敵対的生成ネットワークを利用した学習データの生成

Generation of Learning Data Utilizing Generative Adversarial Networks

早川博章*, 伊藤亮, 中島直樹***, 相原威*,*****

Hirofumi Hayakawa*, Ryo Ito**, Naoki Nakajima*** and Takeshi Aihara *,***

*玉川大学工学部情報通信工学科, 194-8610 東京都町田市玉川学園6-1-1

**玉川大学工学部機械情報システム学科, 194-8610 東京都町田市玉川学園6-1-1

***玉川大学大学院工学研究科, 194-8610 東京都町田市玉川学園6-1-1

*Department of Information & Communication Technology, College of Engineering, Tamagawa University, 6-1-1 Tamagawagakuen Machida-shi Tokyo 194-8610

** Department of Intelligent Mechanical Systems, College of Engineering, Tamagawa University, 6-1-1 Tamagawagakuen Machida-shi Tokyo 194-8610

***Graduate School of Engineering, Tamagawa University, 6-1-1 Tamagawagakuen Machida-shi Tokyo 194-8610

Abstract

The emergence of deep learning, the accuracy of object recognition has been dramatically improved. Currently, upon performing object recognition by deep learning, not only computational resources such as a computer but also a lot of learning data are required. In particular, it has been found that even if this learning data is learned by the same procedure using the same neural network, it significantly affects the learning result (correct answer rate of recognition) due to a bias in the number of data. In addition, although learning data that can be used for deep learning is being disclosed on the Internet, learning data available for a task to be recognized is not always disclosed. In that case, it is necessary to create the data used for learning manually. But if the collected learning data is small, the method to inflate the learning data by parallel shift, rotation, inversion, deterioration processing of the image, etc. has been conventionally proposed. However, all of these methods are mathematically determined processes, and if they are used frequently, learning may be hindered. In this research, we focused on generative adversarial networks and examined whether learning data to improve recognition accuracy could be generated with this.

Keywords: GAN, DCGAN, Generative model, Deep learning, Neural network

1. はじめに

深層学習の登場により，物体認識の精度は飛躍的に向上してきている．現状，深層学習による物体認識を行う場合には，コンピューターなどの計算資源だけでなく多くの学習データが必要である．特にこの学習データはデータ数の偏りなどに

より，同じニューラルネットワークを使用し，同じ手順で学習したとしても，学習結果(認識の正答率)などに多大な影響を及ぼすことが分かっている[1]．また深層学習に使用可能な学習データはインターネット上での公開が進んでいるが，認識したい課題に使用出る学習データが公開されているとは限らない．その場合，学習に使用するデータを自らの手で作成する必要があるが，集めた学習データが少ない場合，従来では，画像の平行移動，回転，反転，劣化処理などにより学習データを水増しする手法[2]が提案されてきた．しかし，それらの手法はどれも数学的に決まった処理であり多用すると学習を阻害する可能性がある．

本研究では敵対的生成ネットワーク[3]に着目し，認識精度を向上させる学習データが生成できないか検討を行った．

2．敵対的生成ネットワーク

敵対的生成ネットワーク（GAN：Generative Adversarial Networks）は生成器（generator）と判別器（discriminator）により構成されており，生成器はノイズを入力として偽物のデータを生成する．一方で判別器は生成器が出力した偽物のデータと敵対的生成ネットワークの学習データである正解のデータの２つの入力を受け，どちらのデータが正解かどうか判別を行う．このとき生成器は判別器が偽物であると区別できないような偽物のデータを生成することを目的として学習し，判別器は入力される真と偽のデータのうちどちらが真か見分けることを目的として学習を行う．この一連の動作を生成器と判別器で同時に繰り返すことによって，２つのネットワークを敵対的に学習させる生成モデルの一種である．

通常の敵対的生成ネットワーク(GAN)に複数のラベル付けされたデータ（例えば手書き数字の０～９の画像）を学習させた場合，生成される画像は完全にランダムである．そこで敵対的生成ネットワークを拡張したネットワークとしてCGAN（Conditional Generative Adversarial Networks)が提案されている[3]．CGANは従来の敵対的生成ネットワークの生成器と判別器に同じラベル情報の入力（例えば１を生成したい場合ラベル情報として1に相当する情報を入力）を行うことで，目的の画像を生成することができる(図１)．また生成器と判別器に畳み込み層など深層学習の手法を取り入れることで，性能を向上させたCGANをDCGAN（Deep Conditional Generative Adversarial Networks）という[4]．

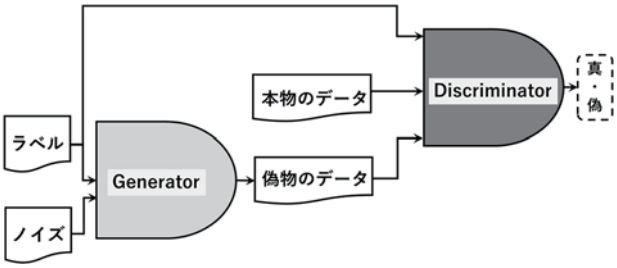

図1．CGANの構成図

3．提案手法

本研究では敵対的生成ネットワーク（Generative Adversarial Networks）[3]により，元々のデータセットにはない特徴量を再現できるのではないかと考えた．２章に述べた通り，敵対的生成ネットワークは偽物のデータを生成する生成器ネットワークと真偽の判別を行う判別器ネットワークを同時に学習することにより構成されている．したがってGANの学習が正しく進んだ場合，最終的にはGANの学習データ（真のデータ）の特徴量を再現した偽のデータを得ることができる．一方で敵対的生成ネットワークの学習途中では入力されている学習対象のデータ（正解データ）の特徴量を少しずつ反映した偽のデータが随時生成されていることになる．本研究では学習途中のGANが生成する偽のデータ（正解データの特徴量を少しずつ反映した偽のデータ）を使用することで，元の学習データのバリエーションを増やすこと提案する．

4. 敵対的生成ネットワークによる画像の生成

本研究では初めにGANを用いて手書き文字の生成を行った． A. Radfordらのモデルを参考に生成器（generator）と判別器（discriminator)は各4層の畳み込み層により構成されたDCGAN（Deep Convolutional GAN）を用いた[4]．

DCGANの学習には手書き数字の学習データセットであるMNISTデータセット[5]を使用した．学習環境及び学習パラメータについては下記のとおりである．

：DCGANの学習パラメータ

- ミニバッチサイズ　64
- 学習回数　2400回
- 学習データ　MNIST(28×28ピクセル)

：学習環境

- CPU　Intel Core i9-7920X
- GPU　GeForce GTX 1080Ti
- OS　Ubuntu 16.04 LTS

：深層学習ライブラリ

- TensorFlow（1.14.0）

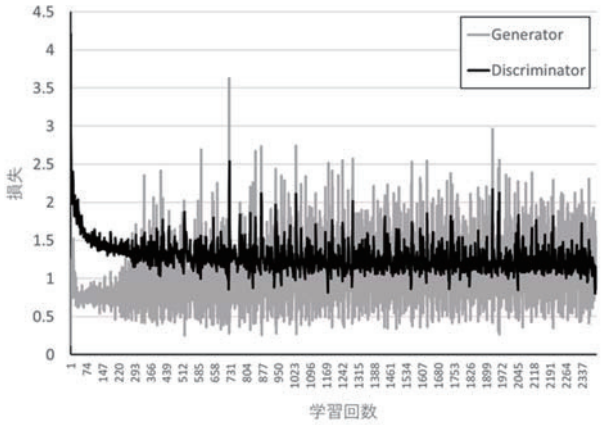

図2．DCGANの学習推移

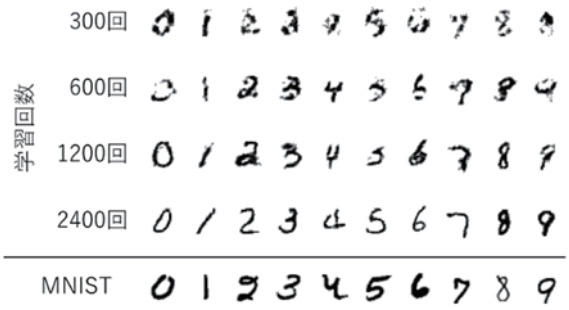

図3．DCGANのGeneratorにより生成された画像
最後の行はMNISTの例を示した．

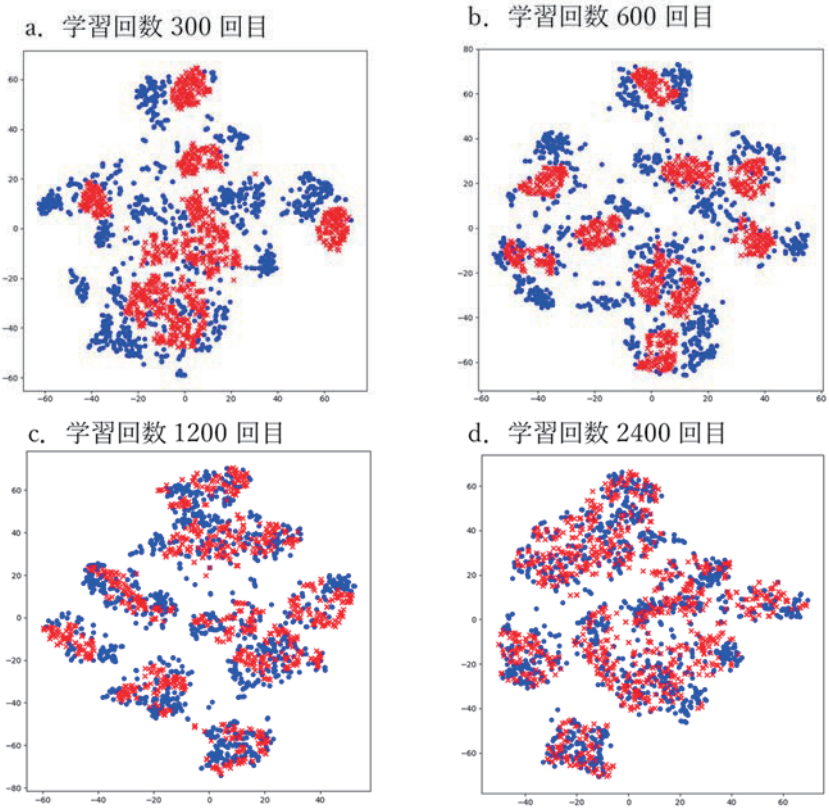

図4．t-SNEによる比較

a） DCGANの学習回数が 300回目のときの生成画像とMNISTをt-SNEにより解析した結果．青丸がMNIST,赤十字がDCGANを示している．

b～d） aと同様にDCGANの学習回数が600回目(b)，1200回目(c)，2400回目(d)のときの結果を示している．

またDCGANの目的関数は次式で与えられており，

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_{gan(z)}}\left[\log\left(1-D\left(G(z)\right)\right)\right] \quad (1),$$

となる．２つのネットワークのうち判別器は真と偽を正しく分類する確率($\log D(x)$)を最大化するように学習し，生成器は判別器が偽のデータを偽であると判別する確率 $\log(1-D(G(z)))$ を最小化するように学習を進める．なおzはノイズ入力，$G(z)$はノイズzにより生成された偽のデータ，xは真のデータ，D( )は判別器の識別結果(真と判断すれば１，偽と判断すれば０)を示している．

したがって生成器と判別器の学習時における損失関数は次の式で定義され，

$$Loss_G = \frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G(z)\right)\right) \quad (2),$$

$$Loss_D = \frac{1}{m}\sum_{i=1}^{m}\left[\log D(x) + \log\left(1-D\left(G(z)\right)\right)\right] \quad (3),$$

で表される．このとき$Loss_G$は生成器の損失関数であり，判別器が生成器の出力した偽データを真(=１)であると判断すると最小になる．またmはミニバッチのサイズを示している．一方で$Loss_D$は判別器の損失関数であり，$D\left(G(z)\right)$は判別器が偽のデータの識別結果が偽(=０)であると判別し，かつ，$D(x)$において真のデータを真であると識別できると最小となる．

表1．各テストデータに対する正答率(Accuracy)

※単位は×100％

	テストデータ				
	MNIST	**DCGAN300**	**DCGAN600**	**DCGAN1200**	**DCGAN2400**
①MNIST	0.915	0.831	0.822	0.954	0.950
②DCGAN300	0.731	1.000	0.647	0.744	0.727
③DCGAN600	0.682	0.679	1.000	0.698	0.697
④DCGAN1200	0.823	0.808	0.835	0.999	0.872
⑤DCGAN2400	0.884	0.831	0.883	0.954	0.992
⑥DCGAN300+MNIST	0.907	0.997	0.831	0.952	0.940
⑦DCGAN600+MNIST	0.908	0.839	0.998	0.957	0.941
⑧DCGAN1200+MNIST	0.911	0.842	0.884	0.995	0.944
⑨DCGAN2400+MNIST	0.912	0.847	0.893	0.967	0.982

表2．各テストデータに対する損失(Loss)

	テストデータ				
	MNIST	**DCGAN300**	**DCGAN600**	**DCGAN1200**	**DCGAN2400**
①MNIST	0.294	0.480	0.482	0.167	0.183
②DCGAN300	1.009	0.010	1.144	0.828	0.940
③DCGAN600	1.120	1.019	0.006	1.012	0.989
④DCGAN1200	0.614	0.554	0.511	0.014	0.402
⑤DCGAN2400	0.434	0.470	0.324	0.159	0.046
⑥DCGAN300+MNIST	0.328	0.046	0.481	0.173	0.205
⑦DCGAN600+MNIST	0.331	0.462	0.033	0.158	0.202
⑧DCGAN1200+MNIST	0.315	0.447	0.363	0.040	0.183
⑨DCGAN2400+MNIST	0.312	0.454	0.333	0.133	0.079

2400回学習したときの生成器（generator）と判別器（discriminator)の損失の推移は図２のようになった．GeneratorとDiscriminatorの損失は1.4付近を推移している．これは判別器における真偽を判断する学習と，生成器において判別されにくい偽データを生成する学習が拮抗しながら(敵対しながら)進んでいることを示している．また学習の途中で生成された画像は図３のとおりである．

生成された画像より，DCGANの学習回数が300回目では数字がはっきりとせず，学習回数が2400回目にはすでに学習データであるMNISTと区別することができないレベルであることがわかる．客観的に解析するためにt-SNE[6]による特徴量の比較を行った．DCGANの各学習段階において1000枚の画像を生成し，MNISTからランダムに1000枚抽出した画像と比較した（図４）．

t-SNEによる解析結果から，DCGANの学習回数が300回のときの生成されたデータはMNISTの特徴から最も離れており（図４a)，2400回目の生成データはほぼMNISTと傾向が同じであることがわかる（図４d)．

5．生成した画像による学習

DCGANにより生成した画像を用いて各テストデータに対する認識精度について検討を行った．使用したネットワークはTensorFlowのチュートリアルに付随するMNISTデータセットの数字を識別するためのサンプルコードmnist_softmax.pyを一部改変し使用した．このサンプルコードは28×28ピクセルの画像入力から10個の出力行い，ソフトマックス関数により，入力された画像が数字の０～９の内どれに相当するのか識別する．したがってこのネットワークは入力層と出力層のみの全結合ネットワークである．また損失関数は以下の式で定義される．

$$\mathrm{Loss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log \acute{y}_i) \qquad (4).$$

このときNはテストデータのサイズ，y_iは教師データ，$\acute{y}_i$は予測結果を示している．

通常，このネットワークでMNISTデータセットを学習した場合正解率は約９１％になる．本研究では，DCGANで生成した画像データが学習可能なデータであるか，また学習したモデルが他のデータセットに対してどのような認識精度を持っているのか検証した．ここでは学習パラメータは下記の通りとし，学習環境はDCGANの学習で使用したものと同一である．

：学習パラメータ

・バッチサイズ　100

・学習回数　1000回

：各学習データにより生成したモデルは下記の通りある

①MNIST　MNISTデータセットの画像により学習したモデル
（学習用 6万枚　テスト用 1万枚）

②DCGAN300　DCGANをMNISTにより300回学習させたネットワークを使用し生成された画像により学習したモデル．使用した枚数はMNISTと同じ学習用6万枚，テスト用1万枚である．

③DCGAN600　DCGANで600回学習したときに生成した画像により学習したモデル

④DCGAN1200　DCGANで1200回学習したときに生成した画像により学習したモデル

⑤DCGAN2400　DCGANで2400回学習したときに生成した画像により学習したモデル

⑥～⑨上記の各DCGAN生成画像にMNISTの画像を混ぜた学習データセットにより学習したモデル．DCGAN生成画像とMNIST画像の枚数の混合比率は1:1とした．また混合の後のデータセットの枚数はMNISTと同になるようにした．

各学習データで1000回学習させたモデル①～⑨に対して，それぞれのテストデータを使用して正答率(Accuracy)と損失(Loss)を算出した（表１，表２）.

正答率は予測した数字と正解が一致しているかどうかの指標であるため，その予測確率がどの程度か損失により評価した．損失については式４で定義されているように，ネットワークの出力である予測確率（どのクラスに属するか予測した結果）の対数をとった値と教師データである正解（０または１)との積ある．したがって損失が小さくかつ正答率が高い場合には，予測確率が高くかつその予測は正しいことを示している．表２にあるように，モデル①とモデル③～⑨はどのテストデータにも正答率が高く，かつ損失の値も小さくなったたため，予測確率が高く正しい予測が多いと考えられる．一方でDCGAN300のモデル②やDCGAN600のモデル③は全体的に正答率が低くかつ損失の値も大きいことから，予測確率が低く予測の間違いも多いことを示している.

正答率(表１)について詳細にみると，同じ学習データとテストデータの組み合わせ(例：MNISTで学習したモデルのMNISTのテストデータに対する正答率)のとき最も正答率が良い結果となった．一方で他のテストデータに対する正答率(例：MNISTで学習したモデル①のDCGAN300のテストデータに対する正答率)は低くなる傾向があった．これはすべてのモデルが学習データに対する過学習を起こしていることを示唆している.

学習データとしてDCGANの生成画像とMNISTの画像を混合させたものを使用した学習モデル（⑥～⑨）のうち，特にDCGAN2400＋MNISTを使用した学習モデル⑨はMNISTに対してもMNIST単体の学習モデルと同等の正答率であり，そのほかのDCGANで生成したテストデータに対してはMNIST単体の学習モデルより認識精度が良い結果となった．第４章で検討したように，DCGANを2400回学習させたときの生成画像とMNISTはt-SNEで解析を行っても客観的にわかるような特徴量の違いはなかった．しかしながらMNISTとDCGANを2400回学習させたときの生成画像は多少異なっている可能性がある．その結果，MNIST単体で作成した学習モデルの過学習を解消し，他のテストデータに対する正答率を向上させる効果があったと考えられる.

6. 他の手書き文字データセットに対する識別性能の検討

DCGANにより生成したデータセットでも文字認識を行うことができることが検証できた．次にMNIST以外の手書き文字データセットに対する識別性能を評価した．DCGANで生成される手書き文字は2400回程度学習させればNMISTと同様な画像となるが，学習回数300回目や600回目ではかすれたような文字となっている．そこでこの特徴を活かすことができないか検証するために，FaxOCRデータセットに対する正答率を調べた．FaxOCRデータセットは災害時にFaxの活用を目的としたオープンソースのプロジェクトが公開しているデータセットであり，Faxで送信された文字による０～９の数字データで構成されている[7]．本研究では，Faxのようにコピーと印刷によりかすれた文字はDCGANで生成したデータセットでより精度よく認識できる可能性があると考えた.

認識精度を確認する前にFaxOCRデータセットに事前処理を行った．MNISTデータセットは20×20ピクセルの数字画像の重心を28×28ピクセルの中心に来るように作成されている．FaxOCRデータセットに対しても同様の処理を行った．図５にした事前処理をしたFaxOCRデータセットの例を示す.

図5. FaxOCRデータセットとMNISTデータセット，DCGANで生成した画像との比較

FaxOCRデータセットに含まれるテスト画像は249枚である．本研究では５章で構築した各モデルにこのテストデータを入力し正答率を算出した（図６）．

その結果，わずかな差ではあるが，MNISTだけを使用し構築したモデルより，DCGANを2400回学習させ生成した画像により構築したモデルのほうが約3%正答率の向上が見られた．

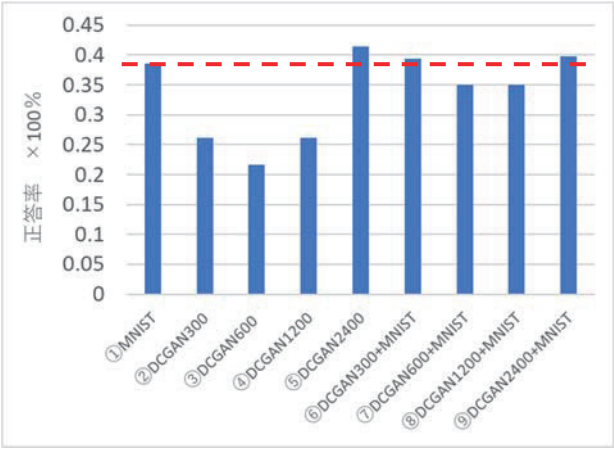

図6. FaxOCRのテストデータに対する各モデルの正答率
赤い線はMNISTだけで学習したモデル①の正答率を示している．

7. 考察・まとめ

本研究では，敵対的生成ネットワークに着目し，古典的な学習データの水増し（画像の平行移動，回転，反転，劣化処理など）ではなく，生成モデルの一種である敵対的生成ネットワークの学習過程を利用し，学習データそのもの生成を試みた．

敵対的生成ネットワークにMNISTデータセットを入力すると先行研究でも示されている通り，MNISTに近い画像を生成できた．t-SNEによりその画像を解析した結果，DCGANを300回学習したとき生成された画像はMNISTの特徴から離れており，2400回学習したときに生成した画像では，MNISTとほとんど同じであることが分かった．したがってDCGANの学習過程ではMNISTとは違う特徴を持つが人間の目では“数字”として認識可能な手書き文字が生成できていることが分かった．

人間の目では“数字”として認識可能な手書き文字が生成できているということは，MNISTデータセットにはない特徴を持つ手書き文字を認識できると考えた．そこで，MNIST以外の他の手書き文字データセットに対する識別性能を評価した．本研究ではFaxOCRデータセットに対する識別性能を評価した．その結果，わずかな差ではあるが，MNISTだけを使用し構築したモデルより，DCGANを2400回学習させ生成した画像により構築したモデルのほうが約３%正答率の向上が見られた．これはMNISTにはない特徴量を，DCGANを2400回学習させ生成した画像データセットは持っていることを示唆している．

以上の結果から敵対的生成ネットワークを使用して，元の学習データとは視覚的には識別可能でも特徴量が異なるデータセットを作成可能であることを示した．敵対的生成ネットワークは学習の仕方を工夫することにより複数の学習データの特徴を混ぜることができることも示唆されている[8]．今後はこのような手法をさらに取り入れることで，学習データを集めるコストを削減し，より高品質な学習データを作るうえで重要なツールになると考えられる．

参考文献

1) M. Hayat, S. Khan, W. Zamir, J. Shen, L. Shao, Max-margin class imbalanced learning with gaussian affinity, 2019, arXiv:1901.07711

2) 山下拓朗,柳澤秀彰,渡辺裕.深層学習を用いたマンガキャラクタの検出における顔変形の影響評価, 情報処理学会第80回全国大会, 2018, 5Y-03

3) T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved Techniques for Training GANs, 2016, arXiv:1606.03498

4) A. Radford, L. Metz, S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015, arXiv:1511.06434

5) Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, 86(11):2278-2324, 1998

6) L. Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research, vol.9:2579-2605, 2008

7) “Shinsai FaxOCR”, https://sites.google.com/site/faxocr2010/home/about-faxocr

8) A. Hertzmann, Visual Indeterminacy in Generative Neural Art, 2019, arXiv:1910.04639

2020年2月28日原稿受付, 2020年3月17日採録決定

Received, February 28,2020; accepted, March 17,2020