

修士論文概要書

Summary of Master's Thesis

Date of submission: (01/29/2020)

専攻名(専門分野) Department	情報理工・ 情報通信専攻	氏名 Name	松田 卓也	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	5117F082-1		
研究題目 Title	画像品質及びデータ分布を考慮したデータ拡張 Data Augmentation Considering Image Quality and Data Distribution				

1. まえがき

近年、深層学習は様々な商品・サービスに導入されてきているが、その精度を最大限高めるには一般的に大量のデータセットが必要である。一方、画像生成技術も最近著しく発展を遂げており、新しい画像を生成することが可能になった。この技術を用いることで、少ないデータセットでもデータ拡張によって深層学習の精度を高めることが期待される。

本研究では、良質な画像を生成してデータ拡張する手法、同一クラス内のデータ分布の偏りを低減してデータ拡張する手法、他クラス間のデータ分布の偏りを低減してデータ拡張する手法を提案する。提案した手法によってクラス分類の精度を向上することを目的とする。

2. 関連技術

2.1. Generative Adversarial Capsule Network (CapsuleGAN)

Convolutional Neural Network (CNN)のプーリング層における空間的な位置情報の喪失を解決するために、Hinton らは Capsule Network[1]を提案した。Capsule Network はプーリング層を減らし、特徴フィルタを複数枚ごとにグループ化してカプセル単位で学習することで、特徴間の位置関係を学習することができる。Jaiswal らはこの Capsule Network を Generative Adversarial Network (GAN)の鑑別器に導入したモデルである CapsuleGAN[2]を提案し、特徴間の正しい画像を生成することを可能とした。

2.2. Variational Auto Encoder (VAE)

VAE[3]とは、ニューラルネットワークを用いた次元削減アルゴリズムの一つの手法であり、潜在変数に正規分布を導入したものである。潜在変数を連続的に変えることで、連続的に変化する画像を生成することができる。

2.3. Adversarial Auto Encoder (AAE)

AAE[4]とは、Auto Encoder(AE)と GAN を組み合わせることで、AE 内の潜在変数を任意の確率分布に学習させることができるアルゴリズムの一つの手法である。VAE とは異なり、潜在変数のクラスタを分離することが可能になった。

2.4. Synthesize Minority Over-sampling Technique (SMOTE)

SMOTE[5]とは、オーバーサンプリング手法の一つであり、不均衡データにおける問題を解決する手法として用いられる。この手法では、少数クラスのサ

ンプルを複数選択し、そのサンプル間に新たなサンプルを生成する。

3. 提案手法

3.1. CapsuleGAN を用いたデータ拡張

CapsuleGAN で生成した画像を用いてデータ拡張を行う。従来の CapsuleGAN では、畳み込み層を通して得られた N チャンネルの特徴マップから隣接し合う特徴マップを M チャンネルずつグループ化してカプセルに変換していた。しかし、畳み込みフィルタの初期値のランダム性により、特徴間の位置関係を保持するためには、隣接していない特徴マップについてもカプセル化する必要があると考えられる。そこで、Primary Caps 層の入力である特徴マップの集合 $A = \{a_n | 1 \leq n \leq N\}$ をランダム置換 σ によって並び替えた特徴マップの集合を B とすると、 B は

$$B = \{b_n | 1 \leq n \leq N\} \quad (1)$$

$$b_n = \sigma(a_n)$$

と表せる。

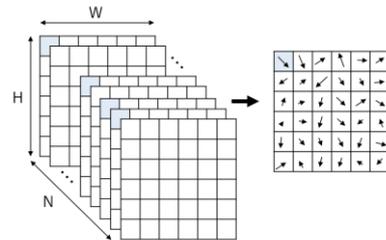


図1 特徴マップのランダムな選択

さらに、カプセル c_k は

$$c_k = \begin{pmatrix} b_{1ij} \\ b_{2ij} \\ \vdots \\ b_{Mij} \end{pmatrix}, 1 \leq k \leq \frac{N \times W \times H}{M}, 1 \leq i \leq H, 1 \leq j \leq W \quad (2)$$

と表せる。従来のカプセル化にこの手法を加えることで、従来の CapsuleGAN で保持できなかった位置関係も保持できると考えられる。

3.2. VAE における潜在変数操作によるデータ拡張

VAE の潜在変数を操作することで、同一クラスデータの分布の偏りを考慮してデータ拡張を行う。まず、サンプル i の N 次元潜在変数を s_i 、各サンプルと原点とのユークリッド距離を D_i としたときの最大距離 D_{MAX} を求める。生成するサンプル数を J としたとき、半径 D_{MAX} の超球体内に一様乱数 $R = \{r_j | 1 \leq j \leq J\}$ を生成する。 R を Decoder に通すことで、画像が得られる。

3.3. AAE と SMOTE によるデータ拡張

AAE の潜在変数に対して SMOTE を用いたオーバーサンプリングを適用することデータ拡張を行う。AAE の Discriminator には真のデータとして 10 個のガウス分布を入力し、偽のデータとして潜在変数を入力する。ある少数クラスの画像データの集合 $\mathbf{X} = \{\mathbf{x}_m | 1 \leq m \leq M\}$ を学習済み AAE に入力したとき、潜在変数の集合 $\mathbf{Z} = \{\mathbf{z}_m | 1 \leq m \leq M\}$ が得られる。 \mathbf{Z} を Decoder に通すことで、画像が得られる。

4. 評価実験

4.1. CapsuleGAN を用いたデータ拡張

以下、実験では MNIST をデータセットとして使用する。CapsuleGAN による生成画像を図 2 上段に示す。また、CapsuleGAN を用いたデータ拡張の評価をするために、CNN モデルを用いてデータセットのクラス分類を行った。さらに、データ拡張しない場合と従来の CapsuleGAN でデータ拡張した場合と同様にクラス分類を行い、比較した。元のデータセットの枚数を 100 枚として、データ拡張でさらに 100 枚生成する。実験結果を表 1 に示す。表 1 より、データ拡張の有効性は確認できるが、従来手法と提案手法は同程度であることがわかる。

4.2. VAE における潜在変数操作によるデータ拡張

生成画像を図 2 中段に示す。また、CNN モデルに元のデータセットのみを学習させた場合と生成画像のみを学習させた場合のクラス分類精度を比較した。その結果を表 2 に示す。表 2 より、少数データセットにおいて、生成画像のみを学習すると、クラス分類精度が向上することが確認できる。

4.3. AAE と SMOTE によるデータ拡張

生成画像を図 2 下段に示す。また、5 クラスを少数クラスとして各データセット数を 50 枚、残り 5 クラスを多数クラスとして各データセット数を 500 枚とする。このとき、データ拡張しない場合、DCGAN でデータ拡張した場合、AAE と SMOTE によるデータ拡張した場合の比較をする。また、データ拡張では、50 枚から 450 枚の画像を新たに生成するものとする。実験結果を表 3 に示す。表 3 より、提案手法によるデータ拡張が有効であることが確認できる。

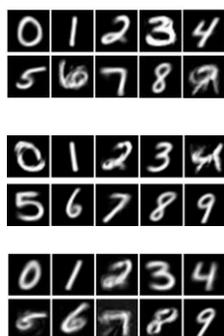


図 2 生成画像(上段: CapsuleGAN, 中段: VAE, 下段: AAE)

表 1 クラス分類精度[%]

	クラス									平均	
	0	1	2	3	4	5	6	7	8		9
データ拡張なし	97	98	87	86	89	85	92	89	83	84	89
従来手法	93	94	90	87	92	93	91	91	83	84	90
提案手法	98	96	87	90	89	88	91	87	90	88	90

表 2 クラス分類精度[%]

		各クラスのデータセット数		
		50	500	1000
学習画像	データセット	65	93	95
	生成画像	74	69	71

表 3 クラス分類精度[%]

	Precision	Recall
データ拡張なし	88	86
DCGAN	90	89
提案手法	91	91

5. むすび

全クラスのデータセットが少なく生成画像の品質を考慮した場合、クラス内においてデータ分布の偏りがある場合、クラス間でデータ分布の偏りがある場合それぞれに対して解決する手法を提案した。提案した CapsuleGAN は強力な Discriminator を持ち、似た画像が多く生成されてしまうことが確認できた。また、データ分布を操作することによるデータ拡張の有効性を確認した。

参考文献

- [1] Sara Sabour, Nicholas Frosst, Geoffrey E Hinton, "Dynamic Routing Between Capsules," Neural Information Processing Systems (NIPS), 2017
- [2] Ayush Jaiswal, Wael AbdAlmageed, Yue Wu, Premkumar Natarajan, "CapsuleGAN: Generative Adversarial Capsule Network," The European Conference on Computer Vision (ECCV) Workshops, pp.0-0, 2018
- [3] Diederik P. Kingma, Max Welling, "Auto-Encoding Variational Bayes," International Conference on Learning Representation, 2014
- [4] Alireza Makhzami, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, Brendan Frey, "Adversarial autoencoders," arXiv preprint arXiv:1511.05644, 2015
- [5] N. C. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research 16, pp.321-357, 2002

2019 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

画像品質及びデータ分布を考慮したデータ拡張

Data Augmentation Considering Image Quality and Data Distribution

松田 卓也

(5117F082-1)

提出日：2020.01.29

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
1.1 研究背景.....	1
1.2 研究目的.....	2
1.3 本論文の構成.....	2
第 2 章 関連技術.....	3
2.1 まえがき.....	3
2.2 データ拡張.....	3
2.3 CapsuleGAN.....	3
2.3.1 GAN.....	3
2.3.2 Capsule Network.....	4
2.3.3 CapsuleGAN.....	7
2.4 AE.....	8
2.4.1 AE.....	8
2.4.2 VAE.....	9
2.4.3 AAE.....	10
2.5 SMOTE.....	11
第 3 章 提案手法.....	12
3.1 まえがき.....	12
3.2 CapsuleGAN を用いたデータ拡張.....	12
3.3 VAE における潜在変数操作によるデータ拡張.....	14
3.4 AAE と SMOTE を用いたデータ拡張.....	14
3.5 むすび.....	16
第 4 章 評価実験.....	17
4.1 まえがき.....	17
4.2 実験概要.....	17
4.2.1 データセット.....	17
4.2.2 評価指標.....	18
4.3 提案した CapsuleGAN による画像生成.....	20
4.4 実験 1：提案した CapsuleGAN による生成画像の品質評価.....	21
4.4.1 実験概要.....	21
4.4.2 実験結果および考察.....	21
4.5 実験 2：提案した CapsuleGAN を用いたデータ拡張の評価.....	23
4.5.1 実験概要.....	23
4.5.2 実験結果および考察.....	23

4.6 潜在変数を一様分布にしたときの VAE による画像生成	24
4.7 実験 3 : VAE による生成画像の多様性評価	25
4.7.1 実験概要	25
4.7.2 実験結果および考察	25
4.8 実験 4 : VAE を用いたデータ拡張の評価	26
4.8.1 実験概要	26
4.8.2 実験結果および考察	26
4.9 実験 5 : AAE と SMOTE を用いたデータ拡張の評価	28
4.9.1 実験概要	28
4.9.2 実験結果および考察	28
4.10 むすび	31
第 5 章 結論と今後の課題	32
5.1 結論	32
5.2 今後の課題	32
謝辞	33
参考文献	34
図一覧	36
表一覧	37
研究業績	38

第1章 序論

1.1 研究背景

近年、AI は技術が向上しているだけでなく、既に様々な商品・サービスに導入されてきている。具体的には、検索サイトやスマートスピーカー、自動車の自動運転などが身近な例として挙げられる。しかし、AI の精度を最大限高めるには一般的に大量のデータセットが必要である。アメリカや中国では、政府や企業の製品・サービスを通じて効率的にデータを収集しており、一部大手 IT 企業はビッグデータを独占しつつある。それゆえ、AI の技術水準は高い。対して、日本にはグローバルなプラットフォーム企業がなく、ビッグデータを確保できていないことから、AI の研究やビジネスにおいて遅れをとっていると考えられる。

国内企業の研究者や開発者がこうした状況の中、世の中にある膨大な量のデータを集めるのは難しい。そこで、既に所有しているデータおよび少ないデータを利活用しようとする研究の取り組みが主に二つある。一つは、少量のデータを効率的に学習する技法であり、もう一つは、生成モデルを使って学習用データを生成する技法である。

前者では、ファインチューニングや転移学習といった技法があげられる。ファインチューニングは、事前学習したモデルの重みパラメータを初期値として、再度新しいデータセットに対して学習するアルゴリズムである。一方、転移学習は事前学習したモデルの重みパラメータを固定して、新たに追加した層の重みパラメータのみ再度学習するアルゴリズムである。どちらのアルゴリズムも、一般的に事前学習の際に用いられるデータセットは大規模なものが多い。

後者の例として、Generative Adversarial Network (GAN)[1]や Auto Encoder (AE)[2]が挙げられる。生成モデルに関する研究は近年数多くなされており、GAN に関する論文や特許出願の件数は増加傾向にある[3]。GAN は新しい画像を生成できるが、収束させることが難しい。対して、AE は安定して学習できるが、データセットと同じような画像しか生成できず、新しい画像を生成することが難しい。また、身近な例として、これらの生成モデルは、ファッションや広告用の新しい画像の生成や 3D モデルの生成、顔画像の老化フィルタ生成などの様々なアプリケーションに使われ始めている[4]。

以上のように、生成モデルを用いることで様々な画像を生成することができる。様々な画像を生成することは、国内企業の研究者や開発者のデータセット不足を解決する支援になると考えられる。しかし、一部のクラスのデータセットは既に多く持っている場合や、同一クラスの似ているデータセットを多く持っている場合など、持っているデータセットの分布にはいくつかのパターンが考えられる。よって、データセットの分布のパターンを考慮して画像を生成する必要がある。

1.2 研究目的

本研究では、画像の品質およびデータ分布を考慮してデータ拡張することで、AI のクラス分類精度を向上することを目的とする。また、画像データセットの分布を 3 パターンに分け、それぞれに対する提案手法を述べる。一つ目に、全てのクラスの画像データセットが少ない場合。二つ目に、同じクラス内に似た画像データセットが多くある場合。三つ目に、画像データセットが多いクラスと、画像データセットが少ないクラスが混在する場合。手法としては、一つ目のパターンに対して、Generative Adversarial Capsule Network (CapsuleGAN)[5]を用いて品質を考慮した画像を生成する。二つ目のパターンに対しては、Variational Auto-Encoder (VAE)[6]の潜在変数を操作して、同一クラス内に偏りが生じないように画像を生成する。三つ目のパターンに対しては、Adversarial Autoencoder (AAE)[7]と Synthetic Minority Over-sampling Technique (SMOTE)[8]を用いて少数クラスのデータを拡張する。

1.3 本論文の構成

以下に本論文の構成を示す。

- 第 1 章 本章であり、研究の背景および目的について述べる。
- 第 2 章 本論文で用いる関連技術について述べる。
- 第 3 章 提案手法について述べる。
- 第 4 章 提案手法の実験概要と実験結果について記述し、実験結果の考察について述べる。
- 第 5 章 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

この章では、本研究の関連技術について詳細に述べる。まず、データ拡張について説明し、つぎに、生成モデルについて説明する。最後に、サンプリング技術について述べる。

2.2 データ拡張

一般的に、機械学習では、学習させるデータセットの量が少ないと過学習に陥る。この問題を解決するために、様々な手法でデータ量を水増しすることをデータ拡張という。画像処理分野では多くの場合、データセット画像に対して輝度変化や回転、平行移動、拡大縮小などの処理を施すことでデータ量を水増しする。画像処理分野だけでなく、自然言語処理分野においてもデータ拡張を行うことが一般的である。

2.3 CapsuleGAN

2.3.1 GAN

GAN[1]は、Goodfellow らが提案した、画像の潜在空間を学習するための手法であり、生成された画像を統計的に本物とほとんど見分けがつかないものにする事で、本物に似た新しい画像の生成を可能にする。GANはGeneratorとDiscriminatorの二つのネットワークで構成されるモデルであり、これらが互いに学習していくことで、最終的にGeneratorが本物に近い画像を生成するようになる。GANの概要を図2.1に示す。

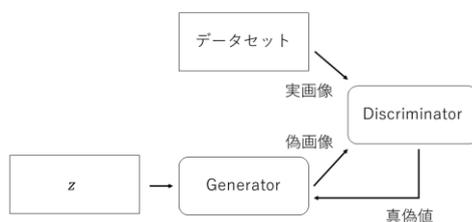


図 2.1 GAN の概要図

Generator は入力を潜在変数 z 、出力を画像とするニューラルネットワークである。対して、Discriminator は入力を画像、出力を真偽値とするニューラルネットワークである。Discriminator には、データセットの画像と Generator の生成画像を交互に入力していき、

Discriminator はそれらの画像に対して、データセット由来か生成器由来かを判断する。その判断が正しくなるように Discriminator は学習し、Generator は生成画像が Discriminator にデータセット由来だと判断させるように学習する。Generator と Discriminator の学習がうまく進むことで、Generator はデータセットの画像と似た画像を生成できるようになり、Discriminator は入力画像がデータセット由来かどうかをより高精度で判断できるようになる。GAN は式(2.1)で表される価値関数 $V(G, D)$ のミニマックス問題を解くことで、最適な Generator と Discriminator を得る。

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (2.1)$$

$D(x)$ は Discriminator の入力 x がデータセット由来である確率を表し、 $G(z)$ は Generator が z から生成するサンプルを表す。また、 $p_{data}(x)$ はデータセットの確率分布を表し、 $p_z(z)$ は z の確率分布を表す。

2.3.2 Capsule Network

Capsule Network[9]とは、Hinton らが提案した、Convolutional Neural Network (CNN)[10]に代わる画像認識手法である。従来、画像認識の分野において様々なモデルの CNN が用いられてきたが、CNN にはプーリング層において画像特徴の位置情報を喪失する問題があった。そこで、Hinton らは画像特徴の位置関係をカプセルという名のベクトルで保持することで、画像をいくつかのカプセルの重み付けで構成することに成功した。図 2.2 に示した簡易な顔をもとに、CNN と Capsule Network が認識できる画像の違いについて述べる。

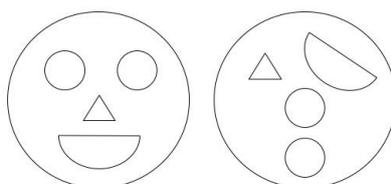


図 2.2 正しい特徴配置の画像(左)と不適切な特徴配置の画像(右)

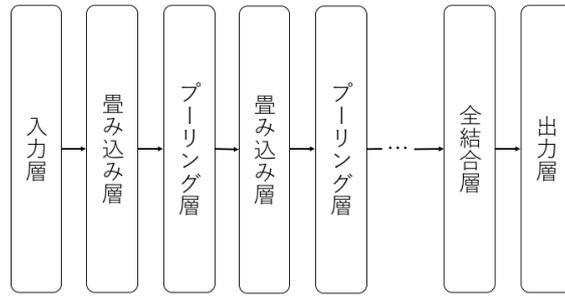


図 2.3 CNN の概要図

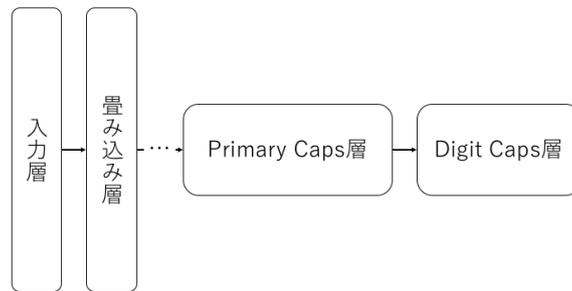


図 2.4 Capsule Network の概要図

CNN は、全結合層と畳み込み層、プーリング層から成り立つモデルである。畳み込み層では特徴を抽出し、プーリング層では特徴マップを圧縮する。しかし、プーリング層の働きにより、特徴間の位置関係の情報を喪失してしまう問題がある。たとえば、図の左の顔は目・鼻・口が正しい位置関係で構成されているが、右の顔のように目・鼻・口のそれぞれの位置関係が正しくなくても、CNN は右の図を顔と認識する。

そこで、Capsule Network ではプーリング層を廃止して、カプセルという新しい構造を導入した。Capsule Network は畳み込み層、Primary Caps 層、Digit Caps 層から成り立つモデルである。畳み込み層を通して得られた特徴マップが、Primary Caps 層によってカプセルに変換される。特徴マップがカプセルに変換される様子を図 2.5 に示す。

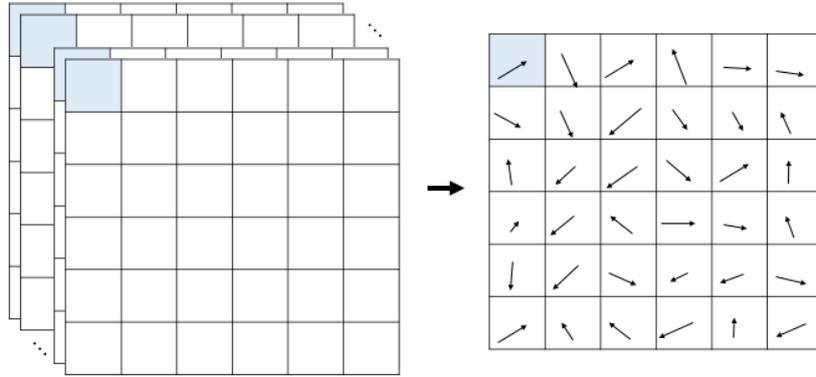


図 2.5 Primary Capsule 層

図 2.5 について説明する．まず，畳み込み層を通して得られた N チャンネルの特徴マップから隣接し合う特徴マップを M チャンネルずつグループ化する． n チャンネル目の特徴マップを \mathbf{a}_n とすれば， N チャンネルの特徴マップは次のように特徴マップの集合として表せる．

$$\mathbf{A} = \{\mathbf{a}_n | 1 \leq n \leq N\} \quad (2.3)$$

各チャンネルの同じ座標のピクセルのスカラ値がカプセルへ変換される．いま， i 行目 j 列目のピクセルに注目する． n チャンネル目の特徴マップ \mathbf{a}_n の i 行目 j 列目のピクセルを a_{nij} とすると， M 個のピクセルのスカラ値からなるカプセル \mathbf{c} は次のように M 次元ベクトルになる．

$$\mathbf{c} = \begin{pmatrix} a_{1ij} \\ a_{2ij} \\ \vdots \\ a_{Mij} \end{pmatrix} \quad (2.4)$$

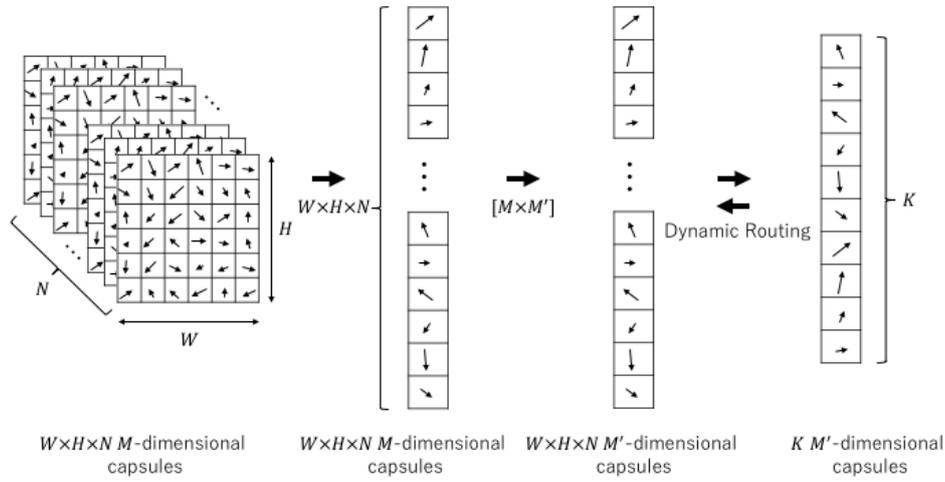


図 2.6 Digit Caps 層

次に、Digit Caps 層について図 2.6 をもとに説明する。Primary Capsule 層の出力のカプセルに $M \times M'$ 重み行列をかけることで、 M' 次元ベクトルを得る。クラス数を K とすると、Digit Caps 層は K 個の M' 次元ベクトルを出力する。クラス予測の際には、 K 個のベクトルそれぞれの L^2 ノルムを求め、最も大きい L^2 ノルムを持つクラスを予測クラス $pred$ とする。これは式 (2.5) で表される。

$$pred = arg \max_k L_k^2 \quad (2.5)$$

また、損失関数は式(2.6)で表される。

$$L_M = \sum_{k=1}^K T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k) \max(0, \|v_k\| - m^-)^2 \quad (2.6)$$

v_k は Digit Caps 層のカプセル k の出力である。なお、画像内にクラスラベル k が存在するとき $T_k = 1$ として、存在しないとき $T_k = 0$ とする。さらに $m^+ = 0.9, m^- = 0.1, \lambda = 0.5$ とする。

2.3.3 CapsuleGAN

CapsuleGAN[5]とは、Jaiswal らが提案したモデルであり、CNN から成る Generator と Capsule Network から成る Discriminator で構成された GAN である。CapsuleGAN における Discriminator の最終層は一つのカプセルのみを持ち、そのベクトルの大きさを、入力画像が真である予測確率としている。

一般的に、Capsule Network はパラメータ数が多い。なぜなら、Capsule Network において、カプセルの出力はスカラー値ではなくベクトルであり、Digit Caps 層において各カプセルが他の全てのカプセルと重みパラメータで結ばれているからである。Capsule Network のパラメータ数が多いことから、Capsule GAN には二つの問題がある。まず、Capsule Network はとても強力なモデルなので、学習過程において Generator を罰しやすく、それゆえ Generator が全く学習できなくなる、あるいはモード崩壊に陥ってしまう。さらに、Dynamic Routing の実行速度は遅い。

価値関数は、GAN における binary cross-entropy と異なり、式(2.6)の margin loss を用いる。モデル全体の価値関数は式(2.7)で表される。

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_t(x)} [-L_M(D(x), T = 1)] + \mathbb{E}_{z \sim p_z(z)} [-L_M(D(G(z)), T = 0)] \quad (2.7)$$

2.4 AE

2.4.1 AE

AE[2]とは、ニューラルネットワークを用いた次元削減アルゴリズムである。AEはEncoderとDecoderで構成され、Encoderを通して入力データを圧縮し、Decoderを通して復元する。Decoderによって復元されたデータと、入力データの二乗誤差を損失関数として、この誤差を小さくするようにAEが学習する。最終的にはDecoderが復元するデータは入力データと似たものになる。AEの概要を図2.7に示す。

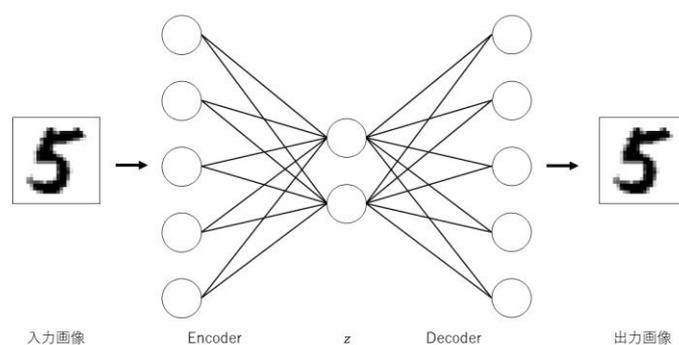


図 2.7 AE の概要図

入力画像と出力画像のピクセル数は同じであり、そのピクセル数を n とする。AEの入力と出力は画像のピクセルを1次元に展開したものである。入力画像を \mathbf{x} 、出力画像を $\hat{\mathbf{x}}$ とすると、損失関数 l_{recon} は次(2.8)で表される。

$$l_{recon} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \hat{\mathbf{x}}_i)^2 \quad (2.8)$$

2.4.2 VAE

VAE[6]とは、AEの潜在変数に確率分布を導入したアルゴリズムである。潜在変数 \mathbf{z} から学習データ \mathbf{x} を生成する確率分布を $P(\mathbf{z}|\mathbf{x})$ とする。Encoderが予測する確率分布を $Q(\mathbf{z}|\mathbf{x})$ とすると、Decoderが予測する確率分布は $P(\mathbf{x}|\mathbf{z})$ として解釈することができる。VAEでは、 $P(\mathbf{z}|\mathbf{x})$ と $Q(\mathbf{z}|\mathbf{x})$ の二つの確率分布のKL Divergenceを誤差関数としており、式(2.9)で表される。

$$\begin{aligned} D_{KL}[Q(\mathbf{z}|\mathbf{x}) \parallel P(\mathbf{z}|\mathbf{x})] &= \int Q(\mathbf{z}|\mathbf{x}) \log \frac{Q(\mathbf{z}|\mathbf{x})}{P(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\ &= \log P(\mathbf{x}) + D_{KL}(Q(\mathbf{z}|\mathbf{x}) \parallel P(\mathbf{z})) - \int Q(\mathbf{z}|\mathbf{x}) \log P(\mathbf{x}|\mathbf{z}) d\mathbf{z} \end{aligned} \quad (2.9)$$

式(2.9)を整理すると、

$$\log P(\mathbf{x}) - D_{KL}[Q(\mathbf{z}|\mathbf{x}) \parallel P(\mathbf{z}|\mathbf{x})] = \int Q(\mathbf{z}|\mathbf{x}) \log P(\mathbf{x}|\mathbf{z}) d\mathbf{z} - D_{KL}(Q(\mathbf{z}|\mathbf{x}) \parallel P(\mathbf{z})) \quad (2.10)$$

となる。ここで、 $D_{KL}[Q(\mathbf{z}|\mathbf{x}) \parallel P(\mathbf{z}|\mathbf{x})] > 0$ より、式(2.11)に書き直せる。

$$\log P(\mathbf{x}) \geq \int Q(\mathbf{z}|\mathbf{x}) \log P(\mathbf{x}|\mathbf{z}) d\mathbf{z} - D_{KL}(Q(\mathbf{z}|\mathbf{x}) \parallel P(\mathbf{z})) \quad (2.11)$$

式(2.11)の右辺は変分下限であるので、左辺の対数尤度を最大化するのではなく、右辺の変分下限を最大化することを考える。そのためには、右辺の第一項を最大化し、第二項を最小化すればよい。また、右辺の第一項はサンプリング近似で求めることができ、式(2.12)で書き表せる。

$$\log P(\mathbf{x}) \geq \mathbb{E}[\log P(\mathbf{x}|\mathbf{z})] - D_{KL}(Q(\mathbf{z}|\mathbf{x}) \parallel P(\mathbf{z})) \quad (2.12)$$

よって、VAE モデルの誤差関数は次の式(2.13)で表される。

$$l_{total} = -\mathbb{E}[\log P(\mathbf{x}|\mathbf{z})] + D_{KL}(Q(\mathbf{z}|\mathbf{x}) \parallel P(\mathbf{z})) \quad (2.13)$$

式(2.13)の右辺の第一項は、実装する上では、式(2.8)を用いることがある。また、式(2.13)の第1項において、Encoderで推定したパラメータから \mathbf{z} をサンプリングする必要があるが、 $\mathbf{z} \sim N(\mu, \sigma)$ としたとき、誤差逆伝播ができないことから、潜在変数を以下の式(2.14)でサンプリングする。これを Reparameterization Trick という [6]。

$$\begin{aligned} \varepsilon &\sim N(0, I) \\ \mathbf{z} &= \mu + \varepsilon\sigma \end{aligned} \quad (2.14)$$

最終的な VAE の概要を図 2.8 に示す。

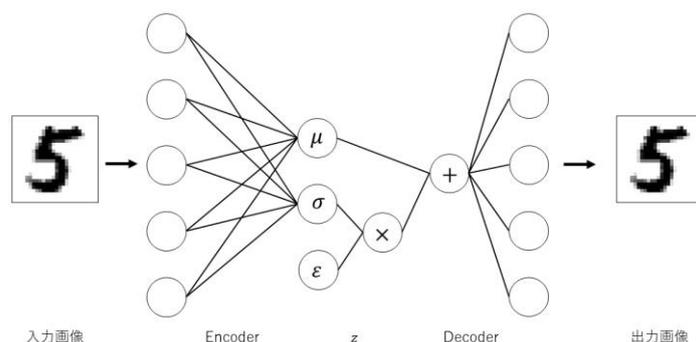


図 2.8 VAE の概要図

2.4.3 AAE

AAE[7]とは、AE と GAN を組み合わせたモデルであり、AE の潜在変数の分布が任意の分布になるように学習することができる。具体的には、GAN の Discriminator に任意の確率分布と AE の潜在変数の分布を入力して学習する。AAE は、AE において式(2.8)の損失関数が定義され、Adversarial Network においては式(2.7)の価値関数が定義される。AAE の概要を図 2.9 に示す

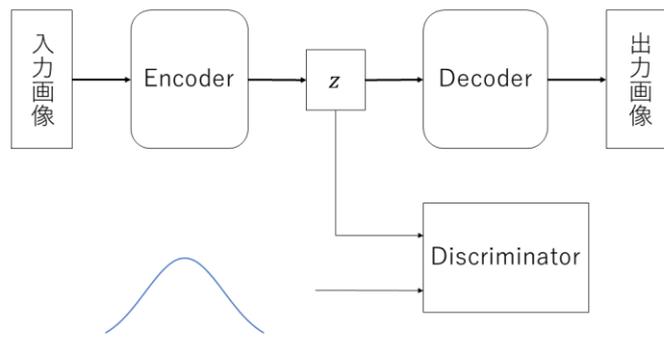


図 2.9 AAE の概要図

2.5 SMOTE

SMOTE[8]とは、オーバーサンプリング手法の一つである。この手法は不均衡データの対策として用いられる。SMOTE の流れは次のようになる。

1. サンプル数の少ないクラスの中から一つのサンプル s を選択する
2. サンプル s の K 近傍を取得する。
3. サンプル s と K 近傍の特徴量の差を求める。
4. 求めた特徴量の差に $0\sim 1$ の乱数をかける。
5. 処理 4 の結果をサンプル s に加える。これが新しく生成されたサンプルである。

SMOTE の一連の流れを 1 次元空間で表したものを図 2.10 に示す。

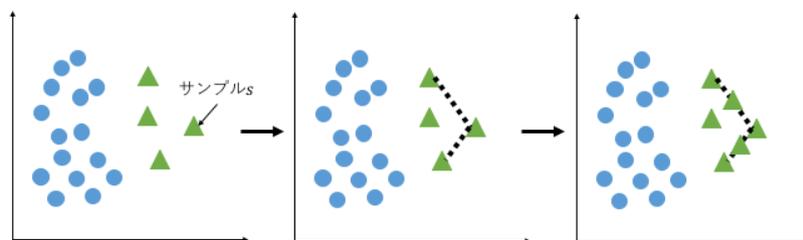


図 2.10 SMOTE の概要図

第3章 提案手法

3.1 まえがき

この章では，提案手法についての説明を行う．

3.2 CapsuleGAN を用いたデータ拡張

本節では，生成画像品質を考慮してデータ拡張を行う手法を提案する．渡部らは，DCGAN[11]を用いてデータ拡張を行った[12]が，DCGAN で生成された画像の中には，特徴の位置関係が不正な画像が見受けられた．これに対して，特徴の位置関係が正しい画像を生成することで，クラス分類精度を向上できると考える．そこで，特徴の位置関係が正しい画像を生成するために，CapsuleGAN を用いて画像を生成する．この手法のフローチャートを図 3.1 に示す．

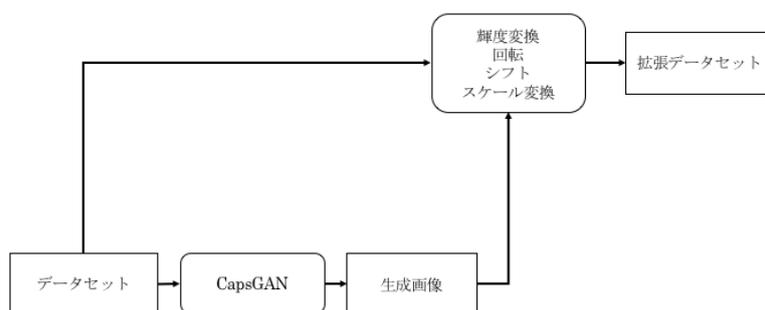


図 3.1 CapsuleGAN を用いたデータ拡張の概要図

図 3.1 について述べる．まず学習用のデータセットを用意する．CapsuleGAN を用いて，データセットから新しい類似画像を生成する．次に，データセットと生成画像の両方に対して輝度変換や回転，平行などの処理をすることで，拡張データセットを得る．

また，本研究では，従来の CapsuleGAN よりも品質の高い画像を生成するために，Discriminator の Primary Caps 層に変更を加えた．Primary Caps 層における特徴マップからカプセルへの変換について，従来手法と提案手法をそれぞれ図 3.2 に示した．

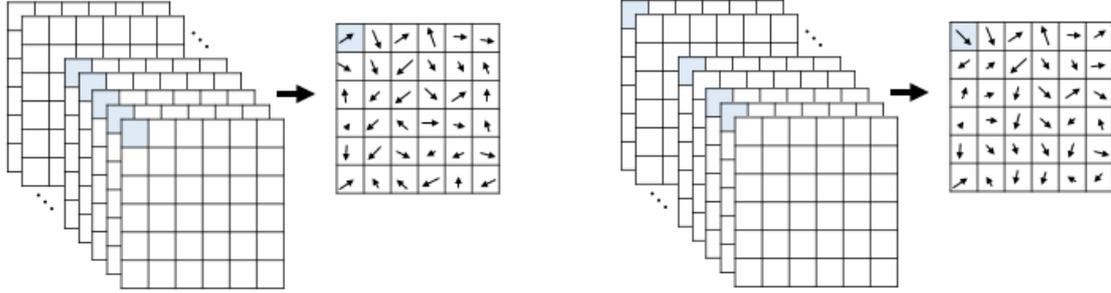


図 3.2 従来の特徴マップ選択方式(左)と提案手法の特徴マップ選択方式(右)

従来手法では、畳み込み層を通して得られた N チャンネルの特徴マップから隣接し合う特徴マップを M チャンネルずつグループ化してカプセルに変換していた。しかし、畳み込みフィルタの初期値のランダム性により、特徴間の位置関係を保持するためには、隣接していない特徴マップについてもカプセル化する必要があると考えられる。Primary Caps 層の入力を N チャンネルの特徴マップとすると、その特徴マップの集合は式(2.3)のように表せる。特徴マップの集合 \mathbf{A} をランダム置換 σ によって並び替えた特徴マップの集合を \mathbf{B} とすると、 σ と \mathbf{B} は式(3.1)、式(3.2)で表せる。

$$\mathbf{B} = \{\sigma(\mathbf{a}_n) | 1 \leq n \leq N\} \quad (3.1)$$

$$\sigma = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \cdots & \mathbf{a}_{N-2} & \mathbf{a}_{N-1} & \mathbf{a}_N \\ \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 & \cdots & \mathbf{b}_{N-2} & \mathbf{b}_{N-1} & \mathbf{b}_N \end{pmatrix} \quad (3.2)$$

また、カプセル \mathbf{c}_k は

$$\mathbf{c}_k = \begin{pmatrix} \mathbf{b}_{1ij} \\ \mathbf{b}_{2ij} \\ \vdots \\ \mathbf{b}_{Mij} \end{pmatrix}, \quad 1 \leq k \leq \frac{N \times W \times H}{M}, \quad 1 \leq i \leq H, \quad 1 \leq j \leq W \quad (3.3)$$

と表せる。ただし、特徴マップの行数を H 、列数を W とする。

従来のカプセル化にこの手法を加えることで、従来の CapsuleGAN で保持できなかった位置関係も保持できると考えられる。また、これらの処理によって得られた拡張データセットを用いて CNN モデルを学習させることで、クラス分類精度が向上すると考えられる。

3.3 VAE における潜在変数操作によるデータ拡張

本節では、同一クラスデータの分布の偏りを考慮してデータ拡張を行う手法を提案する。従来、データ拡張は全クラスもしくは少数クラスに適用されている。しかし、同一クラスにおいても、平均に近い画像は比較的多く、平均から遠い画像は比較的少ないと考えられる。そして、これは CNN のクラス分類において、平均から遠い画像を誤分類することにつながる。そこで、同一クラス内のデータ分布の偏りを解決する手法を提案する。提案手法の流れを次に示す。

1. 学習データセットのクラス数の VAE を用意する。
2. 学習データセットの各クラスに対して各 VAE を学習させる。
3. 学習済みの VAE を用いて、学習データを次元数 N の低次元空間にマッピングする。
4. N 次元空間における原点からユークリッド距離が最も大きい位置にあるデータまでの距離 d を求める。
5. N 次元空間の半径 d の球内に一様分布を生成する。
6. 生成された N 次元の潜在変数を Decoder に通して画像を生成する。

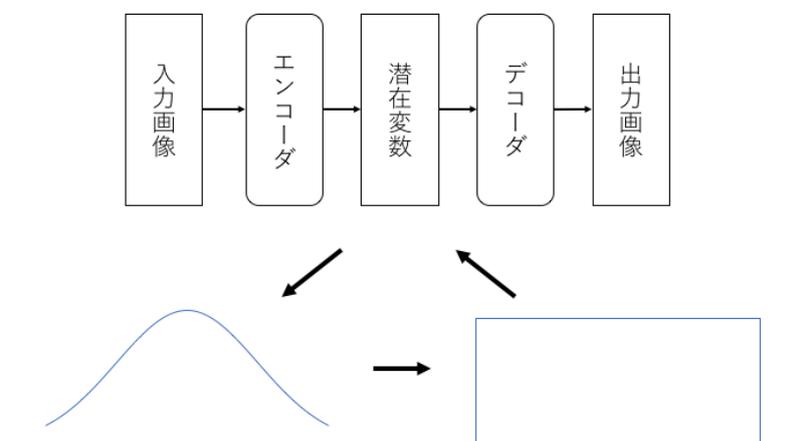


図 3.3 VAE における潜在変数操作の概要図

3.4 AAE と SMOTE を用いたデータ拡張

本節では、クラス分類問題における不均衡データに対して、AAE と SMOTE を用いてデータ拡張する手法を提案する。機械学習において、全クラスの均一な量の画像データセットを

集めるのは容易ではない。そこで、少数クラスのサンプルをオーバーサンプリングする手法を提案する。提案手法の流れを以下に記す。

1. AAE モデルを一つ用意する。
2. Discriminator の真のデータとして 10 個のラベル付き正規分布を用意する。
3. 学習データセット全クラスをまとめて AAE を学習させる。
4. 少数クラスの潜在変数に対して SMOTE を適用。
5. SMOTE によってオーバーサンプリングした潜在変数を学習済み AAE の Decoder に通して、画像を生成する。

手順 2 において用意する真のデータのイメージを図 3.4 に示す。各クラスの潜在変数を 2 次元空間に分布させる際、各クラスが交わらないようにする。各クラスの潜在変数はそれぞれ平均が異なる正規分布に従うように分布させる。また、この提案手法の概要を図 3.5 に示す。

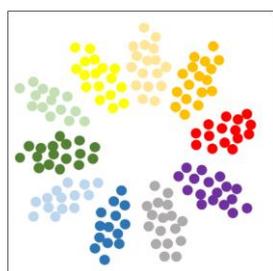


図 3.4 理想的な潜在変数の分布

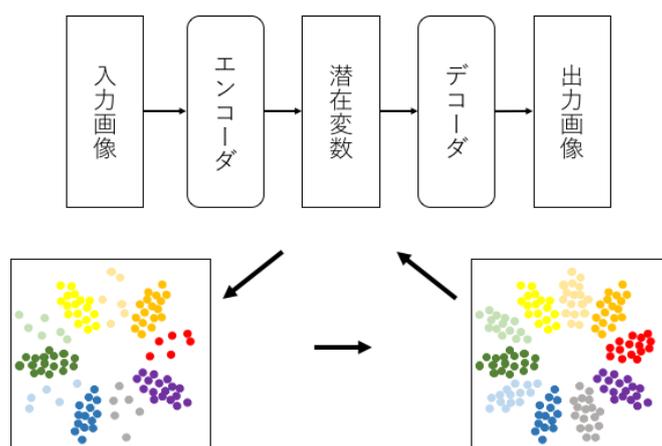


図 3.5 AAE と SMOTE を用いたデータ拡張の概要図

3.5 むすび

本章では、データ拡張として三つの手法について説明した。

第4章 評価実験

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と考察について述べる。なお、本章における実験は全て表4.1の環境下で行なった。

表 4.1 実験環境

OS	Ubuntu 18.04.3LTS
RAM	64GB
CPU	Intel® Core™ i7 – 6700K CPU @ 4.00GHz × 8
GPU	NVIDIA Titan X Pascal

4.2 実験概要

本研究では、提案手法の評価実験として次の評価実験を行う。まず、提案した CapsuleGAN を用いて画像を生成する。さらに、生成した画像の品質評価を行う。これを実験1とする。つぎに、提案した CapsuleGAN を用いたデータ拡張の評価を行う。これを実験2とする。つづいて、VAEにおいて一様分布にした潜在変数を、Decoderに通して画像を生成する。生成した画像の多様性を評価する。これを実験3とする。データセットの同一クラス内に分布の偏りがある場合、VAEを用いてデータ拡張をすることで、クラス分類精度を向上できるか調査する。これを実験4とする。最後に、データセットの他クラス間に分布の偏りがある場合、AAEとSMOTEを用いてデータ拡張をすることで、クラス分類精度を向上できるか調査する。これを実験5とする。

4.2.1 データセット

本研究では、公開データセットである MNIST[13], Fashion-MNIST[14]を使用する。MNISTとは、0~9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ており、1クラスあたりの平均枚数が7000枚のデータセットである。Fashion-MNISTとは、10クラスの衣料品の画像データと、そのラベルがセットとなっており、1クラスあたり平均7000枚の画像データセットである。どちらの画像データセットもグレースケール画像である。これらデータセットについてのまとめを表4.2に示す。また、データセットの例を図4.1に示す。

表 4.2 データセット

	データ数	分類クラス	1クラスあたりの平均データ数
MNIST	70000	10	7000
Fashion-MNIST	70000	10	7000

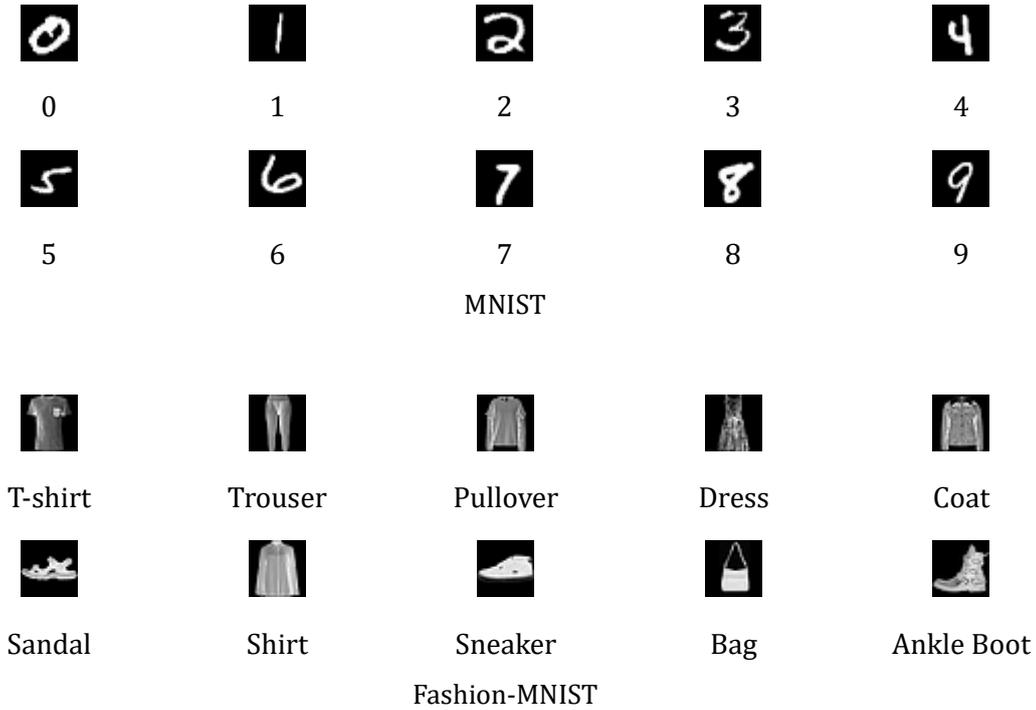


図 4.1 データセットの例

4.2.2 評価指標

本項では、本研究で用いる評価指標について述べる。提案手法によって生成された画像の品質を Inception Score(IS)[15], Fréchet Inception Distance(FID)[16], Multi-Scale Structural Similarity Image Quality(MS-SSIM)[17]の3つの評価指標を用いて評価する。これらの評価指標について説明する。

Inception Score とは、二つの確率分布の KL Divergence であり、式(4.1)で表される。

$$IS = \exp \mathbb{E}_x D_{KL}(p(y|x) \parallel p(y)) \quad (4.1)$$

$p(y|x)$ は inception モデルが画像 x をラベル y として予測する確率を表し、 $p(y)$ は y の周辺確率を表す。ISを用いることで、生成画像の品質と多様性を評価することができる。

FID とは、Inception モデルの中間層の出力によって得られるベクトル \mathbf{h} を使って求められる Fréchet Distance である。まず、 \mathbf{h} の分布が多変量正規分布に従うと仮定する。次に、 \mathbf{h} を用いて平均ベクトル $\boldsymbol{\mu}$ と不偏共分散行列 $\boldsymbol{\Sigma}$ を求める。画像集合を \mathbf{A} とするとき、平均ベクトル $\boldsymbol{\mu}$ と不偏共分散行列 $\boldsymbol{\Sigma}$ は次の式で表せる。

$$\begin{aligned}\boldsymbol{\mu} &= \frac{1}{|\mathbf{A}|} \sum_{\mathbf{h} \in \mathbf{H}} \mathbf{h} \\ \boldsymbol{\Sigma} &= \frac{1}{|\mathbf{A}| - 1} \sum_{\mathbf{h} \in \mathbf{H}} (\mathbf{h} - \boldsymbol{\mu})(\mathbf{h} - \boldsymbol{\mu})^T\end{aligned}\tag{4.2}$$

二つの画像集合 $\mathbf{A}_1, \mathbf{A}_2$ の距離を計算する場合、それぞれの平均ベクトルと不偏共分散行列を $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$ とおくと、Fréchet Distance は式(4.3)で表せる。

$$FID = |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|^2 + \text{tr}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 - 2(\boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_2)^{\frac{1}{2}})\tag{4.3}$$

MS-SSIM は複数の SSIM の平均であり、SSIM とは 2 枚の画像を画素値、コントラスト、構造の三つの観点で評価したときの総乗である。SSIM は画像内のウィンドウ毎に求められ、2 枚の画像 x, y の SSIM は以下の式(4.4)で表される。

$$SSIM(x, y) = \frac{(2\mu_x \mu_y + \varepsilon_1)(2\sigma_{xy} + \varepsilon_2)}{(\mu_x^2 + \mu_y^2 + \varepsilon_1)(\sigma_x^2 + \sigma_y^2 + \varepsilon_2)}\tag{4.4}$$

ここで、 μ はウィンドウ内の平均画素値、 σ はウィンドウ内の画素値の標準偏差、 ε は非常に小さな数の定数である。よって、ウィンドウ数を M としたときの MS-SSIM は次の式(4.5)で表される。

$$MS-SSIM(\mathbf{x}, \mathbf{y}) = \frac{1}{M} \sum_{m \in M} SSIM(x_m, y_m)\tag{4.5}$$

本研究では、生成画像の多様性評価として、生成画像の中からランダムにペアを選び、MS-SSIM の平均を扱う。これを $MSSSIM_{mean}$ とすると、最終的な式は式(4.6)となる。

$$MS-SSIM_{mean} = \frac{1}{N} \sum_{n \in N} MSSSIM(x_n, y_n)\tag{4.6}$$

また、CNN モデルでクラス分類を行う際の評価指標は以下の通りである。

Accuracy：全体の精度

Precision：ラベル L と予測した結果のうち，正解ラベルが L である割合

Recall：正解ラベルが L であるデータのうち，ラベル L と予測できた割合

4.3 提案した CapsuleGAN による画像生成

実験 1 では，提案した CapsuleGAN を用いて画像を生成する．3.2 節で提案した手法において，特徴マップの集合 A から特徴マップの集合 B へのランダム置換 σ を複数種類用意する． θ 種類用意とした場合，置換 σ の集合は $\sigma = \{\sigma_\theta | 1 \leq \theta \leq \theta\}$ となる．特徴マップの集合 A に対して各置換 σ_θ を用いて特徴マップの集合 B へ置換する．このときの置換の様子を図 4.2 に示す．

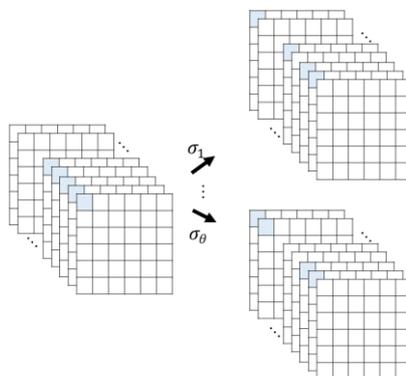


図 4.2 θ 種類のランダム置換による特徴マップの並び替え

本研究では，用いるパラメータ θ を $\theta = 1$ とする．

CapsuleGAN により生成された画像の例を図 4.4 に示す．提案した CapsuleGAN によって生成された画像は概ね，元のデータセットと似ている画像であると確認できる．しかし，MNIST における数字の 6 と 9 の画像が崩れている．Fashion-MNIST においては，バッグの取っ手の部分が消失している．これは Discriminator が Generator よりも早く学習していることが原因であると考えられる．

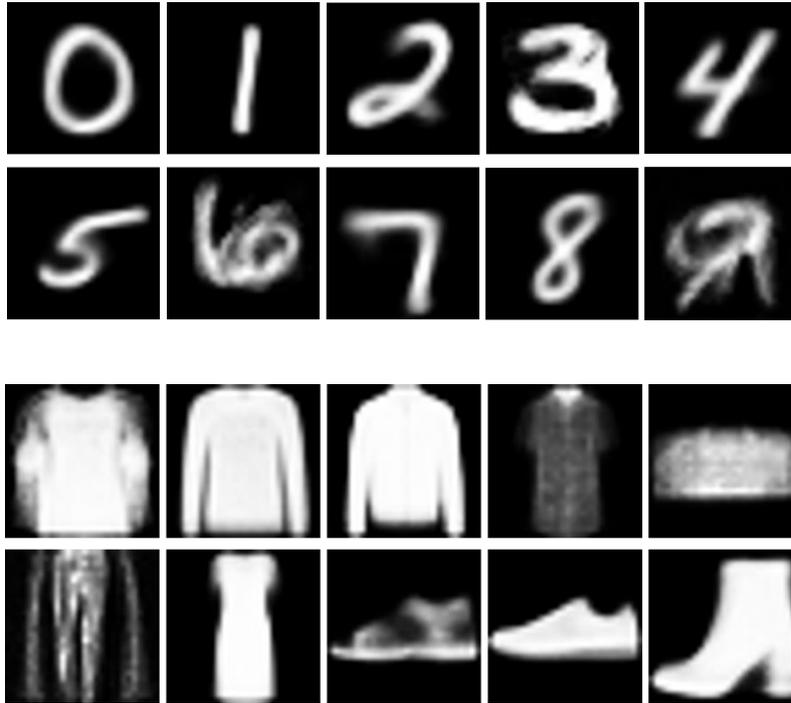


図 4.3 CapsuleGAN の生成画像例(上段：MNIST，下段：Fashion-MNIST)

4.4 実験 1：提案した CapsuleGAN による生成画像の品質評価

4.4.1 実験概要

実験 1 では、4.2.2 項で説明した評価指標を用いて、提案した CapsuleGAN によって生成した画像の品質を評価する。評価するにあたり、従来の CapsuleGAN でも同様に画像生成を行う。なお、各クラスのデータセット数を 100 枚として、生成画像の枚数を 100 枚とする。

4.4.2 実験結果および考察

MNIST の実験結果を表 4.3 に、Fashion-MNIST の実験結果を表 4.4 にそれぞれ示す。

表 4.3 実験 1 の結果(MNIST)

		クラス										平均
		0	1	2	3	4	5	6	7	8	9	
IS	従来手法	2.2	2.1	2.1	1.9	2.1	2.1	1.9	2.3	2.1	1.8	2.1
	提案手法	1.9	2.1	2.2	1.8	2.2	2.1	2.0	2.1	1.9	1.9	2.0
FID	従来手法	166	330	150	134	166	158	193	127	223	178	183
	提案手法	184	169	354	231	299	174	341	237	203	203	240

表 4.4 実験 1 の結果(Fashion-MNIST)

		クラス										平均
		0	1	2	3	4	5	6	7	8	9	
IS	従来手法	2.4	2.0	2.3	2.4	2.6	2.2	2.6	1.8	2.4	2.0	2.3
	提案手法	2.3	2.0	2.2	2.3	2.5	2.1	1.9	1.9	1.8	2.1	2.1
FID	従来手法	222	348	260	232	246	251	278	310	295	259	270
	提案手法	231	411	290	247	289	300	373	299	337	282	306

表 4.3, 表 4.4 より, 提案した CapsuleGAN による生成画像は, 従来の CapsuleGAN による生成画像に比べて, IS は小さい値であった. よって, CapsuleGAN による生成画像の方が画質は悪い. これは Discriminator が強すぎたことが原因であると考えられる. 一方, 提案手法による画像の方が, FID の値が大きいことから, 提案手法による生成画像の方が, 元の画像データセットと似ていない画像を生成できたといえる.

4.5 実験 2：提案した CapsuleGAN を用いたデータ拡張の評価

4.5.1 実験概要

実験 2 では、提案した CapsuleGAN を用いたデータ拡張の評価実験として、CNN の学習データセットに加える、CapsuleGAN による生成画像の枚数と、CNN のクラス分類精度の関係を調査した。1 クラスあたりに加える生成画像の枚数を 0, 50, 100, 500, 1000 枚と変えて CNN の学習を行い、それぞれの CNN のクラス分類精度の比較を行った。ただし、元のデータセットは各クラス 100 枚とした。

4.5.2 実験結果および考察

データ拡張せずにクラス分類した結果を表 4.5 に示す。従来の CapsuleGAN を用いてデータ拡張したときのクラス分類結果を、データセットごとに表 4.6, 表 4.7 にそれぞれ示す。提案した CapsuleGAN を用いてデータ拡張したときのクラス分類結果を、データセットごとに表 4.8, 表 4.9 にそれぞれ示す。

表 4.5 データ拡張しない場合のクラス分類

データセット	クラス										平均
	0	1	2	3	4	5	6	7	8	9	
MNIST	0.97	0.98	0.87	0.86	0.89	0.85	0.92	0.89	0.83	0.84	0.89
Fashion-MNIST	0.73	0.89	0.73	0.89	0.61	0.85	0.23	0.93	0.92	0.88	0.77

表 4.6 従来の CapsuleGAN によるデータ拡張後のクラス分類(MNIST)

追加枚数	クラス										平均
	0	1	2	3	4	5	6	7	8	9	
50	94	94	88	86	90	91	94	83	88	85	89
100	93	94	90	87	92	93	91	91	83	84	90
500	92	92	84	88	94	88	94	90	92	85	90
1000	95	95	85	92	92	84	93	88	92	88	91

表 4.7 従来の CapsuleGAN によるデータ拡張後のクラス分類(Fashion-MNIST)

追加枚数	クラス										平均
	0	1	2	3	4	5	6	7	8	9	
50	82	89	77	83	40	95	24	82	92	85	75
100	75	91	57	86	63	89	49	93	93	91	79
500	76	91	75	86	60	90	43	90	91	91	79
1000	81	92	78	83	61	89	40	92	93	92	80

表 4.8 提案した CapsuleGAN によるデータ拡張後のクラス分類(MNIST)

追加枚数	クラス										平均
	0	1	2	3	4	5	6	7	8	9	
50	97	97	87	86	90	87	92	84	88	90	90
100	98	96	88	82	86	88	92	86	91	89	90
500	97	97	87	90	89	88	91	87	90	88	90
1000	98	98	85	92	90	90	92	87	83	90	91

表 4.9 提案した CapsuleGAN によるデータ拡張後のクラス分類(Fashion-MNIST)

追加枚数	クラス										平均
	0	1	2	3	4	5	6	7	8	9	
50	72	86	76	88	52	91	23	90	93	90	76
100	79	91	71	84	73	90	16	90	92	91	78
500	77	91	64	83	80	89	35	89	94	93	80
1000	74	92	68	78	73	89	35	88	92	93	78

表 4.5, 表 4.6 より, 追加枚数を増やすことで, クラス分類精度が向上することが確認できる. しかし, ある程度増やしたところで, 精度の向上が見られなくなることが分かる. これは正しい画像だけでなくノイズが生成されていることが原因であると考えられる.

4.6 潜在変数を一様分布にしたときの VAE による画像生成

本節では, 潜在変数を一様分布にした VAE によって生成された画像の例を図 4.5 に示す. この実験において, VAE に各クラス 1000 枚の画像データセットを学習させた.



図 4.4 VAE の生成画像例(上段：MNIST, 下段：Fashion-MNIST)

VAE によって生成された画像は概ね、元のデータセットと似ている画像であると確認できる。しかし、MNIST における数字の 0 と 4 の画像が崩れている。Fashion-MNIST においては、ヒールの画像とバッグの画像が一部消失している。これらは同一クラス内において、ある画像の潜在変数と別の画像の潜在変数の間に存在する潜在変数を Decoder に通して得られた画像であると考えられる。

4.7 実験 3：VAE による生成画像の多様性評価

4.7.1 実験概要

実験 3 では、4.2.2 項で説明した評価指標の $MS-SSIM_{mean}$ を用いて、VAE によって生成した画像の多様性を評価する。

4.7.2 実験結果および考察

VAE によって各クラスの画像を 317 枚生成し、生成した画像の中からランダムに 1000 組の画像を選択し、画像間の MS-SSIM を求めた。さらに、MS-SSIM の平均 $MS-SSIM_{mean}$ を求め、求めた結果をデータセットごとに表 4.10、表 4.11 に示す。また、VAE を学習させる際に用いた元データセット 1000 枚の $MS-SSIM_{mean}$ を求め、生成画像の $MS-SSIM_{mean}$ と比較する。

表 4.10 実験 3 の結果(MNIST)

	クラス										
	0	1	2	3	4	5	6	7	8	9	平均
データセット	0.44	0.70	0.35	0.37	0.36	0.30	0.39	0.44	0.42	0.39	0.42
生成画像	0.53	0.61	0.70	0.51	0.50	0.40	0.49	0.47	0.55	0.44	0.52

表 4.11 実験 3 の結果(Fashion-MNIST)

	クラス										
	0	1	2	3	4	5	6	7	8	9	平均
データセット	0.52	0.66	0.33	0.63	0.46	0.23	0.30	0.61	0.24	0.71	0.47
生成画像	0.50	0.53	0.40	0.50	0.57	0.37	0.46	0.57	0.32	0.81	0.50

実験の結果、提案手法において、MNIST の全クラス平均の $MS-SSIM_{mean}$ は 0.52 となり、Fashion-MNIST の全クラス平均の $MS-SSIM_{mean}$ は 0.50 となった。元のデータセットの $MS-SSIM_{mean}$ に比べ、高い値となることを確認した。よって、生成画像は元のデータセットよりも多様性がないことが分かる。また、実データに存在しない潜在変数を Decoder に入力した場合、似たような画像を生成すると考えられる。

4.8 実験 4 : VAE を用いたデータ拡張の評価

4.8.1 実験概要

実験 4 では、VAE における潜在変数の分布を操作することで、データセットの各クラス内のデータ分布の偏りがクラス分類に与える影響を調査する。そこで、元の学習データセットのみを学習した CNN モデルと、VAE で生成した画像のみを学習した CNN モデルの性能を、CNN クラス分類において比較する。この実験では、各クラスのデータセット数を 50, 500, 1000 枚と変えて実験を行い、全ての場合において、学習用画像とバリデーション用画像を 4:1 に分けて学習を行った。

4.8.2 実験結果および考察

MNIST, Fashion-MNIST の実験結果をそれぞれ表 4.12, 表 4.13 に示す。各値は全クラスの平均である。

表 4.12 実験 4 の結果(MNIST)

		各クラスのデータセット数		
		50	500	1000
Accuracy [%]	データセット	65	93	95
	生成画像	74	69	71
Precision [%]	データセット	77	94	95
	生成画像	77	74	77
Recall [%]	データセット	67	94	95
	生成画像	74	69	71

表 4.13 実験 4 の結果(Fashion-MNIST)

		各クラスのデータセット数		
		50	500	1000
Accuracy [%]	データセット	60	78	84
	生成画像	66	62	60
Precision [%]	データセット	69	78	84
	生成画像	66	65	60
Recall [%]	データセット	67	78	84
	生成画像	66	62	60

表 4.12, 表 4.13 より, 元の画像データセットの枚数が少ないとき, VAE による生成画像のみを学習する方が, クラス分類精度が高いことが確認できる. しかし, 枚数が多い時はク

ラス分類精度が低い。枚数が多いときはノイズデータが多く生成されていることが原因であると考えられる。よって、元の画像データセットの枚数が少ないとき、同一クラス内のデータ分布を考慮したデータ拡張が有効である。

4.9 実験 5：AAE と SMOTE を用いたデータ拡張の評価

4.9.1 実験概要

実験 5 では、AAE と SMOTE を用いることで、クラス間の不均衡なデータ分布におけるクラス分類精度を向上できるか調査した。まず、少数派クラスの各データ数を 10, 50, 100 枚、多数派クラスの各データ数を 100, 500, 1000 枚としたとき、データ拡張をしない場合の CNN によるクラス分類精度の調査をした。つづいて、比較実験として、DCGAN を用いて少数派クラスのデータを 100, 500, 1000 枚まで拡張したときのクラス分類精度の調査をした。最後に、AAE と SMOTE を用いて少数派クラスのデータを 100, 500, 1000 枚まで拡張したときのクラス分類精度の調査をした。

4.9.2 実験結果および考察

データ拡張しない場合の結果を表 4.14, 表 4.15, 表 4.16 に示す。また、DCGAN を用いてデータ拡張した場合の結果を表 4.17, 表 4.18, 表 4.19 に示す。さらに、AAE と SMOTE を用いてデータ拡張した場合の結果を表 4.20, 表 4.21, 表 4.22 に示す。

表 4.14 データ拡張なしのクラス分類(少数クラス 10 枚, 多数クラス 100 枚)

		クラス										平均
		0	1	2	3	4	5	6	7	8	9	
		10 枚	10 枚	10 枚	10 枚	10 枚	100 枚	100 枚	100 枚	100 枚	100 枚	
Precision [%]	MNIST	0	0	0	0	0	38	63	59	25	47	23
	Fashion- MNIST	0	77	0	83	0	85	22	76	91	94	53
Recall [%]	MNIST	0	0	0	0	0	80	89	87	88	69	41
	Fashion- MNIST	0	82	0	52	0	85	35	83	90	87	57

表 4.15 データ拡張なしのクラス分類(少数クラス 50 枚, 多数クラス 500 枚)

		クラス										
		0	1	2	3	4	5	6	7	8	9	平均
		50枚	50枚	50枚	50枚	50枚	500枚	500枚	500枚	500枚	500枚	
Precision [%]	MNIST	98	98	95	93	97	84	85	86	70	74	88
	Fashion-MNIST	94	95	100	87	71	95	25	87	89	95	84
Recall [%]	MNIST	85	92	71	77	63	93	96	95	95	93	86
	Fashion-MNIST	28	88	2	64	16	93	93	96	95	90	66

表 4.16 データ拡張なしのクラス分類(少数クラス 100 枚, 多数クラス 1000 枚)

		クラス										
		0	1	2	3	4	5	6	7	8	9	平均
		100枚	100枚	100枚	100枚	100枚	1000枚	1000枚	1000枚	1000枚	1000枚	
Precision [%]	MNIST	97	98	96	96	97	85	90	88	82	82	91
	Fashion-MNIST	81	79	77	71	81	85	84	88	86	89	82
Recall [%]	MNIST	94	93	83	76	80	97	96	97	95	95	91
	Fashion-MNIST	71	69	71	72	76	89	83	88	85	83	79

表 4.17 DCGAN によるデータ拡張後のクラス分類(各クラス 100 枚)

		クラス										
		0	1	2	3	4	5	6	7	8	9	平均
Precision [%]	MNIST	93	92	89	93	93	84	82	81	73	69	85
	Fashion-MNIST	81	94	44	67	50	91	30	85	89	90	72
Recall [%]	MNIST	94	93	70	74	59	90	92	88	89	85	84
	Fashion-MNIST	38	86	48	68	30	88	57	88	92	91	69

表 4.18 DCGAN によるデータ拡張後のクラス分類(各クラス 500 枚)

		クラス										
		0	1	2	3	4	5	6	7	8	9	平均
Precision [%]	MNIST	97	97	96	94	95	89	92	89	79	75	90
	Fashion-MNIST	84	97	74	85	66	97	38	86	92	95	81
Recall [%]	MNIST	94	95	81	82	70	94	94	94	94	94	89
	Fashion-MNIST	50	92	52	77	63	91	72	96	95	90	78

表 4.19 DCGAN によるデータ拡張後のクラス分類(各クラス 1000 枚)

		クラス										
		0	1	2	3	4	5	6	7	8	9	平均
Precision [%]	MNIST	97	97	96	94	97	91	93	92	85	87	93
	Fashion-MNIST	83	99	80	83	69	98	43	90	92	96	83
Recall [%]	MNIST	96	95	88	85	87	96	96	95	95	95	93
	Fashion-MNIST	59	90	49	82	68	94	75	96	96	93	80

表 4.20 AAE と SMOTE によるデータ拡張後のクラス分類(各クラス 100 枚)

		クラス										
		0	1	2	3	4	5	6	7	8	9	平均
Precision [%]	MNIST	93	91	91	93	93	81	82	83	76	75	85
	Fashion-MNIST	83	94	56	69	60	86	43	87	90	88	76
Recall [%]	MNIST	91	90	78	80	65	80	90	91	93	87	85
	Fashion-MNIST	41	88	60	70	41	85	63	85	90	91	71

表 4.21 AAE と SMOTE によるデータ拡張後のクラス分類(各クラス 500 枚)

		クラス										
		0	1	2	3	4	5	6	7	8	9	平均
Precision [%]	MNIST	98	97	92	94	90	89	89	93	86	85	91
	Fashion- MNIST	81	84	76	86	70	75	60	87	86	87	78
Recall [%]	MNIST	96	95	91	94	76	95	91	86	90	91	91
	Fashion- MNIST	84	80	64	77	77	80	69	79	95	93	80

表 4.22 AAE と SMOTE によるデータ拡張後のクラス分類(各クラス 1000 枚)

		クラス										
		0	1	2	3	4	5	6	7	8	9	平均
Precision [%]	MNIST	98	98	95	94	93	90	91	90	89	89	93
	Fashion- MNIST	85	82	78	74	94	93	76	88	80	79	83
Recall [%]	MNIST	98	96	97	96	90	93	95	90	93	93	94
	Fashion- MNIST	88	85	80	76	78	90	89	84	88	76	83

少数クラスに対して AAE と SMOTE を用いることで、クラス分類の精度が向上することが確認できる。さらに、クラス間のデータ分布を考慮せずに DCGAN でデータ拡張を行う場合よりも、データ分布を考慮したデータ拡張する場合の方が、精度が高いと分かる。これによって、SMOTE によるオーバーサンプリングしたデータはクラス分類する上で必要なデータであると分かる。

4.10 むすび

本章では、提案手法の有効性を確認するための評価実験の結果を示した。実験の結果から本提案手法の有効性と問題点を確認した。

第5章 結論と今後の課題

5.1 結論

本論文では、画像認識精度を向上するために、画像の品質およびデータ分布を考慮してデータ拡張する手法を提案した。

生成画像品質を考慮したデータ拡張では、CapsuleGAN を用いた手法を提案した。従来手法と同等の品質であるが、データセットとは似ていない画像を生成できた。

同一クラスデータの分布の偏りを考慮したデータ拡張では、VAE における潜在変数操作による手法を提案した。提案手法による VAE の生成画像は多様性がないことが分かったが、一部の場合において、クラス分類精度を向上させる生成画像を生成できた。

クラス分類問題における不均衡データに対して、AAE と SMOTE を用いたデータ拡張手法を提案した。少数クラスに対して、提案手法によるデータ拡張をすることでクラス分類精度を向上できることが分かった。

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、パラメータ数の削減である。CapsuleGAN におけるランダム選択は、選択の数だけパラメータが線形関数的に増加するため、過学習に陥りやすく計算時間がかかる。また、カラー画像を扱う場合、チャンネル数が増えることからパラメータ数も多くなる。これらを解決するために、パラメータの削減手法などが求められる。

二つ目は、カラー画像や高画質画像への対応である。本研究ではグレースケール画像を扱い、潜在変数を 2 次元としていたが、扱う画像の次元数が増えたときに、最適な潜在変数の次元数の決定をする必要がある。

謝辞

本研究に際して、丁寧かつ熱心なご指導をしてくださり、実験環境および快適な研究環境を与えてくださった渡辺裕教授に深謝いたします。

夏合宿と少数チームゼミに参加していただき、的確で貴重な意見をくださった笠井裕之教授に感謝いたします。

ゼミの場でコメントやアドバイス、技術共有をしてくださった研究室の皆様感謝いたします。特に、日頃より飲み会やゲーム、映画鑑賞などプライベートの面でサポートしてくださった同僚の赤塚紘輝氏、稲田健太郎氏、梅田聖也氏に感謝いたします。

最後に、これまでさまざまな面で多大なサポートをしていただいた家族に深謝いたします。

参考文献

- [1] Ian Goodfellow, Jean Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative Adversarial Networks," Neural Information Processing Systems (NIPS), 2014
- [2] Kramer, Mark A., "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks," AIChE Journal. 37 pp.233-243, Feb 1991
- [3] 酒井美里, "画像や音声を生成する AI, 「GAN」の特許が急増," <<https://tech.nikkeibp.co.jp/atcl/nxt/column/18/00408/00012/>> (参照 2019 年 12 月 5 日)
- [4] Jason Brownlee, "18 Impressive Applications of Generative Adversarial Networks(GANs)," <<https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/>> (参照 2019 年 12 月 7 日)
- [5] Ayush Jaiswal, Wael AbdAlmageed, Yue Wu, Premkumar Natarajan, "CapsuleGAN: Generative Adversarial Capsule Network," The European Conference on Computer Vision (ECCV) Workshops, pp.0-0, 2018
- [6] Diederik P. Kingma, Max Welling, "Auto-Encoding Variational Bayes," International Conference on Learning Representation, 2014
- [7] Alireza Makhzami, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, Brendan Frey, "Adversarial autoencoders," arXiv preprint arXiv:1511.05644, 2015
- [8] N. C. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research 16, pp.321-357, 2002
- [9] Sara Sabour, Nicholas Frosst, Geoffrey E Hinton, "Dynamic Routing Between Capsules," Neural Information Processing Systems (NIPS), 2017
- [10] A. Krizhevsky, I. Sutskever, G.F.Hinton, "ImageNet Classification with Deep Convolutional," Neural Information Processing Systems (NIPS), 2012
- [11] A.Radford, L.Metz, S.Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," In 4th Int. Conf. on Learning Representations (ICLR), No 67, pp.1-16 (May. 2016)
- [12] 渡部, 渡辺, "生成画像品質を考慮した DCGAN を用いたデータオーギュメンテーション," FIT2017(第 16 回情報科学技術フォーラム), H-038, pp.273-274, (Sept. 2017)
- [13] Yann Lecun, Corinna Cortes, Christopher JC Burges, "MNIST hand written digit database," <<https://yann.lecun.com/exdb/mnist/>> (2018 年 4 月 1 日参照)
- [14] Han Xiao, Kashif Rasul, Roland Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," arXiv preprint arXiv: 1708.07747, 2017

- [15] Tim Salimans, "Improved Techniques for Training GANs," Neural Information Processing Systems (NIPS), 2016
- [16] Martin Heusel, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," Neural Information Processing Systems (NIPS), 2017
- [17] Z. Wang, E. P. Simoncelli, A. C. Bovik, "Multiscale structural similarity for image quality assessment," Proceedings of the 37th IEEE Asilomar Conference on Signals, Systems and Computers, Publisher: IEEE Computer Society (2003), pp.1398-1402

図一覧

図 2.1	GAN の概要図.....	3
図 2.2	正しい特徴配置の画像(左)と不適切な特徴配置の画像(右).....	4
図 2.3	CNN の概要図.....	5
図 2.4	Capsule Network の概要図.....	5
図 2.5	Primary Capsule 層.....	6
図 2.6	Digit Caps 層.....	7
図 2.7	AE の概要図.....	8
図 2.8	VAE の概要図.....	10
図 2.9	AAE の概要図.....	11
図 2.10	SMOTE の概要図.....	11
図 3.1	CapsuleGAN を用いたデータ拡張の概要図.....	12
図 3.2	従来の特徴マップ選択方式(左)と提案手法の特徴マップ選択方式(右).....	13
図 3.3	VAE における潜在変数操作の概要図.....	14
図 3.4	理想的な潜在変数の分布.....	15
図 3.5	AAE と SMOTE を用いたデータ拡張の概要図.....	15
図 4.1	データセットの例.....	18
図 4.2	●種類のランダム置換による特徴マップの並び替え.....	20
図 4.3	CapsuleGAN の生成画像例(上段：MNIST, 下段：Fashion-MNIST).....	21
図 4.4	VAE の生成画像例(上段：MNIST, 下段：Fashion-MNIST).....	25

表一覧

表 4.1	実験環境	17
表 4.2	データセット	18
表 4.3	実験 1 の結果(MNIST)	22
表 4.4	実験 1 の結果(Fashion-MNIST)	22
表 4.5	データ拡張しない場合のクラス分類.....	23
表 4.6	従来の CapsuleGAN によるデータ拡張後のクラス分類(MNIST).....	23
表 4.7	従来の CapsuleGAN によるデータ拡張後のクラス分類(Fashion-MNIST)	24
表 4.8	提案した CapsuleGAN によるデータ拡張後のクラス分類(MNIST)	24
表 4.9	提案した CapsuleGAN によるデータ拡張後のクラス分類(Fashion-MNIST).....	24
表 4.10	実験 3 の結果(MNIST)	26
表 4.11	実験 3 の結果(Fashion-MNIST)	26
表 4.12	実験 4 の結果(MNIST)	27
表 4.13	実験 4 の結果(Fashion-MNIST)	27
表 4.14	データ拡張なしのクラス分類(少数クラス 10 枚, 多数クラス 100 枚)	28
表 4.15	データ拡張なしのクラス分類(少数クラス 50 枚, 多数クラス 500 枚)	29
表 4.16	データ拡張なしのクラス分類(少数クラス 100 枚, 多数クラス 1000 枚)	29
表 4.17	DCGAN によるデータ拡張後のクラス分類(各クラス 100 枚).....	29
表 4.18	DCGAN によるデータ拡張後のクラス分類(各クラス 500 枚).....	30
表 4.19	DCGAN によるデータ拡張後のクラス分類(各クラス 1000 枚).....	30
表 4.20	AAE と SMOTE によるデータ拡張後のクラス分類(各クラス 100 枚).....	30
表 4.21	AAE と SMOTE によるデータ拡張後のクラス分類(各クラス 500 枚).....	31
表 4.22	AAE と SMOTE によるデータ拡張後のクラス分類(各クラス 1000 枚).....	31

研究業績

- [1] 松田, 丸壽, 渡辺: “生成画像品質を考慮した CapsGAN によるデータ拡張 (Data Augmentation Using CapsGAN Considering Generated Image Quality)”, 2018 年映像情報メディア学会冬季大会, 12D-4, Dec. 2018
- [2] 青木, 松田, 原, 渡辺: “ステレオ全天球カメラからの距離計測の基礎検討 (Basic Study on Distance Measurement from Stereo Omni-Image Cameras)”, 映像情報メディア学会年次大会, 33B-5, Sep. 2017
- [3] 松田卓也: “魚眼カメラと通常カメラを用いた距離測定 (Estimating Distance by Using Fish-Eye Lens Camera and Conventional Camera)”, 卒業論文 Feb. 2017