4G4-4

深層学習における敵対的ネットワークによるラベル推定と半教師あり学習

Label Estimation and Semi-Supervised Learning Using Adversarial Networks in Deep learning

立花亮介 *1 松原崇 *2 上原邦昭 *2
Ryosuke Tachibana Takashi Matsubara Kuniaki Uehara

*1神戸大学 工学部 情報知能工学科
Department of Computer Science and Systems Engineering, Kobe University

*2神戸大学 大学院 システム情報学研究科
Graduate School of System Informatics, Kobe University

Semi-supervised learning is a topic of practical importance because of the difficulty of obtaining numerous labeled data. In this paper, we apply an extension of adversarial autoencoder to semi-supervised learning tasks. In attempt to separate style and content, we divide the latent representation of the autoencoder into two parts, then we regularize the autoencoder by imposing a prior distribution on two parts to make them independent. As a result, one of the latent representations is associated with content, which is useful to classify the images. We demonstrate that our method disentangles style and content of the input images and achieves less test error rate than vanilla autoencoder on MNIST semi-supervised classification tasks.

1. はじめに

深層学習を用いた手法が画像識別の分野において高い精度を上げており，その有用性が注目を浴びている [Lecun 15]．従来の機械学習手法において困難であった特徴の自動的な獲得が可能となり，人手で特徴を作成するという作業を省略できると考えられている．近年画像識別で高い精度をあげているのは ILSVRC2012 で優勝した手法 [Krizhevsky 12] など，ラベルありデータのみを学習に用いる教師あり学習によるものである．しかし実用的な識別問題の多くにはラベルのないデータをモデルの学習に使用する必要があるため，少量のラベルありデータと大量のラベルなしデータを用いて学習を行う半教師あり学習での識別精度の向上が求められる．

深層学習における半教師あり学習において，高い識別精度を誇る手法として Deep Generative Models (DG) [Kingma 14]，Ladder Networks [Rasmus 15]，Virtual Adversarial Training (VAT) [Miyato 16] などが知られている．VAT はモデル分布と入力に摂動を加えた分布の KL divergence が大きくなる方向に入力を摂動させて学習を行う手法である．DG は変分ベイズを使用して潜在変数を推定する手法である．この手法では，全ラベルに関して誤差を求めるため，ラベルの数に応じて計算量が大きくなる．また，Ladder Networks は自己符号化器の符号化器の各層から復号化器の対応する各層へ情報を伝播することによって各層でデータ多様体を学習するはしご状の深層ニューラルネットワークである．このネットワークでは，細部の情報を低層で保存し，識別に必要な情報のみを高層で保存する．そして高層の情報のみを識別に使用することで識別精度の向上を実現している．しかし，Ladder Networks は複雑な構造を持ち，パラメータの調整を綿密に行う必要がある [Pezeshki 16]．これらの手法の問題点を改善した，構造が単純かつ汎用性を持つ深層学習における半教師あり学習手法が求められる．

本稿では，教師なし学習を行う敵対的自己符号化器 [Makhzani 15] を半教師あり学習に適用したモデルを提案する．これは，自己符号化器の潜在表現を二つに分割し，それぞれに対して任意の事前分布を適合させることにより自己符号化器に正則化を施したものである．この正則化により潜在変数に独立性を持たせることができるため，識別に必要となる情報をそれ以外の情報から切り離すことが可能となる．そして識別に必要となる情報のみを識別に用いることによって半教師あり学習において識別精度の向上を実現させることができる．

連絡先: 立花亮介，神戸大学工学部情報知能工学科，
tachibana@ai.cs.kobe-u.ac.jp

2. 生成的敵対的ネットワーク

データの生成を行う生成器と，データが生成データであるか与えられたデータセットのデータであるか識別を行う識別器を，同じネットワークで敵対的に訓練させるフレームワークを生成的敵対的ネットワーク (Generative Adversarial Networks; GAN) [Goodfellow 14] という．敵対的ネットワークでは，生成器はデータ分布を学習することにより，データセットのデータと同様の特徴を持つデータの生成が可能となる．一方，識別器は入力されたデータがデータセットからのデータである確率を出力する．生成器は識別器の出力する確率を最大化するように学習が進められる．このように生成器と識別器による敵対的な学習過程を経ることにより，二つのモデルに期待する出力をさせることが可能となる．ここで，敵対的ネットワークにおいて期待される出力とは，識別器では全ての入力において出力が 0.5 となることであり，生成器では訓練データの分布と同様の分布に従う標本データを出力することである．データ $\boldsymbol{x}$ についての生成器の分布 p_g の学習において，入力ノイズ $p_{\boldsymbol{z}}(\boldsymbol{z})$ を定義することにより，生成器を $G(\boldsymbol{z};\theta_g)$ と表す．ここで，θ_g は多層ニューラルネットワークにおけるパラメータとする．また，同様に θ_d をパラメータとして，識別器は $D(\boldsymbol{x};\theta_d)$ と定義される．$D(\boldsymbol{x})$ は，入力 $\boldsymbol{x}$ が p_g の標本であるよりも真のデータ分布 p_{data} の標本である確率をスカラー値で出力する．評価

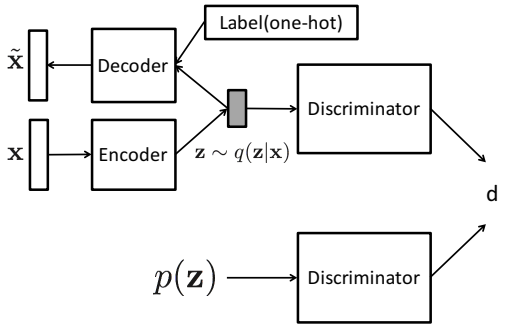

図 1: 敵対的自己符号化器の構造

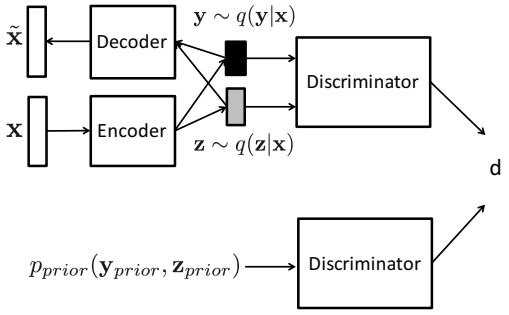

図 2: 提案手法の構造

関数は以下の式 (1) として定式化される．

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

学習の初期段階において G の学習が不十分である時，G が生成するデータは訓練データとは大きく異なるため，D は容易にその２つを見分けることが可能である．この場合，$\log(1-D(G(\mathbf{z})))$ の値は飽和する．G は $\log(1 - D(G(\mathbf{z})))$ を最小化するように学習が進むため，$D(G(\mathbf{z}))$ を最大化することになる．

3. 提案手法

3.1 敵対的自己符号化器

自己符号化器によって獲得した潜在表現を，敵対的ネットワークを用いて任意の事前分布に近づけることで自己符号化器に正則化を施したものを敵対的自己符号化器 (Adversarial Autoencoders; AAE) [Makhzani 15] という．図 1 に敵対的自己符号化器のネットワーク構造を示す．入力を $\boldsymbol{x}$，符号化器が獲得する潜在変数を $\boldsymbol{z}$，潜在変数を適用する事前分布を $p(\boldsymbol{z})$，符号化器の分布を $q(\boldsymbol{z}|\boldsymbol{x})$，復号化器の分布を $p(\boldsymbol{x}|\boldsymbol{z})$ とする．また，データ分布を $p_d(\boldsymbol{x})$ とすると，

$$q(\boldsymbol{z}) = \int_{\boldsymbol{x}} q(\boldsymbol{z}|\boldsymbol{x}) p_d(\boldsymbol{x}) d\boldsymbol{x} \quad (2)$$

となる．敵対的自己符号化器では敵対的ネットワークを用いて式 (2) の $q(\boldsymbol{z})$ を $p(\boldsymbol{z})$ に適合させる．ここで，敵対的ネットワークにおいて生成器に該当するものは符号化器であり，この符号化器は識別器を $q(\boldsymbol{z})$ が $p(\boldsymbol{z})$ からのサンプルであると欺くように学習する．識別器は，入力が事前分布のサンプルである確率 d を出力する．そして，敵対的ネットワークと自己符号化器は再構成誤差と正則化誤差の二つの誤差によりまとめて確率的勾配降下法で学習される．復号化の際，図 1 に示すように潜在変数 $\boldsymbol{z}$ と共にラベルを入力することにより，潜在変数からラベルを分離することが出来る．復号化器にラベルを入力する際は，対応する要素のみが 1 で他が 0 となる one-hot coding されたベクトルを用いる．復号化器は，ラベルと潜在変数 $\boldsymbol{z}$ の両方を使用することにより画像の再構成を行う．このネットワーク構造により，潜在変数はデータからラベル以外の情報のみを分離して学習する．

3.2 敵対的自己符号化器の半教師あり学習への応用

符号化器が獲得する潜在変数として，$\boldsymbol{z}$ に加え $\boldsymbol{y}$ を導入する．符号化器では分布 $q(\boldsymbol{z}|\boldsymbol{x})$ に加えて分布 $q(\boldsymbol{y}|\boldsymbol{x})$ を学習することになる．さらに，潜在変数 $\boldsymbol{z}$ と $\boldsymbol{y}$ を，共に一つの識別器に入力する．$q(\boldsymbol{z})$ は任意の事前分布 $p_{prior}(\boldsymbol{z}_{prior})$ に適合するように，$q(\boldsymbol{y})$ は任意の事前分布 $p_{prior}(\boldsymbol{y}_{prior})$ に適合するように敵対的ネットワークにより学習が行われる．通常の自己符号化器では，符号化器によって潜在変数 $\boldsymbol{z}$ のみが獲得される．潜在変数が $\boldsymbol{z}$ のみであると，再構成に必要な識別に直接関係のないデータまでも保存されてしまうと考えられる．そこで，識別に必要なラベル情報（コンテンツ）とそれ以外の情報（スタイル）の分離行うために単純に潜在変数を二つに分け，コンテンツを保存させる潜在変数をラベルの推定に使用し，スタイルを保存させる潜在変数を再構成時に使用する．これにより，潜在変数を $\boldsymbol{y}$，$\boldsymbol{z}$ に分離した後，各々を独立した分布に適合させることによって潜在変数間に独立性を持たせることが可能となると考えられる．通常の自己符号化器で単に潜在変数を二つに分離した場合，コンテンツとスタイルに十分に分離されないことがわかっている [Tachibana 15]．潜在変数の独立性が保たれることによって，コンテンツとスタイルが十分に分離されることが期待できる．この手法は，識別問題へとモデルを適用する際にデータをコンテンツとスタイルに分離し，コンテンツのみをラベルの推定に使用することが出来るため，識別精度の向上に有効であると考えられる．

本手法では，３つの要素から構成される以下の式を誤差関数とする．

$$L = \lambda_{NLL} L_{NLL} + \lambda_{rec} L_{rec} + \lambda_{reg} L_{reg} \quad (3)$$

ここで，L_{NLL} は符号化器による推定ラベルと正解ラベルの識別誤差，L_{rec} は元のデータと復号化器によって再構成されたデータの再構成誤差，L_{reg} は識別器における正則化誤差を表す．λ_{NLL}，λ_{rec}，λ_{reg} は実数の係数であり，それぞれの要素の影響を調整する役割を果たす．

識別誤差 L_{NLL} には，負の対数尤度

$$L_{NLL} = -\sum_i \sum_j I(y_i = j) \log P(y_i|x_i) \quad (4)$$

を用いる．ここで，$I(\mathrm{cond})$ はある条件 cond が成立すれば 1 を，成立しないならば 0 をとる指示関数，x_i は i 番目の入力，y_i は x_i のラベル，$P(y_i|x_i)$ は x_i が観測されたときラベルが y_i である確率を表すベクトルである．

再構成誤差 L_{rec} には入力データ $\boldsymbol{x}$ と再構成データ $\tilde{\boldsymbol{x}}$ の二乗誤差

$$L_{rec} = ||\boldsymbol{x} - \tilde{\boldsymbol{x}}||^2 \quad (5)$$

を用いる．

正則化誤差 L_{reg} は，生成器である符号化器と識別器による敵対的ネットワークから定義される誤差である．データ分

布を $p_d(\boldsymbol{x})$ とする．符号化器において，$\boldsymbol{x} \sim p_d(\boldsymbol{x})$ に対する潜在変数 $\boldsymbol{y}$ の推定 $q(\boldsymbol{y}|\boldsymbol{x})$ を行い $\boldsymbol{y}$ を出力するネットワークを生成器 G_y，潜在変数 $\boldsymbol{z}$ の推定 $q(\boldsymbol{z}|\boldsymbol{x})$ を行い $\boldsymbol{z}$ を出力するネットワークを生成器 G_z とする．独立で任意の事前分布 $p_{prior}(\boldsymbol{y}_{prior}, \boldsymbol{z}_{prior})$ からのサンプルを $\hat{\boldsymbol{y}}$，$\hat{\boldsymbol{z}}$ とする．また，識別器を D とする．D には，$\boldsymbol{y} = G_y(\boldsymbol{x})$ と $\boldsymbol{z} = G_z(\boldsymbol{x})$，もしくは $\hat{\boldsymbol{y}} \sim p_{prior}(\boldsymbol{y}_{prior}, \boldsymbol{z}_{prior})$ と $\hat{\boldsymbol{z}} \sim p_{prior}(\boldsymbol{y}_{prior}, \boldsymbol{z}_{prior})$ が入力として与えられる．D は入力が $\hat{\boldsymbol{y}} \sim p_{prior}(\boldsymbol{y}_{prior}, \boldsymbol{z}_{prior})$ と $\hat{\boldsymbol{z}} \sim p_{prior}(\boldsymbol{y}_{prior}, \boldsymbol{z}_{prior})$ である確率をスカラー値 d で出力する．このネットワークにおける学習では，G_y は $\boldsymbol{y}$，G_z は $\boldsymbol{z}$ が事前分布からのサンプルであると D に誤認識させるように学習を行う．つまり，符号化器は D の出力を 1 にするように学習が進む．一方，D は入力が $\boldsymbol{y} \sim G_y$，$\boldsymbol{z} \sim G_z$ のとき出力が 0 に，入力が $\hat{\boldsymbol{y}} \sim p_{prior}(\boldsymbol{y}_{prior}, \boldsymbol{z}_{prior})$,$\hat{\boldsymbol{z}} \sim p_{prior}(\boldsymbol{y}_{prior}, \boldsymbol{z}_{prior})$ のとき出力が 1 になるように学習を行う．この学習によって，符号化器の分布 $q(\boldsymbol{y}|\boldsymbol{x})$ が事前分布 $p_{prior}(\boldsymbol{y}_{prior})$ に，分布 $q(\boldsymbol{z}|\boldsymbol{x})$ が事前分布 $p_{prior}(\boldsymbol{z}_{prior})$ に適合される．

以上の学習は以下の誤差関数によって定式化される．

$$
\begin{aligned}
L_{reg} &= \min_{G_y, G_z} \max_{D} V(D, G_y, G_z) \\
&= \mathbb{E}_{\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}} \sim P_{prior}(\boldsymbol{y}_{prior}, \boldsymbol{z}_{prior})}[\log D(\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})] \\
&\quad + \mathbb{E}_{\boldsymbol{x} \sim P_d(\boldsymbol{x})}[\log (1 - D(G_y(\boldsymbol{x}), G_z(\boldsymbol{x})))]
\end{aligned} \tag{6}
$$

教師あり学習を行う場合は，誤差関数 (3) において λ_{NLL} を 0 以外に設定する．これにより，L_{NLL} を学習に使用することになるため，ラベルを用いた学習が行える．一方で，λ_{NLL} を 0 に設定すると，完全な教師なし学習を行うことになる．ラベルありデータの場合に λ_{NLL} を使用して教師あり学習を行い，ラベルなしデータの場合に λ_{NLL} を 0 に設定にして教師なし学習を行うことで，半教師あり学習を実現することが出来る．λ_{reg} を 0 に設定した場合は，通常の自己符号化器と同様になる．また，λ_{rec} と λ_{reg} を共に 0 設定した場合は単なる多層ニューラルネットワークによる学習と同様になる．

4. 評価実験と考察

4.1 データセット

手書き数字のデータセットである MNIST を用いて実験を行った．MNIST は 1 枚が 28×28 の画像サイズを持ち60,000 枚の訓練データと 10,000 枚のテストデータから構成される．60,000 枚の訓練データを二つに分割し，50,000 枚を訓練データとし，10,000 枚を検証データとした．訓練データはモデルの学習に，検証データはハイパーパラメータの決定に使用した．半教師あり学習の実験を行う際は，MNIST の 50,000 枚の訓練データをラベルありデータとラベルなしデータの二種類に分割した．ラベルありデータの数は 100 枚，600 枚，1,000 枚，3,000 枚と変化させて実験を行った．この分割を行う際ラベルありデータの数だけランダムにデータを選択するが，この時選択するラベルありデータのラベルが偏りを持たず一様になるようにした．

4.2 モデル構成

自己符号化器の入力層の次元は $28 \times 28 = 784$，隠れ層の次元は $(1000, 1000)$，潜在変数の次元は $\boldsymbol{z}$ が 50，$\boldsymbol{y}$ が 10 とした．符号化器，復号化器それぞれに対して活性化関数として ReLU 関数を使用しており，符号化器には Dropout [Srivastava 14] とバッチ正則化 [Ioffe 15] を適用した．識別器には，$\boldsymbol{z}$ と $\boldsymbol{y}$ を入力として用い，それぞれの次元は自己符号化器の潜在変数と同様の 50，10 となる．$\boldsymbol{z}$ に対する隠れ層の次元は 1,000，$\boldsymbol{y}$ に対する隠れ層の次元は 1,000 とし，これらの二つの隠れ層はさらに次元 1,000 の隠れ層で統合した．識別器の出力は一次元のスカラー値となる．識別器に対しても活性化関数として ReLU 関数を使用した．また，式 (3) の係数は，ラベルありデータに対しては $\lambda_{NLL} = 1.0$，$\lambda_{rec} = 7.0$，$\lambda_{reg} = 1.0$，ラベルなしデータに対しては $\lambda_{NLL} = 0.0$，$\lambda_{rec} = 7.0$，$\lambda_{reg} = 1.0$ とした．$q(\boldsymbol{z}|\boldsymbol{x})$ に適合する事前分布は標準多変量正規分布 $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ とし，$q(\boldsymbol{y}|\boldsymbol{x})$ に適合する事前分布は一様分布とした．

表 1: 半教師あり学習における識別結果

			Test error(%)
N_l	AE	AAE(z)	AAE(y,z)
100	16.11(±1.92)	17.37(±3.62)	7.07(±2.02)
600	9.46(±0.23)	8.28(±1.78)	5.77(±0.58)
1000	6.93(±0.28)	7.25(±1.15)	5.14(±0.54)
3000	4.88(±0.24)	5.17(±0.61)	4.22(±0.75)

4.3 評価方法

提案手法の有効性を検証するために二つの実験を行なった．まず，敵対的ネットワークが半教師あり学習での画像の識別精度向上に有効であるか検証を行った．これは，通常の自己符号化器，潜在変数 $\boldsymbol{z}$ のみ正則化を行った敵対的自己符号化器，提案手法である潜在変数 $\boldsymbol{y}$ と $\boldsymbol{z}$ に正則化を行った敵対的自己符号化器の三つの手法について，それぞれ半教師あり学習を行い精度を比較を行うことで検証した．

次に，潜在変数が独立となっているかについての検証と，潜在変数が特徴をスタイルとコンテンツに分離して獲得しているかについての検証を行った．独立性の検証は，潜在変数の相関行列のヒートマップを生成し，対角成分以外が 0 に近い値を取っていることを確認することで行う．潜在変数が特徴をスタイルとコンテンツに分離して獲得しているかについて検証は，コンテンツを獲得していると考えられる潜在変数 $\boldsymbol{y}$ を固定して，スタイルを獲得していると考えられる潜在変数 $\boldsymbol{z}$ を変化させることで，コンテンツを保ちながらスタイルが変化することを確認することで行う．

4.4 実験結果

提案手法である敵対的ネットワークを用いた自己符号化器と，通常の自己符号化器のそれぞれを用いた半教師あり学習における MNIST の識別結果を表 1 に示す．表 1 におけるそれぞれの数値は，テストデータにおけるエラー率 (%) である．N_l はラベルありデータの数を表す．AE，AAE(z)，AAE(y,z) はそれぞれ，通常の自己符号化器，潜在変数 $\boldsymbol{z}$ のみを正則化した敵対的自己符号化器，潜在変数 $\boldsymbol{y}$ と $\boldsymbol{z}$ を正則化した敵対的自己符号化器を表している．ラベルありデータの枚数によらず，通常の自己符号化器や $\boldsymbol{z}$ のみを正則化した敵対的ネットワークと比較して，提案手法はエラー率減少を達成している．また，ラベルありデータ数が少ないほど他の手法と比較して提案手法のエラー率減少が大きい傾向がある．

提案手法と通常の自己符号化器の潜在変数の相関行列のヒートマップを以下の図 3 に示す．図 3 は潜在変数 $\boldsymbol{z}$ の 50 次元の各要素と潜在変数 $\boldsymbol{y}$ の 10 次元の各要素の計 60 個の変数による相関行列である．図の右にある数値は相関係数の値であり，−1 から 1 の値を取る．ある二つの確率変数が独立であるならば，相関係数は 0 となる．対角成分は同一の確率変数による相関係数であるので，常に 1 となる．Fig. 3 において，提案手

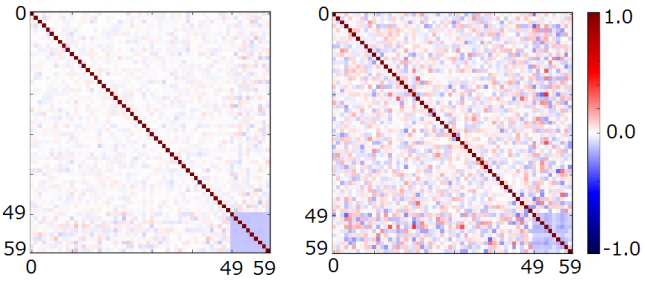

図 3: (左図) 提案手法の相関行列 (右図) 自己符号化器の相関行列

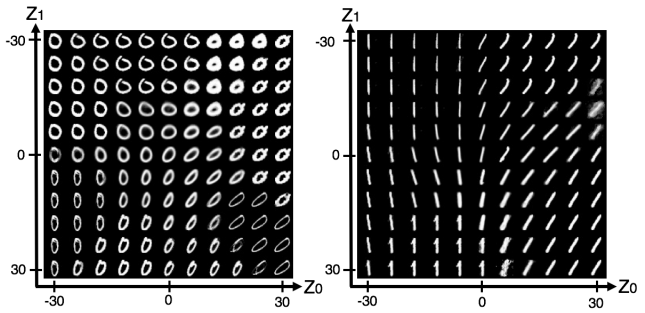

図 4: (左図) y を 0 に固定し z を潜在空間内で動かした時の出力 (右図) y を 1 に固定し z を潜在空間内で動かした時の出力

法の相関行列 (左図) は，自己符号化器の相関行列 (右図) よりも 0 に近くなっている成分が多い．この結果から，通常の自己符号化器に比べて提案手法では潜在変数が独立となるように学習されたことがわかる．自己符号化器の相関行列では右側と下側で色が濃くなっていることから，$\boldsymbol{y}$ と $\boldsymbol{z}$ が独立でなく，スタイルとコンテンツに分離することができていないと考えられる．どちらの相関行列も右下に負の相関がみられるが，これは識別に softmax 関数を使っており，あるラベルに対応する出力が，他のラベルに対応する出力に対して排他的であるからと考えられる．

潜在変数 $\boldsymbol{y}$ を 0 に固定して潜在変数 $\boldsymbol{z}$ の値を変化させて生成した画像と，潜在変数 $\boldsymbol{y}$ を 1 に固定して潜在変数 $\boldsymbol{z}$ を変化させて生成した画像を以下の図 4 に示す．この検証では，二次元ベクトル $\boldsymbol{z}$ の第一要素と第二要素をそれぞれ -30 から 30 まで，6 の間隔を開けて変化させている．Fig. 4 では，コンテンツである数字が固定されたまま，スタイルである文字の書体が変化していることがわかる．つまり，潜在変数 $\boldsymbol{y}$ がコンテンツを，潜在変数 $\boldsymbol{z}$ がスタイルをそれぞれ獲得していると考えられる．さらに，スタイルを変化させたとき左図と右図で同じ値の $\boldsymbol{z}$ のときに同様な特徴変化がみられる．例えば右下の $\boldsymbol{z}$ では，どちらも斜めを向いているという特徴を持っている．このことからも，$\boldsymbol{z}$ がコンテンツと独立したスタイルを獲得していると考えられる．

5. 結論

本論文では，教師なし学習のモデルである敵対的自己符号化器を，半教師あり学習へと応用させた手法を提案した．この手法は，潜在変数を二つに分離しそれぞれを敵対的ネットワークによって正則化することにより独立性を持たせるというものである．独立性を持たせることにより，データからスタイルとコンテンツを分離することが可能となり，コンテンツのみをラベル推定に使用することにより，細部の情報に引きずられない推定を行うことが出来る．実際に，MNIST を使用して半教師あり学習を行い，正則化を施さない自己符号化器と比較してエラー率減少を達成した．また，各潜在変数が独立であることを，相関行列によって検証した．さらに，潜在変数によってデータからコンテンツとスタイルに分離できていることを，画像の生成により検証した．今後の課題として，他の正則化手法との比較が挙げられる．

参考文献

[Goodfellow 14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y.: Generative Adversarial Nets, *Advances in Neural Information Processing Systems 27*, pp. 2672–2680 (2014)

[Ioffe 15] Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *Journal of Machine Learning Research (JMLR): W&P*, Vol. 37, (2015)

[Kingma 14] Kingma, D. P., Rezende, D. J., Mohamed, S., and Welling, M.: Semi-supervised Learning with Deep Generative Models, *Advances in Neural Information Processing Systems* (2014)

[Krizhevsky 12] Krizhevsky, A., Sutskever, I., and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, in *Advances in Neural Information Processing Systems 25*, pp. 1097–1105 (2012)

[Lecun 15] Lecun, Y., Bengio, Y., and Hinton, G.: Deep learning, *Nature*, Vol. 521, pp. 436–444 (2015)

[Makhzani 15] Makhzani, A., Shlens, J., Jaitly, N., and Goodfellow, I.: Adversarial Autoencoders, *arXiv* (2015)

[Miyato 16] Miyato, T., Koyama, M., Nakae, K., and Ishii, S.: Distributional Smoothing by Virtual Adversarial Examples Shin-ichi Maeda, *International Conference on Learning Representation* (2016)

[Pezeshki 16] Pezeshki, M., Fan, L., Brakel, P., Courville, A., and Bengio, Y.: Deconstructing the Ladder Network Architecture, in *International Conference on Learning Representation* (2016)

[Rasmus 15] Rasmus, A., Valpola, H., and Berglund, M.: Semi-Supervised Learning with Ladder Network, in *Advances in Neural Information Processing Systems* (2015)

[Srivastava 14] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Dropout : A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research (JMLR)*, Vol. 15, pp. 1929–1958 (2014)

[Tachibana 15] Tachibana, R., Matsubara, T., and Uehara, K.: 深層学習における教師なし特徴抽出手法の比較, 情報処理学会 関西支部大会 (2015)