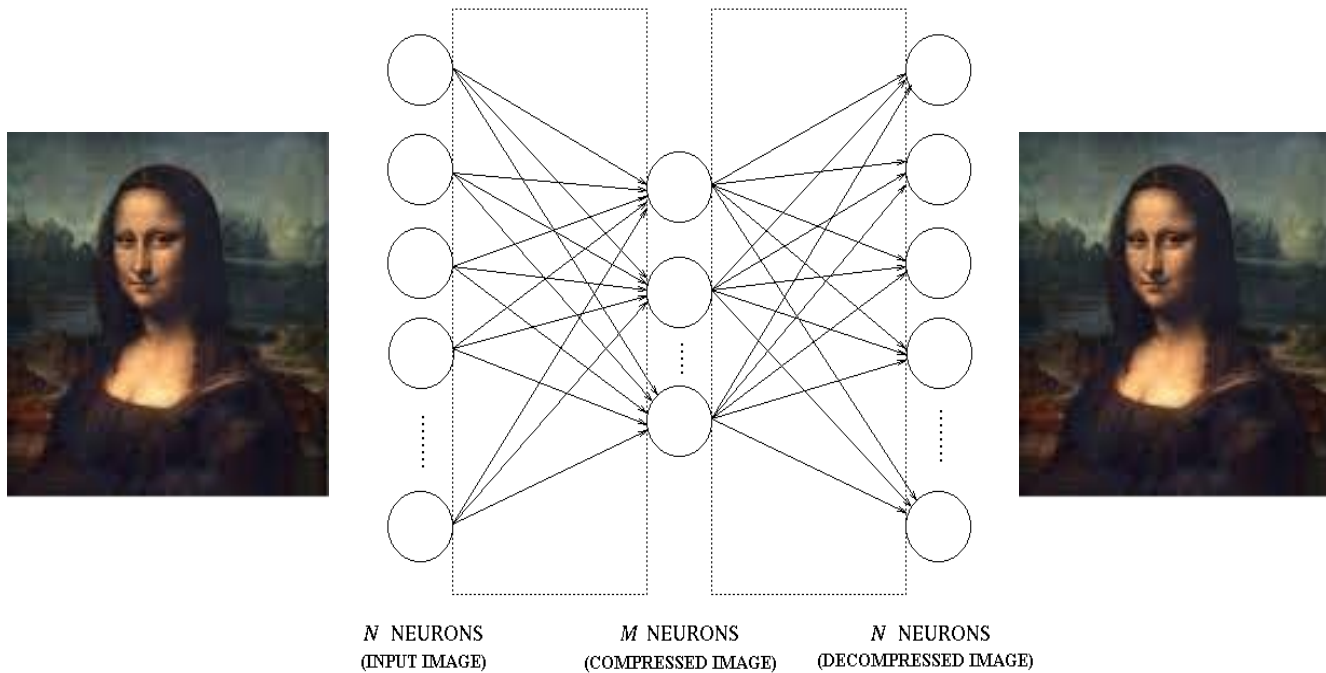


AUTOENCODERS

Primer Cuatrimestre 2025

Compresión de imágenes por encoder/decoder



Entrada=salida (autoasociador)

$M \ll N$

AUTOENCODERS

Objetivo: aprender una representación de un conjunto de datos de entrada, en forma robusta ante ruido y reconstruyendo lo más exactamente posible la entrada original

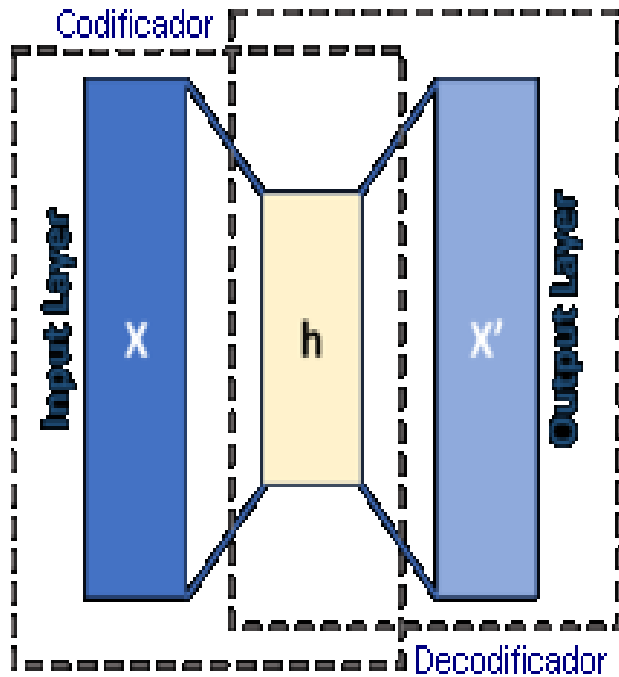
Dos propiedades deseables a balancear:

- Sensibilidad a las entradas → buena reconstrucción de la señal
- Insensibilidad (robustez) a ruido en las entradas → capacidad de generalizar
(= evitar overfitting)
- En general: reducción de dimensión (pero veremos que no siempre)
- Pero no sólo nos interesará la salida (reconstrucción) sino muy especialmente la codificación interna (capa oculta)

Si podemos concatenarlos o apilarlos (stack)

→ *estructura jerárquica con múltiples, progresivos niveles de abstracción.*

AUTOENCODER BÁSICO



p tamaño del conj. entrenamiento

N dimensión entrada / salida

M dimensión capa oculta

X entradas, X' salidas

ξ_μ patrones de entrenamiento ($\mu = 1, \dots, p$)

ζ_μ salidas esperadas ($= \xi_\mu$)

$$h = \text{enc}(X) = f(W_1 X)$$

$$X' = \text{dec}(h) = g(W_2 h)$$

$$E = \sum_{\mu} \|\xi_{\mu} - \text{dec}(\text{enc}(\xi_{\mu}))\|_2^2$$

Impone una restricción (“cuello de botella”) que fuerza al sistema a producir una representación comprimida de la entrada.

SHALLOW (\neq profundos) AUTOENCODERS con UNDERCOMPLETE CODING

Si f es enc y g es dec, queremos minimizar la función de pérdida:

$$L(x; g(f(x))) \text{ con } h = f(x) \text{ y } x' = g(h).$$

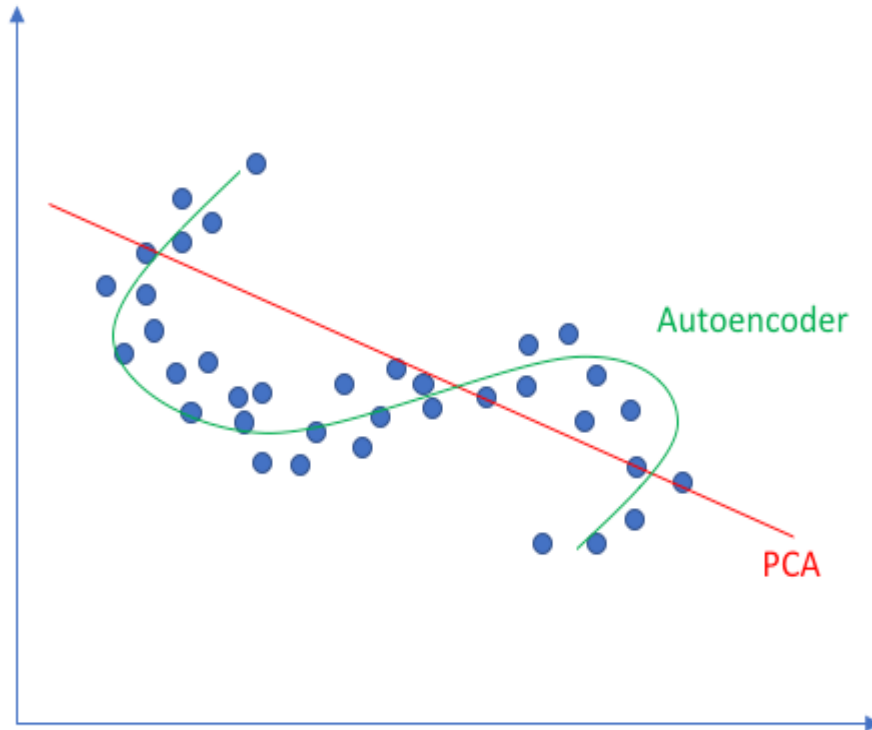
La única restricción es:

$$f: \mathbb{R}^N \rightarrow \mathbb{R}^M \text{ con } N > M \text{ (*undercomplete coding*)}$$

Si activaciones lineales y L error cuadrático medio
 $\rightarrow f(x)$ describe el mismo subespacio que PCA.

GUARDA: Autoencoder básico \neq PCA en general

- PCA *encuentra hiperplano de menor dimensión que describe los datos con mínimo error.*
 - Autoencoder (red neuronal) *encuentra superficie no lineal.*
- AE generaliza PCA



DENOISING AUTOENCODERS

Toman una entrada corrompida y tratan de recuperar la original, eliminando el ruido (denoising).

Asumimos:

- 1- Características de alto nivel son relativamente estables frente al ruido
- 2- Para ser efectivo, el modelo necesita extraer características (features) que representen la estructura de la distribución de la entrada

Entrenamiento:

- Corromper la entrada inicial ξ obteniendo ξ' mediante un mapeo estocástico $R(\xi' | \xi)$
- Mapear ξ' a la representación oculta en la forma habitual:

$$h = f_{\theta}(\xi')$$

- Reconstruir la salida x' en la forma habitual: $x' = g_{\theta'}(h)$

θ y θ' son los parámetros del modelo y son entrenados para minimizar el error medio de reconstrucción sobre todos los datos de entrenamiento, minimizando la diferencia entre x' y la entrada original sin ruido ξ .
(en cada iteración se presenta un ejemplo al que se le ha aplicado ruido)

SPARSE AUTOENCODERS (¿“Dispersos”?)

Útiles para tareas de clasificación.

$M \gg N$ pero sólo algunas (pocas) unidades ocultas pueden activarse al mismo tiempo (= competencia por los recursos sinápticos)

→ *fuerza al modelo a responder a las características específicas del patrón de entrada usado para entrenar*

Función de costo: $L(x, x') + R(h)$

R penaliza la activación simultánea de muchas unidades ocultas.

→ el modelo tenderá a usar pocas unidades en la reconstrucción de la entrada

Se busca capturar características más específicas de cada patrón.

R posibles:

- Vía divergencia de Kullback-Leibler:

$$\sum KL(\rho \parallel \rho'_j) = \sum \rho \log (\rho/\rho'_j) + (1-\rho) \log (1-\rho/1-\rho'_j)$$

suma sobre todas las unidades ocultas.

con ρ'_j activación media de la unidad oculta j sobre todos los elementos del conjunto de entrenamiento

ρ parámetro de una distribución Bernoulli.

→ ρ cercano a 0 penaliza la activación de muchas u. ocultas a la vez.

- $R(h) = \lambda \sum \|h_i\|$ (norma $L1$ o $L2$)

- Manualmente (k -sparse autoencoder), conservando las k mayores activaciones en módulo.

AUTOENCODERS CONTRACTIVOS

Objetivo: codificación robusta = para entradas similares, codificaciones similares

→ *forzar derivadas de las activaciones h a que sean pequeñas con respecto a las entradas.*

Función de costo:

$$L(x, x') + \lambda \sum_i \|J_x h_i\|$$

J en $\mathbb{R}^{m \times p}$ jacobiano de la matriz de activaciones, m tamaño de la capa oculta y p número de elementos en el dataset.

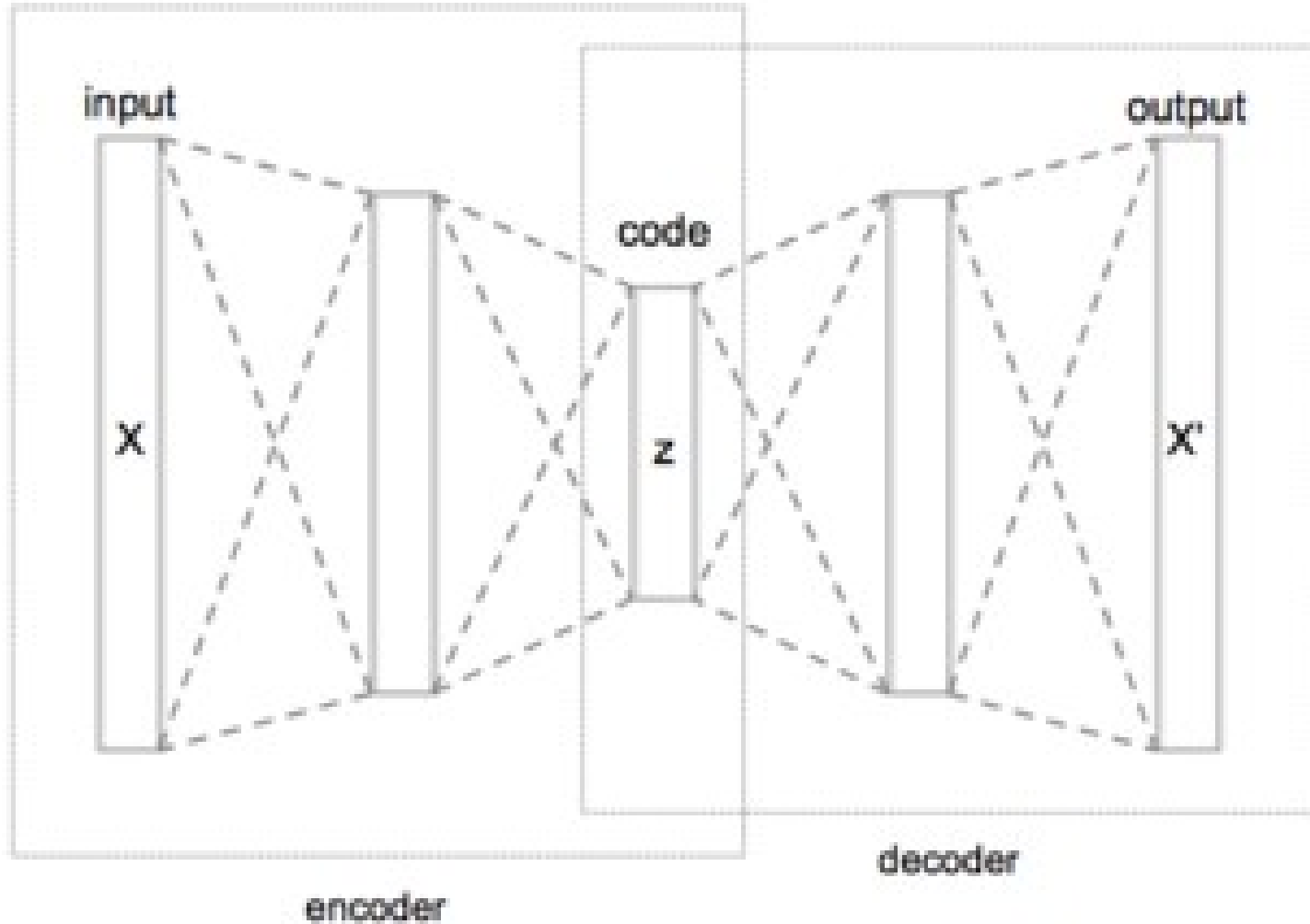
El modelo aprenderá a *contraer un entorno* de las entradas a un entorno más pequeño de las salidas.

NOTAR:

Denoising autoencoders → reconstrucción de la señal (decoder) robusta.

Contractive autoencoders → extracción de características (encoder) robusta.

AUTOENCODERS PROFUNDOS



Más eficientes, requieren menos datos de entrenamiento para aprender una función

ENTRENAMIENTO

De a una capa por vez

→ *salida de un AE = entrada del siguiente*

Luego: backfitting → *ajuste supervisado de toda la red*

→ puede usarse backpropagation

APLICACIONES

- *Reducción de dimensión*
- *Procesamiento de imágenes*
- *Supresión de ruido*
- *Detección de anomalías*
- *Clasificación*