Variational Autoencoders (VAEs)

Representaciones probabilísticas y modelos generativos

Cesar Garcia

2025

Introducción

- Recordatorio: autoencoders aprenden a **reconstruir**
- Problema: el espacio latente **no está organizado para muestrear**
- Solución (VAE): hacer el latente **probabilístico y regularizado**
- Objetivo: **generar** nuevos ejemplos de forma controlada

¿Qué necesitamos agregar para que el espacio latente sea "muestreable"?

Por qué un AE estándar no es generativo

La limitación

En un autoencoder estándar:

- el encoder produce un punto z
- no hay una distribución explícita sobre z
- no sabemos qué regiones del espacio latente son "válidas"

Consecuencia:

- muestrear z al azar suele producir basura al decodificar

¿Qué significa que el espacio latente tenga "agujeros"?

Idea central de un VAE

Latente como variable aleatoria

En un VAE:

- el encoder no produce un solo z
- produce parámetros de una distribución:
  - media $\mu(x)$
  - varianza (o $\log \sigma^2(x)$)

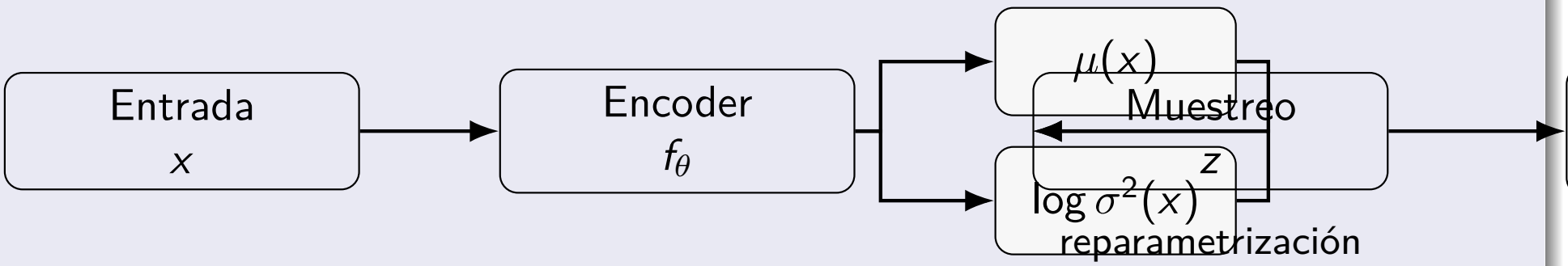
Luego:

- muestreamos $z \sim q_\theta(z \mid x)$
- el decoder reconstruye desde ese z

¿Por qué sería útil introducir "incertidumbre" en el latente?

Arquitectura VAE

Encoder produce μ y $\log \sigma^2$



¿Qué cambia respecto a un autoencoder "determinista"?

Dos objetivos a la vez

Reconstruir y regularizar

El VAE optimiza dos cosas:

1) **Reconstrucción**: que $\hat{x}$ se parezca a x
2) **Regularización**: que el latente se parezca a un prior simple (normal)

Intuición:

- "reconstruye bien"
- "pero no uses un latente raro o caótico"

¿Por qué querríamos que el latente se parezca a una Normal?

KL Divergence (intuición)

Distancia entre distribuciones

El término KL mide qué tan distinta es:

$$q_\theta(z \mid x) \quad \text{vs} \quad p(z)$$

Con $p(z) = \mathcal{N}(0, I)$ típicamente.
Interpretación:

- penaliza "latentes demasiado específicos"
- evita memorizar (en el latente)

¿Qué pasa si el KL es demasiado grande o demasiado pequeño?

La pérdida total (ELBO)

Forma práctica

En entrenamiento usamos (forma típica):

$$\mathcal{L} = \underbrace{\mathcal{L}_{rec}(x, \hat{x})}_{\text{reconstrucción}} + \beta \underbrace{\mathrm{KL}(q_\theta(z \mid x) \,\|\, p(z))}_{\text{regularización}}$$

β controla el balance (*beta-VAE*).

¿Qué efecto tiene aumentar β?

El truco de reparametrización

Problema

Queremos hacer backprop a través de un muestreo.
Pero muestrear es "no diferenciable" si lo hacemos directo.

Solución

Reescribimos:

$$z = \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

Así el azar vive en ϵ y el resto es diferenciable.

¿Por qué esta reescritura sí permite gradientes?

Muestreo y generación

Ahora sí podemos generar

Una vez entrenado el VAE:

- muestreamos $z \sim \mathcal{N}(0, I)$
- decodificamos $\hat{x} = g_\phi(z)$

Además:

- interpolar en z suele producir transiciones suaves

¿Qué condición hizo esto posible en un VAE, pero no en un AE?

Fallos típicos: posterior collapse

Qué es

A veces el modelo aprende:

- $q(z \mid x) \approx \mathcal{N}(0, I)$ para todo x
- el decoder ignora z

Entonces:

- KL muy pequeño
- representación latente inútil

¿Cómo detectarías que el modelo está ignorando el latente?

Idea clave de la sesión

Generar requiere estructura

Un VAE convierte el espacio latente en:

- **continuo**
- **muestreable**
- **regularizado**
  No solo comprimimos: imponemos una geometría probabilística.

¿Qué ganamos al "forzar" estructura en el latente?