフォーマットであるので、アプリケーション間でデータ構造の送受信をする際に少ないデータ量ですむという特徴がある。

4.2 NoteSequence の作成

NoteSequence の作成は図のコマンドで作成できる。

```
root@2e6500d0c5d2:/magenta-data# convert_dir_to_note_sequences \  
> --input_dir=/magenta-data/basemelody \  
> --output_file=/tmp/notesequences.tfrecord \  
> --recursive
```

図 4.1: NoteSequence の作成

`--input_dir` で学習させる MIDI データの絶対パスを指定し、`--output_file` で Notesequence の出力先のディレクトリを指定する。

次に作成した NoteSequence のデータセットを学習用と評価用に分割するために, 下記のコマンドを実行する.

```
root@2e6500d0c5d2:/magenta-data# melody_rnn_create_dataset \  
> --config=basic_rnn \  
> --input=/tmp/notesequences.tfrecord \  
> --output_dir=/tmp/melody_rnn/sequence_examples \  
> --eval_ratio=0.10
```

図 4.2: NoteSequence を学習用と評価表に分割

`--config` で使用する RNN を指定する.`--eval_ratio` で Notesequence の 10 % が評価のためのデータになり, 残りの 90 % が学習用のデータになる.

4.2.1 学習の開始

作成した NoteSequence から学習モデルを作成するために下記のコマンドを実行する.

```
root@2e6500d0c5d2:/magenta-data# melody_rnn_train \  
> --config=basic_rnn \  
> --run_dir=/tmp/melody_rnn/logdir/run1 \  
> --sequence_example_file=/tmp/melody_rnn/sequence_examples/training_melodies.tfrecord \  
> --hparams="batch_size=64,rnn_layer_sizes=[64,64]" \  
> --num_training_steps=500
```

図 4.3: BasicRNN を使用した学習の開始

`--config` で学習に使用する学習モデルを指定,`--rundir` で,`--sequence_examplefile` で学習のために用意した Notesequence を指定する. `--hparams` でメモリの使用量を指定し,`--rnn_layer_size` で中間層のノード数を指定し,`--num_trainingsteps` で学習回数を設定する.

4.2.2 音楽データの作成

下図のコマンドで学習モデルに入力する 音調を MIDI の形式で 指定し音楽データを作成する.

```
root@2e6500d0c5d2:/magenta-data# melody_rnn_generate \  
> --config=attention_rnn \  
> --run_dir=/tmp/melody_rnn/logdir/run1 \  
> --output_dir=/tmp/melody_rnn/generated \  
> --num_outputs=10 \  
> --num_steps=128 \  
> --hparams="batch_size=64,rnn_layer_sizes=[64,64]" \  
> --primer_melody="[60]"
```

図 4.4: 学習モデルを使用し,10 曲を作成

第5章 結論

5.1 今後の課題

aaa

謝辭

参考文献

[1]