

令和2年度 卒業研究

AIによる楽曲の自動編曲に関する研究
Research on Automatic Arrangement of Music using AI

函館工業高等専門学校

生産システム工学科 情報コース

5年 秋田 浩杜

序論

第1節 英文アブストラクト

In recent years, composition by AI using deep learning has been realized in various ways. The compositions in these technologies do not require musical knowledge at the time of design, but are created by AI trained with the data necessary for learning and the network that performs the learning. Focusing on this, the objective of this research is to enable a human without musical knowledge to arrange the target music based on the training data by simply specifying the genre and artist.

Key words : AI, deep learning, ,arrange

第2節 研究背景

近年、ディープラーニングを用いた AI による作曲は様々な形で実現されている。ユーザーは音楽知識を必要とせず、学習に必要なデータと学習を行うネットワークによって訓練された AI によって楽曲が自動作成される。

第3節 研究目的

先行研究によって開発された作曲技術を基に、音楽知識のない人間がジャンルやアーティストを指定するだけで既存楽曲を自動で編曲できるアプリケーションを作成することを本研究の目的とする。

第4節 開発環境

本研究での開発環境は以下の通りである。

第1項 コンピュータ

OS: Windows10 Home 64bit

CPU: Ryzen ThreadRipper 2920X 3.5GHz

RAM: 32GB

GPU: AMD Radeon RX5700

第2項 開発言語

Python 3.8.5

第3項 依存ライブラリ

keras 2.2.4

h5py 2.10.0

music21 6.7.0

mumpy 1.19.2

plaidml 0.9.0

第 5 節 データセット

既存楽曲を編曲するためには学習データセットが必要となる。本研究では、Classical Piano[1]にて公開されているショパンの楽曲データを使用した。なお学習データの形式は 1 つの楽器のみでメロディが構成される MIDI ファイル形式とした。

関連技術

第1節 LSTM (Long Short Term Memory)

LSTM[2]はリカレントニューラルネットワーク(以下 RNN)の一つである。

従来の RNN は、ネットワークからの出力とネットワークへの入力を「 \tanh 層」と呼ばれる層に掛け合わせていくことを繰り返す単純なものであった。

一方 LSTM は、「 \tanh 層」以外にも相互に作用する「忘却ゲート層」「入力ゲート層」「シグモイド層」と呼ばれる層を持っており、それぞれの層は「入力の取捨選択」、「更新する情報の選択」、「出力の決定」の役割を担当している。

図1に各ニューラルネットワークの違いを示す。

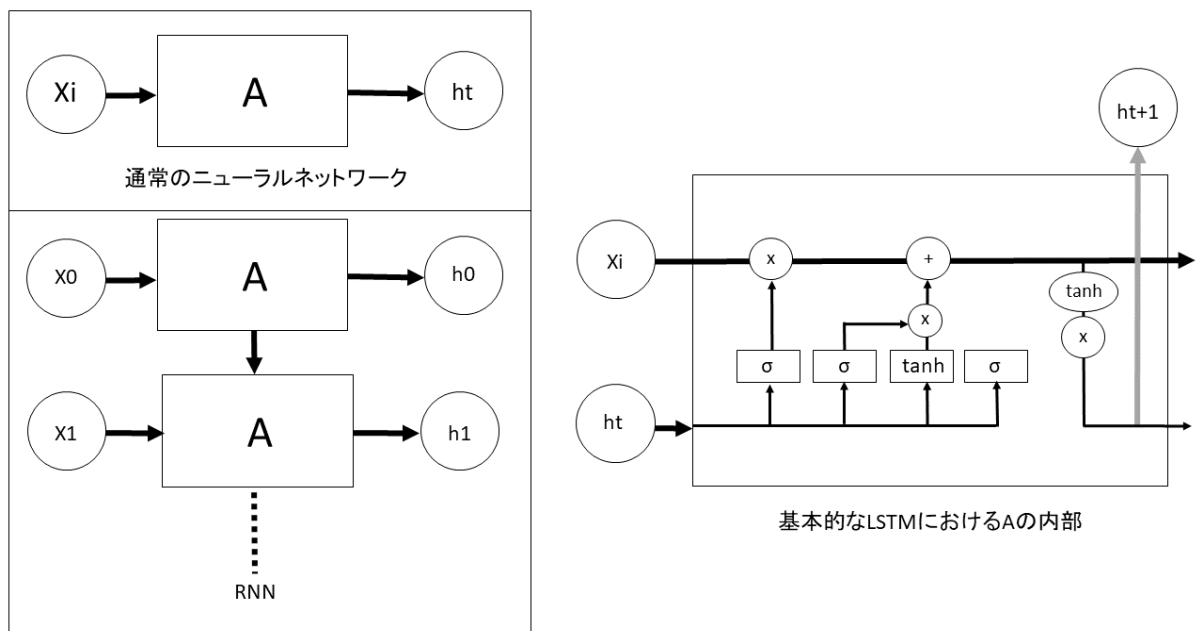


図1 通常のニューラルネットワーク、RNN、LSTM の違い

第 2 節 GAN (Generative Adversarial Networks)

GAN は、「生成器」「識別器」の 2 層から構成されるニューラルネットワークである。ある成功イメージに基づき、生成器が成功イメージに寄せた偽物のイメージを作成する。識別器は生成器が作成した偽物のイメージを成功イメージと比較し、その真偽を判定する。これらの相反する 2 つの機能を相互に動作させることで、生成器、識別器ともに精度を上げていくことができる。

第 3 節 PlaidML

PlaidML[3]は、近年機械学習で取り上げられている GPU を用いた学習を行うライブラリである。現在、GPU を用いた機械学習は殆どが NVIDIA 製の GPU 及び CUDA を用いた環境であり、AMD 製の GPU は機械学習のサポートがあまりされていなかった。しかし PlaidML を用いれば AMD、Intel 両方の GPU 及び OpenCL の環境下においても GPU を用いて機械学習を行うことができるので、本研究では本ライブラリを採用した。

プログラムの開発

第1節 楽曲アレンジのフロー

本研究では以下のフローで楽曲の自動編曲を行う。

1. 編曲を行いたい楽曲データ(以下「対象データ」)を用意する
2. 指定したジャンルアーティストの楽曲データ(以下「学習データ」)を用意する
3. LSTM を用いて楽曲を自動生成する
4. LSTM により自動生成した楽曲と対象データを GAN の生成器に、学習データを GAN の識別機に入力して編曲した楽曲を得る。

以上のフローを図2に示す。

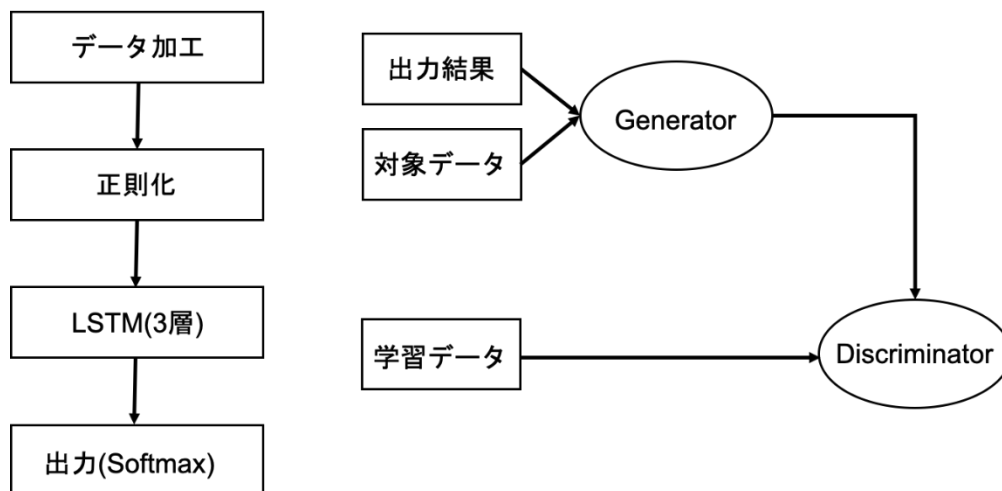


図2 楽曲編曲のフロー

第2節 データセットの加工

機械学習を行う際には MIDI ファイルに含まれるバイナリデータをそのまま入力として扱うことはできないので、`Music21`[4] というライブラリを用いてバイナリデータと数値データを相互に変換した。具体時には、入力前に

MIDI ファイル → 音階を用いた文字列表現 → 音階と数値を対応付けた数値表現

という流れでデータを加工してデータセットを得た。一方、出力を得る際には逆の手順で

音階と数値を対応付けた数値表現 → 音階を用いた文字列表現 → MIDI ファイル

という流れでデータを加工して MIDI ファイルを得た。

実験

第1節 モデル設定

開発したアプリケーションを用いて童謡の「きらきら星」をショパン風に編曲する。今回の実験で用いた LSTM の設定は以下の通りである。

- ・層数 10 層
- ・学習データ数 93 曲
- ・エポック数 200

また GAN の設定は以下の通りである。

- ・層数 2 層
- ・対象データ 「きらきら星」
- ・学習データ 「ショパンの楽曲データ」

第2節 実験結果

図3に実験結果として得られた楽曲の楽譜を示す。生成された楽曲を再生したところ、所々に曲として理解できる部分があるが、ほとんどは音階に規則性のないものとなってしまった。

The musical score consists of 24 measures of music. It is written in a bass clef with a key signature of two flats (B-flat and E-flat) and a 4/4 time signature. The tempo is marked as ♩ = 120. The notation includes various note values, rests, and dynamic markings such as accents and hairpins. The measures are numbered 1 through 24.

図3 実験結果の楽譜

第3節 考察

今回の実験を実施するにあたり、まず先行研究を基に LSTM と GAN を別々に動作させた際にはそれぞれは上手く動作していた。しかし LSTM と GAN を組み合わせると曲と認識できる部分が少なくなり、対象データに似ている楽曲を生成するという実験目的が全く実現できなかった。

したがって、LSTM で生成した楽曲をそのまま GAN に流用する前に、何らかの処理をおこなう必要があると思われる。

まとめ

本研究では、音楽知識のない人間がジャンルやアーティストを指定するだけで既存楽曲を自動で編曲できるアプリケーションを作成することを目指した。その目的のために LSTM と GAN を組み合わせて用いることを考えた。

しかし単純に LSTM と GAN を組み合わせただけでは、ほとんど音階に規則性のない楽曲が生成されてしまった。

そこで今後、LSTM で生成した楽曲をそのまま GAN に流用する前に何らかの処理をおこなったり、思い切って LSTM や GAN 以外の手法を用いたりすることについて検討する。

参考文献

[1] <http://www.piano-midi.de/>

[2] <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

[3] <https://www.intel.com/content/www/us/en/artificial-intelligence/plaidml.html>

[4] <http://web.mit.edu/music21/>