

# Parcial de IA # 1

Sofía Escalante Escobar

Punto ①

- 2.1 Presente el modelo, función de costo y optimización por gradiente automático, de los autoencoderes regularizados, autoencoders varacionales y las redes generativas adversarias (GANs).

## Autoencoder regularizado

Input:  $x \in \mathbb{R}^{N \times M}$

Output:  $x' \in \mathbb{R}^{N \times M}$

Para cada  $x'_n \in \mathbb{R}^M$  en  $X'$

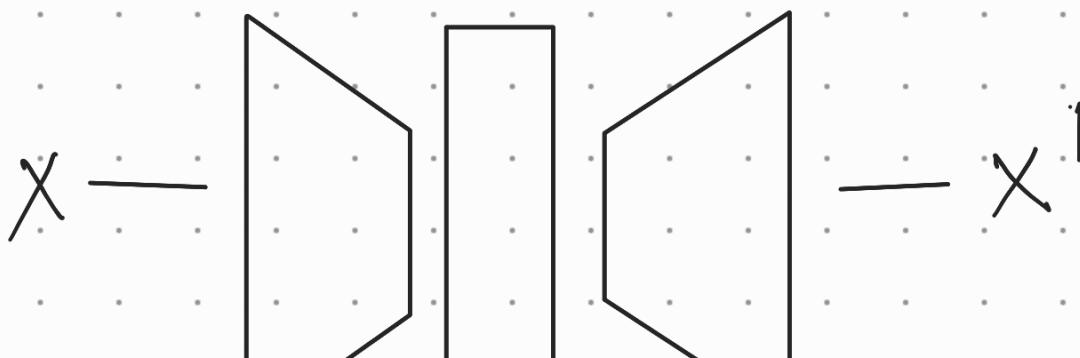
$$x'_n = f_{\text{dec}}(z) \rightarrow \text{decoder}$$

$$z_n = f_{\text{enc}}(x_n) \rightarrow \text{encoder}$$

$$f_{\text{enc}}(x_n) = s_{\text{enc}}(W_{\text{enc}} x_n + b_{\text{enc}})$$

$$f_{\text{dec}}(z_n) = s_{\text{dec}}(W_{\text{dec}} z_n + b_{\text{dec}})$$

$z^j$   
vector de representación  
en el espacio latente



encoder ( $f_{enc}$ ) espacio latente ( $z$ ) decoder ( $f_{dec}$ )

dónde

- $J \ll M$
- $W_{enc} \in \mathbb{R}^{J \times M}$ ,  $b \in \mathbb{R}^J$ ,  $W_{dec} \in \mathbb{R}^{M \times J}$ ,  $b \in \mathbb{R}^M$

### funciones de activación

$$S_{enc}: \mathbb{R}^J \rightarrow \mathbb{R}^J \quad y \quad S_{dec}: \mathbb{R}^M \rightarrow \mathbb{R}^M$$

### función de costo

una función de regularización ( $R$ )

$$\mathbb{E}_{x_n} \{ \mathcal{L}(x_n, x'_n) \} + \lambda R(w)$$

$$= \mathbb{E}_{x_n} \{ \mathcal{L}(x_n, f_{dec}(f_{enc}(x_n)) \} + \lambda R(w)$$

- donde  $\lambda \in \mathbb{R}_+$  y el los  $\mathcal{L}(\cdot)$  puede ser cualquier medida en  $\mathbb{R}^m$
- $R$  puede ser la norma  $L_1, L_2$ , etc.

### función de optimización

$$\hat{\theta} = \arg \min_{\theta} \mathbb{E}_{x_n} \{ \mathcal{L}(x_n, f_{dec}(f_{enc}(x_n)) \} + \lambda R(w)$$

Si tomamos:

$\mathcal{L} \rightarrow$  crossentropy

$R \rightarrow$  norma  $L_2$

Obtenemos:

$$\hat{\theta} = \arg \min_{\theta} \sum_{n=1}^N \sum_{m=1}^M x_{n-m} \log(x'_{n,m}) + \lambda \|w\|_2^2$$

donde:  $\theta = \{ w_{enc}, w_{dec}, b_{enc}, b_{dec} \}$

Parámetros

## Autoencoder Variacional

$$z \sim q(\phi(z|x)) = \mathcal{N}(z; \mu(x), \sigma^2(x))$$

distribución gaussiana

desviación estandar

$x' = f_{dec}(z)$  el decodificador mapea  $z$  de vuelta a una reconstrucción  $x'$

## Modelo:

Input:  $x \in \mathbb{R}^{N \times M}$ ,  $P(z|x_n, \theta)^{(prior)}$

Output:  $x' \in \mathbb{R}^{N \times M}$

función de mito

$$D_{KL}(q(z|x; \phi) || P(z|x; \theta)) + \mathcal{L}_{\text{rec}}(x, x')$$

$$\mathcal{L}_{\text{VAE}}(x, x') =$$

$$- E_{q(\Phi(z|x))} [\log P_\theta(x|z) + D_{KL}(q(\Phi(z|x)) || P(z)) + \log(P(x)) + \mathcal{L}_{\text{rec}}(x, x')]$$

- $\mathcal{L}_{\text{rec}}$  es un loss de reconstrucción en el prior
- $D_{KL} = D_{KL}(q_\phi(z|x) || P(z|x)) = \int_z q_\phi(z|x) \log\left(\frac{q(z|x)}{P(z|x)}\right) dz$
- $q(z|x, \phi)$  es la aproximación del posterior

## Función de optimización

necesitamos que  $\log(P(x))$  sea constante:

$$\phi^* = \arg \min_{\phi} \mathcal{L}_{\text{rec}}(x, x') + D_{KL}(q(z|x, \phi) || P(z)) - E_{q(z|x, \phi)} [\log P(x|z)]$$

$$\phi^* = \arg \min_{\phi} \sum_{n=1}^N \sum_{m=1}^M x_{n,m} \log(x'_{n,m}) + \sum_{n=1}^N D_{KL}(q(x|z, \phi) || P(x|z, \theta))$$

## Redes generativas adversariales (GAN's)

Input:  $x \in \mathbb{R}^{N \times P}$

Output:  $y \in [0,1]^{N+M}$

generador:

$$\phi: \mathbb{R}^Q \rightarrow \mathbb{R}^P$$

$$z \rightarrow x'$$

$$\text{donde: } x' = \phi(z; \theta_0)$$

$z \sim p_t(z) \rightarrow \text{distribución prior}$

$$p_0(x', t)$$

entonces

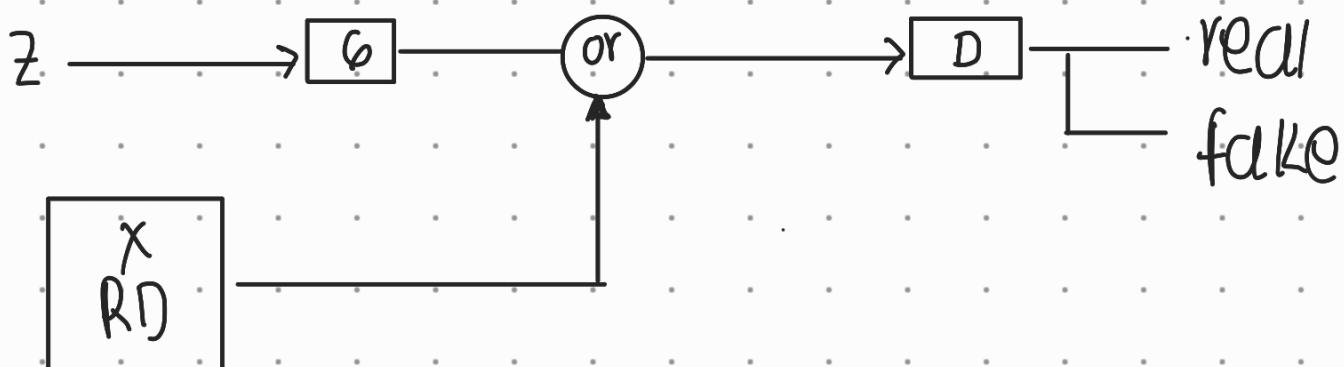
$$x' = \{x' | x' = \phi(z) \text{ para } z \in Z\}$$

discriminador

$$D: \mathbb{R}^P \rightarrow [0, 1] \quad \text{donde}$$

$$x \rightarrow y \quad y = \delta(x; \theta_D)$$

$$D(x) = \begin{cases} 1 & \text{si } x \in X \\ 0 & \text{si } x' \in X' \end{cases}$$



generador

$$\mathcal{L}_G = -\mathbb{E}_{z \sim p_t(z)} [\log D(\phi(z))]$$

## Discriminador

$$\mathcal{L}_D = -\mathbb{E}_{z \sim D(z)} \{1 - \log(D(\phi(x))\} - \mathbb{E}_{x \sim P_{\text{real}}} \{\log(D(x))\}$$

## función de optimización

$$\theta_\phi^*, \theta_D = \min_{\theta_D} \max_{\theta_\phi} \mathcal{L}_D(\theta_D) - \mathcal{L}_\phi(\theta_\phi)$$

Punto ④

## Vector Quantized GAN (VQGAN)

1. Train (Image)  $\rightarrow X \in \mathbb{R}^{H \times W \times 3}$

$$f = \mathbb{R}^{H \times W \times 3}$$

donde

$$E: \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H_2 \times W_2 \times N_2 \text{ (encoder)}}$$

$$E(X) = Z = \{Z_{ij}\}_{i=1, j=1}^{H_2, W_2} \subset \mathbb{R}^{N_2}$$

$$q: \mathbb{R}^{H_2 \times W_2 \times N_2} \rightarrow \mathbb{R}^{H_2 > W_2 > N_2}$$

definido por:

$$x = f(x) = (\ell \circ q \circ \epsilon)(x) \in \mathbb{Z}_q$$

$$= q(z) \left( \operatorname{argmin}_{z_k \in \mathcal{Z}} \|z_{ij} - z_k\| \right) \in \mathbb{R}$$

$\mathbb{H}_Z \times \mathbb{W}_Z \times 3$

$$\ell(z_q) = \ell(q(z)) = x' \in \mathbb{R}^{H \times W \times 3}$$

$$s_{ij} = \operatorname{argmin}_k \|z_{ij} - z_k\| \quad \forall i \in \{1, \dots, H_Z\},$$

$\forall j \in \{1, \dots, W_Z\}$

→ representa la posición en el codebook. Sea

$$S = \{s_{ij}\}_{\substack{i=1, \\ j=1}}^{H_Z, W_Z} \quad \text{los índices del}$$

codebook; definimos una red transformer para

$$T(S) = S' = (\operatorname{argmax} s'_{1,1}, \operatorname{argmax} s'_{1,2}, \dots, \operatorname{argmax} s'_{H_Z, W_Z})$$

LOSS

$$Q^* = \arg \min_{E, \theta, z} \max_D \mathbb{E}_{x \sim p(x)} [L_{v_Q}(E, \theta, z) + \lambda L_{GAN}(E, \theta, z, D)]$$

$L_{total}$

donde

$$Q^* = \{E^*, \theta^*, z^*, D^*\}$$

pero destinado al discriminador

$$\mathcal{L}_{VQ}(E, \phi, z) = \|x - x'\|_2^2 + \|sg(E(x)) - z_q\|_2^2 \\ + \|sg(z_q) - E(x)\|_2^2$$

sg - Stop gradient que aseguran que ciertos terminos del loss, los gradientes no influyan hacia el encoder

$$\mathcal{L}_{GAN}(E, \phi, z, D) = \log(D(x)) + \log(1 - D(x'))$$

loss → para modelar  
(transformer)

$$\mathcal{L}_T = \mathbb{E}_{x \sim p(x)} \{-\log p(s)\}$$

Generalización de swap face. consiste en el intercambio de una imagen A de dimensiones  $H^A \times W^A \times 3$  sobre otra imagen B  $\in \mathbb{R}^{H_B \times W_B \times 3}$ . Para aplicar este proceso en un video de N fotogramas se repetirá este método tantas veces como sea necesario

## • Clasificación y cuantificación de los imágenes

$$Z_q^A = q(Z^A) = q(E(A))$$

$$Z_q^B = q(Z^B) = q(E(B))$$

## • Intercambio de códigos latentes

$$Z_q^A \otimes Z_q^B = Z_{\text{swap}}$$

## • Decodificación y reconstrucción

$$X'_{\text{swap}} = g(Z_{\text{swap}})$$