GANとは

5/18/2018

sakai

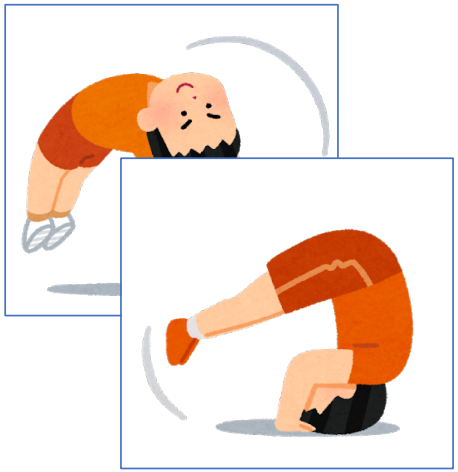
目次

生成モデル

生成モデル

観測データ x

観測データがあるモデルから生成されたと仮定する



学習

生成

$p(x;\theta)$

生成モデル

最近の流行りはこの生成モデルを
NNによって表現すること.

$$z \rightarrow \boxed{\text{生成器}} \rightarrow x$$

$$x = f(z)$$

生成モデルの特徴

- サンプリングが可能.
  - 確率モデルから未知のデータの生成.
- データに対する確率密度が得られる.
- 半教師あり学習に応用ができる.

VAEのぼやけた画像

Input

VAE reconstruction

- 尤度の評価のため単純な分布(ガウス，ベルヌーイ)を仮定していることが原因(多分).

GAN

尤度関数を明示しない生成モデル

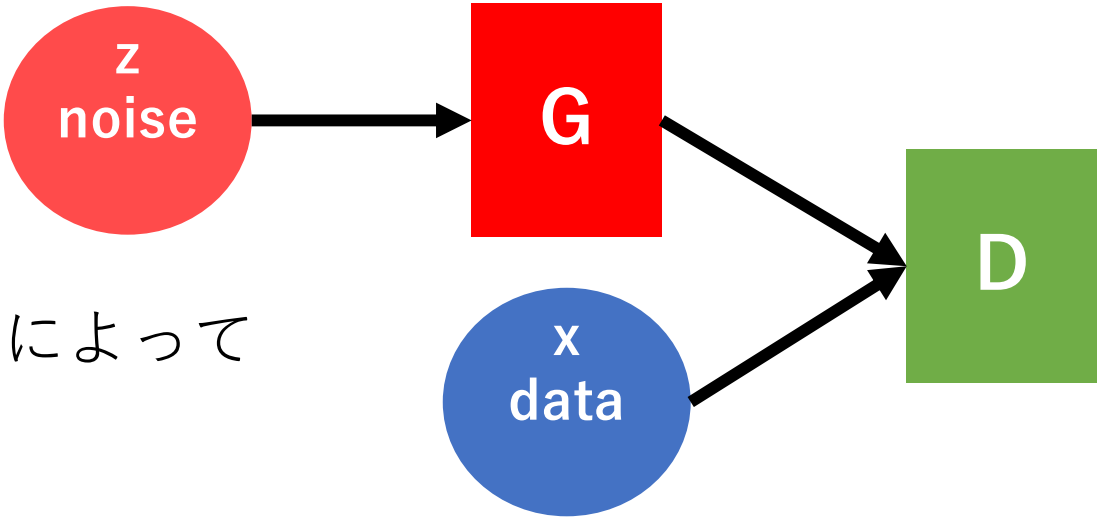
パラメトリックな分布を陽に定義する.

- パラメータをNNで得る.
- 尤度最大化でデータ分布にモデル分布を近づける.

分布を仮定せず
分布の形そのものをNNで表現しよう.

Generative Adversarial Netsの概要

- 訓練データ $x \sim p_{data}(x)$ を表現する生成器Gを得ることが目的.
- つまり生成器Gの出力の分布 $p_g(x)$を $p_{data}(x)$ に近づけることが目的である.



敵対する二つのNNによって
訓練を行うモデル.

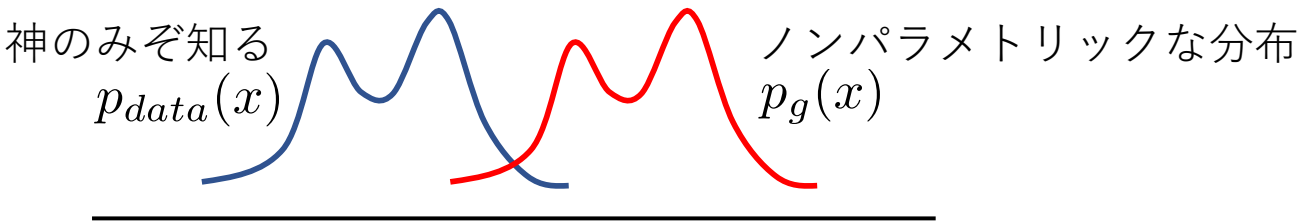
ただし $p_g(x)$ は未定義のままで.

密度比から分布の比較

- 未定義の分布に対して尤度は測れない

目標 データの分布 $p_{data}(x)$ に近いモデル分布 $p_g(x)$ を求める.

モデル分布: 生成器Gの出力が従う分布



- 分布の比較ができればいいので密度比を考える.

$$r(x) = \frac{p_{data}(x)}{p_g(x)}$$

このままではまだカーネル密度推定などで求める必要がある.

密度比からクラス分類問題へ

- データの分布に従うデータと生成器から生成したデータを同じ割合で用意し，ラベル付けされたデータ集合を考える．
- データの分布に従うデータをy=1, 生成器から生成されたデータをy=0とした．

$$\mathcal{D} = \{(x_1, 0), \ldots, (x_N, 0), (x_{N+1}, 1), \ldots, (x_{2N}, 1)\}$$

$$\text{ただし，}\ p(y=1) = p(y=0) = \frac{1}{2}$$

このとき，それぞれの分布は

$p_{data}(x) = p(x|y=1)$ ：データの従う分布

$p_g(x) = p(x|y=0)$ ：生成器の出力値が従う分布

密度比からクラス分類問題へ

密度比は以下のようになる.

$$\frac{p_{data}(x)}{p_g(x)} = \frac{p(x|y=1)}{p(x|y=0)} = \frac{\frac{p(y=1|x)p(x)}{p(y=1)}}{\frac{p(y=0|x)p(x)}{p(y=0)}} = \frac{p(y=1|x)}{p(y=0|x)}.$$

$p(y=1|x)$ を学習で推定したい.
NNで上記を推定する分布をパラメータ化する.

識別器
discriminator

$$x \rightarrow \boxed{D(x;\theta_d)} \rightarrow y$$

$$D(x;\theta_d) = p(y=1|x)$$

密度推定を二値分類問題に置き換えた.

ベルヌーイ分布の対数尤度最大化

$$p(y|x) = \mathrm{Bern}(y|D(x;\theta_d))$$

- 識別器のパラメータθ_dを決定する.

尤度：
$$p(\mathbf{y}|D(\mathbf{x};\theta_d)) = \prod_{i=1}^{2N} D(x_i;\theta_d)^{y_i}(1-D(x_i;\theta_d))^{1-y_i}$$

$$where \quad \mathbf{y} = (y_1,\ldots,y_{2N}) \quad \mathbf{x} = (x_1,\ldots,x_{2N})$$

- 次の負の交差エントロピーを最大化.

$$V(D) = \mathbb{E}_{p(x,y)}\left[y\log(D(x;\theta_d)) + (1-y)\log(1-D(x;\theta_d))\right]$$

識別器の目的関数

- 式変形.

$$
\begin{aligned}
V(D) &= \mathbb{E}_{p(x|y)p(y)}\left[y\log(D(x;\theta_d)) + (1-y)\log(1-D(x;\theta_d))\right] \\
&= \mathbb{E}_{p(x|y=1)p(y=1)}\left[\log(D(x;\theta_d))\right] + \mathbb{E}_{p(x|y=0)p(y=0)}\left[\log(1-D(x;\theta_d))\right] \\
&= \frac{1}{2}\mathbb{E}_{p_{data}(x)}\left[\log(D(x;\theta_d))\right] + \frac{1}{2}\mathbb{E}_{p_g(x)}\left[\log(1-D(x;\theta_d))\right]
\end{aligned}
$$

- 定数倍は無視して,

$$
V(D) = \mathbb{E}_{p_{data}(x)}\left[\log(D(x;\theta_d))\right] + \mathbb{E}_{p_g(x)}\left[\log(1-D(x;\theta_d))\right]
$$

目的関数の意味

目標 データの分布 $p_{data}(x)$ に近いモデル分布 $p_g(x)$ を求める.

識別関数が適切に推定できたなら

$$D^*(x;\theta_d^*) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)} \quad \text{となる(証明略).}$$

このとき，目的関数は

$$V(D^*) = 2\cdot JSD(p_{data}||p_g) - \log 4 \quad \text{となる(証明略).}$$

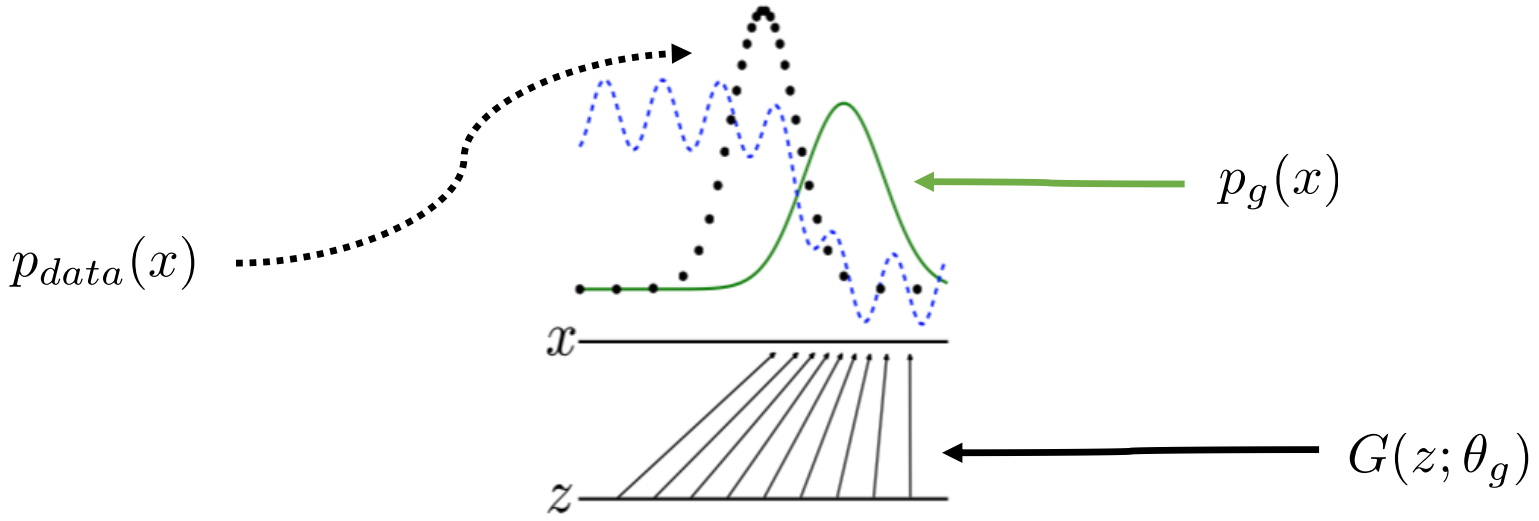
- JSDはJensen-Shannon divergence.
- Kullback-Leiber divergence と違い対称性がある.
- 識別器Dによって生成器Gのための評価指標(JSD)を用意した.

生成器

$p_{data}(x)$ を推定する分布をNNでパラメータ化する.

生成器
generator
$G(z;\theta_d) = x$

$z \rightarrow \boxed{G(z;\theta_g)} \rightarrow x$



生成器の目的関数

目標 データの分布 $p_{data}(x)$ に近いモデル分布 $p_g(x)$ を求める.

- $JSD(p_{data}||p_g)$ を最小にする生成器のパラメータを求める.

$$
\begin{aligned}
JSD(p_{data}||p_g) &= \frac{1}{2}\left\{KL\left(p_{data}||\frac{p_{data}+p_g}{2}\right)+KL\left(p_g||\frac{p_{data}+p_g}{2}\right)\right\} \\
&= \mathbb{E}_{x\sim p_{data}}\left[\log\frac{2p_{data}(x)}{p_{data}(x)+p_g(x)}\right]+\mathbb{E}_{x\sim p_g}\left[\log\frac{2p_g(x)}{p_{data}(x)+p_g(x)}\right] \\
&= \mathbb{E}_{x\sim p_{data}}[\log D^*(x)]+\mathbb{E}_{x\sim p_g}[\log(1-D^*(x))]-\log 4 \\
&= \mathbb{E}_{x\sim p_{data}}[\log D^*(x)]+\mathbb{E}_{z\sim p_z}[\log(1-D^*(G(z)))]-\log 4
\end{aligned}
$$

生成器の目的関数

$$V(D,G) = \mathbb{E}_{p_{data}(x)}\left[\log(D(x;\theta_d))\right] + \mathbb{E}_{p_g(x)}\left[\log(1 - D(G(z;\theta_g);\theta_d))\right]$$

- 上記の式の最小化は以下の正答率の最小化とも解釈できる.

$$p(y = 1|\hat{x}) = D(\hat{x};\theta_d) \qquad \hat{x}\text{: 真の分布から得られたデータ}$$
$$p(y = 0|G(z;\theta_g)) = 1 - D(G(z;\theta_g);\theta_d)$$

- GはDによってもたらされる近似的なJSD距離の最小化によって学習する.

実際の学習アルゴリズム

実際には交互に目的関数を最適化する.

- Gを固定してDの学習.

$$\max_{\theta_d} \mathbb{E}_{p_{data}(x)} \left[\log(D(x;\theta_d))\right] + \mathbb{E}_{p_g(z)} \left[\log(1 - D(G(z;\theta_g);\theta_d))\right]$$

- Dを固定してGの学習.

$$\min_{\theta_g} \mathbb{E}_{p_g(z)} \left[\log(1 - D(G(z;\theta_g);\theta_d))\right]$$

実際の学習アルゴリズム

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

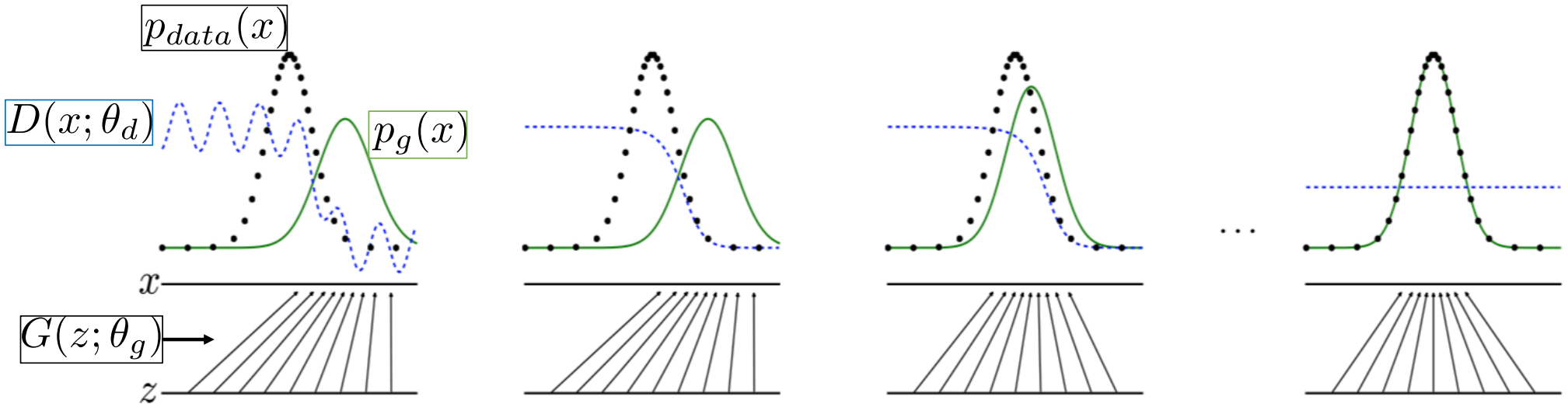
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

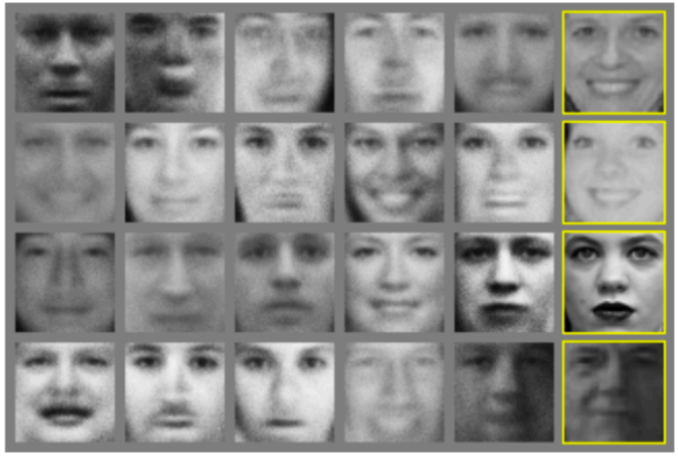
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

GANの学習の様子

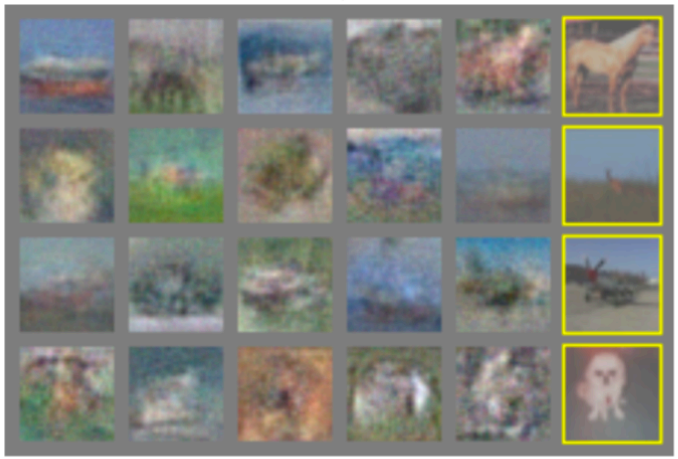


GANの生成物

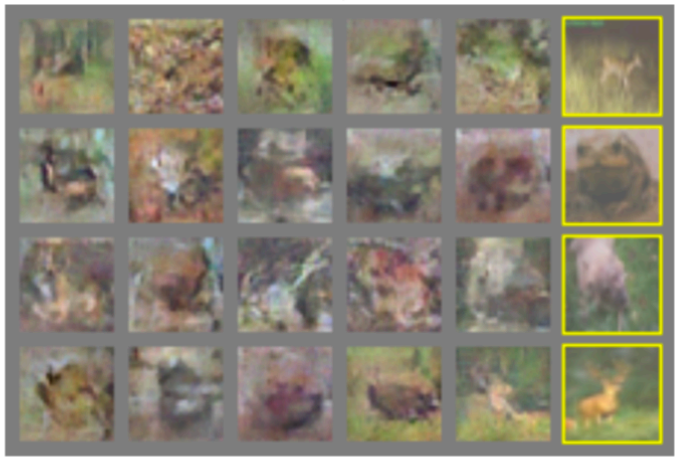
a)



b)



c)



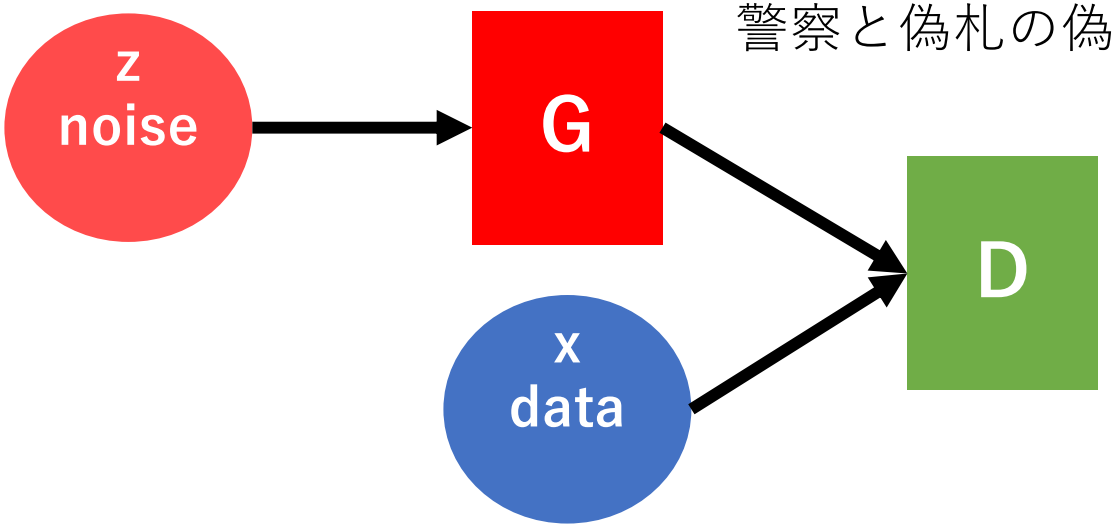
d)

概要再掲，参考

直感的には以下のミニマックスゲームをする．

- GはDを騙すxを生成する．
- DはGに騙されないよう識別する．

$$\min_{\theta_g} \max_{\theta_d} V(D,G)$$



警察と偽札の偽造者の攻防

参考
Mohamed, Shakir, and Balaji Lakshminarayanan. "Learning in implicit generative models." *arXiv preprint arXiv:1610.03483*(2016).

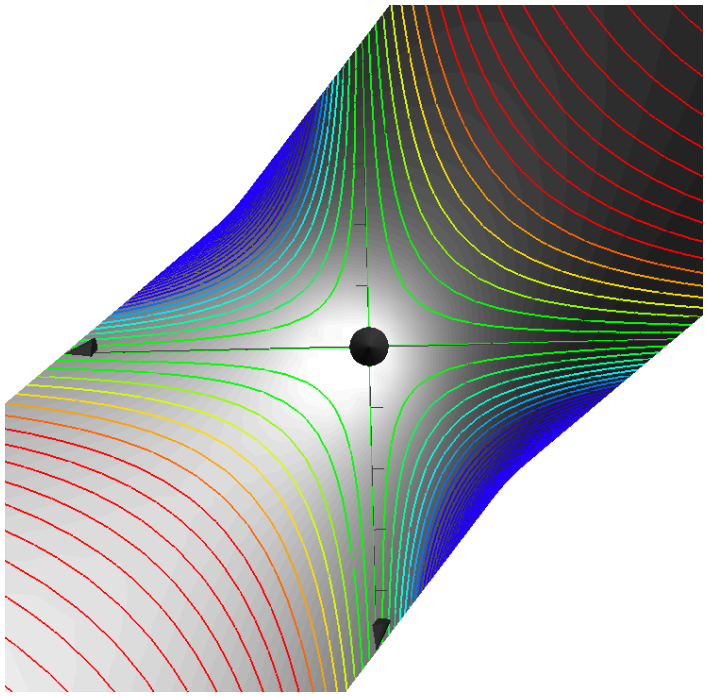
GANの欠点

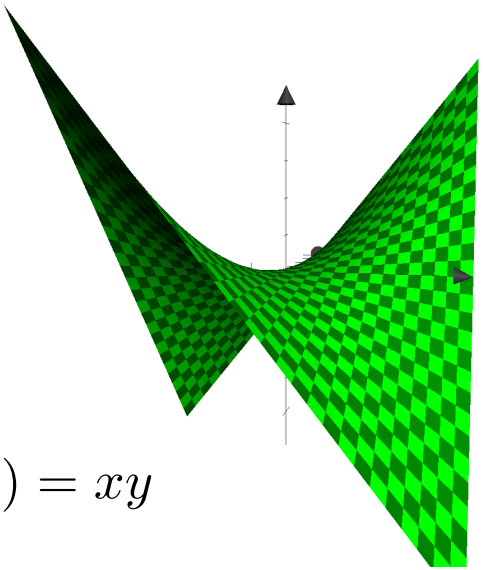
GANの欠点

- 収束性
- Mode collapse
- 勾配消失

収束性

- ミニマックスゲームの平衡点.
  - 例えば $V(x,y)=xy$ 価値関数では $x=y=0$ の鞍点が平衡点.
  - しかしxとyについて交互に片方を固定し最適化すると振動する.

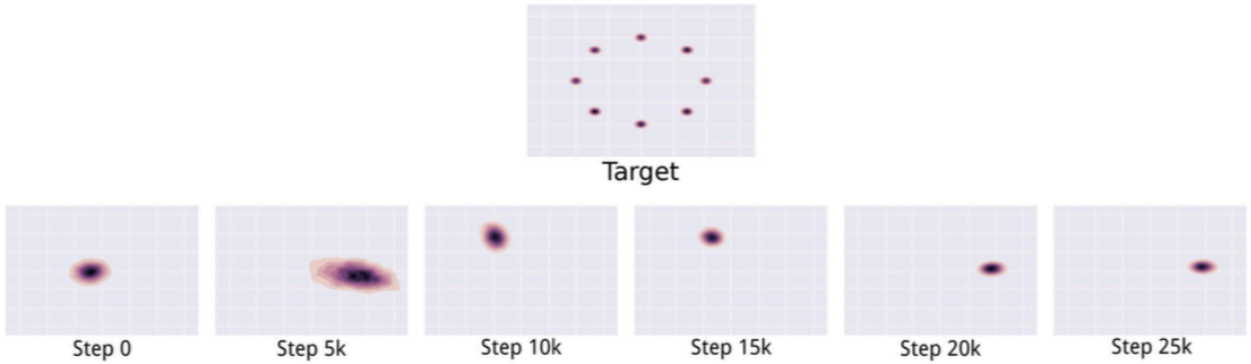


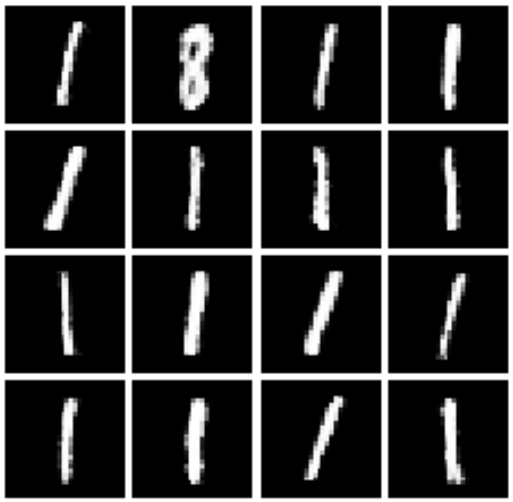


$V(x,y)=xy$

Mode collapse

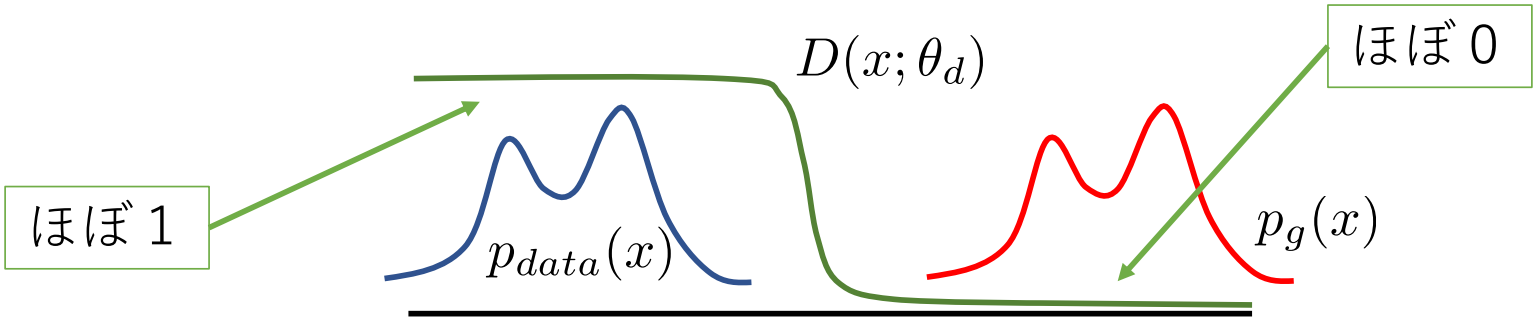
- 学習の不十分なDを固定してGを最適化すると…
  - Gの生成データが一つの峰にフィットする.





勾配消失

- 分布が交わらないとき(学習初期)，識別器が完全に識別できてしまう．



- 以下がサチってGについて更新ができなくなる．

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left[\log \left(1 - D \left(G \left(z^{(i)} \right) \right) \right) \right].$$

勾配消失の対処法

$$\min_G \mathbb{E}_{p_z}\left[\log\left(1 - D(G(z))\right)\right] \quad \Rightarrow \quad \min_G \mathbb{E}_{p_z}\left[-\log\left(D(G(z))\right)\right]$$

- Dがほぼ0を出力しても勾配がでる.
- もはやJSDとは関係がない.

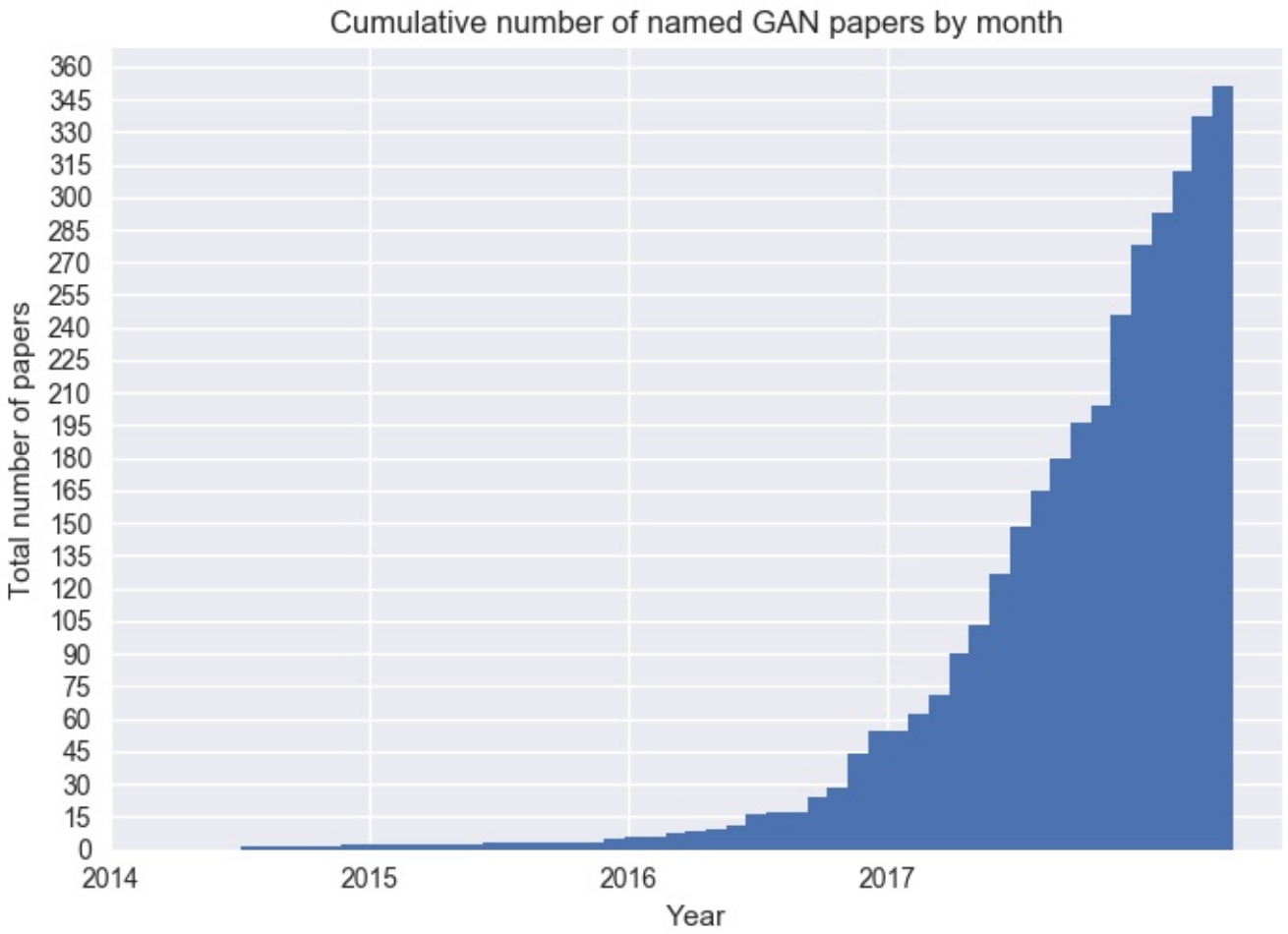
勾配消失

- Dを完全に最適化してしまうと
  - $p_{data}(x)$について１，　$p_g(x)$について０の定数を出力する.
  - 学習初期と同様Gについて勾配が消失する.

- GANの困りどころ
  - 学習が難しい.
  - Gが不完全なうちにDを最適化すると勾配が消失.
  - Dが不完全なうちにGを最適化するとmode collapse.

色々なGAN

爆発



略称は被りがあるので
注意

[引用]GAN zoo
(2018/5/16)

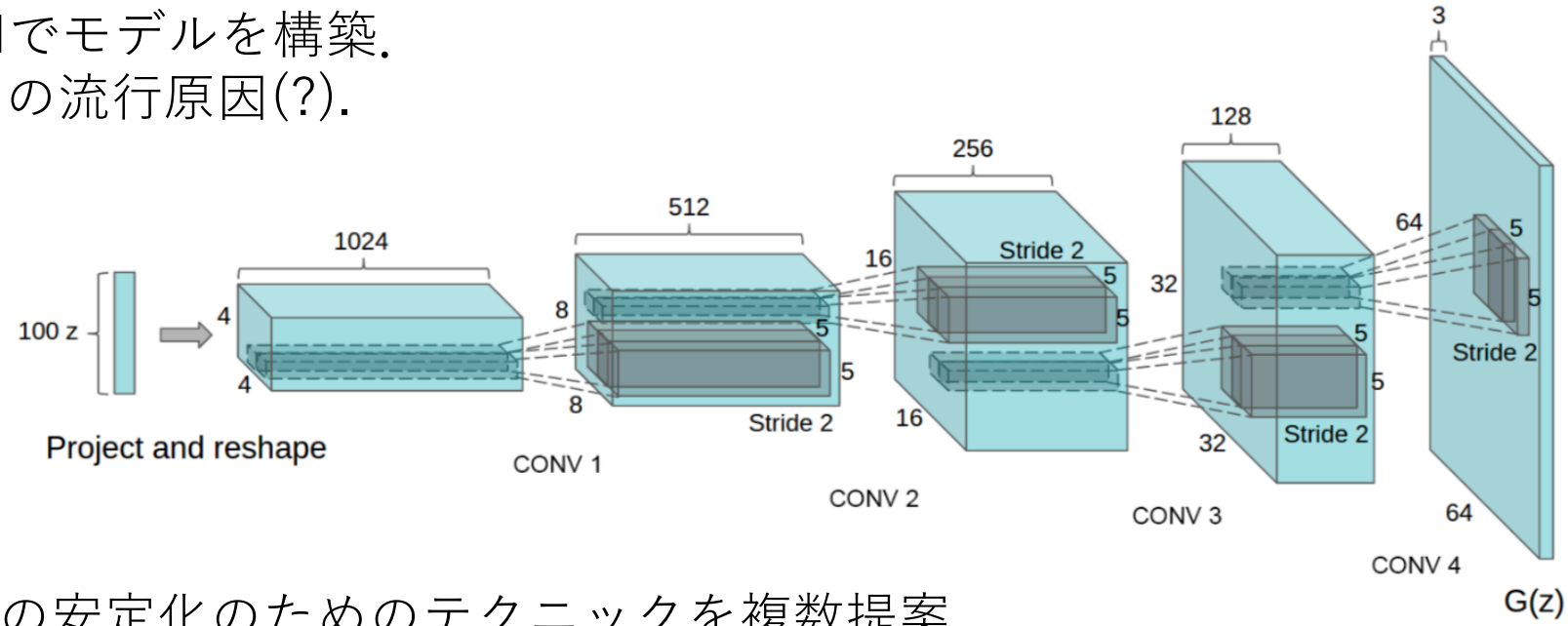
有名なGAN

(僕の中で)有名なGAN

- DCGAN(Deep Convolutional GAN)[2015]
- CGAN(Conditional GAN)[2014]
- Progressive GAN[2017]
- WGAN(Wasserstein GAN)[2017]

Deep Convolutional GAN

- CNNでモデルを構築.
- GANの流行原因(?).



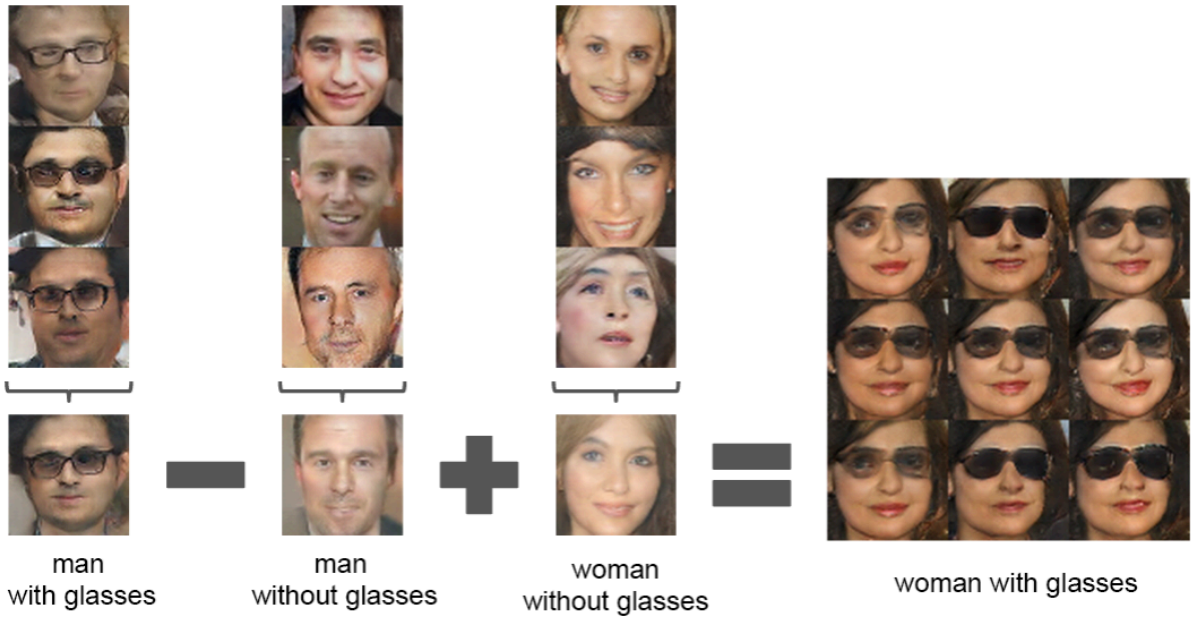
- 学習の安定化のためのテクニックを複数提案.
  - Batch Normalization.
  - Dにleaky ReLUを用いる.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

DCGANの生成物

潜在空間でのベクトル演算

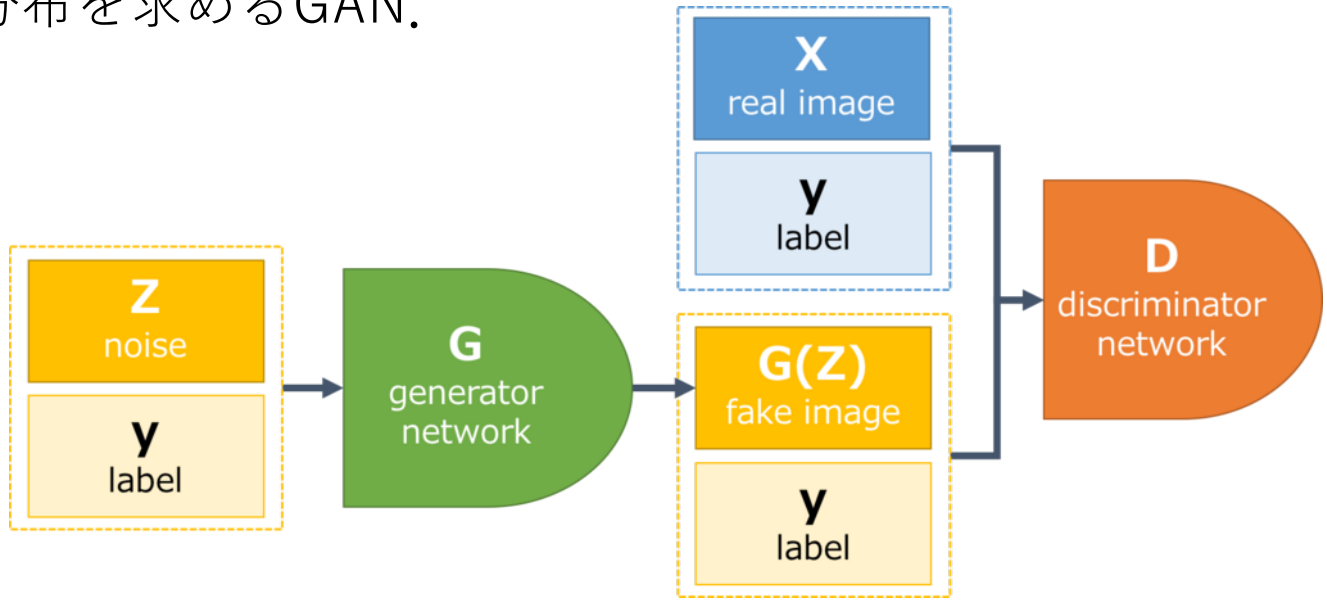
潜在空間での
演算



入力空間での
演算

Conditional GAN

- 条件付き分布を求めるGAN.

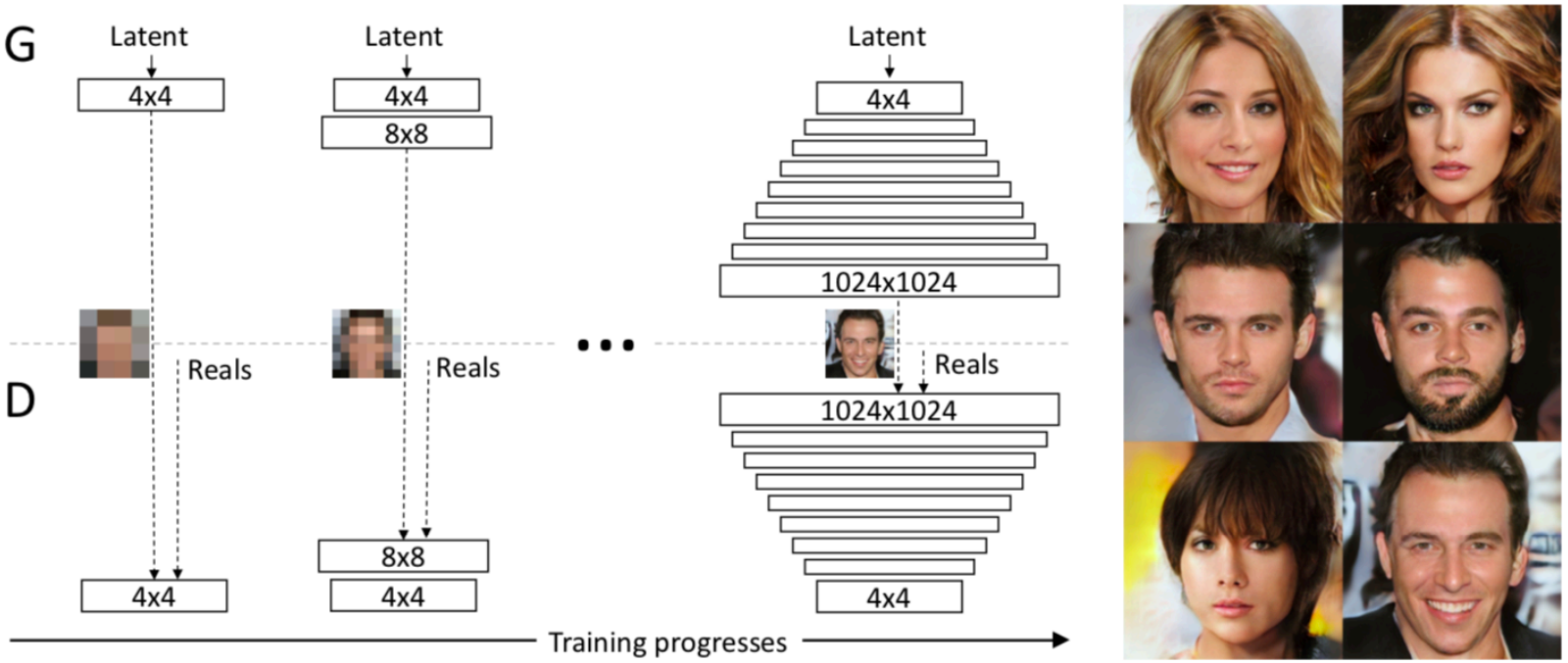


$$\min_G \max_D V(D,G) = \mathbb{E}_{p_{data}(x)}\left[\log(D(x|y))\right] + \mathbb{E}_{p_g(x)}\left[\log(1 - D(G(z|y)))\right]$$

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Progressive GAN

- 段階的に解像度を上げるように学習し，高解像度の画像生成に成功.

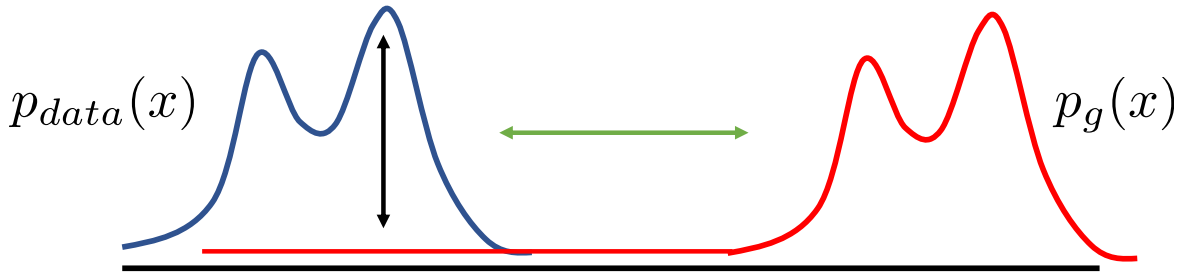


Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

JSDの限界

- 分布が交わらないとき，JSDは定数になる．

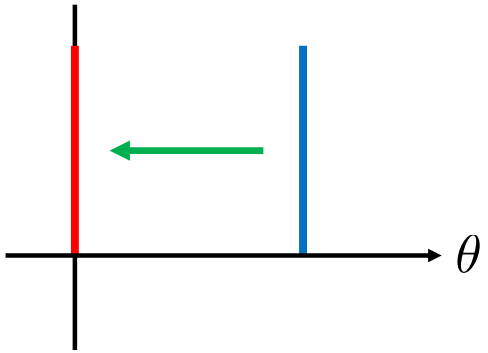
$$JSD(p_{data}||p_g) = \frac{1}{2}\left\{KL\left(p_{data}||\frac{p_{data}+p_g}{2}\right) + KL\left(p_g||\frac{p_{data}+p_g}{2}\right)\right\}$$
$$= \int p_{data}(x)\log\left\{\frac{2p_{data}(x)}{p_{data}(x)+p_g(x)}\right\}dx + \int p_g(x)\log\left\{\frac{2p_g(x)}{p_{data}(x)+p_g(x)}\right\}dx$$

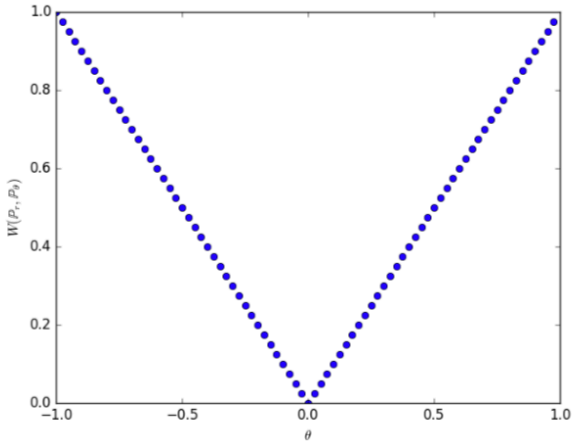


- 違うダイバージェンスを用いる提案が多くされている．
  - EMD(Earth Mover distance)

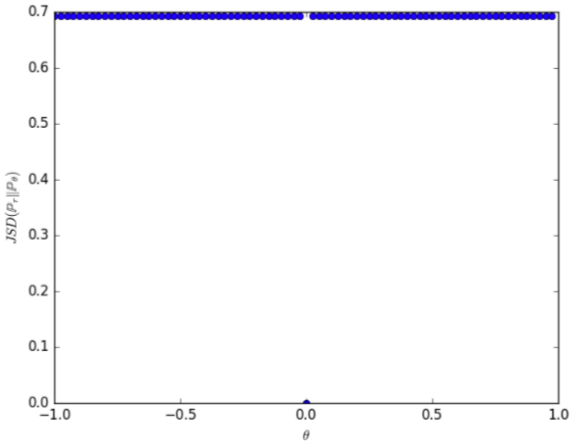
Wasserstein-1 距離 (EM distance) の利点

- 右の確率密度関数を動かした際の距離が以下.
- JSDでは広い範囲で定数，しかも不連続だが EMDは勾配が得られている.





EMD



JSD

Wasserstein GAN

- Wasserstein距離(の双対表現)による目的関数は以下.

$$W(\mathbb{P}_r, \mathbb{P}_g) = \max_w \mathbb{E}_{x\sim\mathbb{P}_r}\left[D(x;w)\right] - \mathbb{E}_{z\sim p(z)}\left[D(G(z);w)\right].$$

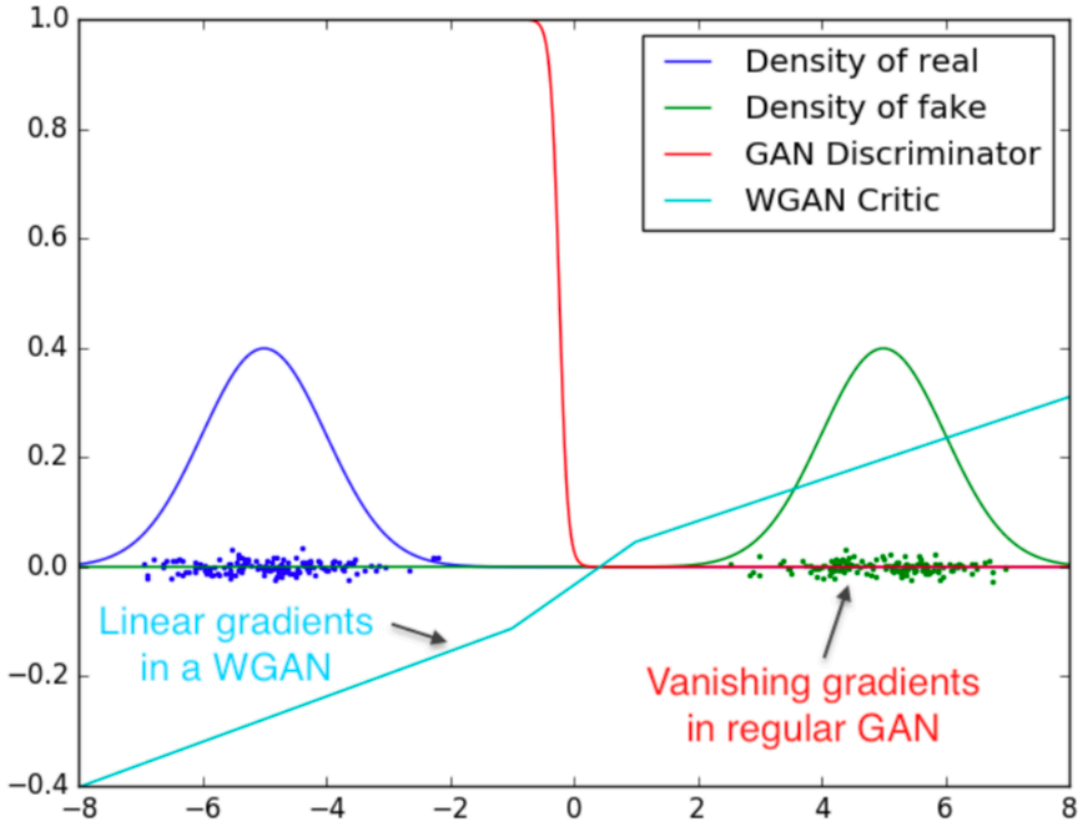
ただし　$\|D(x) - D(y)\| \leq \|x - y\|$　(1-リプシッツ連続)

- ほぼGANと同じ.
  - Dはリプシッツ性をもつ(傾きが急でない).
  - Dは非線型関数を出力に取らない(出力値は確率としての意味を持たない).
  - 付録.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Wasserstein GANの利点

- Critic(GANでのDの出力に相当)は全域で勾配を与えておりGの学習を助ける.
  - Criticは確率ではない.



終わり

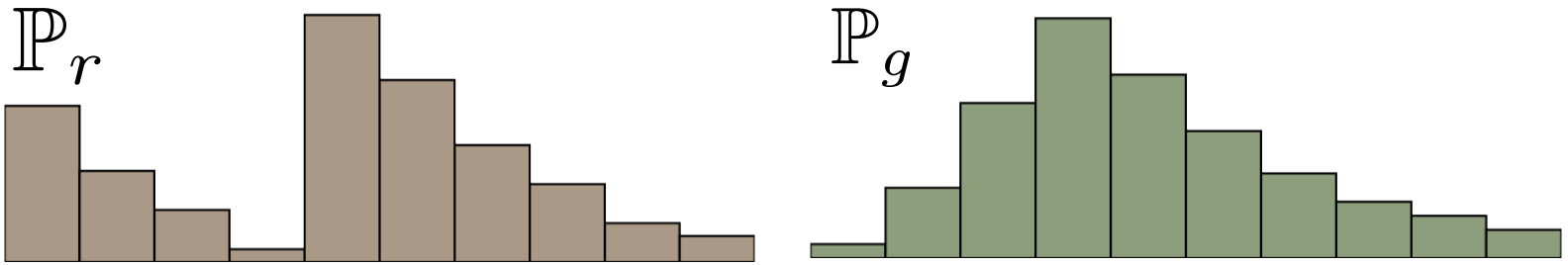
付録，参考文献

Wasserstein-1 距離　概要

- Earth-Mover distance or Wasserstein-1(以下EMD).

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \left[\|x - y\| \right].$$

- 距離とは言うものの，それ自体が最適化問題.
  - 確率密度分布を土の山としてみたとき，片方の確率分布の山をもう一方の確率分布の山に土を輸送して変形させるための最小の労力.



EMDの解釈

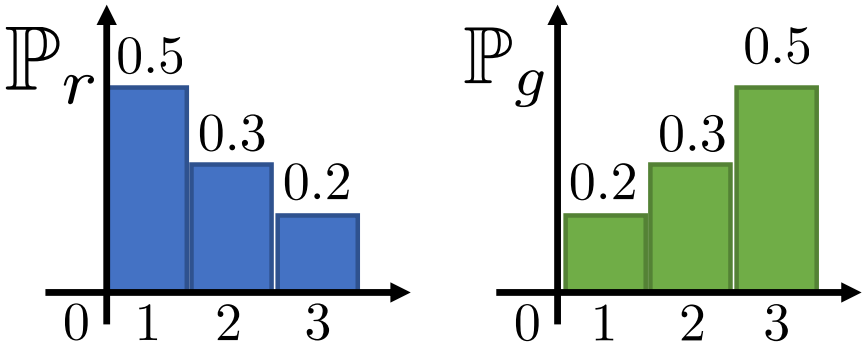
- 山を目的の形へ変形させる輸送方法は無限に存在する.
  - 輸送方法$\gamma(x,y)$を以下を満たすように定義する.

$$\sum_x \gamma(x,y) = \mathbb{P}_r(y) \qquad \sum_y \gamma(x,y) = \mathbb{P}_g(x)$$

- xとyについての上記を満たす同時分布として解釈できる.
- $\gamma(x,y)$ は $\mathbb{P}_g$ を $\mathbb{P}_r$ に変形するためにxからyへ輸送する土の量.

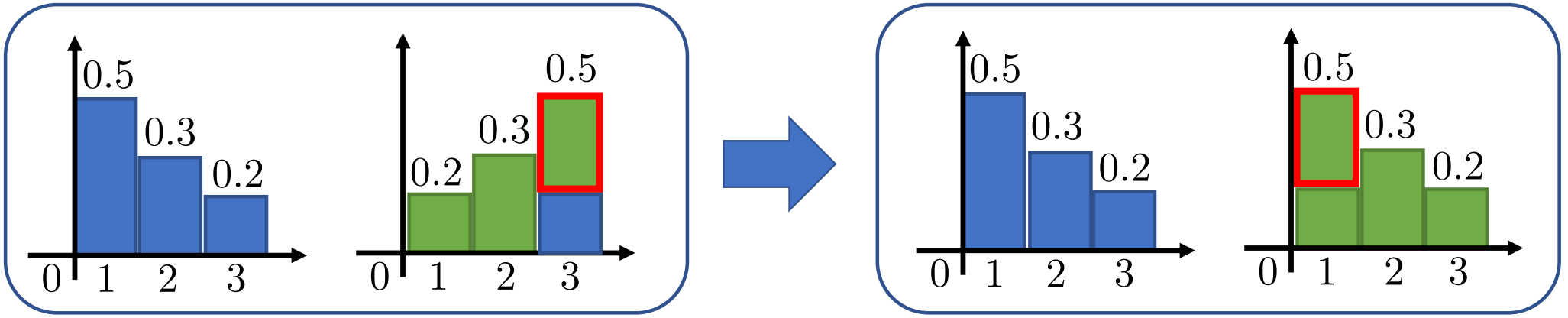
同時分布 γ の解釈

- 簡単のため右記の離散分布を考える.
- $\mathbb{P}_g$を変形させることで$\mathbb{P}_r$の形にする.



$\gamma(x, y)$ はxからyへ輸送される土の量.

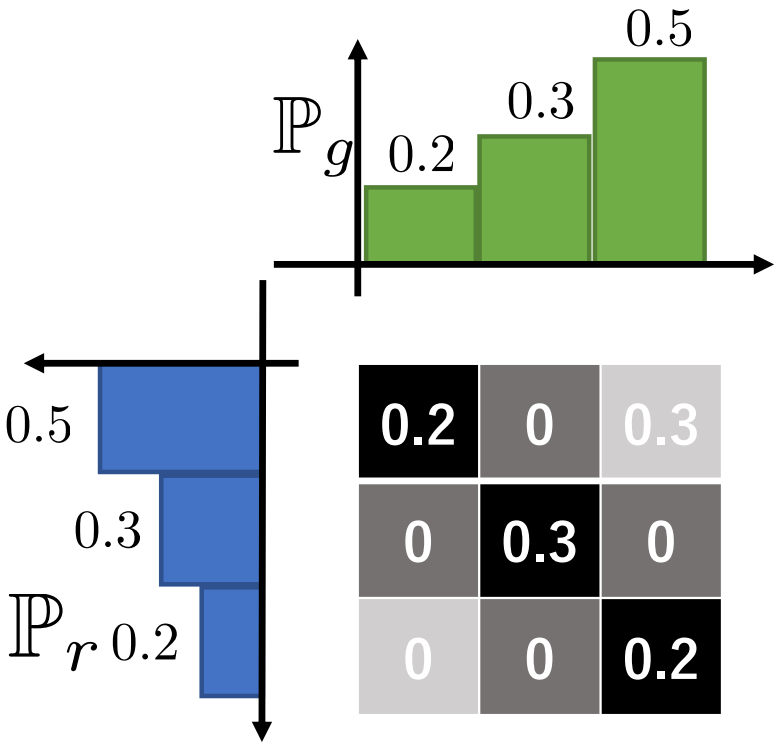
輸送方法として次の手順が考えられる.



3のビンの赤色の量を１のビンへ輸送することで形が一致する.
このとき$\gamma(x, y)$は→

同時分布 γ の解釈

- 先の輸送方法$\gamma(x, y)$は下の 2 次元ヒストグラム(表)で表現できる.



0.2	0	0.3
0	0.3	0
0	0	0.2

$$\sum_x \gamma(x, y) = \mathbb{P}_r(y) \qquad \sum_y \gamma(x, y) = \mathbb{P}_g(x)$$

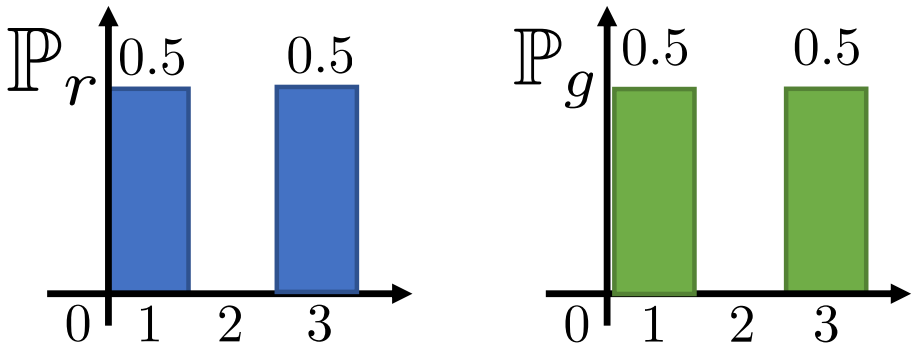
- もちろん上記の制約を満たす.
- 対角成分(x=y)は輸送しない量となる.
- 違う輸送方法については違う表になる.

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \left[\|x - y\|\right].$$

- 対角成分に大きい数値があるほどEMDは小さくなる.

理解のための極端な例

- 次の二つの同じ形の分布を考える.
  - 最適な輸送方法は”輸送しないこと”であることは自明.



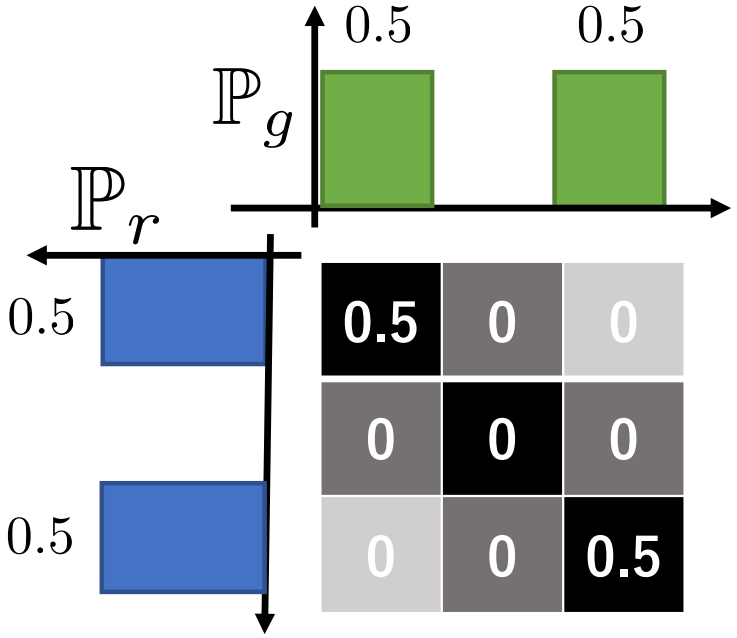
- ただ，1のビンと3のビンを入れ替えるという無駄な輸送方法を考えられる.

最適な輸送方法と無駄な輸送方法を比較してEMDがどうなるか確認する.

※(EMDは最適な輸送方法での距離なので無駄な輸送方法を用いた値はEMDではないことに注意)

理解のための極端な例

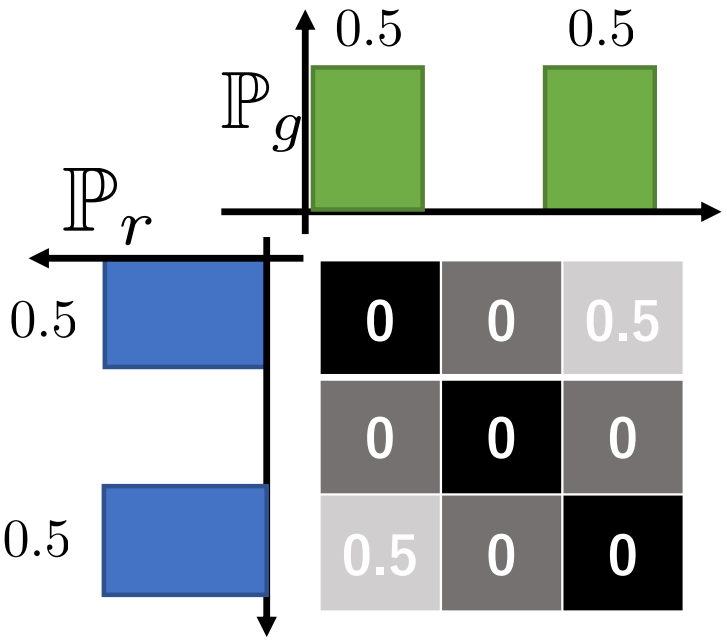
最適



$$\sum_{(x,y)} \left\{ \|x-y\| \gamma(x,y) \right\} = 0$$

こっちがEMDとなる.

無駄な輸送方法



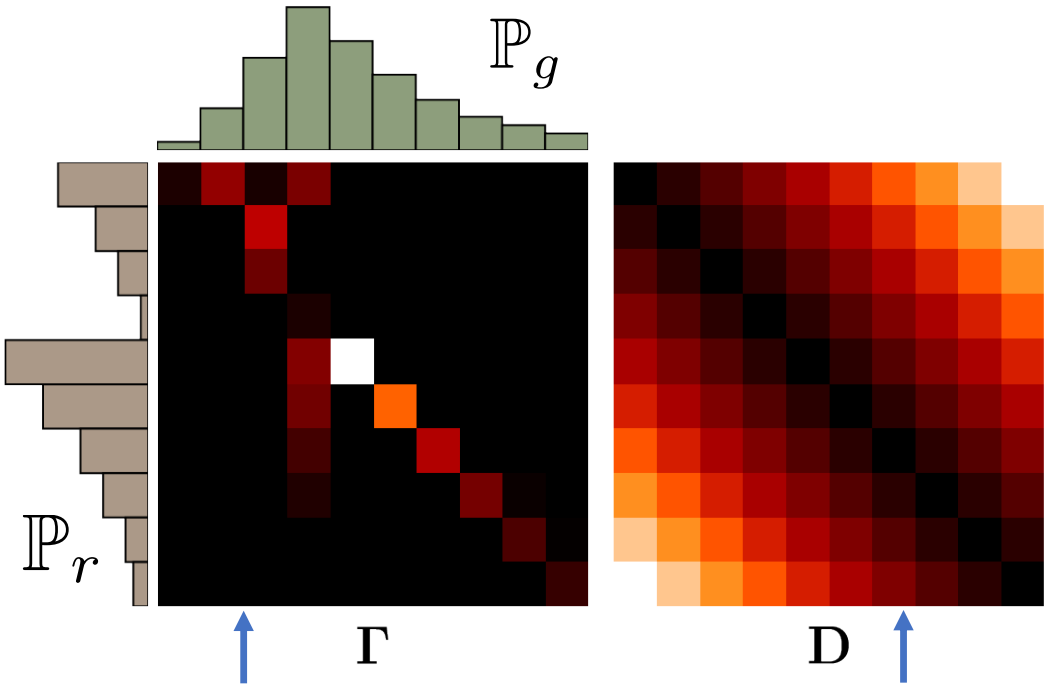
$$\sum_{(x,y)} \left\{ \|x-y\| \gamma(x,y) \right\}$$

$$= \|3-1\| \cdot 0.5 + \|1-3\| \cdot 0.5 = 2$$

EMDまとめ

$$\mathrm{EMD}(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$

$$\mathrm{EMD}(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi} \langle \mathbf{D}, \mathbf{\Gamma} \rangle_{\mathrm{F}}$$



$\langle \mathbf{D}, \mathbf{\Gamma} \rangle_{\mathrm{F}}$ はフロベニウス積

可能な限り対角成分に大きい数値を当てた γ が最適解(EMD)を与える γ となる.

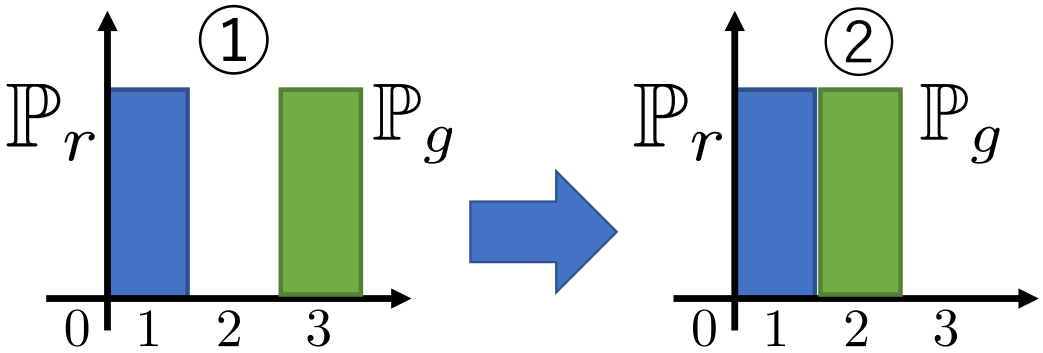
横軸x縦軸y.

ある γ について表示したもの

$\|x - y\|$ をヒストグラムで表示したもの

WGANに有効な理由

- 右のように緑の分布を近づける.
- ①②それぞれにおけるEMDは次のようになる.



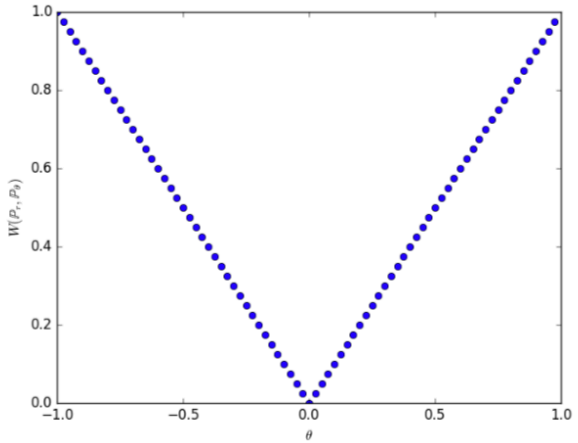
$$\mathrm{EMD}(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y)$$

① $\mathrm{EMD}(\mathbb{P}_r, \mathbb{P}_g) = \|3 - 1\| \cdot 1 = 2$

② $\mathrm{EMD}(\mathbb{P}_r, \mathbb{P}_g) = \|2 - 1\| \cdot 1 = 1$

- 分布が一致したときもちろん0.

緑の分布を動かし続けたときのEMDの値



絶対値がそのまま現れて定数にならず不連続でもない.

参考文献

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

2. Goodfellow, I. (2016). NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

3. Mohamed, S., & Lakshminarayanan, B. (2016). Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*.

4. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

5. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

6. Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

7. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. *arXiv preprint arXiv:1701.07875*.