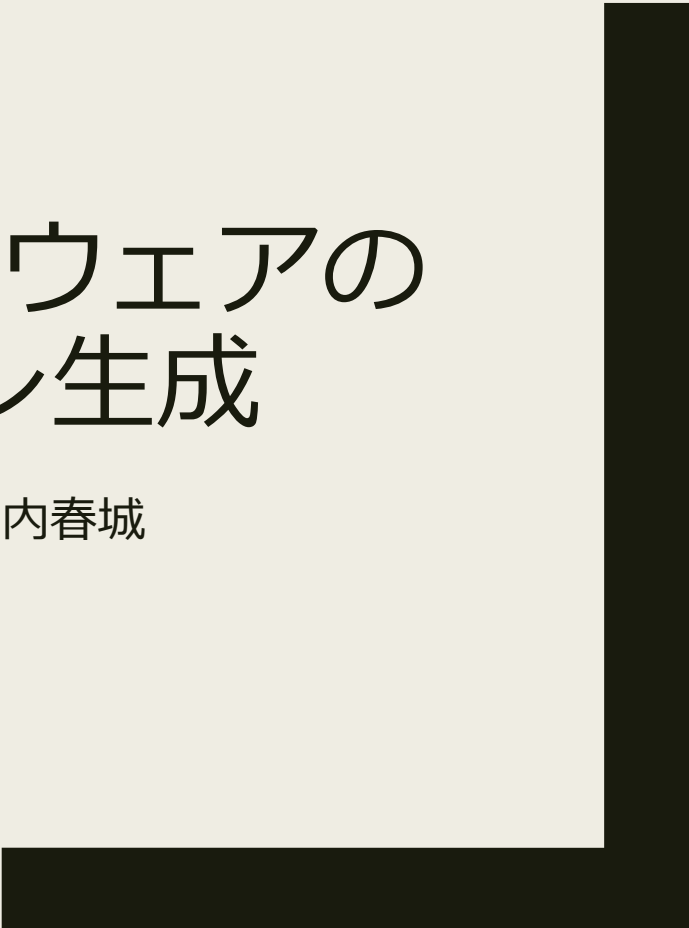




DCGANによるコレクションウェアの 日常利用できるデザイン生成

日本大学 文理学部 情報科学科 谷聖一研究室 堀内春城

目次

1. はじめに
 - 1-1 動機
 - 1-2 目的
2. 準備
 - 2-1 生成モデル
 - 2-2 GAN
 - 2-3 DCGAN
3. 演習内容
 - 3-1 画像データの収集
 - 3-2 画像データの整形
 - 3-3 デザインを生成するモデルの構築
 - 3-4 生成結果
4. まとめ

1. はじめに

1-1 動機

服が好きで、作ってみたいと思ったことがある

しかし服をデザインするのは難しい

そこで機械学習で服をデザインする

1-1 動機

普段着

日常、家庭の中などで着ている衣服



1-1 動機

コレクションウェア

パリ・コレクションやミラノ・コレクションなど
ファッションショーで発表される服を
コレクションウェアとする

1-2 目的 デザインの内容

➤ 普段着
ありきたりなデザイン

➤ コレクションウェア
かっこいい服だが、派手なので着るのに少し抵抗がある



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

1-2 目的

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

普段着とコレクションウェアの中間のデザインを生成



普段着



目的のデザイン



コレクションウェア

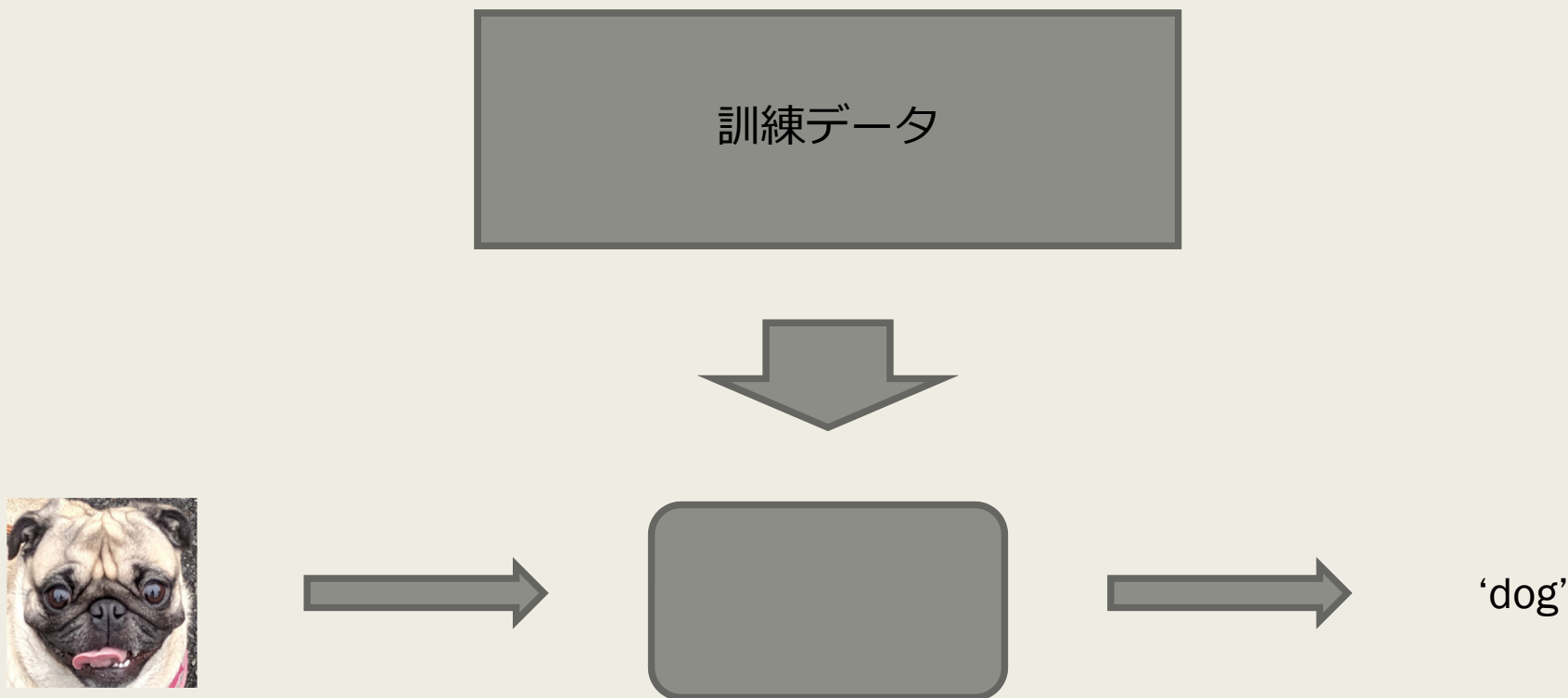
普段着とコレクションウェアの要素を合わせ持つ

目的のデザインが生成されるのではないかな

2. 準備

学習モデル（復習）

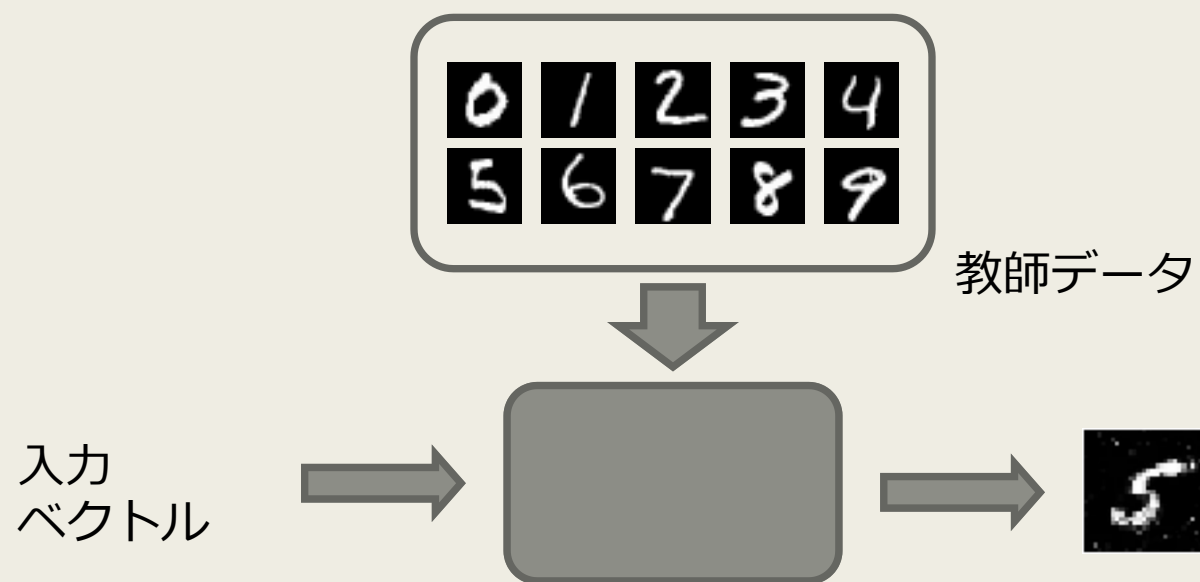
訓練データから学習し
入力を受け取り何かしらの評価・判定をして出力する



2-1 生成モデル

生成モデルとは

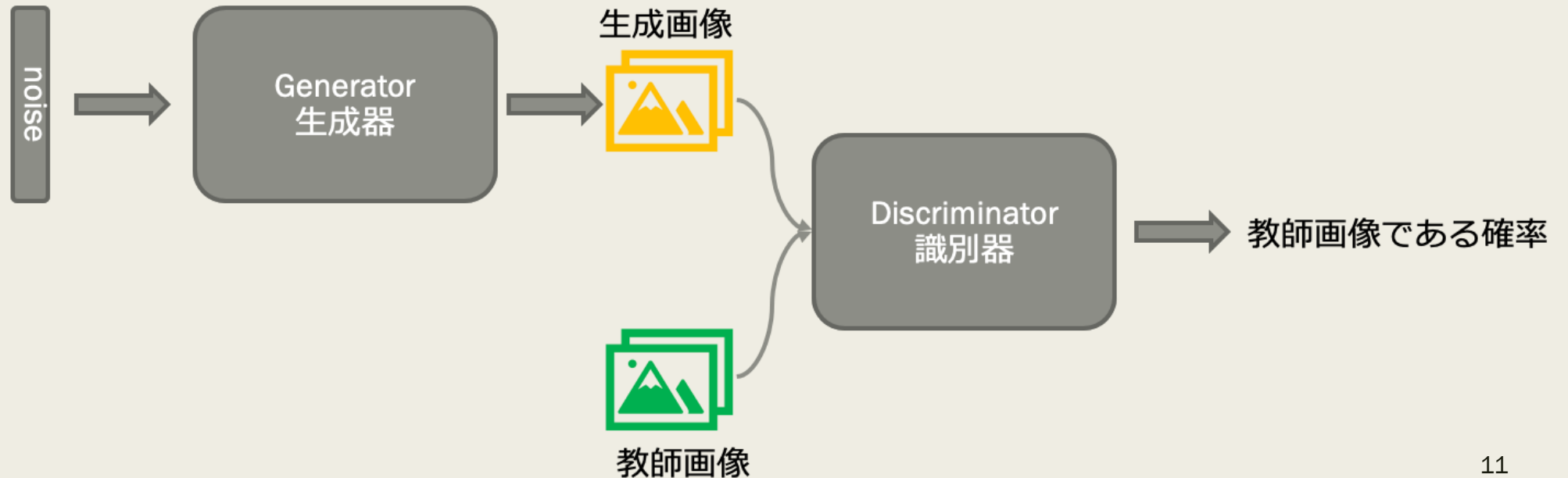
- 教師データから学習し、それらのデータと似たような新しいデータを生成するモデル



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で構築する手法
- 2つの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



2-2 GAN Generator

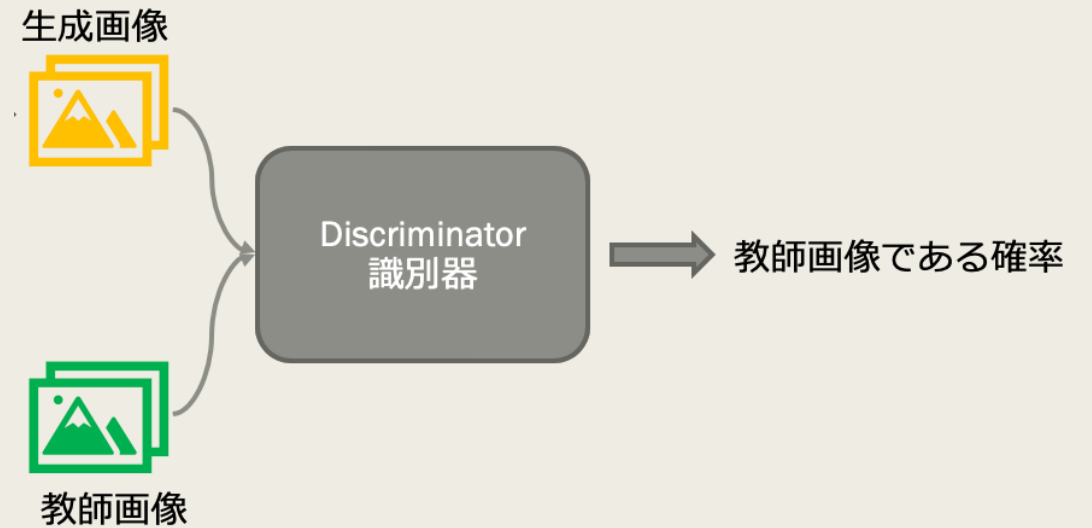


入力としてベクトル受け取り、画像を生成

入力：一様分布からサンプリングした多次元ベクトル

出力：教師画像と同じサイズ、カラーモードの画像データ

2-2 GAN Discriminator

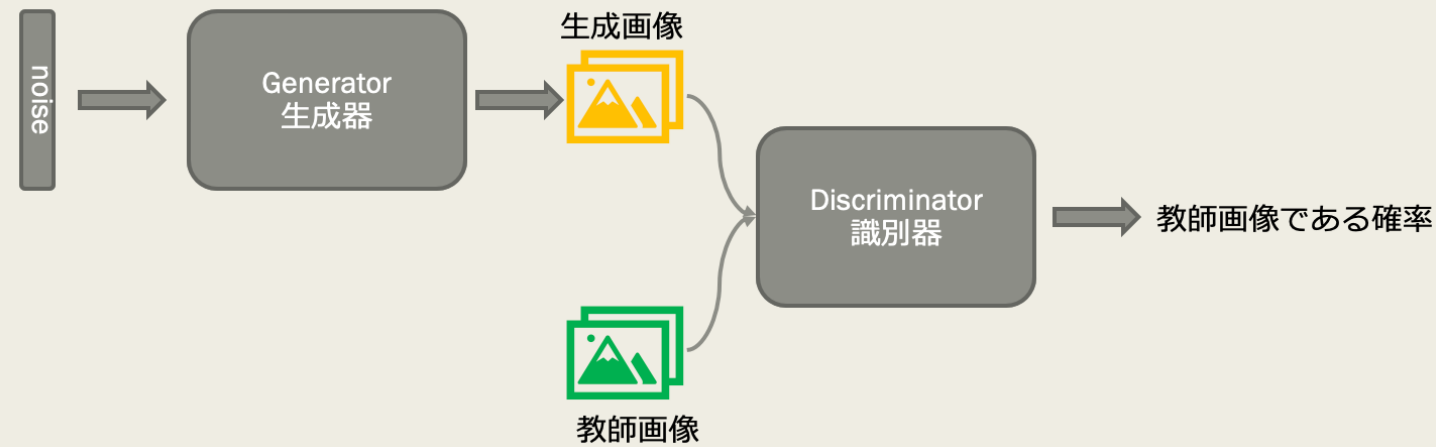


入力データが教師画像かGeneratorの生成画像かを識別

入力：Generatorが生成した画像または教師画像

出力：教師画像である確率

2-2 GAN



➤ Generator
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

➤ Discriminator
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

このような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

2-2 GAN

GANの学習方法

2つのステップを
バッチ毎に繰り返す

1. Discriminator の学習
2. Generator の学習



2-2 GAN GANの学習方法

2つのステップを
バッチ毎に繰り返す

1. Discriminator の学習
2. Generator の学習



生成画像は0、教師画像は1と出力するように学習

2-2 GAN GANの学習方法

2つのステップを
バッチ毎に繰り返す

1. Discriminator の学習
2. Generator の学習



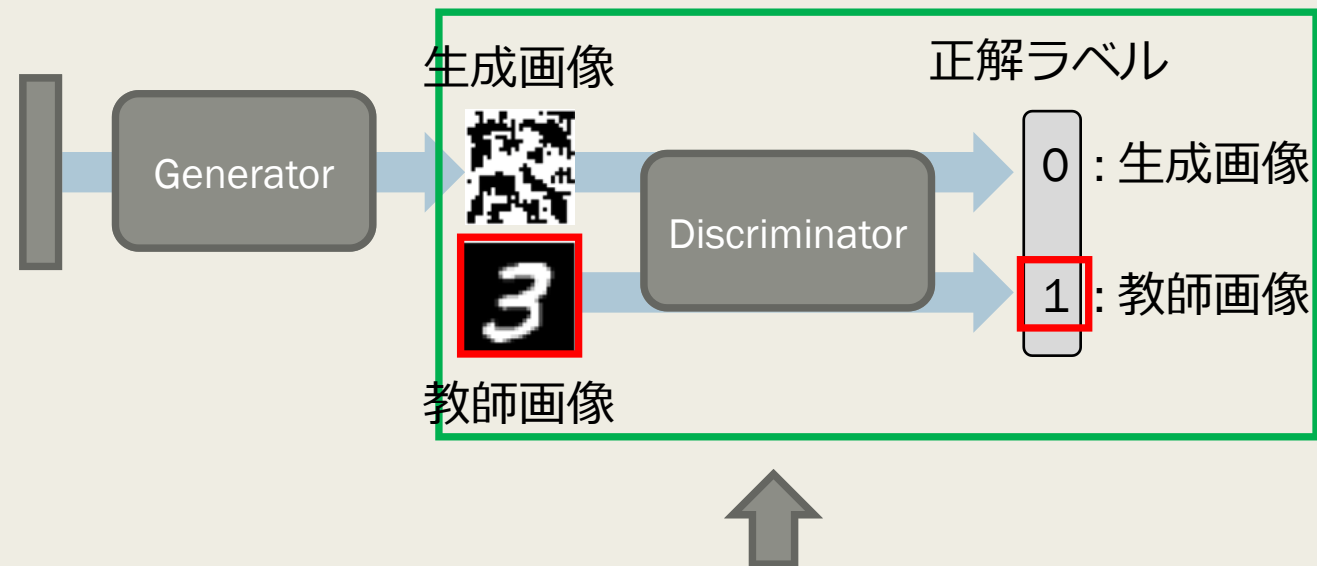
生成画像は0、教師画像は1と出力するように学習

2-2 GAN

GANの学習方法

2つのステップを
バッチ毎に繰り返す

1. Discriminator の学習
2. Generator の学習



生成画像は0、教師画像は1と出力するように学習

2-2 GAN GANの学習方法

2つのステップを
バッチ毎に繰り返す

1. Discriminator の学習
2. **Generator** の学習



Discriminatorの出力が1（教師画像）となるように学習

2-2 GAN GANの学習方法

2つのステップを
バッチ毎に繰り返す

1. Discriminator の学習
2. **Generator** の学習



Discriminatorの出力が1（教師画像）となるように学習

2-2 GAN

GANの学習方法

2つのステップを
バッチ毎に繰り返す

1. Discriminator の学習
2. Generator の学習

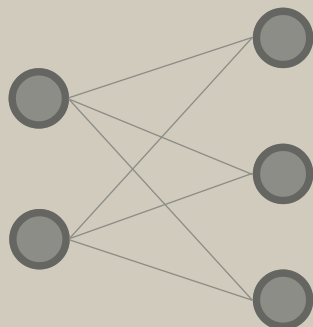


2-2 GAN

GANのネットワーク構造

GANの多くの実装では、
全結合ニューラルネットワークが用いられていた

全結合ニューラルネットワークとは
前層と後層のニューロンが全て接続されている層のこと



2-3 DCGAN

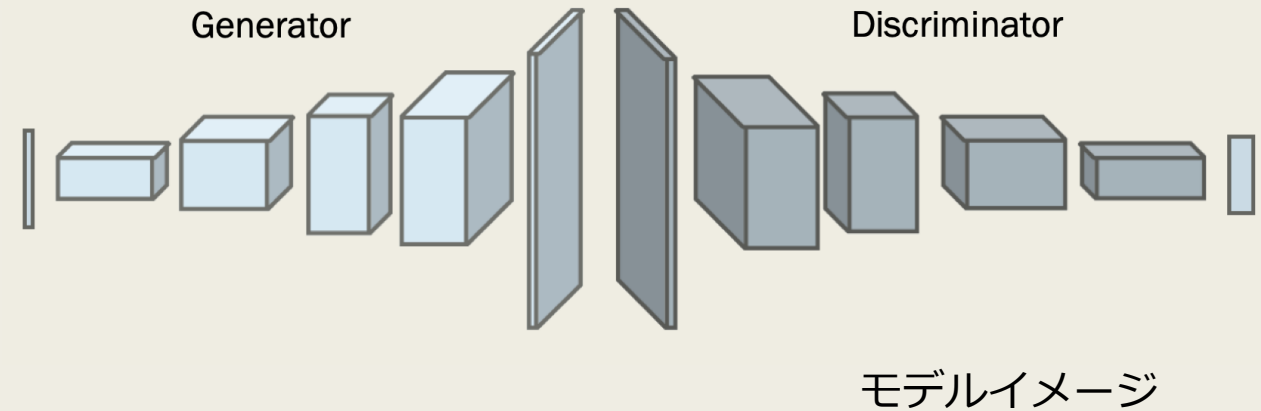
DCGAN (Deep Convolutional GAN) とは

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 (CNN) を用いたGAN

一般的なGANと比べて学習が安定しており、
高解像度の画像の生成が可能とされている

2-3 DCGAN

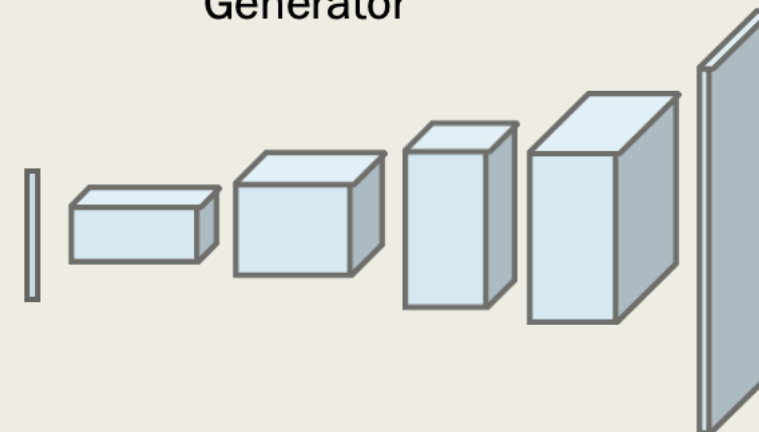
安定したDCGANの構造



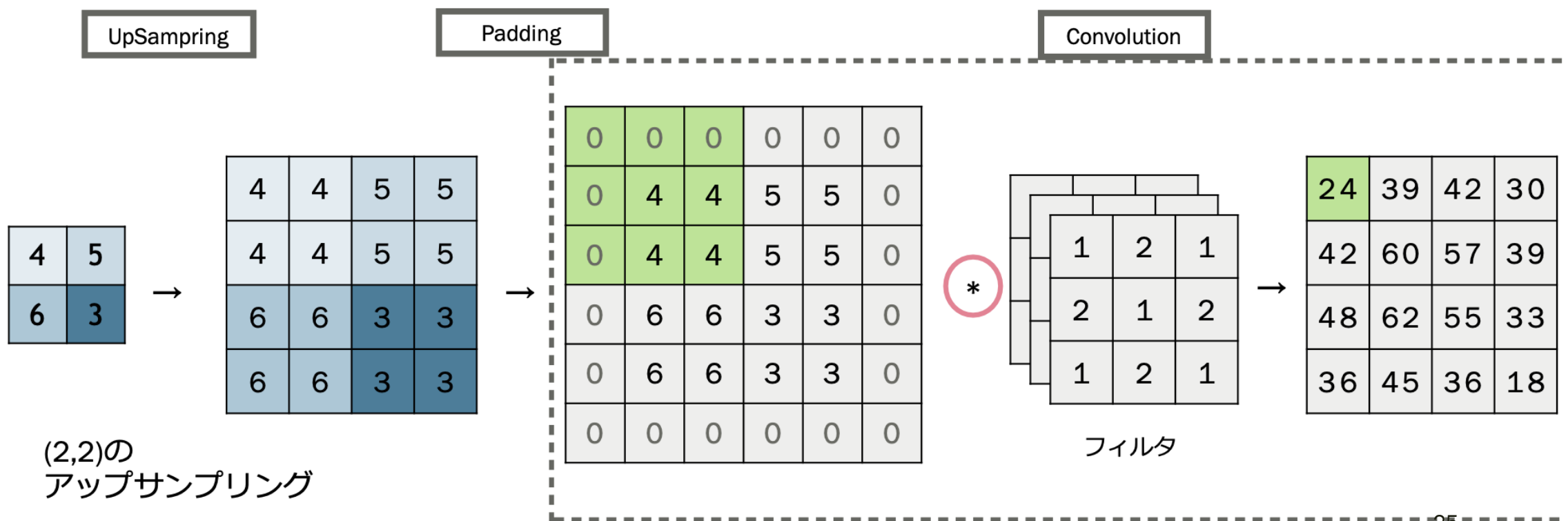
- CNNのプーリング層を下記で置き換える
 - Generator（生成器）：転置畳み込み層
 - Discriminator（識別器）：畳み込み層
- Batch Normalizationを使用
- Generatorの出力層以外の層で ReLU 関数を活性化関数として使用
出力層ではTanh関数を使用
- Discriminatorではすべての層でLeakyReLU関数を活性化関数として使用

2-3 DCGAN 転置畳み込み層

Generator



アップサンプリングと畳み込み層を組み合わせ
転置畳み込みを実装する



2-3 DCGAN

Batch Normalization

ミニバッチ毎に正規化

入力 $B = \{x_1, \dots, x_m\}$ を以下の式で正規化

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

ここで μ_B を平均、 σ_B^2 を分散とする

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

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

正規化した値を以下の式で変換する

$$y_i \leftarrow \gamma \hat{x}_i + \beta$$

γ と β は学習パラメータ

Batch Normalizationの効果

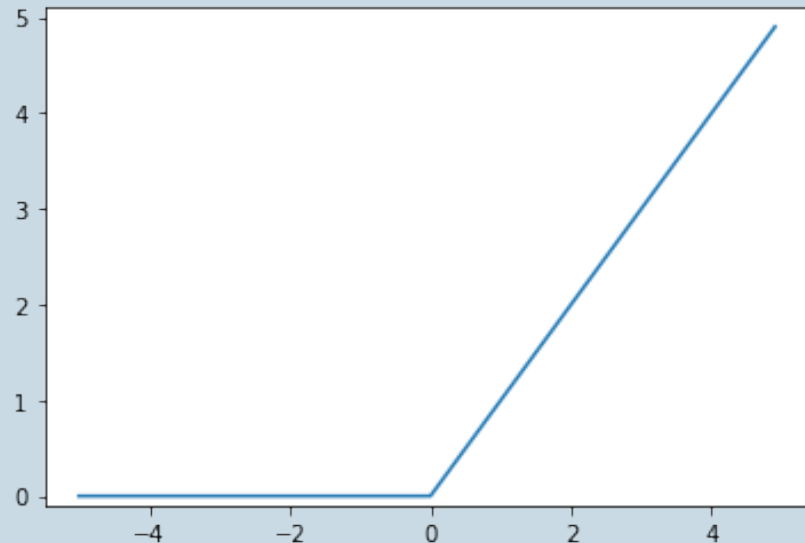
- 学習の高速化
- 過学習の抑制

2-3 DCGAN

ReLU • LeakyReLU

ReLU

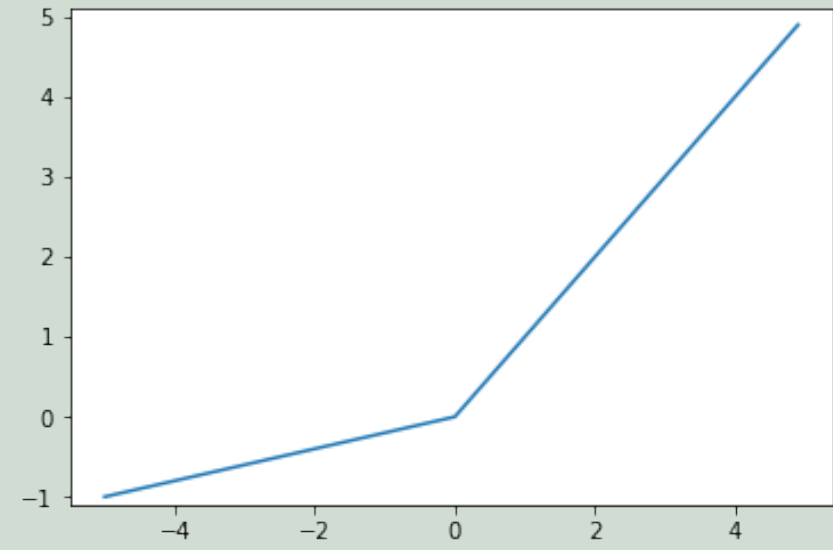
$$\begin{cases} x = x(x > 0) \\ x = 0(x \leq 0) \end{cases}$$



LeakyReLU

$$\begin{cases} x = x(x \geq 0) \\ x = \alpha x(x < 0) \end{cases}$$

$\alpha = 0.2$

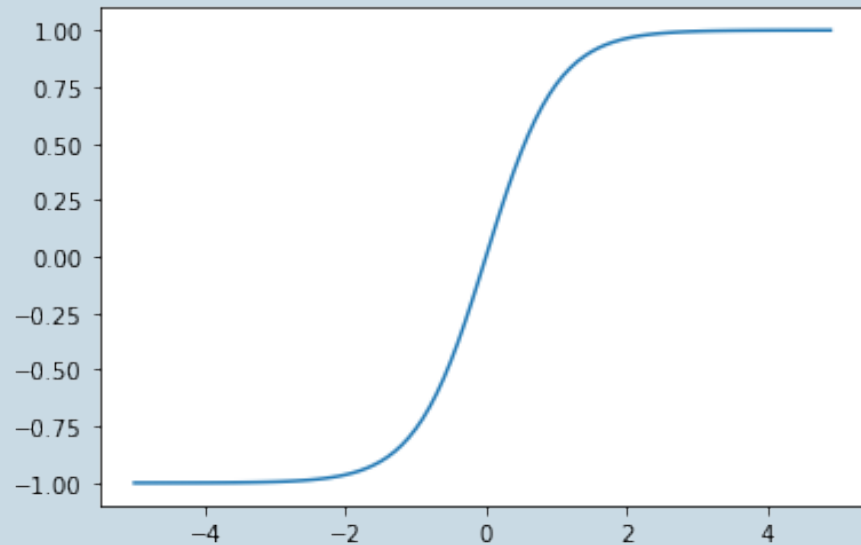


2-3 DCGAN

tanh • sigmoid

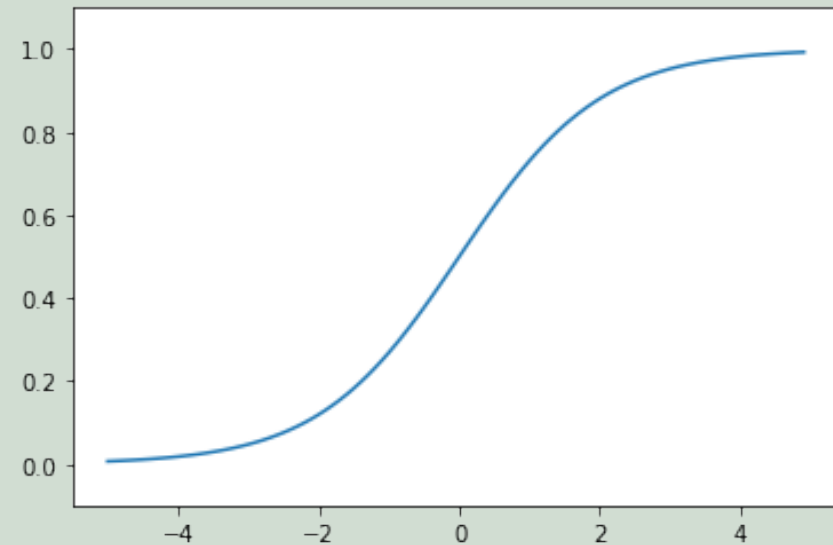
tanh

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



sigmoid

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



3. 演習内容

演習方法

演習の流れ

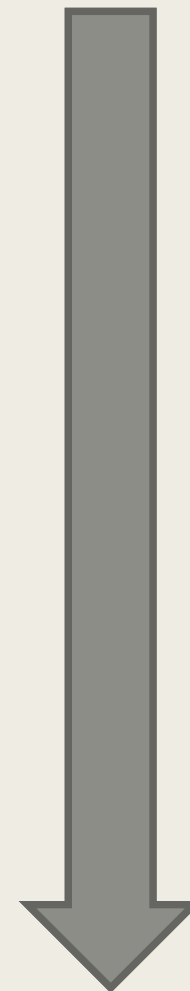
1.画像データの収集

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.まとめ



演習方法

演習の流れ

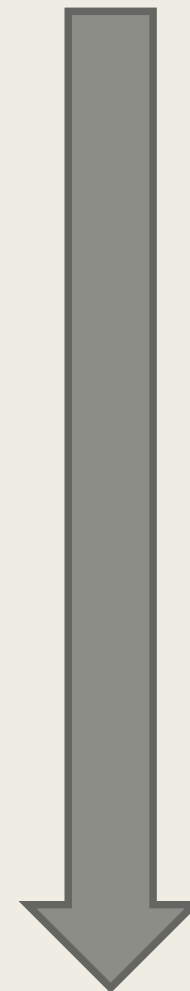
1.画像データの収集

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.まとめ



3-1 画像データの収集 スクレイピング

Excite画像検索から

Webスクレイピングで普段着とコレクションウェアの
画像を収集

MIT License

Copyright © 2018 Doarakko

<https://github.com/Doarakko/scraping-challenge/tree/master/excite-image-scraping>

3-1 画像データの収集 スクレイピング

➤ 普段着

cardigan、fleece、hoody、jacket、polo shirt、shirt、sweats、T-shirt

上記の8種類のトップスの画像を各500枚ずつスクレイピング

➤ コレクションウェア

50種類のブランドから各200枚ずつスクレイピング

演習方法

演習の流れ

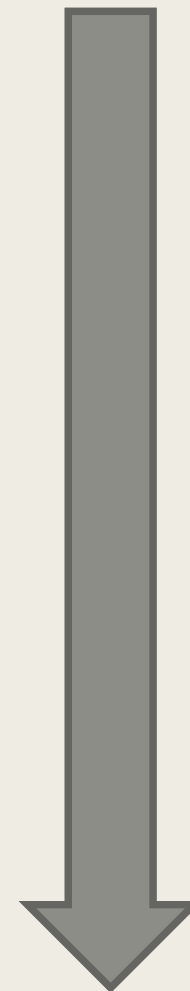
1.画像データの収集

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.まとめ



3-2 画像データの整形

普段着

正面からのトップスの画像を残す（目視で判断）

以下の画像を除去

- パンツなどのトップスでない画像
- 横向き、後ろ向きの画像
- サイズの小さい画像（ 128×128 ピクセル未満）

3-2 画像データの整形

コレクションウェア

モデルが正面を向いている画像を残す（目視で判断）

以下の画像を除去

- モデルが複数人写っていて重なっている画像
- サイズの小さい画像（128×128ピクセル未満）
- コレクションウェアでない画像
- モデルが座っていて服が見えづらい画像

3-2 画像データの整形 トリミング

普段着とコレクションウェアの画像に対してトップスが中心に来るようにトリミング（手作業）

例



3-2 画像データの整形 データの前処理

Pillowを用いて

画像データのカラーモードをRGBに変換

画像サイズを統一（128×128ピクセル）

Pillowとは

Pythonに、各種形式の画像ファイルの読み込み・操作・保存を行う 機能を提供する画像処理ライブラリ

3-2 画像データの整形 結果（データセットの枚数）

	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

画像の整形後、データセットの各500枚はより普段着っぽいデザイン、コレクションウェアっぽいデザインを選別

演習方法

演習の流れ

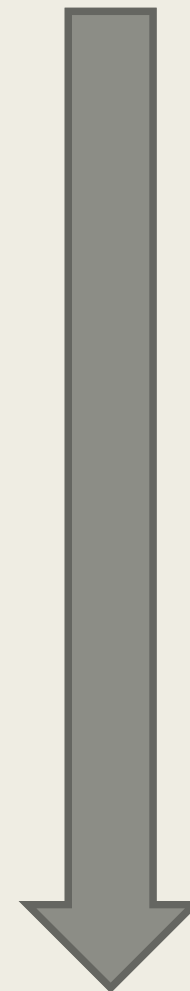
1.画像データの収集

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.まとめ



3-3 デザインを生成するモデルの構築 使用しているライブラリ

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

3-3 デザインを生成するモデルの構築

デザインを生成するモデルの概要

- 128×128ピクセルの画像を生成
- 学習回数は10000エポック

3-3 デザインを生成するモデルの構築

DCGANを以下の文献を参照してKerasを用いて構築

Generator

入力は100次元の各要素が-1~1の範囲の浮動小数点数をとるベクトル

Discriminator

文献[2]よりDropoutを用いている

[1]Radford, Alec, Luke Metz, and Soumith Chintala.

“UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS”

[2]GANについて概念から実装まで ～DCGANによるキルミーベイバー生成～

<https://qiita.com/taku-buntu/items/0093a68bfcae0b0ff879d>

3-3 デザインを生成するモデルの構築

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	○	—
Upsampling2D	—	—	1024	—	—
Conv2D	3×3	1	512	○	ReLU
Upsampling2D	—	—	512	—	—
Conv2D	3×3	1	256	○	ReLU
Upsampling2D	—	—	256	—	—
Conv2D	3×3	1	128	○	ReLU
Conv2D	3×3	1	3	—	tanh

3-3 デザインを生成するモデルの構築

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	Dropout	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

演習方法

演習の流れ

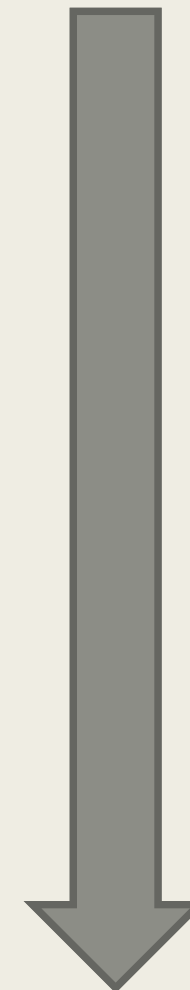
1.画像データの収集

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.まとめ



3-4 生成結果

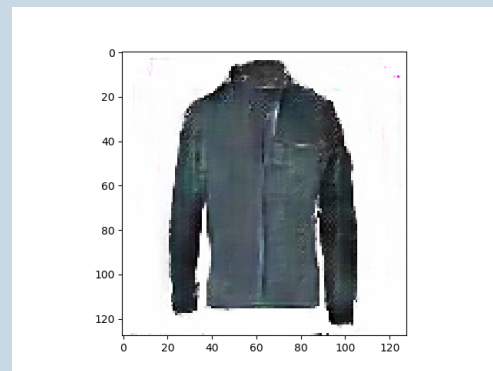
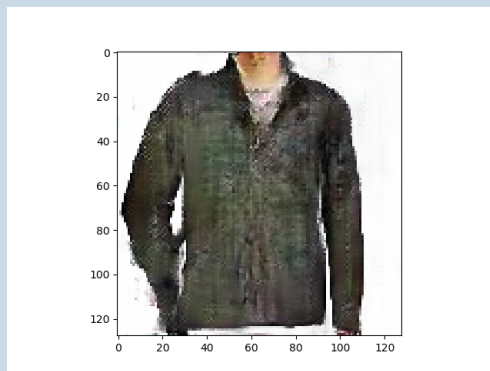
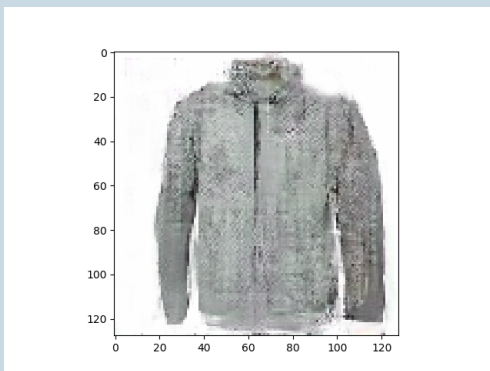
教師画像には普段着とコレクションウェアの2種類



普段着っぽいものとコレクションウェアっぽいもの
どちらか生成されるはず

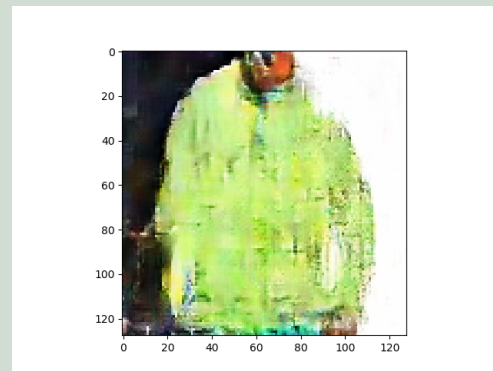
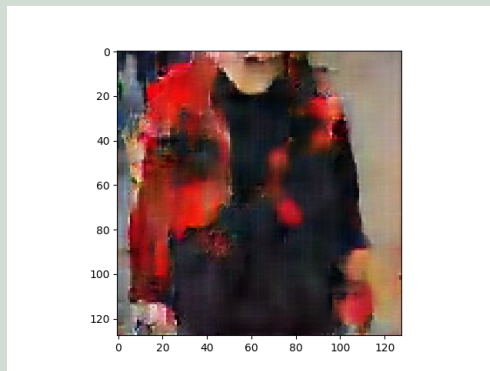
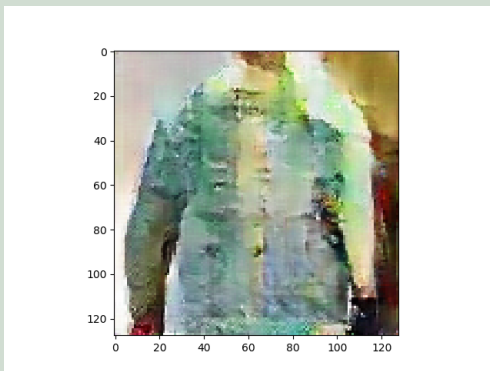
3-4 生成結果

普段着っぽいデザイン



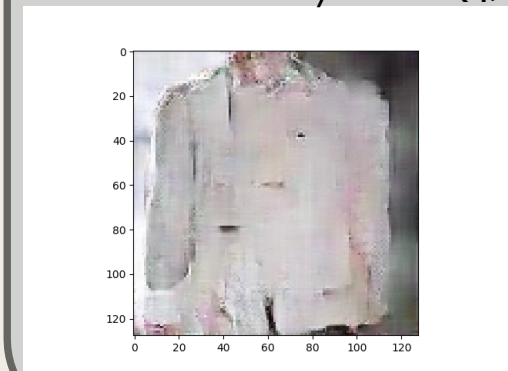
38/100 (枚)

コレクションウェアっぽいデザイン



41/100 (枚)

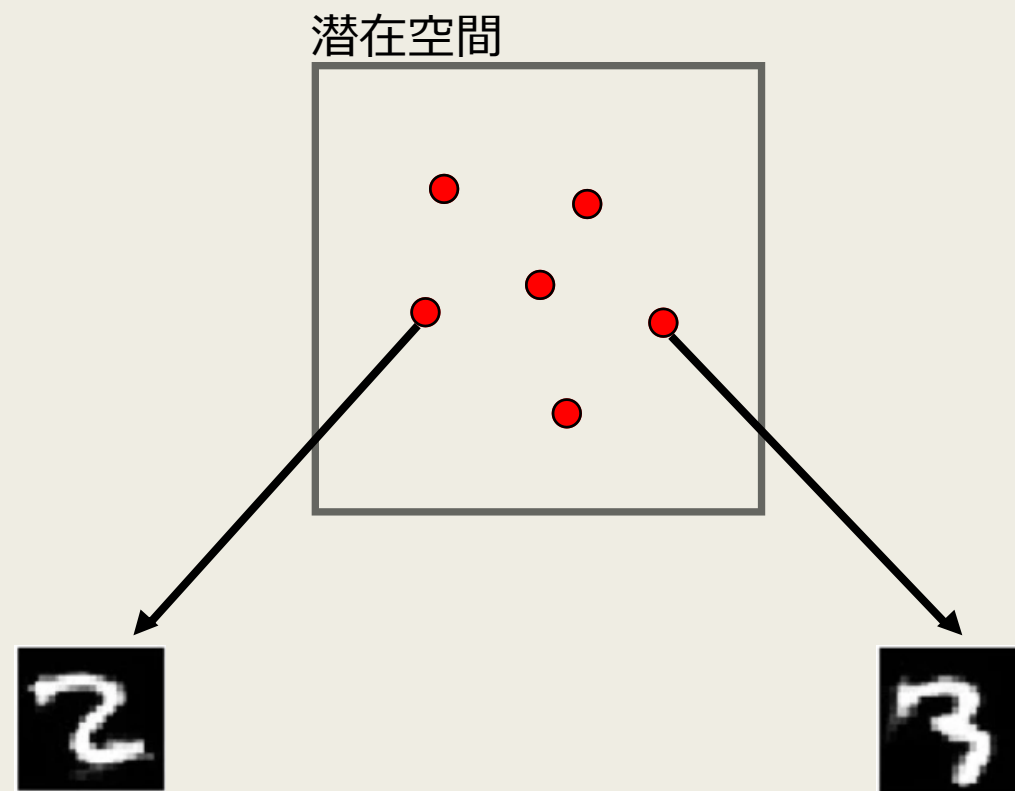
うまく生成されない
デザイン 21/100 (枚)



3-4 生成画像 潜在空間

Generatorの入力の多次元ベクトルが存在する空間を潜在空間という

学習済みのGeneratorでは潜在空間内の1点からは決まった画像が生成される

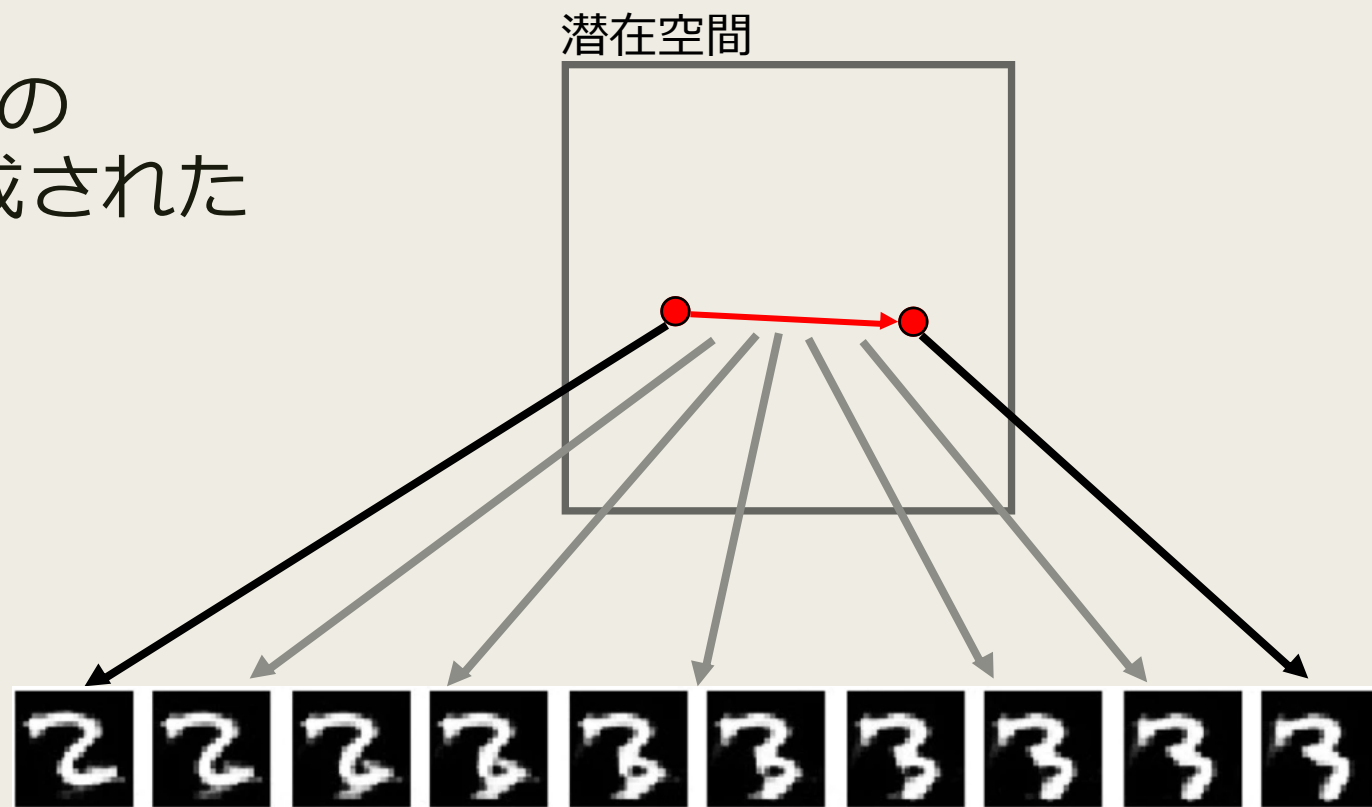


引用元 今さら聞けないGAN (3) 潜在変数と生成画像

<https://qiita.com/triwave33/items/a5b3007d31d28bc445c2>

3-4 生成画像 潜在空間

潜在空間の2点間のパス上の
一連のポイントから、生成された
2つの画像間の遷移を示す
一連の画像を生成可能



引用元 今さら聞けないGAN (3) 潜在変数と生成画像
<https://qiita.com/triwave33/items/a5b3007d31d28bc445c2>

3-4 生成結果

学習済のGeneratorを用いて画像を100枚生成

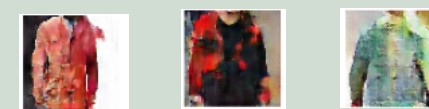
目視で普段着と
コレクションウェアに識別



普段着



コレクションウェア



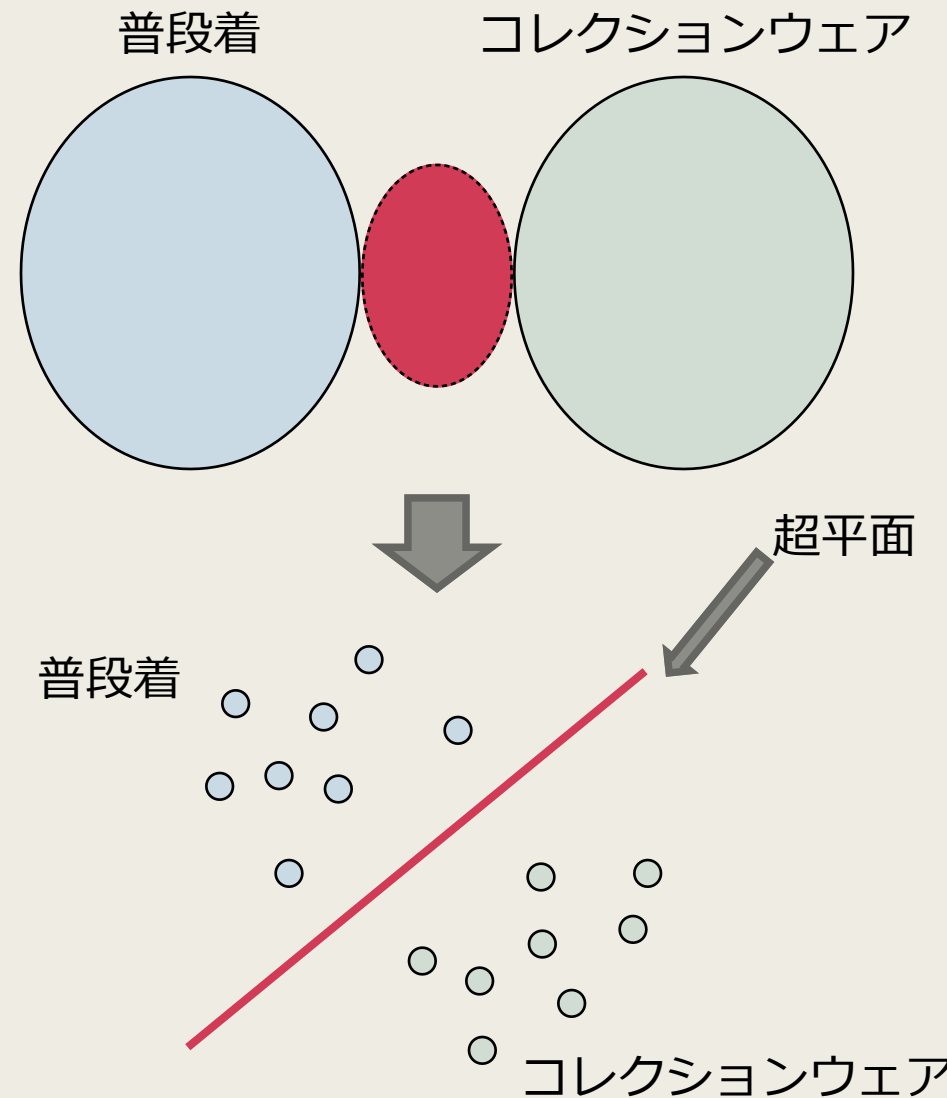
3-4 生成画像

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したい



例)

SVMを用いて分類する超平面を獲得できればよい

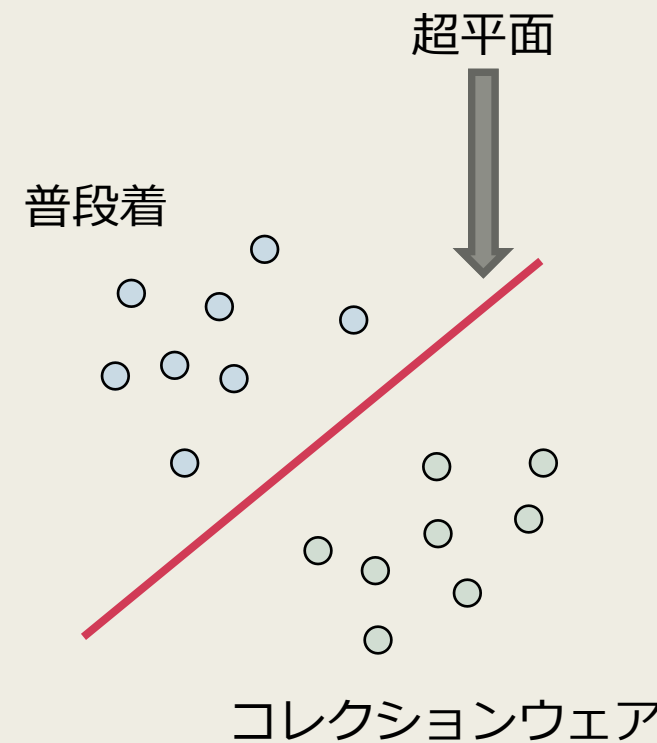


3-4 生成画像

SVMで普段着とコレクションウェアの集合を分類することができた

獲得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

1つの普段着と1つのコレクションウェアの間を
獲得することは可能

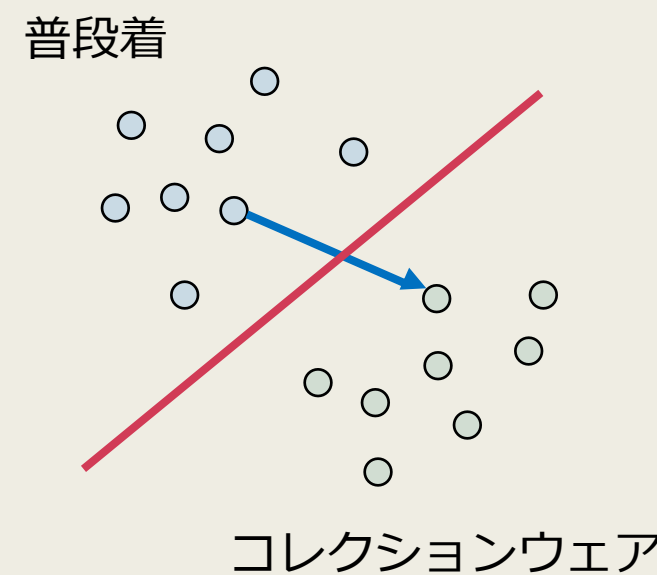


3-4 生成画像

SVMで普段着とコレクションウェアの集合を分類することができた

獲得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

1つの普段着と1つのコレクションウェアの間を
獲得することは可能



3-4 生成画像

1つの普段着と一つのコレクションウェアの中間

普段着



コレクションウェア

3-4 生成画像

1つの普段着と一つのコレクションウェアの中間

普段着



中間

コレクションウェア

演習方法

演習の流れ

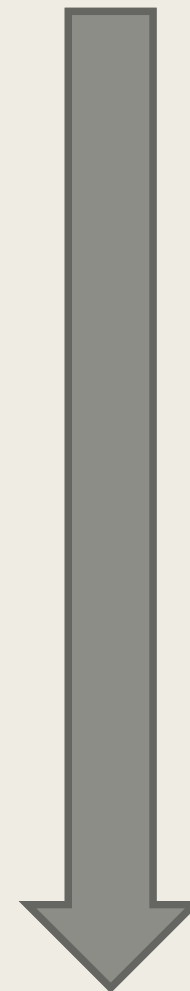
1.画像データの収集

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.まとめ



4. まとめ

4 まとめ 評価

- 1つの普段着と1つのコレクションウェアの間は自分が着てみたいと思えるデザインである



4 まとめ

今後の課題

- SVMで分類した超平面の解析
- 高画質のデータセット作成
- 3Dでのデザイン
- より一般的なものを獲得する