

## リカレントニューラルネットワークを用いた コード進行の自動生成

西田 正洋 菊地 進一 中西 正和

慶應義塾大学大学院 理工学研究科 開放環境科学専攻  
{nishida,shinichi,czl}@nak.ics.keio.ac.jp

本稿ではリカレントニューラルネットワークを用いた単旋律に対応するコード進行の自動生成の手法を提案する。まず、単旋律に対応するコード進行の記憶の実験を行い、3つのリカレントニューラルネットワーク、Elmanのネットワーク、複素リカレントニューラルネットワーク (Multilayer Network using Complex Neuron with Local Feedback, MNCF)、短期記憶を用いたシンプルリカレントニューラル (Simple Recurrent Neural Network with Short-term Memory) の比較を行った。比較の結果、学習時間、収束率の最も良いSRNSMをコード進行の自動生成システムに用いた。本手法により作成されたコード進行は、アンケートによる主観評価において比較的的自然であるという結果を得られた。また、作成されたコード進行の雰囲気が学習データで用いたアーティストの楽曲の雰囲気に似ているなどの感想を得た。

キーワード: 作曲, コード進行, ニューラルネットワーク

## Generating a Chord Progression using Neural Networks

Masahiro NISHIDA Shinichi KIKUCHI Masakazu NAKANISHI

Department of Science for Open and Environmental Systems,  
Keio University  
3-14-1, Hiyoshi, Kouhoku-ku, Yokohama 223-8522, Japan

This paper reports an attempt of generating a chord progression corresponding to a single melody using recurrent neural networks. First, the three types of recurrent neural networks, Elman's recurrent neural network, Multilayer Network using Complex Neuron with Local Feedback(MNCF), and Simple Recurrent Network with Short-term Memory(SRNSM) are compared with respect to memorizing the chord progression. The comparison result shows that SRNSM attains the best learning speed and convergence rate. The generated chord progression was rated "natural" by the questionnaires. Also, comments were given that the mood of the generated chord progression is similar to that of the musician's songs of learning data.

**KeyWord:** composition, chord progression, neural networks

## 1 はじめに

従来コンピュータは事務処理や計算などの実務的なことがらに多く用いられてきたが、近年のコンピュータ技術の発展により、趣味や娯楽などに対する需要が高まっている。音楽の分野においてはコンピュータを用いた演奏や作曲などの応用がなされている。

また、コンピュータ技術においては近年 AI の分野が注目されている。ニューラルネットもそのひとつであり、人間の認識、知覚動作などのメカニズムの理解、シミュレーションのための道具としての幅広い興味を得ている。そのなかにおいて、ニューラルネットは音楽の解析のための道具としても注目されている。

そこで本研究では、音楽に対するコンピュータの応用の一分野である作曲を、ニューラルネットを用いて行う方法に注目した。

楽曲中において、ある一つのコードは先行するいくつものメロディー、コードから決定されると考えられる。また現在、時系列の問題を解くためのニューラルネットのモデルとして、様々なモデルが提案されている。これらの点に注目して本研究では Elman のネットワーク [1]、複素リカレントニューラルネットワーク (Multilayer Network using Complex Neuron with Local Feedback, MNCF) [2][3] と短期記憶を用いたシンプルリカレントネットワーク (Simple Recurrent Network with Short-term Memory, SRNSM)[4][5][6] の 3 つを用いて、与えられたメロディに対応するコード進行の記憶を行い、さらに各ニューラルネットの比較を行う。また、比較により優れた評価を得たニューラルネットを用いてコード進行の自動生成を行う。

以下、2 章で本研究で用いるリカレントニューラルネットワークについて、3 章でコード進行の記憶の実験、4 章でコード進行の自動生成の実験、5 章で考察と今後の課題について述べる。

## 2 リカレントニューラルネットワーク

### 2.1 複素リカレントニューラルネットワーク

本ノ内らはネットワーク内で用いる数値を複素数に拡張することによって、リカレントニューラルネットワークの処理能力を高めた複素リカレントニューラルネットワーク (Multilayer Network using Complex neurons with local Feedback, MNCF) を提案している [2][3]。図 1 に MNCF の構造を示す。

第  $n$  層の  $j$  番目のニューロンの出力は次式によって計算される。

$$y_j^{(n)}(t) = f(u_j^{(n)}(t)) \quad (1)$$

$$u_j^{(n)}(t) = \sum_i \omega_{ji}^{(n)} y_i^{(n-1)}(t) + v_j^{(n)} y_j^{(n)}(t-1) + \theta_j^{(n)} \quad (2)$$

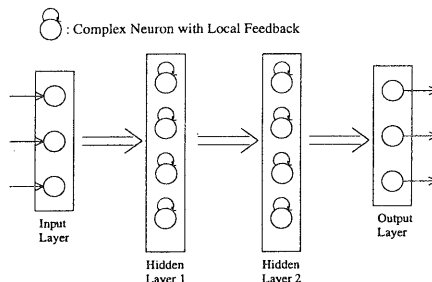


図 1: MNCF

ここで  $u_j^{(n)}(t)$  はニューロンへの入力、 $\omega_{ji}^{(n)}$  は  $(n-1)$  層の  $i$  番目のニューロンから  $n$  層の  $j$  番目のニューロンへの結合荷重、 $v_j^{(n)}$  は自己フィードバックの荷重、 $\theta_j^{(n)}$  は閾値である。また、 $f(x)$  は次式で示される複素シグモイド関数である。

$$f(x) \equiv \tanh(\operatorname{Re}x) + j \tanh(\operatorname{Im}x) \quad (3)$$

MNCF において時系列を扱うための学習アルゴリズムとしては、複素バックプロパゲーション学習 (Complex Back-Propagation for Temporal sequence, CBPT) が用いられる [2][3]。

### 2.2 短期記憶を用いたリカレントニューラルネットワーク

筆者らは、入力にバッファを用いて前後関係の情報を短期記憶として利用するネットワーク (バッファモデル)[7] と、先に述べた中間層の値をコピーした文脈層を持つ Elman のネットワーク [1] を組み合わせた構造を持つ、短期記憶を用いたシンプルリカレントネットワーク (Simple Recurrent Network with Short-term Memory, SRNSM) を提案した [4][5][6]。図 2 に SRNSM の構造を示す。

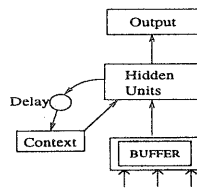


図 2: SRNSM

学習アルゴリズムとしては、Ooyen らによる次の評価関数を用いた誤差逆伝搬学習の高速化手法であるクロスエントロピー法 (Cross Entropy method, CE) を用いる [8]。

$$J = - \sum_i \{ t_i \ln o_i + (1 - t_i) \ln(1 - o_i) \} \quad (4)$$

ここで、 $t_i$  はユニット  $i$  の教師信号、 $o_i$  はその出力値である。これにより、シグモイド関数を微分したときに生じる学習を遅らせたり、収束率を低下させたりする項を打ち消すことができる。

### 3 メロディーに対応するコード進行の記憶

#### 3.1 実験方法

楽曲の中のある一つのコードは、その前後のメロディーと先行するいくつかのコードから決定されると考えられる。すなわちコード進行の規則はオートマトンの推移規則として表現することができる。この点に注目して、本研究では時系列処理能力を持つニューラルネットを用いて与えられたメロディに対応するコード進行の記憶を行い、さらに各ニューラルネットワークの性能の比較を行った。なお学習アルゴリズムとして、Elman のネットワークは BP, MNCF は CBPT( $T = 0$ ), SRNSM は CE を用いた。

まず3つのニューラルネット、Elman のネットワーク, MNCF, SRNSM に既知のメロディーとそのコード進行を学習させることによりその関係をネットワークに記憶させ、その実験の結果により3つのニューラルネットの性能を比較した。

##### 3.1.1 システムの構成

ニューラルネットの入出力部を図3に示す。メロディーやコードは各曲の調に依存するところが大きいので、本実験では相対的な音階を示す音度を用いた表記法を使用した。

入力部では、4分音符1つ分の長さを一単位とする離散時刻  $t$  におけるメロディーの情報を22ビットに符号化した。具体的にはダイアトニックスケール上の7つの度数として7ビット("I" から "VII")を割り当てる。同様に1オクターブ上であることを表すために1ビット(" ")を、前の音と続いていることを表すために1ビット(" ")を、休符に1ビット("R")を割り当てる。これら10ビットを8分音符1つ分の長さに割り当てることにより単位時間あたり20ビット、また現在のメロディーが小節の頭であることを表すために1ビット("bar"), また学習パターンの最後であることを表すために1ビット("E")を用いる。

出力部では、4分音符1つ分の長さの離散時刻  $t$  におけるコードの情報を8ビットに符号化する。具体的にはダイアトニックコードの7つの度数として7ビット("I" から "VII")を割り当てる。同様に7thであることを表すために1ビット("7th")を割り当てる。

次に、ユニット数を表1に示す。これらの値は各ネットワークの性能が良くなるように決定した。なお、SRNSM のバッファ数は4とした。

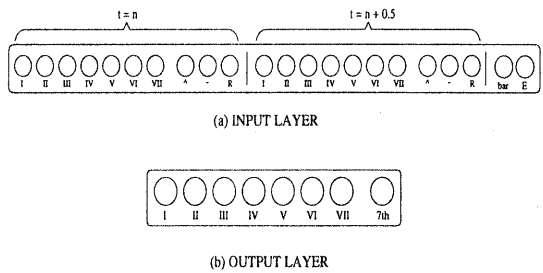


図3: ニューラルネットの入出力層

表1: ユニット数

RNN	ユニット数		
	入力層	隠れ層	出力層
Elman	22+50(context)	50	8
MNCF	22	50 - 50	8
SRNSM	22×4(buffer) +50(context)	50	8

##### 3.1.2 学習データ

入出力としてダイアトニックスケールのみを用いているので、ダイアトニックスケールのみを用いる曲を学習データとした。

子供向けのピアノの稽古用の楽譜[9]より5曲、「やぎさん ゆうびん」、「春の小川」、「めだかの学校」、「母」、「手のひらを太陽に」のメロディーとコードをニューラルネットの学習データとして用いた。曲間には初期化ビット "E" を挿入して区切りをいれた。

##### 3.2 実験結果とその評価

学習定数が0.01, 0.04, 0.08の各場合について、100回の試行による平均学習回数, 平均学習時間, 収束率を表2に示す。

学習回数は平均出力自乗誤差が0.01になるまでのエポック数であり、学習時間はそのときにかかったCPU Time [ s ] である。収束しなかったものは平均の計算から除外した。また学習打ち切り回数を5000エポックとした。収束率は、学習打ち切り回数までに平均出力自乗誤差が0.01に達した割合である。

表2を見ると、Elman のネットワークは学習率を変えても学習が終了せず、他の二つのネットワークに比べてコード進行の記憶を十分に行えないとことがわかる。また学習係数については、収束率、学習時間のどちらで比較しても、MNCFでは0.01, SRNSMでは0.04が最も良い結果を示している。

次に、各ニューラルネットの学習曲線を図4に示す。ここで横軸を学習時間 [ s ] とし、縦軸は平

表 2: 学習回数, 学習時間, 収束率

学習係数	Elman			MNCF			SRNSM		
	学習回数	学習時間	収束率	学習回数	学習時間	収束率	学習回数	学習時間	収束率
0.01	—	—	0%	151	323s	98%	237	288s	100%
0.04	—	—	0%	—	—	0%	107	132s	100%
0.08	—	—	0%	—	—	0%	527	691s	98%

(‘—’は学習が終了しなかったことを示す)

均乗誤差(MSE)とした。学習係数は、先の実験において最も評価の高かった値として、Elmanのネットワークでは0.08, MNCFでは0.01, SRNSMでは0.04とした。

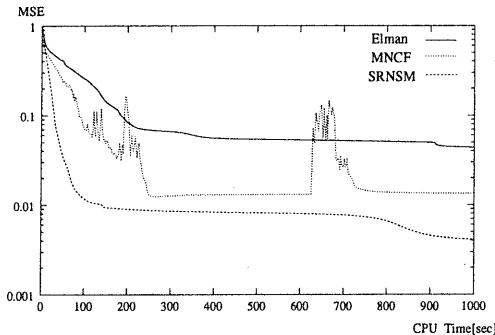


図 4: 学習曲線

学習曲線を見てみると、Elmanは収束が遅く、またMNCFはグラフが振動している部分があり学習が不安定である。SRNSMは他の二つのネットワークに比べて収束が速く安定しているといえる。

以上の実験結果より、コード進行の記憶にはSRNSMが最も適しているという結果を得た。そこで、次節のコード進行の自動生成の実験にはSRNSMを使用することにした。

## 4 メロディーに対応するコード進行の自動生成

### 4.1 実験方法

先の実験においてSRNSMが総合的に最も良い結果が得られた。そこでSRNSMを用いてメロディーに対応するコード進行の自動生成を行った。

まず学習データをSRNSMに学習させた後、学習データ以外のメロディーを入力データとして与えることにより、メロディーに対応するコード進行の自動生成を行った。生成されたコード進行を、カデンツァに解析し、コードとメロディの関係を音楽理論

と比較したまた、実際に生成されたコードに合わせてメロディーを実際に聞くなどにより評価を行った。

#### 4.1.1 システムの構成

コード進行の自動生成を行うシステムとして、異なる入出力をもつ2つのシステム(A), (B)を用いた。(A)はダイアトニックスケールのみに対応するシステムであり、出力はダイアトニックコードとした。(B)はダイアトニックスケール以外のコード進行、メロディーに対応するものであり、出力はコードとした。

(A), (B)のシステムの入出力について以下に示す。

(A) 3節と同様のものを用いる。

(B) 入出力層を図5に示す。

入力部において、4分音符1つ分の長さの単位とする離散時刻 $t$ におけるメロディーの情報を32ビットに符号化する。具体的には音階上の12の度数として12ビット(“I”, “I#”... “VI#”, “VII”)を割り当てる。同様に1オクターブ上であることを表すために1ビット(“”)を、前の音と続いていることを表すために1ビット(“-”)を、休符に1ビット(“R”)を割り当てる。これら15ビットを8分音符1つ分の長さの単位に割り当てることにより単位時間あたり30ビット、また現在のメロディーが小節の頭であることを表すために1ビット(“bar”), また学習パターンの最後であることを表すために1ビット(“E”)を用いる。

出力部において、4分音符1つ分の長さの離散時刻 $t$ におけるコードの情報を52ビットに符号化する。具体的には音階上の12の度数(“I”から“VII”)各々について、major, minor, diminish, augment(“I”, “Im”, “Idim”, “Iaug”, ...)を割り当てることにより48ビット、同様に4和音の場合における7th, major7th, 6th, sus4を表すために4ビット(“7”, “M7”, “6th”, “sus4”)を割り当てる。

また、ユニット数は以下のように決定した。

(A) 3章のSRNSMと同様のユニット数を用いた。

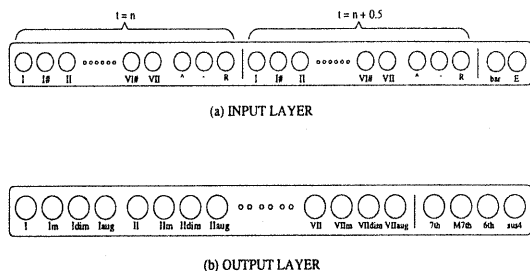


図 5: ニューラルネットの入出力層 (B)

学習定数については、先の実験で最も評価の高かった値を用いた。

- (B) 入出力ユニット数はニューラルネットの入出力の項でも述べたように、入力ユニット数が 32, 出力ユニット数が 52 と決定した。隠れユニット数は (A) と同様 50 とした。学習定数は (A) と同様、3 章の実験で最も評価の高かった値を用いた。

#### 4.1.2 学習データ

(A), (B) のシステムにおける学習データについて以下に示す。

- (A) 3 章で使用した 5 曲に「山口さんちのつとむくん」, 「1 年生になったら」, 「どんぐりころころ」の 3 曲を加え、計 8 曲を学習データとして用いた。曲間には初期化ビット “E” を挿入して区切りをいれた。
- (B) 日本のロックバンドの一つである The Yellow Monkey の曲より短調の曲を 10 曲「天国旅行」, 「BURN」, 「So Young」, 「My Winding Load」, 「追憶のマーメイド」, 「Spark」, 「離れるな」, 「球根」, 「Tactics」, 「熱帯夜」のメロディーとコード (転調部分は除く) を学習データとして用いた。曲間には初期化ビット “E” を挿入して区切りをいれた。

#### 4.2 実験結果とその評価

- (A) 入出力としてダイアトニックスケールのみに対応するシステム (A) によって自動生成されたコード進行の一例として、「お正月」のメロディーを与えた場合の結果を 6 に示す。
- (B) 入出力としてダイアトニックスケール以外にも対応するシステム (B) によって自動生成されたコード進行の一例として、The Yellow Monkey の「花吹雪」のメロディーを与えた場合の結果を図 7 に示す。

なお、譜面上の音符が与えたメロディーであり、五線の下のコード (度数) が自動生成されたコード (度数) である。

図 6: (A) 「お正月」のコード進行出力結果

図 7: (B) 「花吹雪」のコード進行出力結果

アンケートによる主観評価として、作曲の経験者 10 人と未経験者 10 人に、入力したメロディーとともに実験結果により生成されたコード進行を試聴してもらい、自然か不自然かの 5 段階の評価 (1 - 不自然, 5 - 自然) をつけてもらったところ、表 3 のような結果となった。

表 3: 主観評価

入出力	曲名	評価		
		経験者	未経験者	平均
(A)	たきび	3.9	4.1	4.0
	きらきら星	3.8	4.0	3.9
	お正月	4.4	4.4	4.4
(B)	花吹雪	4.6	4.4	4.5
	青い契り	4.3	4.3	4.3

また、感想として次のようなものが挙げられた。

- (1) コード進行に一部不自然な部分がある。
- (2) コードの変化が早すぎてせわしない感がある。
- (3) (B) のシステムによる曲は、The Yellow Monkey のコード進行に雰囲気似ている。

(1) に関しては、コード進行に D-S-T の逆進行のカデンツァや、スケール外のコードがコード進行が不自然な感を与えたと考えられる。

(2) に関しては、(A) のシステムの学習データ自体のコードの変化が早かったので、生成されたコード進行もせわしく感じたと思われる。また、「花吹雪」

の4小節目のI-IV-Vのコード進行は、単位時間おきにコードが変化しているため、せわしない感を与えたと考えられる。

(3)に関しては、IVm, IIIが頻繁にコード進行内に用いられている点や、特徴的な3度進行などがThe Yellow Monkeyの曲中においてよく用いられるからであると考えられる。これは特徴のあるThe Yellow Monkeyらしいコード進行が学習されたという興味深い結果と言える。

## 5 考察および今後の課題

### 5.1 コード進行の自動生成の問題点とその改善法

コード進行の自動生成の実験において、(1)コード進行が一部不自然である、(2)コードの変化が早すぎるといふ二つの問題点が挙げられた。

(1)については、学習データを増やすなどの方法により、ネットワークの汎化能力を上げることにより改善されると考えられる。

(2)については(1)と同様に汎化能力を上げるとともに、学習データをコードの変化が比較的緩やかなものを選び入力として時間的な情報を詳しく与えることにより改善されると考えられる。

### 5.2 コード進行の自動生成のシステムの出力力について

実際の楽曲中におけるコードは前後のメロディー、楽曲中における時間的な情報、以前のコード進行などによって決定される。

本研究ではシステムの入力としてメロディー、小節の頭かどうか、曲の終わりかどうかの情報を与えた(以前のコード進行の情報はシステムの内部状態として表現される)。したがって、時間的な情報、以前のコード進行の情報に比べて、前後のメロディーの情報が比較的大きく出力に反映されていると思われる。そこで、それぞれの情報をどの程度まで与えるかについてより検討していく必要があると思われる。

### 5.3 コード進行の自動生成システムの今後の課題

本研究においては出力結果の評価として前節のような様々な問題点が挙げられたため、実用的に用いるためには様々な改良が必要と思われる。

また、本研究のコード進行自動生成システムを発展させていくにあたって、具体的に次のような案が挙げられる。

#### 5.3.1 作曲者別のコード進行生成

本研究においては、すべて同じ作曲者によって作られた曲を学習データとして用いた。そのため、実験の結果生成されたコード進行はその作曲者の癖を反映したものとなった。今後は学習データとなる曲を複数人に作曲されたものを用いて、入力部に作曲

者ビットを加えて作曲者の情報を学習させることにより、好みの作曲者を選んでその作曲者の癖を反映した出力を得られるようにしていく。

#### 5.3.2 展開ごとのコード進行生成

一般のポップスの楽曲には例えばイントロ、Aメロ、Bメロ、サビなどの展開のパターンがあり、パターンごとのコード進行の流れがある。本研究のシステムにおいてはそのような展開について考慮されていないので、長期の入力に対して展開ごとのコード進行を生成することはできない。

そこで今後はシステムの入力としてイントロ、Aメロ、Bメロ、サビなどの情報を与えて、展開ごとのコード進行生成を可能にしていく。

#### 5.3.3 生成されるコードの種類を増加

本研究のシステムの出力部は5和音やオンコードに対応していない。今後は5和音やオンコードも出力できるようにしていきたい。

## 参考文献

- [1] J.L. Elman, Finding structure in time, "Cognitive Science", vol.14, pp.179-211, 1990.
- [2] 木ノ内 誠, 萩原 将文, "複素ニューロンによる時系列の学習", 電学論, vol.116-C, no.7, 748-754, 1996.
- [3] 木ノ内 誠, 萩原 将文, "複素リカレントニューラルネットワークを用いたメロディの記憶と想起", 情報処理学会論文誌, vol.39, no.5, 1232-1238, 1998.
- [4] 菊地進一, 中西正和, "短期記憶を用いたリカレントニューラルネットワークと高速な構造学習法", 電子情報通信学会技術研究報告, NC99-67, 1999.
- [5] S. Kikuchi and M. Nakanishi, "Learning Time Series by Simple Recurrent Network with Short-term Memory", Proc. of IASTED International Conference on Modelling, Identification and Control, pp.44-48, 2000.
- [6] 菊地進一, 中西正和, "短期記憶を用いたシンプルリカレントネットワークによるカオス時系列の短期予測", 情報研報, MPS-29-4, 2000.
- [7] T.J. Sejnowski and C.R. Rosenberg, "Parallel networks that learn to pronounce English text", Complex Systems, vol.1, pp.145-168, 1987.
- [8] A. van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm", Neural Networks, vol.5, no.3, pp.465-471, 1992.
- [9] 堀野 羽津子, 原 礼彦 "ピアノのおけいこ1", 成美堂出版, 1996.