

LSTM を用いた繰り返し構造を持ったメロディ生成による 作曲支援システム

Composition support system by generating melody with repetitive structure using LSTM

河野 翔太

Shota Kawano

法政大学情報科学部デジタルメディア学科

学籍番号:16K1111

shota.kawano.5k@stu.hosei.ac.jp

abstract

One of the important elements of a song is a melody, which is created by modifying it little by little. In order to automate this work, since the music has a repetitive structure, it is considered that a similar melody that appeared by automatically generating a melody with a repeating structure and a melody created by myself can be used as a correction plan. In this research, an automatic melody correction system using an automatically generated music was created. A generated music includes the melody created by myself and uses a machine learning model that can generate music with repetitive structure. Improv RNN is used for learning and generation. A part of an existing music in which a similar melody appears due to the repetition structure is given as an input to the system, and an evaluation is made as to whether a similar melody similar to the existing music can be presented as a correction plan. When a melody was generated using this system, a melody similar to the input appeared due to the repetitive structure, and it can be presented as a correction plan.

1 はじめに

作曲ではメロディやコード、ドラムなどのリズムを中心に様々なパートを制作必要がある。リズムとコード進行に関してはほとんど決まった形があるが、メロディの正解は1つではなく、少しずつ修正しながら作っていくことがある。

実際に DAW(Digital Audio Workstation) を使用してメロディを作っていく際には、演奏しながらピアノロール上で修正していく。この修正例としては、リズムをずらす、休符を加える、刻んで音程をずらす、メロディの駆け上がりや下がりを加えるなどがある。これらの作業は初心者が躓きやすい部分であったり、今まで作ってきたメロディと似たようなものにならず、躓いてしまうこともある。このような修正作業を自動化する支援システムを作成したい。

まず、どのような観点で修正するのかを考えた。ジャンルによっては、繰り返しの構造を持っている楽曲が多いため、この特徴を利用出来るのではないかと考えた。既存楽曲の中には、4小節同じメロディを繰り返していき、最後の1小節だけを変えて次の展開に繋げるものがある。そこで、本研究では、繰り返しの構造を学習し生成出来る自動作曲を用いて、自分で作成したメロディを含んだ楽曲を自動で生成し、その繰り返しの構造を持つ出力結果からメロディを修正するシステムを作成する。

2 自動作曲の従来研究

機械学習を用いた自動作曲システムの研究は多く行われている。その中には、Recurrent Neural Network(RNN)、ま

たはこれを発展させた Long short-term memory(LSTM) や、Generative Adversarial Network(GAN) などがある。RNN は時系列データを扱えるニューラルネットワークであり、RNN において長期依存の時系列データを扱えるようにしたものが LSTM である。GAN は敵対的生成ネットワークと呼ばれるもので、2014 年に発案されたものである。生成ネットワークと識別ネットワークと呼ばれる 2 つのネットワークから構成されており、この 2 つのネットワークを競合させることによって学習が進められ、教師なし学習が出来る。従来研究によると、機械学習によって楽譜のメロディのパターンとコード進行の構造を学習できる。RNN による自動作曲 [1] では楽譜の生成はできたが、RNN の勾配消滅問題により、短期依存のデータしか学習出来ないことから音楽的構造を学習することが出来ず、音楽として成立していないものしか出来なかった。しかし、LSTM の登場により長期依存の時系列データの学習が可能になったことで、ブルースの構造 [2] やバッハの聖歌 [3]、ポリフォニー音楽 [4] などを学習させることで RNN では出来なかった音楽理論に基づいた自動作曲が出来た。GAN と RNN を組み合わせた研究 [5] や GAN と Convolutional Neural Network(CNN) を組み合わせた研究 [6] でも音楽理論に基づいた曲を自動生成することに成功している。

紹介してきた従来研究では MIDI(Musical Instrument Digital Interface) ファイルを使用している。この MIDI ファイルは、どの音がどのぐらいの長さ鳴っているのか、音量はどうなっているかなど、演奏情報を格納できるファイル形式で、様々な機器やソフトウェアで使用でき、DAW でも入出力が可能なファイル形式である。また、Google はニューラルネットワークを使って美術や音楽を自動生成する Magenta[7] と呼ばれるプロジェクトを行っており、誰でも簡単に学習出来るように様々なモデルを公開している。他にも音声データからスペクトルを特徴量として取り出して学習し、音楽を生成する研究 [8] や、音声データを μ -low というエンコーディングで 8 ビットの情報に変換して学習する研究 [9] や、強化学習 (RL) を用いた研究 [10] もある。

3 提案手法

従来研究の複数のシステムを使用した所、Magenta の ImprovRNN では自分で作成したメロディを入力として与えることで楽譜を生成することができ、繰り返しの構造が現れやすく、様々なパターンを生成可能だったことから、このシステムを使用して、修正システムを作成していく。楽曲の繰り返しには様々なパターンがあるが、本研究では 4 小節同じメロディを繰り返していき、最後の 1 小節だけを変える場合に絞る。ImprovRNN で出力されたメロディのパターンがとても多く、中には休符が多いものや全音符しかないものなどもあり、生成結果を絞る必要がある。そこで、生成結果を絞るために既存楽曲のメロディの類似度からモデルを作成した。

3.1 ImprovRNN

Magentaにて公開されている学習モデルの1つとして、コードを見ながらメロディを学習することが出来る Improv RNN と呼ばれるものがある。MusicXML ファイルで学習を行い、次にどの音が来るのかを学習する。メロディはベクトルとして表現されており、最小単位が16分音符のワンホットベクトルである。1つ目の要素は、メロディが鳴っていない時に1となり、2つ目の要素は前のメロディが鳴り続けている時に1となり、3つ目以降の要素は、MIDI ノートナンバー 48~84(C2からC5)の音が鳴っている時に1となる。コードは MusicXML ではラベルで記載し、このラベルからどの音が鳴っているかをCからBまでの音を表現した12個の要素を持つベクトルで表現している。ただし、1小節につき1つしか与えられない。

$$\begin{bmatrix} 0,0,0,0,0,0,0,0,0,0,0,0, \\ 0,1,0,0,0,0,0,0,0,0,0,0, \\ 0,0,0,0,0,0,0,0,0,0,0,0 \end{bmatrix}$$

図1. メロディのワンホットベクトル表現

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ C & D\flat & D & E\flat & E & F & G\flat & G & A\flat & A & B\flat & B \end{bmatrix}$$

図2. コードCのベクトル表現

この学習モデルにおいて、ネットワークはRNNを使用しているが、数ステップ前までに出力された情報を、入力として扱うことに特徴がある。このようなネットワークにすることで、数小節前の情報を参照しながら楽譜を生成していくことが可能となる。ネットワークの図と各係数の式は下のようになる。 w_1, w_2, w_3 は学習するパラメータであり、 h_i は n ステップ前までの出力 (h_{t-n} から h_{t-1})、 h'_t は時刻 t の出力となっている。本研究では、32ステップ前までの情報、つまり2小節前までの情報を参照している。中間層の情報は、次の時系列へ重みをつけて引き渡される。

$$u_i^t = w_3^T \tanh(w'_1 h_i + w'_2 c_t) \quad (1)$$

$$a_i^t = \text{softmax}(u_i^t) \quad (2)$$

$$h'_t = \sum_{i=t-n}^{t-1} a_i^t h_i \quad (3)$$

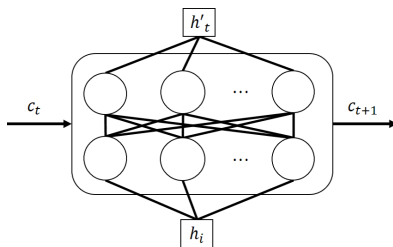


図3. ネットワーク構成図

ネットワークへの入力メロディのベクトルとなっており、出力 h'_t はどの音が来やすいか確率として出力される。この確率からふさわしいメロディを選択する。誤差を求めるには目的関数として下の式のような交差エントロピーを使用する。この交差エントロピーは、行列を扱う学習において誤差を計算するのに適している。

$$E_i = -t_i \ln(y_i) - (1 - t_i) \ln(1 - y_i) \quad (4)$$

中間層の数、バッチサイズ、学習回数などは実行の度に指定する。デフォルトでは、64個の中間層が2つあるネットワークで、バッチサイズは64で学習回数は20000となっており、この設定で学習されたモデルを使用した。

メロディとコードを入力とし、メロディと1小節につき1つのコードを出力する。こちらで楽曲の生成をする際には、メロディの一部と全体のコードを入力として与える。メロディは音価を考慮した音高のベクトルとして入力する。このシステムを使って、学習済みモデルから楽曲を生成した。学習データには、Wikifonia[11]と呼ばれるサイトから入手した MusicXML ファイルが使用されている。約6000個の MusicXML ファイルがあり、ポップス、クラシックなど複数のジャンルが混在している。図5の自ら作成したメロディを入力として与えた。これをベクトル化したものは、 $[67, -2, 67, -2, 65, -2, 67, -2, 70, -2, -2, -2, 67, -2, 74, -2]$ となり、コードは、 $E\flat F Gm F E\flat F Gm F$ を入力として与えた。繰り返しの構造を持つような楽曲も生成されたが、一部は繰り返しの構造がない楽曲も出力されてしまうので、入力に似たメロディが出力されているものだけを取り出す必要がある。



図4. 自ら作成したメロディ



図5. ImprovRNNによって出力された楽譜

3.2 学習データ

学習データは Wikifonia の MusicXML ファイルの他に、TheoryTab[12] と呼ばれる、様々なジャンルの楽譜を MIDI ファイルとして出力できるサイトで収集した。ジャンルは絞らず、様々なアーティストの楽曲の MIDI データを約1万個収集した。この MIDI データに入っているコードは、全て4分音符となっており、音符情報として保存されている。これらの MIDI ファイルは MusicXML ファイルへ変換してから学習で使用される。



図6. 学習データの楽譜例

3.3 類似度の計算

繰り返しの構造を持ったメロディが生成されているかを判断するために、小節同士の類似度を計算する。類似度は、音高、音価、休符の3つに対して計算する。

まず、音高の類似度の計算方法を説明する。メロディを一定の間隔(拍数、秒数)でサンプリングし、1小節から複数の小節の MIDI のノートナンバーが入ったベクトルへ変換する。このベクトル同士でコサイン類似度を(5)式で計算することで類似度を求める。この計算を簡単に行うために、MATLAB の MIDI Toolbox[13]に入っている関数 `meldistance` を使用している。比較方法の名称は、`melcontour` と呼ばれるメロディの輪郭を使うものであり、図7はメロディのサンプリングの例である。初めの1小節目を1/4拍でサンプリングすると、 $[64 \ 64 \ 64 \ 64 \ 71 \ 71 \ 71 \ 71 \ 71 \ 71 \ 69 \ 69]$ というベクトルが得られる。この作業を他の小節に対しても行い、ベクトル間の類似度を求める。

$$\cos(\vec{a}, \vec{b}) = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}} \quad (5)$$

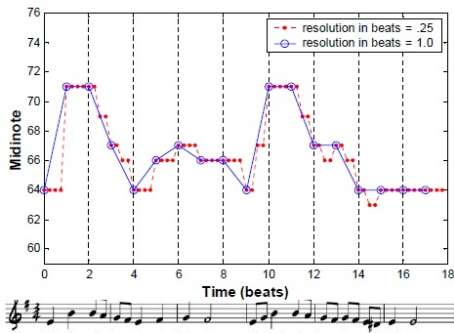


図 7. メロディの輪郭と表記法 [13]

次に、音価の類似度の計算方法を説明する。音価の類似度の計算は、1小節の中にどの音価の音符がどのぐらいの割合出現しているかをベクトル化したものを使用する。全音符から16分音符までの音価に対応した9個の要素を持ったベクトルを作成する。図7の1小節目の音価のベクトルは、4分音符が2つ、8分音符が2つあることから、 $[0\ 0\ 0\ 0\ 0.5\ 0\ 0.5\ 0\ 0]$ のようなベクトルが得られる。この計算には、先程使用したMIDI Toolboxの中に入っているdurdistと呼ばれる比較方法を使用し、コサイン類似度によってベクトル間の類似度を求める。

最後に、休符の類似度の計算方法を説明する。1つの要素が16分音符となっているベクトルで表現し、音符がある時を1、休符がある時を0とする。図7の1小節目は休符がないため、 $[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$ のようなベクトルとなり、休符がある場合、途中の要素が0となる。このベクトル同士の類似度をコサイン類似度によって求める。

3.4 修正システム

修正する際には、まず、自分で作成したメロディをMIDIファイルとして書き出し、ImprovRNNに入力するベクトルへ変換する。生成したい小節数分のコードをラベルとして与えて1000曲分自動生成する。そして、入力したメロディの小節と修正したメロディの小節の類似度を求める。この作業をすべての自動生成された楽曲に対して実行し、類似度が高いものを優先的にユーザーへ提示し、これらの楽譜から修正案として採用するものを選択する。どのようなメロディを提示するべきかは、既存楽曲のメロディの類似度を計算し、どの程度の値が適当か決定した。

3.5 メロディの類似度モデル

小節間の類似度がどの程度の値を取れば、既存楽曲のような繰り返しの構造を持ったメロディを修正案として提示できるか、既存楽曲で学習した。既存楽曲の条件としては、4小節と8小節目が異なり、1から3小節目と5から7小節目がほとんど同じもので、150曲の既存楽曲を使用した。結果は表1のようになり、各ベクトルの類似度から図8のような三次元の共分散行列を計算し、これらの学習から得られた三次元正規分布に従って、取り出す類似度をランダムに決定出来るようにした。

表 1. 既存楽曲のメロディの類似度

	平均値	最大値	最小値	分散
音高	0.271	0.981	-0.646	0.604
音価	0.932	1.000	0.712	0.013
休符	0.926	1.000	0.714	0.012

$$\begin{bmatrix} 0.604 & -0.009 & -0.001 \\ -0.009 & 0.013 & 0.011 \\ -0.001 & 0.011 & 0.012 \end{bmatrix}$$

図 8. 共分散行列

4 実験

本システムを使用して、メロディの修正を実行した。メロディの生成には、WikifoniaのMusicXMLファイルで学習し

たモデルを使用した。評価実験は、モデルの学習に使用していない既存楽曲の7小節を入力として8小節目を生成し、既存楽曲と同じようなメロディを修正案として提示できるかどうかで評価し、生成実験では、自ら作成した4小節目のメロディに対して、8小節目の修正案を提示した。

4.1 評価実験

既存の楽曲の一部を入力として自動生成させることで、既存楽曲と同じような繰り返しの構造を持った楽曲が出力されていることを確認することで評価を行った。今回は、次の展開に繋げるメロディを修正によって作成することを想定し、既存楽曲と同じようなメロディが現れるかを確認した。

評価実験1では、図9の既存楽曲の楽譜を入力として与え、1000個生成したものの中から40個程度提示した所、図10のような既存楽曲と似たメロディを得ることが出来た。図9は既存楽曲の8小節の楽譜、図10は修正案の終わりの4小節を表示している。音高の類似度は0.810、音価の類似度は1.000、休符の類似度は1.000となっている。入力に比べて旋律の形はかなり似ているが、音高が下がったものが出力された。他の出力においても、4小節目より音高が下がったものがほとんどだった。



図 9. 評価実験1に使用した楽譜



図 10. 評価実験1の出力

評価実験2では、図11のような3小節目と7小節は異なるが、繰り返しの構造を持っている既存楽曲を入力として、1000個生成したものの中から60個程度提示した所、図12のような既存楽曲と似たメロディを得ることが出来た。音高の類似度は0.980、音価の類似度は0.640、休符の類似度は0.790となっている。入力と音高はほとんど同じだが、音価が異なっている。他の既存楽曲に対しても評価実験を行ったが、50個程度修正案を提示することで既存楽曲と似たメロディが現れることから、本システムでは作曲者が作成するような修正案を提示することができる。



図 11. 評価実験2に使用した楽譜



図 12. 評価実験2の出力

4.2 生成実験

自ら作成したメロディを入力として与え、類似度モデルによって提示されるメロディの中に修正案として利用できるものがあるかを確認するために、生成実験を行った。評価実験において、50個程度修正案を提示すれば既存楽曲と似たメロディが現れることが多かったため、生成実験では1000個の生成結果に対して50個の修正案を提示するようにした。今回の実験では、自ら作成した4小節目の2つメロディに対して、8小節目の修正案を提示した。実験1では、図13のような4小節のメロディを作成し、7小節まで繰り返して8小節目を本システムで作成した。修正案を50個提示し、20番目に出力された図14のような5から8小節のメロディを修正案として採用した。音高の類似度は0.479、音価の類似度は1.000、休符の類似度は0

.935 となっている。他の出力結果においては、音価と休符の類似度はほとんど変わらなかったが、音高の類似度は幅広い値が現れた。



図 13. 実験 1 の入力



図 14. 実験 1 の出力

次に、実験 2 では図 15 の楽譜を入力として与え、3 番目に提示された図 16 の出力を修正案として選択した。音高の類似度は 0.805、音価の類似度は 0.925、休符の類似度は 0.897 となっている。実験 1 とは異なり、音価と休符の類似度が 1.0 となるような出力は 1000 個の中に数個しかなかった。3 小節目の音高が高くなっているためか、終止の音高が比較的高くなっている。



図 15. 実験 2 の入力



図 16. 実験 2 の出力

5 考察

今回の提案手法によって、既存楽曲にあるような繰り返しのメロディを作成することが可能であり、自ら作成したメロディについても、4 小節目を参考にした 8 小節目の楽譜の提示が可能だった。実際に聴いてみた所、修正案としてふざわしいようなものだった。

修正案として提示されたメロディは、全体的に音価の類似度がほとんど同じだったが、異なるものも提示された。音高については、分散が大きいことから類似度が異なる出力を多く得ることが出来た。学習した類似度モデルを用いて、音高、音価、休符について類似度が異なる様々なパターンが提示が出来た。

しかし、修正案として採用したいと思えるメロディが出てくるまで 50 個では足りないこともあった。この場合は、1000 個の楽譜の生成を再度行い、その中から 50 個提示するようにするべきである。また、本システムで作成したメロディの類似度モデルにおいて、他の特徴量についても学習を行い、生成結果をさらに絞るべきである。

一方でコードが 1 小節につき 1 つまでしか与えられないことで、多様なコード進行を用いたメロディを修正することが出来ない。これによって、システムが使用できる範囲が狭まってしまっている。さらに、学習データも 1 小節につき 1 つのコードしかないものしか使用できないため、使用できるデータかどうか仕分けする必要も出てくる。これらを解決するためには、自動生成の部分において、コードをラベルではなく音符情報で与えられるようなネットワーク、学習を用いれば解決出来る。

6 おわりに

想定していたようなメロディの修正システムを作成させることが出来た。メロディの類似度モデルにおいて音高、音価、休符の 3 つの特徴量のみ用いたが、既存楽曲においては同じような修正案を得ることができ、自ら作成したメロディについては良いと思える修正案を提示することが出来た。

学習データやメロディの類似度モデルにおいて、楽曲のジャンルを絞っておらず様々なジャンルを使用した、ImprovRNN

での生成や修正案の提示において特に問題は発生しなかった。しかし、ジャンルを絞るとどのような結果が得られるかというのも試してみたい。

さらに、今回の学習モデルでは前から後ろへの生成のみを行っており、次の展開のメロディを考慮していない。そこで、後ろから前も同時に生成することでさらに良い結果が得られる可能性もある。

また、修正案の選択はユーザーがするものとなっており、ここを自動化させることは出来なかった。メロディの類似度モデルにおいて、さらなる特徴量の採用や、修正案の選択の自動化などが今後の課題である。

参考文献

- [1] Michael C. Mozer, "Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing", Connection Science volume 6, 1994, pp. 247-280.
- [2] Douglas Eck, Jurgen Schmidhuber, "Finding temporal structure in music: blues improvisation with LSTM recurrent networks", Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing, 2002, pp. 747-756.
- [3] Gaetan Hadjeres, Francois Pachet, Frank Nielsen, "DeepBach: a Steerable Model for Bach Chorales Generation", Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 1362-1371.
- [4] Nicolas Boulanger-Lewandowski, Yoshua Bengio, Pascal Vincent, "Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription", Proceedings of the 29th International Conference on Machine Learning, 2012
- [5] Olof Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training", Constructive Machine Learning Workshop at NIPS 2016
- [6] Li-Chia Yang, Szu-Yu Chou, Yi-Hsuan Yang "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation", ISMIR (International Society of Music Information Retrieval) Conference, 2017
- [7] Magenta: Make Music and Art Using Machine Learning <http://magenta.tensorflow.org/>
- [8] Andy M. Sarroff, Michael Casey, "MUSICAL AUDIO SYNTHESIS USING AUTOENCODING NEURAL NETS", Proceedings of the International Computer Music Conference, 2014, pp. 14-20
- [9] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu "WaveNet: A Generative Model for Raw Audio", Google AI publications, 2016
- [10] Natasha Jaques, Shixiang Gu, Richard E. Turner, Douglas Eck, "Tuning Recurrent Neural Networks With Reinforcement Learning", The International Conference on Learning Representations 2017
- [11] Wikifonia <http://www.wikifonia.org/>
- [12] HookTheory: Software And Books That Help You Make Amazing Music <https://www.hooktheory.com/theorytab>
- [13] Tuomas Eerola, Petri Toiviainen, "MIR IN MATLAB: THE MIDI TOOLBOX", ISMIR (International Society of Music Information Retrieval) Conference, 2004