

# Taller 01

## Factorización de Matrices no Negativas (NMF)

Odin Eufracio

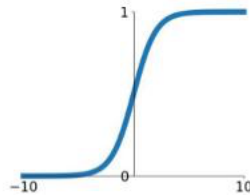
# Factorización de Matrices no Negativas (NMF)

# Función de activación ReLU: no-negatividad

## Activation Functions

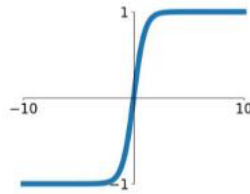
### Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



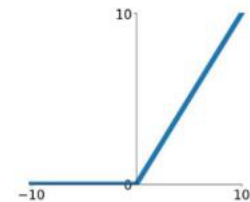
### tanh

$$\tanh(x)$$



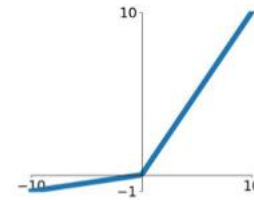
### ReLU

$$\max(0, x)$$



### Leaky ReLU

$$\max(0.1x, x)$$

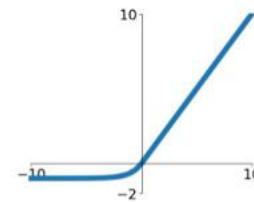


### Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

### ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

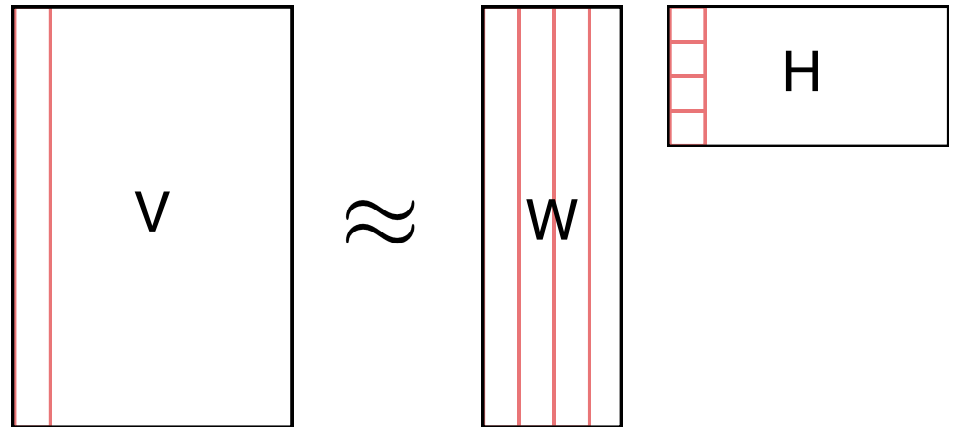


Efecto de no-negatividad? Es lo natural? Solo combinaciones aditivas?

# Factorización de Matrices no Negativas (NMF)

Método de reducción de dimensionalidad propuesto por Lee y Seung [1]

$$\begin{array}{ll} \min_{W,H} & F(V | WH^T) \\ \text{s. t.} & W \geq 0 \\ & H \geq 0 \end{array}$$

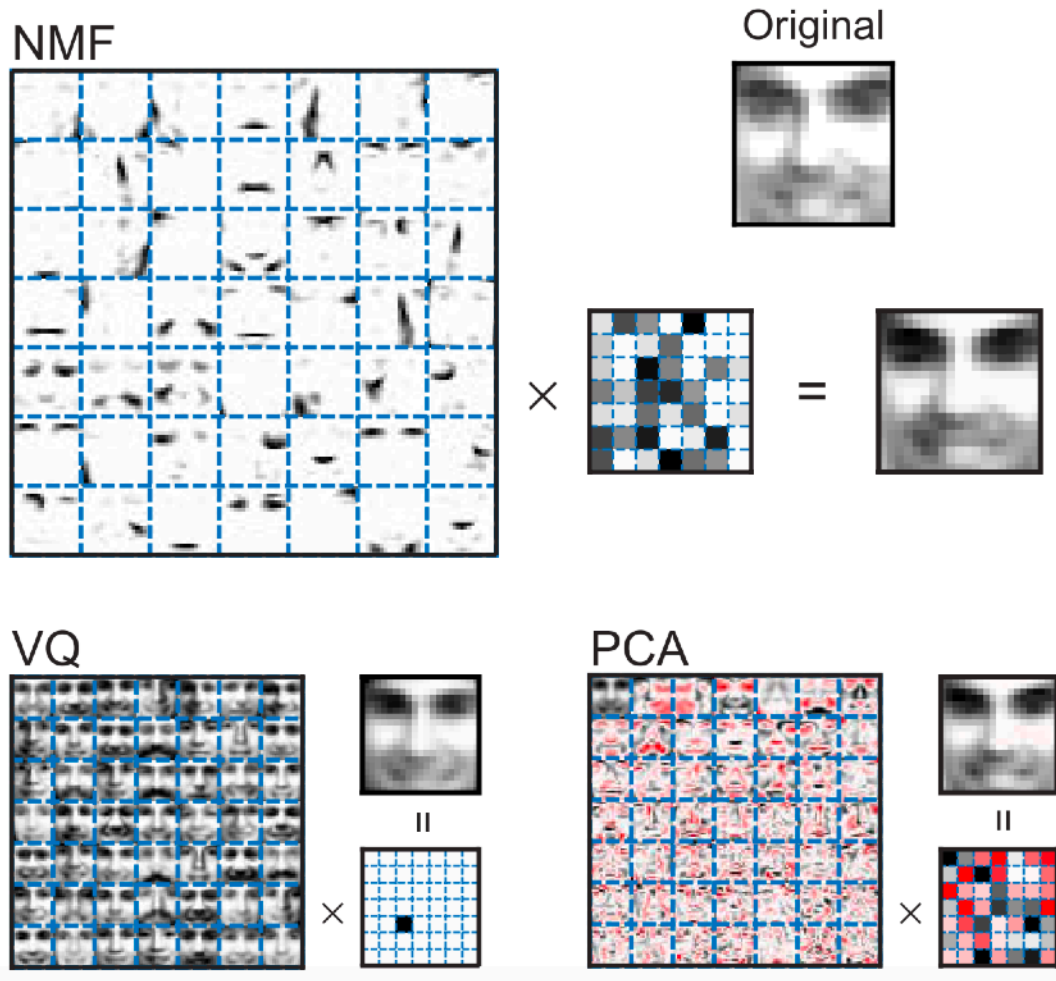


$$\mathbf{v}_j \approx \sum_k h_{kj} \mathbf{w}_k$$

La restricción de no negatividad promueve una representación de los datos basada en partes, lo cual permite “sumar” partes, no restar.

# Factorización de Matrices no Negativas (NMF)

Diferentes métodos de reducción de dimensionalidad: NMF, PCA, VQ [1]



## Características principales de NMF

- ✱ NMF produce de una forma **natural** una representación de los datos basada en partes [2].
- ✱ Las **partes** aprendidas representan rasgos de los datos [2].
- ✱ Los elementos de la **base W son ralas** porque extraen de los datos **rasgos localizados, con poco o nulo traslape** [1].
- ✱ El usuario debe **especificar el rango** de la factorización [2].

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[2] S. Squires, A. Prügel-Bennett, and M. Niranjana, "Rank Selection in Nonnegative Matrix Factorization using Minimum Description Length," *Neural Comput.*, vol. 29, no. 8, pp. 2164–2176, 2017.

# **Hierarchical Alternating Least Squares (HALS)**

# Hierarchical Alternating Least Squares (HALS)

En el contexto de **NMF**, uno de los algoritmos que ha mostrado un **gran desempeño** es **HALS**, el cual está implementado en el módulo de Python **scikit-learn**



<https://github.com/scikit-learn/scikit-learn/blob/1495f6924/sklearn/decomposition/nmf.py#L839>

HALS, en lugar de minimizar una o dos funciones objetivo, minimiza un conjunto de funciones de **costo locales**.

$$V^{(j)} = V - \sum_{r \neq j} w_r h_r^T = V - WH^T + w_j h_j^T,$$

Estas funciones de costo son locales en el sentido que las columnas y renglones de los factores  $W$  y  $H$  en NMF son procesados uno por uno en forma secuencial.

$$\min_{w_j} \quad D(w_j) = \frac{1}{2} \|V^{(j)} - w_j h_j^T\|_F^2 \quad \text{s. t. } w_j \geq 0$$

$$\min_{h_j} \quad D(h_j) = \frac{1}{2} \|V^{(j)} - w_j h_j^T\|_F^2 \quad \text{s. t. } h_j \geq 0$$

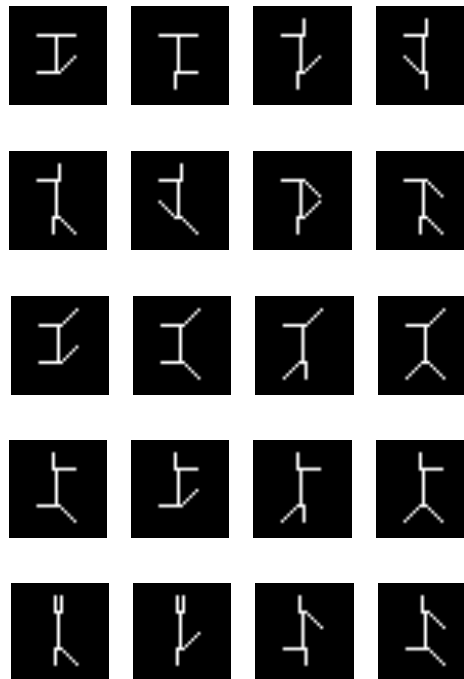


# Actividad 1

Deducir el algoritmo HALS descrito en

[https://www.researchgate.net/publication/220241471\\_Fast\\_Local\\_Algorithms\\_for\\_Large\\_Scale\\_Nonnegative\\_Matrix\\_and\\_Tensor\\_Factorizations](https://www.researchgate.net/publication/220241471_Fast_Local_Algorithms_for_Large_Scale_Nonnegative_Matrix_and_Tensor_Factorizations)

Implementar el algoritmo en el notebook **NMF\_HALS\_Swimmer.ipynb**



A. Cichocki and A.-H. Phan, “Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations,” *IEICE Trans.*, vol. 92-A, no. 3,

D. Donoho and V. Stodden, “When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts?,” in *Advances in Neural Information Processing Systems*, 2004, pp. 1141–1148.

## **Odin Eufracio**

Centro de Investigación en Matemáticas - CIMAT  
Jalisco SN, Mineral de Valenciana Gto. Gto.

Office: D307

Phone: (+52) 473 732 7155 ext. 4730

E-Mail: [odin.eufracio@cimat.mx](mailto:odin.eufracio@cimat.mx)