マルチトラック楽曲におけるGANを用いた大楽節の自動生成*

☆松浦　功一郎, 平林　晃, △北原　大地 (立命館大)

1　はじめに

作曲は多大な時間と労力を要する作業であり，高い品質の楽曲を作成することは容易ではない．これを自動化できれば，作曲者の時間と労力を削減できるだけでなく，作曲者へのアイデア提供も可能になる．また，動画配信サービスなどのマルチメディア技術の発展に伴って，著作権の制約を考慮すること無く自由に取り扱える音楽が求められており，こうした需要に応えることができるようにもなる．これらの要求に応えるために，人工知能が自ら作曲法を学び，楽曲を自動生成する技術が研究されている [1, 2]．

多くの自動作曲研究の中でも本研究では，複数楽器が同時に演奏されているマルチトラック楽曲の生成に着目する．このための MuseGAN と呼ばれる手法 [3] は，敵対的生成ネットワーク (Generative Adversarial Networks: GAN) を利用した手法である [4]．この手法では，各トラックの独立性と共通性に注目し，3 種類のネットワークを提案している．しかしこの手法では，実数値のピアノロール生成の後に行う二値化処理が固定されたものになっており，この処理が音価の細分化を引き起こしてしまう．この問題を解決するために，MuseGAN を改良したネットワークとして Binary-MuseGAN [5] が提案されている．この手法では，MuseGAN で固定されていた二値化処理に対しても学習を導入することによって，音価の細分化を防いでいる．また，判別器のネットワーク構造に大幅な変更を加えることによって，二値化処理への学習導入と併せて判別器の精度を向上させ，これにより複雑なマルチトラック楽曲の学習に成功している．一般に，4 小節から成り，半独立した楽想をもった楽句を小楽節と呼ぶ．更に，小楽節が 2 連接することによる 8 小節を大楽節と呼ぶ [6]．上記のマルチトラック楽曲生成手法は 4 小節からなる小楽節を生成するものであり，2 小楽節からなる大楽節を生成するものではない．個々に生成された小楽節を連結するだけでは，音高やリズム変化に一貫性が無く，高品質の大楽節を生成することはできない．

この問題を解決するために本研究では，大楽節生成のための新しい手法を提案する．提案手法では 2 種類の GAN を用いる．第 1 の GAN は，Binary-MuseGAN を簡略化したネットワークを用いることによって，ランダムノイズから前半小楽節を生成する．さらに第 2 の GAN は，既存楽曲の前半楽節から後半楽節への遷移を学習しておくことにより，生成された前半小楽節にスムーズに繋がる新たな小楽節を生成する．提案手法によって大楽節を生成し，得られた結果を学習データである大楽節，および単純連結によって構成した大楽節と比較する．アンケートによる主観評価と，数値指標による客観評価によって，学習データには及ばないものの，提案法が生成した大楽節が単純連結による大楽節を上回る結果が得られることを示す．

2　提案手法

2.1　大楽節生成 GAN

本節では，提案法である大楽節生成 GAN の概要について述べる．大楽節生成 GAN は，前半楽節生成 GAN と後半楽節生成 GAN によって構成される大規模ネットワークである．前半楽節生成 GAN では，Binary-MuseGAN と同様に，楽曲を特徴付ける小楽節単位での学習と生成を行う．提案法では，後半楽節生成 GAN が既存曲の前半楽節から後半楽節への遷移を学習する．これによって，入力の前半楽節からスムーズに繋がる新たな後半楽節を生成する．詳細を順に説明する．

2.2　前半楽節生成 GAN

前半楽節生成 GAN はランダムノイズから前半楽節を生成する GAN であり，Binary-MuseGAN から精錬器を省いた構造を取る．これにより，学習時間の高速化を行う．また，ピアノロールを 16 分音符単位でサンプリングすることで，音価の過剰な細分化を防ぐ．前半楽節生成器および判別器をそれぞれ G_A, D_A で表す．学習データの前半楽節を $\mathbf{X}_A$, ランダムノイズを $\boldsymbol{z}$，生成される前半楽節を $\tilde{\mathbf{X}}_A$ で表せば $\tilde{\mathbf{X}}_A = G_A(\boldsymbol{z})$ である．D_A および G_A の学習には，Binary-MuseGAN と同様に，次式で定義される Wasserstein GAN-Gradient Penalty (WGAN-GP) [7] を用いる：

$$\begin{aligned}\min_{G_A}\max_{D_A} &\mathbb{E}_{\tilde{\mathbf{X}}_A\sim p_g(\tilde{\mathbf{X}}_A)}[D_A(\tilde{\mathbf{X}}_A)]\\ &-\mathbb{E}_{\mathbf{X}_A\sim p_A(\mathbf{X}_A)}[D_A(\mathbf{X}_A)]\\ &+\lambda\mathbb{E}_{\hat{\mathbf{X}}_A\sim p_{\hat{\mathbf{X}}_A}(\hat{\mathbf{X}}_A)}[(\|\nabla_{\hat{\mathbf{X}}_A}D_A(\hat{\mathbf{X}}_A)\|_2-1)^2]\end{aligned}$$

ここで，

$$\hat{\mathbf{X}}_A = \epsilon\mathbf{X}_A + (1-\epsilon)\tilde{\mathbf{X}}_A \qquad \text{s.t.}\quad \epsilon\sim U[0,1]$$

であり，p_A は前半楽節の真分布，p_g は前半楽節の生成分布である．また，$\hat{\mathbf{X}}_A$ は $\mathbf{X}_A$ と $\tilde{\mathbf{X}}_A$ を結ぶ線分上の任意の点であり，ϵ はその位置を決める乱数である．

2.3　後半楽節生成 GAN

後半楽節生成 GAN は前半楽節から後半楽節を生成する GAN である．前半楽節生成 GAN との違いは

*Automatic Eight Bar Phrase Generation Based on GANs for Multitrack Music.　by MATSUURA, Koichiro, HIRABAYASHI, Akira, KITAHARA, Daichi

生成器の構造にある．後半楽節生成器および判別器をそれぞれ G_B, D_B で表す．判別器 D_B には，前半楽節の判別器 D_A と同じネットワーク構造を用いる．一方，生成器 G_B では，前半楽節との遷移を学習するために特徴量を抽出する．このために，前半楽節生成器 G_A の前段に特徴量抽出のための層を追加した構造を用いる．学習時には学習データの前半楽節 $\mathbf{X}_A$ を入力として，後半楽節 $\tilde{\mathbf{X}}_B$ を出力する：$\tilde{\mathbf{X}}_B = G_B(\mathbf{X}_A)$. 判別器 D_B および生成器 G_B の学習には，前半楽節生成 GAN と同様の次式を用いる：

$$\begin{aligned}\min_{G_B}\max_{D_B}&\mathbb{E}_{\tilde{\mathbf{X}}_B\sim p_g(\tilde{\mathbf{X}}_B)}[D_B(\tilde{\mathbf{X}}_B)]\\&-\mathbb{E}_{\mathbf{X}_B\sim p_B(\mathbf{X}_B)}[D_B(\mathbf{X}_B)]\\&+\lambda\mathbb{E}_{\hat{\mathbf{X}}_B\sim p_{\hat{\mathbf{X}}_B}(\hat{\mathbf{X}}_B)}[(\|\nabla_{\hat{\mathbf{X}}_B}D_B(\hat{\mathbf{X}}_B)\|_2-1)^2]\end{aligned}$$

ここで，

$$\hat{\mathbf{X}}_B = \epsilon\mathbf{X}_B + (1-\epsilon)\tilde{\mathbf{X}}_B \qquad \text{s.t.}\quad \epsilon\sim U[0,1]$$

であり，p_B は後半楽節の真分布，p_g は後半楽節の生成分布である．$\hat{\mathbf{X}}_B$ は $\mathbf{X}_B$ と $\tilde{\mathbf{X}}_B$ を結ぶ線分上の任意の点であり，ϵ はその位置を決める乱数である．

学習して得られた後半楽節生成 GAN を評価する時には，生成器への入力は前半楽節生成 GAN によって生成された小楽節 $\tilde{\mathbf{X}}_A$ となる：$\tilde{\mathbf{X}}_B = G_B(\tilde{\mathbf{X}}_A)$. こうして得られた 2 小楽節 $\tilde{\mathbf{X}}_A$ と $\tilde{\mathbf{X}}_B$ を連結したものが，提案手法が生成する大楽節となる．

3　シミュレーション

3.1　学習データ

本実験では，学習データセットとして Midi collection [8] を使用する．本データセット中の四拍子のポップス 48 曲から，前半楽節と後半楽節のペアを 4 種ずつ取り出し学習データに用いる．さらに，各曲のキーを変えることにより学習データ数を 12 倍にし，計 2,304 種の大楽節が本実験における学習データ数となる．取り扱う MIDI ファイルは General MIDI Level1 (GM1) の規格 [9] に従う．GM1 では 128 種の音色と対応するキーレンジ (発音可能な音域) を切り替えることができるように音色番号 (プログラムチェンジ) の値が規定されている．本実験では，学習の効率化を目的に，使用する音色番号の統一を行う．奏法や音色，キーレンジを考慮し，類似するものを Table 1 のように纏める．これに加え，楽曲の主旋律を担当しているメロディトラックとドラムトラックを合わせた計 9 トラックが，本実験で使用するトラック数 M となる．また，一小節あたりのタイムステップ数 T を 16，キーレンジ S を MIDI ノートナンバー 23 から 106 までの 84 とする．よって，学習に使用する 1 小節あたりのマルチトラックピアノロールは $\mathbf{X}\in\{0,1\}^{16\times84\times9}$ となる．

Table 1　音色番号の統一

元の音色番号	統一後の音色番号
1〜24, 47	1: Acoustic Piano
25〜32	28: Electric Guitar(clean)
33〜40	34: Electric Bass(finger)
41〜46, 49〜55	49: Strings Ensembles 1
57〜64	62: Brass Section
65〜80	81: Lead 1(square)
81〜96	81: Lead 8(sawtooth)
48, 56, 97〜104, 105〜128	削除

3.2　実験条件

本実験では，最適化アルゴリズムに Adam optimizer を使用した．また，WGAN-GP の目的関数における λ は 10 に設定している．バッチサイズは 128 とし，学習を 1000 epoch 行った．判別器が Wasserstein 距離を正確に計算するために，最初の 25 バッチのみ，判別器を 100 回ずつ更新しており，その後は 5 回ずつ更新している．なお，生成器は常に 1 回ずつ更新している．Table 2 に前半楽節生成器 G_A における各層の詳細を示す．全結合層における数値はノード数を表しており，活性化関数には ReLU 関数を使用した．転置畳込み層における各値は，左からフィルター数，フィルターサイズ，ストライド数となっており，活性化関数は，最終層にのみ tanh 関数を使用し，その他には Leaky ReLU 関数を使用した．前半楽節生成器では全層においてバッチ正規化を行っている．

Table 3 に後半楽節生成器 G_B における各層の詳細を示す．G_B は中央の下向き矢印より上までの部分と下からの部分に大きく分かれる．矢印より上の部分が，入力された前半楽節の特徴量を抽出するために追加した層であり，128 次元の特徴ベクトルを出力する．矢印から下の部分は前半楽節生成器 G_A と構造は同一である．しかし，入力がランダムベクトルではなく，特徴ベクトルを受け取って小楽節を生成するように学習している点が異なっている．活性化関数は，全結合層には ReLU 関数を，最終層には tanh 関数を使用し，その他の層では Leaky ReLU 関数を使用した．後半楽節生成器では全層においてバッチ正規化を行っている．

Table 4 に前半楽節判別器と後半楽節判別器で共通に用いているネットワーク構造の詳細を示す．全ての

Table 2　前半楽節生成器

入力: $\mathbb{R}^{128}$							
全結合層	1536						
(3,1,1) × 512 チャネルに reshape							
転置畳込み層	256	2 × 1 × 1	(1,1,1)				
転置畳込み層	128	1 × 2 × 1	(1,2,1)				
転置畳込み層	128	1 × 1 × 3	(1,1,3)				
転置畳込み層	64	1 × 2 × 1	(1,2,1)				
転置畳込み層	64	1 × 1 × 3	(1,1,2)				
	substream 1			substream 2			
転置畳込み層	64	1 × 1 × 12	(1,1,12)	64	1 × 4 × 1	(1,4,1)	
転置畳込み層	32	1 × 4 × 1	(1,4,1)	32	1 × 1 × 12	(1,1,12)	... × 9
チャネル軸方向に concat							
転置畳込み層	1	1 × 1 × 1	(1,1,1)				
チャネル軸方向に stack							
出力: $\mathbb{R}^{4\times16\times84\times9}$							

Table 3　後半楽節生成器

入力: $\mathbb{R}^{4\times16\times84\times9}$								
トラック軸方向に分割								
	substream 1			substream 2			... × 9	E_c	E_o
畳込み層	32	1 × 1 × 12	(1,1,12)	32	1 × 4 × 1	(1,4,1)			
畳込み層	64	1 × 4 × 1	(1,4,1)	64	1 × 1 × 12	(1,1,12)			
チャネル軸方向に concat									
畳込み層	64	1 × 1 × 1	(1,1,1)						
チャネル軸方向に concat									
畳込み層	128	1 × 2 × 3	(1,2,2)						
畳込み層	256	1 × 2 × 3	(1,2,3)						
チャネル軸方向に concat									
畳込み層	128	4 × 1 × 1	(1,1,1)						
全結合層	128								

⇓

全結合層	1536						
(3,1,1) × 512 チャネルに reshape							
転置畳込み層	256	2 × 1 × 1	(1,1,1)				
転置畳込み層	128	1 × 2 × 1	(1,2,1)				
転置畳込み層	128	1 × 1 × 3	(1,1,3)				
転置畳込み層	64	1 × 2 × 1	(1,2,1)				
転置畳込み層	64	1 × 1 × 3	(1,1,2)				
	substream 1			substream 2			
転置畳込み層	64	1 × 1 × 12	(1,1,12)	64	1 × 4 × 1	(1,4,1)	... × 9
転置畳込み層	32	1 × 4 × 1	(1,4,1)	32	1 × 1 × 12	(1,1,12)	
チャネル軸方向に concat							
転置畳込み層	1	1 × 1 × 1	(1,1,1)				
チャネル軸方向に stack							
出力: $\mathbb{R}^{4\times16\times84\times9}$							

層において，Leaky ReLU 関数を活性化関数として使用している．判別器ではいずれの層においてもバッチ正規化を行っていない．Table 3.2 に終止判別器 D_o [5] および終始特徴抽出器 E_o の詳細を示す．Table 3.2 に終止判別器 D_c [5] および終始特徴抽出器 E_c の詳細を示す．

3.3　主観評価

生成された大楽節に対してアンケートによる主観評価を行った．実験参加者は 20 代の健常な男女 10 人である．各参加者には，学習データに用いた大楽節，提案手法で生成された大楽節，既存手法 (Binary-MuseGAN) で生成された小楽節二つを連続再生したものを 4 曲ずつブラインドテスト形式で試聴してもらった．各ピアノロールを Roland の Hyper Canvas を用いて wav ファイルに書き出し，オーディオインターフェイスに Native Instruments の Komplete Audio 6，ヘッドホンに audio-technica の ATH-M50x を用いて再生した．評価項目は，ハーモニーの良さ，リズムの良さ，人間による作曲らしさ，小楽節接続のスムーズさ，大楽節の総合評価の 5 種である．各項目に対して，実験参加者に 5 段階評価を行ってもらい，得られた評価値の平均値と標準偏差を Table 7 に示す．どの評価項目においても，提案手法による生成曲の評価値は既存手法の評価値を上回ったが，学習データの評価値を下回る結果となった．既存手法で生成した楽曲よりも，提案手法で生成した楽曲は人間が作った楽曲に近いと推察できる．

Table 4　前半楽節判別器および後半楽節判別器

入力: $\mathbb{R}^{4\times16\times84\times9}$									
トラック軸方向に分割									
	substream 1			substream 2			... × 9	D_c	D_o
畳込み層	32	1 × 1 × 12	(1,1,12)	32	1 × 4 × 1	(1,4,1)			
畳込み層	64	1 × 4 × 1	(1,4,1)	64	1 × 1 × 12	(1,1,12)			
チャネル軸方向に concat									
畳込み層	64	1 × 1 × 1	(1,1,1)						
チャネル軸方向に concat									
畳込み層	128	1 × 2 × 3	(1,2,2)						
畳込み層	256	1 × 2 × 3	(1,2,3)						
チャネル軸方向に concat									
畳込み層	512	2 × 1 × 1	(1,1,1)						
全結合層	1536								
全結合層	1								
出力: $\mathbb{R}$									

Table 5　D_o および E_o

入力: $\mathbb{R}^{4\times16\times1\times9}$				
畳込み層	32	1 × 4 × 1	(1,4,1)	
畳込み層	64	1 × 2 × 1	(1,2,1)	
畳込み層	128	1 × 2 × 1	(1,2,1)	
出力: $\mathbb{R}^{4\times1\times1\times128}$				

Table 6　D_c および E_c

入力: $\mathbb{R}^{4\times4\times12\times9}$			
畳込み層	64	1 × 1 × 12	(1,1,12)
畳込み層	128	1 × 4 × 1	(1,4,1)
出力: $\mathbb{R}^{4\times1\times1\times128}$			

Table 7　アンケート結果

評価項目＼曲種	学習データ	提案手法	既存手法
ハーモニーの良さ	3.83±0.97	3.1±0.94	2.25±0.83
リズムの良さ	4.10±0.77	3.63±0.80	3.08±0.84
人間による作曲らしさ	3.98±0.88	3.00±0.77	2.43±0.77
接続のスムーズさ	4.1±0.77	3.48±0.81	2.90±0.80
総合評価	3.98±0.82	3.25±0.73	2.73±0.71

(5 段階評価: 1:悪い 2:やや悪い 3:普通 4: やや良い 5:良い)

3.4　客観評価

人間に作曲された楽曲と提案手法による自動生成曲の客観評価を行った．評価指標には，先行研究 [3] で使用されている 3 種類の指標 Empty Bar Rate, Qualified Note Rate，Polyphonicity Rate を用いる．Empty Bar Rate は，あるトラックにおける全てのタイムステップ数に対する無音のタイムステップ数の比率である．この値が小さいと，そのトラックの楽曲中の出現頻度が高いことを示す．Qualified Note Rate は，あるトラックにおける全ての音符に対する 8 分音符以上の長さを持つ音符の比率である．この値が小さいと，そのトラックを構成する音符が細かいことを示す．Polyphonicity Rate は，あるトラックにおける全てのタイムステップ数に対する複数音を有するタイムステップ数の比率である．この値が小さいと，そのトラックを構成する音に単音が多いことを示す．学習データにおいて，最も Empty Bar Rate が小さな 5 トラックの結果をまとめる．なお，ドラムトラックは音高情報を有さないため，Empty Bar Rate のみ算出した．また，Table 8 に学習データと提案手法の前半楽節の評価を，Table 9 に学習データと提案手法

Table 8　前半楽節の客観評価

評価指標 / トラック	Empty Bar [%]		Qualified Note [%]		Poliphonicity [%]	
	学習データ	提案手法	学習データ	提案手法	学習データ	提案手法
ピアノ	25.13	24.88	70.20	57.47	64.29	46.60
ギター	48.18	47.50	58.11	53.54	49.84	32.42
ベース	18.01	31.12	58.18	47.69	2.46	3.79
メロディ	3.78	12.03	61.70	46.38	0.58	7.67
ドラム	5.46	2.34	-	-	-	-

Table 9　後半楽節の客観評価

評価指標 / トラック	Empty Bar [%]		Qualified Note [%]		Poliphonicity [%]	
	学習データ	提案手法	学習データ	提案手法	学習データ	提案手法
ピアノ	25.92	26.13	71.03	57.42	64.81	41.60
ギター	46.74	57.50	60.69	48.69	49.62	20.98
ベース	16.82	28.05	57.41	50.63	2.80	4.44
メロディ	4.95	15.47	63.17	40.03	0.86	5.81
ドラム	4.56	3.59	-	-	-	-

の後半楽節の評価を示す．

提案手法では，全トラックにおいて Qualified Note Rate が小さくなっている．つまり，16 分音符の割合が増えていることになる．ピアノトラックとギタートラックにおいては Polyphonicity が約 20%減少している．このことは，学習データの楽曲では和音演奏が多い傾向に対して，提案手法では単音演奏が増えていることを示している．ベーストラックとメロディトラックに着目すると，Empty Bar Rate と Polyphonicity Rate の上昇が大きく，学習データとの差が見られる．一般的にベースは楽曲の低音域を担当し，ボーカルは楽曲の主旋律を担当するため，楽曲中で絶えず演奏することが多い，よって，Empty Bar Rate の上昇は望ましくない．また両トラックは単音で演奏されることが多いと考えられるので，Polyphonicity Rate の上昇も望ましくない．ドラムトラックに着目すると，Empty Bar Rate が減少していることが分かる．音が過密していき，単調なリズムになることが懸念される．

4　おわりに

本研究では，二種類の GAN を用いて大楽節を生成する手法について提案した．提案手法では，一方の GAN がランダムノイズから前半楽節となる小楽節を生成し，他方の GAN が前半楽節から後半楽節への遷移を学習して後半楽節を生成する．これら 2 種類の GAN を組み合わせることで，小楽節単位の学習のみで大楽節を生成することができる．実験の主観評価では，アンケート調査を行い，生成した楽曲とその他の楽曲をブラインドテスト形式で評価した．その結果，提案手法による自動生成曲の主観評価が，従来法による小楽節を単純に接続した場合よりも高い値となった．一方，客観評価では，各トラックにおける音の細かさや和音の多さを学習データと比較した．その結果，提案手法による自動生成曲と学習データの楽曲では異なる傾向がみられた．提案手法によって生成した楽曲は音楽的である一方，人間に作曲された楽曲と雰囲気の異なる楽曲であると推察できる．

参考文献

[1] O. Lopez-Rincon, O. Starostenko, and G. A. Martin, “Algorithmic music composition based on artificial intelligence: A survey,” in 2018 International Conference on Electronics, Communications and Computers, pp.187–193, Cholula Puebla, 2018.

[2] 東山恵祐, “作曲支援システムにおけるコード進行及びキーの決定方法,” 情報処理学会研究報告, vol. 2014-MUS-103, no. 52, pp. 1-6, 2014.

[3] 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 Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI), pp.34-41, Louisiana, 2018.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y Bengio, “Generative adversarial nets,” Advances in Neural Information Processing Systems 29 (NIPS 2016), pp.4790-4798, Barcelona, 2016.

[5] H.-W. Dong, and Y.-H. Yang, “Convolutional generative adversarial networks with binary neurons for polyphonic music generation,” in The 19th International Society for Music Information Retrieval Conference (ISMIR), pp.190-196, Paris, 2018.

[6] 石桁真礼生, “新版　楽式論,” 音楽之友社, 東京, 2017.

[7] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of Wasserstein GAN,” Advances in Neural Information Processing Systems 30 (NIPS 2017), pp.5767–5777, California, 2017.

[8] Reddit midi_man, “Midi collection,” https://goo.gl/4moEZ3.

[9] 音楽電子事業協会, “MIDI 1.0 規格書 MIDI Standard & Recommended Practice(日本語版 98.1),” http://amei.or.jp/midistandardcommittee/MIDI1.0.pdf, 2016.