Package inputenc Error: Unicode character (U+0008)not set up for use with LaTeXSee the inputenc package documentation for explanation.You may provide a definition with 曲を作成 4.47

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

図 目 次

表 目 次

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 によって商品化され，現在ではスマートフォンにも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．
また囲碁や将棋などの競技においても，プロに勝利するなどその精度は高くなっており，その成長は著しい．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GitHab上に公開している．Magentaではまず学習させたい音楽のMIDIデータをファイルに格納しNoteSequence（magentaが扱うファイル形式）に変更する．それを学習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．

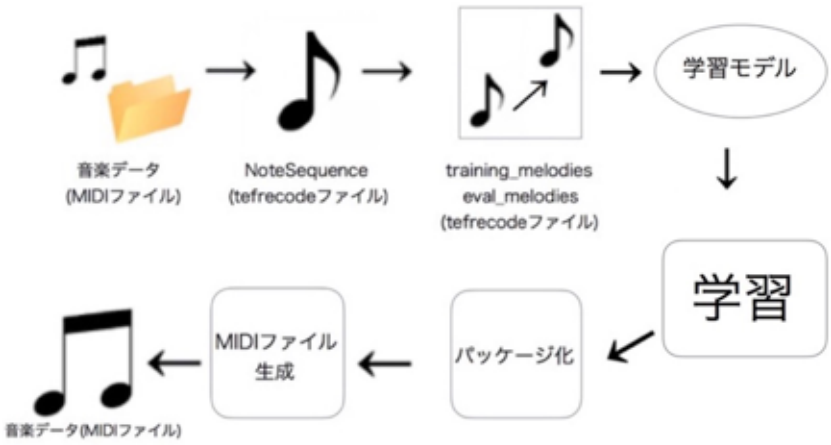

図 2.1: magenta による MIDI 音楽データ生成までのプロセス

2.2 開発環境の構築

開発環境の構築にはコンテナ型仮想環境を提供するオープンソフトウェアである Docker を用いた.

Docker には仮想環境を配布可能な形にする事ができる DockerImage があり, その Image を用いる事で同一の実行環境が作成できる. また, クラウド上で DockerImage を配布できる DockerHub というサービスがあり, そのサービス上にすでに Magenta の開発環境を構築済みの仮想環境があるため、その環境を今回は利用した.

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$_r nn$

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．$Lookback_r nn$ と $Attention_r nn$ はこれを基に機能を追加したものである．

(2)$lookback_r nn$

$Basic_r nn$ 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)$attention_r nn$

$basic_r nn$ 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。

3.2　学習回数による違い

3.3　ノード数による違い

第4章　楽曲制作

4.1　NoteSequenceの作成

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 は 2008 年にオープンソース化したスキーマ言語であり, スキーマ言語とは. Magenta は Python で作成されているので, 実行するには Python ファイルを指定し実行するか, 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

–inputdir で学習させる MIDI データの 絶対パスを指定し,–outputfile 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.–evalratio 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.

4.1.1 学習の開始

```
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,–sequenceexamplefile 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,–rnnlayersize で中間層のノード数を指定し,–numtrainingsteps で学習回数を設定する.

```
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

4.2 PolyfonyRNN

第5章　結論

5.1　今後の課題

aaa

謝辞

参考文献

[1]