DCGANによるコレクションアイテムの日常利用できるデザイン生成

谷聖一 研究室　堀内 春城
Haruki Horiuchi

概要

本演習では，DCGAN を用いて普段着とコレクションウェアのデザインを生成するモデルを作成した．学習した生成モデルを用いて普段着とコレクションウェアのデザインを生成し，生成された普段着とコレクションウェアのデザインの中間をとることで日常利用できる服のデザインを試作した．

1　はじめに

報告者は服のデザインに興味があり，服をデザインしてみたいと考えたことがある．しかし，服のデザインに関する専門知識のない報告者には，デザインは難しい．そこで，デザインの自動生成を試みた．

本演習には，普段着とコレクションウェアの 2 種類の服を用いる．普段着は日常，家庭の中などで着ている衣服とする．コレクションウェアはパリ・コレクションやミラノ・コレクションなど，ファションショーで発表される服とする．

普段着はありきたりなデザインであるが，コレクションウェアは派手で着るのに抵抗があるデザインなので，両方を掛け合わせる事で，日常利用できるコレクションウェアのデザインを生成したい．

訓練データから学習し，新しいデータを生成するものを生成モデルという．生成モデルの優れた学習手法として Deep Convolutional Generative Adversarial Networks（DCGAN）が知られている．DCGAN はベクトルを用いてデータを生成し，[1] によれば，ベクトルの値に応じて画像を連続的に変化させることができる．

普段着とコレクションウェアのデザインを生成するモデルを機械学習技術を用いて構築する．生成された普段着とコレクションウェアの中間を生成できれば 2 つの要素を合わせ持つ目的のデザインが生成されるのではないかという問題意識のもと演習を行うことにした．

2　準備

この節では，生成モデル，GAN，DCGAN についての説明をする．

2.1　生成モデル

生成モデルとは，教師データから学習し，それらのデータと似たような新しいデータを生成するモデル．

2.2　GAN

GAN（Generative Adversarial Networks）とは，教師データから学習し，生成モデルをニューラルネットワークを用いて構築する手法である ([2])．Generator という新しいデータを生成するネットワークと Discriminator という Generator の生成した画像と教師画像を識別するネットワークを競わせながら学習させるアーキテクチャである．GAN のネットワーク構造を図 1 に示す．

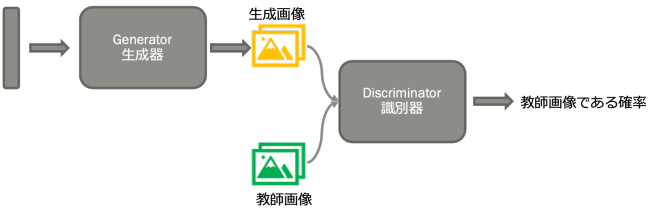


図 1: GAN のアーキテクチャ

Generator

一様分布からサンプリングした多次元ベクトルを入力として受け取り，教師画像と同じデータ，カラーモードの画像データを出力する．

Discriminator

入力として Generator の生成した画像または教師画像を受け取り，識別し，教師画像である確率を出力する．

Generator は Discriminator に生成画像を教師画像と誤認識させるようにデータを生成することを目指し，Dis-

criminator は Generator の生成画像と教師画像を誤認識しないように正しく識別することを目指す．このように 2 つのネットワークが相反する目的のもとに学習することから，敵対的生成ネットワークと呼ばれる．

2.3 DCGAN

DCGAN は GAN を元にしたネットワークで，GAN のニューラルネットワーク部分に畳み込みニューラルネットワークを用いたものである．DCGAN は一般的な GAN と比べて学習が安定しており，高解像度の画像が生成可能と言われている．GAN では具体的に明示されていなかったネットワーク構造について，言及しており，DCGAN では以下のようなネットワーク構造を提案している ([1])．

- プーリング層を Generator では転置畳み込み層，Discriminator では畳み込み層で置き換える
- 深い構造では全結合の隠れ層を除去する
- Generator と Discriminator で BatchNormalization を行う
- Generator の出力層以外のそうで ReLU 関数を活性化関数として使用し，出力層では Tanh 関数を使用
- Discriminator では全ての層で LeakyReLU 関数を活性化関数として使用

転置畳み込み層

入力をアップサンプリングしてから，畳み込み演算を行う．入力をアップサンプリングで徐々に大きくして，画像サイズに拡大していく．

Batch Normalization

学習の高速化や学習の安定性を向上させる手法 ([3])．ミニバッチ毎に入力を正規化する．入力 $B = \{x_1, \ldots, x_m\}$ を式 (1) により，正規化．

$$\widehat{x_i} \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{1}$$

ここで μ_B を平均，σ_B^2 を分散とする．

$$\mu_B \leftarrow \sum_{i=1}^{m} x_i$$

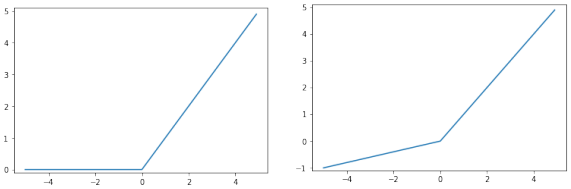
$$\sigma_B^2 \leftarrow \sum_{i=1}^{m} (x_i - \mu_B)^2$$

正規化した値を式 (2) で変換する．

$$y_i \leftarrow \gamma \widehat{x_i} + \beta \tag{2}$$

ReLU・LeakyReLU

出力層以外の層の活性化関数は Generator では図 2 の「ReLU」，Discriminator では図 2 の「LeakyReLU」を用いた．



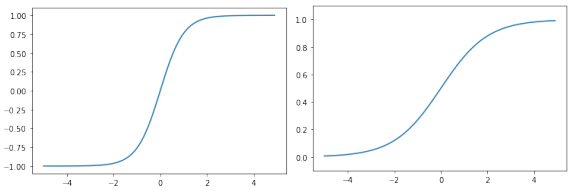
$$\begin{cases} x\,(x > 0) \\ 0\,(x \leq 0) \end{cases} \qquad \begin{cases} x\,(x > 0) \\ \alpha x\,(x \leq 0) \end{cases} \quad \alpha = 0.2$$

ReLU　　LeakyReLU

図 2: 出力層以外の活性化関数

tanh・sigmoid

出力層の活性化関数は Generator では図 3 の「tanh」，Discriminator では図 3 の「sigmoid」を用いた．



$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad h\,(x) = \frac{1}{1 + e^{-x}}$$

tanh　　sigmoid

図 3: 出力層の活性化関数

3 演習内容

本演習では普段着とコレクションウェアのデザインを生成するモデルを構築し，生成された普段着とコレクションウェアの中間を獲得する．
演習の手順を以下に示す．

1. 画像データの収集
2. 画像データの整形
3. デザインを生成するモデルの構築
4. 生成結果
5. 評価

以下に，各手順の詳細を示す.

3.1 画像データの収集

普段着の画像を 8 種類，コレクションウェアの画像を 50 種類収集した．画像収集は Excite 画像検索からスクレイピング で行った.

3.2 画像データの整形

画像データの選別

収集した普段着とコレクションウェアの画像を目視で正面を向いたトップスの画像だけに選別した.
以下のような画像は除去.

- パンツなどトップスでない画像
- 該当のデザインでない画像
- 画像のサイズが極端に小さい画像
- トップスが重なり合っている画像

トリミング

選別した画像に対し，トップスが画像の中心に来るように手作業でトリミングを行った.

画像データの前処理

「Pillow」([4]) を用いて画像データのカラーモードを「RGB」に変換し，画像サイズを「128 × 128」に統一した.

画像データ整形後の枚数

整形結果を表 1 に示す.

表 1: 整形結果

	収集枚数	整形後枚数	データセット
普段着	4000	747	500
コレクションウェア	10000	1452	500

学習のデータセット用により普段着っぽいデザイン，よりコレクションウェアっぽいデザインを選別し，各 500 枚に揃えた.

3.3 デザインを生成するモデルの構築

使用したライブラリを表 2 に示す.

表 2: 使用したライブラリ

ライブラリ	概要
Keras	深層学習ライブラリ
TensorFlow	計算ライブラリ
NumPy	データ分析
OpenCV	画像解析ライブラリ
Matplotlib	グラフ描画ライブラリ

デザインを生成するモデルは「128 × 128」の画像を生成する.
Generator と Discriminator のネットワーク構成を [1], [5] を参考に Keras[6] を用いて構築した.
Generator のネットワークの構成を表 3，Discriminator のネットワーク構成を表 4 に示す.

表 3: Generator のアーキテクチャ

操作	カーネル	ストライド	チャンネル数	Batch Normalization	活性化関数
入力：100	−	−	−	−	−
Dense	−	−	−	−	ReLU
Reshape	−	−	1024	○	−
Upsampuling2D	−	−	1024	−	−
Conv2D	3 × 3	1	512	○	ReLU
Upsampuling2D	−	−	512	−	−
Conv2D	3 × 3	1	256	○	ReLU
Upsampuling2D	−	−	256	−	−
Conv2D	3 × 3	1	128	○	ReLU
Conv2D	3 × 3	1	3	−	tanh

表 4: Discriminator のアーキテクチャ

操作	カーネル	ストライド	チャンネル数	Batch Normalization	活性化関数
入力：128 × 128 × 3	−	−	3	−	−
Conv2D	3 × 3	2	32	−	LeakyReLU
Conv2D	3 × 3	2	64	○	LeakyReLU
Conv2D	3 × 3	2	128	○	LeakyReLU
Conv2D	3 × 3	2	256	−	LeakyReLU
Flatten	−	−	1	−	−
Dense	−	1	1	−	sigmoid

3.4 生成結果

生成結果は，教師画像に普段着とコレクションウェアの 2 種類を用いているので，そのどちらかが生成されるはずである．生成結果は図 4 のようになった.

図 4: 生成画像

100 枚画像を生成し，目視による判断で普段着とコレクションウェアを識別したところ 38 枚が普段着，41 枚がコレクションウェアとなった．残りはうまく生成されず，服には見えないデザインとなった．

生成された普段着の集合とコレクションウェアの集合の中間を獲得する方法として，本演習では SVM により分類する超平面を獲得することを目指した．

本演習では SVM により普段着とコレクションウェアの集合を分類することはできたが，分類した超平面を取り出すツールを見つけることができなかった．そのため集合の中間ではなく，1 つの普段着と 1 つのコレクションウェアのここの中間を獲得した．その例を図 5 に示す．図 5 の左端が普段着，右端がコレクションウェアとなっており，中は 2 つの画像の遷移を表す画像である．

図 5: 1 つの普段着と 1 つのコレクションウェアの中間

3.5 評価

図 5 の生成結果から，1 つの普段着と 1 つのコレクションウェアの中間は自分が着てみたいと思えるデザインとなった．また，色や形なども，どちらの要素もあると感じた．

4 今後の課題

今後の課題は 4 点ある．1 点目は，SVM で分類した超平面の解析を行うことである．2 点目は，本演習ではデータセットの画像サイズは「128 × 128」で行ったが，さらに高画質の画像を用いることでより鮮明な画像の生成を行う．3 点目は，本演習では 2D のデザイン生成を行なったが，実際，デザインから服を作る時は 3D の方がわかりやすいので 3D でのデザイン生成を目指す．4 点目は，本演習では自分が着たいと思えるデザイン生成を行なったので，目視で判断した部分など，第三者に判断してもらうなどして，より一般的に普段着とコレクションウェアの中間となるデザインを生成する方法を考えることである．

参考文献

[1] Alec Radford, Luke Metz, Soumith Chintala
Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
arXiv preprint arXiv:1511.06434, 2015.

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio
Generative Adversarial Networks
arXiv:1406.2661,2014.

[3] Sergey Ioffe, Christian Szegedy
Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift
arXiv:1502.03167,2015.

[4] Pillow
http://pillow.readthedocs.io/en/latest/(参照:2020-02-06)

[5] GAN について概念から実装まで
https://qiita.com/taku-buntu/items/oo93a68bfae0b0ff879d(参照:2020-02-06)

[6] Keras
https://keras.io/ja/(参照:2020-02-06)