

平成 30 年度 卒業論文

AI を用いた音楽生成の研究

函館工業高等専門学校 生産システム工学科 情報コース

3 番 小川 桃佳

指導教員 東海林 智也

目次

第1章 序論

- 第1節 英文アブストラクト
- 第2節 研究目的
- 第3節 研究背景
- 第4節 開発環境

第2章 AI を用いた音楽生成システムの開発

- 第1節 音楽生成プログラムの概要
- 第2節 音楽再生プログラムの概要
- 第3節 midiOut 関数
- 第4節 教師データ
- 第5節 出力ファイル

第3章 結果

第4章 課題

参考文献

第 1 章 序論

第 1 節 英文アブストラクト

It is a purpose to generate music using deep learning which is one of AI techniques.

In this research, development was done in c++ environment using Visual Studio. Tiny-DNN was used as the AI library. Create training data in the CSV file, load it and do learning. Similarly, output can be done with a CSV file, and the generated music can be played by loading the outputted CSV file.

In conclusion, although I was able to produce music, it is not accurate and I have to think about improving accuracy. Lack of training data and input perceptron is considered as a cause.

Key words : AI, deep learning, Tiny-DNN,

第 2 節 研究目的

本研究では唯一 C 言語で深層学習ができるライブラリを用いて、楽曲を手軽に生成できるようにすることを目的とする。

第 3 節 研究背景

音楽知識のある人向けの作曲ツールが多く存在する中で、音楽知識のない人でも作曲したい曲の特徴を入力し出力するツールを見かけなかったことでこの開発に至った。

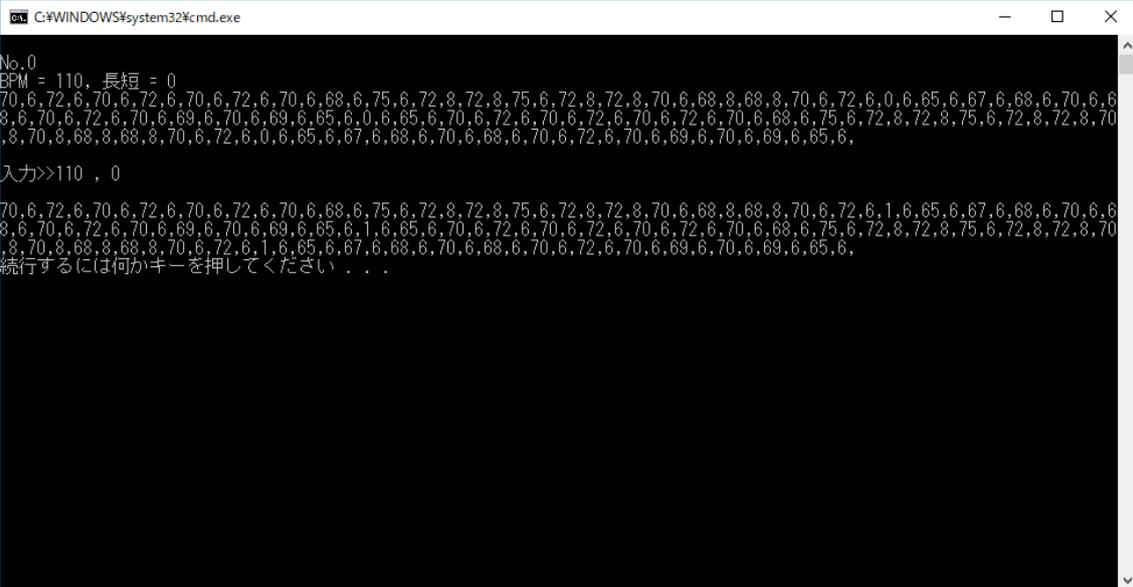
第 4 節 開発環境

| | |
|---------|------------------------------|
| 使用 OS | Windows10 Education |
| 使用ソフト | Microsoft Visual Studio 2017 |
| 使用ライブラリ | Tiny-Dnn[1] |
| 使用言語 | C++ |

第2章 AI を用いた音楽生成システムの開発

第1節 音楽生成プログラムの概要

このプログラムは入力パーセプトロンに任意の BPM と作曲者の識別番号の2つのデータを入力することで、30秒程度の楽曲をニューラルネットワークから出力させる機能を持つ(図1)。

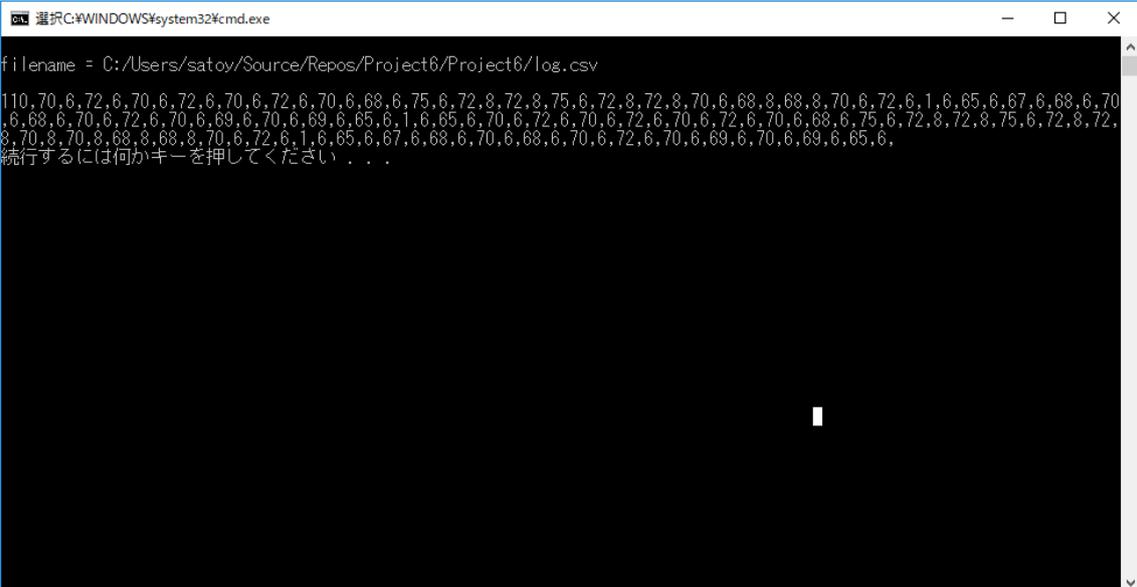


```
C:\WINDOWS\system32\cmd.exe
No.0
BPM = 110, 長短 = 0
70,6,72,6,70,6,72,6,70,6,72,6,70,6,68,6,75,6,72,8,72,8,75,6,72,8,72,8,70,6,68,8,68,8,70,6,72,6,0,6,65,6,67,6,68,6,70,6,6
8,6,70,6,72,6,70,6,69,6,70,6,69,6,65,6,0,6,65,6,70,6,72,6,70,6,72,6,70,6,72,6,70,6,68,6,75,6,72,8,72,8,75,6,72,8,72,8,70
,8,70,8,68,8,68,8,70,6,72,6,0,6,65,6,67,6,68,6,70,6,68,6,70,6,72,6,70,6,69,6,70,6,69,6,65,6,
入力>>110, 0
70,6,72,6,70,6,72,6,70,6,72,6,70,6,68,6,75,6,72,8,72,8,75,6,72,8,72,8,70,6,68,8,68,8,70,6,72,6,1,6,65,6,67,6,68,6,70,6,6
8,6,70,6,72,6,70,6,69,6,70,6,69,6,65,6,1,6,65,6,70,6,72,6,70,6,72,6,70,6,72,6,70,6,68,6,75,6,72,8,72,8,75,6,72,8,72,8,70
,8,70,8,68,8,68,8,70,6,72,6,1,6,65,6,67,6,68,6,70,6,68,6,70,6,72,6,70,6,69,6,70,6,69,6,65,6,
続行するには何かキーを押してください...
```

図1 音楽生成画面

第 2 節 音楽再生プログラムの概要

このプログラムは出力結果の楽曲を CSV ファイルに変換して保存し、その CSV ファイルから `midiOut` 関数を用いて音声に変換してスピーカーから出力する (図 2)。



```
選択C:\WINDOWS\system32\cmd.exe
filename = C:/Users/satoy/Source/Repos/Project6/Project6/log.csv
110,70,6,72,6,70,6,72,6,70,6,72,6,70,6,68,6,75,6,72,8,72,8,75,6,72,8,72,8,70,6,68,8,68,8,70,6,72,6,1,6,65,6,67,6,68,6,70
,6,68,6,70,6,72,6,70,6,69,6,70,6,69,6,65,6,1,6,65,6,70,6,72,6,70,6,72,6,70,6,72,6,70,6,68,6,75,6,72,8,72,8,75,6,72,8,72,
8,70,8,70,8,68,8,68,8,70,6,72,6,1,6,65,6,67,6,68,6,70,6,68,6,70,6,72,6,70,6,69,6,70,6,69,6,65,6,
続行するには何かキーを押してください . . .
```

図 2 音楽生成画面

第3節 midiOut 関数

生成した楽曲の音を出す関数として midiOut 関数を使用した。下記の三行で簡単に音を出すことができる。

```
midiOutShortMsg(h, 0x7f●●90);  
Sleep(1000);  
midiOutShortMsg(h, 0x7f●●90);
```

1 行目の下 2 ケタを 90 にすると ON で音になる。2 行目の Sleep の中はミリ秒表記で音の長さを表しており、この Sleep 処理がなければ音を出すことができない。3 行目の下 2 ケタを 80 にすると OFF になり音を止めることができる。

1 行目と 3 行目の●●には表 1 に記載されている対応した音のノート番号を 16 進数に直した値を入力する [2]。また、0x の後の 7f は 16 進数に直したベロシティ(鍵盤を押す速さ)の値で、実験では一番早い 127 に設定した。

例) オクターブ 4

| | | | | | | |
|-----|----|----|----|-----|----|-----|
| コード | C | #D | D | b E | E | ... |
| ノート | 60 | 61 | 62 | 63 | 64 | ... |

表 1. 音階表

第5節 出力ファイル

生成された曲は CSV ファイルで曲の速さ (BPM) を 1 つ、midiOut 関数で使用する鍵盤の番号 (ノート番号) ・音符の長さ (4 分音符 → 4) を音符 90 個分の計 181 個の値をカンマ (,) 区切りで出力させた (図 4)。

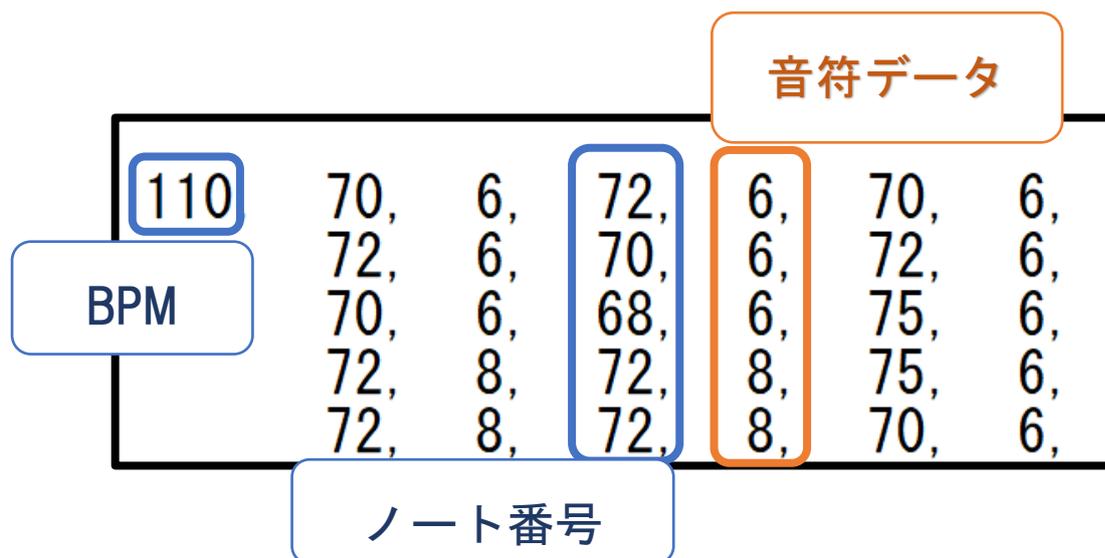


図 1 出力データ

第3章 結果

試作として教師データを1つだけ学習させたものから出力させたところ、教師データとほぼ同一のものが出力された(図5,6)。

| | | | | | | | |
|------|----|-----|----|-----|----|-----|----|
| 110, | 0, | 70, | 6, | 72, | 6, | 70, | 6, |
| | | 72, | 6, | 70, | 6, | 72, | 6, |
| | | 70, | 6, | 68, | 6, | 75, | 6, |
| | | 72, | 8, | 72, | 8, | 75, | 6, |
| | | 72, | 8, | 72, | 8, | 70, | 6, |
| | | 68, | 8, | 68, | 8, | 70, | 6, |
| | | 72, | 6, | 0, | 6, | 65, | 6, |
| | | 67, | 6, | 68, | 6, | 70, | 6, |
| | | 68, | 6, | 70, | 6, | 72, | 6, |
| | | 70, | 6, | 69, | 6, | 70, | 6, |
| | | 69, | 6, | 65, | 6, | 0, | 6, |
| | | 65, | 6, | 70, | 6, | 72, | 6, |
| | | 70, | 6, | 72, | 6, | 70, | 6, |
| | | 72, | 6, | 70, | 6, | 68, | 6, |

図5 BPM110の教師データ

| | | | | | | |
|------|-----|----|-----|----|-----|----|
| 110, | 70, | 6, | 72, | 6, | 70, | 6, |
| | 72, | 6, | 70, | 6, | 72, | 6, |
| | 70, | 6, | 68, | 6, | 75, | 6, |
| | 72, | 8, | 72, | 8, | 75, | 6, |
| | 72, | 8, | 72, | 8, | 70, | 6, |
| | 68, | 8, | 68, | 8, | 70, | 6, |
| | 72, | 6, | 1, | 6, | 65, | 6, |
| | 67, | 6, | 68, | 6, | 70, | 6, |
| | 68, | 6, | 70, | 6, | 72, | 6, |
| | 70, | 6, | 69, | 6, | 70, | 6, |
| | 69, | 6, | 65, | 6, | 1, | 6, |
| | 65, | 6, | 70, | 6, | 72, | 6, |
| | 70, | 6, | 72, | 6, | 70, | 6, |
| | 72, | 6, | 70, | 6, | 68, | 6, |

図 6 出力データ

次に BPM の値が異なる 2 つの教師データを学習させ、入力する BPM の値を片方のデータに近い値にして出力したところ片方のデータに偏った楽曲が生成された(図 7~9)。

| | | | | | | | |
|------|----|-----|-----|-----|-----|-----|----|
| 110, | 0, | 70, | 8, | 72, | 8, | 70, | 8, |
| | | 72, | 8, | 70, | 8, | 72, | 8, |
| | | 70, | 8, | 68, | 8, | 75, | 8, |
| | | 72, | 16, | 72, | 16, | 75, | 8, |
| | | 72, | 16, | 72, | 16, | 70, | 8, |
| | | 68, | 16, | 68, | 16, | 70, | 8, |
| | | 72, | 8, | 0, | 8, | 65, | 8, |
| | | 67, | 8, | 68, | 8, | 70, | 8, |
| | | 68, | 8, | 70, | 8, | 72, | 8, |
| | | 70, | 8, | 69, | 8, | 70, | 8, |
| | | 69, | 8, | 65, | 8, | 0, | 8, |
| | | 65, | 8, | 70, | 8, | 72, | 8, |
| | | 70, | 8, | 72, | 8, | 70, | 8, |
| | | 72, | 8, | 70, | 8, | 68, | 8, |

図 7 BPM110 の教師データ

| | | | | | | | |
|------|----|-----|----|-----|----|-----|----|
| 210, | 0, | 73, | 4, | 73, | 4, | 73, | 8, |
| | | 71, | 8, | 69, | 8, | 69, | 8, |
| | | 76, | 4, | 71, | 8, | 73, | 5, |
| | | 73, | 4, | 71, | 8, | 69, | 8, |
| | | 71, | 8, | 69, | 8, | 71, | 8, |
| | | 71, | 8, | 73, | 8, | 73, | 8, |
| | | 71, | 4, | 69, | 8, | 69, | 4, |
| | | 0, | 8, | 71, | 4, | 73, | 4, |
| | | 73, | 4, | 73, | 8, | 71, | 8, |
| | | 69, | 8, | 69, | 8, | 76, | 4, |
| | | 71, | 8, | 73, | 5, | 73, | 4, |
| | | 71, | 8, | 69, | 8, | 71, | 8, |
| | | 69, | 8, | 71, | 8, | 0, | 8, |
| | | 73, | 8, | 73, | 8, | 71, | 8, |
| | | 71, | 8, | 69, | 8, | 69, | 8, |

図 8 BPM210 の教師データ

| | | | | | | |
|------|-----|----|-----|----|-----|----|
| 110, | 70, | 6, | 72, | 6, | 70, | 7, |
| | 72, | 7, | 70, | 7, | 72, | 7, |
| | 70, | 5, | 68, | 7, | 75, | 6, |
| | 72, | 6, | 72, | 8, | 75, | 6, |
| | 72, | 8, | 72, | 8, | 70, | 7, |
| | 68, | 8, | 68, | 8, | 70, | 6, |
| | 72, | 5, | 28, | 7, | 65, | 6, |
| | 42, | 7, | 68, | 6, | 70, | 6, |
| | 68, | 6, | 70, | 7, | 72, | 7, |
| | 70, | 7, | 69, | 7, | 70, | 6, |
| | 69, | 7, | 65, | 6, | 32, | 5, |
| | 65, | 7, | 70, | 7, | 72, | 7, |

図 9 教師データを 2 つ学習した出力データ

更にデータの数を8つに増やし BPM 入力を 60, 120, 180 に、作曲者識別番号を 0、0.5、1 に固定した後、それぞれで比較を行った。作曲者識別番号の 0.5 については、同作曲者だけではなく異なった作曲者の特徴を合わせて出力させるために設定した。

結果は、作曲者識別番号が 0 の場合は 0 の作曲データの中から入力した BPM に近い値を持つデータに偏って参照された。識別番号 1 の場合も同様の結果が得られた。また、識別番号 0.5 の場合は、0 と 1 どちらにも偏りのないデータが生成された。

2つ以上の教師データを学習して出力された楽曲は、いずれも異なった曲が混ざっているため不協和音が多くみられた。

第4章 課題

今後の課題として以下のものが考えられる。

- ・教師データの総数を増やす

教師データの総数が8つでは望んだデータが出力出来なかったため、学習させる数を増やす必要がある。

- ・教師データの作成法則の変更

今までに作成した教師データは、音符の総数を揃えて作成していたが今回の結果から1小節ごとに作成するのが望ましいと考えられる。

- ・入力パーセプトロンを増やす

現段階ではBPMと作者識別番号の2つのみであったため、メロディー・リズムなどの判断材料を増やすことで精度を高くする。

参 考 文 献

[1] Tiny-Dnn 配布元,

<https://github.com/tiny-dnn>

[2] midi ノート番号表

<http://yamatyuu.net/computer/program/vc2013/midi/note.html>