

# LSTM を用いた既存楽曲の学習に基づく作曲支援手法

久原 聖志<sup>†</sup> 牛尼 剛聡<sup>‡</sup>

<sup>†</sup>九州大学大学院芸術工学府 〒815-8540 福岡県福岡市南区塩原 4-9-1

<sup>‡</sup>九州大学大学院芸術工学研究院 〒815-8540 福岡県福岡市南区塩原 4-9-1

E-mail: <sup>†</sup> kuhara.s.000@s.kyushu-u.ac.jp, <sup>‡</sup> ushiama@design.kyushu-u.ac.jp

**あらまし** 本研究では、音楽に対する専門的な知識を持たないユーザに対して、楽曲の制作を支援する手法の開発を目的とする。そのために、機械学習を用いて既存の楽曲の特徴を学習し、ユーザが制作したメロディの後に適したメロディの候補を提示する。本研究では、音楽が時系列データであることに着目し、時系列データを対象とした代表的な機械学習手法である RNN (Recurrent Neural Network) を拡張した LSTM (Long Short-Term Memory) を利用する。ユーザが指定した楽曲を学習させたモデルを生成し、ユーザが提示した楽曲メロディの一部分入力として、その部分の続きに適した音の候補の推定し、作曲活動を支援する。

**キーワード** 深層学習 作曲支援 楽曲分析

## 1. はじめに

近年、インターネットでは大容量のデータを高速に送受信できるようになり、インターネットを用いたコンテンツ配信が急速に普及している。また、それに伴って YouTube に代表される動画投稿サイトなどに、一般個人でもコンテンツを制作し、アップロードできる環境が整備された。しかし、そのような環境においては、専門的な知識や能力を持たない利用者が作品を投稿できるため、質が低い作品が存在する可能性がある。そうした中で、音楽的に専門的な知識や能力を持たない利用者が、高品質な作品を制作できるように支援する環境が期待されている。

一方、深層学習に代表される機械学習技術の飛躍的な進歩に伴い、人が制作した様々なコンテンツの特徴をコンピュータが学習し、人間の創作活動を支援する研究に注目が集まっている。これらの研究では、人間が制作したコンテンツを学習したモデルを構築し、そのモデルを利用して、高度なコンテンツの制作を支援する。例えば、Gatys ら[1]は、既存の絵画の画風を学習して、任意の写真の画風を変換する手法を提案している。また Google の AutoDraw[2]では、ユーザのラフスケッチを高品質なイラストに変換するサービスを提供している。

本研究では、楽曲を対象として、機械学習技術を利用して、専門的な音楽知識や能力を有さないユーザの、高品質で高評価を得られる作品の制作を支援する手法を開発する。具体的には、高い評価を得ている既存の楽曲の特徴を学習したモデルを構築し、そのモデルに基づいてユーザが生成したメロディの後に続くのにふさわしい音の候補を提示することにより、ユーザを支援する。

既存の楽曲の特徴を学習するモデルを構築するためには、深層学習を利用する。深層学習には様々な手

法が提案されているが、本研究では、その中でも時系列のデータを学習することができるアルゴリズムとして知られている LSTM (Long Short-Term Memory) を用いる。既存楽曲を構成するそれぞれの音に関して、次の音を正解データとして LSTM で学習しモデルを構築する。次に、ユーザによって作曲されたメロディの一部分を、モデルに入力データとして与え、ユーザの入力したメロディの次の音を予測することによって候補を作成してユーザに提示する。

提案手法に基づいて、PC 上で動作するプロトタイプを制作し、被験者実験により提案手法の有効性を評価する。

本論文の構成は以下のとおりである。2 では、本研究に関連する研究を説明する。3 では、本研究で利用する機械学習アルゴリズムである LSTM について説明する。4 では、本研究における作曲支援のアプローチを説明する。5 では提案手法について述べる。6 では提案手法の評価について述べる。7 はまとめである。

## 2. 関連研究

これまでにも、コンピュータを利用して、高品質な楽曲を簡単に制作するための研究が幾つか存在する。コンピュータの計算による作曲は 1950 年代の ILLIAC I による弦楽四重奏曲 (ILLIAC Suite) を始めとして、コンピュータが発明された初期から行われてきた。深山らによると、代表的な自動作曲の手法として、音楽理論を明示的に実装するという Orpheus でも採用された手法であるルールベース、既存作曲家のスタイルを模倣し、その断片を組み合わせるなどして利用する事例ベース、そして確率モデルなどを用いた機械学習手法の発展に伴い、楽譜のデータベースから学習した音高のマルコフ連鎖を用いて旋律や伴奏の生成などを行う確率モデルベースの 3 種類が存在する[3]。

深山ら[4]が提案する Orpheus では、音楽理論を明示的にシステムに取り込み、メロディを生成する事が可能である。一方、安藤ら[5]が提案している作曲支援システムでは、対話型進化論的計算を応用して作曲過程を実装し、クラシック作品を作曲できる。これらのシステムの特徴は、人間が音楽的な知識を明示的に与えることにより、コンピュータが楽曲を自動的に生成することである。この手法では、生成される楽曲はシステム開発者の音楽の知識に依存するが、音楽のジャンルや雰囲気などをユーザが指定することは困難である。また、高品質な楽曲の知識をすべて知識として抽出することは現実的ではない。

一方、近年は既存のコンテンツのデータを利用して、コンピュータがその特徴を自動的に学習し、学習した特徴に基づいて、コンテンツを生成する研究が行われている。ここで対象とするコンテンツの種類としては、イラスト、文章、楽曲等様々である。これらの研究で利用されているのは、近年飛躍的に進歩した深層学習である。楽曲を深層学習で学習した研究としては、楽曲が時系列データであることに着目し、時系列のデータを学習するのに適した RNN (Recurrent Neural Network), また RNN を改良した LSTM が用いられている。

姫野による研究[6]では、LSTM による自動作曲を目指し、MIDI データを時刻で区切り得られた音の集合を時系列データとしてまとめ、モデルの学習を行っている。生成された作品は人間によるアンケートにより適切な創作物として評価された。

また、音楽情報をテキスト化し、LSTM を用いて学習した Choi らの研究[7]が存在する。この研究では、文字に基づいた方法と単語に基づいた方法でコード進行とドラムパターンの学習と生成が行われており、コード進行の学習は両方の方法において、ドラムパターンは単語に基づいた方法において有用とされる結果が示されている。

本研究では、楽曲の自動的な生成を目的とするのではなく、ユーザが生成したメロディの一部に対して、そのメロディの続きにふさわしいメロディの候補を提示することによってユーザの作曲活動を支援することが、従来研究とは異なる。

### 3. LSTM

本研究では、深層学習のアルゴリズムとして LSTM[8]を用いる。時系列データを学習する深層学習アルゴリズムとして RNN があり、ニューラルネットワークの中間層の出力値を次のデータの入力とするモデルであるが、学習するデータの時刻が増えていくことで中間層の深さが増大していく順伝播型ニューラル

ネットワークと同じようになり、勾配消失や勾配爆発の問題が発生する。それらを解決するために改良を加えたものが LSTM であり、入力ゲート、出力ゲート、忘却ゲートを取り入れることによって情報の取捨選択を行うことにより時系列データの学習精度が向上している。LSTM は、人間の短期記憶と長期記憶を模倣するような機構を有している点に特徴がある。

## 4. 作曲支援について

### 4.1. 作曲支援の意義

1 では、現在の Web 上のユーザが投稿したコンテンツには質が低い作品が存在する場合があると述べた。しかし、質が低くても、作品を自分で創作することはユーザにとって楽しみとなる。ユーザが自分で発想したアイデアを作品に反映することができ、その質が高ければ、ユーザの満足度は高いと考えられる。自動作曲は、創作を自ら行いたいと思うユーザの要求には応えることができず、また、自動作曲においては意図しない結果が生成されることがある。

本研究では、ユーザにメロディを入力させることによってユーザが発想したアイデアを取り入れ、自ら創作したいという要求を満たす機構を開発する必要があると考え、自動作曲ではなく作曲支援を行う手法を開発する。

### 4.2. 創作過程における本研究の位置づけ

松原らのサーベイ[9]によると、創作行為は Shneiderman による

- Collect (既存作品から学びとること)
- Relate (既存作品の分析や解釈を行うこと)
- Create (解釈から独自の表現を見出すこと)
- Donate (表現を外在化させること)

という Genex Phases[10]で提案された 4 段階 (以下 C-R-C-D とする) によって行われており、必ずしもこの順番でなく、逆に戻ることも認められている。

自動作曲においては 4 段階全てがコンピュータによって行うことができるのが望ましい。一方、作曲支援とは、何かしらの意図をユーザが持っており、その意図を作品として実現する際に必要な作業をコンピュータが行うという形で補助するシステムである。

Shneiderman の提案に沿って本研究を述べるとするならば、Collect と Relate をコンピュータによって行い、Create をユーザの意図を取り入れてコンピュータが行う、いわゆる共同で行うというシステムである。なお Donate は一般的には作曲の後で行う過程であるため、本研究では Collect, Relate, Create をシステムとして実装する。

## 5. 提案手法

### 5.1. 全体の概要

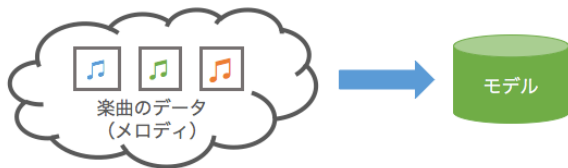


図 2：既存楽曲の学習の流れ



図 3：ユーザの入力と生成の流れ

本研究では、まず図 2 のように既存楽曲を教師データとしてモデルを構築する。次に図 3 のようにユーザの入力をモデルを通して分析し、モデルに沿って続きとして推測されたメロディを出力する。ここで、モデルとは、LSTM の学習で更新および保存された重みを意味する。

### 5.2. 学習の手法

本研究では、既存楽曲に関するデータとして、音楽情報をテキスト化したものを用いる。具体的には音高を表す数字をカッコ記号 ([ ]) で囲んだもので一つの音を表し、音の系列として楽曲を表現する。また、本研究では、メロディの音高のみを対象とし、音の長さやリズムは考慮しない。

既存楽曲のデータを作成する方法として、Songle[11] 等の楽曲情報を提供するサービスから取得する方法や、あるいは直接手入力で行うという方法が考えられる。本研究では、手作業で既存楽曲のデータを作成し利用する。

図 4 の楽譜の一番左に位置する “ド” の音（英語名 C, 国際式[12]記述 C4）の音符が、テキスト化した数字（以下ピッチ数とする。）の 60 に相当し、数字が 1 増えるごとに半音上がり、1 下がるごとに半音下がる。例えば、図 4 の楽譜で表されたフランス童謡である “きらきら星” をピッチ数で表現すると、  
[60][60][67][67][69][69][67][65][65][64][64][62][62][60] のようになる。図 4 の楽曲を既存楽曲のデータとすると、このピッチ数が並んだテキストでファイルが構成される。このテキストファイルを学習用のプログラムに読み込ませ、モデルを構築していく。

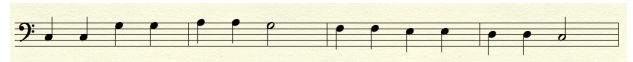


図 4 きらきら星の楽譜

本研究では学習用のプログラムとして、Python 上で動作する深層学習用のライブラリの一つである Tensorflow を利用し、LSTM で学習を行う。本研究では 1 曲毎に学習を行い、また後述の生成過程においても 1 曲の学習をもとに出力するメロディを決定する。

次は、具体的な学習のアルゴリズムについて述べる。入力層と出力層のベクトルは、メロディを構成するピッチ数の集合の長さがベクトルの次元となる。例えば図 4 のきらきら星であれば、音の集合  $M$  は

$$M = \{60, 62, 64, 65, 67, 69\}$$

となり、ベクトル  $v$  は

$$v = \begin{pmatrix} p_{60} \\ p_{62} \\ p_{64} \\ p_{65} \\ p_{67} \\ p_{69} \end{pmatrix}$$

となる。 $p_x$  は入力時にピッチ数が  $x$  であれば 1 となり、 $v$  のそれ以外の要素は 0 となる。例えば、最初の入力が [64] であった場合、

$$v = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

と表現され、このような表現方法を One-hot ベクトルと呼ぶ。出力層のベクトルは、入力層のベクトルと同じ次元数となり、学習の過程においては出力層も One-hot ベクトルとなる。

本研究の学習過程では中間層の次元数は 128 とし、中間層のベクトルは 128 個の LSTM のメモリユニットで構成される。時刻  $t$  における中間層のベクトル  $h_t$  は、 $x_t$  を入力ベクトルとし  $W^{in}$  を入力層から中間層への重み、 $W$  を 1 つ前の時刻  $t-1$  の中間層から時刻  $t$  の中間層への重みとすると、活性化関数  $f$  を適用して、

$$h_t = f(W^{in}x_t + Wh_{t-1})$$

と表される。出力層の  $y_t$  は中間層の出力に Softmax 関数を適用したものであり、中間層から出力層への重みを  $W^{out}$  とすると、

$$y_t = \text{softmax}(W^{out}h_t)$$

となる。つまり、 $x_t$  と  $y_t$  が与えられたとき、

$$y_t = \text{softmax}(W^{out}f(W^{in}x_t + Wh_{t-1}))$$

を満たすように  $W^{out}$ ,  $W^{in}$ ,  $W$  を更新する。例えば、時刻  $t$  におけるピッチ数の入力が 60, 出力が 67 とすると、

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \text{softmax} \left( W^{out} f \left( W^{in} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + W h_{t-1} \right) \right)$$

を満たすような  $W^{out}$ ,  $W^{in}$ ,  $W$  となるように更新していく。

本研究では,  $W^{out}$ ,  $W^{in}$ ,  $W$  と呼ぶ重みや, 入力層, 中間層, および出力層の次元数などをまとめたものをモデルと呼び, 学習過程ではそのモデルが構築される。

### 5.3. 生成の手法

本研究では, ユーザが入力したメロディの続きを, 深層学習を用いて推測する。

Python のプログラムは, 後述のプロトタイプからユーザが入力したメロディを表す文字列, および出力するメロディの長さなどを受け取り, Python 上で動作している LSTM のプログラム(以降, LSTM ユニットとする)に入力し, 指定された長さの出力を得てプロトタイプに送信するシステムとなっている。

LSTM ユニットでは, 5.2 節で述べたモデルに入力の One-hot ベクトルを与えることによって, 出力層ベクトルの各成分は, 該当する音への遷移確率となる。そしてその出力層のベクトルに log 関数を適用してそのベクトルを多様性パラメータで割り, さらにその次に Softmax 関数をベクトルに適用した後に, 多項分布でサンプリングを行い, その結果最大となった確率が, 入力した要素の次にくる要素として出力される。

### 5.4. インターフェース

5.2 節および 5.3 節で述べた手法を簡単に用いることができるようにするため, インターフェースを提案する。また提案手法とインターフェースの有用性を実証するため, PC 上で動作するインターフェースのプロトタイプを制作した。プロトタイプの GUI を実装するプラットフォームとして Processing を利用した。一方で, LSTM ユニットとはプログラミング言語や動作環境が異なるため, プロトタイプと LSTM ユニット間の通信にはソケット通信を用いる。

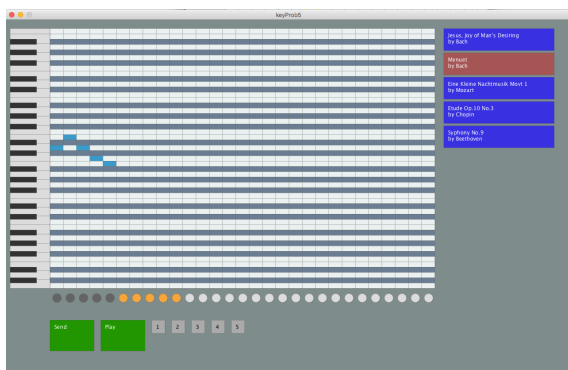


図 5 プロトタイプの全体画面

プロトタイプの動作例を図 5 に示す。プロトタイプにはピアノロールがあり, 縦軸が音高で横軸が時刻である。ピアノの鍵盤部分のクリックまたは MIDI キーボードを使用することによりユーザはメロディを入力できる。入力されたメロディはピアノロール上に青く表示される。図 5 の右側にあるのは学習した楽曲のリストであり, リストにある楽曲は予め登録されている。ユーザはメロディの提示の元となる学習された曲を選択する。また, ピアノロールの真下に位置する円は提示されるメロディの長さを指定するものである。黒い円はすでにメロディが存在している時刻である。白い円を押すことによって最後の黒い円の次の円から押された白円の時刻までの円がオレンジに変化し, オレンジの円の数が提示されるメロディの長さに相当する。図 5 に沿って述べると, すでに 5 時刻分メロディが入力されているため, 左から 5 つの円は黒くなっている。一方, オレンジの円は 5 つになっているため, 提示されるメロディの長さは 5 時刻分である。そして円の下に緑色の "Send" ボタンを押すことにより, ボタンは "Analyzing Now" に赤く変化し, LSTM ユニットによる解析が開始される。

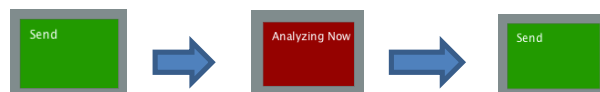


図 6 解析開始前, 解析中, 解析終了後のボタンの変化

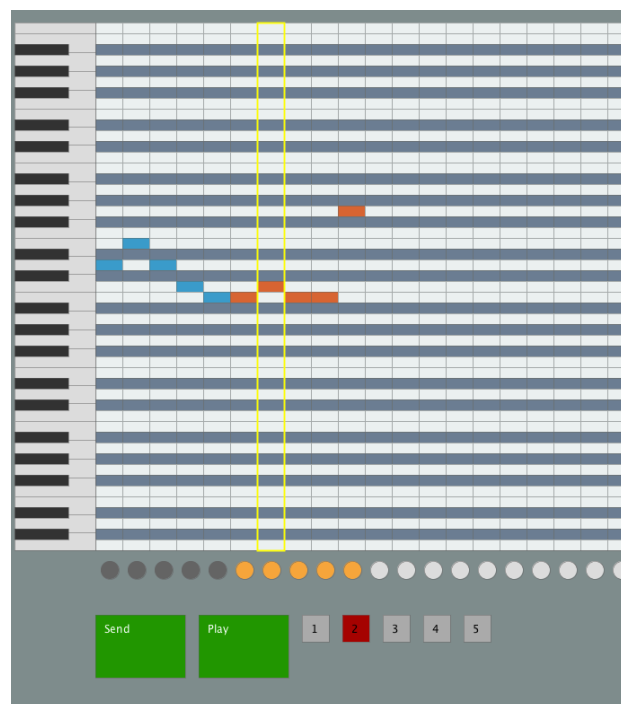


図 7 解析終了後の画面

LSTM ユニットによる解析が終了すると再び緑色の”Send”ボタンに戻り、図7のようにピアノロール上には提示されたメロディが赤く表示される。”Play”ボタンをクリックすると、ピアノロール上にあるすべての音が順番に再生される。再生中は時刻に沿って黄色い四角が移動し、その時刻で再生されている音がわかる。また、”Play”ボタンの右側にある数字が書かれている5つのボタンは、メロディの候補である。1回の”Send”ボタンのクリックで5回解析を行い、その分の結果を1つ1つの候補に格納する。ユーザは提示されたメロディが気に入らない場合、別の候補を選ぶことでそのメロディを聞くことができ、すべての候補が気に入らなければ再び”Send”ボタンをクリックすることにより解析をやり直すことができる。なお、提示されたメロディは、5.3節で述べたように LSTM ユニットが受け取った入力で得られた出力を更にサンプリングした結果であり、必ずしも同じ入力で同じ結果が得られるとは限らない。ユーザは、LSTM ユニットにメロディを再生成させることができる一方、提示されたメロディの一部を図8のように修正することができる。修正された箇所は黄色で表示される。修正が加えられたメロディに関しても、”Play”ボタンで再生することができる。ユーザが図8の状態ですべてのメロディの入力を開始すると、ピアノロールにあるメロディは保存されるが、候補にあったメロディはリセットされる。再び”Send”ボタンが押すことでピアノロールにあるメロディ全体を入力として LSTM ユニットで新たな候補を生成することができる。



図8 ユーザによる修正が行われたときの画面

## 6. 評価実験

本研究の有用性を示すため、被験者による評価実験を行った。本章では2つの実験を述べるが、両実験とも LSTM の他に時系列のデータを学習する手法として RNN と GRU (Gated Recurrent Units) を新たに準備し、LSTM との比較実験を行った。手法ごとに3つのサンプル、計9つのサンプルの候補を生成している。これらのサンプルについて、印象および入力したメロディの続きとして生成されたメロディがふさわしいかどうかを質問した。この実験で学習した楽曲は、J.S.Bach 作曲の“Jesus bleibet meine Freude(カンタータ BWV147 より、日本名「主よ、人の望みの喜びを」)”の一部分である。最後に、この提案手法およびインターフェースに対して可能性や期待を感じるかどうかを5段階で評価してもらった。

### 6.1. 実験1(プロトタイプに触れてもらう実験)

はじめに、被験者3人に実際にプロトタイプに触れ、メロディを入力してもらい、生成されたメロディについて、以下の質問をして評価してもらった。

- 生成された結果の中に、修正しなくても良いと思えるメロディはあったか
- 修正すれば良くなると思えるメロディはあったか
- 上記で答えたサンプルのうち、続きとしてふさわしいと思えるメロディはあったか

該当する候補があればその番号を記入し、なければメロディの入力や生成を繰り返させ、何度繰り返しても該当する候補が出てこない場合は無回答とするように指示した。

回答結果として、図9に評価された手法とその評価数のヒストグラムを示す。

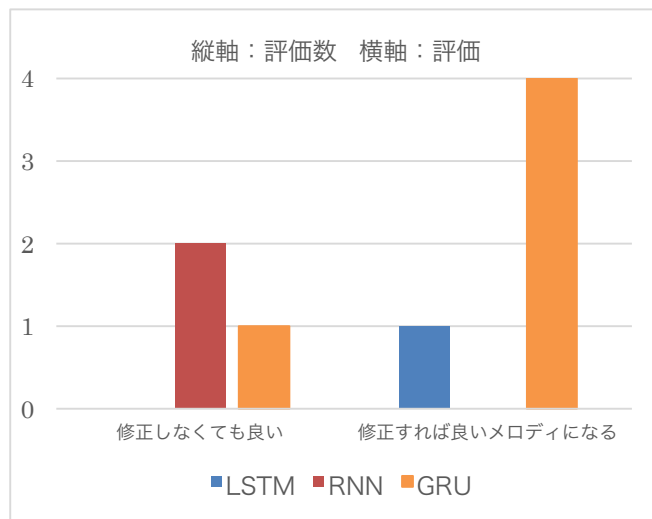


図9 実験1の回答結果

修正しなくても良いと評価された候補のうち、GRU

一つが一番良いという評価となり、かつその候補が入力したメロディの続きとしてふさわしいと評価された。

一方修正すれば良いメロディになると評価された候補のうち、LSTMとGRUで生成された候補がそれぞれ一番良いという評価となり、LSTMで生成された候補を選んだ被験者は、同候補が入力したメロディの続きとしてふさわしいという評価をした。そして、インターフェースへの期待の評価値の平均は5段階中3.66であった。

以上の結果から、印象とふさわしさを両方満たす手法はLSTMとGRUであり、LSTMとGRUがモデルの構築手法としては有効であると考えられる。一方で、候補の選び方がわからないといった作曲未経験者の意見があり、作曲が全くできない状態から作曲ができるまでに至らせる思考プロセスの研究も行う必要がある。

## 6.2. 実験2(生成物を評価してもらう実験)

次に、筆者がプロトタイプを用いてメロディを入力し、生成された候補について、被験者14人にメロディの全体的な印象を5段階、入力されたメロディの続きとしてふさわしいかどうか(以下、ふさわしさとする)を4段階で評価してもらった。メロディを聞いてもらい評価する理由として、この研究の目的が高品質で高評価が得られる楽曲を創作する支援であるため、他人からの評価が高ければ提案手法は目的を満たすのに有効であると考えられることができるということが挙げられる。またプロトタイプを使用している場面の動画を被験者に見せ、実験1と同じくインターフェースに対する期待を5段階で質問した。また、9つのサンプルのうち、一番良いものを選んでもらった。以下に、各手法と各評価項目の評価値の平均を割合で示しまとめたものを示す。印象は5段階で、ふさわしさは4段階の評価になっているので、それぞれの評価値の平均を段階数で割っている。

表1: 各手法と各評価項目

	LSTM	RNN	GRU
印象の評価値の平均÷5	0.700	0.557	0.576
ふさわしさの評価値の平均÷4	0.679	0.565	0.583

印象もふさわしさも、有意水準5%でT検定を行った結果、LSTMと他の2手法との間に有意差が見られた。

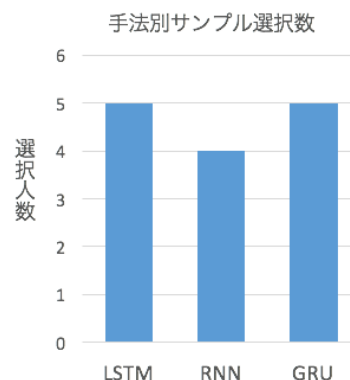


図10 手法別サンプル選択数

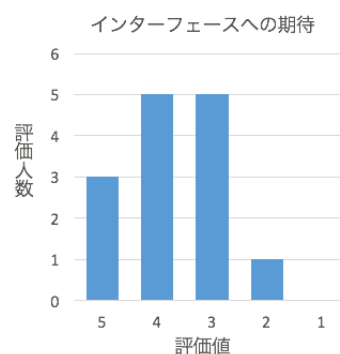


図11 インターフェースへの期待

9つのサンプルから一番良いものを選んでもらった結果を、手法別に図10にまとめた。LSTMとGRUがRNNを上回っている。

また、インターフェースへの期待について、被験者の大半がどちらでもないを意味する3、およびそれ以上の評価を行っており、おおむね良い評価が得られた。

以上の実験結果から、他の2手法と比較してLSTMがモデルの構築手法、学習手法として有効であると考えられる。また、LSTMを用いたときの印象の評価値の割合は7割と高く、また5段階中3以上という評価が半数以上を占めたことから、LSTMは他人からの評価が高いメロディを予測し生成できると考えられ、本研究の目的を満たす手法として、LSTMは有効であると考えられる。

## 7. おわりに

本論文では、時系列を対象とした代表的な機械学習手法であるLSTMを利用して、音楽に関する専門的な知識や能力を有さないユーザが、高品質な楽曲の制作を支援する手法を提案した。そして、LSTMが手法として有効であることを被験者実験によって示した。今回はメロディの流れのみを学習しているが、考慮しなかったリズムについても学習と生成の手法を検討していくことが今後の課題として挙げられる。また、コー

ド進行も学習してシステムに取り入れることを検討中である。

## 謝 辞

本研究は（公財）電気通信普及財団の助成およびJSPS 科研費 JP16K12534 の助成をうけたものです。

## 参 考 文 献

- [1] L. A. Gatys, A. S. Ecker, and M. Bethge, A Neural Algorithm of Artistic Style, arXiv, 2015
- [2] <https://www.autodraw.com/>
- [3] 深山覚, 後藤真孝, ”特集 音楽を軸に広がる情報科学 4. 音楽とコンテンツ生成”, 情報処理 Vol.57 No.6 June 2016, pp.516-518
- [4] 深山覚, 中妻啓, 米林裕一郎, 酒向慎司, 西本卓也, 小野順貴, 嵯峨山茂樹, ”Orpheus: 歌詞の韻律に基づいた自動作曲システム”, 情報処理学会研究報告 2008-MUS-76
- [5] 安藤大地, Dahlstedt Palle, Nordahl Mats, 伊庭斉志, ”対話型 GP を用いたクラシック音楽のための作曲支援システム”, 芸術科学会論文誌, Vol.4(2005) No.2 pp.77-87
- [6] 姫野雅大, ”LSTM による自動作曲システムの構築”, 学位論文, 2016.3
- [7] Keunwoo Choi, George Fazekas, Mark Sandler, ”Text-based LSTM networks for Automatic Music Composition”, arXiv:1604.05358v1 [cs.AI] 18 April 2016
- [8] S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural computation, 9(8), pp. 1735-1780, 1997
- [9] 松原正樹, 深山覚, 奥村健太, 寺村佳子, 大村英史, 橋田光代, 北原鉄朗, ”創作過程の分類に基づく自動音楽生成研究のサーベイ”, コンピュータソフトウェア 30(1), 1\_101-1\_118, 2013
- [10] Shneiderman, B, ”Creating Creativity”, User-Interfaces for Supporting Innovation, ACM Transactions on Computer-Human Interaction(ToCHI), Vol.7, No.1 (2000), pp.114-138.
- [11] <http://songle.jp/>
- [12] [http://www.flutopedia.com/octave\\_notation.htm](http://www.flutopedia.com/octave_notation.htm)