The story about WGAN

Ngày 23 tháng 11 năm 2019

GAN - What is the optimal value for D ?

Loss function of GAN

$$L = \mathbb{E}_{x \sim P_r} \left[\log D\left(x\right)\right] + \mathbb{E}_{x \sim P_g} \left[\log\left(1 - D\left(x\right)\right)\right]$$

$$f\left(D\left(x\right)\right) = P_r\left(x\right) \log D\left(x\right) + P_g\left(x\right) \log\left(1 - D\left(x\right)\right)$$

$$\frac{\partial f\left(D\left(x\right)\right)}{\partial D\left(x\right)} = P_r\left(x\right) * \frac{1}{D\left(x\right)} - P_g\left(x\right) \frac{1}{1 - D\left(x\right)} = 0$$

$$D^*\left(x\right) = \frac{P_r\left(x\right)}{P_r\left(x\right) + P_g\left(x\right)}$$

Nếu hình ảnh là một hình ảnh thật thì Discriminator trả về xác suất là 0.5. Mặt khác, nếu hình ảnh là giả thì Discriminator trả về 0

The main problem : optimal discriminator

Loss function of GAN

$$L = \mathbb{E}_{x\sim P_r}\left[\log D\left(x\right)\right] + \mathbb{E}_{x\sim P_g}\left[\log\left(1 - D\left(x\right)\right)\right]$$

$$\begin{aligned}
&L = \mathbb{E}_{x\sim P_r}\left[\log D\left(x\right)\right] + \mathbb{E}_{x\sim P_g}\left[\log\left(1 - D\left(x\right)\right)\right] \\
&= \mathbb{E}_{x\sim P_r}\left[\log \frac{P_r\left(x\right)}{P_r\left(x\right) + P_g\left(x\right)}\right] + \mathbb{E}_{x\sim P_g}\left[\log\left(1 - \frac{P_r\left(x\right)}{P_r\left(x\right) + P_g\left(x\right)}\right)\right] \\
&= \mathbb{E}_{x\sim P_r}\left[\log \frac{P_r\left(x\right)}{2 \times \frac{1}{2}\left(P_r\left(x\right) + P_g\left(x\right)\right)}\right] + \mathbb{E}_{x\sim P_g}\left[\log\left(\frac{P_g\left(x\right)}{2 \times \frac{1}{2}\left(P_r\left(x\right) + P_g\left(x\right)\right)}\right)\right] \\
&= \mathbb{E}_{x\sim P_r}\left[\log \frac{P_r\left(x\right)}{\frac{1}{2}\left(P_r\left(x\right) + P_g\left(x\right)\right)}\right] + \mathbb{E}_{x\sim P_g}\left[\log\left(\frac{P_g\left(x\right)}{\frac{1}{2}\left(P_r\left(x\right) + P_g\left(x\right)\right)}\right)\right] - 2\log 2 \\
&= \mathbb{E}_{x\sim P_r}\left[\log \frac{P_r\left(x\right)}{P_{average}\left(x\right)}\right] + \mathbb{E}_{x\sim P_g}\left[\log\left(\frac{P_g\left(x\right)}{P_{average}\left(x\right)}\right)\right] - 2\log 2 \\
&= 2JS\left(P_r|P_g\right) - 2\log 2
\end{aligned}$$

Như kết quả trên, nếu ta tiếp trục training Discriminator thì giá trị của hàm loss sẽ bằng 0, và JS-divergence sẽ bằng $\log 2$

Different distance

- The Total Variation (TV) distance is :

$$\delta\left(P_r, P_g\right) = \sup_{A \in \sum} \left|P_r\left(A\right) - P_g\left(A\right)\right|$$

- The Kullback-Leibler (KL) divergence is

$$KL\left(P_r \,\|P_g\right) = \int \log\left(\frac{P_r\left(x\right)}{P_g\left(x\right)}\right) P_r\left(x\right) d\mu\left(x\right)$$

- The Jenson-Shannon (JS) divergence

$$JS\left(P_r, P_g\right) = KL\left(P_r \,\|P_m\right) + KL\left(P_g \,\|P_m\right)$$

- The Earth Mover (EM) or Wasserstein distance

$$W\left(P_r, P_g\right) = \inf_{\gamma \in \prod\left(P_r, P_g\right)} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x - y\|\right]$$

Different distance

Example

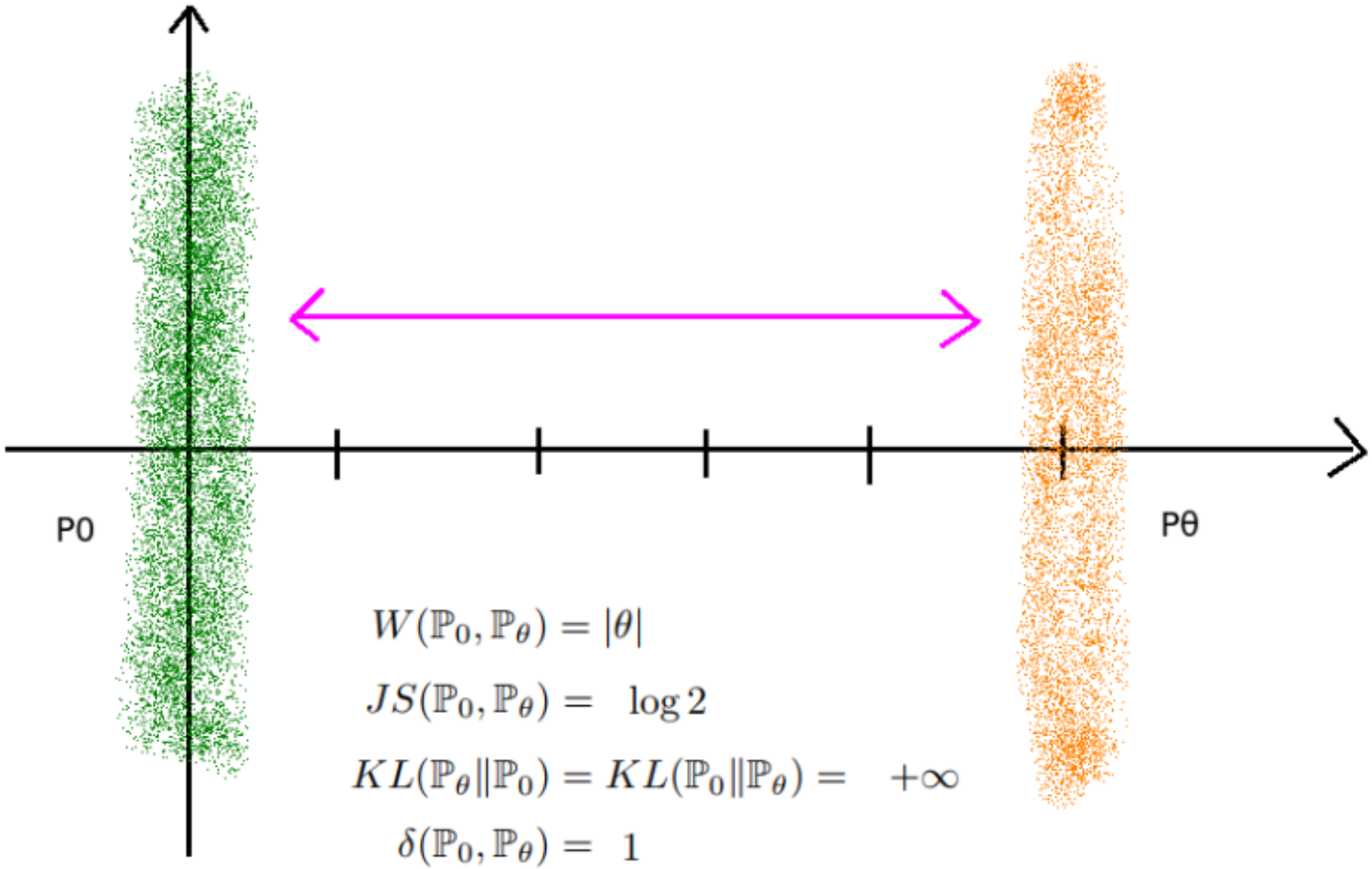
Giả sử ta có hai distributions, $\mathbb{P}$ và $\mathbb{Q}$:

$$\forall (x, y) \in \mathbb{P}, x = 0 \text{ và } y \sim U(0, 1)$$
$$\forall (x, y) \in \mathbb{Q}, x = \theta, 0 \leqslant \theta \leqslant 1 \text{ và } y \sim U(0, 1)$$

- $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$
- $JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} 2 \log 2 \text{ nếu } \theta \neq 0 \\ 0 \text{ nếu } \theta = 0 \end{cases}$
- $KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = KL(\mathbb{P}_\theta \| \mathbb{P}_0) = \begin{cases} +\infty \text{ nếu } \theta \neq 0 \\ 0 \text{ nếu } \theta = 0 \end{cases}$
- $\delta(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} 1 \text{ nếu } \theta \neq 0 \\ 0 \text{ nếu } \theta = 0 \end{cases}$

Chỉ có duy nhất Wasserstein metric mới có a smooth measure, phù hợp để sử dụng gradient.

Different distance



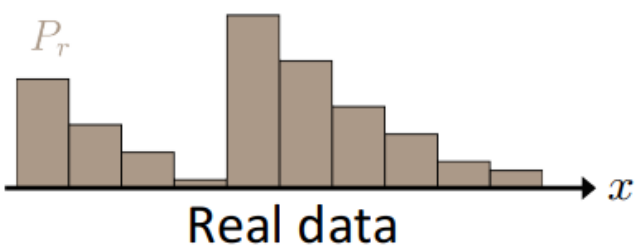
$$W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$$

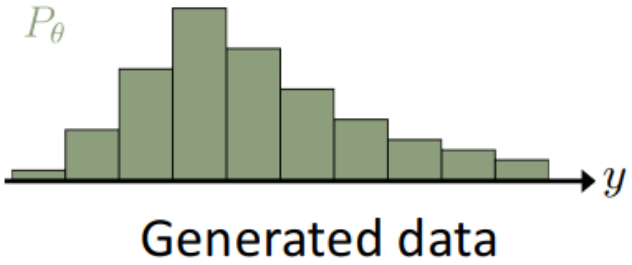
$$JS(\mathbb{P}_0, \mathbb{P}_\theta) = \log 2$$

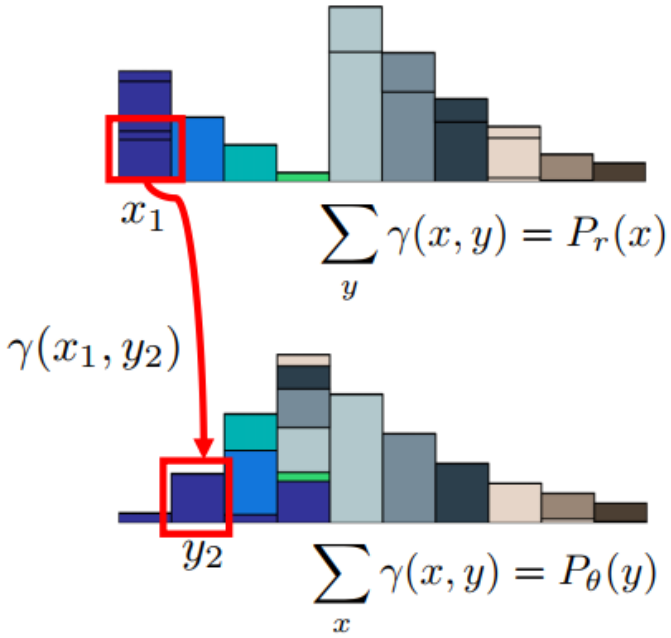
$$KL(\mathbb{P}_\theta \| \mathbb{P}_0) = KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = +\infty$$

$$\delta(\mathbb{P}_0, \mathbb{P}_\theta) = 1$$

Earth-Mover (EM) distance/ Wasserstein Metric



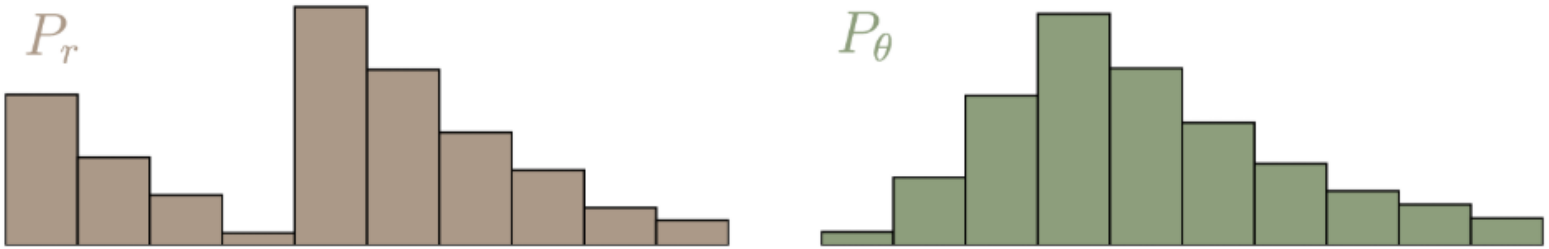




Earth-Mover (EM) distance/ Wasserstein Metric

$$\mathrm{EMD}\left(P_r, P_\theta\right) = \inf_{\gamma \in \prod} \sum_{x,y} \|x - y\| \gamma\left(x, y\right) = \inf_{\gamma \in \prod} E_{(x,y)\sim\gamma} \|x - y\|$$

Earth-Mover (EM) distance/ Wasserstein Metric



$$\text{EMD}\left(P_r, P_\theta\right) = \inf_{\gamma \in \prod} \sum_{x,y} \|x - y\| \gamma\left(x, y\right) = \inf_{\gamma \in \prod} E_{(x,y)\sim\gamma} \|x - y\|$$

Đặt

$$\begin{cases} \Gamma = \gamma\left(x, y\right) \\ D = \|x - y\| \end{cases}$$

Ta viết lại :

$$EMD\left(P_r, P_\theta\right) = \inf_{\gamma \in \prod} \langle D, \Gamma \rangle_F$$

Để tính được phương án di chuyển tối ưu Γ, ta dùng Linear Programming

Linear Programming

Our Problem

Objective function : $\mathrm{EMD}\left(P_r, P_\theta\right) = \inf_{\gamma \in \Pi\left(P_r, P_\theta\right)} \langle \mathrm{D}, \Gamma \rangle_F$

Constraint :

$\Sigma_y \gamma\left(x, y\right) = P_r\left(x\right)$

$\Sigma_x \gamma\left(x, y\right) = P_\theta\left(y\right)$

$\forall x, y \quad \gamma(\ x, y) \ \geqslant 0$

Linear Programming

Objective function : minimize $z = c^T x$

Constraint :

$Ax = b$

$x \geqslant 0$

Linear Programming

Objective function :

Minimize $z = c^T x$

$$\mathrm{EMD}\left(P_r, P_\theta\right) = \inf_{\gamma \in \Pi(P_r, P_\theta)} \langle \mathrm{D}, \Gamma \rangle_F$$

$$c = \mathrm{vec}\left(D\right) \qquad x = \mathrm{vec}\left(\Gamma\right)$$

$$\begin{bmatrix} \|x_1 - y_1\| \\ \|x_1 - y_2\| \\ \vdots \\ \|x_2 - y_1\| \\ \|x_2 - y_2\| \\ \vdots \\ \|x_n - y_1\| \\ \|x_n - y_2\| \\ \vdots \end{bmatrix} \qquad \begin{bmatrix} \gamma\left(x_1, y_1\right) \\ \gamma\left(x_1, y_2\right) \\ \vdots \\ \gamma\left(x_2, y_1\right) \\ \gamma\left(x_2, y_2\right) \\ \vdots \\ \gamma\left(x_n, y_1\right) \\ \gamma\left(x_n, y_2\right) \\ \vdots \end{bmatrix}$$

Linear Progamming

Điều kiện :

$Ax = b$
$\Sigma_y \gamma(x, y) = P_r(x)$
$\Sigma_x \gamma(x, y) = P_\theta(y)$

$$\underbrace{\begin{bmatrix} 1 & 1 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots \\ 0 & 0 & \cdots & 1 & 1 & \cdots & 0 & 0 & \cdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 & 1 & \cdots \\ \hline 1 & 0 & \cdots & 1 & 0 & \cdots & 1 & 0 & \cdots \\ 0 & 1 & \cdots & 0 & 1 & \cdots & 0 & 1 & \cdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots \end{bmatrix}}_{\mathrm{A}} \underbrace{\begin{bmatrix} \gamma(x_1, y_1) \\ \gamma(x_1, y_2) \\ \vdots \\ \hline \gamma(x_2, y_1) \\ \gamma(x_2, y_2) \\ \vdots \\ \hline \gamma(x_n, y_1) \\ \gamma(x_n, y_2) \\ \vdots \end{bmatrix}}_{x = \mathrm{vec}(\Gamma)} = \underbrace{\begin{bmatrix} P_r \\ P_\theta \end{bmatrix}}_{b} \begin{bmatrix} P_r(x_1) \\ P_r(x_2) \\ \vdots \\ P_r(x_n) \\ \hline P_\theta(y_1) \\ \vdots \\ P_\theta(y_2) \\ P_\theta(y_n) \end{bmatrix}$$

Dual form

Primary form

$$\begin{aligned} &\text{minimize} \quad z = c^T x \\ &\text{so that} \quad Ax = b \\ &\text{and} \quad x \geqslant 0 \end{aligned}$$

Dual form

$$\begin{aligned} &\text{maximize} \quad \widetilde{z} = b^T y \\ &\text{so that} \quad A^T y \leqslant c \end{aligned}$$

Dual form

Objective function

$$\widetilde{z} = b^T y$$

$$b = \begin{bmatrix} P_r \\ P_\theta \end{bmatrix} \quad y = \begin{bmatrix} f \\ g \end{bmatrix}$$

$$\begin{bmatrix} P_r(x_1) \\ P_r(x_2) \\ \vdots \\ P_r(x_n) \\ P_\theta(y_1) \\ P_\theta(y_n) \\ \vdots \\ P_\theta(y_n) \end{bmatrix} \quad \begin{bmatrix} f(x_1) \\ f(x_1) \\ \vdots \\ f(x_1) \\ g(x_1) \\ g(x_1) \\ \vdots \\ g(x_1) \end{bmatrix}$$

$$\Longrightarrow \mathrm{EMD}(P_r, P_\theta) = f^T P_r + g^T P_\theta$$

Dual form

Điều kiện

$$A^T y \leqslant c$$

$$\underbrace{\left[\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 & 0 & \cdots & 0 \\ & & & & & & \vdots & \\ 0 & 0 & \cdots & 1 & 0 & 1 & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}\right]}_{A^T} \quad \underbrace{\left[\begin{array}{c} f \\ g \end{array}\right]}_{y} \quad \left[\begin{array}{c} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \\ g(x_1) \\ g(x_2) \\ \vdots \\ g(x_n) \end{array}\right] \quad \leqslant \quad \underbrace{\left[\begin{array}{c} \|x_1 - x_1\| \\ \|x_1 - x_2\| \\ \vdots \\ \|x_2 - x_1\| \\ \|x_2 - x_2\| \\ \vdots \\ \|x_n - x_1\| \\ \|x_n - x_2\| \\ \vdots \end{array}\right]}_{c = vec(D)}$$

$$\Rightarrow \forall i, j \quad f(x_i) + g(x_j) \leqslant \|x_i - x_j\|$$

Dual form

Điều kiện

$$\forall i, j \quad f(x_i) + g(x_i) \leqslant \|x_i - x_j\|$$

If $i = j$:

$$(x_i) + g(x_i) \leqslant \|x_i - x_i\| = 0$$

Ta muốn maximize : $\mathrm{EMD}(P_r, P_\theta) = f^T P_r + g^T P_\theta$

$$\Longrightarrow f(x_i) = -g(x_i)$$

Từ (1) và (2), ta được :

$$\begin{cases} f(x) - f(x) \leqslant \|x - x\| \\ f(x) - f(x) \geqslant -\|x - x\| \end{cases}$$

$$\Rightarrow -1 \leqslant \frac{f(x) - f(x)}{\|x - x\|} \leqslant 1$$

$$\Rightarrow \|f\|_{L \leqslant 1}$$

Vậy dạng Dual form của EMD là :

$$\mathrm{EMD}(P_r, P_\theta) = \sup_{\|f\|_{L \leqslant 1}} \mathbb{E}_{x \sim P_r} f(x) - \mathbb{E}_{x \sim P_\theta} f(x)$$

WGAN

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do** ← Train until convergence

  for $t = 0, ..., n_{\text{critic}}$ **do** — Critic Training

    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.

    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.

    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))\right]$

    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$

    $w \leftarrow \text{clip}(w, -c, c)$

  end for

  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples. — Generator Training

  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$

  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$

end while

Ta nhận thấy được quá trình training của WGAN khác với GAN thông thường :

- The critics sẽ được cập nhật nhiều lần
- Trong quá tình loss, WGAN không sử dụng hàm logarit (không sử dụng cross entropy)
- Sử dụng weight clipping
- Không sử dụng các momentum

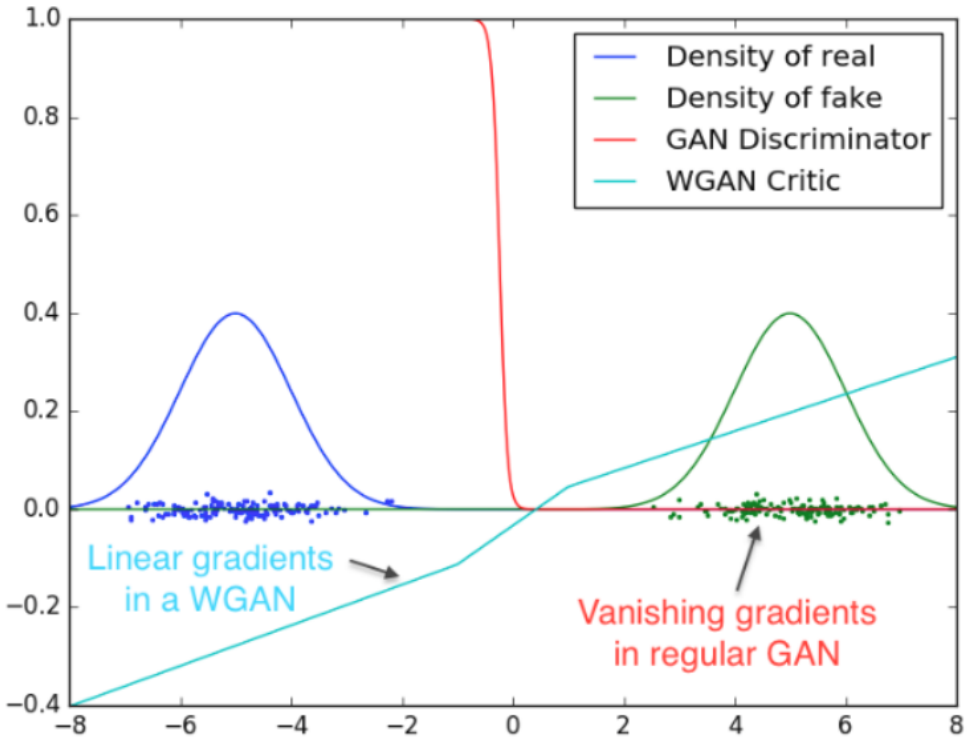
WGAN & GAN

- Trong GAN, Discriminator cố gắng maximizes :

$$\frac{1}{m}\left(\sum_{i=1}^{m}\log D\left(x^{(i)}\right)+\sum_{i=1}^{m}\log\left(1-D\left(g_\theta\left(z^{(i)}\right)\right)\right)\right)$$

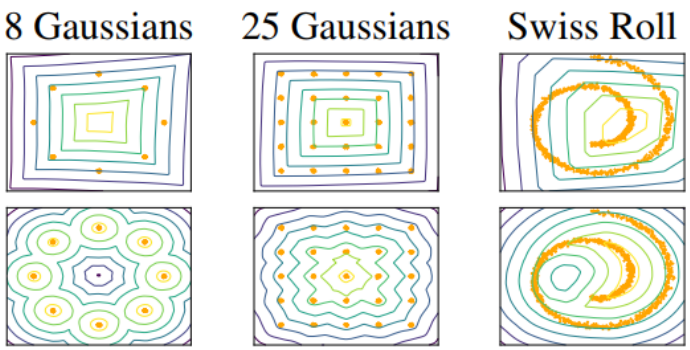
  trong đó, $D(x)$ là một sác xuất $p \in (0,1)$ Trong WGAN, nó không đưa ra một xác suất cụ thể, mà nó đưa ra khoảng cách EMD của hai distribiution
- Trong paper gốc của GAN, các độ đo khoảng cách sẽ dựa vào JS divergence Trong WGAN, nó được thay thế bằng Wasserstein distance

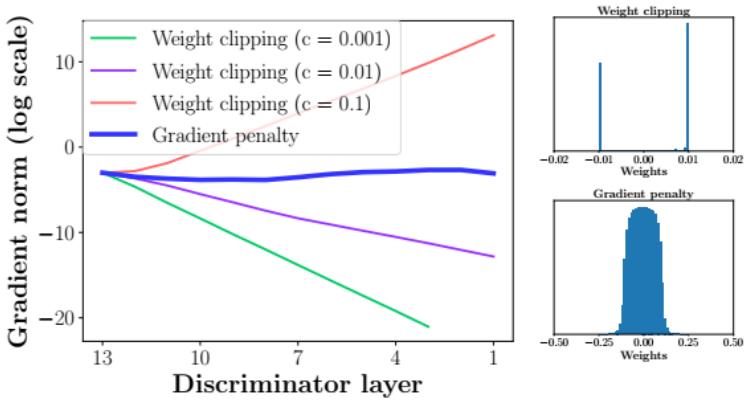
Result of WGAN



$$\max_{w \in W} \mathbb{E}_{x \sim P_r} \left[f_w \left(x \right) \right] - \mathbb{E}_{z \sim p(z)} \left[f_w \left(g_\theta \left(z \right) \right) \right]$$

Problem in WGAN

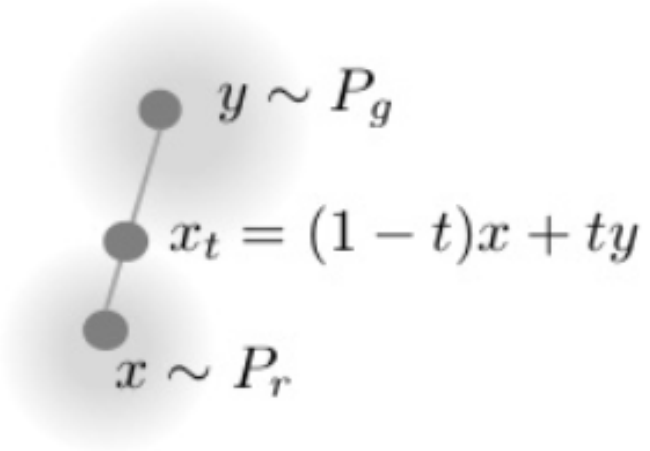




- Sau khi WGAN được công bố, rất nhiều developer tìm ra được vấn đề của WGAN là tốc độ hội tụ của nó chậm và chất lượng hình ảnh được tạo ra không được tốt
- Nếu như ta chọn giá trị của weight clipping quá nhỏ thì sẽ bị dẫn đến hiện tượng vanishing gradient, ngược lại nếu quá lớn thì sẽ dẫn đến hiện tượng exploding gradient
- Việc sử dụng weight clipping sẽ làm giảm khả năng học được các distribution phức tạp của tập dataset, chỉ học được các function đơn giản

Gradient penalty

Optimal critic has gradient with norm 1 almost everywhere under P_r và P_g



$$\nabla f^*(x_t) = \frac{y - x_t}{\|y - x_t\|}$$

$$\Rightarrow \quad \|\nabla f^*(x_t)\| = 1$$

Loss function

$$L = \underbrace{\mathop{\mathbb{E}}_{\widetilde{x}\sim P_g}\left[D\left(\widetilde{x}\right)\right] - \mathop{\mathbb{E}}_{x\sim P_r}\left[D\left(x\right)\right]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\widehat{x}\sim P_{\widehat{x}}}\left[\left(\|\nabla_{\widehat{x}} D\left(\widehat{x}\right)\|_2 - 1\right)^2\right]}_{\text{Gradient penalty}}$$

WGAN-GP

Algorithm 1 WGAN with gradient penalty. We use default values of $\lambda = 10$, $n_{\text{critic}} = 5$, $\alpha = 0.0001$, $\beta_1 = 0$, $\beta_2 = 0.9$.

Require: The gradient penalty coefficient λ, the number of critic iterations per generator iteration n_{critic}, the batch size m, Adam hyperparameters α, β_1, β_2.
Require: initial critic parameters w_0, initial generator parameters θ_0.

while θ has not converged **do**
  for $t = 1, ..., n_{\text{critic}}$ **do**
    for $i = 1, ..., m$ **do**
      Sample real data $\boldsymbol{x} \sim \mathbb{P}_r$, latent variable $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random number $\epsilon \sim U[0, 1]$.
      $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z})$
      $\hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1 - \epsilon)\tilde{\boldsymbol{x}}$
      $L^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + \lambda(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
    end for
    $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha, \beta_1, \beta_2)$
  end for
  Sample a batch of latent variables $\{\boldsymbol{z}^{(i)}\}_{i=1}^{m} \sim p(\boldsymbol{z})$.
  $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -D_w(G_\theta(\boldsymbol{z})), \theta, \alpha, \beta_1, \beta_2)$
end while

Ở thuật toán này ta đã có thể sử dụng được momentum để cập nhật được model. Mỗi khi cập nhật critics, gradient penalty được thêm vào hàm loss