

# **Generative Adversarial Networks**

---

Davi Barreira

FGV - Escola de Matemática Aplicada

# Table of contents

1. Introdução
2. Formalização Teórica
3. Implementações e Desafios
4. Implementações e Desafios

# Introdução

**Generative Adversarial Networks (GAN)** foram originalmente introduzidas por Goodfellow et al. (2014). Essas redes são utilizadas com o objetivo de gerar dados sintéticos realísticos a partir de dados reais.



**Figure 1:** Faces geradas por GAN <sup>1</sup>.

---

<sup>1</sup><https://towardsdatascience.com/generating-modern-arts-using-generative-adversarial-network-gan-on-spell-39f67f83c7b4>

# Introdução

A geração de novas amostras sintéticas tem diferentes utilidades, como aprendizado semi-supervisionado, geração de exemplos adversariais, *style transfer*, entre outros.



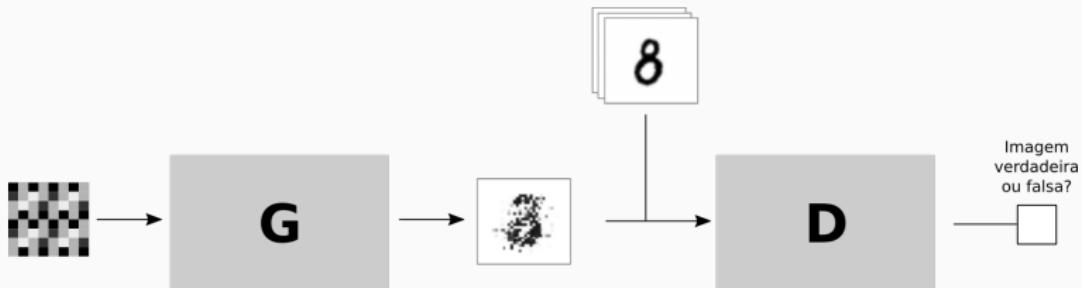
**Figure 2:** Style transfer utilizando CycleGan <sup>2</sup>.

---

<sup>2</sup><https://towardsdatascience.com/style-transfer-with-gans-on-hd-images-88e8efcf3716>

# Introdução

A ideia geral por trás das GANs é utilizar duas redes neurais competindo uma com a outra, sendo uma rede responsável por gerar amostras parecidas com os dados reais (*gerador*) , enquanto a outra busca identificar quando o dado é real ou sintético (*descriminador*).



**Figure 3:** Desenho esquemático de uma GAN "convencional".

## Formalização Teórica

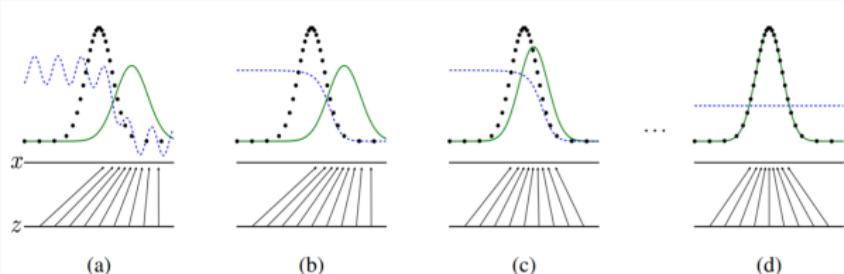
Na formalização teórica da modelagem das redes adversariais, consideraremos que o gerador e o descriminador são ambos *multilayer perceptrons*. Os dados reais possuem uma distribuição  $p_{data}(\mathbf{x})$ , enquanto  $p_g$  é a distribuição do gerador e  $p_z(z)$  é a priori do ruído de entrada. A função  $G(z, \theta_g)$  é a função diferenciável que transforma  $\mathbf{z}$  no dado sintético, onde  $\theta_g$  são os parâmetros da rede. A função  $D(\mathbf{x}, \theta_d)$  retorna a probabilidade de  $\mathbf{x}$  ter sido amostrada de  $p_{data}$  invés de  $p_g$ .

- $p_g$  - Distribuição dos dados sintéticos;
- $p_z$  - Distribuição priori dos rúidos de entrada;
- $p_{data}$  - Distribuição real dos dados;
- $G(z, \theta_g)$  - Função geradora;
- $D(\mathbf{x}, \theta_d)$  - Função discriminadora.

# Formalização Teórica

Nós treinamos  $D$  buscando maximizar a capacidade de discernir dados de  $p_{data}$  de  $p_g$ . Ao mesmo tempo que treinamos  $G$  para minimizar  $\log(1 - D(G(z)))$ . O treino da rede se resume ao problema de otimização dado pela seguinte função objetivo:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{z \sim p_z(z)} [1 - \log(D(G(z)))]$$



**Figure 4:** De (a) até (d), o desenho ilustra a evolução do algoritmo ao ser treinado. A linha azul representa a distribuição do discriminador, a linha verde representa a  $p_g$ , e os pontos pretos representam  $p_{data}$ <sup>3</sup>.

<sup>3</sup>Imagem de Goodfellow et al. (2014)

# Formalização Teórica

---

## Algorithm 1: GAN descrita em Goodfellow et al. (2014)

---

**for** número de iterações de treino **do**

**for**  $k$  passos **do**

        Amostre  $m$  valores  $\{z^{(1)}, \dots, z^{(m)}\}$  da priori  $p_z(z)$ ;

        Amostre  $m$  exemplos  $\{x^{(1)}, \dots, x^{(m)}\}$  da função dos dados  $p_{data}(x)$ ;

        Atualize o *discriminator* utilizando *stochastic gradient descent*:

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

**end**

    Amostre  $m$  valores  $\{z^{(1)}, \dots, z^{(m)}\}$  da priori  $p_z(z)$ ;

    Atualize o *generator* utilizando *stochastic gradient descent*:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \log(1 - D(G(z^{(i)})))$$

**end**

---

# Formalização Teórica

Vamos estabelecer alguns resultados teóricos do funcionamento do algoritmo.

**Proposição 1.** Para  $G$  fixo, o discriminador  $D$  ótimo é  $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ .

**Teorema 1.** O mínimo global da função objetivo é atingido se, e somente se,  $p_g = p_{data}$ . Neste ponto, o mínimo é  $-\log 4$ .

**Proposição 2.** Se  $G$  e  $D$  tiverem capacidade suficiente, e, em cada passo do Algoritmo 1, o discriminador atingir o seu ótimo dado  $G$  com  $p_g$  sendo atualizado para melhorar o critério

$$\mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{x \sim p_g(x)} [1 - \log(D(x))]$$

então  $p_g$  converge para  $p_{data}$ .

# Formalização Teórica

**Proposição 1.** Para  $G$  fixo, o discriminador  $D$  ótimo é  $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ .

Demonstração:

---

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{z \sim p_z(z)} [1 - \log(D(G(z)))]$$

# Formalização Teórica

**Proposição 1.** Para  $G$  fixo, o discriminador  $D$  ótimo é  $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ .

Demonstração:

$$\begin{aligned} V(D, G) &= \mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{z \sim p_z(z)} [1 - \log(D(G(z)))] \\ &= \int_x p_{data}(x) \log(D(x)) dx + \int_z p_z(z) \log(1 - D(G(z))) dz \end{aligned}$$

# Formalização Teórica

**Proposição 1.** Para  $G$  fixo, o discriminador  $D$  ótimo é  $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ .

Demonstração:

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{z \sim p_z(z)} [1 - \log(D(G(z)))]$$

$$= \int_x p_{data}(x) \log(D(x)) dx + \int_z p_z(z) \log(1 - D(G(z))) dz$$

$$x = G(z) \implies z = G^{-1}(x) \implies dz = (G^{-1}(x))' dx$$

$$p_g(x) = p_z(G^{-1})(G^{-1})'(x) dx$$

# Formalização Teórica

**Proposição 1.** Para  $G$  fixo, o discriminador  $D$  ótimo é  $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ .

Demonstração:

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{z \sim p_z(z)} [1 - \log(D(G(z)))]$$

$$= \int_x p_{data}(x) \log(D(x)) dx + \int_z p_z(z) \log(1 - D(G(z))) dz$$

$$x = G(z) \implies z = G^{-1}(x) \implies dz = (G^{-1}(x))dx$$

$$p_g(x) = p_z(G^{-1})(G^{-1})'(x)dx$$

$$= \int_x p_{data}(x) \log(D(x)) dx + \int_x p_z(G^{-1}(x)) \log(1 - D(x))(G^{-1})'(x)dx$$

# Formalização Teórica

**Proposição 1.** Para  $G$  fixo, o discriminador  $D$  ótimo é  $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ .

Demonstração:

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{z \sim p_z(z)} [1 - \log(D(G(z)))]$$

$$= \int_x p_{data}(x) \log(D(x)) dx + \int_z p_z(z) \log(1 - D(G(z))) dz$$

$$\begin{aligned} x = G(z) &\implies z = G^{-1}(x) \implies dz = (G^{-1}(x))dx \\ p_g(x) &= p_z(G^{-1})(G^{-1})'(x)dx \end{aligned}$$

$$= \int_x p_{data}(x) \log(D(x)) dx + \int_x p_z(G^{-1}(x)) \log(1 - D(x))(G^{-1})'(x) dx$$

$$= \int_x p_{data}(x) \log(D(x)) dx + \int_x p_g(x) \log(1 - D(x)) dx$$

# Formalização Teórica

**Proposição 1.** Para  $G$  fixo, o discriminador  $D$  ótimo é  $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ .

Demonstração:

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{z \sim p_z(z)} [1 - \log(D(G(z)))]$$

$$= \int_x p_{data}(x) \log(D(x)) dx + \int_z p_z(z) \log(1 - D(G(z))) dz$$

$$\begin{aligned} x = G(z) &\implies z = G^{-1}(x) \implies dz = (G^{-1}(x))dx \\ p_g(x) &= p_z(G^{-1})(G^{-1})'(x)dx \end{aligned}$$

$$= \int_x p_{data}(x) \log(D(x)) dx + \int_x p_z(G^{-1}(x)) \log(1 - D(x))(G^{-1})'(x) dx$$

$$= \int_x p_{data}(x) \log(D(x)) dx + \int_x p_g(x) \log(1 - D(x)) dx$$

$$= \int_x p_{data}(x) \log(D(x)) + p_g(x) \log(1 - D(x)) dx$$

# Formalização Teórica

**Proposição 1.** Para  $G$  fixo, o discriminador  $D$  ótimo é  $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ .

Demonstração:

---

$$\max_D V(D, G) = \max_D \int_x p_{data}(x) \log(D(x)) + p_g(x) \log(1 - D(x)) dx$$

# Formalização Teórica

**Proposição 1.** Para  $G$  fixo, o discriminador  $D$  ótimo é  $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ .

Demonstração:

$$\max_D V(D, G) = \max_D \int_X p_{data}(x) \log(D(x)) + p_g(x) \log(1 - D(x)) dx$$

$$\frac{\partial}{\partial D(x)} (p_{data}(x) \log(D(x)) + p_g(x) \log(1 - D(x))) = 0$$

# Formalização Teórica

**Proposição 1.** Para  $G$  fixo, o discriminador  $D$  ótimo é  $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ .

Demonstração:

$$\max_D V(D, G) = \max_D \int_X p_{data}(x) \log(D(x)) + p_g(x) \log(1 - D(x)) dx$$

$$\frac{\partial}{\partial D(x)} (p_{data}(x) \log(D(x)) + p_g(x) \log(1 - D(x))) = 0$$

$$\implies \frac{p_{data}(x)}{D(x)} - \frac{p_g(x)}{1 - D(x)} = 0$$

$$\implies D(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

□

# Formalização Teórica

**Teorema 1.** O mínimo global da função objetivo é atingido se, e somente se,  $p_g = p_{data}$ . Neste ponto, o mínimo é  $-\log 4$ .

Demonstração:

---

$\implies$  ) Seja  $p_g = p_{data}$ ,  $D_G^*(x) = \frac{1}{2}$ . Assim,

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log(1/2)] + \mathbb{E}_{x \sim p_g(x)} [\log(1/2)] = -\log 4$$

# Formalização Teórica

**Teorema 1.** O mínimo global da função objetivo é atingido se, e somente se,  $p_g = p_{data}$ . Neste ponto, o mínimo é  $-\log 4$ .

Demonstração:

---

$\Leftarrow$  ) Seja  $C(G) = \max_D V(G, D)$ , assim

$$C(G) = \int_x p_{data}(x) \log(D_G^*(x)) + p_g(x) \log(1 - D_g^*(x)) dx$$

# Formalização Teórica

**Teorema 1.** O mínimo global da função objetivo é atingido se, e somente se,  $p_g = p_{data}$ . Neste ponto, o mínimo é  $-\log 4$ .

Demonstração:

---

$\Leftarrow$  ) Seja  $C(G) = \max_D V(G, D)$ , assim

$$\begin{aligned} C(G) &= \int_x p_{data}(x) \log(D_G^*(x)) + p_g(x) \log(1 - D_g^*(x)) dx \\ &= \int_x p_{data}(x) \log\left(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right) + p_g(x) \log\left(\frac{p_g(x)}{p_{data}(x) + p_g(x)}\right) dx \end{aligned}$$

# Formalização Teórica

**Teorema 1.** O mínimo global da função objetivo é atingido se, e somente se,  $p_g = p_{data}$ . Neste ponto, o mínimo é  $-\log 4$ .

Demonstração:

---

$\Leftarrow$  ) Seja  $C(G) = \max_D V(G, D)$ , assim

$$\begin{aligned} C(G) &= \int_x p_{data}(x) \log(D_G^*(x)) + p_g(x) \log(1 - D_g^*(x)) dx \\ &= \int_x p_{data}(x) \log\left(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right) + p_g(x) \log\left(\frac{p_g(x)}{p_{data}(x) + p_g(x)}\right) dx \\ &= \int_x p_{data}(x) \log\left(2^{-1} \cdot \frac{p_{data}(x)}{\frac{p_{data}(x) + p_g(x)}{2}}\right) + p_g(x) \log\left(2^{-1} \cdot \frac{p_g(x)}{\frac{p_{data}(x) + p_g(x)}{2}}\right) dx \end{aligned}$$

# Formalização Teórica

**Teorema 1.** O mínimo global da função objetivo é atingido se, e somente se,  $p_g = p_{data}$ . Neste ponto, o mínimo é  $-\log 4$ .

Demonstração:

$\Leftarrow$  ) Seja  $C(G) = \max_D V(G, D)$ , assim

$$\begin{aligned} C(G) &= \int_x p_{data}(x) \log(D_G^*(x)) + p_g(x) \log(1 - D_g^*(x)) dx \\ &= \int_x p_{data}(x) \log\left(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right) + p_g(x) \log\left(\frac{p_g(x)}{p_{data}(x) + p_g(x)}\right) dx \\ &= \int_x p_{data}(x) \log\left(2^{-1} \cdot \frac{p_{data}(x)}{\frac{p_{data}(x) + p_g(x)}{2}}\right) + p_g(x) \log\left(2^{-1} \cdot \frac{p_g(x)}{\frac{p_{data}(x) + p_g(x)}{2}}\right) dx \\ &= \mathbb{E}_{x \sim p_{data}(x)} \left[ -\log(2) + \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right] + \mathbb{E}_{x \sim p_g(x)} \left[ -\log(2) + \frac{p_g(x)}{p_{data}(x) + p_g(x)} \right] \end{aligned}$$

# Formalização Teórica

**Teorema 1.** O mínimo global da função objetivo é atingido se, e somente se,  $p_g = p_{data}$ . Neste ponto, o mínimo é  $-\log 4$ .

Demonstração:

---

$$\begin{aligned} C(G) &= KL \left[ p_{data}(x) \parallel \frac{p_{data}(x) + p_g(x)}{2} \right] + KL \left[ p_g(x) \parallel \frac{p_{data}(x) + p_g(x)}{2} \right] - \log 4 \\ &= 2 \cdot JSD [p_{data} \parallel p_g] - \log 4 \end{aligned}$$

Onde  $KL$  é a distância Kullback-Leibler e  $JSD$  é a divergência de Jensen-Shannon. Assim:

$$\min_G C(G) = \min_G (2 \cdot JSD [p_{data} \parallel p_g] - \log 4)$$

O mínimo da divergência  $JSD$  é zero e só é atingido se, e somente se,  $p_g = p_{data}$ .<sup>4</sup>

<sup>4</sup>Estamos assumindo que o modelo gerativo é capaz de reproduzir perfeitamente a distribuição dos dados

□

## Formalização Teórica

**Proposição 2.** Se  $G$  e  $D$  tiverem capacidade suficiente, e, em cada passo do Algoritmo 1, o discriminador atingir o seu ótimo dado  $G$  com  $p_g$  sendo atualizado para melhorar o critério

$$\mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{x \sim p_g(x)} [1 - \log(D(x))]$$

então  $p_g$  converge para  $p_{data}$ .

Demonstração:

---

Considere  $V(G, D) = U(p_g, D)$ . Assim, para um  $D$  fixo,  $U$  é função de  $p_g$ . Note que  $U(p_g, D)$  é convexo, pois

$$U(p_g, D) = \mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{x \sim p_g(x)} [1 - \log(D(x))] \therefore$$

# Formalização Teórica

**Proposição 2.** Se  $G$  e  $D$  tiverem capacidade suficiente, e, em cada passo do Algoritmo 1, o discriminador atingir o seu ótimo dado  $G$  com  $p_g$  sendo atualizado para melhorar o critério

$$\mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{x \sim p_g(x)} [1 - \log(D(x))]$$

então  $p_g$  converge para  $p_{data}$ .

Demonstração:

Considere  $V(G, D) = U(p_g, D)$ . Assim, para um  $D$  fixo,  $U$  é função de  $p_g$ . Note que  $U(p_g, D)$  é convexo, pois

$$U(p_g, D) = \mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{x \sim p_g(x)} [1 - \log(D(x))] \therefore$$

$$\begin{aligned} U(\alpha p + (1 - \alpha)q, D) &= \int_x \alpha \cdot p(x) \log(D(x)) + (1 - \alpha)q(x) \log(1 - D(x)) dx \\ &= \alpha \mathbb{E}_{x \sim p(x)} [\log(D(x))] + (1 - \alpha) \mathbb{E}_{x \sim q(x)} [1 - \log(D(x))] \\ &= \alpha U(p, D) + (1 - \alpha)U(q, D) \end{aligned}$$

## Formalização Teórica

**Proposição 2.** Se G e D tiverem capacidade suficiente, e, em cada passo do Algoritmo 1, o discriminador atingir o seu ótimo dado G com  $p_g$  sendo atualizado para melhorar o critério

$$\mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{x \sim p_g(x)} [1 - \log(D(x))]$$

então  $p_g$  converge para  $p_{data}$ .

*Demonstração:*

---

Como  $U(p_g, D)$  é uma função convexa, podemos utilizar um algoritmo de descida de gradiente para atingir o seu mínimo no ponto onde esse gradiente é igual a zero, e que como provado no **Teorema 1**, é um mínimo global.  $\square$

# Formalização Teórica

**Proposição 2.** Se  $G$  e  $D$  tiverem capacidade suficiente, e, em cada passo do Algoritmo 1, o discriminador atingir o seu ótimo dado  $G$  com  $p_g$  sendo atualizado para melhorar o critério

$$\mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{x \sim p_g(x)} [1 - \log(D(x))]$$

então  $p_g$  converge para  $p_{data}$ .

Demonstração:

---

Como  $U(p_g, D)$  é um função convexa, podemos utilizar um algoritmo de descida de gradiente para atingir o seu mínimo no ponto onde esse gradiente é igual a zero, e que como provado no **Teorema 1**, é um mínimo global.  $\square$

Na prática, a GAN otimiza os parâmetros  $\theta_g$  invés de  $p_g$ , então a prova não se aplica, já que o *multilayer perceptron* aproxima um subconjunto da família de  $p_g$ .

# Implementações e Desafios

---

## Desafios - Dissipação dos Gradientes

**Dissipação dos gradientes:** Pesquisas indicam que se  $D$  for muito bom, então o treinamento de  $G$  pode falhar por dissipação dos gradientes. Ou seja, sob certas condições, foi provado que

$$\lim_{||D - D^*|| \rightarrow 0} \nabla_\theta \mathbb{E}_{z \sim p(z)} [\log (1 - D(G_\theta(z))) = 0]$$

# Desafios - Dissipação dos Gradientes

## Solução:

(i) Alterar a função objetivo do gerador para  $C(G) = \max - \log(D^*(G(z)))$ .

Dessa forma foi provado que

$$\mathbb{E}_{z \sim p(z)} [-\nabla_{\theta} \log D^*(G_{\theta}(z))|_{\theta=\theta_0}] = \nabla_{\theta} [KL(p_{g_{\theta}} || p_{data}) - 2JSD(p_{g_{\theta}} || p_{data})]|_{\theta=\theta_0}$$

Note que:

- JSD's estão com sinais opostos. Significa que estão pressionando para que as distribuições sejam diferentes, o que parece ser uma falha na atualização.
- $KL(p_{g_{\theta}} || p_{data})$  não é equivalente ao da máxima verossimilhança.

Isso explica porque os GANs (quando estabilizados) criam amostras de boa aparência e justifica porque os GANs sofrem com uma quantidade grande de "Mode Dropping".

# Desafios - Dissipação dos Gradientes

## Solução:

(ii) (WGAN) - Utilizar métrica Wasserstein, definida abaixo, como função objetivo.

$$W(p_{data}, p_g) = \inf_{\gamma \in \Gamma(p_{data}, p_g)} \mathbb{E}_{(x,y) \sim \gamma} [|x - y|]$$

# Desafios - Dissipação dos Gradientes

## Solução:

(ii) (WGAN) - Utilizar métrica Wasserstein, definida abaixo, como função objetivo.

$$W(p_{data}, p_g) = \inf_{\gamma \in \Gamma(p_{data}, p_g)} \mathbb{E}_{(x,y) \sim \gamma} [|x - y|]$$

Porém, implementar essa definição é impraticável. Em seu lugar, através da dualidade Kartrovich - Rubinstein, podemos aproxima-la por

$$W(p_{data}, p_g) = \sup_{\|f\|_L \leq 1} \frac{1}{K} [\mathbb{E}_{x \sim p_{data}} [f(x)] - \mathbb{E}_{x \sim p_g} [f(x)]]$$

# Desafios - Dissipação dos Gradientes

## Solução:

(ii) (WGAN) - Utilizar métrica Wasserstein, definida abaixo, como função objetivo.

$$W(p_{data}, p_g) = \inf_{\gamma \in \Gamma(p_{data}, p_g)} \mathbb{E}_{(x,y) \sim \gamma} [|x - y|]$$

Porém, implementar essa definição é impraticável. Em seu lugar, através da dualidade Kartrovich - Rubinstein, podemos aproxima-la por

$$W(p_{data}, p_g) = \sup_{\|f\|_L \leq 1} \frac{1}{K} (\mathbb{E}_x p_{data}[f(x)] - \mathbb{E}_x p_g[f(x)])$$

Suponha que  $f$  pertença à família de funções K-Lipschitz  $\{f_w\}_{w \in W}$ . Temos então

$$W(p_{data}, p_g) = \max_{w \in W} \mathbb{E}_x p_{data}[f_w(x)] - \mathbb{E}_z p_z [f_w(G(z))], \text{ tal que } \|f_w\|_L \leq K$$

# Desafios - Dissipação dos Gradientes

## Solução:

(ii) (WGAN) - Utilizar métrica Wasserstein, definida abaixo, como função objetivo.

$$W(p_{data}, p_g) = \inf_{\gamma \in \Gamma(p_{data}, p_g)} \mathbb{E}_{(x,y) \sim \gamma} [|x - y|]$$

Porém, implementar essa definição é impraticável. Em seu lugar, através da dualidade Kartrovich - Rubinstein, podemos aproxima-la por

$$W(p_{data}, p_g) = \sup_{\|f\|_L \leq 1} \frac{1}{K} [\mathbb{E}_x p_{data}[f(x)] - \mathbb{E}_x p_g[f(x)]]$$

Suponha que  $f$  pertença à família de funções K-Lipschitz  $\{f_w\}_{w \in W}$ . Temos então

$$W(p_{data}, p_g) = \max_{w \in W} \mathbb{E}_x p_{data}[f_w(x)] - \mathbb{E}_z p_z [f_w(G(z))], \text{ tal que } \|f_w\|_L \leq K$$

O Discriminador é usado para "aprender"  $w$  que permita encontrar uma boa  $f_w$ . A função  $W$  então se torna.

$$W(p_{data}, p_g) = \max_{w \in W} \mathbb{E}_x p_{data}[D_w(x)] - \mathbb{E}_z p_z [D_w(G(z))], \text{ tal que } \|D_w\|_L \leq K$$

## Desafios - Dissipação dos Gradientes

Dessa forma, D deixa de ser um Discriminador (com saída [0,1]) que fica tentando descobrir quais amostras são falsas ou verdadeiras e se transforma em um Crítico (com saída  $\geq 0$ ) que é treinado para aprender funções contínuas K-Lipschitz que ajudarão a computar a métrica Wasserstein. A função objetivo da nova GAN (WGAN) passa a ser então.

$$V(G, D) = \min_{\theta} \max_{w \in W} \mathbb{E}_{x \sim p_{data}} [D_w(x)] - \mathbb{E}_{z \sim p_z} [D_w(G_{\theta}(z))], \text{ tal que } \|D_w\|_L \leq K$$

Uma vez que a metrica Wasserstein tem um comportamento suave ela elimina o problema da dissipação do gradiente.

## Desafios - "Mode Colapse"

**"Mode Colapse"** Normalmente, você deseja que sua GAN produza uma ampla variedade de saídas. No entanto, se um gerador produz uma saída especialmente plausível, ele pode aprender a produzir apenas essa saída.

Neste caso, a melhor estratégia do discriminador é aprender a sempre rejeitar essa saída. Mas se a próxima geração de discriminador ficar presa no mínimo local e não encontrar a melhor estratégia, será muito fácil para a próxima iteração do gerador encontrar a saída mais plausível para o atual discriminador.

### Solução:

- (i) WGAN - Se o discriminador não ficar preso num mínimo local, então ele aprende a rejeitar as saídas estabilizadas do gerador.
- (ii) Unrolled GAN - Unrolled GAN usam uma função objetivo para o gerador que incorpora não apenas as classificações atuais do discriminador, mas também os resultados de futuras versões do discriminador. Portanto, o gerador não pode otimizar demais para um único discriminador.

# Desafios - Falha de Convergência

**"Falha de Convergência"** GAN's frequentemente não convergem.

**Solução:**

- (i) Adicionar ruído nas entradas do discriminador. Sabemos que o suporte de  $p_{data}$  e  $p_g$  são disjuntos num espaço de grande dimensão. Para artificialmente espalhar a distribuição e aumentar a chance das duas distribuições se sobrepor, uma solução é adicionar ruído nas entradas do discriminador D.
- (ii) Penalizar pesos do discriminador.

## References i

---

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.