Vol.2018-MUS-120 No.1
2018/8/21

ニューラルネットワークによる自動和声付けのための和音表現方法の検討

コンヴェール マクシム[1,a)]　深山 覚[2,b)]　中野 倫靖[2,c)]　高道 慎之介[1,d)]　猿渡 洋[1,e)]
後藤 真孝[2,f)]

概要：ニューラルネットワークは自動和声付けにおいて有望な技術である．膨大なデータセットを元に，入力と出力の複雑な依存関係を学習することができるため，旋律と和音の依存関係も扱うことができる．ニューラルネットワークの性能はその入力と出力情報の表現方法が強く影響する．しかし，従来の自動和声付け研究では，出力情報である和音の表現方法について深くは検討されておらず，テンションノートといった和音の詳細な構造が最大限活用されてこなかった．和音の表現方法を変えることで，旋律と和音の関係を更に細かく学習できると考えられる．そこで本研究では，和音の表現方法の違いが Recurrent Neural Network (RNN) による自動和声付けの性能にどれほど影響するかを調査する．従来の表現方法を含む 4 つの異なる和音表現方法に基づいて Gated Recurrent Unit (GRU) を用いたニューラルネットワークを構築し，それらの性能を比較した．実験の結果，和音の構成音を陽に表現した表現方法を用いると，従来の和音ラベル形式を使った場合に近い性能に達成するだけでなく，構成音の細かな違いに対応できる多機能な自動和声付けモデルの構築を可能とすることがわかった．

1. はじめに

和声付けとは与えられた旋律に適切な和音をつけることである．旋律を和声付けする際，和声付けの方法は一つに限らないため，その違いによってある作曲家のスタイルや，特定の音楽ジャンルの特徴を作曲で表現できる．さらに，和声付けは音楽の経験と知識を必要とするため，自動で和声付けをすることができれば，技術によって人々の作曲を助けることができ，結果としてより多くの人が多様な音楽を作曲できるようになる．自動和声付けには二つの課題があり，一つめは合唱における 4 つの声部 (ソプラノ，アルト，テノール，バス) の内一つの声部 (多くの場合ソプラノかバス) が与えられた際に，他の 3 つの声部を作曲する問題である．また，二つ目は，決まった旋律に対し適切な和音の系列を作曲する課題であり，本稿では後者を扱う．

従来，自動和声付けには様々な方法が提案されてきたが，中でも統計的機械学習による方法が現在の最先端である [1]．特に，Hidden Markov Model (HMM) による手法がこの 10 年多く提案されていて [2–4]，ニューラルネットワークを用いた手法も提案されている [5–15]．Hild らが提案した HARMONET はバッハスタイルのコラールの生成を目的とし [5]，その後，MELONET という旋律の和声付けの様々なサブタスクを複数のニューラルネットワークで扱う手法に応用された [6]．リアルタイムに旋律の和声付けを行うことに特化したモデルも提案され，多層パーセプトロンをルールベースシステムで拡張する手法 [7] や，旋律の系列情報を活用するモデルもある [8]．近年 Recurrent Neural Network (RNN) による旋律生成モデルも提案されている．Long Short-Term Memory (LSTM) [16] や Gated Recurrent Unit (GRU) [17,18] が多声部の旋律 [9,10]，和音の系列 [19] やコラール [11] の生成に用いられている．Hadjeres らは DeepBach で擬似ギブスサンプリングを用いた Bidirectional RNN でバッハスタイルのコラールを生成している [12]．Generative Adversarial Network (GAN) による教師なし学習手法も近年音楽生成に数多くの応用例が提案されている．MidiNet と MuseGAN は畳み込みニューラルネットワーク (Convolutional Neural Network:

[1] 東京大学 大学院情報理工学系研究科
The University of Tokyo, Graduate School of Information Science and Technology, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.
[2] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8568, Japan.
[a)] maxime_convert@ipc.i.u-tokyo.ac.jp
[b)] s.fukayama@aist.go.jp
[c)] t.nakano@aist.go.jp
[d)] shinnosuke_takamichi@ipc.i.u-tokyo.ac.jp
[e)] hiroshi_saruwatari@ipc.i.u-tokyo.ac.jp
[f)] m.goto@aist.go.jp

CNN) を GAN によって学習し，旋律・和音・リズムなどを同時に生成している [13, 14]．また SeqGAN は多声部の旋律を生成する [15].

従来の研究では，自動和声付けモデルの構築にあたり，それぞれの和音をラベルで表現したデータを用いていた．しかし，この方法では，テンションノートなどを含む複雑な和音を表現するには多くのラベルが必要となってしまう．これは根音と Major/Minor の組合せだけからなるシンプルなラベルの場合と比べて，ラベルごとのデータ数が非常に少なくなる要因となり，結果としてモデルの学習が難しくなる．コラール生成の研究では，和音のラベルを使う代わりに，各声部の音高そのものを生成するが，この方法によってテンションノートを含む和音のような複雑な和音が生成できるのかは検証されていない．このように，現状としてはシンプルな和音ラベルでモデルを構築するしかない状況であり，これはテンションを含む和音を使うといった多機能な自動和声付けモデルを開発する上での問題となっている．

そこで本研究では, 和音の表現方法に応じて複数のニューラルネットワークを学習し，それらモデルの性能を比較検証する．テンションを多く含む音楽ジャンルでの検証のため，モデルの学習には 456 曲のジャズ楽曲を用いる．実験の結果，和音構成音を陽に表す表現方法を用いたモデルの性能は，従来のラベルを用いる場合とほぼ同等な性能を達成した上，構成音の細かな違いに対応できる機能的な自動和声付けシステムの構築に役立つことがわかった．

2. 準備：和音に関連する音楽用語

本研究を理解する上で必要な和音に関する音楽用語について述べる．

2.1 和音の表記

現代の和音の記法では，和音は根音を表す root label と和音構成音を表す suffix の組み合わせで表現される．根音は 1 オクターブに含まれる 12 個のピッチクラスの音名で表し，根音以外の和音構成音は根音からの音程の組合せによって表される．

2.2 三和音，四和音，テンション

最も基礎的な和音は，三つの音高を長 3 度や短 3 度の音程で組合せることで形成され，これを三和音と呼ぶ．三和音には長三和音 (Major)，短三和音 (Minor)，増三和音 (Augmented)，減三和音 (Diminished) などがある．和音構成音のうち第 3 音を根音から長 2 度や完全 4 度の音程に加えた和音を，特にサスペンデッド (Suspended) コード (それぞれ Suspended-2nd と Suspended-4th) と呼ぶことがある．

三和音は更に音高を重ねることで四和音へ拡張できる．例として，長三和音の根音から長 6 度の音程にある音高を重ねた 6th の和音や，短三和音の根音から短 7 度の音程にある音高を重ねた minor7th の和音などがある (例：F^6, Cm^7)．更に追加すると，追加された音高はテンションと呼ばれることがあり，根音からの音程が追記されて表記される (例：$B^{\flat 7\flat 9\flat 13}$).

2.3 和音の転回・スラッシュコード

根音が和音構成音のうち最も低い音高であるように和音を構成した場合，この和音は基本型にあるという．根音以外の和音構成音を最も低い音高とすると，同じ和音であっても響きの異なる和音をつくることができる．このような和音の変換のことを転回という．また様々な効果を狙って，和音構成音にない音を敢えて最も低い音高とすることもある．このような和音は転回の場合を含めてスラッシュコード (slash chord) と呼ばれることがある．スラッシュコードは，最も低い音高の音名をスラッシュに続いて明記することで表記される (例：$D^{\flat}m^7/G^{\flat}$).

3. 検証に用いるデータセットの構築

本研究では，Weimar Jazz Database (WJD) [20, 21] による 456 曲のジャズ曲のソロと和音の系列を元として学習データを構築する．ジャズにおいては，他のジャンルよりも多様な和音が用いられているため，本研究の目的に適切であると考え，本データセットを選んだ．

3.1 学習に用いる曲の選定

データセットから学習に用いる曲を選定するにあたって，自動和声付けに適した曲のみを選出する必要がある．具体的には，曲中で一種類の和音しか使われていない場合，あるいは和音系列が非常に単純な曲 (二種類の和音の繰り返しによる曲など) はモデルの学習において偏ったデータと見なして用いない．また西洋音楽で一般的な拍子である四分の四拍子の曲のみ選曲する．

3.2 三和音データセットと四和音データセットの構築

WJD には合計 399 個の和音ラベルが存在し，テンション付きの和音やスラッシュコードなどの三和音や 6th と 7th 以外の比較的頻度の低い suffix を含む和音ラベルもある．これら頻度の低いラベルを頻度の高いラベルと一緒に学習データに含めると，ニューラルネットワークの学習に必要とされるラベルごとの膨大の学習データという要件を満たすことができない．そこで，本来の和音ラベルの複雑な部分を切り捨て，三和音と四和音のどちらであるかに着目してデータを分割し，三和音データセット (Triads data) と四和音データセット (Sevenths data) の二つのデータセットを構築した．

三和音データセットと四和音データセットはそれぞれ 5

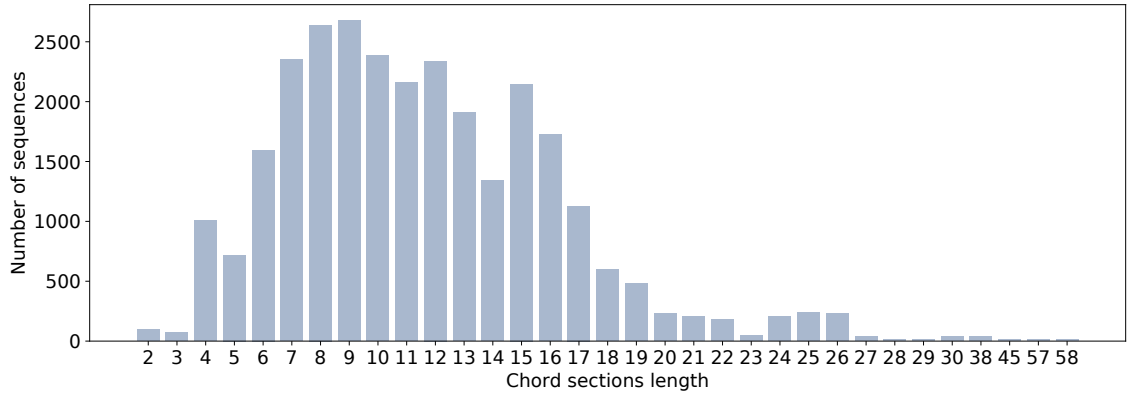

図 1 和音系列の長さごとの頻度．系列の長さは拍単位ではなく和音単位で求めているため，同じ和音系列の長さでも複数小節のものと 1 小節以下のものが同じ系列長のサンプルとして扱われている．

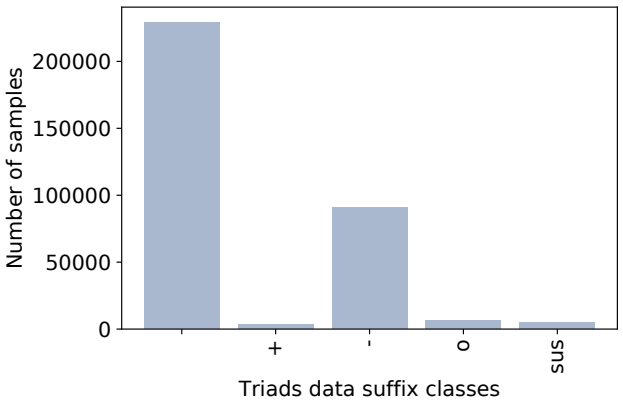

図 2 データ中の三和音の頻度．左から右の順に Major，Augmented，Minor，Diminished，Suspended の和音の頻度を表す．ここでは Suspended-2nd と Suspended-4th の区別をしていない．

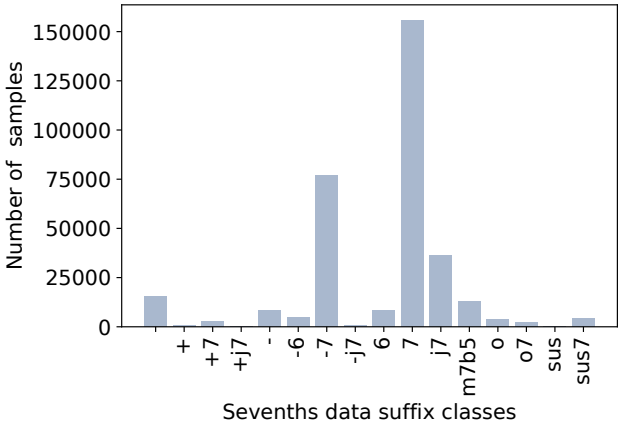

図 3 データ中の四和音の頻度．左から右の順に Major，Augmented，Augmented-7th，Augmented major-7th，Minor，Minor-6th，Minor-7th，Minor-major-7th，Major-6th，7th，Major-7th，Half-diminished-7th，Diminished，Diminished-7th，Suspended，Suspended-7th を表す．

つと 16 つの suffix のラベルを含み，12 種類の根音を表す表記と組み合わせると，三和音データセットには 50 種類，四和音データセットには 192 種類の和音ラベルが存在する．それぞれのデータセットにおける suffix のラベルの頻度を図 2 と図 3 に示す．なお根音の表記にシャープ (♯) が含まれる場合には，異名同音であるフラット (♭) を使った表記へと変換し，根音の表記を統一した．

3.3 旋律の扱い

データのスパースネスを軽減するため，データセット中の旋律に含まれる C3 から C8 の音域にわたる 60 種類の音高は，オクターブを無視した 12 種類のピッチクラスへと変換する．また無音を示すクラスを追加する．さらに各旋律の音高系列長を統一するために，最長の系列長に長さを揃えるようにパディング用のクラスを追加し，合計 14 クラスとした (音高 12 クラス，無音 1 クラス，パディング 1 クラス)．

3.4 和音系列の扱い

自動和声付けの学習データとしては，和音のデータは系列としての情報を保ったままであることが望ましい．系列の長さが長いほど，時間的に離れた位置にある和音同士の関係が分析できる一方，曲中には楽曲構造の区切りがあり，その区切りをまたいでの和音同士の関係は考慮する必要は低いと考えられる．そこで本研究では WJD でアノテーションされているセクションの境界を区切りとして，和音系列を各曲からサンプルした．この方法でサンプルされた和音系列の系列長ごとの頻度を図 1 に示す．なお曲中に繰り返しがあって同じ和音系列が複数回サンプルされる場合があっても，その和音系列とペアとなるソロの旋律の音高が異なっていることが多いため，多重なデータのサンプルが行われている可能性は低いと考えた．

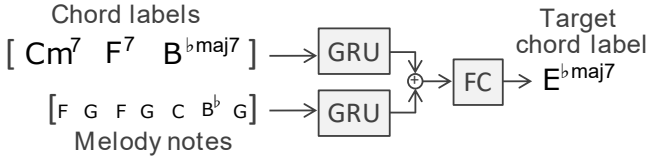

図 4 モデル 1：ベースラインモデル．ラベル式で和音を表現する．(+) はエンコード後のベクトルの連結を表し，[FC] は fully-connected 層を表す．

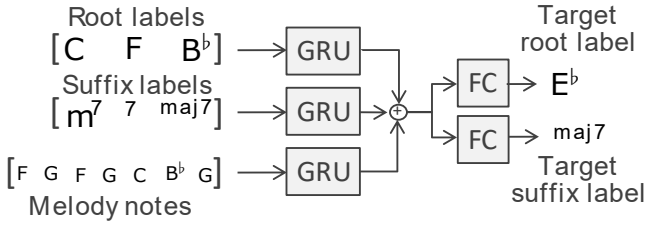

図 5 モデル 2：根音＋音程モデル．根音のラベルと音程のラベルを別々に扱うモデル．

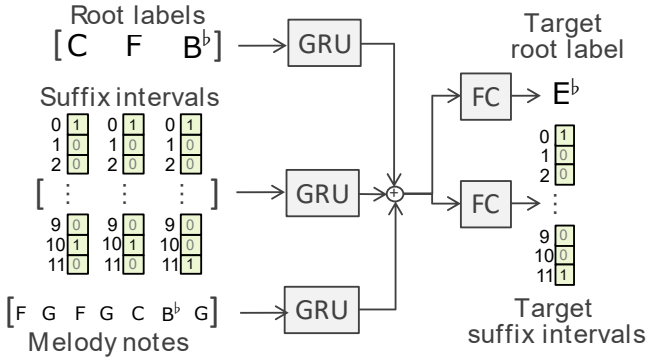

図 6 モデル 3：根音＋間隔モデル．根音をラベルとして扱い，音程の有無をバイナリベクトルで表す．

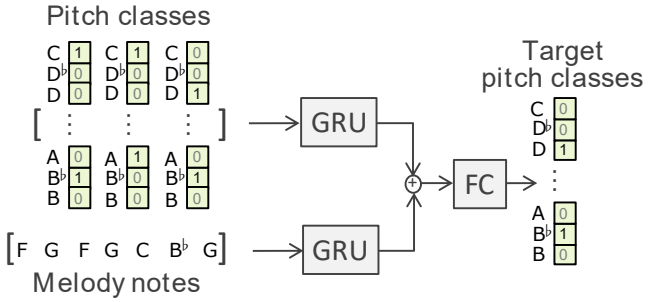

図 7 モデル 4：構成音モデル．和音を 12 個の構成音の有無をベクトルで表現する．

3.5 データの拡張

旋律と和音のペアを移調することで学習データ量を増やすデータ拡張を行った．元データのコードの根音とメロディーの音高を元の調以外の 11 個の調へと移調した．このデータ拡張法は従来法においても有用な方法として用いられている [12]．

4. 自動和声付けのためのモデル構造

和音の表現方法が異なる 4 つのモデルの性能を比較する．そのため，層の数を増やすことによるモデルの記述能力を制限し，なるべく少ない層の数でモデルを設計し，性能を比較する．モデルは現在の和音に伴って演奏される旋律の音高系列と，現在の和音に先行する和音の系列の 2 つの系列を受け付ける．そして出力として和音の推定結果を出力する．

それぞれのモデル間において，和音の表現方法のみが異なる点であるようにし，その他の過程については共通のニューラルネットワークの構造を用いるようにする．モデルの入力層では，系列情報が扱えるように，RNN を用いる．本研究では Recurrent Cell として，Gated Recurrent Unit (GRU) [18] を用いた．また出力層においては Fully-Connected (FC) 層を用いた．

和音の表現方法に応じて変形を加えた 4 つの自動和声付けのためのモデルを設計した．それぞれの構造を図 4，5，6，7 に示す．それぞれのモデルにおける和音の表現方法を以下に述べる．

モデル 1 (図 4): コードネームをそのままラベルとして用い和音を表現する．コードネームは音楽において一般的に用いられている表現であり，その表現をそのまま用いているため，このモデルをベースラインモデルと呼ぶことにする．

モデル 2 (図 5): 和音の根音の音名と，コードネーム中で和音の種類を表している suffix をそれぞれ独立なラベルとして扱い，和音をこれら 2 種類のラベルの組合せで表現する

モデル 3 (図 6): 和音の根音の音名をラベルとして扱い，コードネーム中の suffix を 12 次元のバイナリベクトルで表現する．このベクトルは，和音構成音の根音からの半音単位による音程を表す．13 半音以上の音程については，12 で割った余りの音程と同じとみなした．

モデル 4 (図 7): 和音の構成音をピッチクラスに対応する 12 次元のバイナリベクトルによって表現する．このモデルにおいては，根音も和音構成音のいずれもラベルを用いて表現されていない点に注意する．

モデル 1〜3 でラベルをニューラルネットワークで扱うにあたっては，one-hot ベクトルを用いた．

モデル 1 とモデル 2 は，コードネーム表記の根音と suffix を独立に扱うかどうかに違いがある．モデル 2 において根音と suffix を独立に扱い，それらの組合せを可能とすることで，根音と suffix の組合せによって，学習データでは現れなかった和音が扱える．

モデル 2 とモデル 3 の間では，和音の構成音が suffix によるラベルで表現されるか，根音からの音程の組合せに

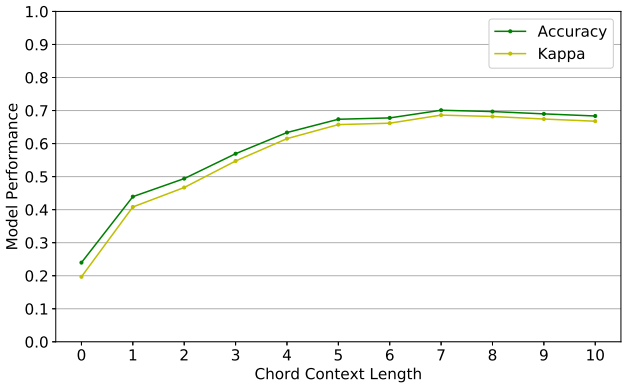

図 8 学習に用いた先行和音系列の長さによるベースラインモデルの性能.

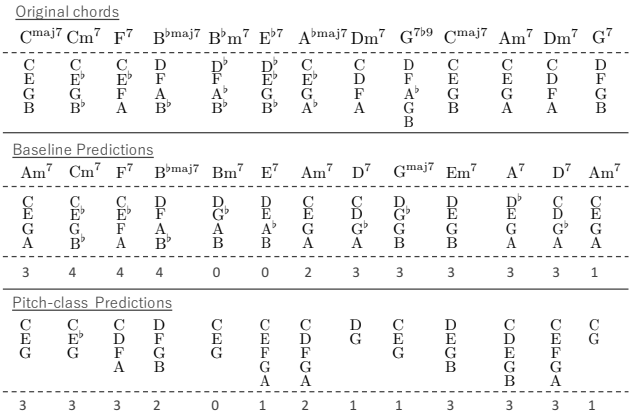

図 9 四和音データセットによる学習を元にアフタヌーン・イン・パリの旋律に対して行った和声付け．一行目は本来の和音とそれに対応する構成音を示す．二行目と三行めはそれぞれベースラインと構成音モデルの結果を表し，下に正解の構成音の数を述べる．

よって表現されるかという違いがある．モデル 3 においては，根音からの音程関係によってあらゆる和音構成音が表せるので，学習データに現れなかった和音の構成音を表現できる．

最後にモデル 4 では，和音をラベリングするという概念から完全に離れて，和音の構成音をピッチクラスの組合せによって表現している．これによって，モデル 4 は学習データに含まれない和音についても表現できる．

モデル 1 からモデル 4 に拡張するにあたって，和音を構成する音の組合せの自由度が大きくなり，学習データに含まれていないような和音構成音の組合せによる多彩な自動和声付けが行える可能性がある．一方で和音構成音の組合せ方の自由度が大きくなるに伴い，モデルのパラメータ数は増大しモデルの学習が難しくなると考えられる．

このように，モデルによって表現できる和音の自由度と，学習の結果得られるモデルによる自動和声付けの性能にはトレードオフに近い関係があると考えられることから，どの和音の表現が実質的に自動和声付けにおいて適しているのかを検証するために，次節に述べるような実験を行う．

5. 実験：自動和声付け性能の比較検証

異なった和音表現方法に基づくニューラルネットワークを，準備した 2 種類のデータセットを用いて学習し，その結果を自動和声付けの性能によって評価することで，それぞれの和音表現の性質を議論する．

5.1 ニューラルネットワークの実装と学習

ニューラルネットワークは TensorFlow [22] を通じて実行されるライブラリ Keras [23] を用いて構築した．学習率などのハイパーパラメータによってモデル間で性能差が生じるのを避けるため，すべてのモデルを同じハイパーパラメータを用いて学習した．また実験途中でハイパーパラメータを調節するといった，性能を上げるためのチューニングを行わなかった．GRU のユニット数は 128 とした．学習はミニバッチを用いて行い，バッチサイズは 32 に設定した．最適化法には Adam [24] を用い，エポック数は 50 を上限として，その中で得られた最善のものを結果とした．データは層化抽出法によって 80%-10%-10%の割合で train-validation-test のセットへと分割して用いた．

5.2 評価方法

自動和声付けの性能は実際の曲における和音と一致しているかによる推定精度によって評価した．なおテストに用いる楽曲によっては，和音ごとの頻度が大きく異なることがあるため，推定精度とともに Cohen's Kappa スコア [25] を算出した．

推定精度は異なるコード表現方法ごとに求めることができるが，表現方法がことなっているために，精度の値をそのまま比較して議論をすることが難しい．そこで和音の推定結果を全てモデル 4 の和音表現へと変換し，その表現を用いて計算した精度を比較した．

5.3 事前実験：入力の和音系列長の決定

モデルへの入力である一定長の和音系列に，どのような系列の長さを用いればよいのかを決定するため，事前実験を行った．三和音データセットを用い，ベースラインモデルを用いて学習・推定を行い，系列長に応じた精度を比較した．結果を図 8 に示す．系列長が長くなるほど推定精度が向上し，系列長が 7 のときに最も高い推定精度が得られることがわかった．これに従い，以下においてモデルへの入力系列の系列長を 7 として実験を行った．

5.4 三和音データセットを用いた評価

三和音データセットを用いた場合の和音の表現方法の異なるモデル間の精度比較結果を表 1 に示す．またモデル

表 1 異なる和音表現方法による自動和声付け性能比較．モデル 1 はコードラベル，モデル 2 は根音とコードラベルの suffix それぞれのラベルの組合せ，モデル 3 は根音と suffix に表された音程を 12 次元バイナリベクトルで表現したもの，モデル 4 は和音構成音をすべて 12 次元バイナリベクトルで表現したもの．精度はすべてモデル 4 の表現である和音構成音のベクトル表現へと変換して求めた．参考値としてそれぞれのモデルでの和音の表現を用いて計算された制度を括弧内に示す．

	Model 1 Baseline	**Model 2 Root+Suffix**	**Model 3 Root+Intervals**	**Model 4 Pitch-class**
Accuracy	0.8865 (0.7009)	0.8451 (root:0.6438, suffix:0.8272)	0.8465 (root:0.6741, suffix:0.9574)	0.8803
Kappa	0.6969 (0.6862)	0.5867 (root:0.6115, suffix:0.5819)	0.5900 (root:0.6445, suffix:0.8864)	0.6632
F1	0.7725	0.6899	0.6922	0.7403
# Parameters	145,215	171,286	175,131	112,524

表 2 ベースラインモデルと構成音モデルの結果比較．三和音データセット (Triads) および四和音データセット (Sevenths) を用いて精度を計算した．またモデルのパラメータ数も併せて示した．

	Model 1: Baseline		**Model 4: Pitch-class**	
	Triads	**Sevenths**	**Triads**	**Sevenths**
Accuracy	0.8865	0.8810	0.8803	0.8622
F1	0.7725	0.8178	0.7403	0.7768
Kappa	0.6969	0.7295	0.6632	0.6775
Parameters	145,215	229,827	112,524	112,524
Notes per prediction	2.994	3.926	2.997	3.892

4 による和音表現方法へと変換する前の表現を用いたときの，各モデルの推定精度を参考値として括弧の中に示す．モデル 2 とモデル 3 はベースラインモデルと構成音モデルよりも精度が低かった．根音と音程を別々に学習すると，学習パラメータ数が他モデルより多いにもかかわらず，性能向上に貢献していないと考えられる．モデル 4 はベースラインよりやや低い精度であるものの，同等な性能を達成した (モデル 1: 0.8865，モデル 4: 0.8803)．

5.5 四和音データセットを用いた評価

更に，和音の構成音が多い場合の評価を行うことを目的とし，四和音データセットを用い，三和音データセットを用いた際に性能の良かったベースラインと構成音モデルを比較した．表 2 に示すように，ベースラインモデルの方が構成音モデルより高性能を達成した．

5.6 自動和声付け結果の例

ジャズスタンダードである「アフタヌーン・イン・パリ」のメロディに対して和音を推定し，それぞれの和音の表現方法の性質を分析した．自動和声付けの結果を図 9 に示す．この曲は学習データセットに存在しない曲であり，各モデルにとって未知なものである．1 小節中の平均音符数は 3.92 であり旋律がやや複雑な曲である．曲冒頭のコードは正解とほぼ一致した．

ベースラインモデル (モデル 1) は Minor-7th を 7th (例：$Dm^7 \rightarrow D^7$，$Am^7 \rightarrow A^7$) に推定したり，Major-7th を転回した場合それに最も近い Minor-7th(例：$C^{maj7} \rightarrow Am^7$，$Em^7$) に誤ったりするが，5 つめと 6 つめの和音を除けば構成音としてはほぼ正解している．一方モデル 4 は，ベースラインモデルの結果を超えるような良い和音系列が生成できなかった．構成音モデルは和音ごとの構成音の数を誤る問題があり，二和音や五和音などに誤って推定することがあった．

6. 考察

三和音データセットを用いた実験では，コードネームをそのままラベルとするベースラインモデルと，和音構成音を 12 次元ベクトルで表現するモデルが高い性能を達成した．四和音データセットを用いた実験には，ベースラインモデルの性能がもっとも良かった．

和音構成音を 12 次元ベクトルで表現する方法は，コードネームをそのままラベルとする場合と比べて，モデルのパラメータ数の増加を抑えつつもテンションを含む和音を表現できる利点があることもわかった．ラベル形式で学習を行う際，和音が複雑なものになるに伴い，推定クラス数が増加し，それにともなってモデルパラメータが増加する．実際に実験では，用いるデータセットを三和音データセットから四和音データセットに変更したことで，ベースラインモデルにおけるコードラベルの種類が 60 種から 192 種に増加し，その結果ベースラインモデルで必要とされる学習パラメータが大幅に増加している (145,215 から 229,827)．一方，和音構成音を 12 次元ベクトルで表すモデル (モデル 4) はデータの複雑さにかかわらず学習パラメータ数は一定である．

Confusion matrix を算出し，比較的性能が高かったコードネームをそのままラベルとする方法と構成音を 12 次元ベクトルで表現する方法の性能を詳細に比較し

表 3 ベースラインモデルの推定結果の confusion matrix.

Triads		Sevenths	
TN = 0.6937	FP = 0.0567	TN = 0.6139	FP = 0.0601
FN = 0.0568	TP = 0.1928	FN = 0.0589	TP = 0.2671

表 4 構成音モデルの推定結果の confusion matrix.

Triads		Sevenths	
TN = 0.7096	FP = 0.0407	TN = 0.6243	FP = 0.0521
FN = 0.0790	TP = 0.1705	FN = 0.0858	TP = 0.2398

た．その結果を表 3 と表 4 に示す．ベースラインモデルの False-Positive と False-Negative の比率は三和音データセットと四和音データセットで 0.0567/0.0568 ≈ 0.998 と 0.0601/0.0589 ≈ 1.02 になり，双方のデータセットでほぼ 1.0 となった．一方, 和音構成音を 12 次元ベクトルで表すモデルでは 0.0407/0.0790 ≈ 0.515 と 0.0521/0.0858 ≈ 0.607 となり，False-Positive より False-Negative の方が倍近く (およそ 0.5) 高かった．

自動和声付けを実際に用いる場面を考えると，大きな False-Positive は耳障りな余計な音を和音に混ぜてしまう可能性を示唆する．したがって，False-Positive と False-Negative を比較するならば，False-Negative の場合の方が好ましいと考えられる．構成音を 12 次元ベクトルで表す表現方法 (モデル 4) はベースラインモデル (モデル 1) より精度が低いものの，このような誤った和音構成音を生成することを避けられるという点では，応用可能性が高いといえる．

今後の課題として，モデルのハイパーパラメータの調節や，5 つ以上の構成音を持つ和音を用いた実験などを行い，それぞれの表現方法の特質を明らかにすることが挙げられる．特に和音構成音を 12 次元ベクトルで表す方法は，モデルパラメータの増加を抑えながらも，和音の転回型やテンションを含むデータを用いた学習が可能な方法であるため，その方法の詳細な検討は，5 つ以上の和音構成音を含む和音を生成，ヴォイシングを同時に考慮するといった，より高度な自動和声付けモデル構築の足掛かりになると考えられる．

7. まとめ

本稿では，ニューラルネットワークによる和音表現方法の異なるモデルを検討し，精度を比較した．和音構成音を 12 次元ベクトルによって和音を表現する場合，コードネームをラベルとする表現方法を用いた場合に匹敵する精度が得られた．この方法はテンションを含む和音といった複雑な和音をパラメータ数の増加を抑えつつ扱うことが可能な方法であり，5 つ以上の和音構成音を含む和音の生成やヴォイシングの考慮といった高度な自動和声付けを行うシステムの構築に有用なことがわかった．

謝辞 本研究の一部は JST ACCEL (JPMJAC1602) の支援を受けた．

参考文献

[1] D. Makris, I. Kayrdis, and S. Sioutas, “Automatic melodic harmonization: An overview, challenges and future directions,” *Trends in Music Information Seeking, Behavior, and Retrieval for Creativity*, pp. 146–165, 2016.

[2] R. Groves, “Automatic harmonization using a hidden semi Markov model,” in *Proc. AIIDE*, Boston, U.S.A., Oct. 2013, pp. 48–54.

[3] S. A. Raczyński, S. Fukayama, and E. Vincent, “Melody harmonization with interpolated probabilistic models,” *Journal of New Music Research*, vol. 42, no. 3, pp. 223–235, Jun. 2013.

[4] M. Kaliakatsos-Papakostas and E. Cambouropoulos, “Probabilistic harmonization with fixed intermediate chord constraints,” in *Proc. ICMC-SMC*, Athens, Greece, Sep. 2014, pp. 1083–1090.

[5] H. Hild, J. Feulner, and W. Menzel, “HARMONET: a neural net for harmonizing chorales in the style of J.S.Bach,” in *Proc. NIPS*, Denver, U.S.A., Dec. 1991, pp. 267–274.

[6] J. Feulner and D. Hönel, “MELONET: Neural networks that learn harmony-based melodic variations,” in *Proc. ICMC*, Aarhus, Denmark, Sep. 1994, pp. 121–124.

[7] U. S. Cunha and G. Ramahlo, “An intelligent hybrid model for chord prediction,” *Organised Sounds*, vol. 4, no. 2, pp. 115–119, Jun. 1999.

[8] D. Gang, D. Lehman, and N. Wagner, “Tuning a neural network for harmonizing melodies in real-time,” in *Proc. ICMC*, 1998.

[9] K. Goel, R. Vohra, and J. K. Sahoo, “Polyphonic music generation by modeling temporal dependencies using a RNN-DBN,” in *Proc. ICANN*, Hamburg, Germany, Sep. 2014, pp. 217–224.

[10] Q. Lyu, Z. Wu, J. Zhu, and H. Meng, “Modeling high-dimensional sequences with LSTM-RTRBM: application to polyphonic music generation,” in *Proc. IJCAI*, Buenos Aires, Argentina, Jul. 2015, pp. 4138–4139.

[11] F. Liang, M. Gotham, M. Johnson, and J. Shotton, “Automatic stylistic composition of Bach chorales with deep LSTM,” in *Proc. ISMIR*, Suzhou, China, Oct. 2017, pp. 449–456.

[12] G. Hadjeres, F. Pachet, and F. Nielsen, “Deepbach: a steerable model for bach chorales generation,” in *Proc. ICML*, Sydney, Australia, Aug. 2017, pp. 1362–1371.

[13] L. C. Yang, S. Y. Chou, and Y. H. Yang, “MidiNet: A convolutional generative adversarial network for symbolic-domain music generation,” in *Proc. ISMIR*, Suzhou, China, Oct. 2017, pp. 324–331.

[14] H. W. Dong, W. Y. Hsiao, L. C. Yang, and Y. H. Yang, “MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment,” in *Proc. AAAI*, New Orleans, U.S.A., Feb. 2018, pp. 34–41.

[15] S. Lee, U. Hwang, S. Min, and S. Yoon, “A SeqGAN for polyphonic music generation,” vol. arXiv:1710.11418, 2017. [Online]. Available: https://arxiv.org/abs/1710.11418

[16] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[17] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," vol. arXiv:1409.1259, 2014. [Online]. Available: https://arxiv.org/abs/1409.1259

[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," vol. arXiv:1412.3555, 2014. [Online]. Available: https://arxiv.org/abs/1412.3555

[19] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using BLSTM networks," in *Proc. ISMIR*, Suzhou, China, Oct. 2017, pp. 621–627.

[20] J. Abeßer, K. Frieler, M. Pfleiderer, and W. G. Zaddach, "Introducing the Jazzomat project - jazz solo analysis using music information retrieval methods," in *Proc. CMMR*, Marseille, France, Oct. 2013.

[21] "Weimar Jazz Database, https://jazzomat.hfm-weimar.de/index.html."

[22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," in *Proc. of the 12th USENIX Conference on Operating Systems Design and Implementation*, Savannah, GA, U.S.A., 2016, pp. 265–283.

[23] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, U.S.A., Oct. 2013.

[25] J. Cohen, "A coefficient of agreement for nominal scales," *Educational Psychology and Measurement*, vol. 2, no. 1, pp. 37–46, Apr. 1960.