

GANにおける多様な画像生成のための損失関数の提案

A Proposal of a loss function of GAN to generate various images

児玉涼次*1

Kodama Ryoji

中村剛士*1

Tsuyoshi Nakamura

加納政芳*2

Masayoshi Kanoh

山田晃嗣*3

Koji Yamada

*1名古屋工業大学

Nagoya Institute of Technology

*2中京大学

Chukyo University

*3情報科学芸術大学院大学

Institute of Advanced Media Arts and Sciences

We propose a new loss function based on variance of generated images that we introduce to GAN(Generative Adversarial Networks). Recent image-generation methods adopt a neural-network based generative model. Auto-generators of illustrations can contribute to assist creative activities and entertainment. This paper focuses on suppression of collapse and its benefit to GAN training. Using our new technique, we attempted to generate various images of illustrations.

1. はじめに

画像生成とは、コンピュータグラフィクス(CG)によって様々な画像の自動生成を行う技術である。特に、絵画やイラストなどの創作画像を生成する研究分野は、Non-Photorealistic Rendering(NPR)と呼ばれ、CG研究において注目される分野である。画像生成という創作活動は、計算で表すことが容易でないが、Neural Network(NN)を用いたGANの技術によって、その実現可能性が示されている[Goodfellow 14]。GANの発展形であるDeep Convolutional GAN(DCGAN)[Radford 15]を用いた手法も存在する。

GANの学習においてcollapseという現象[Salimans 16]が確認されていて、パラメータに依らず類似した画像ばかりを生成する現象である。また文献[Arora 17]では、GANの生成データが学習データに比べて多様性に欠けると報告されている。これらの関連研究から示される課題は、GANの学習過程において生成データの多様性が低下し得るということである。

そこで本研究はGANにイラストを学習させ、生成画像における多様性を向上させることを目的として、分散を考慮した損失関数の導入を提案する。この提案により、コミックに描かれた絵柄を学習させた際にも、それに類似しつつ同一でない多様な絵柄の獲得が期待できる。

2. 損失関数に対する分散の導入

GANにおける生成画像の多様性を向上させる為に、生成画像の分散に着目した手法を提案する。GANによって生成される画像群が正規分布し、分散が生成データの多様性に関係していると仮定する。分散を大きくするため、生成画像の分散をGANの損失関数に加えることを考える。

式(1)では生成画像のリスト X を定義する。

$$X \stackrel{\text{def}}{=} [x_1, x_2, \dots, x_n, \dots, x_N] \quad (1)$$

生成画像の総数を N とし、そのうちの一つを x_n とする。 x_n はカラー画像であるためサイズ $C \times H \times W$ の3階テンソルと捉え、 C はカラーチャンネル数、 H と W は縦横の長さとする。 N, C, H, W は1以上の整数とし、 x_n の一画素を $x_{n,c,h,w}$ と表す。

式(2)では分散計算のために x_n のノルムを計算する。

$$\|x_n\| = \sqrt{\sum_{c,h,w} (x_{n,c,h,w})^2} \quad (2)$$

式(3),(4)は生成画像における平均 μ_X と分散 Σ_X を示す。分散は平均とノルムの差 σ_n を定義し、その2乗和の平均として計算している。 σ_n は後の正規化のために定義した。

$$\mu_X = \frac{1}{N} \sum_{n=1}^N \|x_n\|, \sigma_n \stackrel{\text{def}}{=} \|\mu_X - \|x_n\|\| \quad (3)$$

$$\Sigma_X = \frac{1}{N} \sum_{n=1}^N \sigma_n^2 \quad (4)$$

式(5),(6)では分散の正規化を行う。まず σ_n を正規化した σ_n^{normal} を計算する。 $\sigma_{\max}, \sigma_{\min}$ は σ_n のうちそれぞれ最大、最小のものである。次に正規化した分散 Σ_X^{normal} を計算する。

$$\sigma_n^{\text{normal}} = \frac{\sigma_n - \sigma_{\min}}{\sigma_{\max} - \sigma_{\min}} \quad (5)$$

$$\Sigma_X^{\text{normal}} = \frac{1}{N} \sum_{n=1}^N (\sigma_n^{\text{normal}})^2 \quad (6)$$

実際の学習においては、 $\alpha(1 - \Sigma_X^{\text{normal}})$ を判別器の損失関数に加えた式(式(7))とする。 m はミニバッチサイズ、 d はデータのミニバッチ、 z はノイズのミニバッチである。 D, G はそれぞれ判別器、生成器を表す。

$$L_{\text{disc}} = -\frac{1}{m} \sum_{i=0}^m \left[\log D(d^{(i)}) + \log(1 - D(G(z^{(i)}))) \right] + \alpha(1 - \Sigma_X^{\text{normal}}) \quad (7)$$

3. 実験

我々は提案手法を導入したDCGANにイラストを学習させ、分散の変化を観察した。実験では生成モデルとしてDCGANを利用し、NVIDIA GeForce GTX 980Tiを積んだ計算機で学習を行った。

学習データとして検索エンジン*1から集めたイラストデータを使用する。3人のイラスト作家を定めてイラストを収集し、学習を簡単にするための前処理として右目領域を切り抜いた。

*1 <https://safebooru.org/>

それを $3 \times 96 \times 96$ のサイズに変形し、三つのデータセットを構築した (表 1). 学習の際には 9 割を訓練データとし、1 割をテストデータとして使用する. データの順序はそれぞれの学習ごとにランダムである.

表 1: イラストデータセット

データ名	枚数 [枚]	前処理後 [枚]	学習時間 [h]
kantoku	1492	1427	34
muririn	360	297	8
tsunako	1146	811	20

パラメータ α に関して分散の値などを観察する. $\alpha = 0.0, 0.1, 1.0, 10.0$ の 4 パターンを設定し, 訓練データを用いて DCGAN を 10000epoch 学習させた. α のパターン違いによる学習時間の変化はほとんど見られなかった.

4. 結果

提案手法による学習結果を観察し, パラメータ α に関して比較を行う. 図 1 は学習過程での分散の推移を表しており, 三つのグラフはデータセットそれぞれの結果を示す. グラフでは横軸が epoch, 縦軸が分散の値で, すなわち学習進度に対する分散の変化を表している. 実線は全 10000 点のうち 1000 点毎の平均の 10 点を結んだもので, 破線は全体平均である. 色の違いはパラメータ α の違いを表す. 図 2 には生成画像からランダムに抽出したサンプルを示す.

kantoku データではパラメータを変化しても, 結果はほぼ同じである. 1000 点平均では $\alpha = 1.0$ の場合に分散の上昇が早く発生しているが, 全体平均を見るとほぼ同じ分散値を取っていることがわかる. このデータの傾向として学習終盤に分散が上昇するという特徴が見られる.

muririn データではパラメータによる変化が見られる結果となった. 1000 点平均では $\alpha = 1.0$ の場合に最大値を取り, 全体平均でも最大となっている. このデータでは学習が進むほど分散が小さくなっていることがわかる.

tsunako データでもパラメータによる変化が見られる. 1000 点平均では最大値の点は取らないものの, $\alpha = 1.0$ の場合に分散の値が落ち込みにくくなっていることがわかる. 結果として全体平均では最大値となっている. このデータでも学習が進むにつれて分散が小さくなるが, $\alpha = 1.0$ では終盤において僅かに上昇していると確認できる.

今回のデータセットについては, 分散を導入した損失関数によって, collapse を回避し多様な画像を生成することができた. また, $\alpha = 1.0$ の場合が各データセットに共通して分散上昇に有効だと思われる.

5. まとめ

我々は GAN に対して分散を考慮した損失関数を提案し, イラストデータ学習において多様性を向上させる可能性があることを確認した. GAN における collapse という課題を解決するために, 生成画像の分散を組み込んだ学習を行った. 分散の値の確認によって, 提案手法の中でも最適なパラメータが存在する可能性がある. 最適なパラメータがわかれば, それによって分散を上昇させ, 多様性を持った生成画像を得られるのではないかと期待される.

課題として, 分散計算の適切さと実験回数の不十分さが考えられる. 今回の手法では生成画像のノルムとして単純な和を使用した, これは分散を適切に計測できているのか, という懸念がある. また今回の実験回数は, 提案手法の影響を分析するには不十分であると考えられる.

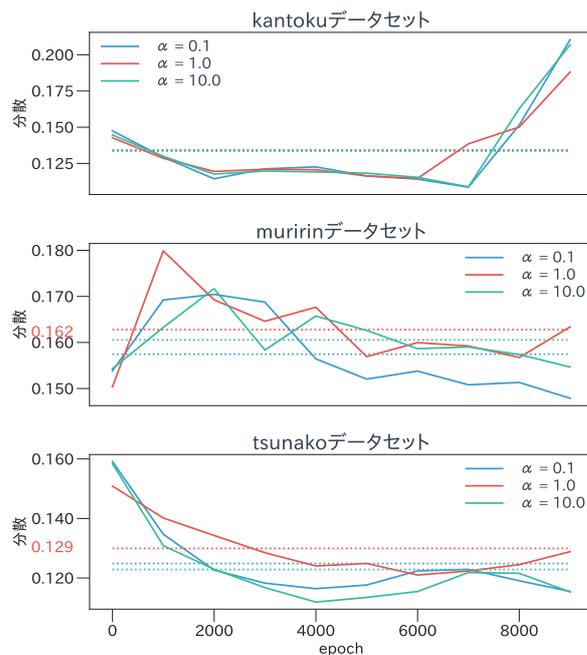
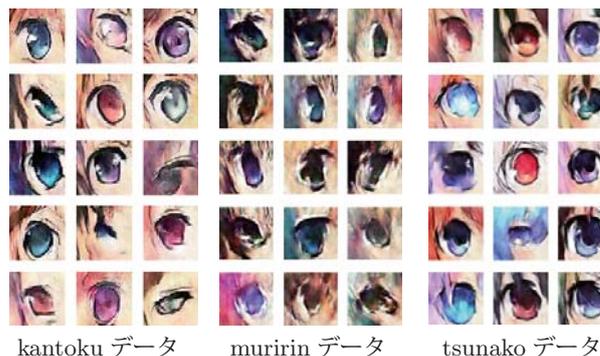


図 1: 分散の推移

図 2: 生成画像例 (6000epoch, $\alpha = 1.0$)

参考文献

- [Goodfellow 14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, Yoshua Bengio. Generative Adversarial Networks. Advances in Neural Information Processing Systems 27 (NIPS 2014), pp.2672–2680. 2014.
- [Radford 15] Alec Radford, Luke Metz, Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. ICLR 2016.
- [Salimans 16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen. Improved Techniques for Training GANs. 2016.
- [Arora 17] Sanjeev Arora, Yi Zhang. Do GANs actually learn the distribution? An empirical study. 2017.