

# Generative Adversarial Networks (GANs)

Ion Goodfellow, 2014

- Dos redes compitiendo entre sí (Adversarial)
- 1° Generador (datos sintéticos)
- 2° Discriminador (clasificar reales vs sintéticos)

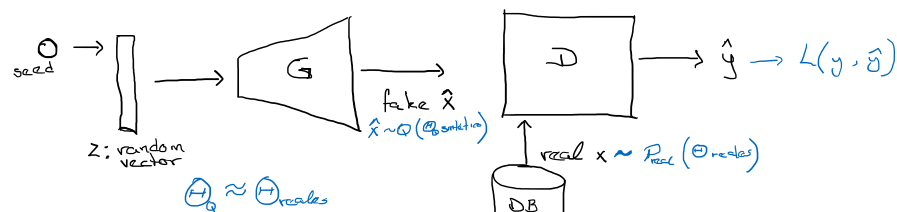
## Generadores:

- Muestra números aleatorios: vector  $z$
- Con layers ocultos mapea  $z$  en datos sintéticos (genera fakes)

## Discriminador:

- Se entrena para no dejarse engañar

$$x_{\text{real}} \sim \mathcal{R}(\Theta_{\text{real}})$$



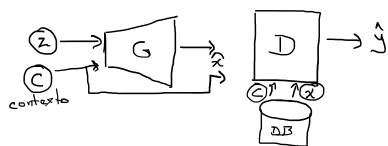
## Optimización

- Fakes, maximizar  $\log p(\hat{y}=0|x)$
- Real, maximizar  $\log p(\hat{y}=1|x)$

$$\begin{aligned} \mathcal{L} &= E[\log D(x)] + E[\log (1 - D(G(z)))] \\ &= BCE(D(x), 1) + BCE(D(G(z)), 0) \quad ; \quad BCE: \text{binary cross entropy} \\ &= D_{JS} : \text{Divergence Jensen-Shannon} \end{aligned}$$

## Variantes

- **Conditional GAN (cGAN)**
- Incluimos información sobre contexto (ej. clases)



- **Wasserstein GAN (WGAN)**

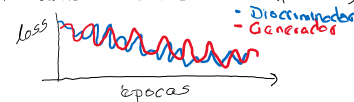
- Optimización unificada
- Sub modelos ya no compiten en términos de optimización, sino que colaboran
- la pérdida comienza a parecerse a un curva monótona
- Implementación:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N y_i \cdot \hat{y}_i \quad ; \quad N: \text{tamaño de la muestra}$$

- 1- Entreno D con ejemplos reales - etiqueta 1 y ejemplo fake - etiqueta 0. Backprop modifica los pesos de D únicamente.

- 2- Entreno G con ejemplos fake - etiqueta 1 usando el error del discriminador. Backprop bypasses D y modifica los pesos de G

- Entrenamiento adversarial (competitivo)



Paréntesis: Divergence Kullback-Leibler

- Divergencia entre dos pdf's  $P$  y  $Q$
- Notión del costo a pagar por suponer que nuestros datos son generados por  $P$ , cuando realmente son generados por  $Q$

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \log \left( \frac{P(x)}{Q(x)} \right)$$



Divergence Jensen-Shannon

$$D_{JS}(P||Q) = \frac{1}{2} D_{KL}(P||M) + \frac{1}{2} D_{KL}(Q||M)$$

donde  $M = \frac{1}{2}(P+Q)$

## EN MEMORIA

