

Self-Attention GAN における調和学習の提案

日大生産工 ○佐々木 創太 日大生産工 山内 ゆかり

1. まえがき

Generative Adversarial Networkの世界では安定した動作で精巧な画像を作成することが必要とされている。Ian Goodfellow氏らはSelf-Attention GANを提案し、GANの安定化手法として有効であったSpectral NormalizationをDiscriminator, Generator共に適応させ更にSelf-attention機構とTwo Time-Scale Update Ruleを追加することでImageNetに対して高画質な画像生成に成功したと報告がされている。しかしDとGを1:1のアップデートで行うにはバッチサイズをかなり大きなサイズにしなければならず、またImageNet以外のケースでの報告はあったが他のケースでの報告は無く、別の失敗が起きる可能性も孕んでいるという問題がある。

本研究では、こういったGANにおける安定化の問題において、DiscriminatorとGeneratorのアップデート回数を可変にすることを提案する。

2. 従来研究

2.1 Generative Adversarial Network

Generative Adversarial Networkとは、入力画像の真贋を判定するDiscriminatorと教師データに似せた入力を作成するGeneratorの二つからなるデータ作成のアルゴリズム。

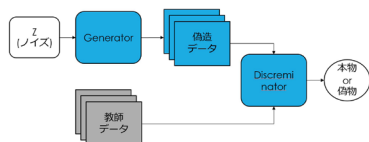


Figure 1 GAN のアルゴリズム

2.2 Self-Attention GAN

従来研究のSelf-Attention機構について、畳み込みの出力結果として得られたInputの画像群に対して1*1convolution等を行いQuery, Key, Valueを作成する。それぞれのピクセルに値するベクトルを参照して今後は計算を行う。まず左上のQueryベクトルと他のすべてのピクセルのKeyベクトルとの内積を求めてSoftmaxを作成する。(1)

$$\beta_{j,i} = \frac{\exp(S_{ij})}{\sum_{i=1}^N \exp(S_{ij})}, f(x_i)^T g(x_i), \quad (1)$$

これを行うことで左上のピクセルを他のどのピクセルで特徴づけるべきかが決まる。そしてValueをこのSoftmaxの結果に基づいてすべてのピクセルに対し重みづけしながら足し合わせる。(2)

$$o_j = v \left(\sum_{i=1}^N \beta_{j,i} h(x_i) \right), \quad (2)$$

$$h(x_i) = W_h x_i, v(x_i) = W_v x_i,$$

ここまでの処理をすべてのピクセルに対して行ったものを、入力に足し合わせることでそれをoutputとし、この一連の流れをSelf-Attentionとする。(3)

$$y_j = \gamma o_i + x_i \quad (3)$$

このような動作を行うことによって何が特徴づけられるかということ、convolutionでは表現しづらいほどに離れてしまっていた画像の特徴同士を結び付ける際などに活躍する。

Two Time-Scale Update Ruleとは、強化学習で用いられるactor-critic学習の知見をGANに応用したものである。actor-critic学習におけるactorとcriticの関係がGANのGeneratorとDiscriminatorの関係によく似ている。actor-critic学習で用いられていたcriticをactorよりも早く学習させる事でナッシュ均衡に近づくという知見に基づき、GANにおいてもdiscriminatorをgeneratorより速く学習させるようにすれば、より学習が安定し、収束が早まることを示している。

3. 提案手法

TTURによりDiscriminatorとGeneratorの学習率を個別に操作すること、Self-Attention機構を追加することでD:Gを1:1のアップデートでも安定化すると報告されているが、学習を続けていく際にDiscriminatorが優位すぎる状況に陥ってしまったりGeneratorの学習が優位に立ちすぎてしまったりすることが考えられる。その為の安定化手法でもあるTTURだが、ここにさらなる工夫を加えることで一度片側優位に

陥ってしまった状況などからでも安定させた学習が可能になるのではないだろうか。
実際にどのように行うかを次に示す。0~1 の範囲で出力がされる Discriminator のクラス分類結果を Batch Size で平均化する。Generator の入力で得られたクラス分類平均が 0.65 を超えた際には Generator 優位とし、Generator の学習回数を減らすことで互いの学習量の調和を図る。また教師データを入力として得られたクラス分類平均が 0.65 を下回ってしまっている場合でも Generator 優位と見做すことで、学習量の調整を行い調和を図る。また、Discriminator 優位に関しては教師データを入力として得られたクラス分類平均から、Generator を入力として得られたクラス分類平均で引くことで差を求めその差の数値が 0.75 より大きくなってしまった際に Discriminator 優位とし、Generator の学習回数を増やすことで調和を図る。ここでは Discriminator になるべく優位にある状況で安定させることで、学習速度を下落させることなくさらなる安定化を目指す。

4. 実験および検討

Optidigit及びMNISTデータセットを用いて手書き文字の判別及び生成を行う。評価方法としては損失関数の動きを確認し安定化しているかどうかを確認する。また、実際に得られた出力画像を目視での確認を行い正常に画像が出力されているか否かの判別も行う。

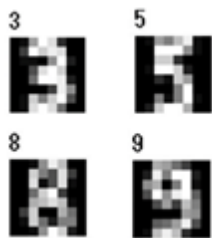


Figure 2 入力画像の例



Figure 3 出力画像の例

5. まとめ

安定した学習の為にDiscriminatorの出力値を参照した調和学習を提案し、計算機実験により有効性を検討した。

参考文献

- 1) Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative Adversarial Nets",
- 2) Han Zhang, Ian Goodfellow, Dimitris Metaxas, Augustus Odena, "Self-Attention Generative Adversarial Networks"