深層ネットワークによる作曲者の分類と作曲の試み

2019SS058 落合真嗣
指導教員：小市俊悟

1　はじめに

作曲者によって，作曲する楽曲に特徴や癖があると言う人もいるが，楽曲を聞いただけで，作曲者を当てることは難しい．本研究では，作曲者の特徴のあるなしについて，深層学習を用いて，作曲者ごとに楽曲が分類可能かを検証する．深層学習は画像の分類が得意であるので，画像に変換した楽曲ファイルを対象に分類を行う．楽曲データが限定される中で，深層学習がどれほどの精度を示すかに関心がある．また，敵対的生成ネットワークを用いて作曲者の特徴を学習し，別の作曲者の楽曲から学習した作曲者の楽曲に寄せた楽曲ファイルを作ることを試みる．

2　深層学習と敵対的生成ネットワーク

2.1　深層学習について

深層学習で用いる深層ネットワークとは，単純には，ニューラルネットワークの層を多層化したものである．昨今では，Python 等から利用できる深層学習用のフレームワークも充実し，その活用がますます進んでいる [1].

2.2　敵対的生成ネットワークについて

敵対的生成ネットワークとは GAN(Generative Adversarial Network) の訳である．GAN は 2 つのネットワークが競い合うように学習することで，最終的に訓練データには存在しなかったデータを新たに生成できることが特徴である．しかし，現状では，画像に関連するものが中心である．本研究は，そのような GAN を楽曲データに用いようとするものであり，GAN の新たな可能性を探るものである．

3　データとメルスペクトログラム

本研究では 10 名の作曲者の楽曲ファイルを使用する．各曲の長さは平均で 4 分程度であるが，それを 60 秒ごとに分割し，図 1 のようなメルスペクトログラムと呼ばれる画像として保存する．実験に使用する画像では縦横軸の目盛や大きさの指標は取り除いている．メルスペクトログラムでは，縦軸が周波数 (Hz)，横軸が時間 (秒)，色の濃さが音の大きさ (dB) を表し明るいほど音が大きく，暗いほど小さい．このようなメルスペクトログラムは短時間フーリエ変換により得られる．

4　研究の方法と結果

4.1　研究方法

文献 [2] を参考に，3 節で説明したデータに対して物体識別でよく用いられる次の 3 つの深層ネットワークを用いた．

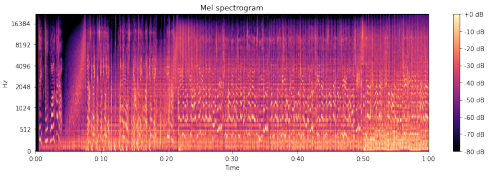

図 1　楽曲から得られたメルスペクトログラムの例

1. AlexNet
2. resnet
3. densenet

ただし，パラメータは学習により決定される．本研究では，学習用データに対する損失関数値を学習損失と呼び，判別結果の正解率を学習正解率と呼ぶ．同様に検証用データに対するものを検証損失と検証正解率とそれぞれ呼ぶ．一般に損失は小さいほど，正解率は高いほど良い．

4.2　結果

各深層ネットワークの学習における正解率の変化を表したのが図 2〜図 4 である．青線は学習正解率を示し，緑の点線は検証正解率を示す．まず 5 人の作曲者の分類をした．学習は 100 回に及ぶパラメータの更新からなる．

AlexNet　図 2 を見ると 0.2〜0.45 の範囲で両正解率は推移し値は低い．両正解率の差が小さいため，学習と検証であまり違いがない．

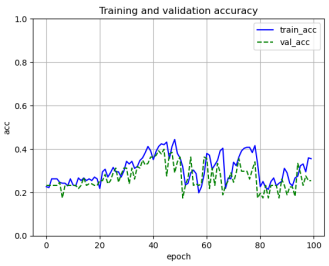

図 2　Alexnet の学習正解率と検証正解率

resnet　図 3 を見ると，学習正解率は上昇し 0.9 を超えている．検証正解率はパラメータ更新が約 40 回のところまでは上昇するが，それ以降は 0.4〜0.6 を推移する．学習用データでの結果は良いが，検証用データに対しては，あまり良いとは言えない．

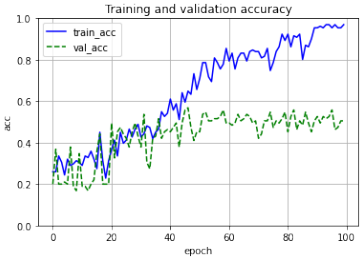

図 3　resnet の学習正解率と検証正解率

densenet　図 4 を見ると，0.9 を超える．検証正解率はほとんど横ばいで，0.3〜0.6 を推移する．各値の動きは resnet と似ているが，正解率ともに resnet と比べるとそれぞれ 0.1 程度低く，また学習と検証損失はやや大きい．

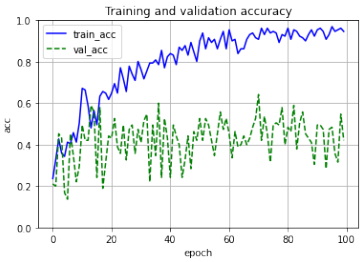

図 4　densenet の学習正解率と検証正解率

結果として，検証正解率を十分に上げることは難しかったが，学習正解率は，resnet や densenet であれば 1 に近い値まで出せる．したがって，学習データを十分に用意できれば，作曲者の分類は十分に可能ではないかと考える．

3 つのネットワークを比較したとき，最も検証損失が低く，検証正解率が高い値を出す resnet を使い，一度に分類する人数を増やしたところ，学習正解率の上がり方は 5 人のときと比べて緩やかであるが 1 に近づく．

5　GAN による作曲

5.1　研究方法

1. 楽曲ファイルを短時間フーリエ変換した結果を npy ファイルで保存し，楽曲をデータベースとする．
2. 作曲者 A と作曲者 B の楽曲の分類を学習し，作曲者 A と B を見分ける識別器 F を作成する．
3. GAN を応用した学習により，Generator G と Discriminator D を作成する．G は作曲者 A の楽曲を入力として，出力を行う．G の目的は，その出力を D と F の双方に作曲者 B の楽曲であると判定させることである．一方，D の目的は，入力データが，G が作成したものか，作曲者 B の楽曲であるかを正しく判定することである．F は，この G と D の学習においては，変化させない．
4. 学習が終了した後の G が生成した npy ファイルからメルスペクトログラムを作成する．
5. 作曲者の分類で作成した判別器が，手順 4 のメルスペクトログラムを正しく判定するかを検証する．判別器が正しく答えないようであれば，「機械」を騙す程度には，作曲することができたことになると考える．

5.2　結果

表 1　作曲者 B の楽曲と Generator G の出力に対する判別結果 (個数)

判別結果＼正解	作曲者 B	G の生成物	計
作曲者 B	89	8	97
G の生成物	9	21	30
計	98	29	127

表 1 より，G が生成した楽曲を作曲者 B の楽曲であると判別器に誤認させたのは，30 曲中 9 曲，割合で 0.3 であった．この結果は，判別器の検証正解率 0.7368 すなわち，誤認率 $1 - 0.7368 = 0.2632$ を踏まえるとやや高い程度なので，期待通りに誤認させることができたとは言い切れない結果となった．これを改善するのに，気づいた点を以下に記す．

- 識別器 F の検証正解率が 0.7 程度で頭打ちになる．分類と同様に識別器 F の精度を上げるために，十分な学習データが必要である．
- 手順 3 の GAN の実行では，特に初期段階において G が生成するデータは，素朴に考えると，作曲者 B の楽曲からは遠いと考えられるが，このようなデータは，手順 2 における識別器 F の学習には用いておらず，F がどのように識別するか不明である．しかし，F が作曲者 B の楽曲ではないと識別すべきものであると考えられるので，このようなデータも含めて，F を再学習させることが良いかもしれない．

6　おわりに

本研究では作曲者の特徴という言語化が難しいものについて深層学習を用いた分類を試みた．学習データを十分に用意できれば，分類できるような結果も得られたので，深層学習の新たな適用方法を探ることができたと考える．一方で，GAN により人間が「曲」と感じられるものまでを生成することは，画像以上に難しい．画像であれば，多少の変化に特に違和感を感じないものであるのに対して，楽曲は少しの変化でも不協を感じ得る．このような点を考慮できれば，より「曲」らしいデータを生成できるのではないかと考える．

参考文献

[1] PyTorch ニューラルネットワーク ハンドブック：株式会社秀和システム，東京，2019

[2] Platinum Data Blog: https://blog.brainpad.co.jp/entry/2018/04/17/143000