

Generative Adversarial Networks の共同学習

塚原 拓也[†] 平川 翼[†] 山下 隆義[†] 藤吉 弘亘[†]

[†] 中部大学 〒487-8501 愛知県春日井市松本町 1200

E-mail: [†] {tsukahara,hirakawa}@mprg.cs.chubu.ac.jp, {takayoshi,fujiyoshi}@isc.chubu.ac.jp

あらまし Generative Adversarial Networks (GANs) は生成モデルおよび識別モデルを敵対的に学習させることで、実在しない画像を生成する手法である。GANs は単一の生成モデルと識別モデルを用いており、複数のモデルそれぞれの特性を活かした学習が行えていないため、性能を最大限に発揮できていないと考えられる。一方、画像分類問題では、複数のニューラルネットワーク間において知識転移をする共同学習により、認識精度が向上することが知られている。そこで、本研究では複数の生成モデルと1つの識別モデルによる GANs により、それぞれの生成モデルで情報を転移しながら共同学習を行う手法を提案する。評価実験により、生成モデルが生成する画像の質が向上することを確認した。

キーワード 敵対的生成モデル, 相互学習, 深層学習, 畳み込みニューラルネットワーク

Collaborative learning for generative adversarial networks

Takuya TSUKAHARA[†], Tsubasa HIRAKAWA[†],

Takayoshi YAMASHITA[†], and Hironobu FUJIYOSHI[†]

[†] Chubu University 1200 Matsumoto, Kasugai, Aichi, 487-851 Japan

E-mail: [†] {tsukahara,hirakawa}@mprg.cs.chubu.ac.jp, {takayoshi,fujiyoshi}@isc.chubu.ac.jp

Abstract Generative adversarial networks (GANs) adversarially trains generative and discriminative models. And this is how to generate a nonexistent image. Common GANs use only a single generative model or discriminant model, and are considered to be unable to maximize their performance. On the other hand, in the image classification problem, it is known that the recognition accuracy is improved by collaborative learning in which knowledge transfer is performed among a plurality of neural networks. Therefore, in this research, we propose a method that uses multiple generative models and one discriminant model to perform collaborative learning while transferring information in each generative model. Experimental results show that the quality of images generated by the proposed method was improved.

Key words Generative adversarial networks, Deep mutual learning, Deep learning, Convolutional neural network

1. はじめに

一般社会において、優れた教師の存在は、生徒の学習効果に大きく寄与する。また、学習時における生徒同士の適切な情報交換も、学習効率の向上に効果的である。これらの一般的な考え方は、ニューラルネットワークの学習においても同様のことが言える。画像分類問題における教師あり学習は、通常、教師ラベルを用いて単一のニューラルネットワークで学習を行う。複数のニューラルネットワーク間において知識転移をする共同学習では、相互に知識転移した学習を行うことで、通常の教師あり学習と比較して認識精度が向上することが知られている [1]~[5]。

ニューラルネットワークにおける知識の転移には、一方向性の Knowledge Distillation (KD) [1] および双方向性の Deep Mutual

Learning (DML) [5] がある。KD は、大規模で認識精度の優れた事前学習済みのニューラルネットワークである teacher ネットワークを使用し、軽量のニューラルネットワークである student ネットワークを学習する手法である。これにより、通常の教師あり学習を行った student ネットワークと比較して優れた性能を獲得できる。DML は student ネットワーク同士で相互学習を行う。構造の異なる student ネットワークの組み合わせによる学習のみならず、同一構造のネットワーク同士においても認識精度の向上に効果的であり、3つ以上のニューラルネットワークを同時に学習することが可能である。

Generative Adversarial Networks (GANs) [6] は、生成モデルおよび識別モデルを敵対的に学習させることで、実在しない画像を生成する手法である。識別モデルである Discriminator では、実画像と生成モデルにより生成された生成画像を識別でき

るように学習される。生成モデルである Generator は、潜在変数を用いて画像を生成し、生成画像が識別モデルにより実画像と識別されるように学習される。これにより、GANs は実在しないデータの生成が可能となる。高精細な画像の生成が可能な GANs には、Deep Convolutional GANs (DCGANs) [7], Progressive Growing of GANs (Progressive GANs) [8], Self-Attention GANs (SAGANs) [9] 等がある。DCGANs は Convolutional Neural Network (CNN) [10] を用いることで、Progressive GANs は Generator および Discriminator を学習と共に深くすることで、Self-Attention GAN は Self-Attention 機構を導入することで生成画像の質を向上させている。これらの手法は、ニューラルネットワークの構造を工夫することで高精細な画像の生成を実現しているが、複数のニューラルネットワークを用いておらず、それぞれのニューラルネットワークの特性をうまく活かすことができない。そのため、画像分類問題における通常の教師あり学習と同様にニューラルネットワークの性能を最大限に発揮できていないと考えられる。

GANs に対して互いのニューラルネットワークの知識を転移させながら学習を行う仕組みを導入することで、ニューラルネットワークの性能を最大限に発揮することが可能となると思われる。そこで本研究では、複数の Generator を用いて学習を行い、Generator が互いの知識を転移しながら共同学習を行う手法を提案する。これにより、複数の Generator それぞれの特性を活かした学習を実現する。また、Generator 同士での知識の伝達方法を最適化することによりニューラルネットワークの性能を最大限に発揮させ、生成画像の質の向上が期待できる。評価実験により、知識の転移による効果および提案手法の有効性を示す。

2. 関連研究

画像分類問題における共同学習には、KD および DML を内包しつつ、様々な知識の転移を用いた学習を統一的に表現可能な手法として知識転移グラフ [11] がある。また、GANs において複数のニューラルネットワークを用いた手法が提案されている [12]~[15]。複数の Generator を用いて学習を行う手法として、Multi-Agent Diverse GANs (MAD-GANs) [12] および Mixture GANs (MGANs) [13] がある。一方で、2つの Discriminator を用いて学習を行う手法として Dual Discriminator GANs (D2GANs) [14] があり、複数の Discriminator を用いて学習を行う手法として Generative Multi-Adversarial Networks (GMAN) [15] がある。ここでは本研究に、より関連度の高い知識転移グラフおよび MAD-GANs について説明する。

2.1 知識転移グラフ

知識転移グラフは、画像分類問題において KD および DML を内包しつつ、様々な知識の転移を用いた学習を統一的に表現することができる。3つのニューラルネットワークを用いた場合における知識転移グラフを図1に示す。学習に使用するニューラルネットワークの数を k とすると、各ニューラルネットワークは $m_1, \dots, m_k$ で表され、ノードとして表す。また、各ノードの間には2つのエッジを定義し、それぞれ異なる向きの有向グ

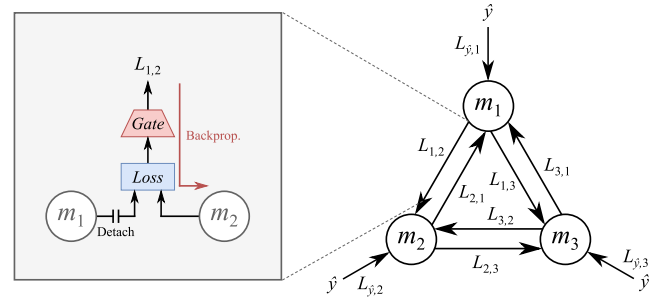


図1: 知識転移グラフ

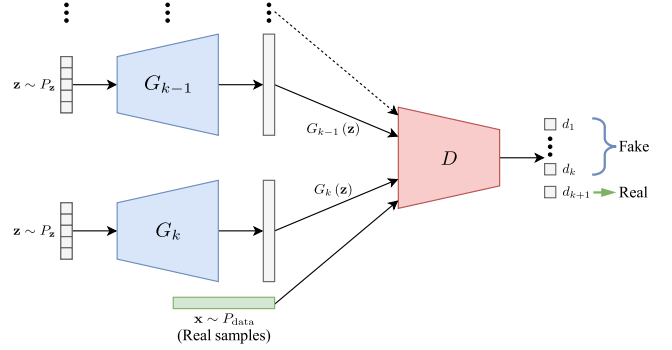


図2: MAD-GANs のネットワーク構造

ラフで表す。このエッジは、学習時に勾配の情報が伝わる向きを表している。各エッジには個別の損失関数を定義し、損失関数の内部にゲート関数を導入することで、伝播される損失を制御する。このような表現を用いることで、KD および DML を表現することが可能となる。また、選択するモデルや損失関数の組合せによって、多様な学習方法を表現できる。

2.2 Multi-Agent Diverse GANs

MAD-GANs は、複数の Generator および 1つの Discriminator で構成された GANs の手法である。MAD-GANs のネットワーク構造を図2に示す。各 Generator は潜在変数を用いて生成画像を生成し、生成画像が Discriminator により実画像と識別されるように学習される。Discriminator は実画像と生成画像を識別するのみならず、どの Generator から生成された生成画像であるかを識別するように学習する。これにより、Generator が似た画像ばかり生成すると Discriminator により生成画像と識別されやすくなるため、Generator はより多様な生成画像を生成するように学習される。GANs の学習において問題とされるモード崩壊 [16] は、Generator が多様性を失うことが原因のため、MAD-GANs を用いることでモード崩壊の発生を抑制することが可能となる。

3. 提案手法

本研究では複数の Generator を用いて学習を行い、Generator が互いの知識を転移しながら学習を行う GANs の共同学習を提案する。ここでは、提案手法の構築方法、知識転移グラフの導入、学習方法について説明する。

3.1 構築方法

提案手法は、複数の Generator および 1つの Discriminator で

構成される。また、ベースである GANs の手法に対していくつかの変更を行う必要がある。提案手法のネットワーク構造を図 3 に示す。

図 3 において、学習に使用する Generator の数を k とすると、各 Generator は $G_1, \dots, G_k$ で表される。各 Generator は、潜在変数 $\mathbf{z} \sim P_{\mathbf{z}}$ を用いて生成画像を生成する。そのため、 k 枚の生成画像が得られる。また、各 Generator の特徴マップまたは生成画像を知識転移グラフを用いて転移させることで、Generator 同士での知識の伝達を可能にしている。

図 3 において、Discriminator は D で表される。Discriminator は実画像と生成画像を識別するのみならず、どの Generator から生成された生成画像であるかを識別する必要がある。そのため、最終層の出力値のユニットの数は $k+1$ 個となっており、ソフトマックス関数を用いてクラススコア $d_1, \dots, d_{k+1}$ を出力する。スコア $d_1, \dots, d_k$ は各 Generator から生成された生成画像である確率を表し、スコア d_{k+1} は実画像である確率を表す。

ネットワーク内部には、学習の安定化を図るため、Batch Normalization [17] が用いられる場合が多い。しかし、GANs における学習の安定化には Batch Normalization は不十分である。そこで、本手法では Discriminator に Spectral Normalization (SN) [18] を導入する。ベースである GANs の手法が Discriminator に Batch Normalization を用いている場合は、SN に変更する。Discriminator に SN を導入することで、リップシツ連続を満たすという制約を設けることができる。これにより、損失関数に変更を加えることなく Wasserstein GANs (WGAN) [19] や gradient penalty [20] を導入した WGAN (WGAN-GP) と同様にモード崩壊を抑制し、GANs における学習の安定化を図ることが可能となる。

3.2 知識転移グラフの導入

提案手法は、知識転移グラフを導入することで、各 Generator が互いの知識を転移しながら学習することを可能にしている。3 つの Generator を用いた場合における知識転移グラフを導入した提案手法を図 4 に示す。学習に使用する Generator の数を k とすると、各 Generator は $G_1, \dots, G_k$ 、Discriminator は D となり、ノードとして表す。各 Generator のノード間には 2 つのエッジを定義し、それぞれ異なる向きの有向グラフで表す。各エッジの向きは、学習時に勾配の情報が伝わる方向を表している。各 Generator のノード間のエッジは Generator 同士の知識の伝達を表し、 i 番目の Generator のエッジ $L_{D,i}$ は i 番目の Generator が生成した生成画像が Discriminator に実画像と判断されたかどうかの情報の伝達を表す。また、エッジ $L_{\text{adversarial}}$ は、Discriminator の識別結果が正しいかどうかの情報の伝達を表す。各エッジには個別の損失関数を定義し、各 Generator の方向を向いたエッジの損失関数にはゲート関数を導入することで、伝播される損失を制御する。

本手法では、ゲート関数として Through Gate および Cutoff Gate を用いる。Through Gate は、入力されたサンプルごとの損失をそのまま通すゲートである。一方で、Cutoff Gate は損失を 0 とし、損失の計算を行わないゲートである。これにより、任意のエッジを切断することができる。

3.3 学習方法

提案手法では、Discriminator の学習および Generator の学習を繰り返すことで GANs の学習を行う。学習はすべての実画像に対しての学習が終わるまで繰り返し行い、終わった時点で 1 回の学習となる。Discriminator の学習および Generator 学習について以下で説明する。

3.3.1 Discriminator の学習

Discriminator は実画像とどの Generator から生成された生成画像であるかを識別するように学習を行う。使用する Generator の数を k とすると、Discriminator に対して実画像または Generator で生成された生成画像を入力した際の識別結果は $\mathbf{d} = d_1, \dots, d_{k+1}$ で表される。また、正解ラベルは $\mathbf{t}^d = t_1^d, \dots, t_{k+1}^d$ で表され、Discriminator に対して i 番目の Generator で生成された生成画像を入力した際は t_i^d が 1 でそのほかが 0 のベクトルとなり、実画像を入力した際は t_{k+1}^d が 1 でそのほかが 0 のベクトルとなる。Discriminator の学習に用いる損失関数 $L_{\text{adversarial}}$ は式 (1) で表される。

$$L_{\text{adversarial}} = - \sum_{m=1}^{k+1} t_m^d \log_e d_m \quad (1)$$

Discriminator の学習は以下の 3 つの手順で行われる。

手順 1 Discriminator へ実画像を入力し、損失関数 $L_{\text{adversarial}}$ を用いて Discriminator の勾配を更新する。

手順 2 Discriminator へ i 番目の Generator で生成した生成画像を入力し、損失関数 $L_{\text{adversarial}}$ を用いて Discriminator の勾配を更新する。このとき、Generator へ入力する潜在変数は各 Generator で異なる。

手順 3 手順 1 および手順 2 を k 回繰り返すことで、すべての Generator で生成した画像に対して学習を行う。

3.3.2 Generator 学習

各 Generator は、生成画像が Discriminator により実画像と識別されるように学習を行う。また、提案手法では知識転移グラフに応じて、各 Generator が互いの知識を転移しながら学習する。各 Generator のノード間のエッジは Generator 同士の知識の伝達を表し、 i 番目の Generator のエッジ $L_{D,i}$ は i 番目の Generator が生成した生成画像が Discriminator に実画像と判断されたかどうかの情報の伝達を表す。Generator のノード間のエッジで、 t 番目の Generator の知識を s 番目の Generator に転移する際の損失関数を $L_{t,s}$ とし、 i 番目の Generator が生成した生成画像が Discriminator に実画像と判断されたかどうかを表す損失関数を $L_{D,i}$ とする。また、各エッジにおけるゲート関数は $\text{Gate}(\cdot)$ とする。Generator の学習では、各 Generator に入力される潜在変数 $\mathbf{z} \sim P_{\mathbf{z}}$ は同一とする。最終的な各 Generator の損失関数はエッジから得られた $L_{t,s}$ および $L_{D,i}$ の総和から算出される。

使用する Generator の数を k とすると、Discriminator に対して i 番目の Generator で生成された生成画像を入力した際の識別結果は $\mathbf{d}^{(i)} = d_1^{(i)}, \dots, d_{k+1}^{(i)}$ で表される。また、正解ラベルは $\mathbf{t}^g = t_1^g, \dots, t_{k+1}^g$ で表され、 t_{k+1}^g が 1 でそのほかが 0 のベクトルとなる。このとき、損失関数 $L_{D,i}$ は式 (2) で表される。

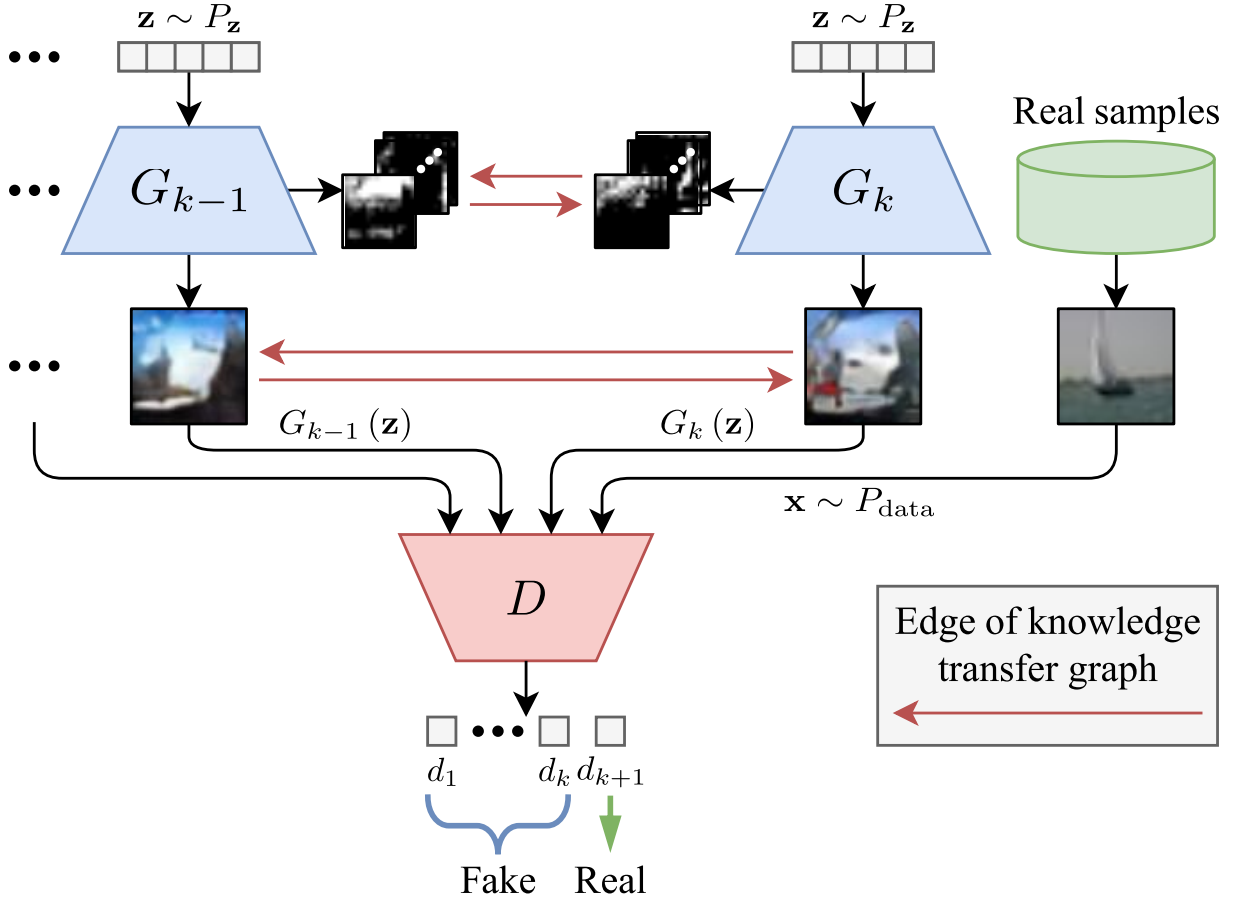


図 3: 提案手法のネットワーク構造

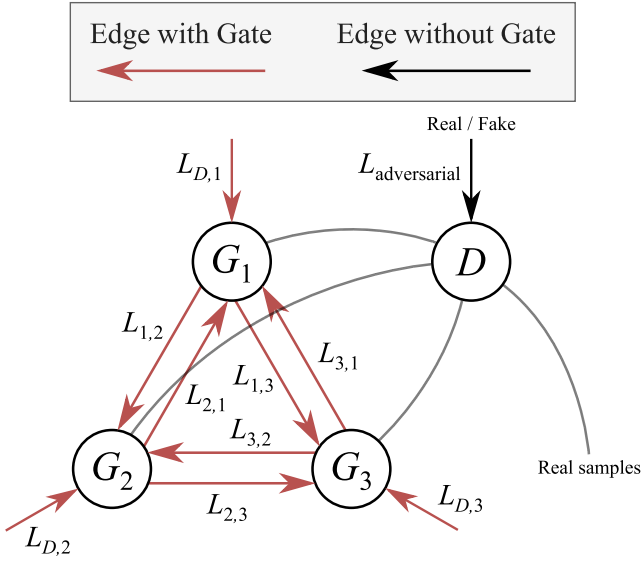


図 4: GANs に知識転移グラフを導入した提案手法 ($k=3$)

$$L_{D,i} = \text{Gate} \left(- \sum_{m=1}^{k+1} t_m^g \log_e d_m^{(i)} \right) \quad (2)$$

提案手法には、特徴マップを転移する手法と生成画像を転移する手法の2種類があり、それぞれの手法で損失関数 $L_{t,s}$ が異なる。特徴マップのチャンネル番号を $c = 1, \dots, C$ 、特徴マップの層の位置を $m = 1, \dots, M$ 、特徴マップのベクトル番

号を $n = 1, \dots, N$ とすると、 i 番目の Generator の特徴マップは $F_{i,m,c,n}$ で表される。このとき、ハイパーパラメータを α とすると、特徴マップを転移する手法の損失関数 $L_{t,s}$ は式 (3) および式 (4) で表される。

$$Q_i^{(m,n)} = \frac{1}{C} \sum_c F_{i,m,c,n}^2 \quad (3)$$

$$L_{t,s} = \text{Gate} \left(\frac{\alpha}{MN} \sum_m \sum_n \left(\frac{Q_s^{(m,n)}}{\|Q_s^{(m,n)}\|_2} - \frac{Q_t^{(m,n)}}{\|Q_t^{(m,n)}\|_2} \right)^2 \right) \quad (4)$$

生成画像のベクトル番号を $n = 1, \dots, N$ とし、潜在変数 $\mathbf{z} \sim P_z$ を i 番目の Generator に入力した際の生成画像を $G_i^{(n)}(\mathbf{z})$ とする。このとき、ハイパーパラメータを α とすると、生成画像を転移する手法の損失関数 $L_{t,s}$ は式 (5) で表される。

$$L_{t,s} = \text{Gate} \left(\frac{\alpha}{N} \sum_n \left(\frac{G_s^{(n)}(\mathbf{z})}{\|G_s^{(n)}(\mathbf{z})\|_2} - \frac{G_t^{(n)}(\mathbf{z})}{\|G_t^{(n)}(\mathbf{z})\|_2} \right)^2 \right) \quad (5)$$

4. 評価実験

従来手法と比較を行うことにより、提案手法の有効性を調査し、知識を転移させながら学習を行う仕組みを導入することによる有効性を調査する。また、提案手法に知識転移グラフを導入し傾向を調査することにより、最適な知識の転移方法を探る。

表 1: 知識の転移による効果

	SN	知識の転移	IS $\uparrow$	FID $\downarrow$
DCGANs		—	4.57 ± 0.03	25.79
	✓	—	4.77 ± 0.05	21.17
MAD-GANs	✓		4.81 ± 0.05	27.02
提案手法	✓	✓	5.10 ± 0.07	21.31
実画像	—	—	8.70 ± 0.14	—

4.1 実験条件

実験には、CIFAR-10 データセット [21] を用いる。CIFAR-10 データは、10 クラスの一般物体のカラー画像で構成されており、学習用のサンプルとして 50,000 枚、評価用のサンプルとして 10,000 枚の画像が含まれている。本実験では、学習用のサンプルである 50,000 枚のみを使用する。また、CIFAR-10 データの画像サイズは 32×32 ピクセルだが、 64×64 にリサイズして使用する。

本実験では、ベースの GANs の手法として DCGANs を用いる。また、知識転移グラフにおけるゲート関数の探索回数を 1,500 とし、各 Generator において最大となる Inception Score (IS) [22] を最大化するゲート関数の組み合わせを探索する。

学習時におけるバッチサイズを 256 とし、学習回数を 50、各 Generator に入力される潜在変数の次元数を 100、ハイパーパラメータ α を 1,000 とする。

生成画像の質を評価する指標として IS および Fréchet Inception Distance (FID) [23] を用いる。IS は、ニューラルネットワークによって画像分類がしやすく、生成画像のクラスが豊富なほど生成画像の質が高く評価される。また、IS は高い値になるほど生成画像の質が高いことを表している。FID は、IS が実画像の分布を考慮していないという欠点を改善し、実画像と生成画像の分布間の距離により、生成画像の質が評価される。また、FID は低い値になるほど生成画像の質が高いことを表している。

4.2 知識転移の効果の調査

Generator が知識の転移を行うことによる効果を調査する。本実験では、DCGANs および MAD-GANs、提案手法を用いる。DCGANs は SN の導入の有無による比較を行い、MAD-GANs および提案手法は SN を導入して比較を行う。MAD-GANs は、2つの Generator を用いる。提案手法は、2つの Generator を用い、生成画像を転移する手法を用いる。また、本実験ではゲート関数の探索は行わず、Through Gate のみを用いる。学習回数を 200 Epoch とし、IS の最も高い時点でのモデルパラメータを用いて評価を行う。IS および FID を比較した結果を表 1 に示す。結果から、IS で用いた評価では、提案手法が最も生成画像の質が高いことが確認できる。また、FID を用いた評価では SN を導入した DCGANs が最も質が高い画像が生成できるが、提案手法においても同程度の画像が生成できることが確認できる。MAD-GANs と提案手法の構造は同じだが、生成画像を転移して学習を行うことで、大幅な改善が見られる。よって、Generator の知識の転移は GANs における生成画像の質を向上させることが可能である。

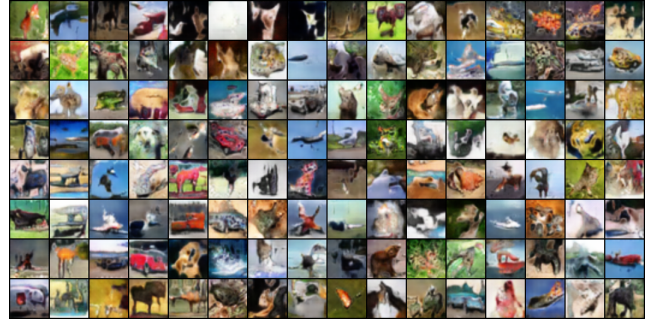
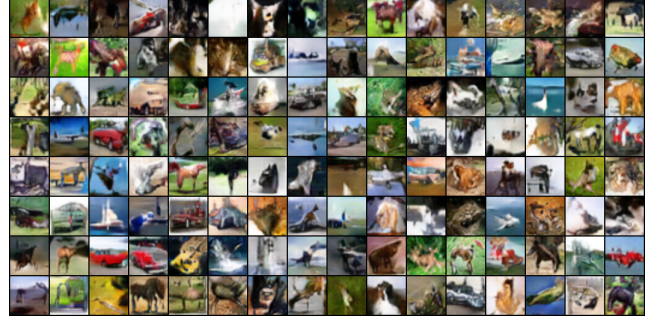
(a) 1 つ目の Generator の生成画像 $G_1(\mathbf{z})$ (b) 2 つ目の Generator の生成画像 $G_2(\mathbf{z})$

図 5: 生成画像の可視化

提案手法において、1 つ目および 2 つ目の Generator に同一の潜在変数 $\mathbf{z} \sim P_{\mathbf{z}}$ を 128 通り入力した際の生成画像を可視化する。Generator によって生成された 128 枚の生成画像を可視化した結果を表 5 に示す。各 Generator における同じ位置の画像が同じ潜在変数を入力した際の生成画像に対応している。1 つ目および 2 つ目の Generator の生成画像を比較すると、同じ潜在変数を入力した際の画像が似ていることが確認できる。これは、互いの生成画像を転移することで似た画像を生成するように学習され、Generator の特性が似るためだと思われる。

4.3 知識転移グラフの導入による傾向の調査

知識転移グラフを導入した提案手法を用いて Generator が知識の転移を行う際の傾向を調査する。本実験では、DCGANs および MAD-GANs、提案手法を用いる。DCGANs は SN の導入の有無による比較を行い、MAD-GANs および提案手法は SN を導入して比較を行う。MAD-GANs および提案手法は、3つの Generator を用いる。提案手法は、生成画像を転移する手法および入力層付近 (DCGANs の 2・3 層目) の特徴マップを転移する手法、出力層付近 (DCGANs の 3・4 層目) の特徴マップを転移する手法を用い、それぞれの傾向を調査する。また、本実験ではゲート関数として Through Gate および Cutoff Gate を用い、知識転移グラフを用いた探索を行う。学習回数を 50 Epoch とし、IS の最も高い時点でのモデルパラメータを用いて評価を行う。IS および FID を比較した結果を表 2 に示す。結果から、いずれの指標においても、出力層付近の特徴マップを用いた提案手法が最も生成画像の質が高いことが確認できる。よって、出力層付近の特徴マップを転移した学習は、生成画像の質を向上に有効的に働く。

表 2: 知識転移グラフの導入による傾向

	SN	知識の転移	IS $\uparrow$	FID $\downarrow$
DCGANs		—	3.90 ± 0.03	53.44
	✓	—	4.12 ± 0.07	40.91
MAD-GANs	✓		3.99 ± 0.02	49.95
提案手法	✓	生成画像	4.42 ± 0.04	46.06
	✓	入力層付近	4.46 ± 0.04	50.09
	✓	出力層付近	4.48 ± 0.05	40.93
実画像	—	—	8.70 ± 0.14	—

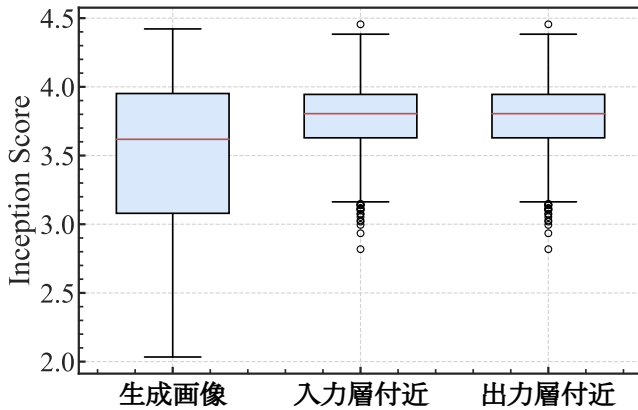


図 6: 提案手法における各手法の傾向

各提案手法において、知識転移グラフを用いたゲート関数の探索を 1,500 回行った際に、IS の傾向を調査した。各探索における IS のばらつきを図 6 に示す。生成画像を転移した手法と比較して、特徴マップを転移した手法は IS が高い値で推移しており、学習が安定する傾向があることが確認できる。また、入力層付近および出力層付近の特徴マップを転移した手法は IS の分布の傾向が似ているが、表 2 における FID から出力層付近の特徴マップを転移させた方が質に高い画像を生成していることが確認できる。

5. おわりに

本研究では複数の生成モデルおよび 1 つの識別モデルを用いて、それぞれの生成モデルで情報を転移しながら共同学習を行う手法を提案した。評価実験により複数の Generator を用いて知識の転移を行うことにより、生成モデルが生成する画像の質が向上することを確認した。また、特徴マップを転移した手法は学習が安定化する傾向があることを確認した。今後は複数の Generator および複数の Discriminator を用いて、Generator および Discriminator にそれぞれ知識転移グラフを導入し、最適な知識の転移方法を探索することで、さらなる生成画像の質の向上を目指す。

文 献

- [1] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” NIPS Deep Learning and Representation Learning Workshop, 2014.
- [2] A. Romero, N. Ballas, S.E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, “Fitnets: Hints for thin deep nets,” International Conference on Learning Representations, 2015.
- [3] X. Zhu, S. Gong, et al., “Knowledge distillation by on-the-fly native

ensemble,” Neural Information Processing Systems, pp.7517–7527, 2018.

- [4] G. Song and W. Chai, “Collaborative learning for deep neural networks,” Neural Information Processing Systems, pp.1832–1841, 2018.
- [5] Y. Zhang, T. Xiang, T.M. Hospedales, and H. Lu, “Deep mutual learning,” Computer Vision and Pattern Recognition, pp.4320–4328, 2018.
- [6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” Neural Information Processing Systems, pp.2672–2680, 2014.
- [7] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” International Conference on Learning Representations, 2016.
- [8] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” International Conference on Learning Representations, 2018.
- [9] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” International Conference on Machine Learning, pp.7354–7363, 2019.
- [10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol.86, no.11, pp.2278–2324, 1998.
- [11] S. Minami, T. Hiraoka, T. Yamashita, and H. Fujiyoshi, “Knowledge transfer graph for deep collaborative learning,” arXiv preprint arXiv:1909.04286, 2019.
- [12] A. Ghosh, V. Kulharia, V.P. Namboodiri, P.H. Torr, and P.K. Dokania, “Multi-agent diverse generative adversarial networks,” Computer Vision and Pattern Recognition, pp.8513–8521, 2018.
- [13] Q. Hoang, T.D. Nguyen, T. Le, and D. Phung, “Mgan: Training generative adversarial nets with multiple generators,” International Conference on Learning Representations, 2018.
- [14] T. Nguyen, T. Le, H. Vu, and D. Phung, “Dual discriminator generative adversarial nets,” Neural Information Processing Systems, pp.2670–2680, 2017.
- [15] I. Durugkar, I. Gemp, and S. Mahadevan, “Generative multi-adversarial networks,” International Conference on Learning Representations, 2017.
- [16] I. Goodfellow, “Nips 2016 tutorial: Generative adversarial networks,” arXiv preprint arXiv:1701.00160, 2016.
- [17] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” International Conference on Machine Learning, vol.37, pp.44–456, 2015.
- [18] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” International Conference on Learning Representations, 2018.
- [19] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” International Conference on Machine Learning, vol.70, pp.214–223, 2017.
- [20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A.C. Courville, “Improved training of wasserstein gans,” Neural Information Processing Systems, pp.5767–5777, 2017.
- [21] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Technical Report, University of Tront, 2009.
- [22] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” Neural Information Processing Systems, pp.2234–2242, 2016.
- [23] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” Neural Information Processing Systems, pp.6626–6637, 2017.