1 はじめに

GAN の最適解はナッシュ均衡として知られている．ここでいうナッシュ均衡とは，Discriminator の Loss が DIscriminator のパラメータに関して最小であり，かつ Generator の Loss も Generator のパラメータに関して最小な点の事である．従来の GAN は，アルゴリズムが収束するという保証が無い Gradient descent を使用していたためナッシュ均衡を得ることができなかった．しかし，GAN のようなコスト関数が非凸であり，パラメータが連続値かつ非常に高次元な場合で，ナッシュ均衡を求めるアルゴリズムは未だ知られていない．

Feature matching, Minibatch Discrimination はそんな GAN に対して，収束を促進させるようにヒューリスティックに動機付けられた手法を提案している．つまり，近似的に解を得ようとする発想である．

2 Feature Matching

feature matching は現在の Discriminator に関しての overtraining を防ぐ新しい目的関数を作ることで，GAN の不安定さに対処する．具体的には，Generator が Dicriminator の中間層の特徴量の期待値にマッチするように学習させる手法であり，以下のように Generator の目的関数を定義する．

$$\min_G \|\mathbb{E}_{\boldsymbol{x}\sim p_{data}}\boldsymbol{f}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boldsymbol{f}(G(\boldsymbol{z}))\|_2^2 \tag{1}$$

ただし，$f(\boldsymbol{x})$ は Discriminator の中間層の出力を表す．

Discriminator はトレーニングによって，本物か生成されたものかを最も判別可能な特徴量を見つけようとするので，この手法は自然な選択である．オリジナルの GAN の Loss ではデータ分布と全く同じになる最適解が存在したが，feature matching は Generator の損失関数に手を加えるため，この最適解に達するかの保証はない．しかし，オリジナルでは不安定だったシチュエーションにおいて実験的に feature matching は効果的であったと言っている．

疑問点

- feature matching を使うときの activation とは活性化関数をかけた後のことを言っているのか．
- feature matching を適用するときはどの層の出力後に適用すべきか．出力層前の Conv Layer に適用するのが一般的?
- feature matching の Loss に係数 λ をかけ，通常の Generator の Loss に加えるとどうなるのか?

$$\min_G \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}} \log D\left(G(\boldsymbol{z}_i)\right) + \lambda\|\mathbb{E}_{\boldsymbol{x}\sim p_{data}}\boldsymbol{f}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boldsymbol{f}(G(\boldsymbol{z}))\|_2^2$$

Algorithm 1 に，Feature matching の Generator の最適化アルゴリズムを示す．

Algorithm 1 Feature Matching

Require: $\boldsymbol{x}_n, n \in [N]$: ミニバッチデータセット, $\boldsymbol{z}_n, n \in [N]$: コード

Ensure: $\boldsymbol{x}_n \in \mathbb{R}^D, \boldsymbol{z}_n \in \mathbb{R}^{100}$

1: $f(\boldsymbol{x})$:本物データを Discminator に入れ，適当な中間層 (出力層の一つ手前など) の出力を得る

2: $f\left(G(\boldsymbol{z})\right)$: コードを Discminator に入れ，適当な中間層 (出力層の一つ手前など) の出力を得る

3: 次の関数を計算し，最小化するように誤差逆伝播を行い最適化する．

$$\left\| \frac{1}{N} \sum_n \boldsymbol{f}(\boldsymbol{x_n}) - \frac{1}{N} \sum_n \boldsymbol{f}(G(\boldsymbol{z_n})) \right\|_2^2$$

3 Minibatch Discrimination

GAN の主な失敗の 1 つに，Generator が常に同じ出力をだす mode collapse がある．これは Discriinator が各データを独立に処理し，Generator の出力が各データとより異なるようになれというメカニズムがないために引きおこされる問題である．

このタイプの失敗を避けるのに適した方法は，Discriminator に複数のデータを組み合わせて見せることを可能にさせ，以下の minibatch discrimination を実行することである．

複数のデータを個別ではなく組み合わせてみる Discriminator は，Generator の mode collapse を避けるのを助けることができるのかもしれない．実際，DCGAN の論文 (by Radford) であった，batch normalization の応用の成功はこの観点から，うまく説明することができる．ミニバッチ内のデータの近さをモデリングする方法の 1 つは以下の通りである．

ミニバッチを $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ としたとき，ミニバッチの n 番目のサンプル $\boldsymbol{x}_n$ を入れた時の，Discriminator のある中間層の特徴ベクトルを $\boldsymbol{f}(\boldsymbol{x_n}) \in \mathbb{R}^A$ とする．そして，$\boldsymbol{f}(\boldsymbol{x_n})$ にテンソル $T \in \mathbb{R}^{A \times B \times C}$ をかけ，$M_n \in \mathbb{R}^{B \times C}$ を得る．M_n を c 次元ん行ベクトルが B 個並んだものと考え，行ごとに $n \in [N]$ との L_1-distance をとる．さらに，負の指数をとり，以下を得る．

$$c_b(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\|M_{i,b} - M_{j,b}\|_{L_1}\right) \tag{2}$$

ここで，$b \in [B], i, j \in [N]$ である．minibatch discrimination layer の $\boldsymbol{x}_n$ についての出力 $o(x_n)$ は他のミニバッチ内のデータ全てとの和で定義される．

$$o(\boldsymbol{x}_n)_b = \sum_{j=1}^{N} c_b(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbb{R} \tag{3}$$

$$o(\boldsymbol{x}_n) = [o(\boldsymbol{x}_n)_1, o(\boldsymbol{x}_n)_2, \ldots, o(\boldsymbol{x_n})_B] \in \mathbb{R}^B \tag{4}$$

$$o(\boldsymbol{X}) = \begin{pmatrix} o(\boldsymbol{x}_1) \\ o(\boldsymbol{x}_2) \\ \vdots \\ o(\boldsymbol{x}_N) \end{pmatrix} \in \mathbb{R}^{N \times B} \tag{5}$$

最後に，minibatch discrimination layer の出力 $o(\boldsymbol{x_n})$ と中間層の特徴ベクトル $\boldsymbol{f}(\boldsymbol{x_n})$ とを結合し，結果を

Algorithm 2 Minibatch Discrimination

Require: $\boldsymbol{x}_n, n \in [N]$: ミニバッチデータセット, $\boldsymbol{z}_n, n \in [N]$: コード

Ensure: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$: ミニバッチ, $\boldsymbol{x}_n \in \mathbb{R}^D, \boldsymbol{z}_n \in \mathbb{R}^{100}$

1: $\boldsymbol{f}(\boldsymbol{X}) \in \mathbb{R}^{N \times A}$:ミニバッチデータを Discminator に入れ，適当な中間層 (出力層の一つ手前など) の出力を得る

2: $\boldsymbol{f}(\boldsymbol{X})$ の個々のデータ $\boldsymbol{f}(\boldsymbol{x}_n)$ について以下のように $o(\boldsymbol{x}_n)$ を計算する

$$M = \boldsymbol{f}(\boldsymbol{x}_n) \cdot T \in \mathbb{R}^{B \times C}$$
$$c_b(\boldsymbol{x}_n, \boldsymbol{x}_k) = \exp(-\|M_{n,b} - M_{k,b}\|) \in \mathbb{R}$$
$$o(\boldsymbol{x}_n)_b = \sum_{k=1}^{N} c_b(\boldsymbol{x}_n, \boldsymbol{x}_k) \in \mathbb{R}$$
$$o(\boldsymbol{x}_n) = \left[o(\boldsymbol{x_n})_1, o(\boldsymbol{x_n})_2, \ldots, o(\boldsymbol{x_n})_B\right] \in \mathbb{R}^B$$

3: そして，$o(\boldsymbol{x}_n)$ を行ベクトルとみなし，ミニバッチ全てについての結果 $o(\boldsymbol{X})$ を以下のようにする．

$$o(\boldsymbol{X}) = \begin{pmatrix} o(\boldsymbol{x}_1) \\ o(\boldsymbol{x}_2) \\ \vdots \\ o(\boldsymbol{x}_N) \end{pmatrix} \in \mathbb{R}^{N \times B}$$

4: 最後に，$o(\boldsymbol{X})$ を元の活性 $f(\boldsymbol{X})$ と結合させ，次の層に渡す

次の層の入力として与える．この計算を Generator からのサンプルのミニバッチ内と学習データからのサンプルのミニバッチ内でそれぞれ別々に行う．従来のように，Discriinator は各データに対して，そのデータが本物である確率を表す数字を出力するようにする．Discriminator のタスクは実質的に，そのデータが本物かどうかを識別することである．しかし，今ではサイド情報として，ミニバッチ内の他のサンプルを使うことができる．Minibatch Discrimination は視覚的に魅力的なサンプルを素早く生成させることを可能にさせる．Algorithm2. に Minibatch Discrimination のアルゴリズムを示す．

4 Inception Score

GAN はモデル同士を比べる評価関数が十分でない．最も直感的で理解しやすい指標は，Annotator にサンプルの視覚的なクオリティを測ってもらうことである Tim Salimans, et al. による論文 [**?**] では，GAN のモデル比較のために，はじめは Amazon Mechanical Turk を用いて人に生成したデータが本物かどうかを識別してもらい，その正解率を指標としていた．しかし，人による識別では問題設定，Annotator のモチーベーションに左右されてしまう．そこで Inception score を提案した．まず条件付き分布 $p(y|x)$ を得るために全ての生成画像に Inception Model を適用する．y はあるラベルを表す．論文では TensorFlow の事前学習済みのモデルを使用した[*1]．

意味のある画像ならば条件付き分布 $p(y|\boldsymbol{x})$ は，高い確率を持って識別されるため，低いエントロピーを持つ

[*1] http://download.tensorflow.org/models/ image/imagenet/inception-2015-12-05.tgz.

はずである．また，私たちはモデルが多様な画像を生成することを期待しているため，$\int_{\boldsymbol{z}\sim p_{\boldsymbol{z}}} p(y|\boldsymbol{x}=G(\boldsymbol{z}))dz$ は低いエントロピーを持つはずである．以上２点を考慮すると，Inception score は以下のように定義される．

$$\exp\left(\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}} KL\left(p(y|\boldsymbol{x})||p(y)\right)\right) \tag{6}$$

この Inception score は人間の判断と強い相関をももち，良い指標と言うことがわかった．また，この基準は多様性を測るので十分に大きいデータ数 (50000) で評価することが重要であるということもわかった．

参考文献

[1] Tim Salimans, et al., Improved Techniqes for Training GANs
https://arxiv.org/abs/1606.03498