修 士 論 文 概 要 書

Summary of Master's Thesis

Date of submission: 01/25/2021

専攻名（専門分野） Department	情報理工・ 情報通信専攻	氏名 Name	丸寄 佳奈子	指 導 教 員 Advisor	渡辺 裕 印
研究指導名 Research guidance	オーディオビジュアル 情報処理研究	学籍番号 Student ID number	CD 5119F090-2		
研究題目 Title	Capsule Network を識別器と生成器に用いた敵対的生成ネットワークによる画像生成 Image Generation by Generative Adversarial Networks Using Capsule Network for Discriminator and Generator				

1. まえがき

近年, Convolutional Neural Network (CNN)[1]を用いた画像処理が数多く提案されている．新しいデータを生成する敵対的生成ネットワークである Generative Adversarial Networks (GAN)[2]もその一つであり，CNN がよく用いられている．しかしながら，GAN による画像生成は難しく，生成画像の品質が安定しない．

CNN には画像の特徴間の空間的情報が失われるという欠点がある．2017 年に，CNN の欠点を補った Capsule Network[3]が発表された．Capsule Network を GAN に取り入れれば，より品質のよいデータを生成することができると考えられる．

本研究では，Capsule Network を GAN の Discriminator および Generator の両方の構造に組み込んだ Capsule GAN を提案する．

2. 関連研究

Capsule Network を用いた GAN として，Huseyn Gadirov らの研究[4]や Ayush Jaiswall[5]らの研究が挙げられる．これら二つの GAN は，Discriminator のみに Capsule Network を使用し，Generator には CNN を用いている．本論文では，Huseyn Gadirov らが提案した GAN を Capsule GAN1 と定義する．

Capsule GAN1 は，MNIST[6]および CIFAR-10[7]で CNN を用いた代表的な GAN である Deep Convolutional GAN (DCGAN)[8]よりも高い品質の画像を生成することに成功している．

3. 提案手法

3.1. Discriminator の層を Generator に使用した構造 (Capsule GAN2)

この構造では，Discriminator および Generator の両方に Capsule Network を使用している．Discriminator には Capsule Network を CNN のように組み込む．Generator には，Discriminator から取り出した画像の特徴を保持している DigitCaps 層を使用する．取り出した DigitCaps 層と乱数を掛け合わせた値を Generator の入力とする．本論文では，この Capsule GAN を Capsule GAN2 と定義する．

3.2. Capsule Network を Generator に使用した構造 (Capsule GAN3)

この構造では，Discriminator および Generator の両方に Capsule Network を使用している．Capsule GAN2 と同様，Discriminator には Capsule Network を CNN のように組み込む. Generator には, Capsule Network の流れを逆にした構造を組み込む．本論文では，この Capsule GAN を Capsule GAN3 と定義する．Capsule GAN3 では，生成クラスのラベルを使用し，生成クラスごとに画像を生成する構造も提案する．

4. 評価実験

4.1. 比較実験

比較実験では，CNN を用いた GAN と Capsule Network を用いた Capsule GAN1，Capsule GAN2 および Capsule GAN3 の 4 種類の GAN の比較を行った．本実験で用いた GAN の概略図を図 1 に示す．

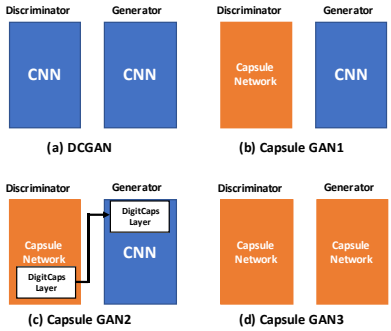

図 1　実験に用いた GAN の概略図

使用したデータセットは白黒画像の MNIST および FashionMNIST[9]とカラー画像の猫画像である．また，猫画像においては GAN の安定化手法である Wasserstein GAN-gp (WGAN-gp)[10]の手法を取り入れて画像を生成した．したがって，比較した CNN を用いた GAN は WGAN-gp である．生成した猫画像の結果を図 2 に示す．

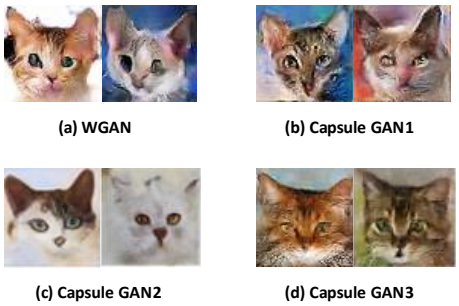

図 2　生成画像 (猫画像)

また，Inception Score (IS)[11]および Geometry Score (GS)[12]を用いて評価した．結果を表 1 に示す．

表 1　評価結果

データセット	評価手法	GAN (Using CNN)	Capsule GAN1	Capsule GAN2	Capsule GAN3
MNIST	IS	2.32	2.35	2.37	**2.57**
	GS (×100)	3.89	5.01	3.48	**1.46**
Fashion MNIST	IS	4.39	4.34	4.48	**4.54**
	GS (×100)	0.123	0.129	0.205	**0.103**
猫画像	IS	4.17	4.53	4.68	**4.72**
	GS (×100)	0.192	0.221	3.84	**0.106**

図 2 より，目視での比較では Capsule GAN2 および Capsule GAN3 が他の二つの結果に比べて猫の形を上手く捉えて生成できていることがわかる．また表 1 より，どのデータセットにおいても Capsule GAN3 の評価が一番良い結果となった．したがって，比較した 4 種類の GAN においては Capsule GAN3 が一番品質の良い画像を生成できることが示された．このことから，CNN を使用した GAN よりも Capsule Network を使用した GAN の方が品質の良い画像を生成することがわかる．

4.2.　ラベルを使用した検証実験

検証実験では，Capsule GAN3 の構造においてクラスごとに画像を生成できるか検証した．MNIST の生成結果を図 3 に示す．図 3 の結果は，行ごとに使用しているラベルが違う結果を示す．

図 3　検証結果

図 3 より，生成クラスのラベルが示すクラスごとに，画像を生成できていることがわかる．

5. 結論

本研究では，Capsule Network を Discriminator および Generator に使用した Capsule GAN2 および Capsule GAN3 を提案した．Capsule GAN3 においては，実験に用いた全てのデータセットにおいて一番品質の良い画像を生成した．また，Capsule GAN3 はラベルを用いることで生成クラスごとに画像を生成することが可能であることを確認した．

参考文献

[1] A. Krizhevsky, I. Sutskever and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, Neural Information Processing Systems (NIPS), pp. 1106-1114, Dec. 2012.

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative Adversarial Networks”, Neural Information Processing Systems (NIPS), pp. 2672-2680, Dec. 2014.

[3] S. Sabour, N. Frosst and G. E. Hinton, “Dynamic Routing Between Capsules”, Neural Information Processing Systems (NIPS), pp. 3859-3869, Dec. 2017.

[4] H. Gadirov, M. Tamošiūnaité and D. Vitkute-Adzgauskiene, “Capsule Architecture as a Discriminator in Generative Adversarial Networks”, Vytautas Magnus University, Feb. 2018, M. D. thesis.

[5] A. Jaiswal, W. AbdAlmageed, Y. Wu and P. Natarajan, “CapsuleGAN: Generative Adversatial Capsule Network”, European Conference on Computer Vision (ECCV), pp-526-535, Sep. 2018.

[6] Y. LeCun, C. Cortes and C. J. C. Burges, “The MNIST Database of Handwritten Digits”, http://yann.lecun.com/exdb/mnist/, 1998.

[7] Alex Krizhevsky, “Convolutional deep belief networks on CIFAR-10”, Aug. 2010.

[8] A. Radford, L. Metz and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, International Conference on Learning Representations (ICLR), Jan. 2016.

[9] H. Xiao, K. Rasul and R. Vollgraf, “Fashion-MNIST: A Novel Image Dataset”, arXiv preprint arXiv: 1708.07747, 2017.

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. Courville, “Improved Training of Wasserstein GANs”, Neural Information Processing System (NIPS), pp. 5769-5779, Dec. 2017.

[11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, “Improved Techniques for Training GANs”, Neural Information Processing Systems (NIPS), pp.2234-2242, Dec. 2016.

[12] V. Khrulkov and I. Oseledets, “Geometry Score: A Method for Comparing Generative Adversarial Networks”, International Conference on Machine Learning (ICML), pp. 2621-2629, Jul. 2018.

2020 年度

早稲田大学大学院基幹理工学研究科情報理工・情報通信専攻　修士論文

Capsule Network を識別器と生成器に用いた敵対的生成ネットワークによる画像生成

Image Generation by Generative Adversarial Networks Using Capsule Network for Discriminator and Generator

丸嵜　佳奈子

(5119F090-2)

提出日：2021.01.25

指導教員：渡辺　裕　印

研究指導名：オーディオビジュアル情報処理研究

目次

第1章　序論

1.1　研究の背景

近年，Convolutional Neural Network (CNN)[1]を用いた画像処理が数多く提案されている．顔認識[2]，白黒画像のカラー化[3]などが例に挙げられる．新しいデータを生成する敵対的生成ネットワークである Generative Adversarial Networks (GAN)[4]もその一つである．最近では，GAN を用いた漫画生成[5]やゲーム生成[6]などが発表されている．近年，研究が活発に行われている分野の一つである．しかし，GAN などの多くの画像処理に用いられている CNN には，画像の特徴間の空間的情報が失われるという欠点がある．2017 年に CNN の欠点を補った Capsule Network[7]が発表された．CNN に比べ画像の特徴を捉えることに優れる Capsule Network を GAN に取り入れれば，より品質の良いデータを生成することができると考えられる．

そこで，本研究では Capsule Network を GAN の Discriminator および Generator の両方の構造に組み込んだ Capsule GAN を提案する．

1.2　本研究の目的

CNN を用いた GAN は数多く研究されている．代表的なものに Deep Convolutional GAN (DCGAN)[8]が挙げられる．また，画像を生成する際に画像の種類を示すラベルを付与した Convolutional GAN[9]，Auxiliary Classifier GAN[10]や，GAN の学習過程におけるモード崩壊を，Wasserstein 距離を使用し防ぐ Wasserstein GAN (WGAN)[11]，Wasserstein GAN-gp (WGAN-gp)[12]などもある．しかしながら，GAN による画像生成は難しく，モード崩壊などが起こりやすい欠点がある．そのため，生成画像の品質が安定しない．

CNN の欠点を補った Capsule Network を GAN の構造に組み込めば，より品質の良い画像を生成できると考えられる．したがって，本研究では Capsule Network を GAN に組み込み，より品質の良い画像を生成する手法を提案する．

1.3　関連研究

Capsule Network を用いた GAN の研究として，Huseyn Gadirov らの研究[13]や Ayush Jaiswall らの研究[14]が挙げられる．Huseyn Gadirov らは，MNIST[15]および CIFAR-10[16]で CNN を用いた標準的な GAN である DCGAN よりも高い品質の画像を生成することに成功している．これらの手法は，GAN の Discriminator のみに Capsule Network を使用している．しかしながら，Generator の構造は DCGAN のように CNN を用いてお

り，Capsule Network を使用していない．我々の過去の研究[17]では，GAN の Discriminator から取り出した Capsule Network の層を Generator の入力に使用した．本論文ではこの GAN に加え，Generator の構造自体に Capsule Network を組み込んだ Capsule GAN を提案する．

1.4 本論文の構成

本論文の構成を以下に示す．

第 1 章は本章であり，本研究の背景，目的及び関連研究について述べる．

第 2 章では関連技術について述べる．

第 3 章では本論文で提案する手法について述べる．

第 4 章では提案手法の実験，結果及び考察について述べる．

第 5 章では本論文の結論と今後の課題について述べる．

第2章　関連技術

2.1 まえがき

本章では，本論文で用いる関連技術について述べる．

2.2 CNN

2.2.1 CNN の概略

CNN とは，画像認識や音声認識などで多く用いられているニューラルネットワークの一種である[18]．CNN の基本構造を図 2.1 に示す．各層は複数のノードからなる．CNN では，畳み込み層，プーリング層および全結合層を隠れ層に持つ．畳み込み層とプーリング層は交互に組み込まれる．組み込んでいく層の数は，構造によって異なる．これら二つの層により，入力されたデータの特徴を得ることができる．全結合層では，得られた特徴を一つのノードに結合し，活性化関数により変換された値を出力する。全結合層の出力を出力層で確率に変換することで，入力データを識別および分類することが可能となる．

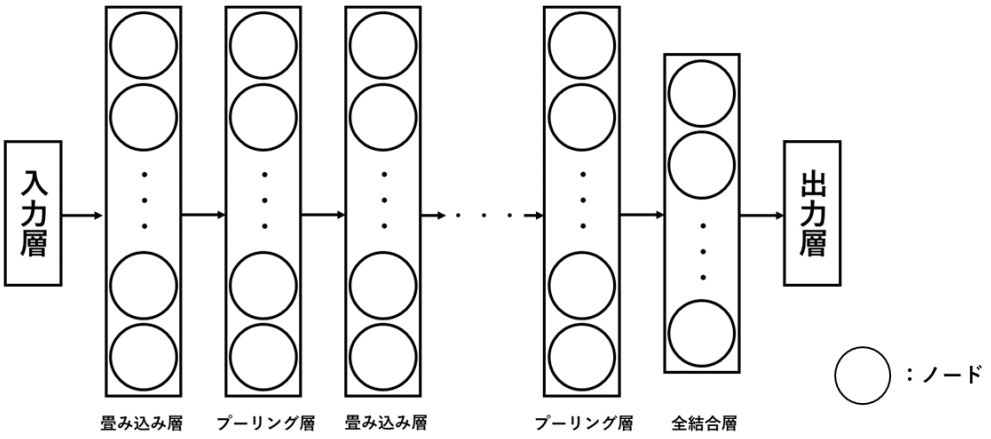

図 2.1　CNN の基本構造

2.2.2 CNN の欠点

CNN は，プーリング層により画像の位置変化に強くなる．しかしながら，その一方で画像の空間的情報を損失してしまうという欠点がある．画像の空間的情報の損失について，図

2.2 より人の顔を例にして説明する．CNN では，人の顔のパーツである目，鼻などの特徴を畳み込み層で捉え特徴マップを生成する．その後，プーリング層で特徴マップを縮小する．畳み込み層が顔のパーツの特徴をそれぞれ捉えると，画像内でパーツの配置がバラバラになっていたとしても，プーリング層での処理によりその画像を人の顔だと判断してしまう．それゆえに，パーツの特徴間の位置関係を無視してしまう[19]．したがって，プーリング層での処理により CNN は画像の特徴間の位置関係である空間的情報を損失する．

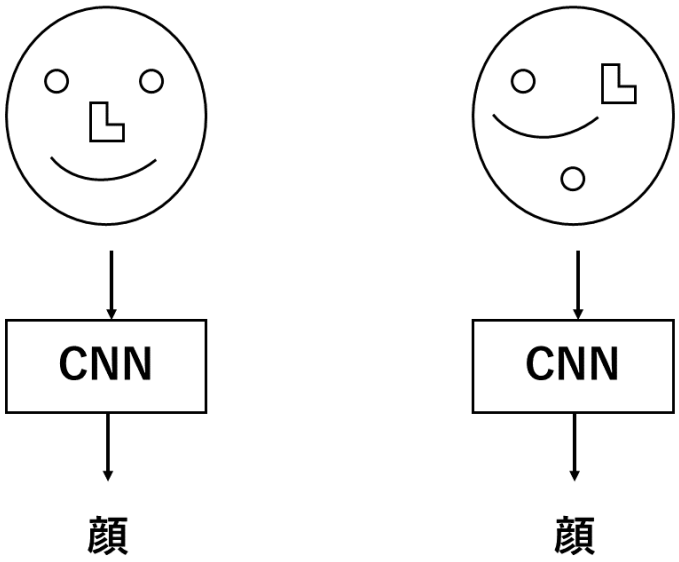

図 2.2　CNN の誤認識

2.3 Capsule Network

Capsule Network とは，CNN をベースとしたニューラルネットワークである[20]．CNN には，2.2.2 節で述べた通りプーリング層により画像の特徴間の空間的情報が失われるという欠点がある．Capsule Network ではプーリング層をなくし，各ニューロンへの入力をスカラーではなくベクトルとすることで，その欠点を補うことを可能とした．このベクトルをカプセルと呼んでいる．図 2.3 に Capsule Network の基本構造を示す．図 2.3 に示している構造は，28×28 pixels の画像を入力した場合の構造である．レイヤーl_iの入力ベクトル$\boldsymbol{u}_i$に対して，対応する重み行列$\boldsymbol{W}$がかけられる．$\boldsymbol{U}_i$の大きさは，対応するオブジェクトの存在確率を表す．$\boldsymbol{U}_i$の向きは，そのオブジェクトの空間的な情報を表す．式を以下に示す．

	$\hat{\boldsymbol{u}}_{j\|i} = \boldsymbol{W}_{ij}\boldsymbol{u}_i$	(2.1)

Capsule Network では，重みは dynamic routing という手法で学習させる．重みをc_{ij}とする．iはl層に含まれるカプセル，jは$l+1$層に含まれるカプセルとする．c_{ij}は以下のように定義される．

	$$c_{ij} = \frac{exp(b_{ij})}{\sum_k exp(b_{jk})}$$	(2.2)

b_{ij}は，はじめに 0 で初期化され，以下のように更新される．

	$$b_{ij} \leftarrow b_{ij} + \hat{\boldsymbol{u}}_{j\|i} \cdot \boldsymbol{v}_j$$	(2.3)

$\boldsymbol{v}$は出力ベクトルである．この処理を繰り返し行う．繰り返し行うことにより，この処理において重要な値が大きくなる．Dynamic routing 後の出力層を DigitCaps 層と呼ぶ．

Capsule Network は，活性化関数に squash 関数を使用する．入力ベクトルを$\boldsymbol{s}$，出力ベクトルを$\boldsymbol{v}$とする．出力ベクトル$\boldsymbol{v}$は以下の式で表される．

	$$\boldsymbol{v} = \frac{\|\boldsymbol{s}\|^2}{1+\|\boldsymbol{s}\|^2}\frac{\boldsymbol{s}}{\|\boldsymbol{s}\|}$$	(2.4)

Squash 関数はベクトルの向きを変えずに大きさを 1 にスケーリングする．

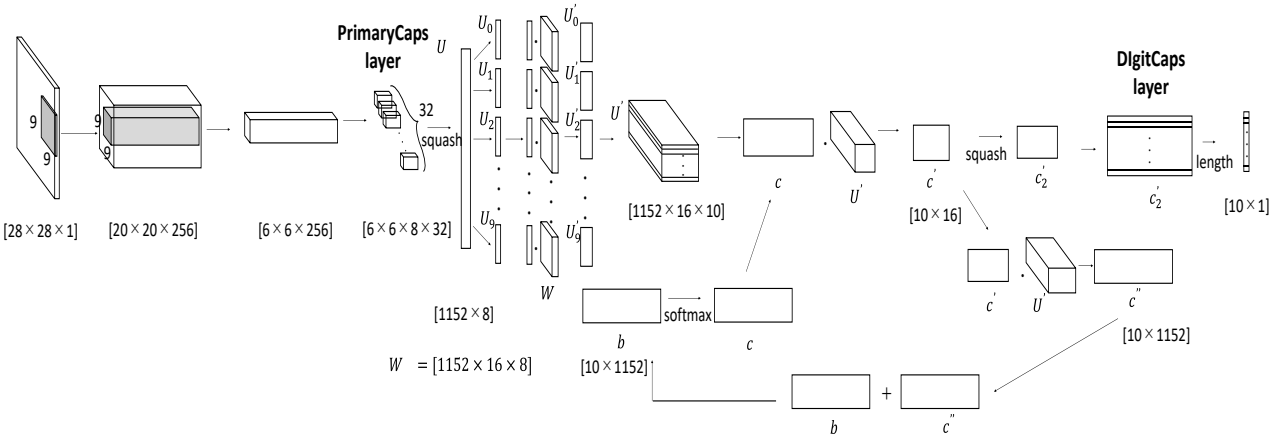

図 2.3　Capsule Network の構造

2.4 GAN

2.4.1 GAN の概要

GAN とは，Discriminator，Generator と呼ばれる二つのネットワークを用いて，入力データと似たようなデータを生成するモデルである[21]．図 2.4 に GAN の概略図を示す．Generator は乱数を入力とし，データセットと似たようなデータを生成し，出力する．Discriminator は，Generator が生成したデータと訓練データセットを入力とし，入力データが訓練データセット（本物）か生成データ（偽物）かを識別する．Generator は Discriminator を騙せるような本物そっくりなデータを生成できるように学習していく．Discriminator は Generator に騙されないように入力データを識別できるように学習していく．すなわち，Discriminator と Generator は互いに競い合いながら学習する．Discriminator の出力をD，Generator の出力をG，訓練データセットを$\boldsymbol{x}$，乱数を$\boldsymbol{z}$，訓練データセットの画像分布を$p_{data}(\boldsymbol{x})$，乱数の分布を$p_z(\boldsymbol{z})$とする．損失関数の式を以下に示す．

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{2.5}$$

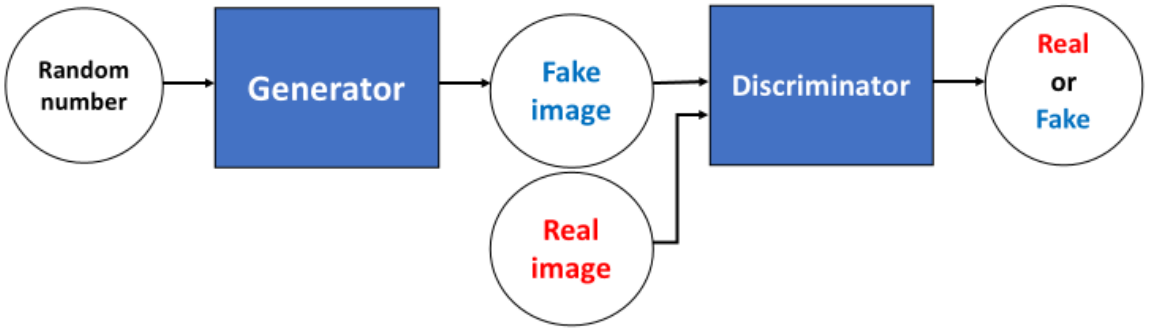

図 2.4　GAN の概略図

しかしながら，GAN にはいくつかの問題点がある．Generator が同じような画像しか生成しなくなるモード崩壊が例に挙げられる．また，学習の途中でパラメータの勾配消失が起こりやすいという問題点もある．

2.4.2 DCGAN

DCGAN とは，GAN の Discriminator，Generator の構造に CNN を組み込んだ GAN である．GAN は学習を進めていくことが難しいが，DCGAN ではプーリング層，全結合層の

廃止などといった手法[22]を提案することにより学習の安定に成功している．

2.4.3 WGAN

WGAN とは，2017 年に Martin Arjovsky らによって発表された学習の安定化手法を取り入れた GAN の一つである[23]．従来の GAN では，損失関数の計算に Jensen-Shannon divergence (JS-divergence)を使用していた. JS-divergence を用いた損失関数の計算では，パラメータの勾配消失が起こりやすいという欠点がある．WGAN では JS-divergence ではなく Wasserstein distance を用いて損失関数を計算している．Wasserstein distance を用いることで，勾配消失が起こりにくくなる．Wasserstein distance をW，1 次のリプシッツな関数を$f: \boldsymbol{x} \to \mathbb{R}$とする．Wasserstein distance を以下に示す.

$$W\left(p_{data(\boldsymbol{x})}, p_{z(\boldsymbol{z})}\right) = \sup_{\|f\| \leq 1} \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[f(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[f(\boldsymbol{x})] \tag{2.6}$$

また，式 (2.6) をパラメータ$\boldsymbol{w}$のニューラルネットワークで近似したものを以下に示す.

$$W\left(p_{data(\boldsymbol{x})}, p_{z(\boldsymbol{z})}\right) = \max_{w \in W} \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[f_w(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[f_w(G(\boldsymbol{z}))] \tag{2.7}$$

f_wを WGAN では Discriminator の出力とする．Wasserstein 距離を用いた損失関数を以下に示す.

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}\left[\left(D(G(\boldsymbol{z}))\right)\right] \tag{2.8}$$

WGAN は Wasserstein 距離を用いることにより，学習途中の勾配消失やモード崩壊を避けることに成功している．

2.4.4 WGAN-gp

WGAN-gp とは，2017 年に Ishaan Gulrajani らによって発表された学習の安定化手法を取り入れた GAN の一つであり, WGAN を改良した手法である[24]. 損失関数は, 式 (2.8) に示す WGAN の損失関数の式に，gradient penalty 項と呼ばれる制約項を加えたものである．WGAN よりも学習の安定化に成功している．

2.4.5 CapsuleGAN (Capsule GAN1)

CapsuleGAN とは，2018 年に Huseyn Gadirov らが発表した Capsule Network を用いた GAN である．GAN の Discriminator に Capsule Network，Generator に CNN を用いた構造である[25]．本論文では，この CapsuleGAN を Capsule GAN1 と定義する．

2.4.6 評価方法

2.4.6.1 Inception Score (IS)

Inception Score[26]とは，2016 年に Tim Salimans らが発表した GAN を評価する際に使用される評価指標である．画像が Inception モデル[27]で識別しやすく，かつ識別されるラベルの種類が多いほど Inception Score が高くなる．Inception Score は，値が大きいほど生成画像が良い画像であると評価する[28]．$\boldsymbol{x}_i$をi番目の画像データ，yをラベル，i番目の画像を Inception モデルに入力して得られるラベルyの確率を$p(y|\boldsymbol{x}_i)$，画像データ全体のラベルyの確率を$p(y)$，使用する画像の集合をXとする．Inception Score は，$p(y|\boldsymbol{x}_i)$と$p(y)$の確率分布の KL-divergence を求めたものである．式を以下に示す．

$$\mathrm{IS} = \exp\left(\frac{1}{X}\sum_{\boldsymbol{x}_i \in X} p(y|\boldsymbol{x}_i)\log\frac{p(y|\boldsymbol{x}_i)}{p(y)}\right) \tag{2.9}$$

$p(y|\boldsymbol{x}_i)$と$p(y)$の確率分布の差が大きいほど，Inception Score は大きくなる．

2.4.6.2 Geometry Score (GS)

Geometry Score[29]とは，2018 年に Valentin Khrulkov らが発表した GAN を評価する際に使用される評価指標である．機械学習には，高次元空間に存在するデータは低次元の非線形多様体に近似できるという多様体仮説[30]がある．Geometry Score は多様体仮説に基づき，訓練データセットの多様体と生成データの多様体を幾何学的に比較する手法である．データ点を中心とする半径α_1の複数の円を考える．半径α_1をα_2 $(\alpha_1<\alpha_2)$ としたとき，円と円の距離が小さくなり穴が形成される．この過程を図 2.5 に示す．

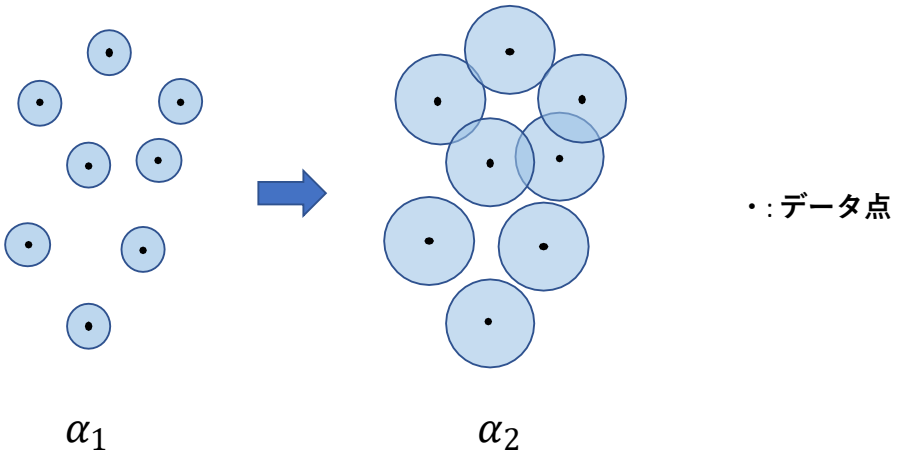

図 2.5　データ間の穴の形成過程

図 2.5 より，円の半径がα_2のとき，上の 5 個のデータ間で穴が形成されている．半径α_2を大きくしてくと，下の 4 個のデータで穴が形成されることが推測できる．また，半径を大きくしていくと，円の半径がα_2のときに形成されていた穴が消失することが推測できる．Geometry Score ではこのデータ間の穴の出現，消失に着目する．穴が出現した時間と消失した時間を計算し，穴が形成されていた時間の分布を比較する．訓練データセットと GAN が生成したデータの分布が似ていれば，モード崩壊が少なく訓練データセットと似た画像を生成していると評価する．データ点の円の半径を$\alpha \in [0, \alpha_{max}]$とする．$b_i$，$d_i$を$k$次元の穴の出現時間と消失時間，$n$を生じた$k$次元の穴の総数とする．$k$次元の穴が半径$\alpha$のときに存在する数を$k$位ベッチ数$\beta_k(\alpha)$とする．$k$位ベッチ数$\beta_k(\alpha)$を以下の式で定義する．

	$\beta_k(\alpha) \triangleq \left\lvert \{[b_i, d_i] \in \{[b_i, d_i]\}_{i=1}^{n} \} : \alpha \in [b_i, d_i] \right\rvert$	(2.10)

データセットをX，XからランダムにサンプリングしたデータをLとする．式 2.10 より，穴の相対的生存時間RLTを以下の式で定義する．

	$RLT(i, k, X, L) \triangleq \dfrac{\mu(\{\alpha \in [0, \alpha_{max}] : \beta_k(\alpha) = i\})}{\alpha_{max}}$	(2.11)

ランダムにサンプリングしたLに関する相対的生存時間RLTの平均を表す平均相対的生存時間 MRLT を以下の式で定義する．

	$MRLT(i,k,X) \triangleq \mathbb{E}_L[RLT(i,k,X,L)]$	(2.12)

Geometry Score では$k=1$とする．比較する二つのデータセットX_1，X_2の Geometry Score は以下の式で定義される．

	$GS(X_1,X_2) \triangleq \sum_{i=0}^{i_{max}-1} \left(MRLT(i,1,X_i) - MRLT(i,1,X_2)\right)^2$	(2.13)

Geometry Score では，Inception Score では特定できなかったモード崩壊を特定することが可能である．

2.5 むすび

本章では，本論文で用いるディープラーニングの技術である CNN，Capsule Network，GAN，および GAN の評価方法について述べた．

第3章　提案手法

3.1　まえがき

本章では，本研究で提案する手法の概要について述べる．本研究では，GAN の Discriminator および Generator の両方に Capsule Network を組み込んだ Capsule GAN を二つ提案する．一つは，Discriminator の層を Generator の入力に使用した GAN（Capsule GAN2）である．もう一つは，Capsule Network を CNN のように Generator に使用した GAN（Capsule GAN3）である．

3.2　Capsule GAN の構造

3.2.1　Capsule GAN2（DigitCaps 層を Generator の入力に組み込んだ構造）

図 3.1 に本研究で提案する Capsule GAN2 の構造を示す．

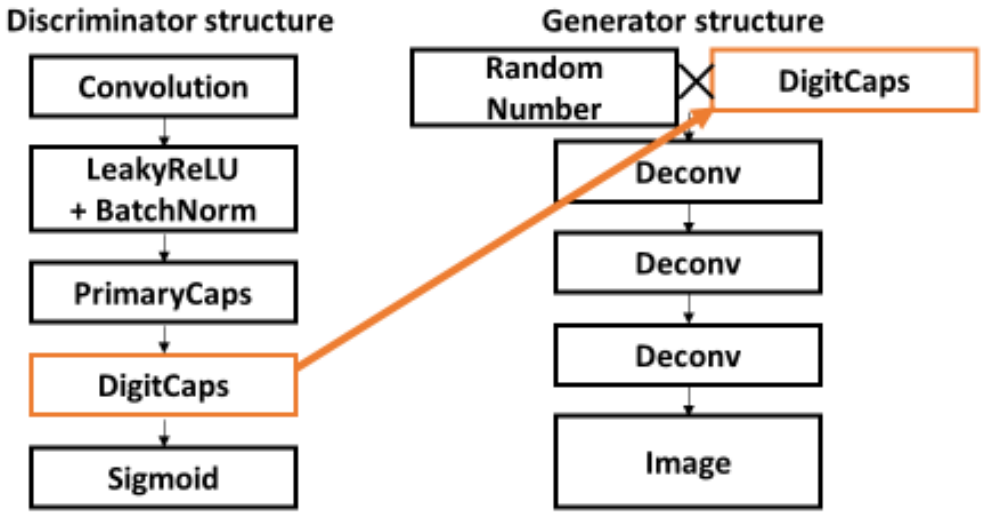

図 3.1　Capsule GAN2 の構造

まず，Discriminator の構造について説明する．Discriminator には Capsule Network の構造をそのまま使用する．Discriminator は，訓練データセットの画像および Generator が

生成した画像を入力とする．画像を Capsule Network に通し，出力として入力された画像が訓練データセットの画像（本物）か生成画像（偽物）かを出力する．Capsule Network の DigitCaps 層は画像の特徴を含んでいる層である． Discriminator に入力する画像は，訓練データセットの画像，生成画像の順に入力される．そのため，DigitCaps 層の出力も訓練データセットの画像の特徴，生成画像の特徴の順番で出力される． この DigitCaps 層を取り出し, Generator の入力に使用する. 次に, Generator の構造について説明する. Generator は Discriminator から取り出した DigitCaps 層と乱数を掛け合わせた値を入力とする． このときに使用する DigitCaps 層は，訓練データセットの画像の特徴を入力した場合のみである. そのため, 取り出した訓練データセットの画像の特徴を含む DigitCaps 層の出力を, 2 回繰り返し使用する．DigitCaps 層の使用の流れを図 3.2 に示す．図 3.2 では，画像をx枚入力した場合の例を示す.

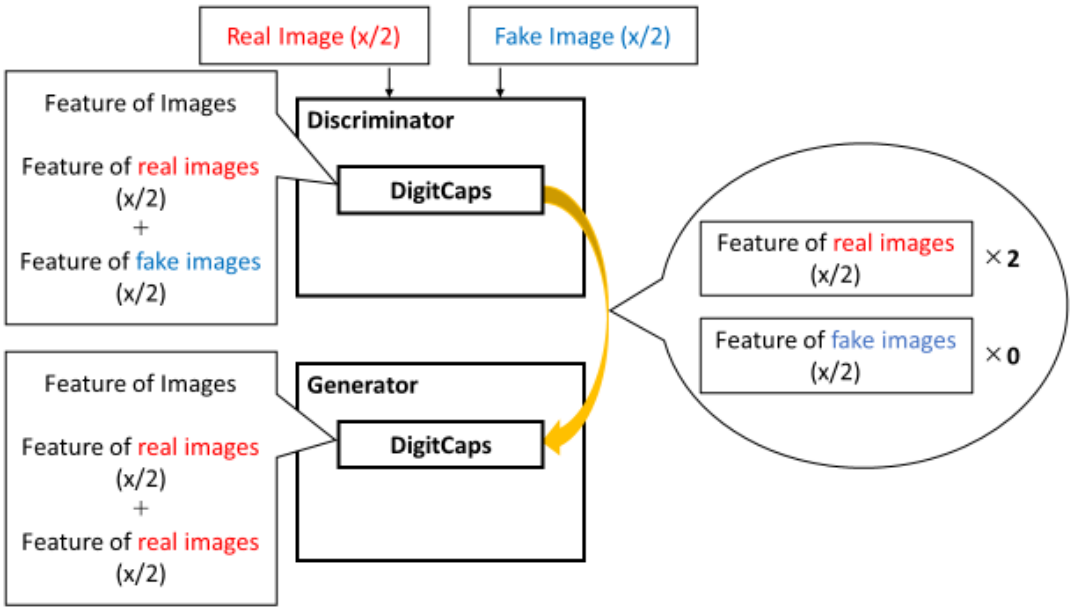

図 3.2　DigitCaps 層の使用の流れ

生成画像の特徴を持つ DigitCaps 層を使用した場合, 学習の初期段階において Generator は, 品質の悪い生成画像の特徴をもつ DigitCaps 層を使用してしまう. なぜなら, Generator は学習の初期段階では画像を上手く生成できないためである. そのため, 生成画像の特徴を持つ DigitCaps 層を使用すると，生成画像の品質が悪くなる．Generator に使用する DigitCaps 層による生成画像の品質の違いを図 3.3 に示す．図 3.3 より，白黒の数字の

MNIST の画像では，生成画像の品質の差はあまり見られない．一方，カラー画像の猫画像では，品質に大きな差があることがわかる．これら二つの結果の違いは，Generator の学習速度の違いが影響していると考えられる．白黒画像よりカラー画像の方が学習するパラメータ数が多い．そのため，猫画像では使用する DigitCaps 層により生成画像の品質に大きな差がでたと考えられる．多くのデータセットに対応するため，本研究で提案する構造では訓練データセットの画像の特徴をもつ DigitCaps 層のみを Generator の入力に用いる．

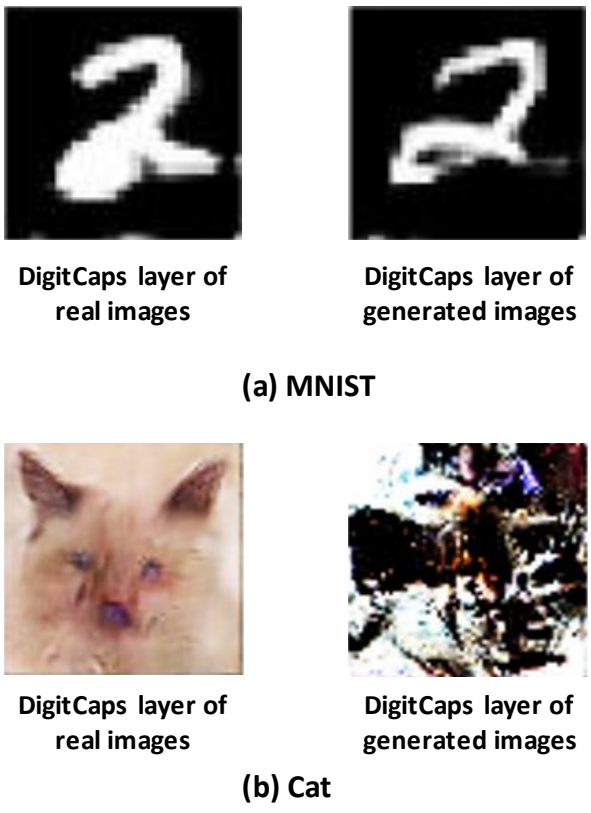

図 3.3　DigitCaps 層の違いによる生成画像の差

DigitCaps 層と乱数を掛け合わせた後，DCGAN と同様に Deconvolution 層と呼ばれる Transposed Convolution 層に入力値を通し画像を生成する．

Capsule GAN2 では，DigitCaps 層を Generator の入力に使用することにより，Discriminator と Generator の両方に Capsule Network を用いた構造となっている．

3.2.2 Capsule GAN3 (Capsule Network を Generator に使用した構造)

3.2.2.1 Capsule GAN3 の概要

本研究で提案する Capsule GAN3 は二つの構造がある．一つは，DCGAN のように一つの重み行列を，生成する全てのクラスで共有する構造である．もう一つは，Convolutional GAN のように生成するクラスごとに重み行列を使用する構造である．この構造では，学習の際に生成クラスを示すラベルを使用する．

3.2.2.2 Capsule GAN3 (一つの重み行列を共有する構造)

図 3.4 に一つの重み行列を共有する場合の Capsule GAN3 の Generator の構造を示す．図 3.4 に示す構造は，28×28 pixels の画像を生成する場合である．

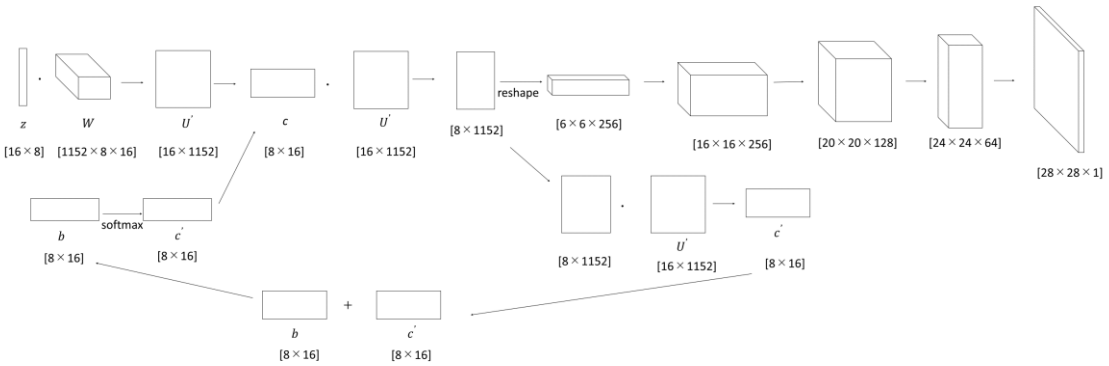

図 3.4　Capsule GAN3 の Generator の構造（一つの重み行列を共有する場合)

Discriminator の構造は Capsule GAN2 と同様である．Generator の構造について説明する．Generator の構造は，図 2.3 で示した Capsule Network の流れを逆にした構造である．Deconvolution 層に入力する前に dynamic routing で画像を生成するに際に重要な値を大きくする．その結果，生成画像の品質を安定化することができる．

3.2.2.3 Capsule GAN3 (複数の重み行列を使用する構造)

図 3.5 に生成クラスごとに重み行列をそれぞれ使用する場合の Capsule GAN3 の Generator の構造を示す．図 3.5 に示す構造は，図 3.4 と同様 28×28 pixels の画像を生成

する場合の構造である．生成クラスは 10 クラスの場合である．行列$\boldsymbol{U}'$ 以降は図 3.4 と同様の流れである．生成クラスごとに違った重み行列を使用し学習する．そのため，学習の際にはデータセットのクラスを示すラベルを使用する．

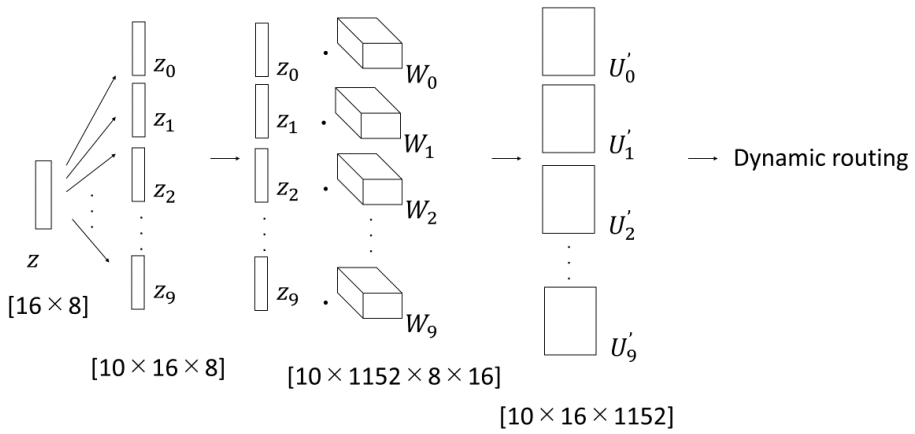

図 3.5　Capsule GAN3 の Generator の構造（複数の重み行列を使用する場合）

3.2.2.2 節と本節より，Capsule GAN3 は Discriminator と Generator の両方に Capsule Network を CNN のように組み込んだ構造となっている．

3.3 むすび

本章では，本研究で提案する Capsule GAN2 および Capsule GAN3 の構造について述べた．

第4章　実験

4.1 まえがき

本章では，第 3 章で提案した手法の評価実験を行う．また，従来手法との比較を行い，考察する．実験に使用したデータセット，4 種類の GAN，実験概要，実験結果及び考察について述べる．

4.2 データセット

4.2.1 データセットの概要

本研究では, 2 種類の白黒画像のデータセットと 1 種類のカラー画像のデータセットを用いて実験を行った．使用したデータセットを以下に示す.

4.2.2 MNIST

MNIST とは，0~9 までの手書き数字の画像データと，その数字のラベルデータがセットとなったデータセットである．学習用に 60000 枚，検証用に 10000 枚用意されている．画像のサイズは 28×28 pixels であり，白黒画像である．本実験では，学習用の画像 60000 枚を使用した.

4.2.3 FashionMNIST

FashionMNIST[31]とは, T シャツやズボンなどの洋服の画像データとその洋服のクラスを表すラベルデータがセットとなったデータセットである．クラスとラベルの対応関係を表 4.1 に示す．MNIST と同様，学習用に 60000 枚，検証用に 10000 枚用意されている．画像のサイズは 28×28 pixels であり，白黒画像である．本実験では，学習用の画像 60000 枚を使用した.

表 4.1　FashionMNIST のクラスとラベルの対応関係

ラベル	クラス
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

4.2.4 猫画像

本実験では，白黒画像である MNIST および FashionMNIST の他に，カラー画像でも実験を行うため猫の画像を使用した．オックスフォード大学が公開している動物画像データセット，”The Oxford-IIIT Pet Dataset”[32] から猫の画像 4978 枚を使用した．また，画像数を増やすためクローリングを行い，総画像数を 7836 枚とした．画像のサイズは，64×64 pixels に正規化して使用した．

4.3 実験に使用した GAN

本実験では，CNN を用いた GAN と Capsule Network を用いた 3 種類の GAN の計 4 種類の GAN を用いて実験を行った．Capsule Network を用いた GAN は，2.4.5 節で述べた関連研究である Capsule GAN1 と，第 3 章で提案した Capsule GAN2 および Capsule GAN3 である．CNN を用いた GAN は DCGAN である．本実験で用いた GAN の概略図を図 4.1 に示す．また，猫画像においてはそれぞれに WGAN-gp の手法を用いた場合で同様に比較実験を行った. 28×28 pixels の白黒画像を生成する場合の DCGAN の構造を表 4.2, Capsule GAN1 の構造を表 4.3，Capsule GAN2 の構造を表 4.4，Capsule GAN3 の構造を表 4.5 にそれぞれ示す．表 4.5 に示す Capsule GAN3 の構造は，3.2.2.2 節で提案した一つの重み行列を全生成クラスで共有する場合の構造である．

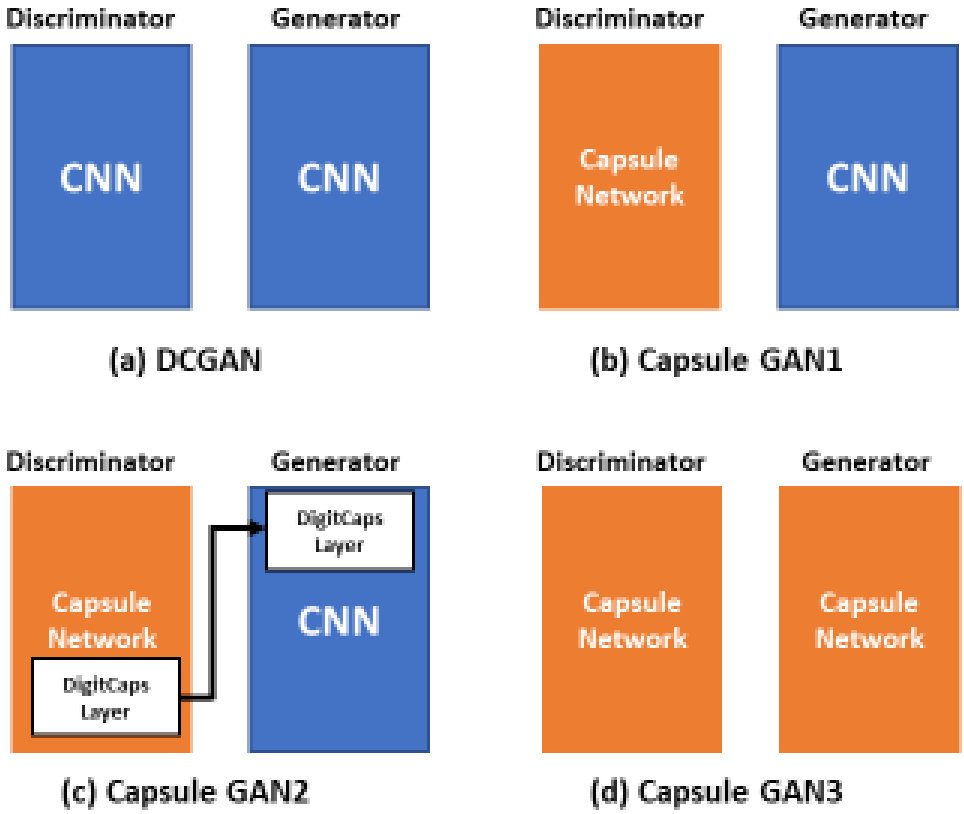

図 4.1　実験に用いた GAN の概略図

表 4.2　DCGAN の構造

Discriminator		Generator	
層	出力サイズ	層	出力サイズ
Convolution, LeakyRelu	32×14×14	Dense	6272
Dropout	32×14×14	Reshape	128×7×7
Convoluiton, Batch Normalization, LeakyRelu	64×8×8	Deconvolution, Batch Normalization, Relu	128×14×14
Dropout	64×8×8	Deconvolution, Batch Normalization, Relu	64×28×28
Convolution, Batch Normalization, LeakyRelu	128×4×4	Convolution, Tanh	1×28×28
Dropout	128×4×4		
Convolution, Batch Normalization, LeakyRelu	256×4×4		
Dropout	256×4×4		
Dense	1		

表 4.3　Capsule GAN1 の構造

Discriminator		Generator	
層	出力サイズ	層	出力サイズ
Convolution, LeakyRelu	32×14×14	Dense	6272
Dropout	32×14×14	Reshape	128×7×7
Convolution, Batch Normalization, LeakyRelu	64×8×8	Deconvolution, Batch Normalization, Relu	128×14×14
Dropout	64×8×8	Deconvolution, Batch Normalization, Relu	64×28×28
Convolution, Batch Normalization, LeakyRelu	128×4×4	Convolution, Tanh	1×28×28
Dropout	128×4×4		
Convolution, Batch Normalizaion, LeakyRelu	256×4×4		
Dropout	256×4×4		
Dense	1		

表 4.4　Capsule GAN2 の構造

Discriminator		Generator	
層	出力サイズ	層	出力サイズ
Convolution, LeakyRelu	256×20×20	Multiply, Batch Normalization, LeakyRelu	16×100
Primary, squash	256×6×6	Dense, BN, LeakyRelu	100
DigitCaps	16×10	Dense	6272
Mask	16	Reshape	128×7×7
Dense	1	Deconvolution, Batch Normalization, Relu	128×14×14
		Deconvolution, Batch Normalization, Relu	64×28×28
		Convolution, Tanh	1×28×28

表 4.5　Capsule GAN3 の構造

Discriminator		Generator	
層	出力サイズ	層	出力サイズ
Convolution, LeakyRelu	256×20×20	Reshape	8×16
Primary, squash	256×6×6	DigitCaps	8×1152
DigitCaps	16×10	Reshape	256×6×6
Mask	16	Deconvolution, Batch Normalization, Relu	256×16×16
Dense	1	Deconvolution, Batch Normalization, Relu	128×20×20
		Deconvolution, Batch Normalization, Relu	64×24×24
		Deconvolution, Tanh	1×28×28

4.4 実験

4.4.1 実験の概要

本研究では，二つの実験を行った．一つ目の実験は，従来手法との比較実験である．本実験では，4.3 節で述べた 4 種類の GAN を用いて MNIST，FashionMNIST および猫画像を訓練データセットとして画像の生成を行った．また，生成画像を Inception Score および Geometry Score を用いて評価した．

二つ目の実験は，3.2.2.3 節で述べた各生成クラスにそれぞれ重み行列を用いた Capsule GAN3 の検証実験である．ラベルごとに画像が生成されているかを確認する．

4.4.2 従来手法との比較実験

4.4.2.1 実験概要

MNIST，FashionMNIST および猫画像を用いて，4.3 節で述べた 4 種類の GAN による生成画像の比較実験を行った．また，Inception Score および Geometry Score を用いて生成画像の品質を評価した．

4.4.2.2 MNIST を用いた実験

4.4.2.2.1 実験結果

MNIST を用いて画像の生成および評価を行った．4 種類の GAN はそれぞれ収束するまで学習した．生成画像の結果を図 4.2，図 4.3，図 4.4 および図 4.5 に示す.

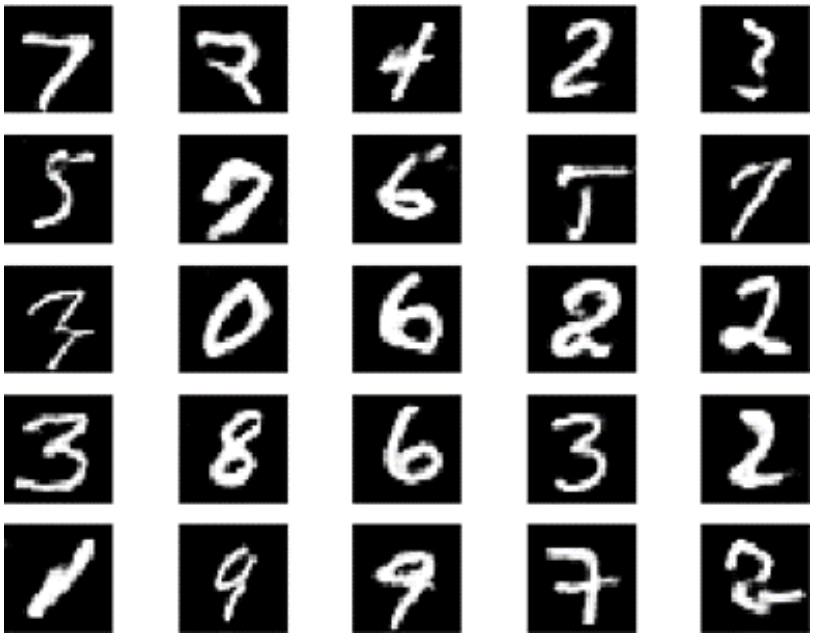

図 4.2　MNIST の生成画像（DCGAN）

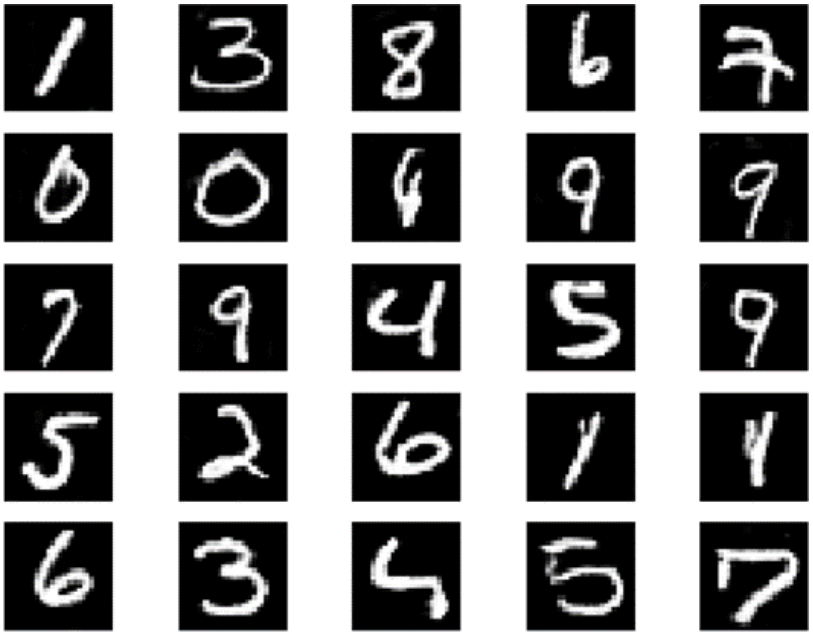

図 4.3　MNIST の生成画像（Capsule GAN1）

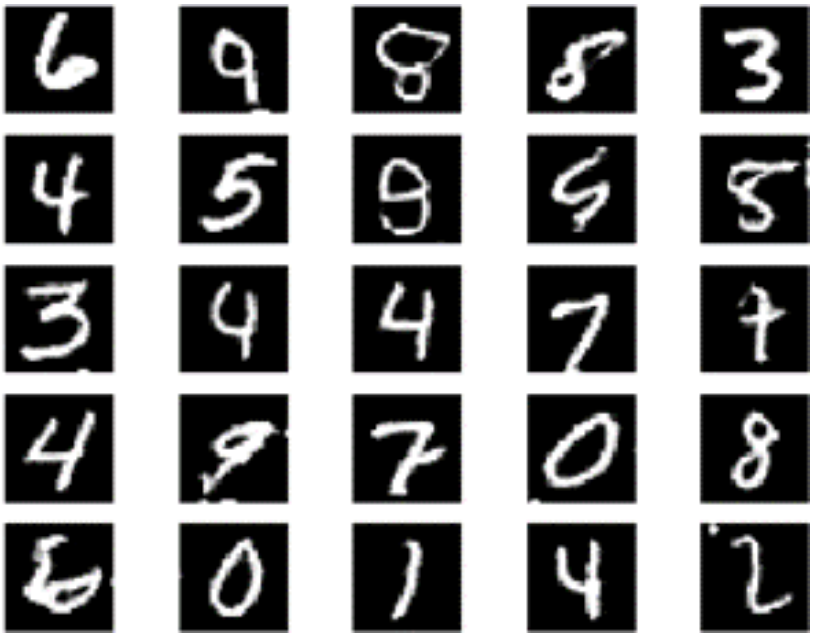

図 4.4　MNIST の生成画像（Capsule GAN2）

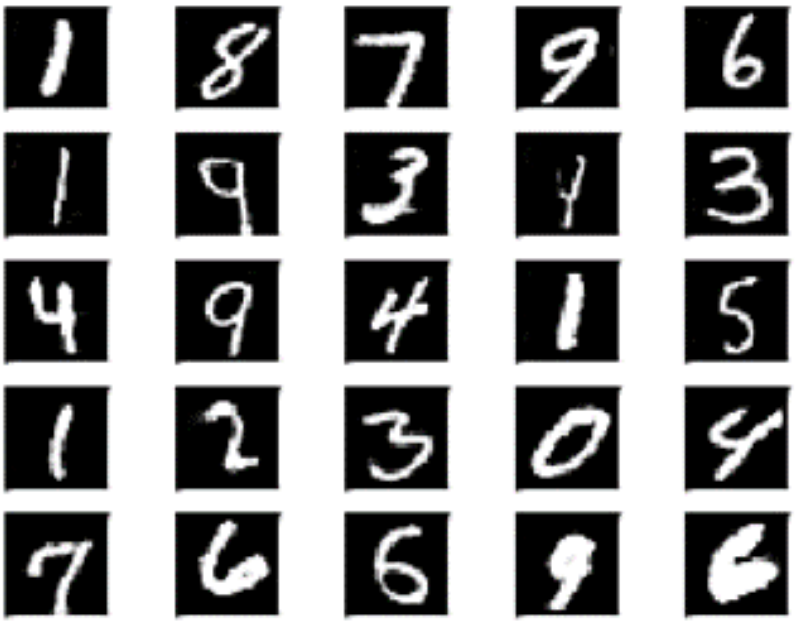

図 4.5　MNIST の生成画像（Capsule GAN3）

4.4.2.2.2 評価および考察

また, 生成画像から Inception Score および Geometry Score を計算し評価した. Inception Score には画像を 10000 枚，Geometry Score には画像を 1000 枚用いた．結果を表 4.6 に示す.

表 4.6　評価結果（MNIST）

	DCGAN	Capsule GAN1	Capsule GAN2	Capsule GAN3
Inception Score	2.32	2.35	2.37	**2.57**
Geometry Score (×100)	3.89	5.01	3.48	**1.46**

図 4.2，図 4.3，図 4.4 および図 4.5 より生成画像を比較する．4 種類の GAN は，それぞれ数字を生成できていることが確認できる．目視での比較では，生成画像の品質に大きな差は見られない．これは，MNIST は GAN において比較的生成しやすい画像であることが理

由として考えられる．そのため，どの GAN においても数字の生成に成功した．

表 4.6 より Inception Score および Geometry Score を比較する．Inception Score では，Capsule Network を用いた 3 種類の GAN 全てが CNN を用いた GAN である DCGAN よりも評価値が良い結果となった．Inception Score は，Inception モデルで識別しやすい画像ほど画像の品質が良いことを表す．したがって，MNIST においては Capsule Network を使用した GAN の方が，CNN を用いた GAN よりも品質の良い画像を生成できることがわかる．Geometry Score を比較すると，Capsule GAN3，Capsule GAN2，DCGAN，Capsule GAN1 の順で評価値が良い結果となった．Capsule GAN3 では DCGAN よりも 0.0243 良い結果となった．Geometry Score は画像の分布を比較し，モード崩壊を特定する．Capsule GAN1 は他の 3 種類の GAN と比較すると似たような画像を生成したと考えられる．

Inception Score および Geometry Score の結果から，MNIST においては Capsule Network を Generator 用いた Capsule GAN3 が一番品質の良い画像を生成できたことがわかる．したがって，Capsule Network を用いた GAN の方が，CNN を用いた GAN よりも MNIST の画像を上手く生成できたといえる．

4.4.2.3 FashionMNIST を用いた実験

4.4.2.3.1 実験結果

FashionMNIST を用いて画像の生成および評価を行った．4 種類の GAN は MNIST 同様それぞれ収束するまで学習した．生成画像の結果を図 4.6，図 4.7，図 4.8 および図 4.9 に示す．

図 4.6　FashionMNIST の生成画像（DCGAN）

図 4.7　FashionMNIST の生成画像（Capsule GAN1）

図 4.8　FashionMNIST の生成画像（Capsule GAN2）

図 4.9　FashionMNIST の生成画像（Capsule GAN3）

4.4.2.3.2 評価および考察

また，生成画像から Inception Score および Geometry Score を計算し評価した．使用した画像枚数は 4.4.2.2.2 節と同様である．結果を表 4.7 に示す.

表 4.7　評価結果（FashionMNIST）

	DCGAN	Capsule GAN1	Capsule GAN2	Capsule GAN3
Inception Score	4.39	4.34	4.48	**4.54**
Geometry Score (×100)	0.123	0.129	0.205	**0.103**

図 4.6，図 4.7，図 4.8 および図 4.9 より生成画像を比較する．図 4.6 より DCGAN の生成画像は Capsule Network を用いた GAN の生成画像と比較すると洋服の形が崩れていることがわかる．また，sandal や bag などの衣服以外の画像の生成が少ない，図 4.7 より Capsule GAN1 の生成画像について考察する. 図 4.6 の DCGAN の生成画像と比較すると，洋服の輪郭がはっきりと生成できていることがわかる．生成している洋服の種類は，DCGAN と同様，衣服以外の画像の生成が少ない．図 4.8 より Capsule GAN2 の生成画像について考察する．Capsule GAN1 と同様，洋服の輪郭をはっきりと生成している．生成している洋服の種類は衣服以外の生成が多くなっている．図 4.9 より Capsule GAN3 の生成画像について考察する．Capsule GAN1 および Capsule GAN2 と同様，洋服の輪郭をはっきりと生成できていることがわかる．また，他の 3 種類の GAN に比べて様々な種類の洋服をバランス良く生成できていることがわかる.

表 4.7 より Inception Score および Geometry Score を比較する. MNIST の場合と同様，Inception Score においては Capsule Network を用いた 3 種類の GAN 全てが DCGAN よりも評価値が良い結果となった. したがって, FashionMNIST においても Capsule Network を使用した GAN の方が CNN を用いた GAN よりも品質の良い画像を生成できることがわかる．次に Geometry Score を比較する．Geometry Score は Capsule GAN3，Capsule GAN1，DCGAN，Capsule GAN2 の順で評価値が良い結果となった．目視での生成画像の種類の分布の比較結果と同じく，Capsule GAN3 が一番バランスよく洋服の種類を生成していることがわかる.

Inception Score および Geometry Score の結果から, FashionMNIST においても Capsule GAN3 が一番品質の良い画像を生成できたことがわかる．したがって，Capsule Network を用いた GAN の方が，CNN を用いた GAN よりも FashionMNIST の画像を上手く生成できたといえる.

4.4.2.4 猫画像を用いた実験

4.4.2.4.1 実験概要

カラー画像の猫画像を用いて画像の生成および評価を行った．Capsule GAN3 は，パラメータが収束せず画像の生成を行うことが出来なかった．そのため，GAN の安定化手法の一つである WGAN-gp の手法を取り入れて実験を行った．まず，WGAN-gp の手法を使用しない場合の実験結果を 4.4.2.4.2 節に示す．これは DCGAN，Capsule GAN1 および Capsule GAN2 の 3 種類の GAN のみの実験結果である．次に，WGAN-gp の手法を使用した場合の結果を 4.4.2.4.3 節に示す．4 種類の GAN すべてに WGAN-gp の手法を使用した．そのため，比較する CNN を用いた GAN は DCGAN ではなく WGAN-gp である．

4.4.2.4.2 WGAN-GP の手法を使用していない場合

4.4.2.4.2.1 実験結果

DCGAN, Capsule GAN1 および Capsule GAN2 を用いて画像を生成した. 3 種類の GAN はそれぞれ収束するまで学習した．生成画像の結果を図 4.10，図 4.11 および図 4.12 に示す.

図 4.10　猫画像の生成画像（DCGAN）

図 4.11　猫画像の生成画像（Capsule GAN1）

図 4.12　猫画像の生成画像（Capsule GAN2）

4.4.2.4.2.2 考察

図 4.10, 図 4.11 および図 4.12 より生成画像を比較する．生成画像を目視で比較すると，Capsule GAN2 が一番猫の形を捉えながら生成できていることがわかる．特に，目の形および位置が他の二つの GAN よりも上手く生成できている．これは，Capsule Network の画像のパーツの位置関係を保持しながら処理を行う点が影響していると考えられる．

Capsule GAN3 は学習が安定しなかった．これは，Capsule GAN3 が他の 3 種類の GAN よりも学習パラメータ数が多く，学習が収束するのが難しいためであると考えられる．白黒画像の MNIST および FashionMNIST に比べ，猫画像はカラー画像でありパラメータ数が多くなる．また，画像のサイズも 28×28 pixels から 64×64 pixels と大きい．そのため，同様にパラメータ数が多くなる．これらも Capsule GAN3 のパラメータの収束に影響を与えたと考えられる．

MNIST，FashionMNIST と同じように 4 種類の GAN で比較するには，Capsule GAN3 でも猫画像を生成する必要がある．そのため，WGAN-gp に用いられている安定化手法を取り入れて Capsule GAN3 で実験を行った．また，比較のために他の 3 種類の GAN にも同様に安定化手法を取り入れて再度猫画像を生成した．結果を 4.4.2.4.3 節に示す．

4.4.2.4.3 WGAN-GP の手法を使用した場合

4.4.2.4.3.1 実験結果

WGAN-gp と WGAN-gp の手法を適用した Capsule GAN1，Capsule GAN2 および Capsule GAN3 を用いて画像を生成した．4 種類の GAN はそれぞれ収束するまで学習した．生成画像の結果を図 4.13，図 4.14，図 4.15 および図 4.16 に示す．

図 4.13　猫画像の生成画像（WGAN-gp）

図 4.14　猫画像の生成画像（Capsule GAN1，WGAN-gp の手法を適用）

図 4.15　猫画像の生成画像（Capsule GAN2，WGAN-gp の手法を適用）

図 4.16　猫画像の生成画像（Capsule GAN3，WGAN-gp の手法を適用）

4.4.2.4.3.2 評価および考察

また，生成画像から Inception Score および Geometry Score を計算し評価した．使用した画像枚数は 4.4.2.4.2 節と同様である．結果を表 4.8 に示す.

表 4.8　評価結果（猫画像，WGAN-gp の手法を適用）

	WGAN-gp	Capsule GAN1	Capsule GAN2	Capsule GAN3
Inception Score	4.17	4.53	4.68	**4.72**
Geometry Score (×100)	0.192	0.221	3.84	**0.106**

図 4.13，図 4.14，図 4.15 および図 4.16 より生成画像を比較する．WGAN-gp の手法を用いることで，4.4.2.4.2 節では学習が収束しなかった Capsule GAN3 でも画像を生成することができた．生成画像を目視で比較すると，Capsule Network を用いた 3 種類の GAN の方が CNN を用いた GAN である WGAN-gp よりも猫の形を捉えながら生成できていることがわかる．また，Capsule GAN2 および Capsule GAN3 は，猫の顔および目の輪郭を比較的崩すことなく生成できていることがわかる.

表 4.8 より Inception Score および Geometry Score を比較する．Inception Score は Capsule GAN3，Capsule GAN2，Capsule GAN1，WGAN-gp の順で評価値が良い結果となった．これは目視での比較結果と一致する．このことから，猫画像においても Capsule Network を用いた GAN の方が CNN を用いた GAN よりも品質の良い画像を生成できることがわかる. 次に Geometry Score を比較する. Geometry Score は Capsule GAN3, WGAN-gp，Capsule GAN1，Capsule GAN2 の順で評価値が良い結果となった．Capsule GAN2 の評価値が他の 3 種類の GAN と大きく差が出ている．これは，Capsule GAN2 が画像を生成する際に DigitCaps 層を利用している点が影響していると考えられる．3.2.1 節より，Capsule GAN2 は一つの DigitCaps 層の値を 2 回繰り返し使用している．同じ DigitCaps 層の値を使用しても，全く同じ画像が生成されることはない．これは値を乱数と掛け合わせるためである．しかしながら，似たような画像は生成されやすい．また，猫画像は MNIST や FashionMNIST と違いはっきりとしたクラス分けがない．そのため，Capsule GAN2 の同じ DigitCaps 層を 2 回使用する点が Geometry Score に影響したと考えられる. しかし，前述したとおり猫画像ははっきりとしたクラス分けがない．したがって，MNIST および FashionMNIST の場合と比べると Geometry Score は重視する必要がないと考える. また，WGAN-gp は Geometry Score は 2 番目に良い評価値を出しているが, 目視および Inception

Score の結果が一番低い．この点からも，猫画像においては Geometry Score を重視する必要がないと言える．

Inception Score および Geometry Score の結果から，猫画像においても Capsule GAN3 が一番品質の良い画像を生成できたことがわかる．したがって，Capsule Network を用いた GAN の方が，CNN を用いた GAN よりも猫画像を上手く生成できたといえる．

4.4.3 Capsule GAN3 におけるラベルを使用する場合の検証実験

4.4.3.1 実験概要

3.2.2.3 節で述べた生成クラスごとに重み行列を使用する Capsule GAN3 の構造の検証実験を行った．実際にラベルが示す生成クラスを生成できるかを検証した．使用したデータセットは MNIST および FashionMNIST の二つである．

4.4.3.2 MNIST を用いた実験結果

3.2.2.2 節で述べた生成クラスごとに重み行列を使用する Capsule GAN3 で MNIST の画像を生成した．生成画像を図 4.17 に示す．図 4.17 に示す結果は，行ごとに使用している重み行列が違う結果を示す．

図 4.17　MNIST の生成画像（生成クラスごとに重み行列を使用した場合）

4.4.3.3 FashionMNIST を用いた実験結果

4.4.3.2 節同様, 3.2.2.2 節で述べた生成クラスごとに重み行列を使用する Capsule GAN3 で FashionMNIST の画像を生成した．生成画像を図 4.18 に示す．図 4.17 と同様，図 4.18 に示す結果は 行ごとに使用している重み行列が違う結果を示す.

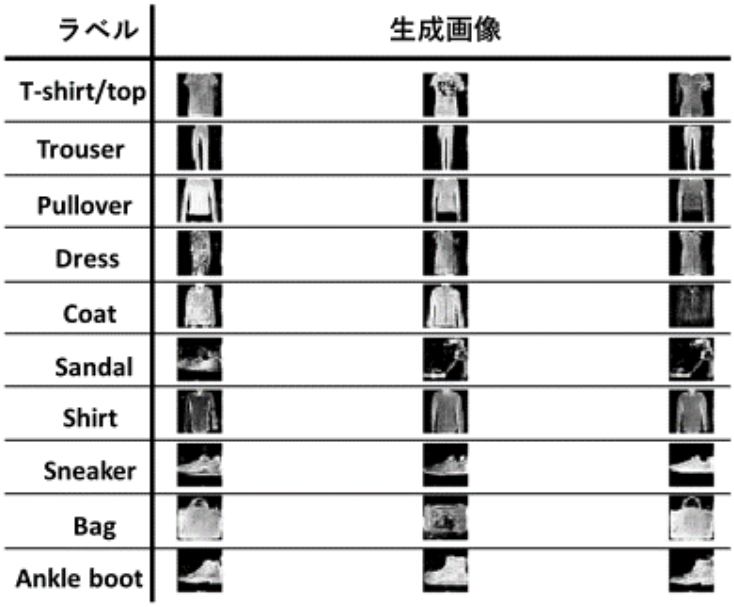

図 4.18　FashionMNIST の生成画像（生成クラスごとに重み行列を使用した場合

4.4.3.4 考察

図 4.17 および図 4.18 より考察する．生成結果より，MNIST および FashionMNIST のどちらの場合においてもラベルが示す生成クラスごとに画像を生成していることがわかる．このことから，3.2.2.2 節で述べた Capsule GAN3 の構造は生成クラスごとに重み行列を使用できていることが確認できる.

4.5 むすび

本章では，第 3 章の提案手法の評価実験の概要，結果および考察について述べた.

第5章　結論と今後の課題

5.1 結論

本研究では，Capsule Network を Discriminator および Generator に用いた GAN である Capsule GAN を 2 種類提案した．

CNN を用いた GAN と従来の Capsule Network を用いた GAN と比較したところ，本論文で提案した Capsule GAN3 が実験で用いた全てのデータセットにおいて一番品質の良い画像を生成した．比較実験の結果から，Capsule Network を用いた GAN の方が CNN を用いた GAN よりも品質の良い画像を生成できると言える．

また，Capsule GAN においても CNN を用いた Conditional GAN 同様，ラベルを使用することでクラスごとの画像生成が可能であることが確認できた．クラスごとに画像を生成できることで，CNN を用いた GAN で提案されている従来手法をより多く適用することが可能となる．

5.2 今後の課題

今後の課題として二つ挙げられる．

一つ目は，学習の安定化である．パラメータ数が多いため，学習の収束が難しい点がある．今回は WGAN-gp の手法を取り入れることで学習の収束に成功した．しかしながら，全ての画像生成で同様の結果が得られるとは限らない．そのため，Capsule GAN に適した学習の安定化手法を見つける必要がある．

二つ目は，高解像度の画像生成である．本論文では 64×64 pixels の画像生成までしか行っていない．そのため，より高解像度の画像生成が可能かを検証する必要がある．

第6章　謝辞

本研究に際して，コロナ下という不安定な状況の中，研究テーマに対して熱心かつ丁寧に指導してくださった渡辺教授に心より深く感謝申し上げます．

また，日頃から研究の問題点に対して相談に乗っていただき，アドバイスをくださった研究室の皆様に心より感謝申し上げます．

最後に，常に心身ともに支えながら私をここまで育ててくださり，学費を工面してくださった家族に対して深く感謝致します．

第7章　参考文献

[1] A. Krizhevsky, I. Sutskever and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, Neural Information Processing Systems (NIPS), pp. 1106-1114, Dec. 2012.

[2] Q. Le, M. Ranzato, R. Monga, M. Devin, G. Corrado, K. Chen, J. Dean and A. Ng, “Building High-level Features Using Large Scale Unsupervised Learning”, International Conference on Machine Learning (ICML), pp. 81-88, June 2012.

[3] S. Iizuka, E. Simo-Serra and H. Ishikawa, “Let there be Color!：Joint End-to end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification”, ACM Transactions on Graphics (TOG), 35(4), July 2016.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative Adversarial Networks”, Neural Information Processing Systems (NIPS), pp. 2672-2680, Dec. 2014.

[5] KIOXIA, “#世界新記憶 01 TEZUKA2020”, https://tezuka2020.kioxia.com/ja-jp/, (2020年1月現在)

[6] S. W. Kim, Y. Zhou, J. Philion, A. Torralba and S. Fidler, “Learning to Simulate Dynamic Environments with GameGAN”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1231-1240, June 2020.

[7] S. Sabour, N. Frosst and G. E. Hinton, “Dynamic Routing Between Capsules”, Neural Information Processing Systems (NIPS), pp. 3859-3869, Dec. 2017.

[8] A. Radford, L. Metz and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, International Conference on Learning Representations (ICLR), Jan. 2016.

[9] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets”, arXiv preprint arXiv: 1411.1784, 2014.

[10] A. Odena, C. Olah and J. Shlens, “Conditional Image Synthesis with Auxiliary Classifier GANs”, International Conference on Machine Learning (ICML), pp. 2642-2651, Aug. 2017.

[11] M. Arjovsky, S. Chintala and L. Bottou, “Wasserstein GAN”, International Conference on Machine Learning (ICML), pp. 214-223, Aug. 2017.

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. Courville, “Improved Training of Wasserstein GANs”, Neural Information Processing System (NIPS), pp.

5769-5779, Dec. 2017.

[13] H. Gadirov, M. Tamošiūnaité and D. Vitkute-Adzgauskiene, “Capsule Architecture as a Discriminator in Generative Adversarial Networks”, Vytautas Magnus University, Feb. 2018, M. D. thesis.

[14] A. Jaiswal, W. AbdAlmageed, Y. Wu and P. Natarajan, “CapsuleGAN: Generative Adversatial Capsule Network”, European Conference on Computer Vision (ECCV), pp-526-535, Sep. 2018.

[15] Y. LeCun, C. Cortes and C. J. C. Burges, “The MNIST Database of Handwritten Digits”, http://yann.lecun.com/exdb/mnist/, 1998.

[16] Alex Krizhevsky, “Convolutional deep belief networks on CIFAR-10”, Aug. 2010.

[17] K. Marusaki and H. Watanabe, “A Study on GAN using Capsule Network”, IEICE General Conference, D-12-8, Mar. 2019.

[18] 斎藤康毅, ゼロから作る Deep Learning―Python で学ぶディープラーニングの理論と実装ー, オライリー・ジャパン, p.205, 2016.

[19] 清水亮, “深層学習を根底から覆すカプセルネットワークの衝撃”, https://wirelesswire.jp/2018/03/64233/, (2020 年 1 月現在)

[20] S. Sabour, N. Frosst and G. E. Hinton, “Dynamic Routing Between Capsules”, Neural Information Processing Systems (NIPS), pp. 3859-3869, Dec. 2017.

[21] 株式会社システムインテグレータ, “敵対的生成ネットワーク_GAN (Vol.19)”, https://products.sint.co.jp/aisia/blog/vol1-19, (2020 年 1 月現在)

[22] Yuki Shinya, “はじめての GAN”, https://elix-tech.github.io/ja/2017/02/06/gan.html, （2019 年 1 月現在）

[23] M. Arjovsky, S. Chintala and L. Bottou, “Wasserstein GAN”, International Conference on Machine Learning (ICML), pp. 214-223, Aug. 2017.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. Courville, “Improved Training of Wasserstein GANs”, Neural Information Processing System (NIPS), pp. 5769-5779, Dec. 2017.

[25] H. Gadirov, M. Tamošiūnaité and D. Vitkute-Adzgauskiene, “Capsule Architecture as a Discriminator in Generative Adversarial Networks”, Vytautas Magnus University, Feb. 2018, M. D. thesis.

[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, “Improved Techniques for Training GANs”, Neural Information Processing Systems (NIPS), pp.2234-2242, Dec. 2016.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, “Going Deeper with Convolutions”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1-9, June 2015.

[28] “Inception score”, http://bluewidz.blogspot.com/2017/12/inception-score.html https://elix-tech.github.io/ja/2017/02/06/gan.html, (2020 年 1 月現在)

[29] V. Khrulkov and I. Oseledets, “Geometry Score: A Method for Comparing Generative Adversarial Networks”, International Conference on Machine Learning (ICML), , pp. 2621-2629, Jul. 2018.

[30] S. Rifai, P. Vincent, X. Muller, X. Glorot and Y. Bengio, “Contrative Auto-Encoders: Explicit Invariance During Feature Extraction”, International Conference on Machine Learning (ICML), pp. 833-840, June 2011.

[31] H. Xiao, K. Rasul and R. Vollgraf, “Fashion-MNIST: A Novel Image Dataset”, arXiv preprint arXiv: 1708.07747, 2017.

[32] O. M. Parkhi, A. Vedaldi, A. Zisserman and C. V. Jawahar, “Cats and Dogs”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3498-3505, June 2012.

第8章　図一覧

第9章　表一覧

第10章　研究業績

[1] 松田, 丸嵜, 渡辺, ”生成画像品質を考慮した CapsGAN によるデータ拡張”, 映像情報メディア学会冬季大会, 12D-4, Dec. 2018.

[2] 丸嵜, 渡辺, ”A Study on GAN Using Capsule Network”, 電子情報通信学会総合大会, D-12-8, Mar. 2019.

[3] 丸嵜, 渡辺, ”Capsule Network を生成器に使用した敵対的生成ネットワークの一検討”, 2019 年画像符号化シンポジウム・2019 年映像メディア処理シンポジウム (PCSJ/IMPS2019), P-1-03, Nov. 2019.