Ejercicio: Obtener una factorización SVD de la matriz:

$$A = \begin{pmatrix} 1 & -1 \\ -2 & 2 \\ 2 & -2 \end{pmatrix}$$

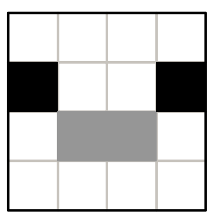
Ejercicio: Calcular una factorización SVD de la matriz:

$$A = \begin{pmatrix} 4 & -2 \\ 2 & -1 \\ 0 & 0 \end{pmatrix}$$

- Compresión de imágenes con SVD :

La factorización SVD puede utilizarse en compresión de imágenes digitales.

Una imagen digital (en blanco y negro) se representa con una matriz:

4 px 

4 px

$\xrightarrow{8 \text{ bits}}$

$L = 2^8 = 256$ niveles

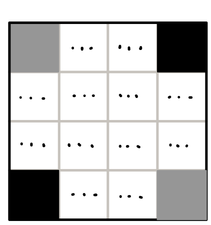
$$A = \begin{pmatrix} 255 & 255 & 255 & 255 \\ 0 & 255 & 255 & 0 \\ 255 & 127 & 127 & 255 \\ 255 & 255 & 255 & 255 \end{pmatrix}$$

Imagen digital en BN con resolución de 4x4 píxeles

matriz 4 x 4 con 256 niveles de gris (0 - 255)

0 ↑ negro, 255 ↑ blanco

Supongamos que tenemos una imagen digital de m x n píxeles:

m px 

n px

$$\longrightarrow \quad A = \Big(\qquad \Big)_{m \times n}$$

Podemos aplicar la factorización SVD a la matriz A de la imagen.

$$\boxed{A = U \cdot \Sigma \cdot V^t = \sigma_1 \vec{u}_1 \vec{v}_1^{\,t} + \cdots + \sigma_r \vec{u}_r \cdot \vec{v}_r^{\,t}}$$

$\lambda_{r+1} = \cdots = \lambda_n = 0$

Como los valores singulares están ordenados de mayor a menor:

$$\sigma_1 \geq \sigma_2 \geq \ldots \sigma_k \geq \ldots \geq \sigma_r \qquad (k \leq r)$$

Los k primeros términos de A son los más importantes porque son los que más suman a A.

Por tanto, podemos aproximar A usando los k 1^{os} términos:

$$\boxed{A \approx A_k = \sigma_1 \vec{u}_1 \vec{v}_1^{\,t} + \cdots + \sigma_k \vec{u}_k \vec{v}_k^{\,t} = U_k \cdot \Sigma_k V_k^t}$$

A_k es una versión comprimida de A $\rightarrow$ usa menos información.

$$\underset{m \times n}{A} = \underset{m \times m}{U} \cdot \underset{m \times n}{\Sigma} \cdot \underset{n \times n}{V^t}$$

$$\approx$$

$$\underset{m \times n}{A_K} = \underset{m \times K}{U_K} \cdot \underset{K \times K}{\Sigma_K} \cdot \underset{K \times n}{V_K^t}$$

Diagonal

$$\begin{cases} \text{Imagen original : } m \cdot n \text{ bytes} \\ \text{Imagen comprimida : } m \cdot K + K + K \cdot n = K(m+n+1) \text{ bytes} \end{cases}$$

Para que exista compresión : $K(m+n+1) < m \cdot n$

$$\boxed{K < \frac{m \cdot n}{m+n+1}}$$

$$K_{MÁX} = \text{floor}\left(\frac{m \cdot n}{m+n+1}\right)$$

redondear a la baja

$$\boxed{\% \text{ almacenamiento} = \frac{K(m+n+1)}{m \cdot n} \cdot 100}$$

Ejercicio: Mediante Python, utilizar la factorización SVD para comprimir la imagen "cameraman.tif" (256 x 256 píxeles) según los valores de K: 1, 10, 20, 30, ..., inferiores a $K_{MÁX}$. Mostrar en cada caso el % de almacenamiento de la imagen comprimida vs la original.