

平成 31 年度 修士論文

深層学習を用いた楽曲ジャンル分類における  
ジャンル境界の可視化

Visualization of Boundaries in Music Genre Classification

by using Deep Learning

所属	新潟大学 自然科学研究科
在籍番号	F18C039J
氏名	山川 颯人

主指導教員：

林 隆史 教授

副指導教官：

中野 啓介 教授

元木 達也 教授

## 概要

近年ではインターネットが普及し、デジタルミュージックコンテンツが劇的に増加している。効率の良い楽曲検索システムを実現するためにも、楽曲ジャンルを自動分類するモデルが必要不可欠である。しかしながら、楽曲ジャンル間を分類する基準は不明瞭であるため、モデル設計が困難となっている。

そこで本研究では、分類基準の特徴を自動で抽出する深層学習を用いてモデルを設計し、楽曲ジャンル分類を行う。さらに学習した分類モデルから、楽曲ジャンル間の境界を可視化することを目的とする。

提案手法において、畳み込みニューラルネットワーク (Convolutional Neural Network : CNN) による楽曲ジャンルの分類を行った結果、77.4%の分類精度を確認した。さらに学習済みの CNN に対し、敵対生成ネットワーク (Generative Adversarial Network : GAN) を用いることで、CNN がジャンル分類するための入力データを生成した。最後に GAN の入力ノイズベクトルと生成データの分類結果の関係を表す 2 次元ジャンルマップとして可視化を行い、ジャンル確率の連続変化を確認できた。

## Abstract

In recent years, the Internet has spread and digital music content has increased dramatically. In order to realize an efficient music search system, the model classifies music genres automatically is essential. However, designing a classification model is difficult because the criteria for classifying music genres is unclear.

Therefore, in this study, we design a model using deep learning that automatically extracts features of classification criteria, and perform music genre classification. Then we visualize the boundaries among music genres from the learned classification model.

In our proposed method, we classify the music genre by Convolutional Neural Network (CNN). As a result, we confirmed 77.4 % classification accuracy. Then we generate data which is classified by the learned CNN by using Generative Adversarial Network (GAN). Finally, we visualize the relationship between the GAN input noise vector and the classification result of the generated data as a Two-Dimensional Genre Map.

# 目 次

第 1 章 はじめに	1
第 2 章 ニューラルネットを用いた機械学習	3
2.1 深層学習 . . . . .	3
2.2 損失関数と重み最適化 . . . . .	4
2.2.1 損失関数 . . . . .	4
2.2.2 重み最適化 . . . . .	4
2.2.3 Adam . . . . .	5
2.3 疊み込みニューラルネットワーク . . . . .	6
2.3.1 疊み込み層 . . . . .	7
2.3.2 プーリング層 . . . . .	7
2.3.3 全結合層と活性化関数 . . . . .	8
2.4 敵対生成ネットワーク . . . . .	9
2.4.1 GAN の損失関数 . . . . .	10
2.4.2 GAN の最適解 . . . . .	10
2.4.3 Wasserstein GAN . . . . .	11
2.5 ニューラルネットによる相互情報量の推定と最大化 . . . . .	13
2.5.1 相互情報量 . . . . .	13
2.5.2 Mutual Information Neural Estimation . . . . .	13
2.5.3 Mutual information estimation and maximization . . . . .	14
第 3 章 メル周波数スペクトログラム	15
3.1 短時間フーリエ変換 . . . . .	15
3.2 メル周波数 . . . . .	15

## 第4章 関連研究 16

4.1 Convolutional Neural Network Achieves Human-level Accuracy in Music Genre Classification . . . . .	16
4.1.1 学習データセット作成 . . . . .	16
4.1.2 ネットワーク構成 . . . . .	16
4.1.3 各ジャンルにおける分類精度 . . . . .	18
4.1.4 問題点 . . . . .	18
4.2 Grad-CAM . . . . .	19
4.2.1 入力画像のヒートマップ化 . . . . .	19
4.2.2 問題点 . . . . .	21
4.3 Harmonic/Percussive Separation using Median Filtering . . . . .	22
4.3.1 メディアンフィルタ . . . . .	22
4.3.2 ハーモニー成分とパーカッショングループ . . . . .	23
4.3.3 ハーモニー成分とパーカッショングループの分離 . . . . .	24
4.3.4 メリット . . . . .	25

## 第5章 提案手法 26

5.1 データセットの作成 . . . . .	26
5.1.1 GTZAN データセットの修正 . . . . .	26
5.1.2 学習用データセット作成 . . . . .	26
5.2 ジャンル分類器の構築 . . . . .	30
5.2.1 ジャンル分類器 . . . . .	30
5.3 データ生成器の構築 . . . . .	31
5.3.1 生成器モデル . . . . .	32
5.3.2 ランダムノイズと分類結果の従属性 . . . . .	34
5.4 ジャンル境界の可視化 . . . . .	35

## 第6章 実験と検証 37

6.1 分類モデルの評価 . . . . .	37
------------------------	----

6.1.1	学習毎の損失と精度のグラフ	37
6.1.2	1曲分の分類精度	43
6.2	生成器モデルの評価	50
6.2.1	学習毎の損失	50
6.2.2	生成されるスペクトログラム	51
6.2.3	生成スペクトログラムのジャンル確率分布	56
6.3	2次元ジャンルマップの検証	57
第7章 おわりに		60
謝辞		61
参考文献		62
付録A プログラムのソースコード		64
付録B Grad-CAMによる予備実験		65
付録C MNISTを用いた予備実験		66

# 第1章 はじめに

インターネットの普及に伴い、様々なマルチメディアデバイスが増加する中、音楽コンテンツのデジタル化も劇的に増加している。ユーザの嗜好に合わせたより良い楽曲を提示するためにも、自動で楽曲を特徴づけるアルゴリズムないしは方法・枠組みといったものが必要不可欠である。この課題は音楽情報検索 (Musical Information Retrieval : MIR) という研究テーマとして様々な事例が挙げられており、その中の一つとして楽曲ジャンル分類がある。楽曲ジャンルの分類基準は一般に曖昧かつ不明瞭であるため、明確なアルゴリズムで自動的に分類することはとても困難である。

このような背景の下、近年では機械学習を用いて楽曲ジャンルを分類を行う研究が注目されている。機械学習を用いたモデルは、分類する基準を統計的に最適化していくため、明確なアルゴリズムを作ることが困難であるパターン認識において優れた手法となっている。さらに機械学習の一種である深層学習においては、データの特徴量を自動で抽出するという特長を有し、非線形な分類問題にも対応できることから、複雑な問題の近似解を求めるための手法として様々な分野で活用されている。こうした理由から、深層学習を楽曲ジャンル分類に適用した研究も行われており、一例として Mingwen らによる畳み込みニューラルネットワークを用いたものが挙げられる [1]。

深層学習はエンドツーエンドな学習でモデル設計が比較的簡単な一方、学習した中身がブラックボックスであり、判断根拠が不明という欠点がある。楽曲ジャンル分類を例にとるならば、どういった音色や音の大きさがジャンル分類に寄与しているかがわからないことになる。これは誤った判別をした際にどのようにモデルを修正すればよいかといった問題が挙げられる。そのため、人間に解釈可能な表現をあてはめることによってモデルの信頼性を高めるといったことも重要であり、深層学習の判断根拠を定量化する研究も行われている。その中でも、分類時の判断根拠となるデータ箇所をヒートマップ化をして可視化を行うという観点で、Grad-CAM という手法がある [2]。しかしながら、この方法は学習済みモデルにおける

る特徴マップの出力サイズに大きく依存してしまい、モデル設計の仕方によっては意味のないヒートマップを作ってしまうことがある。

以上の点を踏まえて本研究では、深層学習を用いることによって、楽曲ジャンルという分類基準が不明瞭なデータを自動分類し、学習した分類モデルからジャンル境界を可視化する新規手法を提案することを目的とする。分類モデルはパターン認識に優れた Convolutional Neural Network (CNN) を用いることによって分類精度の向上図る。また、CNN で得られた学習済み分類モデルと Generative Adversarial Network (GAN) を組み合わせることによって、CNN が分類するためのデータを生成する。これにより GAN への入力ノイズベクトルと生成データの分類結果の関係を 2 次元マップとして可視化をする。

第 2 章では機械学習の一つであるニューラルネットを用いた学習法の全体概要を説明した後、分類モデルでよく用いられる CNN と、生成モデルで用いられる GAN について細かく説明する。さらにニューラルネットを用いた相互情報量の推定と最大化をする手法についても説明する。第 3 章ではモデルの入力として用いるメル周波数スペクトログラムについて説明する。第 4 章では CNN を用いた楽曲ジャンル分類の関連研究として [1] と可視化手法である [2] の手法を説明し、それぞれの問題点を述べる。また、周波数スペクトログラムをハーモニー成分とパーカッション成分に分ける手法を説明する。第 5 章では提案手法について述べる。はじめに学習用データセットについて述べ、CNN を用いて楽曲ジャンルを分類するモデルを構築する。次に学習済み CNN と GAN を用いて、CNN が分類するための入力データを生成するモデルを構築する。最後に生成モデルを用いて 2 次元ジャンルマップの可視化を行う手法を提案する。第 6 章では、提案手法における分類モデルと生成モデルの評価を行い、作成した 2 次元ジャンルマップに対し考察を行う。

# 第2章 ニューラルネットを用いた機械学習

本章では機械学習の中の一つであるニューラルネットについて基本的なことから説明し、次に CNN, GAN, MINE の仕組みについて詳しく説明する。

## 2.1 深層学習

深層学習は、図 2.1 に示すような生物の脳の神経細胞をモデル化したニューロンを基にして、図 2.2 のようにニューロンを多層に結合したモデル（ニューラルネットワーク）を用いる学習法である。個々のニューロン間の結合には重み  $w$  というパラメータが与えられており、 $w$  が更新されていく（学習する）ことで、問題にあった最適解を導く。この重み  $w$  の学習法として、誤差逆伝播法を用いる。誤差逆伝播法は教師信号値  $t$  と出力結果値  $h(x)$  の誤差の大きさを表す損失関数  $E$  を定義し、 $E$  に対し各層の  $w$  の微分係数（勾配）を求めることで  $w$  を更新していく。

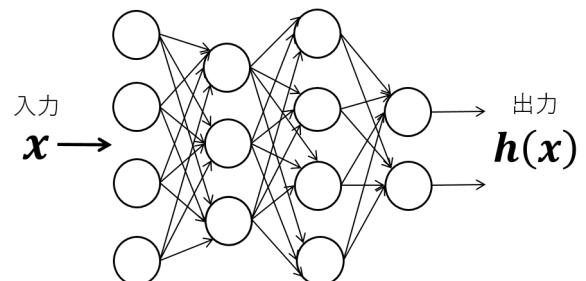
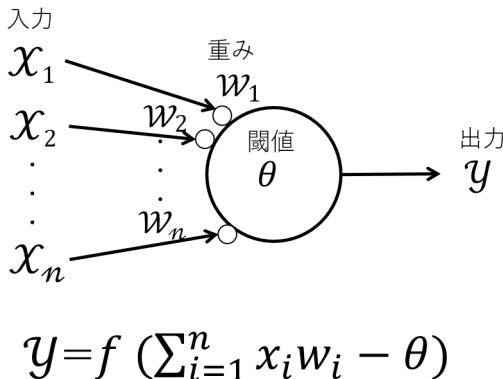


図 2.2 多層パーセプトロン

図 2.1 ニューロンモデル

## 2.2 損失関数と重み最適化

### 2.2.1 損失関数

損失関数  $E$  は様々な種類があり、一般的によく使われるものとして式 (2.1)～式 (2.3) の、交差エントロピー誤差、平均 2 乗誤差、平均絶対誤差などがある。

$$E = - \sum_k^n t_k \log(h_k(\mathbf{x})) \quad (2.1)$$

$$E = \frac{1}{n} \sum_k^n (t_k - h_k(\mathbf{x}))^2 \quad (2.2)$$

$$E = \frac{1}{n} \sum_k^n |t_k - h_k(\mathbf{x})| \quad (2.3)$$

### 2.2.2 重み最適化

$\mathbf{w}$  を更新する手法は様々あるが、基本となっているものは式 (2.4) の勾配降下法であり、損失関数の勾配が減少する方向に学習率  $\eta$  を乗算した値を加えていくことで  $\mathbf{w}$  の最適値を見つけていく。

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{\partial E}{\partial \mathbf{w}} \quad (2.4)$$

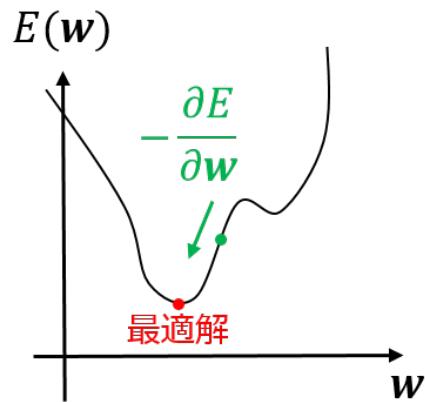


図 2.3 勾配降下法

### 2.2.3 Adam

Adam は式 (2.4) を派生させた  $\mathbf{w}$  の更新方法で現在よく用いられている最適化アルゴリズムである [3]. 2015 年に Diederik P. Kingma らが提唱した手法であり, 式 (2.5) のように学習ステップごとに過去の勾配の値から勾配の重みつき平均と重みつき分散を推定している. これにより, 更新が多い重みの学習率を低く, 更新が少ない重みの学習率を高くするように設定され, 学習の収束が早くなることが期待できる. 式 (2.6), 式 (2.7) における  $\beta_1, \beta_2$  はハイパーパラメータを表し, 実装する側が指定する値である.

$$\begin{aligned} \mathbf{w}_{t+1} &= \mathbf{w}_t - \eta \frac{\hat{\mathbf{m}}}{\sqrt{\hat{\mathbf{v}}} + \epsilon} \\ \hat{\mathbf{m}} &= \frac{\mathbf{m}_{t+1}}{1 - \beta_1^t} \\ \hat{\mathbf{v}} &= \frac{\mathbf{v}_{t+1}}{1 - \beta_2^t} \end{aligned} \tag{2.5}$$

$$\mathbf{m}_{t+1} = \beta_1 \mathbf{m}_t + (1 - \beta_1) \frac{\partial E}{\partial \mathbf{w}_t} = (1 - \beta_1) \sum_{i=1}^t \beta_1^{t-i} \mathbf{m}_i \tag{2.6}$$

$$\mathbf{v}_{t+1} = \beta_2 \mathbf{v}_t + (1 - \beta_2) \left( \frac{\partial E}{\partial \mathbf{w}_t} \right)^2 = (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} \mathbf{v}_i \tag{2.7}$$

## 2.3 畳み込みニューラルネットワーク

畳み込みニューラルネットワーク (CNN : Convolutional Neural Network) は、人間の視覚野の神経細胞の二つの働きである「画像の濃淡パターンを検出する (特徴抽出)」、及び「物体の位置が変動しても同一の物体であるとみなす (位置ズレの考慮)」を組み合わせたものとなっており、画像分野において高い評価を持つニューラルネットワークとなっている [4]。また、入力データのパターンをうまく学習できるという点で、画像だけでなく様々な問題設定で CNN が広く用いられ、高い精度を出している。

図 2.4 は画像分類問題を例にとった CNN のモデルを示している。入力画像は特徴抽出部で特徴が抽出され、その特徴をもとに識別部でパターン分類を行う。特徴抽出部では数層の畳み込み層とプーリング層から構成され、識別部は全結合層から構成されている。

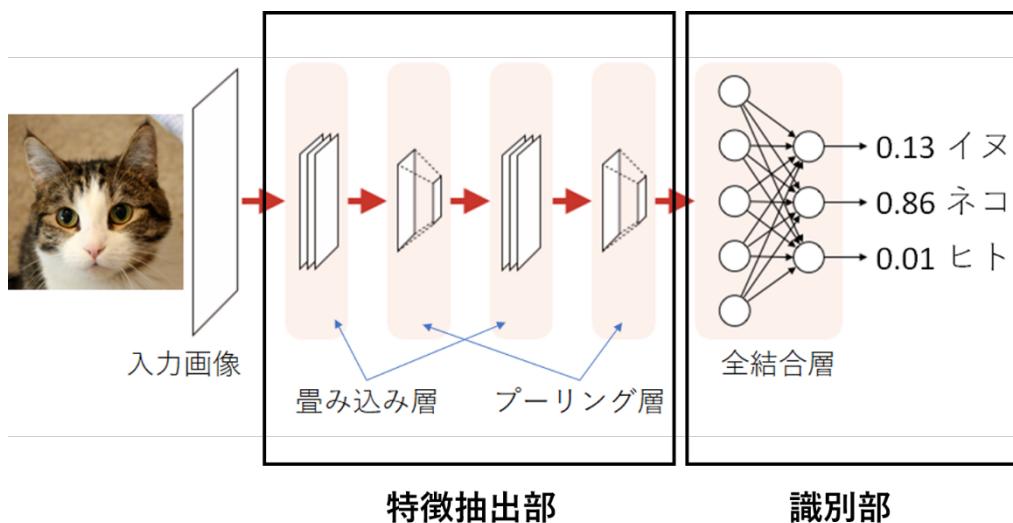


図 2.4 畳み込みニューラルネットワーク

### 2.3.1 疊み込み層

疊み込み層は画像の濃淡パターンを検出するための層に相当する。図 2.5 に示すように、入力画像に対し各ピクセル値にフィルタを適用し、フィルタをスライドさせながら画像を圧縮し、特徴マップを作成する。学習時にはフィルタの値が更新される。

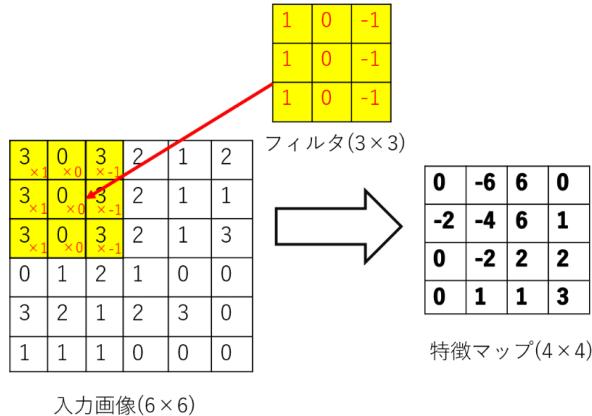


図 2.5 疊み込み

### 2.3.2 プーリング層

プーリング層は位置に対する感度を低くする代わりに、位置変化に対する認識能力を上げるための層に相当する。疊み込み層で得られた特徴マップに対しさらに圧縮をかけることで位置ズレの変化に対応する仕組みとなっている。圧縮のかけ方としては、最大プーリングと平均プーリングがある。図 2.6 に示すのは最大プーリング (max pooling) であり、特徴マップの小領域の中から最大のピクセル値を得る操作となっている。対して平均プーリング (average pooling) は小領域の中の値を平均した値を得る操作となる。プーリング層においては学習時に更新されるパラメータは存在しない。

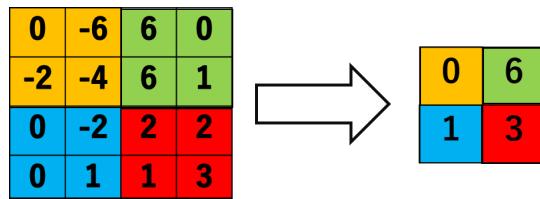


図 2.6 最大プーリング

### 2.3.3 全結合層と活性化関数

全結合層は、特徴抽出部から得られた特徴マップの値を入力とし、図 2.2 のような多層パーセプトロンのニューラルネットワークによりパターン分類が行われる層である。最後のニューロンの出力数は問題設定に応じて変更される。分類問題の場合は分類したいクラス数であったり、画像生成問題の場合は画像のピクセルサイズ分の出力を持ったりなどをする。

各層のニューロンの出力値は活性化関数が通された後の値となっている。これは、ニューロンの出力が線形であるため、多層パーセプトロンと等価な 1 層のパーセプトロンの対が必ず存在してしまうことを避けるためである。活性化関数には、ランプ関数 (ReLU), ソフトマックス関数、シグモイド関数などといったものが存在する。

#### Relu 関数

中間層のニューロンの出力を非線形にするためによく使用されている関数である。非負の値を出力する。

$$f(x) = \max(0, x) \quad (2.8)$$

#### ソフトマックス関数

ニューロンの出力を確率として扱う場合に使用される関数である。分類問題において最終層のニューロンの活性化関数として使用され、式 (2.9) で表される。 $a_k$  は  $k$  番目のニューロンの出力値であり、 $n$  は最終層のニューロンの出力数である。

$$f_k(a_k) = \frac{e^{a_k}}{\sum_{i=1}^n e^{a_i}} \quad (2.9)$$

#### シグモイド関数

出力値を 0~1 に収める関数である。ニューロンの出力値を非線形したいときや、確率とみなしてマルチクラス分類をする際にも使用されている。

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.10)$$

## 2.4 敵対生成ネットワーク

敵対生成ネットワーク (GAN : Generative Adversarial Network) は、学習データと似たような新しいデータを生成する生成モデルの一種であり、言い換えれば生成データの分布を学習データの分布に近づけていくように学習するモデルである。GAN の構造を図 2.7 に示す。GAN は図 2.7 に示すように、生成器 (Generator) と識別器 (Discriminator) から構成され、Generator はランダムノイズからオリジナルと似たデータを生成し、Discriminator は入力されるデータが Generator によって作られたデータか、それともオリジナルのデータかの識別を行う。これら二つのモデルが互いを見抜く・騙すように学習するため、十分に学習が進むと Generator はオリジナルのデータと見分けがつかないようなデータを生成するようになる。

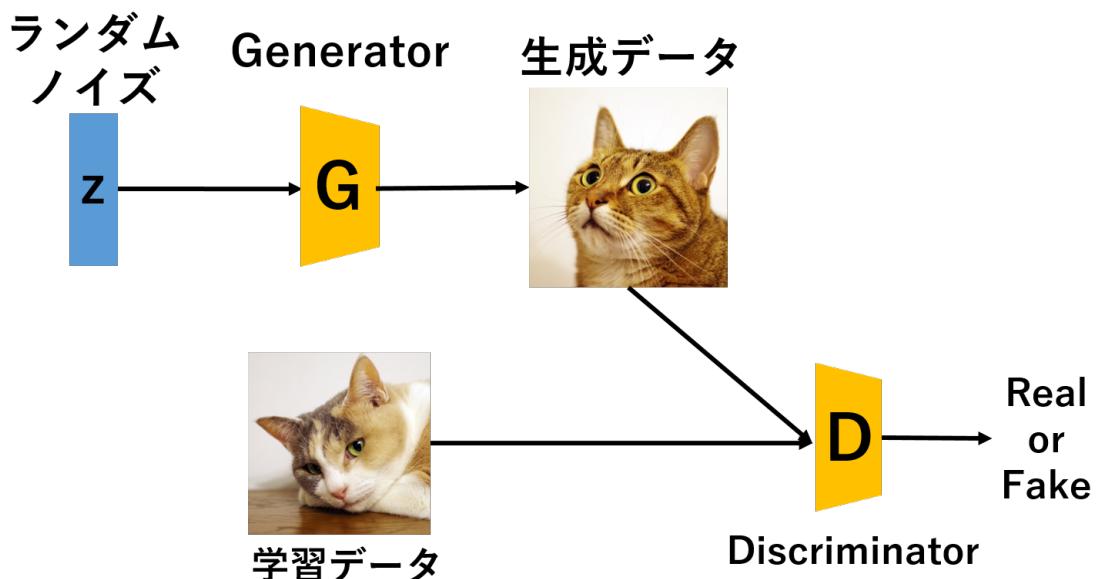


図 2.7 敵対生成ネットワーク (GAN)

### 2.4.1 GAN の損失関数

具体的に評価関数を導入して学習を行う場合、式 (2.11) に関して、Discriminator に対して最大化、Generator に対して最小化するミニマックスゲームを考えればよい [5].

$$\min_G \max_D V(G, D) = \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [1 - \log D(G(\mathbf{z}))] \quad (2.11)$$

式 (2.11) において、 $\mathbf{x}$  はオリジナルデータ、 $p_r(\mathbf{x})$  はオリジナルデータの確率分布、 $\mathbf{z}$  はランダムノイズ、 $p_z(\mathbf{z})$  はランダムノイズの確率分布を示す。この時、 $D$  がオリジナルのデータを正しく判定できれば  $\log D(\mathbf{x})$  が大きくなり、 $D$  が  $G$  の生成データをオリジナルのデータと誤って判定すると  $\log(1 - D(G(\mathbf{z})))$  が小さくなる。

### 2.4.2 GAN の最適解

式 (2.11) において、 $\mathbf{z} \sim p_z(\mathbf{z})$  から  $G$  が生成するデータの分布を  $p_g(\mathbf{x})$  とし、 $V(G, D)$  を書き直すと、

$$V(G, D) = \int_{\mathbf{x}} \{p_r(\mathbf{x}) \log D(\mathbf{x}) + p_g(\mathbf{x}) \log(1 - D(\mathbf{x}))\} d\mathbf{x} \quad (2.12)$$

となる。Discriminator の最適解は  $V(G, D)$  を最大化することであるため、積分の中身の最大化をすればよい。よって中身を  $D(\mathbf{x})$  に関して微分し、その時の導関数が 0 になる  $D(\mathbf{x})$  を  $D^*(\mathbf{x})$  とすると、式 (2.13) となる。

$$D^*(\mathbf{x}) = \frac{p_r(\mathbf{x})}{p_r(\mathbf{x}) + p_g(\mathbf{x})} \quad (2.13)$$

またこのとき、Generator の最適化を考える。式 (2.13) の右辺を式 (2.12) に代入し式を整理すると、式 (2.14) となる。

$$V(G, D) = 2D_{JS}(p_r || p_g) - \log 4 \quad (2.14)$$

式 (2.14) を最小化することは、Jensen-Shannon ダイバージェンスの最適化と等しく、 $p_g = p_r$  のときに最小値  $-\log 4$  をとる。

### 2.4.3 Wasserstein GAN

二つの確率分布の距離として定義した Jensen-Shannon ダイバージェンスを損失関数としたとき、関数に不連続な箇所が見受けられ、勾配が求められないという問題があった [6]。そこで、関数に連続性を持たせるために新たに Wasserstein 距離を二つの確率密度関数の距離を測る指標として導入し、それを用いた GAN を Wasserstein GAN(WGAN) と言う。

#### Wasserstein 距離

Wasserstein 距離は、式 (2.15) で与えられる

$$W(p_r, p_g) = \inf_{\gamma \sim \Pi(p_r, p_g)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} [\|\mathbf{x} - \mathbf{y}\|] \quad (2.15)$$

$\Pi(p_r, p_g)$  は  $p_r$  と  $p_g$  の同時分布を示し、 $\gamma(\mathbf{x}, \mathbf{y})$  は  $p_r$  のある地点  $\mathbf{x}$  を  $p_g$  のある地点  $\mathbf{y}$  に移動させる量である。その量にノルム  $\|\mathbf{x} - \mathbf{y}\|$  をかけたものをコストとして定義し、コストを最小にしたもののが Wasserstein 距離である。この関数はどの点においても連続になるため、勾配がすべての点で存在する。また式 (2.15) は Kantrovich-Rubinstein 双対性を用いて、式 (2.16) に変形できる。

$$\begin{aligned} W(p_r, p_g) &= \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{\mathbf{x} \sim p_r} [f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_g} [f(\mathbf{x})] \\ &= \frac{1}{K} \sup_{\|D\|_L \leq K} \mathbb{E}_{\mathbf{x} \sim p_r} [D(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim p_z} [D(G(\mathbf{z}))] \end{aligned} \quad (2.16)$$

#### 損失関数

式 (2.16) から Wasserstein 距離を求めるためには最大化問題を解かなければならない。Wasserstein GAN ではこの問題を Discriminator が担い、より正確な Wasserstein 距離を求めようとする。対して Generator は Discriminator で求めた Wasserstein 距離を最小化するように学習を行う。つまり、Discriminator 損失関数は式 (2.16) である。一方 Generator の損失関数は式 (2.16) を Generator が持つ学習パラメータ  $\theta$  で微分した式 (2.17) とすることで、 $W(p_r, p_g)$  を最小化すればよいことがわかる。式 (2.17) における  $M$  はバッチサイズを示す。

$$\begin{aligned}\frac{\partial W(p_r, p_g)}{\partial \theta} &= -\mathbb{E}_{z \sim p_z} \left[ \frac{\partial D(G(z))}{\partial \theta} \right] \\ &\simeq -\frac{1}{M} \sum_{m=1}^M \frac{\partial}{\partial \theta} D(G(z_m))\end{aligned}\tag{2.17}$$

## Gradient Penalty

WGAN には式 (2.16) からわかるように Discriminator の制約条件として  $K$ -リップシツツ連続の関数であることが前提である。この制約から [6] では Discriminator の学習パラメータの値が  $[-c, c]$  ( $c$  は任意値) になるようにクリッピングを行っている。しかしクリッピングでは勾配が爆発したり消失したりするのに加え、学習の収束が遅いという欠点があった。そこで、Ishaan らによる勾配に制約を設けた WGAN-gp が提案されている [7].

WGAN-gp では、「最適化された WGAN の Discriminator は  $p_r, p_g$  下のほぼすべての点において大きさ 1 の勾配を持つ」という性質を利用して、Discriminator の損失関数の項に、

$$\lambda \mathbb{E}_{\hat{x} \sim p_{\hat{x}}} [(|\frac{\partial D(\hat{x})}{\partial \hat{x}}| - 1)^2]$$

を加えたものを新たに損失関数として定義する。ここで、 $\lambda$  はハイパーパラメータであり、 $\hat{x}$  は

$$\hat{x} = \epsilon x + (1 - \epsilon) \tilde{x}$$

$$\epsilon \sim U[0, 1], \quad x \sim p_r, \quad \tilde{x} \sim p_g$$

で表され、 $U[0, 1]$  は 0~1 の一様分布に従う乱数を示す。

## 2.5 ニューラルネットによる相互情報量の推定と最大化

### 2.5.1 相互情報量

相互情報量は二つの確率変数を測る尺度であり、二つの確率変数を  $X, Y$  とすると 式(2.18)で定義される。

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ H(X) &= -\mathbb{E}_{P(X)}[\log P(x)] \\ H(X|Y) &= -\mathbb{E}_{P(X),P(Y)}[\log P(X|Y)] \end{aligned} \quad (2.18)$$

$H$  は情報量エントロピーを示し、 $H(X|Y)$  は条件付エントロピーを表す。さらに式(2.18)を変形していくと、

$$\begin{aligned} I(X;Y) &= \mathbb{E}_{P(X),P(Y)}[\log P(X|Y)] - \mathbb{E}_{P(X)}[\log P(X)] \\ &= \mathbb{E}_{P(X),P(Y)}[\log P(X|Y)P(X)] \\ &= \mathbb{E}_{P(X),P(Y)}[\log P(X,Y)P(X)P(Y)] \\ &= D_{KL}(P(X,Y)||P(X)P(Y)) \\ &= \sup_{T:\Omega \rightarrow \mathbb{R}} \mathbb{E}_{P(X,Y)}[T] - \log(\mathbb{E}_{P(X),P(Y)}[e^T]) \end{aligned} \quad (2.19)$$

と表せる。

### 2.5.2 Mutual Information Neural Estimation

相互情報量は二つの確率変数間の依存関係を測る指標であるが、一般的に計算するのが難しい。そこでニューラルネット(NN)を用いて相互情報量を推定する方法が Mutual Information Neural Estimation(MINE)である[8]。

式(2.19)の関数  $T$  をパラメータ  $\theta$  を持つ NN で表現された関数  $T_\theta$  と考える。このとき、相互情報量は、式(2.20)となる

$$I(X;Y) \geq I_\Theta(X;Y) = \sup_{\theta \in \Theta} \mathbb{E}_{P(X,Y)}[T_\theta] - \log(\mathbb{E}_{P(X),P(Y)}[e^{T_\theta}]) \quad (2.20)$$

式 (2.20) における期待値は  $P(X, Y), P(X)P(Y)$  からのサンプリングを用いて計算され, 上限を求める際は勾配法による最大化を行う. NN は表現力に優れているため, 任意の精度で相互情報量を近似することが可能となる. プログラムで実装する場合は式 (2.20) の最大化問題を式全体にマイナスを掛けることにより最小値問題に置き換え, 下限を求めるために勾配降下法を用いる.

### 2.5.3 Mutual information estimation and maximization

MINE の枠組みに従って得られる相互情報量を最大化し, 確率変数  $X, Y$  に従属関係を持たせる場合を考える.  $Y$  の確率変数をパラメータ  $\omega$  に従う NN,  $Y_\omega = F_\omega(X)$  から得られる確率変数として置き換える. このとき確率変数  $X, Y_\omega$  の相互情報量を最大化するためには, 式 (2.21) を解くことに等しい.

$$\operatorname{argmax}_{\theta, \omega} I_\theta(X; F_\omega(X)) \quad (2.21)$$

ここで, 相互情報量の推定と最大化に NN を用いて最適化を行っていることから, 目的関数を一つで表し両者の最適化を同時に行う.  $f_\omega, C_\omega, D_\theta$  を任意の NN とし,  $F_\omega = f_\omega \circ C_\omega$ ,  $T_{\theta, \omega} = D_\theta \circ C_\omega$  のように組み合わせる. このとき相互情報量の推定と最大化を行うための目的関数式 (2.21) は, Jensen-Shannon ダイバージェンスを用いて式 (2.22) と表すことができる [9].

$$\operatorname{argmax}_{\theta, \omega} \mathbb{E}_{P(X, F_\omega(X))} [-\text{sp}(-T_{\theta, \omega}(x, F_\omega(x))) - \mathbb{E}_{P(X, F_\omega(X))} [\text{sp}(T_{\theta, \omega}(\bar{x}, F_\omega(x)))] \quad (2.22)$$

式 (2.22) における  $x, \bar{x}$  はそれぞれ異なる入力サンプルで,  $\text{sp}(z) = \log(1 + e^z)$  である. プログラムで実装する場合は式 (2.22) にマイナスを掛け最小化問題にすることで勾配降下法を用いる.

# 第3章 メル周波数スペクトログラム

メル周波数スペクトログラムは、時間信号を短時間フーリエ変換して得られた振幅スペクトログラムをメル尺度に直したものである [10]. 本章では短時間フーリエ変換とメル周波数について説明する.

## 3.1 短時間フーリエ変換

短時間フーリエ変換 (Short-Term Fourier Transform : STFT) は、時間変化する信号  $f(t)$  に対し窓関数  $w(t)$  をずらしながら掛けていったものをフーリエ変換していく手法である. 時間変化に対する周波数変化の関係を見ることができる. 式 (3.1) は離散時間に関する STFT を示す. また、短時間フーリエ変換して得られたスペクトログラムの絶対値をとったものを振幅スペクトログラムという.

$$\text{STFT}(t, \omega) = \sum_{t=-\infty}^{\infty} f(\tau + t)w(t)e^{-i\omega t} \quad (3.1)$$

## 3.2 メル周波数

メル周波数は人間の音高知覚が考慮された周波数の尺度であり、メル周波数の差が同じであれば、人間の感じる音高の差が同じになることを意図している. 人間は可聴域の下限に近い音は高めに、上限に近い音は低めに聞こえる性質をもつ. メル周波数の単位は mel で表され、1000mel を 1000Hz として基準とし、式 (3.2) で計算される.  $f_0$  は 1000mel = 1000Hz という制約から、式 (3.3) で算出される従属パラメータとなる.

$$m = m_0 \log\left(\frac{f}{f_0} + 1\right) \quad (3.2)$$

$$m_0 = \frac{1000}{\log\left(\frac{1000\text{Hz}}{f_0} + 1\right)} \quad (3.3)$$

# 第4章 関連研究

本章では、本研究に関連がある研究事例を三つを紹介する。はじめに CNN を用いた楽曲ジャンル分類の従来研究について説明し、問題点となる部分を考察する。次に、ジャンル分類基準の可視化という点で、分類した際に強く見ている部分をヒートマップ化する手法である Grad-CAM について説明し、問題点となる部分を示す。最後に、本研究で用いるスペクトログラムをパーカッション成分とハーモニー成分に分ける手法について説明する。

## 4.1 Convolutional Neural Network Achieves Human-level Accuracy in Music Genre Classification

音楽ジャンル分類問題タスクにおいて、CNN を用いることで分類精度を向上させた研究である。CNN の畳み込みフィルタは人間の脳の知覚反応に一致したという結果が報告されている [1]。

### 4.1.1 学習データセット作成

楽曲 10 ジャンルをもつデータセット GTZAN を用いる [11]。モデルの入力データは図 4.1 のように作成する。初めに楽曲信号をオーバーラップ 50% として三秒間毎にメル周波数スペクトログラム  $z_i$  に直していく。次に得られた  $z_i$  に対数を取り、 $f(z_i) = \ln(z_i + 1)$  することでメル周波数スペクトログラムの値の範囲を正規化する。

### 4.1.2 ネットワーク構成

学習に用いるネットワーク構成を図 4.2 に示す。input の次元は(メルスケール、時間)に対応している。また input と最終層以外の層において ReLU、最終層にはソフトマックス関

数を活性化関数として使用している。

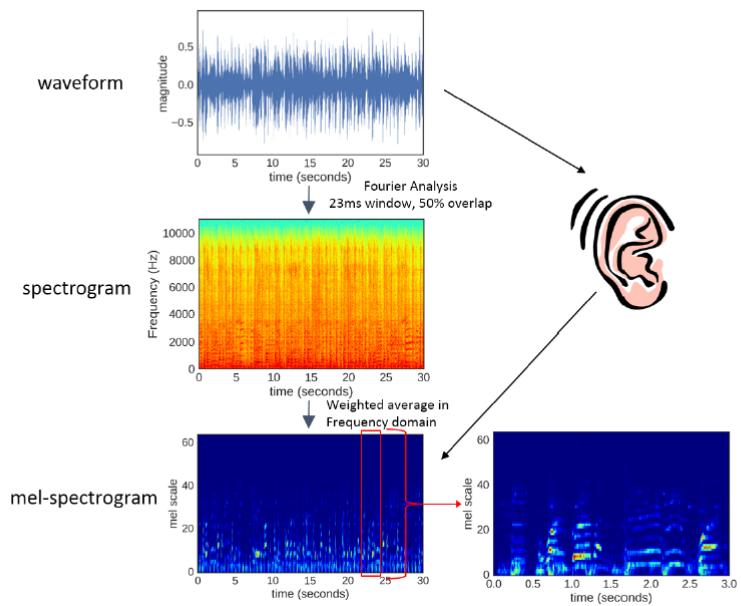


図 4.1 データ前処理

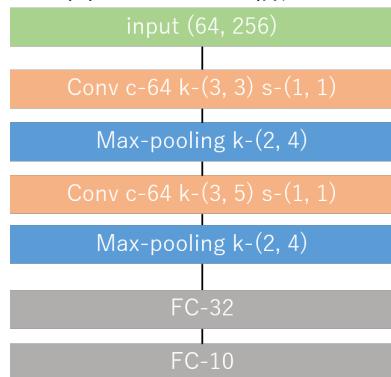


図 4.2 ネットワーク構成

#### 4.1.3 各ジャンルにおける分類精度

ジャンル毎における分類精度を表した混同行列を図 4.3 に示す。図 4.3 から各ジャンルにおいて分類精度にはばらつきがみられ、特に country と rock のジャンルにおいて精度が著しく低くなっている。

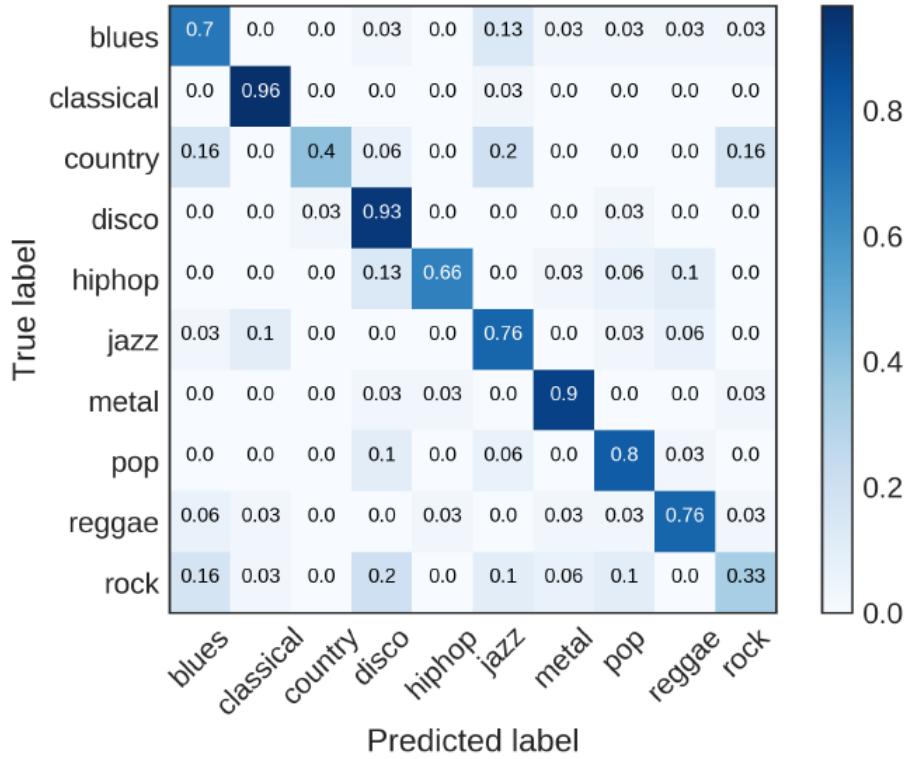


図 4.3 GTZAN を用いた場合の各ジャンルにおける分類精度

#### 4.1.4 問題点

初めに使用しているデータセットに問題点が挙げられ、GTZAN データセットは欠点が存在すると調査されている [12]。欠点の例として、ノイズしか鳴っていないデータがあったり、データセットに重複があるといったものがある。特に重複データにおいては異なるジャンル間で同じ楽曲データが存在してしまっている。そのため、テストデータに学習データが含まれる。

れてしまっている可能性があったり、ラベル付けが不適切であったりなど、実験結果の分類精度に信頼性と説得力が欠けている。

次にモデルの観点から見た問題点を述べる。これは学習済みモデルがブラックボックスなため、どういった点でジャンル分類しているかが不明であるということである。分類精度に改善の余地がみられる点から、モデルを修正する必要性があると考えられる。しかしながらジャンル間の境界面が不明瞭なことから、モデルをどのように修正すればよいのかという指標が立てにくく、試行錯誤的にネットワーク構成を変えながら実験するしかないのが現状である。

## 4.2 Grad-CAM

Grad-CAM は学習済みの CNN が画像を分類した際、画像のどの部分を強く見ているかをヒートマップ化する手法である [2]。

### 4.2.1 入力画像のヒートマップ化

Grad-CAM による入力画像のヒートマップ化の全体の流れを図 4.4 に示す。

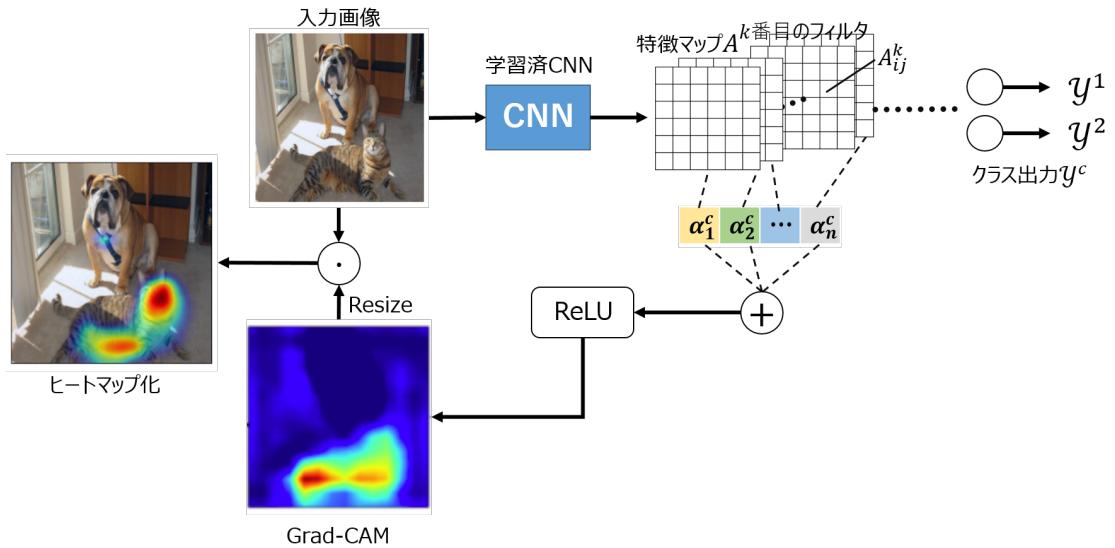


図 4.4 Grad-CAM によるヒートマップ化

学習済み CNN に画像を入力したとき、畳み込み層で得られた  $k$  番目の特徴マップを  $A_{ij}^k$ 、最終層で得られた  $c$  クラスの確率スコアを  $y^c$  とする。このとき式 (4.1) のように、確率スコアに対し特徴マップの勾配をとって平均化したものを重要度  $\alpha_k^c$  とする。

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (4.1)$$

この  $\alpha_k^c$  を用いて、式 (4.2) で示すように  $k$  個の特徴マップで加重平均を計算し、活性化関数 ReLU を通したものヒートマップ出力として定義する。これは、 $A_{ij}^k$  の値の大きさに勾配の大きさも加味することでより重要な箇所を限定していることになる。

$$L_{Grad-CAM}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right) \quad (4.2)$$

最後にヒートマップを入力画像に合わせてリサイズし、入力画像との畳み込み演算により入力画像のヒートマップ化を行っている。

この手法の重要な点は  $A_{ij}^k$  の値をうごかした時に、 $y^c$  のスコアがどのように変化するかという点である。 $\frac{\partial y^c}{\partial A_{ij}^k} > 0$  の場合は  $A_{ij}^k$  が増加する方向に  $y^c$  が増加し、 $\frac{\partial y^c}{\partial A_{ij}^k} < 0$  の場合は  $A_{ij}^k$  が減少する方向に  $y^c$  が増加する。このとき、 $A_{ij}^k$  は ReLU を通した後の値と仮定すれば、 $A_{ij}^k \geq 0$  である。そのため、 $\frac{\partial y^c}{\partial A_{ij}^k} < 0$  の箇所は非活性なピクセルであると考えられ、式 (4.2)において ReLU を通することで、勾配が正の部分だけでヒートマップ化を行っている。

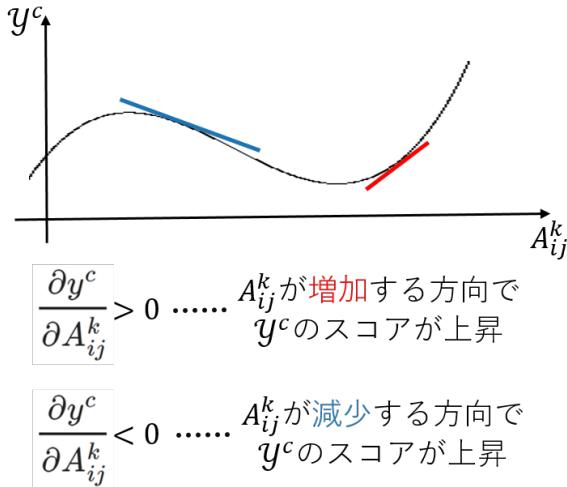


図 4.5  $A_{ij}^k$  に関する勾配

#### 4.2.2 問題点

特徴マップによる勾配をとることから、ヒートマップの形が特徴マップのサイズや形に大きく依存してしまうという欠点がある。極端な例では、畠み込み後の特徴マップが $1 \times 1$ の場合ヒートマップ出力が $1 \times 1$ になってしまい、入力画像全体がヒートマップ化されてしまう。

また、勾配をとるという点で勾配消失の問題がある。例えば学習が十分に進んだモデルに対し、学習データに含まれる画像を入力した場合、 $y_c$ の確率スコアが限りなく1に近づく。この時ソフトマックス関数の勾配は限りなく0に近づくため、プログラム上で実装したとき勾配が0となってしまう。そのため重要度  $\alpha_k^c$  が0になってしまい、入力画像のヒートマップ化が不可能となる。

さらに、画像を入力としてクラス出力を行うため、画像を一枚一枚用意するのに手間がかかる。クラス間をまたがって画像は連続変化していくという前提を考えたとき、クラス境界となる画像が必ず存在する。しかし、その境界画像を確認するためには試行錯誤に画像を用意して探さなければならない。

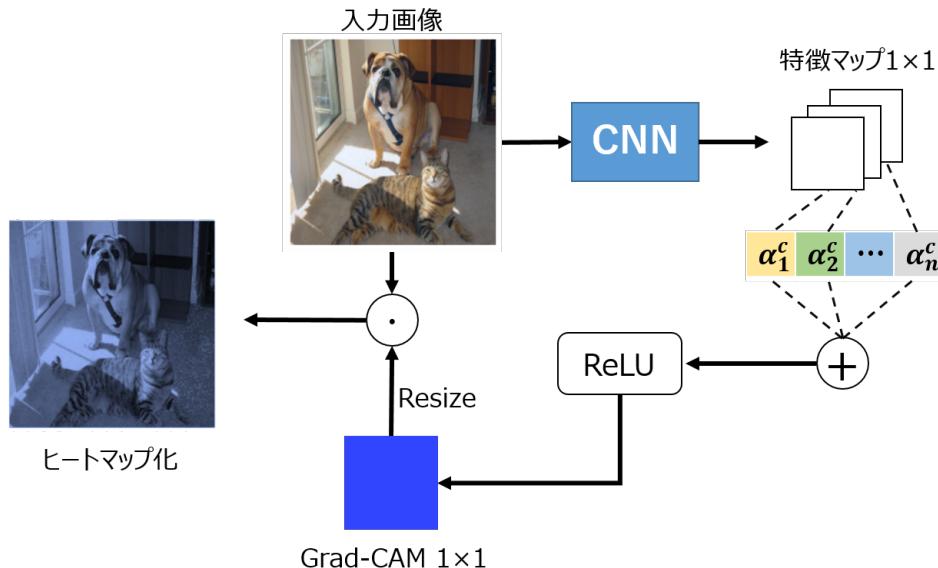


図 4.6  $1 \times 1$  の特徴マップによる Grad-CAM

## 4.3 Harmonic/Percussive Separation using Median Filtering

楽曲の周波数振幅スペクトログラムをメディアンフィルタを用いてハーモニー成分とパーカッション成分のスペクトログラムに分けた研究である [13]. ハーモニー成分のスペクトログラムは周波数軸方向に表れ、パーカッション成分のスペクトログラムは時間軸方向に表れるという仮定のもと推定を行い、オリジナルのスペクトログラムにマスクすることによって一方の成分を取り出している。

### 4.3.1 メディアンフィルタ

メディアンフィルタは画像処理の分野で多く用いられており、主にノイズ除去のために使われている。メディアンはその名の通り中央値を表しており、図 4.7 のような 1 次元データがあったとき、値を小さい順に並べていき中央になった値をデータ配列の中心の値と置き換えるフィルタである。画像などといった 2 次元データの場合は図 4.8 のようにフィルタの中央となる部分の値が置き換わる。



図 4.7 1 次元メディアフィルタの動作

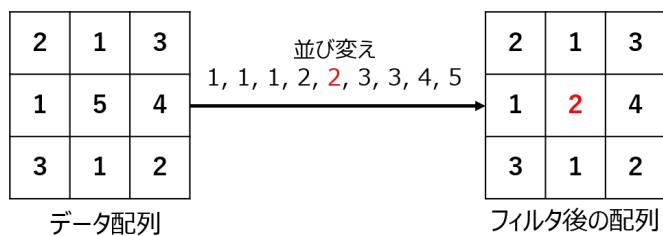


図 4.8 2 次元メディアフィルタの動作

#### 4.3.2 ハーモニー成分とパーカッション成分

ハーモニー成分のスペクトログラムは周波数軸方向に表れやすいと仮定する。この仮定により、周波数軸方向にメディアンフィルタを適用していけばハーモニー成分を取り除くことができる。図 4.9 上はスネアとピアノがミックスされた音の特定時間におけるスペクトラムであり、図 4.9 下は周波数軸方向にメディアンフィルタを掛けた後のスペクトラムである。図 4.9 より、ピアノのハーモニーの主となるスペクトラムの倍音のピークがメディアンフィルタによって消えていることから、メディアンフィルタはハーモニー成分を取り除いていることが分かる。

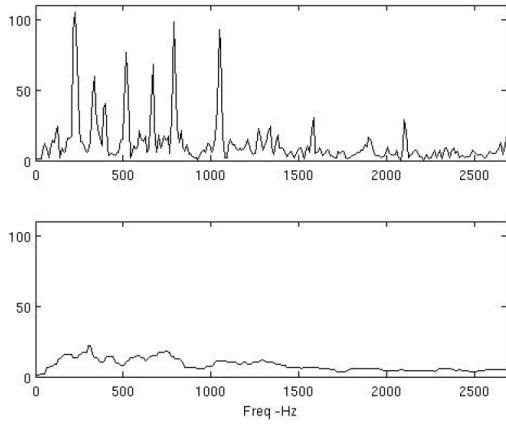


図 4.9 周波数軸に対するメディアンフィルタ

次に、パーカッション成分は時間軸方向のオンセットに表れやすいと仮定する。同様に、時間軸方向にメディアンフィルタを適用していけばパーカッション成分を取り除くことができる。図 4.10 上はスネアとピアノがミックスされた音の特定周波数における時間軸の増減を表しており、図 4.10 下は時間軸方向にメディアンフィルタを掛けた後の特定周波数における時間軸の増減である。図 4.10 より、スネアの主となるオンセットがメディアンフィルタによって抑制されているため、パーカッション成分を取り除いていることが分かる。

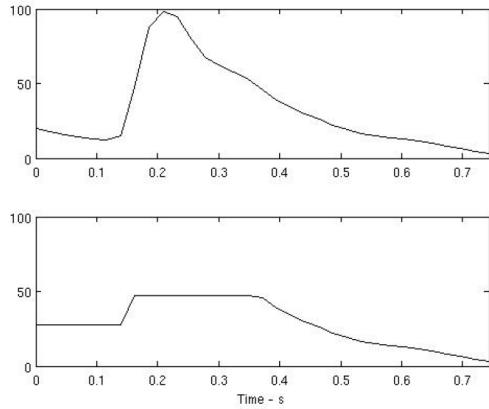


図 4.10 時間軸に対するメディアンフィルタ

#### 4.3.3 ハーモニー成分とパーカッション成分の分離

楽曲のスペクトログラムをハーモニー成分とパーカッション成分に分離することを考える。楽曲信号をフーリエ変換して得られたスペクトログラムを、周波数軸方向を  $h$  と時間軸方向を  $i$  として  $S_{h,i}$  と表し、ハーモニー成分が抑制されたスペクトログラムを  $H_{h,i}$ 、パーカッション成分が抑制されたスペクトログラムを  $P_{h,i}$  とすると、式(4.3)、式(4.4)が成り立つ。

$$H_{h,i} = \text{MF}(S_h, l_h) \quad (4.3)$$

$$P_{h,i} = \text{MF}(S_i, l_i) \quad (4.4)$$

MF はメディアンフィルタを示し、 $l_h, l_i$  はそれぞれフィルタの長さとする。この  $H_{h,i}, P_{h,i}$  を用いて式(4.5)、式(4.6)もしくは式(4.7)、式(4.8)のようにして、ハーモニー成分のマスクスペクトログラム  $M_{H_{h,i}}$  とパーカッション成分のマスクスペクトログラム  $M_{P_{h,i}}$  を作成する。

$$M_{H_{h,i}} = \begin{cases} 1 & (H_{h,i} > Ph, i) \\ 0 & (\text{otherwise}) \end{cases} \quad (4.5)$$

$$M_{P_{h,i}} = \begin{cases} 1 & (Ph, i > Hh, i) \\ 0 & (\text{otherwise}) \end{cases} \quad (4.6)$$

$$M_{H_{h,i}} = \frac{H_{h,i}^k}{H_{h,i}^k + P_{h,i}^k} \quad (4.7)$$

$$M_{H_{h,i}} = \frac{P_{h,i}^k}{H_{h,i}^k + P_{h,i}^k} \quad (4.8)$$

式(4.7), 式(4.8)における $k$ は1か2の値をとる。

最後に式(4.9), 式(4.10)のように, 得られたマスクスペクトログラムをオリジナルのスペクトログラムに畳み込みこむことでハーモニー成分 $\hat{\mathbf{H}}$ とパーカッション成分 $\hat{\mathbf{P}}$ を得る。

$$\hat{\mathbf{H}} = \hat{\mathbf{S}} \otimes \mathbf{M}_{\mathbf{H}} \quad (4.9)$$

$$\hat{\mathbf{P}} = \hat{\mathbf{S}} \otimes \mathbf{M}_{\mathbf{P}} \quad (4.10)$$

#### 4.3.4 メリット

メディアンフィルタという特定のフィルタを用いていることから, 検出方法が自明である。そのため成分を分ける操作において, 機械学習などを用いたときに現われるブラックボックスといった問題点がないということが良い点であると言える。また, アルゴリズムが比較的容易なため, プログラムの実装が簡単であり, さらにはライブラリ等でも実装されている。また, パーカッション成分とハーモニー成分をマスクするデータを作る際に, それぞれの成分で仮定した定義が直感的にも当てはまる。以上の点から本研究で楽曲データセットに対し, ハーモニー成分とパーカッション成分を分ける際に使用する。

# 第5章 提案手法

4.1.4, 4.2.2 で述べた問題点を解決するための手法を提案する。はじめに 5.1 節で GTZAN データセットを修正し、学習用データセットを作成する。次に 5.2 節で楽曲ジャンルを分類するモデルを構築し、5.3 節で学習済みの分類モデルを用いて、スペクトログラムを生成するモデルを構築する。最後に 5.4 節で、生成モデルを用いてジャンル境界を可視化する手法を述べる。

## 5.1 データセットの作成

### 5.1.1 GTZAN データセットの修正

本研究では、楽曲データセットとして GTZAN を使用する。GTZAN データセットは図 5.1 のように Blues, Country, Classical, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, Rock の 10 ジャンルの楽曲から構成される。各ジャンルを説明した表は表 5.1 に示す [15]。1 データ当たりは 30 秒間の楽曲データであり、1 ジャンルにつき 100 曲ずつ用意されている。GTZAN データセットには、ノイズデータや重複したデータが存在しているため、ノイズデータと重複した片方の楽曲データを除去する必要がある [12]。そのため最終的なデータセット数は表 5.2 のようになった。

### 5.1.2 学習用データセット作成

一般に音を分析するために使用されるデータとしては、音信号を STFT した周波数情報が用いられる。そのため、楽曲ジャンルを分類においても、どの周波数がどの程度含まれるかということが重要であると考えられる。さらに楽曲という点でリズムとメロディーが存在するため、パーカッション成分とハーモニー成分に分けて分析することがより良い特徴が得

表 5.1 各ジャンルの主な説明

ジャンル	説明
Blues	米国深南部でアフリカ系アメリカ人の間から発生した音楽の一種およびその楽式。ギターを用いた歌が主役である。
Country	1920 年代にアメリカ合衆国南部で発祥したとされる音楽。シンプルなハーモニーを形成し、バラードからダンス音楽まで幅広い音楽性を持つ。
Classical	バロック音楽、古典派音楽、ロマン派音楽に当たる 1550 年頃から 1900 年頃の音楽。
Disco	一定のリズムを刻む 4 つ打ち、8 分音符ないし 16 分音符刻みかつオフビートでオープンするハイハットパターンがある音楽。さらに突出したシンコペーションを持ったり、時にはオクターブでなるエレキベースのベースラインの上で演奏される。
Hip-hop	1970 年代のアメリカ合衆国ニューヨークのブロンクス区で、アフロ・アメリカンやカリビアン・アメリカン、ヒスパニック系の住民のコミュニティで行われていたブロックパーティから生まれた音楽。MC によるラップを乗せた音楽形態を指すことが一般化している。
Jazz	19 世紀末から 20 世紀初頭にかけてアメリカ合衆国南部の都市を中心に派生した音楽。演奏の中にブルー・ノート、シンコペーション、スウィング、コールアンドレスポンス、即興演奏、ポリリズムなどの要素を組み込んでいることが大きな特徴とされている。
Metal	ギター、ドラム、ボーカル、ベースを主軸とし、一般的には音の「ヘヴィさ」を重視した音楽。そのためギターやベースのチューニングを下げ、通常より低い音が出せるようにしている場合もある。
Pop	1950 年代から 1960 年代にかけて西洋でロックンロールから派生して現代的形態で始まったポピュラー音楽。動きのあるメロディが重視され、基本的な楽式を楽曲中で繰り返すといった普遍的な特徴を持つ。
Reggae	ジャマイカで成立したポピュラー音楽全般。4 分の 4 拍子の第 2・第 4 拍目をカッティング奏法で刻むギター、各小節の 3 拍目にアクセントが置かれるドラム、うねるようなベースラインを奏てるベースなどの音楽的特徴を持つ。
Rock	1950 年代にアメリカ合衆国の黒人音楽である Rocknroll や Blues, Country を起源とし、1960 年代以降にイギリスやアメリカ合衆国で幅広く多様な様式へと展開した音楽。サウンドは伝統的にエレクトリックギターが中心となる。

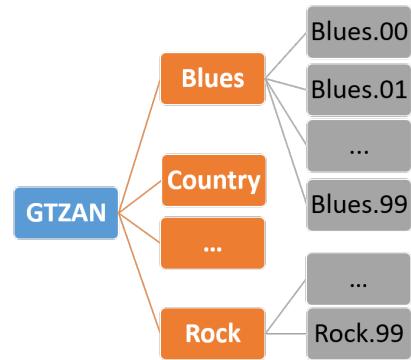


図 5.1 GTZAN データセット

表 5.2 ジャンル毎のデータセット数

ジャンル	データセット数
Blues	100
Country	100
Classical	100
Disco	94
Hip-hop	98
Jazz	87
Metal	91
Pop	91
Reggae	89
Rock	100

られると考えられる。そこで楽曲をパーカッション成分とハーモニー成分に分けた周波数振幅スペクトログラムを生成する [13].

また、人間の聴覚に合わせた周波数がさらに効果的であると考えられるため、楽曲を周波数振幅スペクトログラムに変換した後、メルスケールに直したメル周波数スペクトログラムを学習データとする。学習させる楽曲データの時間的な長さに関しては、Weibin らの研究により 3 秒間が一番良い結果を得ているため。それに倣い 3 秒間のスペクトログラムを用いる [14]。以上の点を踏まえて図 5.2 のように、学習に用いるデータセットを作成する。

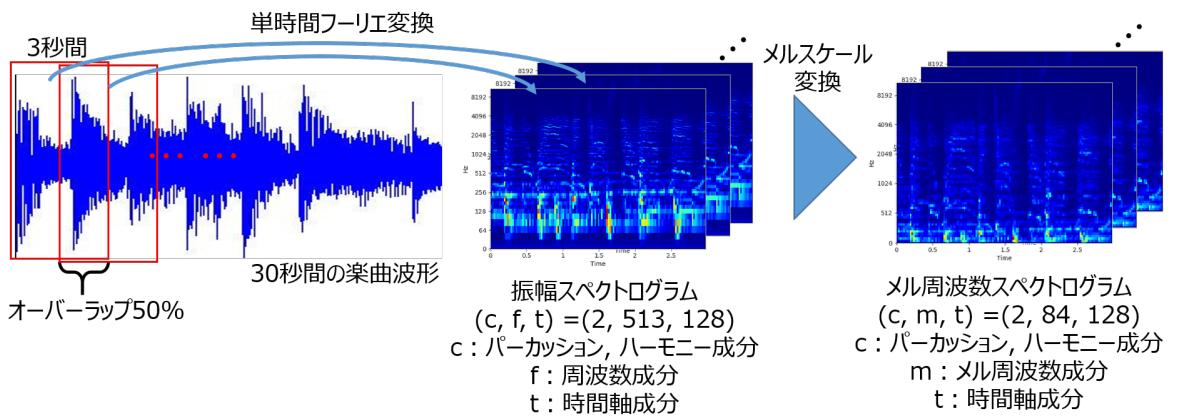


図 5.2 学習データセット作成

ここで、データセットを増やすという目的で、STFT する際の窓のずらし方をオーバーラップを 50% として 3 秒間ごとに切り抜いていく。

また音量をバランスを統一するために 3 秒間の波形の振幅値を式 (5.1) を用いて -1~1 に正規化する。さらにモデルへの入力スケールを合わせるために、式 (5.2) のように最大値で除算しメル周波数スペクトログラムの値を 0~1 の範囲で正規化する。

$$signal = \frac{signal}{max(abs(signal))} \quad (5.1)$$

$$mel = \frac{mel}{max(mel)} \quad (5.2)$$

## 5.2 ジャンル分類器の構築

楽曲 10 ジャンルを分類する学習済みモデルを構築する。データセットは 5.1 節で述べたものを用いて、学習用データとテスト用データに分ける。

### 5.2.1 ジャンル分類器

5.1 節で作成したスペクトログラムを用いて、図 5.3 のような入力をスペクトログラム、出力を楽曲 10 ジャンルに設定した CNN を分類器として構築する。CNN のネットワーク構成は図 5.4 に示す。なお学習時の損失関数は交差エントロピー誤差とする。

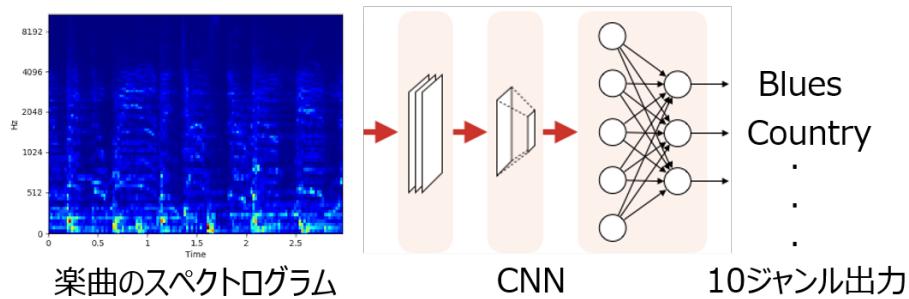


図 5.3 楽曲ジャンル分類モデル

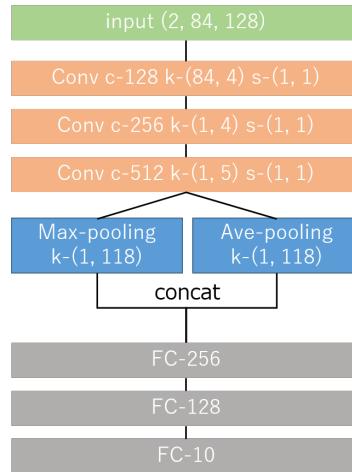


図 5.4 CNN のネットワーク構成

### 5.3 データ生成器の構築

5.2節で構築した学習済みCNNを用いて、ジャンル毎で異なるデータを生成するようなモデルを構築する。本稿で構築した生成器は $-1\sim1$ の一様分布に従うランダムノイズを入力とし、メル周波数スペクトログラムを出力する。さらに生成されたスペクトログラムを学習済CNNによって分類する。このときノイズベクトルの値を連続的に変化させることで、生成データの連続変化とジャンル出力の連続変化が同時に確認できる。図5.5はデータ生成器を学習する際の全体像を示しており、図5.6は学習後に使用するモデルの全体像を示した図である。

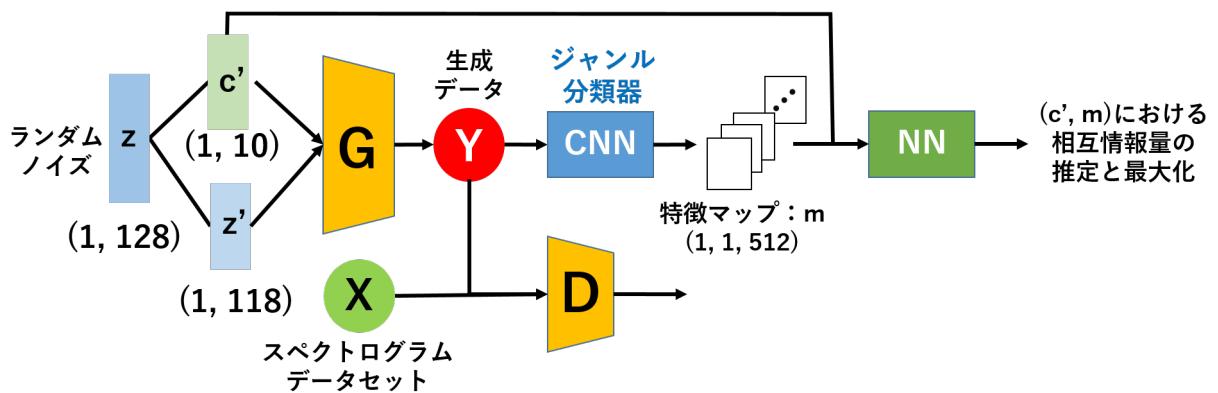


図5.5 データ生成器の学習

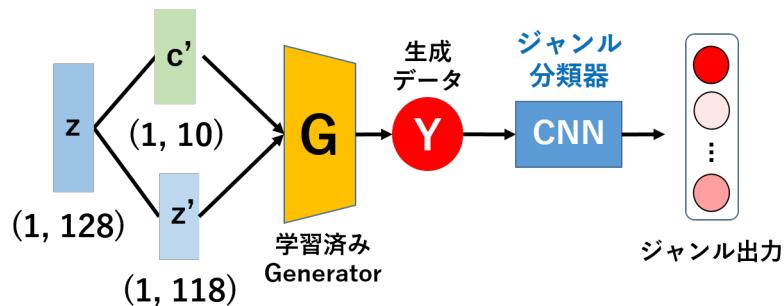


図5.6 生成データの分類

### 5.3.1 生成器モデル

図 5.7 のような GAN モデルを構築する。 $-1 \sim 1$  の一様分布に従うランダムノイズを Generator に入力し、Generator が output したデータと GTZAN データセットのスペクトログラムを Discriminator によって判定する。これにより生成するデータをスペクトログラムに近づけていく。使用する Generator と Discriminator のネットワーク構成を図 5.8 と図 5.9 に示す。

Generator の入力されるノイズベクトルの次元は 128 次元であり、全結合層で次元を増加させ、逆畳み込みによって最終的にデータセットのメル周波数スペクトログラムと同じ次元の  $(2, 84, 128)$  の配列を出力する。一方 Discriminator の入力はデータセット配列または Generator によって生成された配列を入力とし、畳み込み層で特徴マップを得たのち全結合層によって 1 次元の値まで次元が縮小されていく。この 1 次元の出力値が Wasserstein 距離となる。

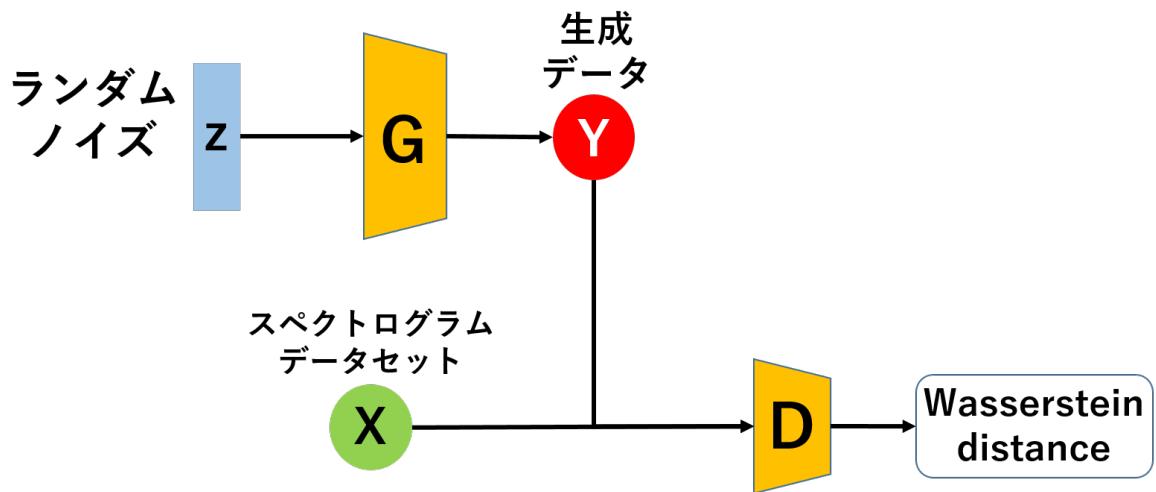


図 5.7 生成器モデル

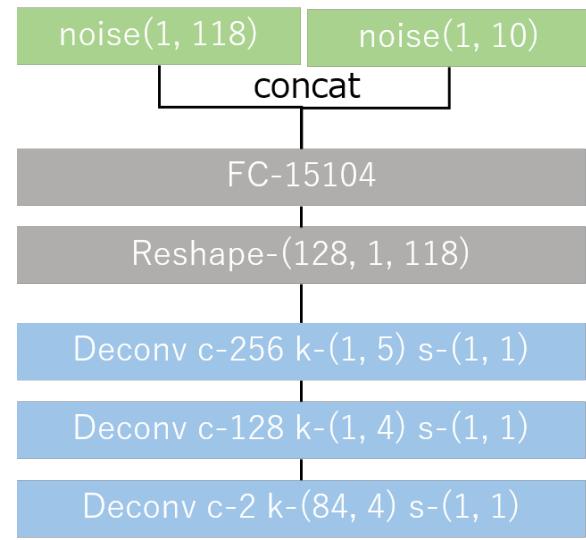


図 5.8 Generator のネットワーク構成

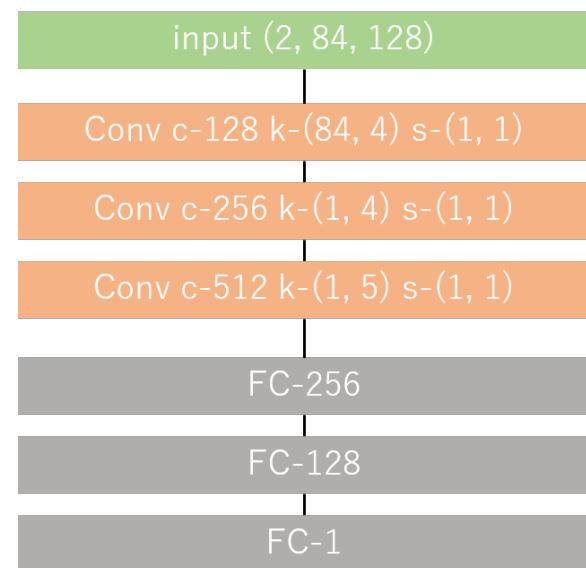


図 5.9 Discriminator のネットワーク構成

### 5.3.2 ランダムノイズと分類結果の従属性

生成データの分類結果と、Generator の入力ランダムノイズに従属関係を持たせたい。そこで、NN を用いた相互情報量を推定し最大化する手法を用いる。図 5.10 のように Generator の入力ランダムノイズの 128 次元ベクトルを 118 次元と 10 次元のベクトルに分け、10 次元のベクトル  $c'$  と CNN のプーリング後の特徴マップ  $m$  との間に、相互情報量を推定し最大化するような NN の学習を行う。これにより 10 次元の入力ベクトルと出力される特徴マップとの間に従属関係ができるため、特徴マップ後の全結合層の推論にも影響を与える。よって Generator の入力変数値とジャンル出力値が互いに影響しあうようなモデルが構築できる。また、図 5.10 における NN のネットワーク構成を図 5.11 に示す。

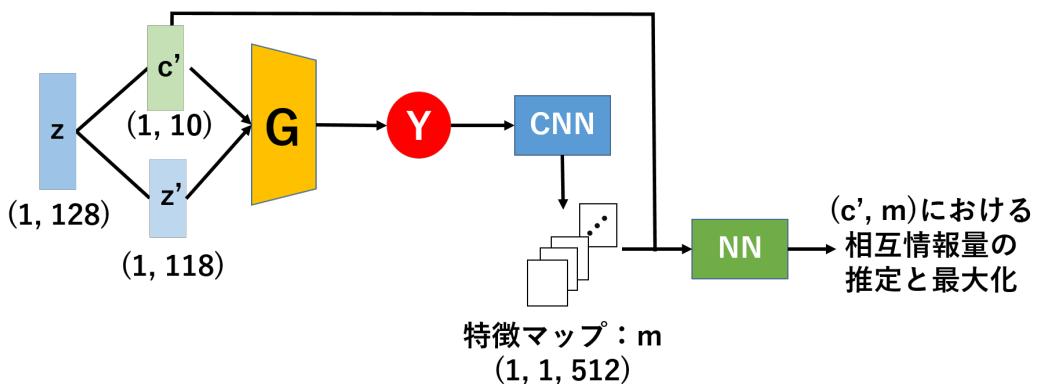


図 5.10 相互情報量の推定と最大化

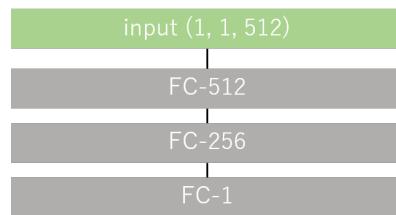


図 5.11 NN のネットワーク構成

## 5.4 ジャンル境界の可視化

5.3 節で構築した学習済み Generator を用いて、入力ノイズベクトルとジャンル出力の関係を二次元ジャンルマップ空間として可視化する。これにより、ジャンル間の境界を可視化することに加えスペクトログラムも生成することができるため、ジャンルにまたがったスペクトログラムの変化の過程を追うことができ、分類器がどのようにジャンルを分けているかという点において人間が意味解釈をする際の手助けとなる。

図 5.12 のように初めに初期値として  $-1 \sim 1$  の一様分布に従うランダムノイズを 128 次元生成する。次に図 5.13 のように従属関係にある 10 次元のベクトル  $c$  から 2 次元だけを取り出しその値を変化させいく。このとき、値の変換に応じて生成するスペクトログラムとジャンル分類結果が変化するため、その時の 2 次元ベクトルの値とジャンル分類結果を図 5.14 のようにジャンルマップにプロットしていく。これにより、ジャンル境界となる座標が明らかとなる。

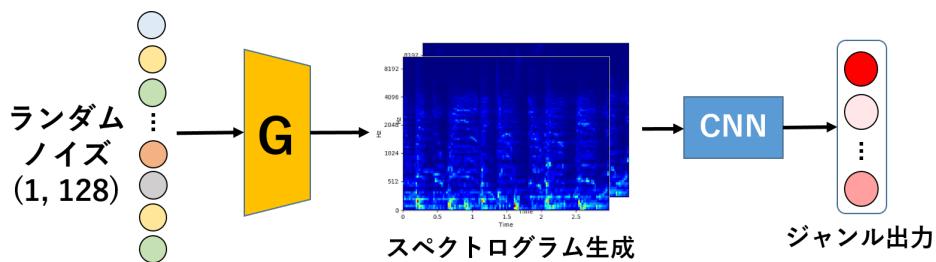


図 5.12 初期値ランダムノイズ生成

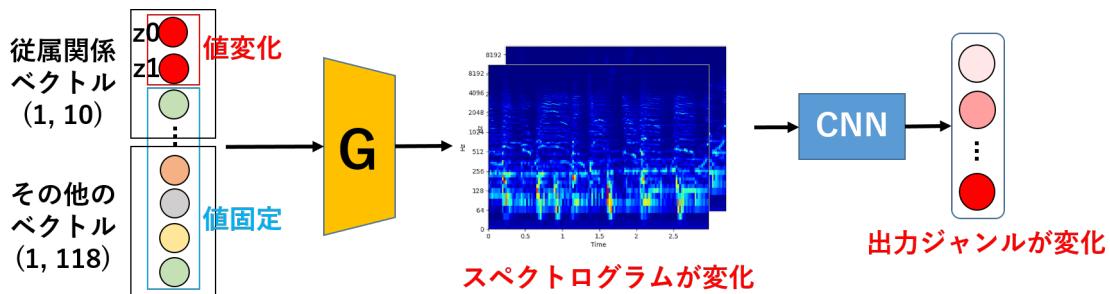


図 5.13 従属関係のベクトル値を変化

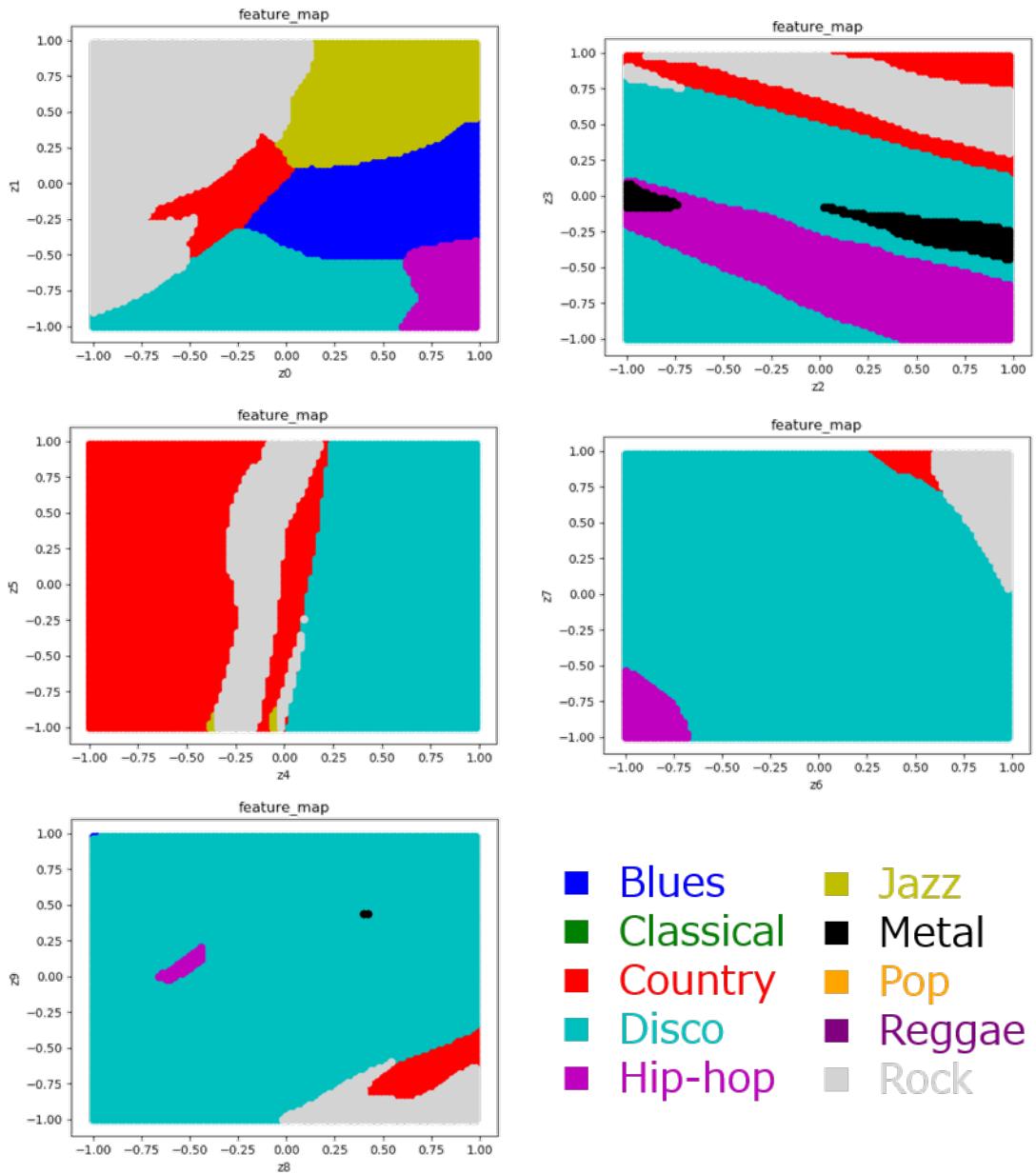


図 5.14 軸ごとにおける 2 次元ジャンルマップの例

# 第6章 実験と検証

提案手法によって 5.2 節で構築した分類モデルと 5.3 節で構築した生成モデルの評価を行う。また 5.4 節のように生成モデルから得られる 2 次元ジャンルマップ空間に対し検証を行う。

## 6.1 分類モデルの評価

### 6.1.1 学習毎の損失と精度のグラフ

5.2 節の学習毎の CNN において、1 データ当たり 3 秒間の学習データとテストデータの組み合わせを変えた実験を行う。合計 10 通りの組み合わせを行ったとき、それぞれの学習毎のモデルにおける分類時の損失と精度をグラフにプロットしたもの図 6.2～図 6.11 に示す。横軸は学習回数 (epoch)，縦軸はその時の損失と精度を表している。

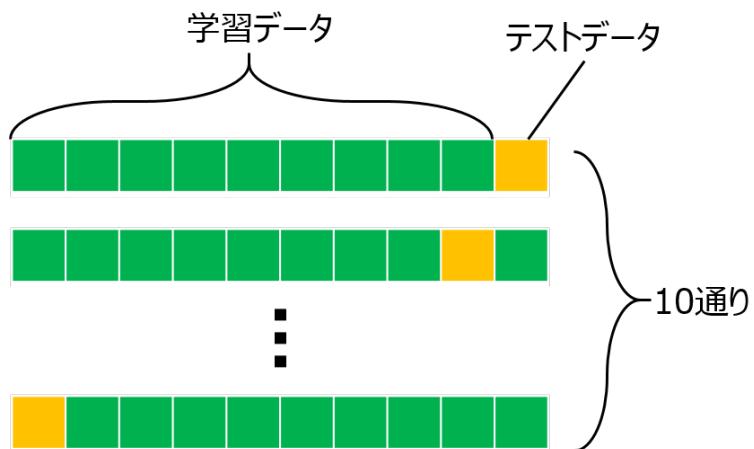


図 6.1 10 分割交差検証

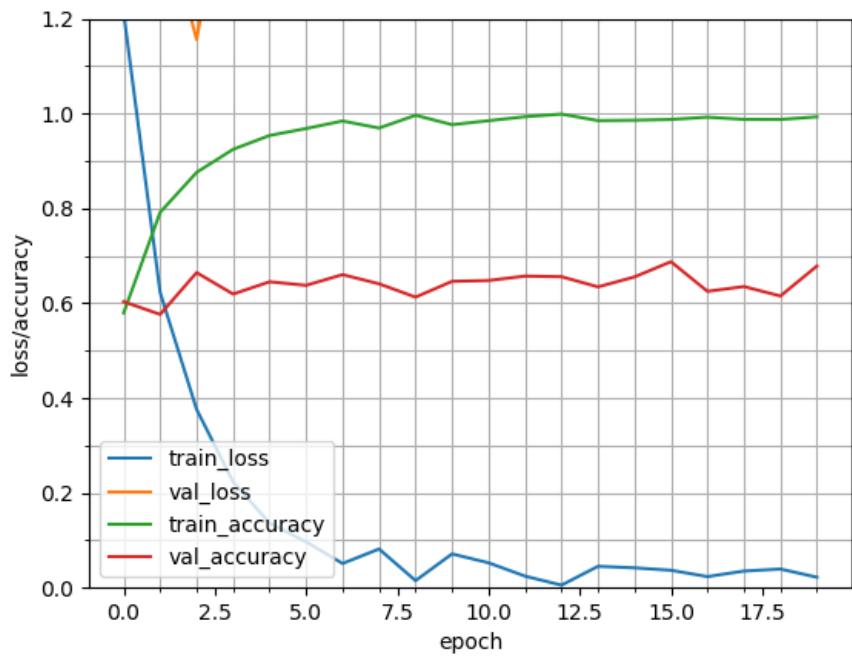


図 6.2 損失と精度パターン 0

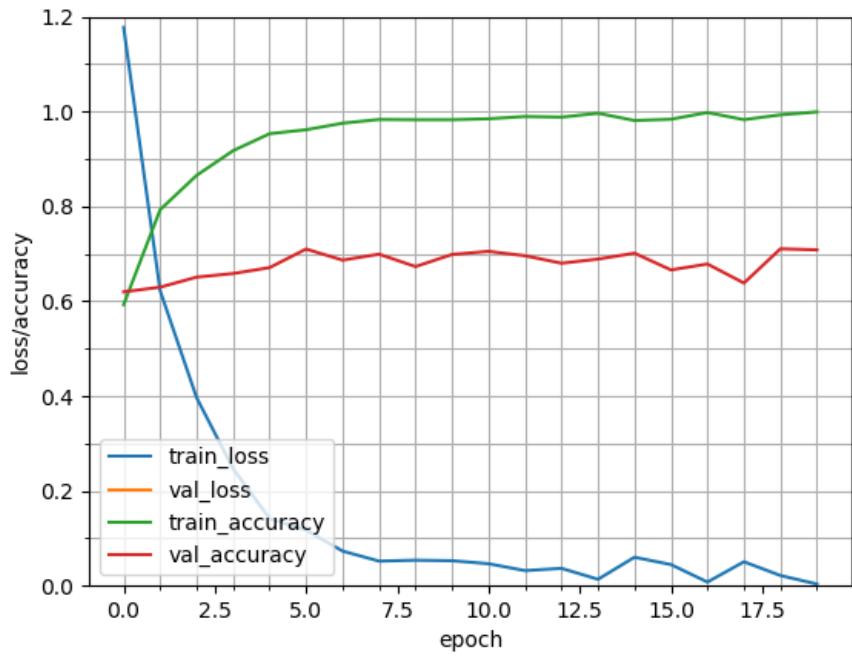


図 6.3 損失と精度パターン 1

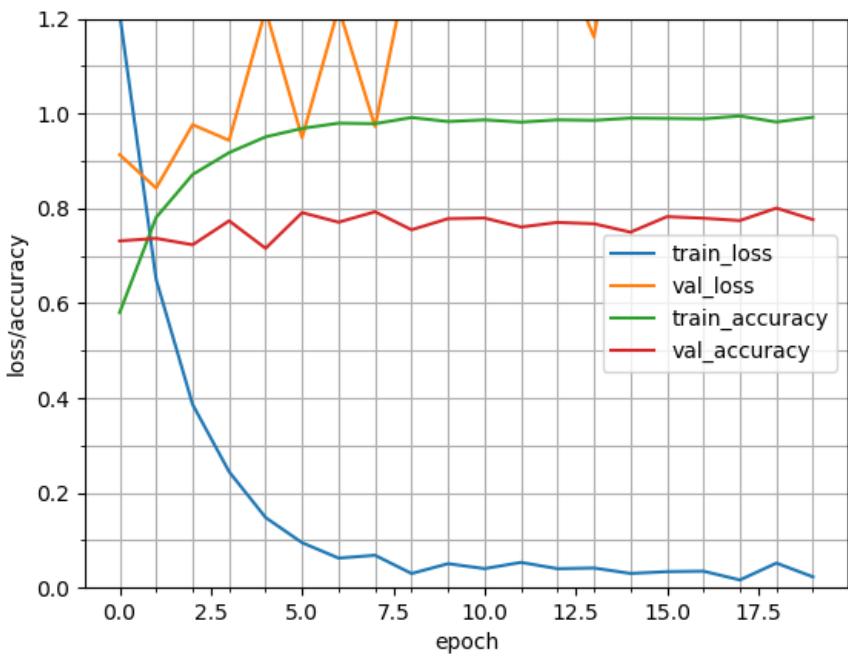


図 6.4 損失と精度パターン 2

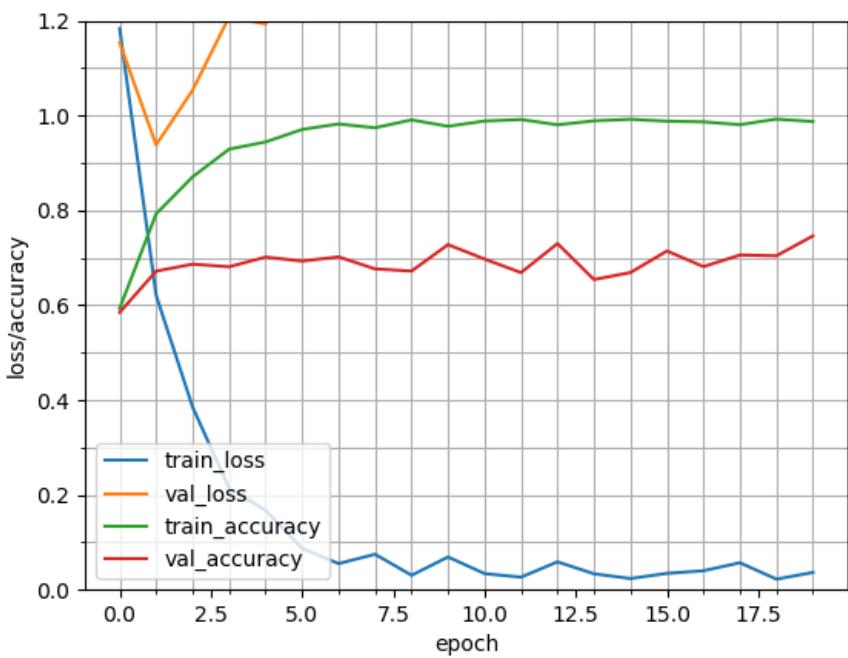


図 6.5 損失と精度パターン 3

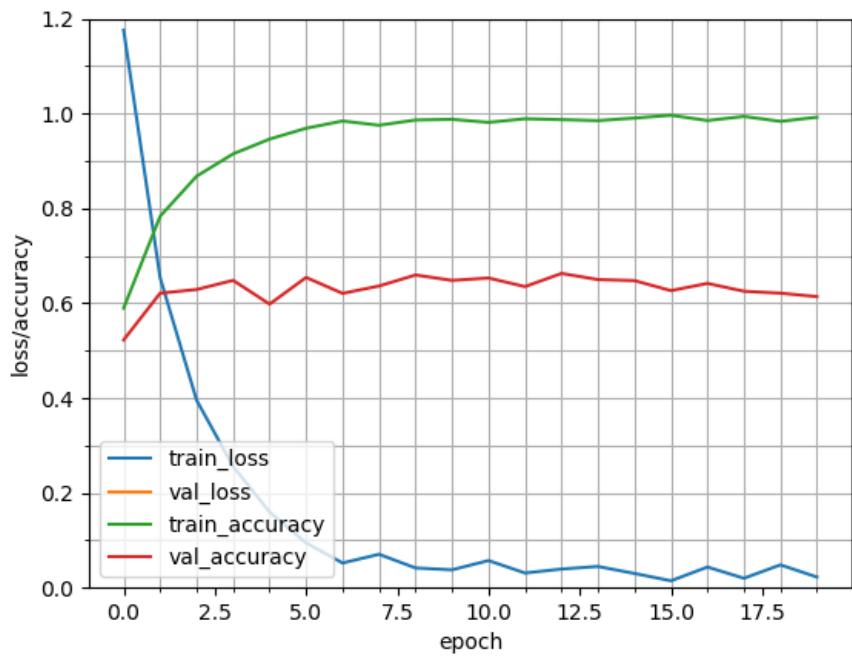


図 6.6 損失と精度パターン 4

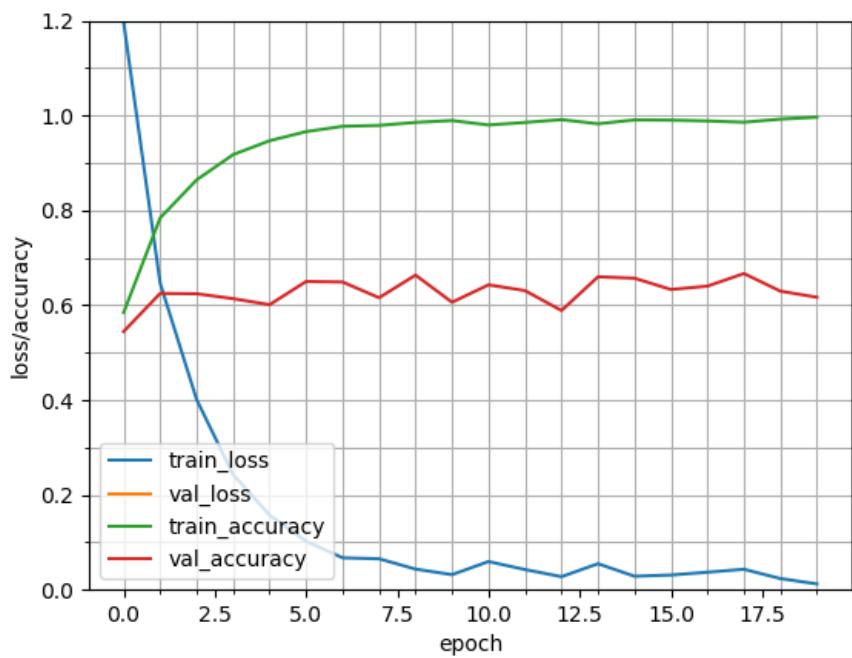


図 6.7 損失と精度パターン 5

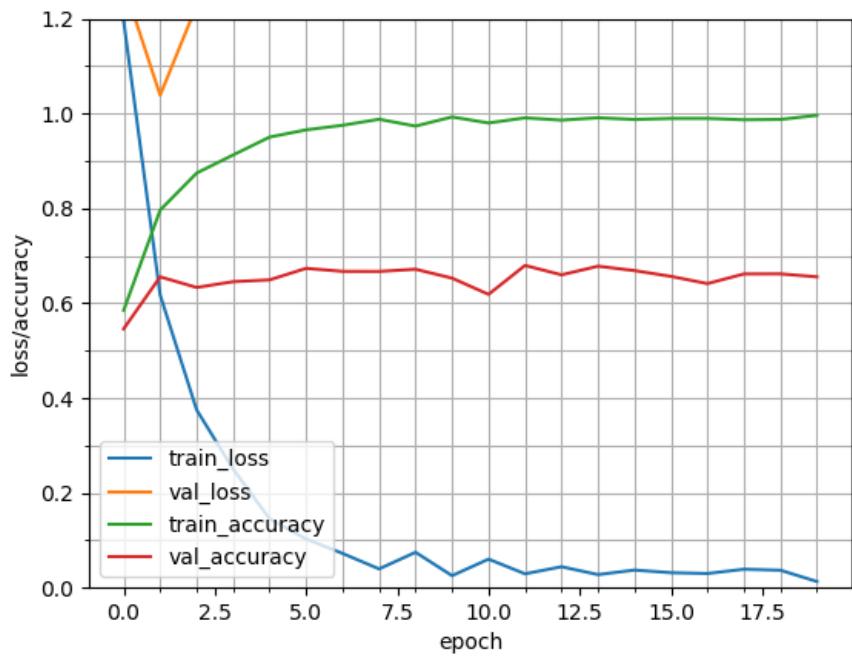


図 6.8 損失と精度パターン 6

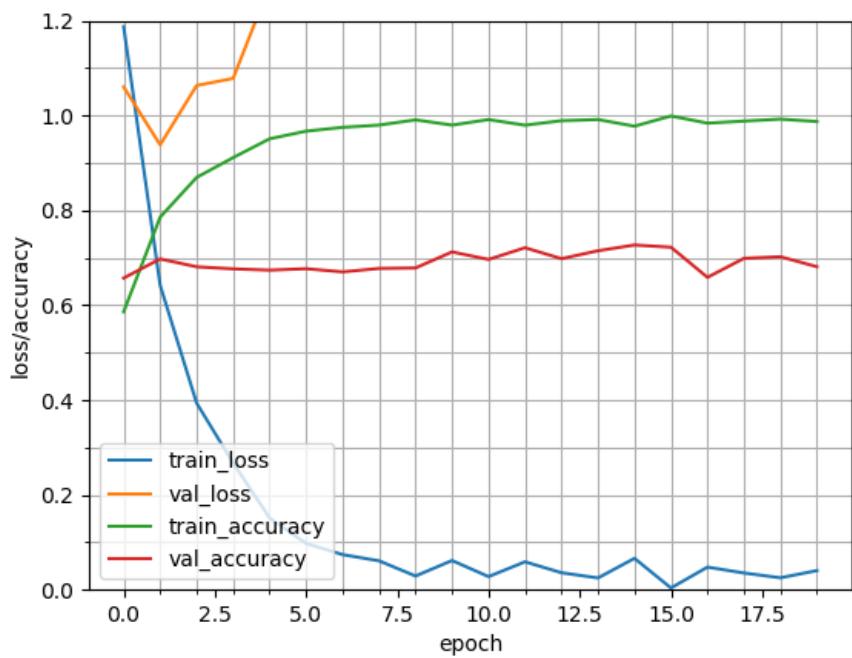


図 6.9 損失と精度パターン 7

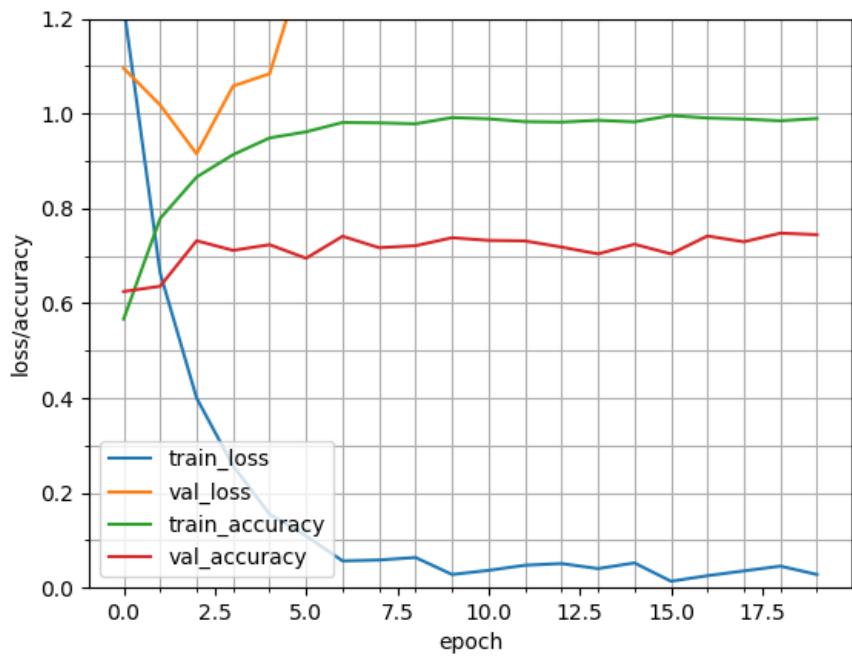


図 6.10 損失と精度パターン 8

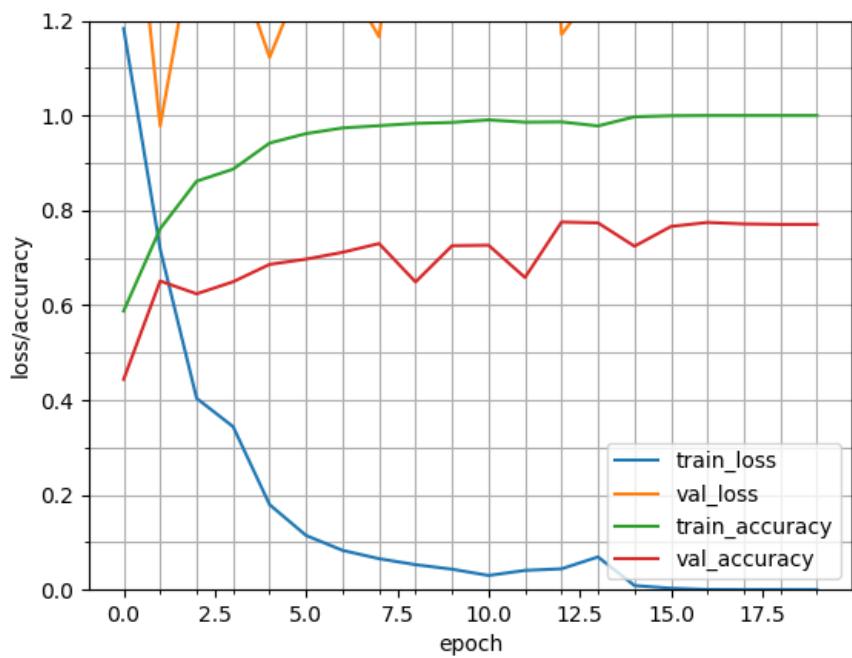


図 6.11 損失と精度パターン 9

図 6.2～図 6.11 より、学習データの組み合わせによって、テストデータに対する分類精度が大きく影響しているということが読み取れる。特に図 6.4 より、パターン 2 のデータセットを用いたときにおけるテストデータの精度が一番高いため、データセット 2 を用いた学習済み CNN はより最適解に近い学習を行ったと言える。よって学習済み CNN から特徴を分析する際には、学習データセットをパターン 2 に設定したモデルを使用することで、より信頼性のあるジャンル境界線を可視化できると考えられる。

### 6.1.2 1曲分の分類精度

図 6.12 のように 1 曲 30 秒間のテストデータを 3 秒ごとに分類し、多数決をとったものを 1 曲分のジャンル分類結果として評価した場合の精度を示す。10 通りのモデルセットにおけるテストデータを図 6.12 のように分類していく、精度を算出した際の混同行列を図 6.13～図 6.22 に示す。また 10 パターンのモデル精度を平均した時の混同行列を図 6.23 に示す。

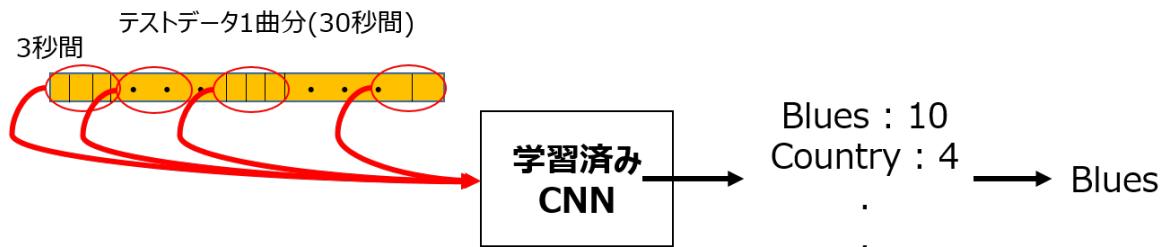


図 6.12 1 曲分の分類精度

ラベル 予測 \	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Classical	0.00	1.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00	0.00
Country	0.00	0.00	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.30
Disco	0.00	0.00	0.00	0.89	0.11	0.00	0.00	0.00	0.13	0.00
Hiphop	0.00	0.00	0.00	0.00	0.78	0.00	0.00	0.00	0.00	0.00
Jazz	0.00	0.00	0.20	0.00	0.00	0.88	0.00	0.00	0.00	0.00
Metal	0.00	0.00	0.00	0.00	0.00	0.00	0.78	0.00	0.00	0.00
Pop	0.00	0.00	0.40	0.00	0.00	0.00	0.11	0.89	0.00	0.00
Reggae	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.75	0.00
Rock	0.10	0.00	0.10	0.11	0.00	0.00	0.11	0.11	0.13	0.70

図 6.13 ジャンル毎の分類精度：パターン 0

ラベル 予測 \	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	1.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00	0.20
Classical	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.00
Country	0.00	0.00	0.70	0.00	0.00	0.00	0.00	0.44	0.11	0.00
Disco	0.00	0.00	0.10	0.67	0.00	0.00	0.00	0.00	0.11	0.00
Hiphop	0.00	0.00	0.00	0.11	0.89	0.00	0.00	0.00	0.00	0.20
Jazz	0.00	0.00	0.10	0.00	0.00	0.88	0.00	0.00	0.00	0.00
Metal	0.00	0.00	0.00	0.11	0.11	0.00	1.00	0.00	0.00	0.30
Pop	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00
Reggae	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.78	0.00
Rock	0.00	0.00	0.10	0.11	0.00	0.00	0.00	0.00	0.00	0.30

図 6.14 ジャンル毎の分類精度：パターン 1

予測\ラベル	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Classical	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Country	0.10	0.00	0.90	0.11	0.00	0.00	0.11	0.00	0.00	0.30
Disco	0.00	0.00	0.00	0.44	0.00	0.00	0.00	0.00	0.00	0.00
Hiphop	0.00	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.11	0.00
Jazz	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Metal	0.00	0.00	0.00	0.00	0.00	0.00	0.89	0.00	0.00	0.10
Pop	0.00	0.00	0.10	0.44	0.00	0.00	0.00	1.00	0.11	0.00
Reggae	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.78	0.00
Rock	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.60

図 6.15 ジャンル毎の分類精度：パターン 2

予測\ラベル	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	0.90	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10
Classical	0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Country	0.00	0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.10
Disco	0.00	0.00	0.10	0.89	0.10	0.00	0.00	0.00	0.00	0.10
Hiphop	0.00	0.00	0.00	0.00	0.60	0.00	0.00	0.00	0.00	0.00
Jazz	0.10	0.00	0.10	0.00	0.00	1.00	0.00	0.00	0.00	0.20
Metal	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Pop	0.00	0.00	0.00	0.00	0.20	0.00	0.00	1.00	0.00	0.10
Reggae	0.00	0.00	0.00	0.11	0.10	0.00	0.00	0.00	1.00	0.00
Rock	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.40

図 6.16 ジャンル毎の分類精度：パターン 3

ラベル 予測	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	0.60	0.00	0.30	0.00	0.00	0.11	0.00	0.00	0.44	0.00
Classical	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Country	0.00	0.00	0.40	0.11	0.00	0.00	0.00	0.00	0.00	0.10
Disco	0.00	0.00	0.00	0.56	0.00	0.00	0.00	0.00	0.00	0.00
Hiphop	0.00	0.00	0.00	0.00	0.70	0.00	0.00	0.00	0.00	0.00
Jazz	0.00	0.00	0.00	0.00	0.00	0.78	0.00	0.00	0.00	0.10
Metal	0.00	0.00	0.00	0.00	0.00	0.00	0.89	0.00	0.00	0.00
Pop	0.00	0.00	0.00	0.11	0.20	0.00	0.00	0.78	0.00	0.00
Reggae	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00
Rock	0.40	0.00	0.30	0.22	0.10	0.11	0.11	0.22	0.11	0.80

図 6.17 ジャンル毎の分類精度：パターン 4

ラベル 予測	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Classical	0.00	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Country	0.20	0.10	0.90	0.00	0.00	0.22	0.00	0.00	0.00	0.00
Disco	0.00	0.00	0.00	0.78	0.00	0.00	0.00	0.00	0.00	0.00
Hiphop	0.00	0.00	0.00	0.00	0.70	0.00	0.00	0.00	0.11	0.00
Jazz	0.00	0.00	0.00	0.00	0.00	0.78	0.00	0.00	0.00	0.00
Metal	0.20	0.00	0.00	0.00	0.00	0.00	0.89	0.00	0.00	0.10
Pop	0.00	0.10	0.10	0.22	0.30	0.00	0.00	1.00	0.22	0.90
Reggae	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.67	0.00
Rock	0.10	0.10	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00

図 6.18 ジャンル毎の分類精度：パターン 5

ラベル 予測	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	0.40	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00
Classical	0.00	1.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00
Country	0.00	0.00	0.90	0.00	0.00	0.00	0.00	0.11	0.00	0.10
Disco	0.00	0.00	0.00	0.80	0.00	0.00	0.11	0.11	0.22	0.00
Hiphop	0.00	0.00	0.00	0.10	0.80	0.00	0.00	0.11	0.00	0.00
Jazz	0.00	0.00	0.00	0.00	0.00	0.78	0.00	0.00	0.00	0.10
Metal	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	0.00
Pop	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.10
Reggae	0.00	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.67	0.00
Rock	0.60	0.00	0.10	0.10	0.00	0.00	0.44	0.00	0.11	0.70

図 6.19 ジャンル毎の分類精度：パターン 6

ラベル 予測	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.00
Classical	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Country	0.00	0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.20
Disco	0.10	0.00	0.00	0.60	0.00	0.00	0.00	0.22	0.11	0.10
Hiphop	0.00	0.00	0.00	0.20	1.00	0.00	0.00	0.22	0.11	0.00
Jazz	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.00	0.00	0.00
Metal	0.00	0.00	0.10	0.20	0.00	0.11	1.00	0.00	0.00	0.30
Pop	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.44	0.11	0.00
Reggae	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.67	0.00
Rock	0.10	0.00	0.10	0.00	0.00	0.11	0.00	0.00	0.00	0.40

図 6.20 ジャンル毎の分類精度：パターン 7

予測ラベル	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.00
Classical	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Country	0.10	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.11	0.10
Disco	0.00	0.00	0.00	0.80	0.00	0.00	0.00	0.00	0.11	0.10
Hiphop	0.00	0.00	0.00	0.10	0.90	0.00	0.00	0.00	0.22	0.00
Jazz	0.10	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Metal	0.10	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Pop	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.89	0.22	0.00
Reggae	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.22	0.00
Rock	0.20	0.00	0.00	0.10	0.10	0.00	0.00	0.11	0.00	0.80

図 6.21 ジャンル毎の分類精度：パターン 8

予測ラベル	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.10
Classical	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Country	0.10	0.00	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Disco	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.11	0.00
Hiphop	0.00	0.00	0.00	0.00	1.00	0.00	0.10	0.00	0.00	0.00
Jazz	0.20	0.00	0.00	0.00	0.00	0.89	0.00	0.00	0.00	0.00
Metal	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.00	0.00	0.00
Pop	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Reggae	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.67	0.00
Rock	0.00	0.00	0.10	0.00	0.00	0.11	0.30	0.00	0.11	0.90

図 6.22 ジャンル毎の分類精度：パターン 9

予測\ラベル	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	0.72	0.02	0.03	0.00	0.00	0.03	0.00	0.01	0.07	0.04
Classical	0.00	0.95	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00
Country	0.05	0.01	0.76	0.02	0.00	0.02	0.01	0.05	0.02	0.12
Disco	0.01	0.00	0.02	0.74	0.02	0.00	0.01	0.03	0.08	0.03
Hiphop	0.00	0.00	0.00	0.05	0.83	0.00	0.01	0.03	0.06	0.02
Jazz	0.04	0.00	0.04	0.00	0.00	0.86	0.00	0.00	0.00	0.04
Metal	0.03	0.00	0.01	0.03	0.01	0.01	0.85	0.00	0.00	0.08
Pop	0.00	0.01	0.06	0.07	0.07	0.01	0.01	0.81	0.07	0.11
Reggae	0.00	0.00	0.00	0.01	0.04	0.00	0.00	0.00	0.66	0.00
Rock	0.15	0.01	0.08	0.06	0.03	0.03	0.11	0.04	0.04	0.56

図 6.23 平均の分類精度

図 6.23 における提案手法によるモデルの分類精度と、図 4.3 における Mingewn らによるモデルの分類精度を比較すると、極端に低かった Country と Rock の分類精度が向上していることがわかる。また、全体的な分類精度も 77.4% となり向上した。図 6.13～図 6.22 から、誤分類するジャンルがモデルによって変化するが、特に Rock のジャンルが全体のジャンルにわたって誤分類されやすいということが分かる。すなわち Rock というジャンルは他のジャンルに共通する特徴を持ちやすいということが考えられる。

## 6.2 生成器モデルの評価

### 6.2.1 学習毎の損失

5.3 節において、学習毎の Discriminator が output する wasserstein 距離と、相互情報量を推定し最大化する NN の損失 (mutual\_loss) を学習事にプロットしたものを図 6.24 に示す。横軸は学習回数 (epoch)，縦軸はその時の mutual\_loss または wasserstein 距離を表している。

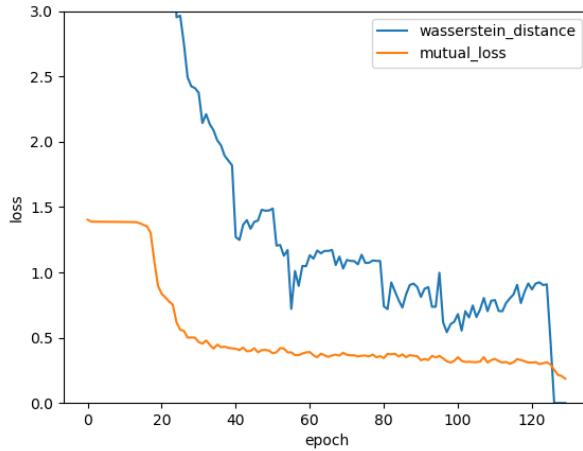


図 6.24 学習毎の wasserstein 距離と NN の損失

図 6.24 より、Discriminator が output する wasserstein 距離が下がっていくことが読み取れる。wasserstein 距離は生成データとオリジナルデータの分布間距離を示しているため、生成データがオリジナルのスペクトログラムに近づいているということがわかる。しかし、epoch が 125 を超えたあたりから wasserstein 距離が 0 となり勾配が消失し、学習が失敗していることが読み取れる。そのため、wasserstein 距離の学習が収束傾向であり、学習が失敗する寸前の epoch が 120 回目のモデルで学習を止めることが最善であると考えられる。一方 mutual\_loss は式 (2.22) を最小値問題としたときの目的関数の値を示している。mutual\_loss が減少していることから、Generator の入力ランダムノイズ  $z$  と、 $z$  から得られた生成データ  $Y$  を CNN に入力した時に得られる特徴マップ  $m$  との間で相互情報量が推定され、さらに最大化されているということがわかる。よって入力ランダムノイズ  $z$  と CNN のジャンル出力との間に従属関係があると言える。

### 6.2.2 生成されるスペクトログラム

学習済 Generator が出力するスペクトログラムとそれらを CNN で分類した結果の一例を図 6.25～図 6.34 に示す。

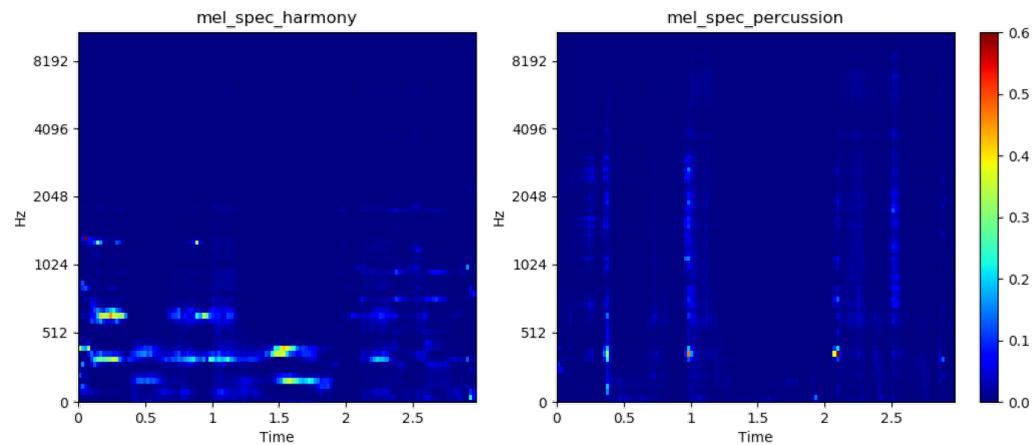


図 6.25 Blues と分類した生成スペクトログラム

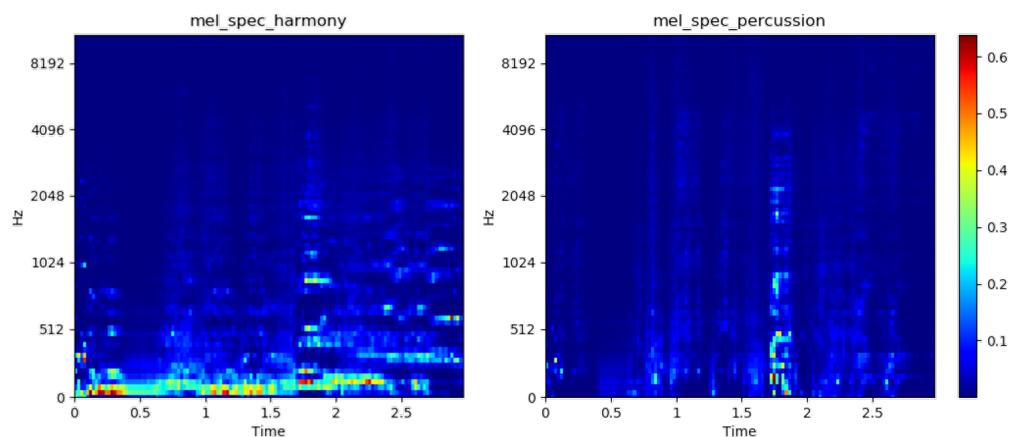


図 6.26 Classical と分類した生成スペクトログラム

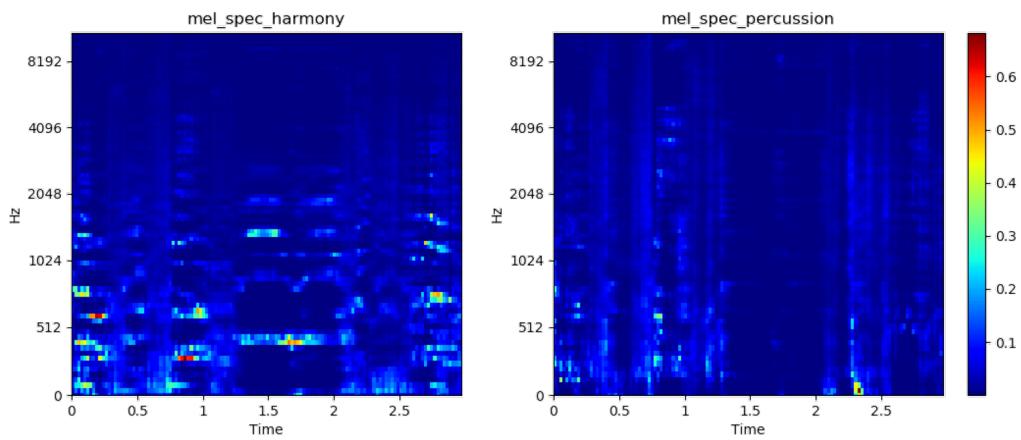


図 6.27 Country と分類した生成スペクトログラム

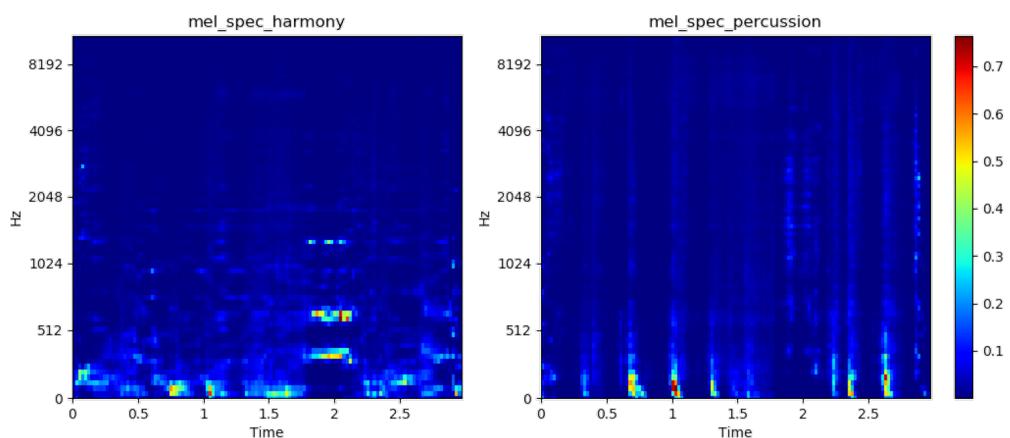


図 6.28 Disco と分類した生成スペクトログラム

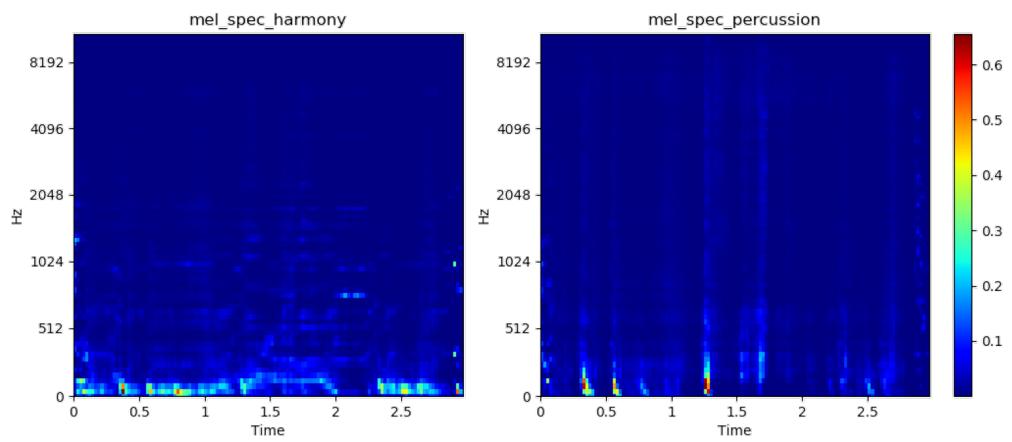


図 6.29 Hiphop と分類した生成スペクトログラム

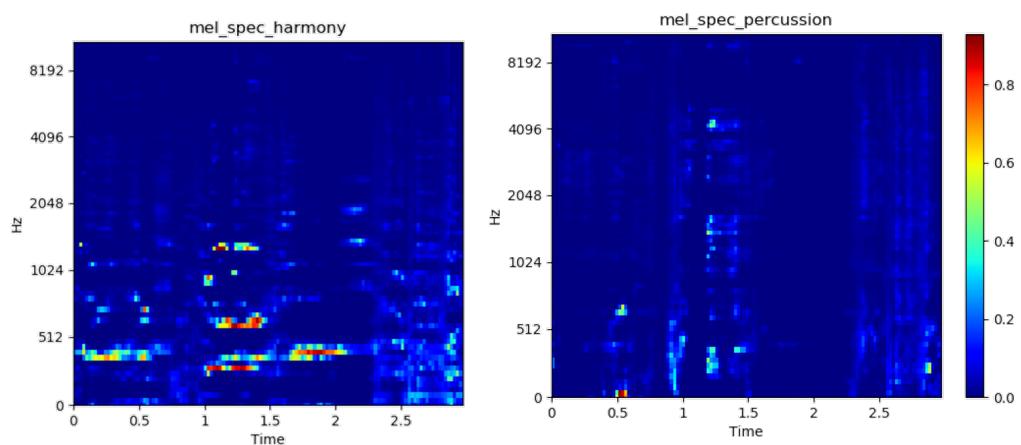


図 6.30 Jazz と分類した生成スペクトログラム

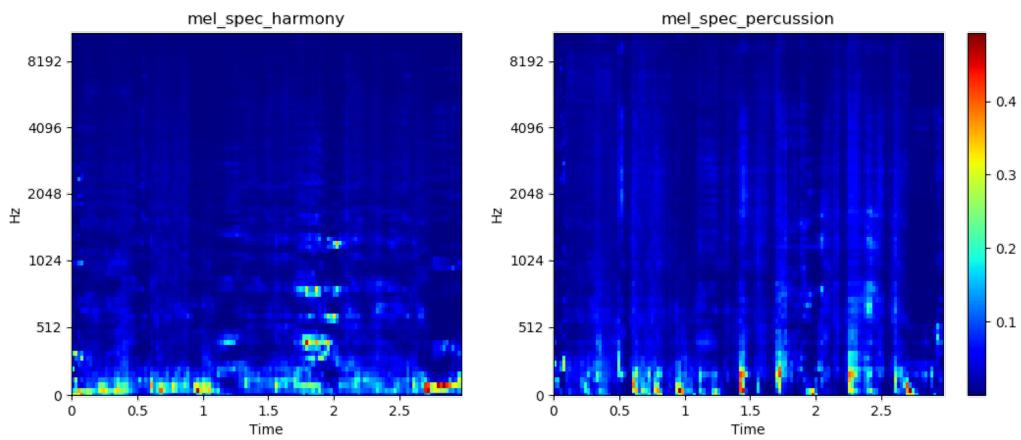


図 6.31 Metal と分類した生成スペクトログラム

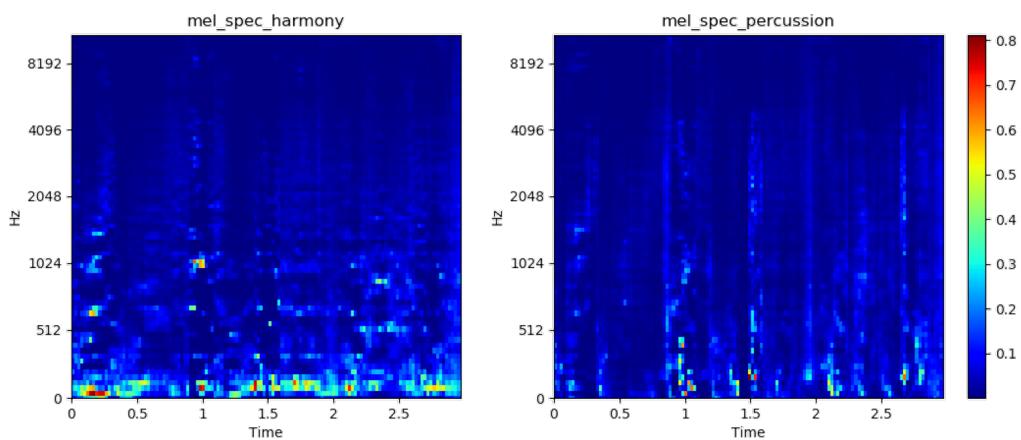


図 6.32 Pop と分類した生成スペクトログラム

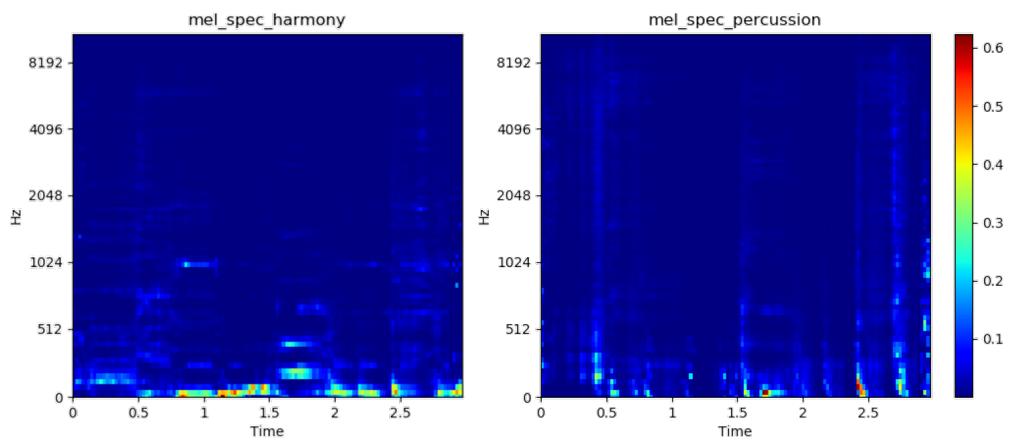


図 6.33 Reggae と分類した生成スペクトログラム

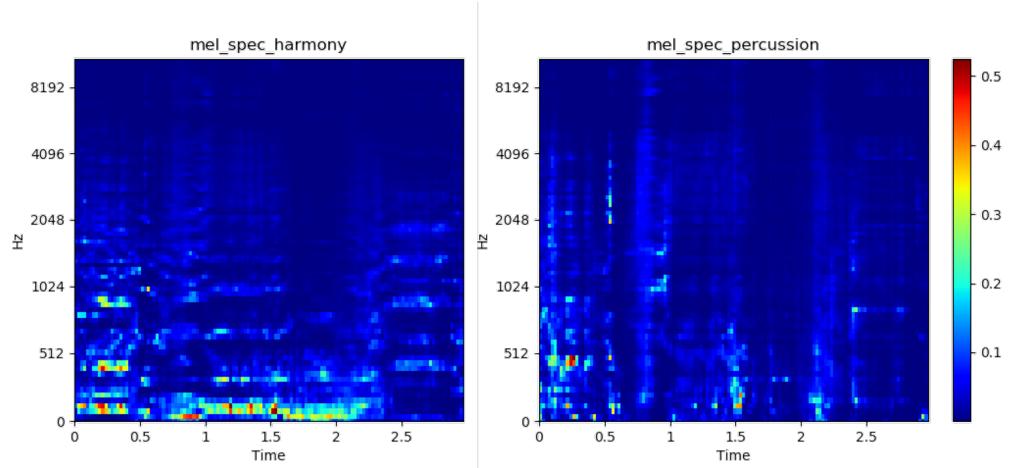


図 6.34 Rock と分類した生成スペクトログラム

図 6.28 と図 6.29 より, Disco と Hip-hop においてはパーカッション成分で低周波に強めのビートが, 特徴として表れやすいということが分かる. これは Disco や Hip-hop などのダンス系の音楽は, バスドラムの音を他の音より強めに出す傾向があることと一致している. また, Metalにおいては, ハーモニー成分とパーカッション成分の両方のにおいて, 強調される周波数が全帯域に広がりやすい傾向があることも読み取れる. これはギターの音を歪ませることにより多くの倍音が含まれるといった楽曲が多いためであると考えられる.

### 6.2.3 生成スペクトログラムのジャンル確率分布

生成されるスペクトログラムのジャンルの偏りを調べるために, 一様乱数のシード値を変化させながら, バッチサイズ 128 のスペクトログラムを生成していく. これらを学習済み CNN で分類していく, 最後にデータ総数で除算する. これにより生成されるスペクトログラムのジャンルの確率分布を求める. ジャンル毎の確率分布を表 6.1 に示す. 表 6.1 より, 生成されるスペクトログラムのジャンルに偏りがあることが分かる. これは, Generator がほぼ同一のデータしか出力しなくなる Mode Collapse という現象に陥っているため, 数ジャンルにだけ特化した生成モデルとなっている. それぞれのジャンルを均一に生成するためにはさらなる工夫が必要であると考えられる.

表 6.1 ジャンル毎の生成確率分布

ジャンル	確率分布 (%)
Blues	16.97
Country	0.85
Classical	17.56
Disco	13.75
Hip-hop	22.82
Jazz	18.13
Metal	3.67
Pop	0.31
Reggae	4.46
Rock	4.79

### 6.3 2次元ジャンルマップの検証

図 6.35 のように、2次元ジャンルマップ空間において、矢印の方向でジャンル境界を跨ぐように座標をずらしていく、そのときのジャンル出力確率の変化をグラフにしたものを見た。図 6.36 は、横軸にマップの座標、縦軸にその時のジャンル毎の確率を表している。

次に  $z_1 = -0.57$  で値を固定し、別の次元の従属ベクトル  $z_6$  を取り出し、 $(z_0, z_6)$  に関する2次元ジャンルマップ空間を作成したものを図 6.37 に示す。また  $z_0 = 0.57$  の軸で変化させたときのマップと出力確率の変化のグラフを図 6.38 に示す。

図 6.36 よりジャンル同士の出力確率が連続変化していることから、生成されるスペクトログラムはジャンルの特徴となるものが連続変化をしていると言える。また、確率の交差する点がジャンル境界となるため、その時に生成されるスペクトログラムがジャンルの中間データであると考えられる。図 6.35 と図 6.36 から、マップの座標がジャンルの境界面に近づくにつれ、両者の確率がトレードオフに変化していく、境界上で確率がイーブンなる。すなわちマップはジャンル間に対する距離的な領域空間を表していると考えらえる。

一方、マップの座標  $(z_0, z_1) = (0.57, -0.57)$  付近の Disco の領域において、Metal の確率が上がっていることについて考える。一見、マップには Metal ジャンルの領域は示されていないが、 $z_1 = -0.57$  で値を固定した図 6.37 から、別次元のマップで Metal の領域が現れていることがわかる。さらに図 6.38 からもわかるように、 $(z_0, z_1) = (-0.57, 0.57)$  で値を固定し  $z_6$  を動かした時、Disco と Metal の確率がトレードオフに変化していることからも、Disco と Metal の距離空間的も近いことが示されている。

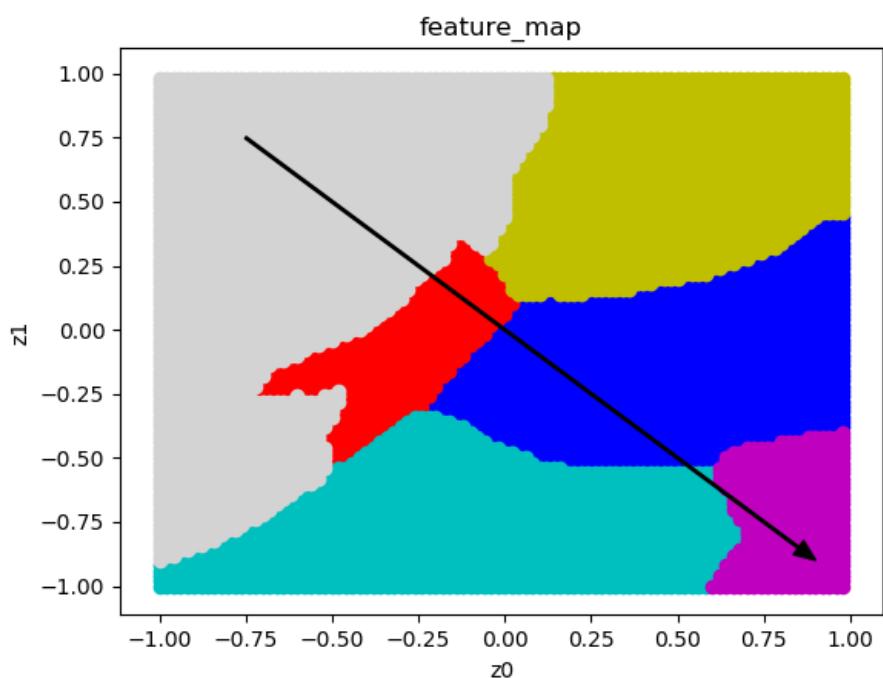


図 6.35 ジャンルデータの変化

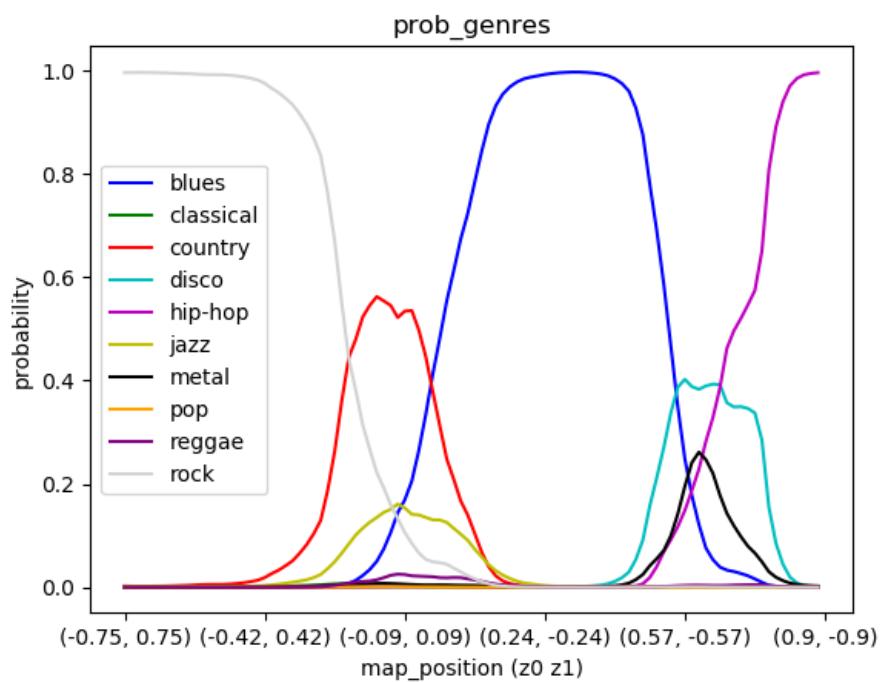


図 6.36 ジャンル毎の確率変化

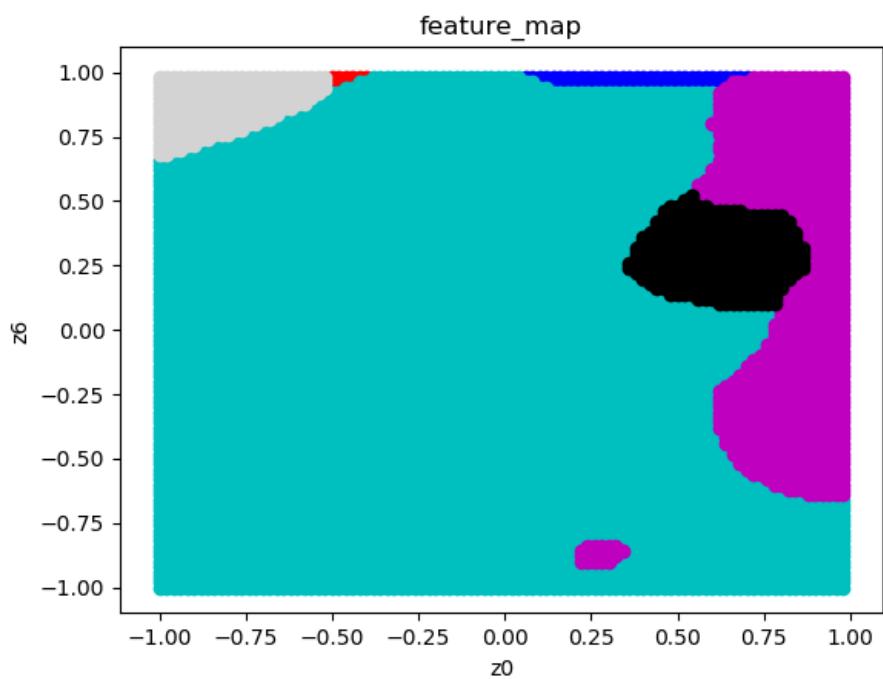


図 6.37  $z_1 = -0.57$  における  $(z_0, z_6)$  のジャンルマップ

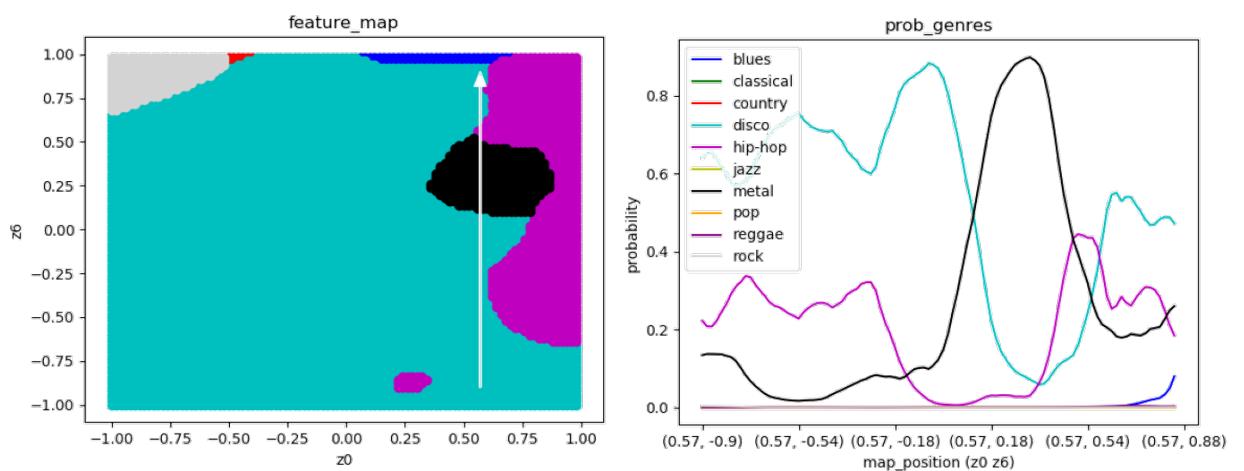


図 6.38  $z_0 = 0.57$  におけるジャンル変化

## 第7章 おわりに

本研究では、特徴量を自動で抽出する深層学習を用いて楽曲ジャンルの分類を行った。さらに学習して得られた分類モデルから、楽曲ジャンルの境界となる部分の可視化する手法を提案した。

楽曲データとしては 10 ジャンルを持つ GTZAN データセットを用いて評価を行った。このとき、提案手法では重複のあるデータの削除を行うことでより精度評価の妥当性を確保した。

分類モデルを構築する際には、分類精度で高い評価を受けている CNN を用いた。ここでの入力をパーカッション成分とハーモニー成分に分けたメル周波数スペクトログラムを正規化したデータを用いることにより、従来よりも分類精度を向上させることができた。さらに 10 分割交差検証を行ったところ、学習データの組み合わせによって精度に偏りが出やすいという事が分かった。

次に構築した分類モデルと GAN と組み合わせることにより、ノイズベクトルからメル周波数スペクトログラムの生成を行った。生成されるスペクトログラムの音楽ジャンルにおいて、Disco と Hip-hop は低周波に強めの一定のビートが特徴として表れやすいことが確認できた。また、生成されるスペクトログラムは連続変化が可能なため、分類モデルのジャンル出力確率も連続で変化することができるモデルとなった。

さらに GAN の入力である 2 次のノイズベクトルの値とジャンル分類結果の関係を 2 次元ジャンルマップ空間に示した。マップ内の値を連続変化させたときの出力確率の変化をグラフで表したところ、ジャンルの確率変化が連続であることを確認でき、ジャンル境界となる部分の可視化を可能にし、提案手法のジャンル分類器の評価にも用いた。

今後の課題としては、連続変化するジャンル確率とスペクトログラムの関係性の分析を行っていきたいと考えている。

## 謝辞

本論文を執筆するにあたり、多大なるご指導ご協力を頂きました、新潟大学大学院自然科学研究科の林隆史教授に心より感謝いたします。また本研究の遂行に際し、様々にご指導、ご助言を頂いた、研究室の大学院修士課程学生並びに学部学生の皆様に深謝いたします。

# 参考文献

- 1) Mingwen Dong, "Convolutional Neural Network Achieves Human-level Accuracy in Music Genre Classification," Feb. 27, 2018.
- 2) Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, "Grad-CAM Visual Explanations from Deep Networks via Gradient-based Localizations," March 21 2017.
- 3) Diederik P. Kingma, Jimmy Lei Ba, "ADAM: A Method for Stochastic Optimizer," Jan. 30, 2017
- 4) "畳み込みニューラルネットワーク," <https://ml4a.github.io/ml4a/jp/convnets/>, 参照 Dec. 12. 2019.
- 5) "GAN と損失関数の計算についてまとめた," <https://qiita.com/kzkadc/items/f49718dc8aedbe8a1bee>, 参照 Dec. 12. 2019.
- 6) Martin Arjovsky, Soumith Chintala, Leon Bottou, "Wasserstein GAN," Dec. 6, 2017.
- 7) Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, "Improved Training of Wasserstein GANs," Dec. 25, 2017
- 8) Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, R Devon Hjelm, "Mutual Information Neural Estimation," June 7, 2018.

- 9) R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, Yoshua Bengio, "Leaning Deep Representations by Mutual Information Estimation and Maximization," Feb. 22, 2019.
- 10) "メル尺度," <https://ja.wikipedia.org/wiki/メル尺度>, 参照 Dec. 12. 2019.
- 11) "MARSYAS, Music Analysis Retrieval and SYnthesis for Audio Signals, <http://marsyas.info/downloads/datasets.html>, Sep. 2018
- 12) Bob L. Sturm, "The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use," June 10, 2013
- 13) Derry FitzGerald, "Hamonic/Percussive Separation Using Median Filtering," Sep. 6-10, 2010, Austria.
- 14) Weibin Zhang, Wenkang Lei, Xiangmin Xu, Xiaofeng Xing, "Improved Music Genre Classification with Convolutional Neural Network," Sep. 8-12, 2016, SanFrancisco, USA.
- 15) "音楽のジャンル一覧," <https://ja.wikipedia.org/wiki/音楽のジャンル一覧>, 参照 Dec. 12. 2019

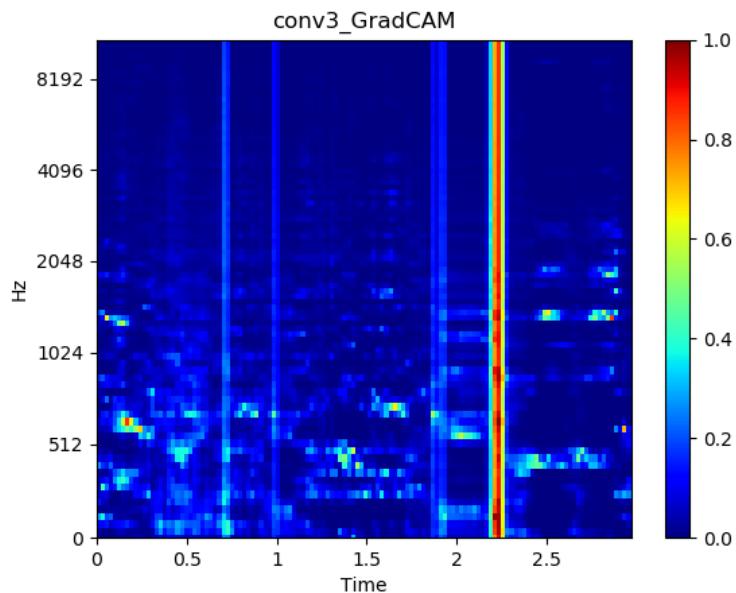
## 付 錄A プログラムのソースコード

モデル構築と2次元ジャンルマップ作成の際に用いたプログラムのソースコードを示す。

[https://github.com/hayashi-labratory/yamakawa\\_master\\_paper\\_sorcecode](https://github.com/hayashi-labratory/yamakawa_master_paper_sorcecode)

## 付録B Grad-CAMによる予備実験

提案手法によって5.2節で得られた学習済ジャンル分類器にGrad-CAMを適用する。この時に得られるヒートマップ化された入力のメル周波数スペクトログラムの一例を図B.1に示す。



図B.1 Grad-CAMによる入力スペクトログラムのヒートマップ化

図B.1は図5.4における3層目の畠み込み層の特徴マップの勾配を平均化した際のヒートマップ出力を表している。このとき、特徴マップの大きさが(1, 118)であるためヒートマップ出力が縦長になってしまっていることがわかる。このことから数値的にクラスの確率を上げるために入力データの重要な箇所を特定することはできるが、人間が解釈可能ではないヒートマップ出力となってしまっている。

## 付 錄C MNIST を用いた予備実験

手書き数字データセット MNIST を用いたときの二次元ジャンルマップを作成する。この時のジャンルマップの変化と確率変化を動画にしたものを作成する。

[https://github.com/hayashi-labratory/yamakawa\\_master\\_paper\\_sorcecode/tree/master/images](https://github.com/hayashi-labratory/yamakawa_master_paper_sorcecode/tree/master/images)