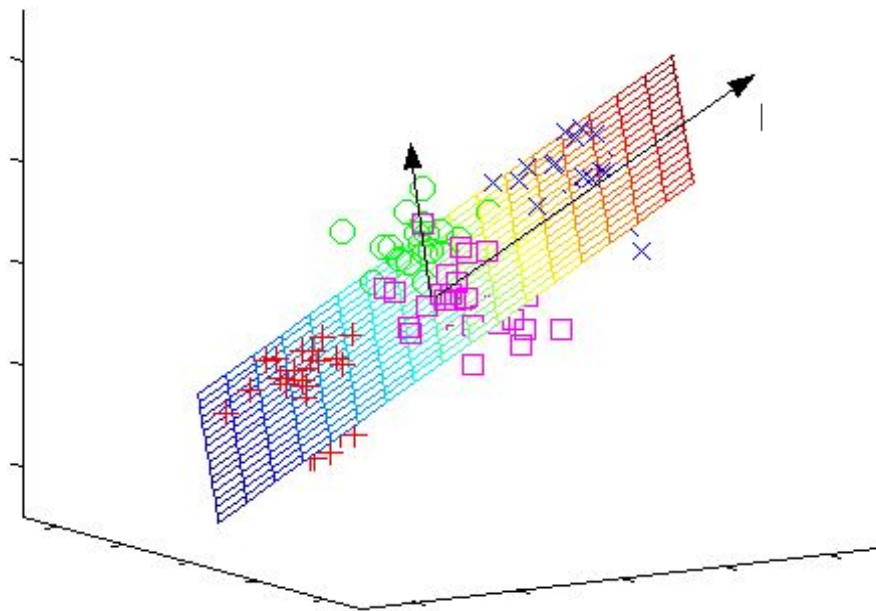


# Embeddings

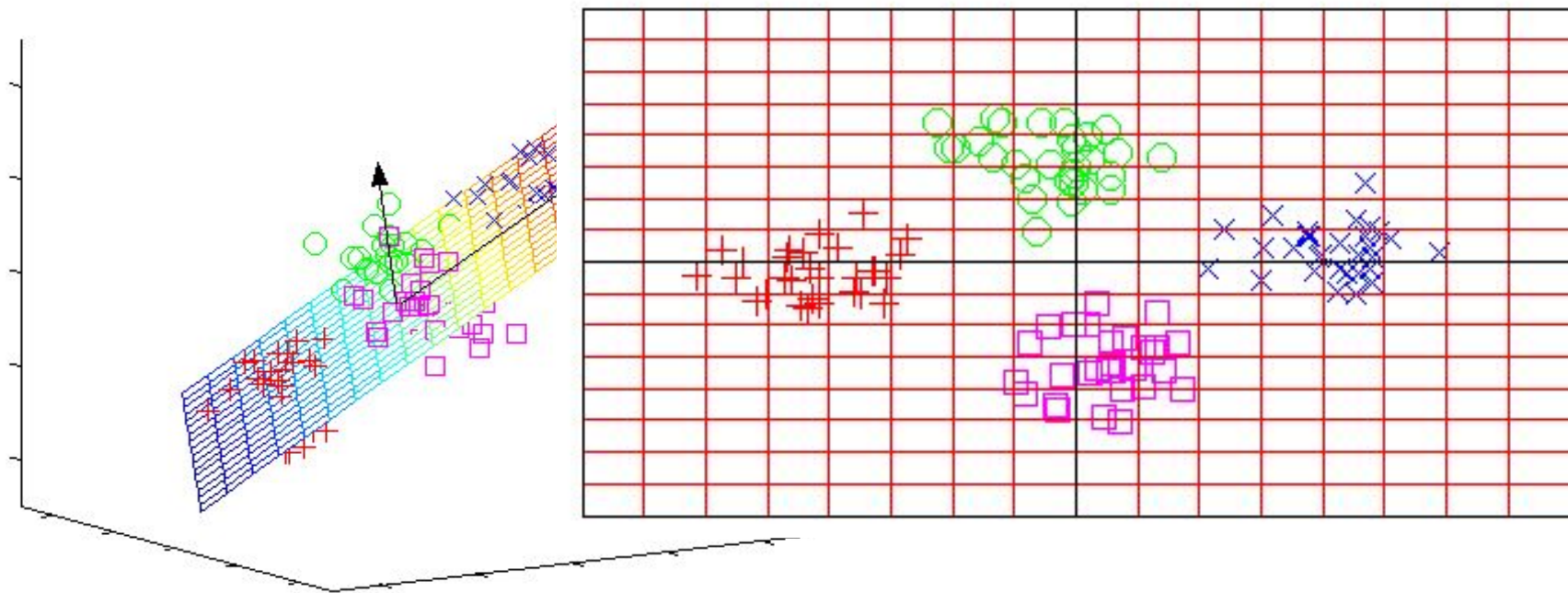


Diplomatura en Ciencia de Datos,  
Aprendizaje Automático y sus Aplicaciones  
FaMAF-UNC  
septiembre 2020

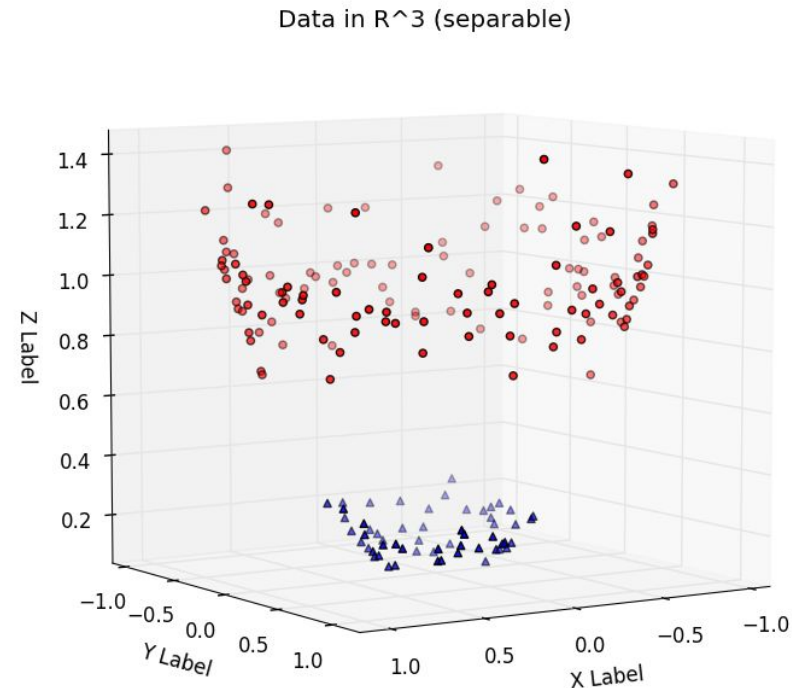
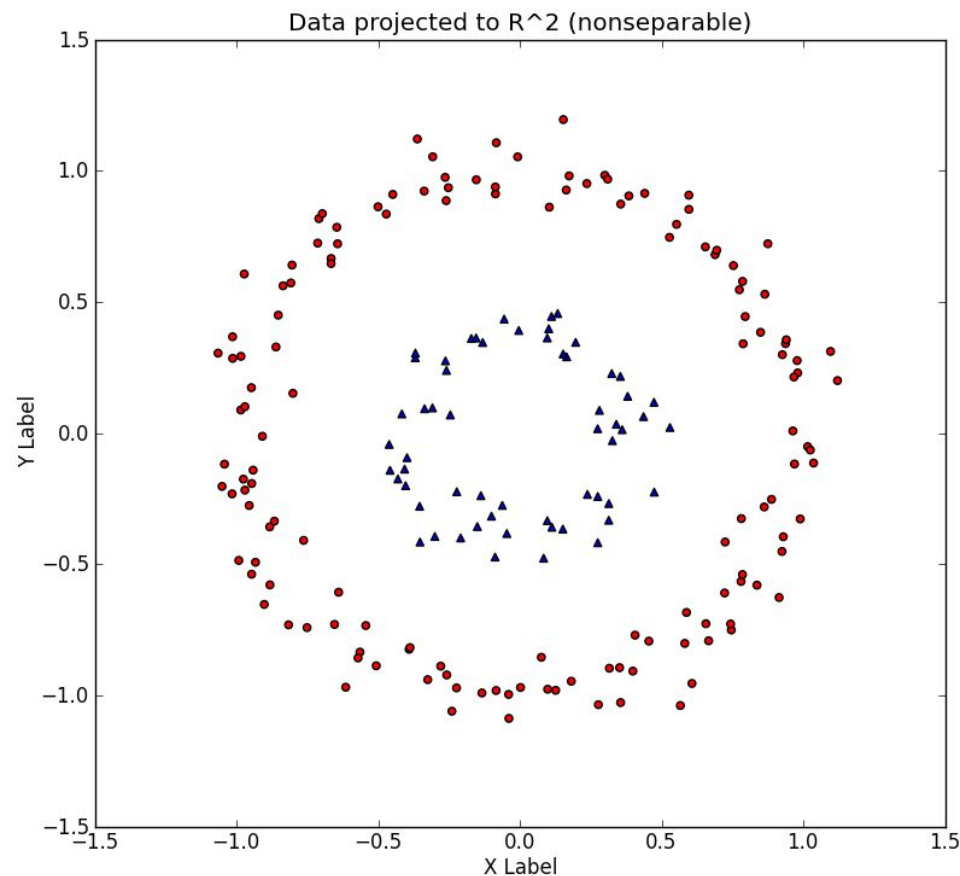
# Qué es un embedding (proyección)



# Qué es un embedding (proyección)



# Qué es un embedding (proyección)



# Qué es un embedding

Y un videíto sobre el kernel trick

<https://www.youtube.com/watch?v=3liCbRZPrZA>

# Tipos de embeddings

Técnicas populares dentro de la familia de los embeddings

- Selección de características → supervisado o no supervisado
- Agrupamiento de características → supervisado o no supervisado
- The kernel trick → un espacio de mayor dimensionalidad!
- Principal Component Analysis
- Latent Dirichlet Allocation
- t-sne
- Neural embeddings

# Objetivos de los embeddings

- En lugar de elegir un subconjunto de características, crear nuevas
- Sin tener en cuenta etiquetas de clase
- Proyectar a menos dimensiones preservando la mayor cantidad de información posible → minimizando el error cuadrado de reconstruir los datos originales

# Para qué sirven?

- Reducción de dimensionalidad
- Reducir overfitting
- Generalización
- Acercamiento a las causas latentes
- Reducir el tiempo en ingeniería de características
- Reducir el sesgo del científico



# Qué perdemos?

- Información
- Interpretabilidad

# Selección de Características

Reducción de dimensionalidad simplemente eliminando características

- Intuición: eliminamos ruido

Pero... la selección de características se hace en relación a una clase!

[https://scikit-learn.org/stable/modules/feature\\_selection.html](https://scikit-learn.org/stable/modules/feature_selection.html)

Cómo hacemos si no tenemos clases?

Aplicamos conocimiento de dominio!

- P.ej., en lenguaje natural:
  - eliminamos palabras poco frecuentes
  - eliminamos palabras muy frecuentes

# Selección de Características

Reducción de dimensionalidad simplemente eliminando características

- Intuición: eliminamos ruido

También tenemos métodos basados en varianza:

- Eliminar características con poca varianza (en scikit learn, [VarianceThreshold](#))
- Eliminar características redundantes con otras (en scikit learn, [mutual\\_info\\_classif](#))

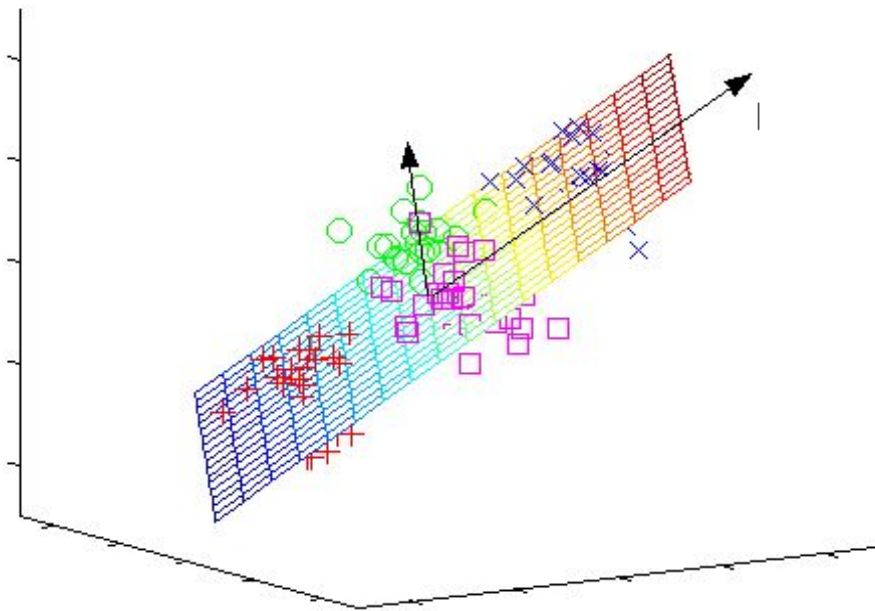
# Agrupamiento de Características

- Combinación de características dependientes (redundantes)
  - Por ejemplo, combinación lineal de el número de paradas recorridas por un colectivo y la distancia
- Combinación de características que sabemos que se pueden representar unidas
  - Por ej., sustituir viento, temperatura y humedad por sensación térmica
- Podemos sustituir características por la clase a la que pertenecen!
  - Por ej., en lenguaje natural, sustituir “corríamos” por “correr” o por V

# Kernel Trick

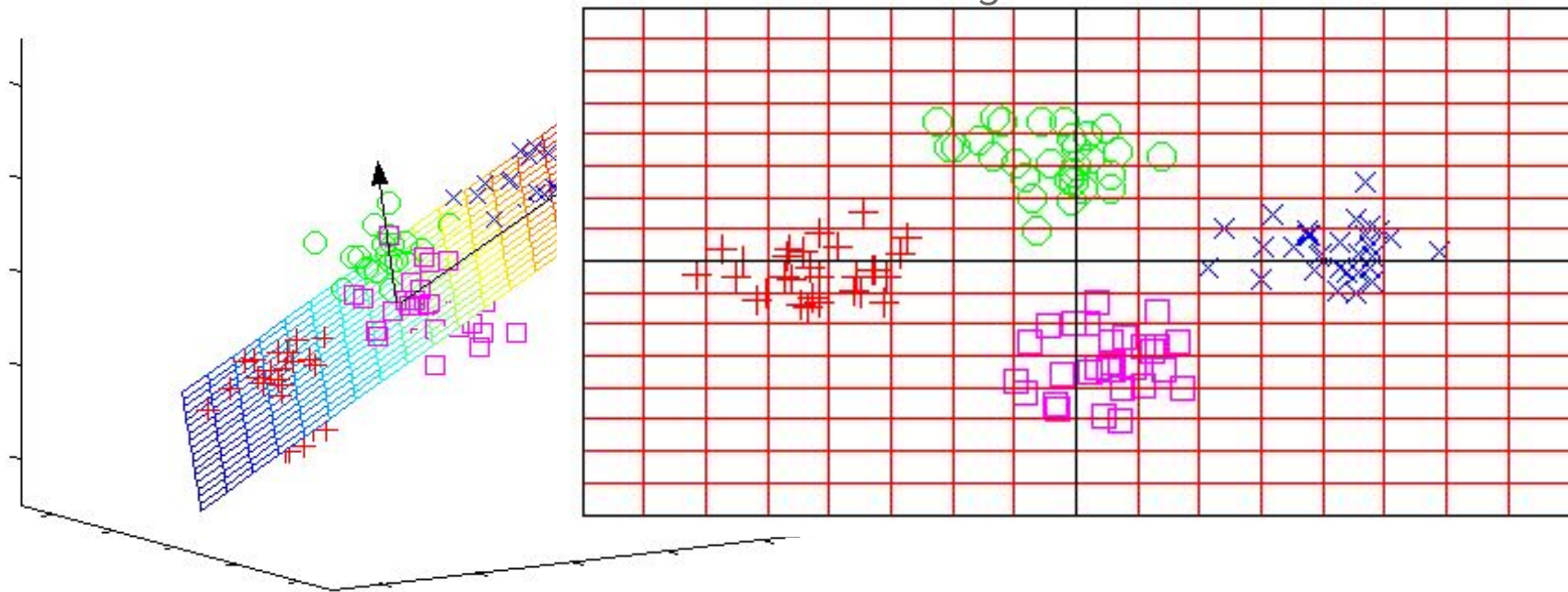
# Principal Component Analysis

Minimiza el error cuadrado de reconstruir los datos originales



# Principal Component Analysis

Minimiza el error cuadrado de reconstruir los datos originales



# Descomposición en Valores Singulares

Los componentes principales se encuentran descomponiendo una matriz en valores singulares (eigenvalues) → singular value decomposition (SVD)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix} \times \begin{bmatrix} \lambda_1 & \emptyset \\ \emptyset & \lambda_2 \end{bmatrix} \times \begin{bmatrix} \text{---} & v_1 & \text{---} \\ \text{---} & v_2 & \text{---} \end{bmatrix}$$



# Latent Semantic Analysis

Los componentes de la matriz en valores singulares

Términos x Documentos

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

=

$$\begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix}$$

Documentos x Conceptos

$$\times \begin{bmatrix} \lambda_1 & \emptyset \\ \emptyset & \lambda_2 \end{bmatrix}$$

Fuerza de cada concepto

Términos x Conceptos

$$\begin{bmatrix} \text{---} & v_1 & \text{---} \\ \text{---} & v_2 & \text{---} \end{bmatrix}$$

# Latent Semantic Analysis

retrieval  
inf. ↓ brain lung

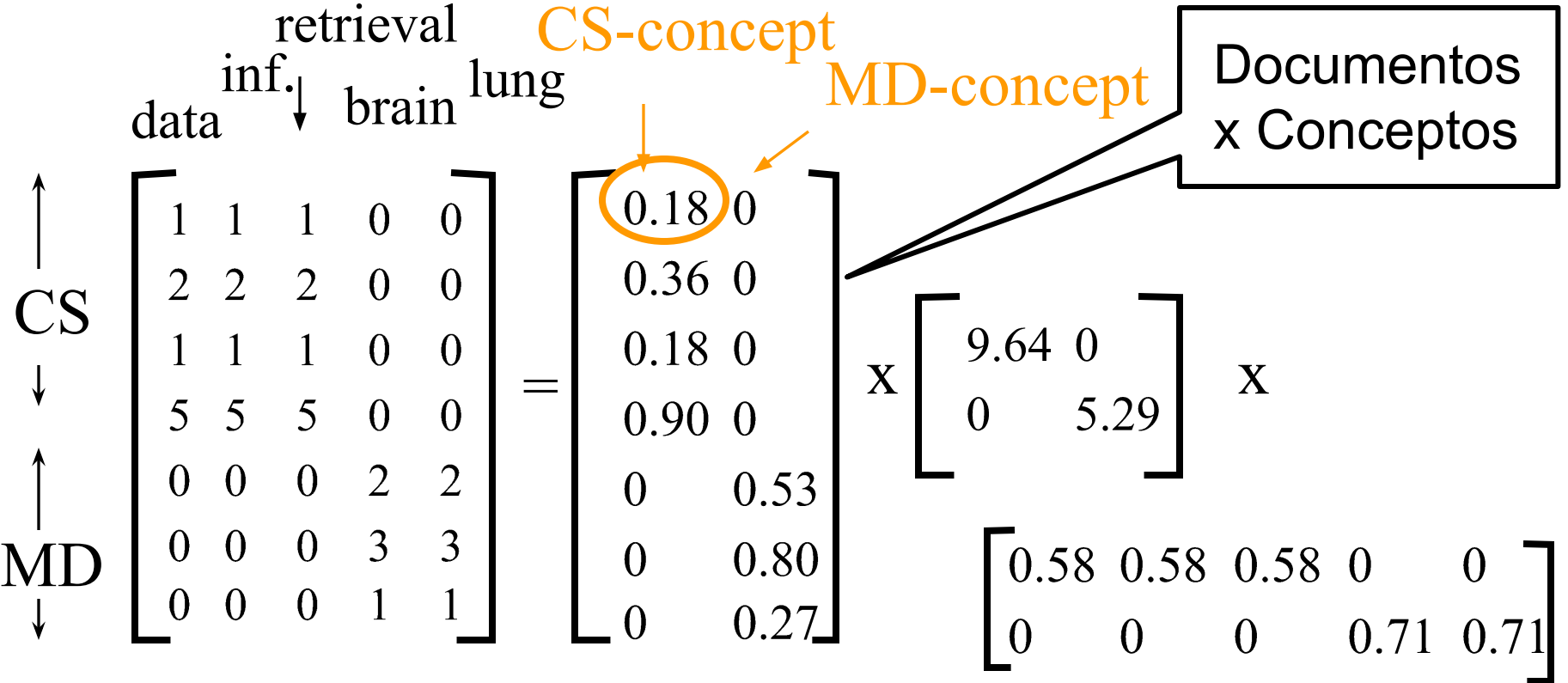
data

CS

MD

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

# Latent Semantic Analysis



# Latent Semantic Analysis

retrieval  
inf. ↓ brain lung

data

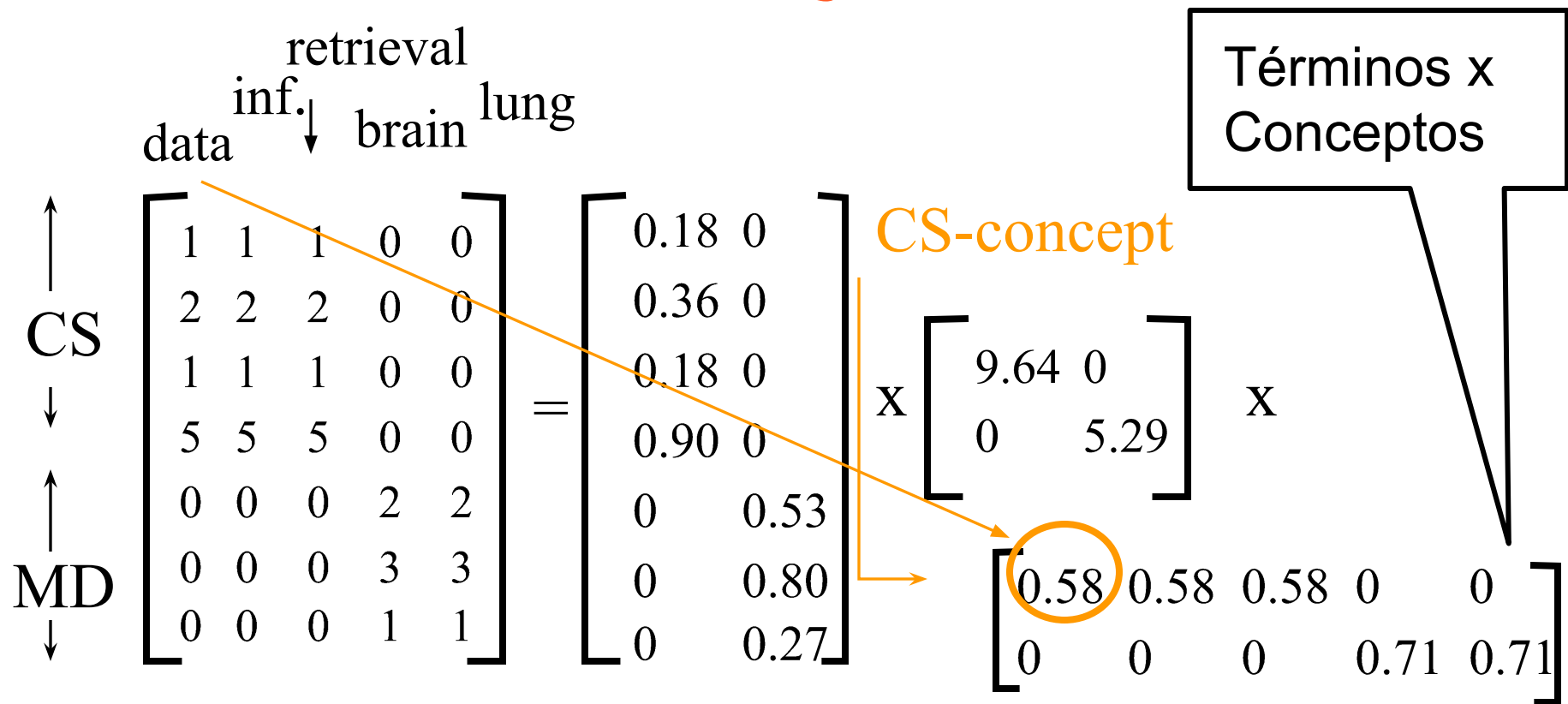
CS

MD

'strength' of CS-concept

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

# Latent Semantic Analysis



# Latent Semantic Analysis: Reducción de dimensionalidad

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

Diagram illustrating the reduction of dimensionality in Latent Semantic Analysis (LSA). The input matrix (7x5) is decomposed into three matrices: a 7x2 matrix, a 2x2 matrix, and a 2x5 matrix. The second matrix (2x2) is crossed out with an orange X, indicating it is the singular value matrix. The first and third matrices are crossed out with green lines, indicating they are the left and right singular vectors, respectively. The resulting matrices are:

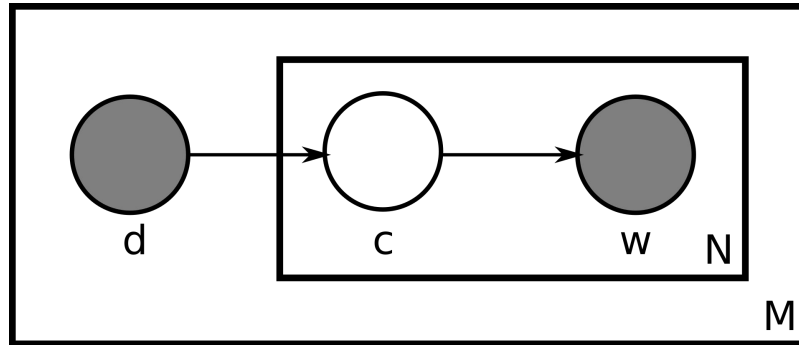
- Input matrix (7x5):
- Left singular vectors (7x2):
- Singular values (2x2):
- Right singular vectors (2x5):

# Latent Semantic Analysis: Reducción de dimensionalidad

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0.18 \\ 0.36 \\ 0.18 \\ 0.90 \\ 0 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} 9.64 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \end{bmatrix}$$

# Probabilistic Latent Semantic Analysis

- Modela la distribución de cada co-ocurrencia como una mezcla de distribuciones multinomiales independientes o clases latentes o tópicos (el  $n$  de tópicos es un parámetro)

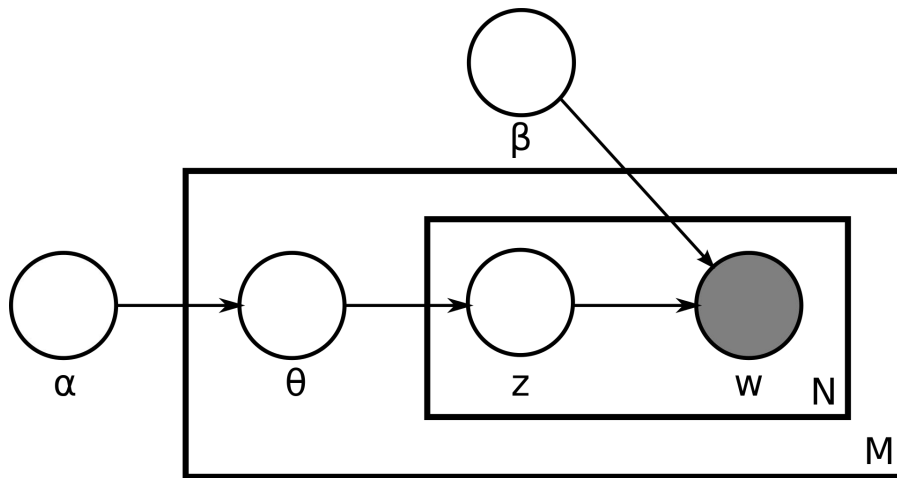


- $d$  es el documento
- $c$  es un tópico obtenido de la distribución de tópicos del documento  $P(c|d)$
- $w$  es una palabra obtenida de la distribución de palabras de  $c$

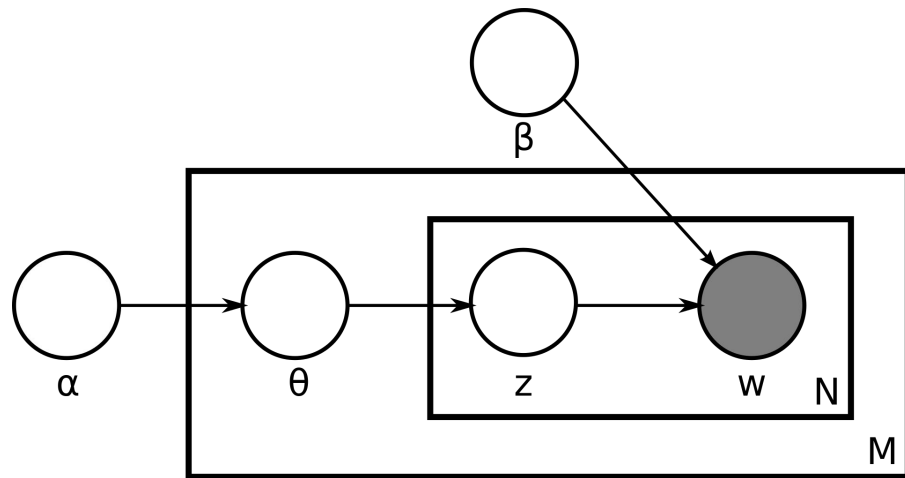


# Latent Dirichlet Allocation

- Modela la distribución de cada co-ocurrencia como una mezcla de distribuciones multinomiales (clases latentes o tópicos)
- Se asume que las clases latentes están distribuidas según la distribución de Dirichlet, una distribución de probabilidad continua multivariada



# Latent Dirichlet Allocation



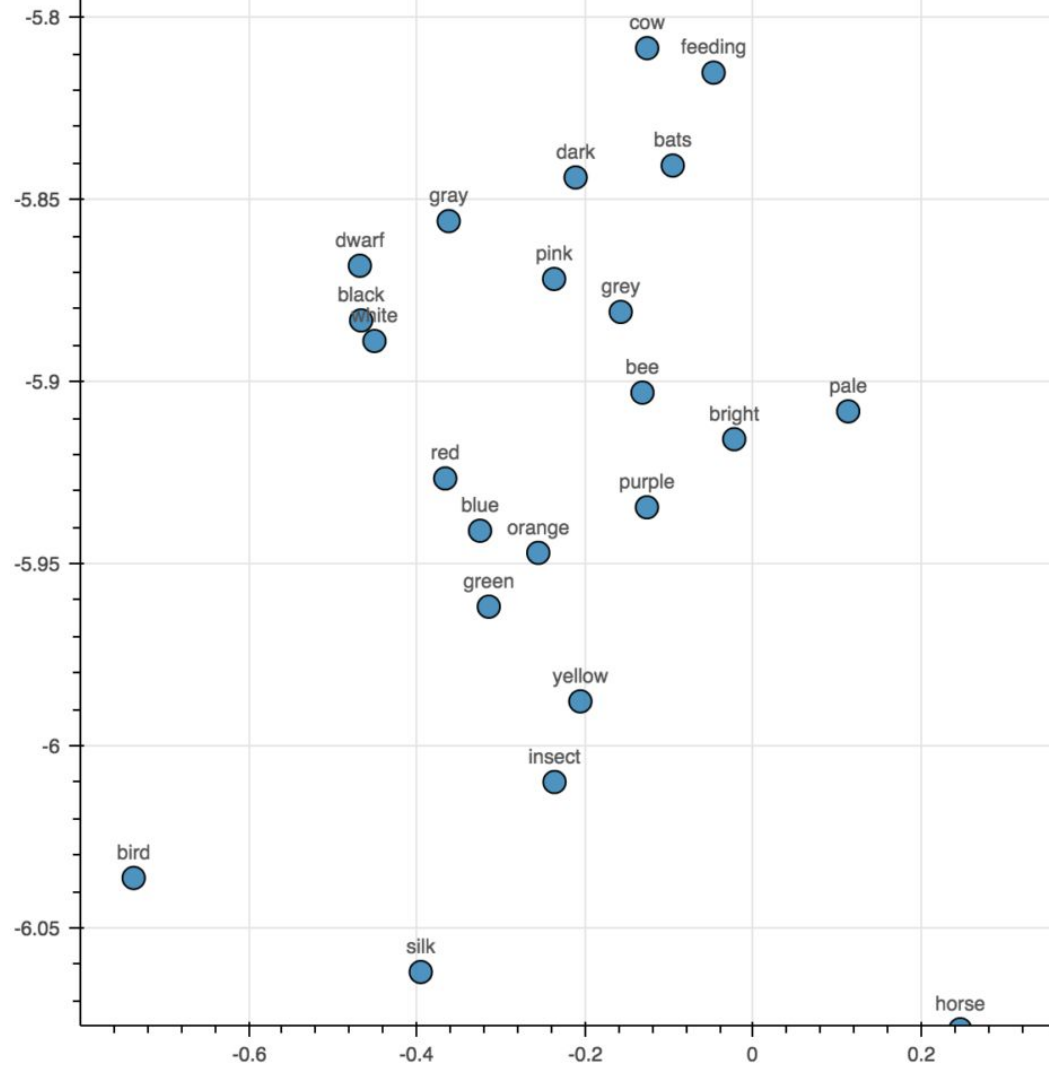
- $\alpha$  es el parámetro de Dirichlet en la distribución de tópicos por documentos
- $\beta$  es el parámetro de Dirichlet en la distribución de palabras por tópicos
- $\theta$  es la distribución de tópicos para el documento  $i$
- $\phi$  es la distribución de palabras para el tópico  $k$
- $z$  es el tópico para la  $j$ -ésima palabra del documento  $i$
- $w$  es la  $j$ -ésima palabra del documento  $i$

# t-SNE

t-distributed stochastic neighbor embedding (t-SNE)

- reducción de dimensionalidad no lineal
- para visualización en dos o tres dimensiones
- los objetos semejantes quedan cercanos y los más diferentes quedan más distantes, con alta probabilidad

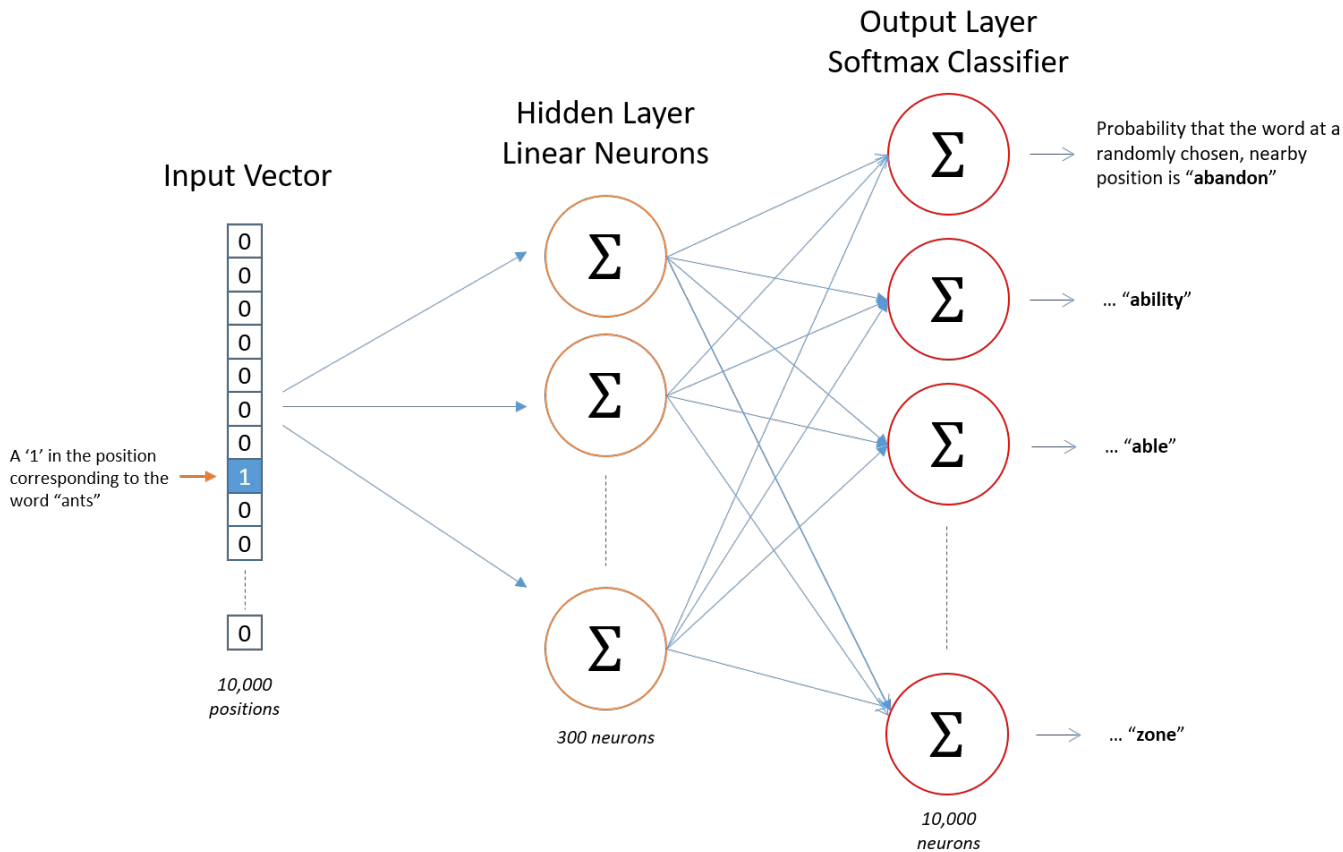
## t-SNE



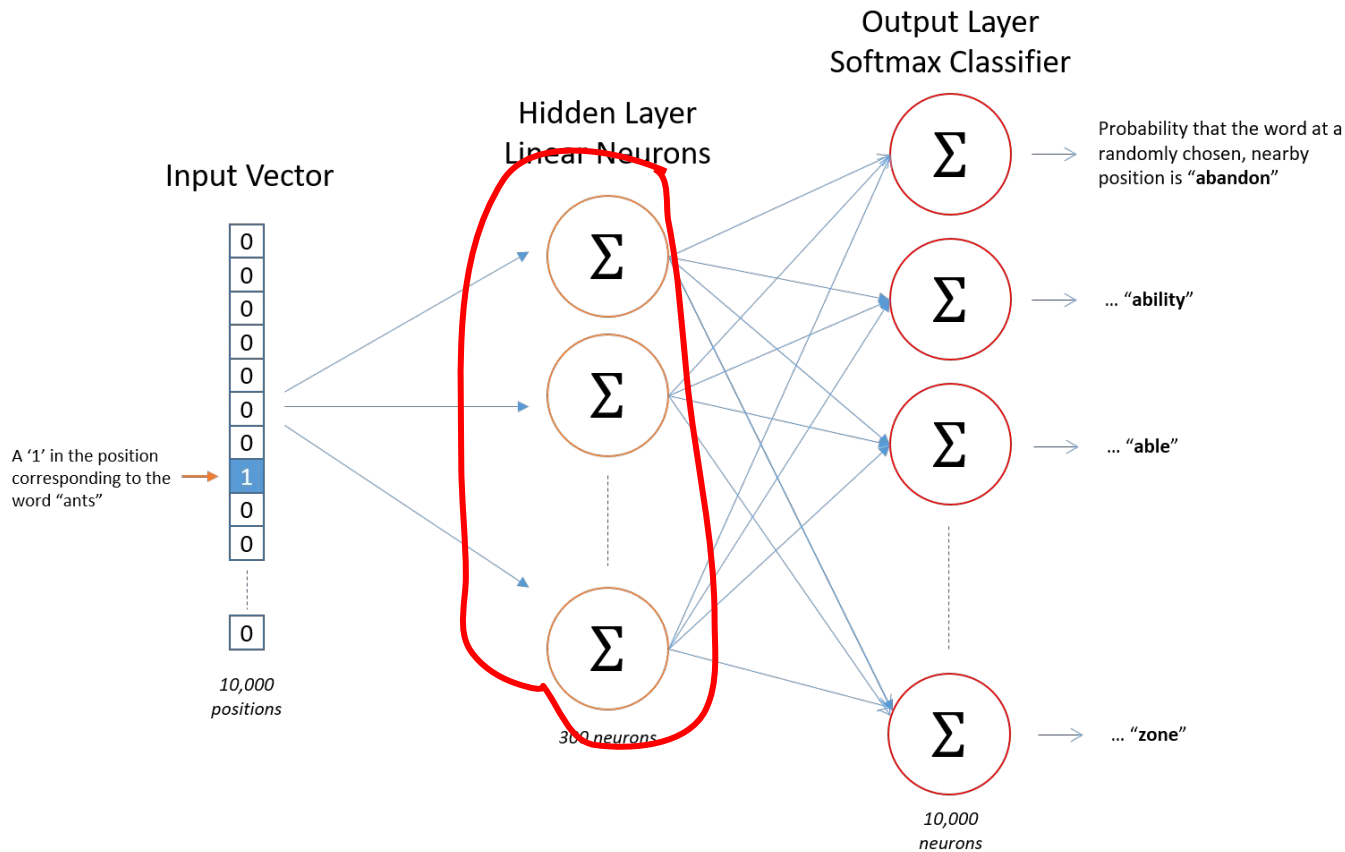
# no linealidad: embeddings neuronales

1. entrenar una red neuronal
2. eliminar la capa de predicción
3. la capa anterior a la de predicción es el nuevo espacio
4. el camino hasta esa capa es el mecanismo de proyección

# no linealidad: embeddings neuronales



# no linealidad: embeddings neuronales



# Embeddings neuronales

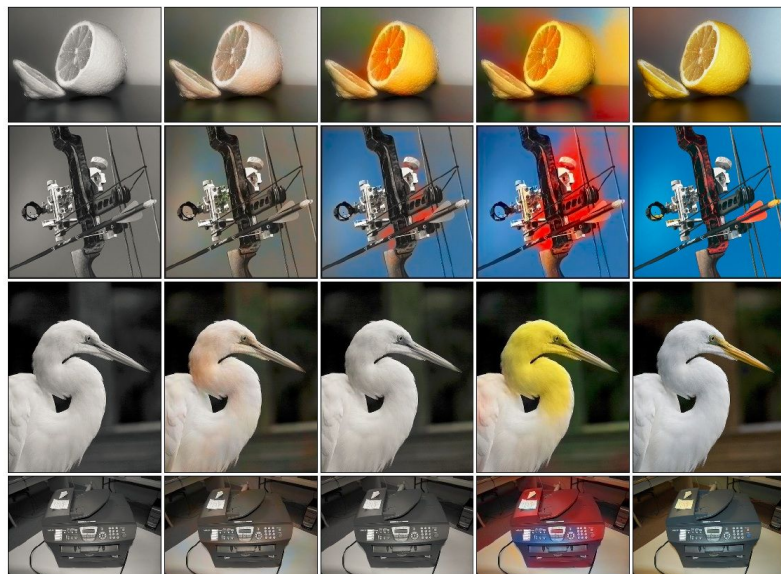
- Entrenar una red neuronal con una **tarea de pretexto** para la que tenemos muchos ejemplos naturalmente
  - Predecir una palabra dado su contexto, o un contexto dada una palabra
  - Reconstruir una imagen
- Eliminar la capa de predicción de la red
- La capa anterior a la de predicción es la nueva caracterización de los objetos
  - Menos características → acercándonos a las causas latentes!
- Se usa la red para convertir los objetos del espacio original al espacio de embeddings
- Es relativamente barato de obtener
- Ahora podemos caracterizar datos supervisados con información poblacional de grandes cantidades de datos no supervisados



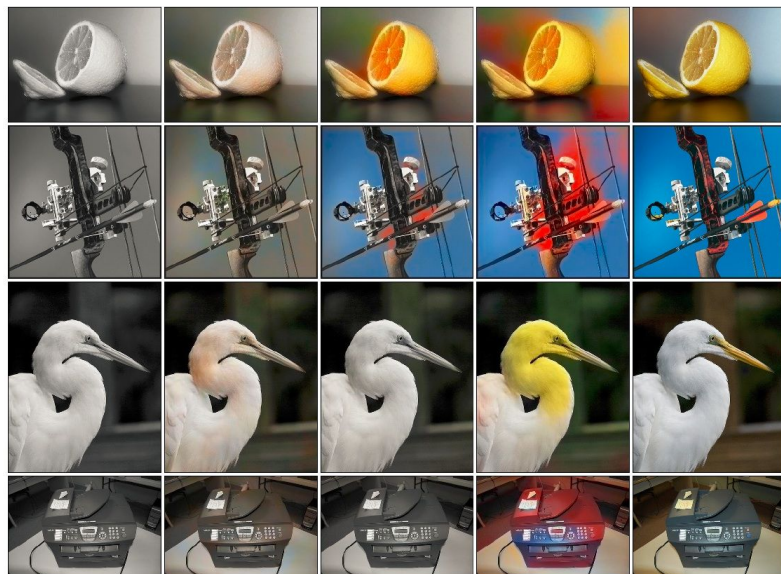
# Tareas de pretexto



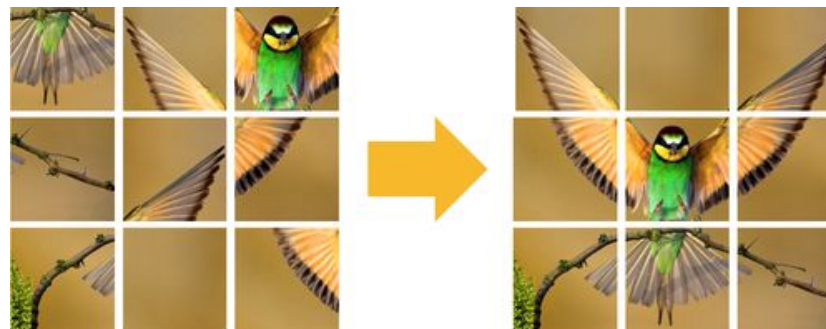
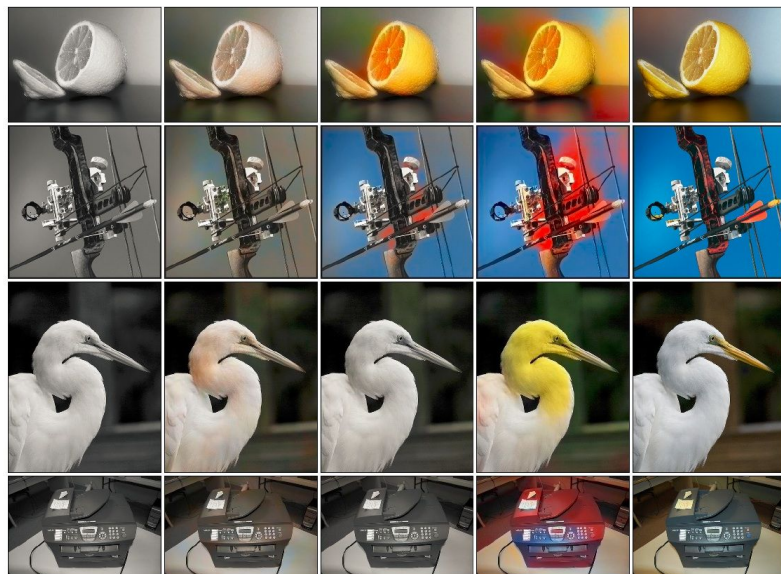
# Tareas de pretexto



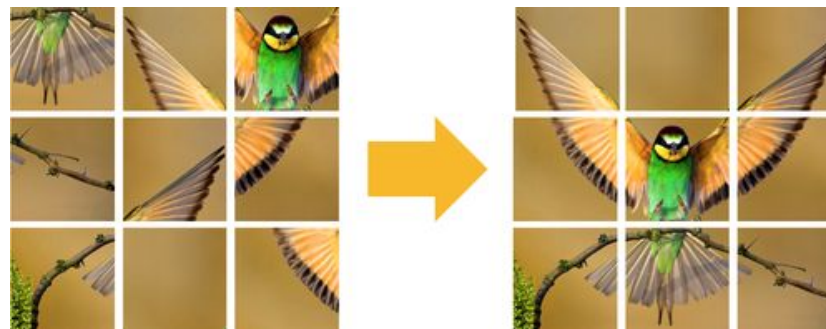
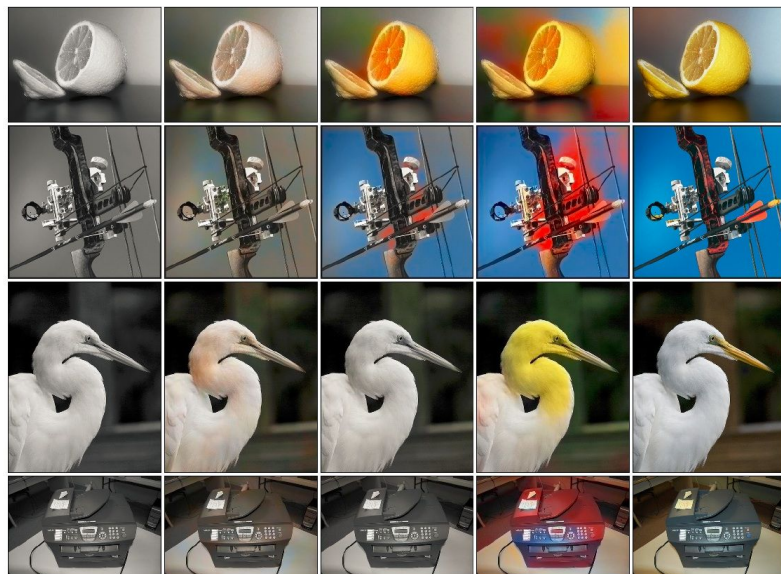
# Tareas de pretexto



# Tareas de pretexto



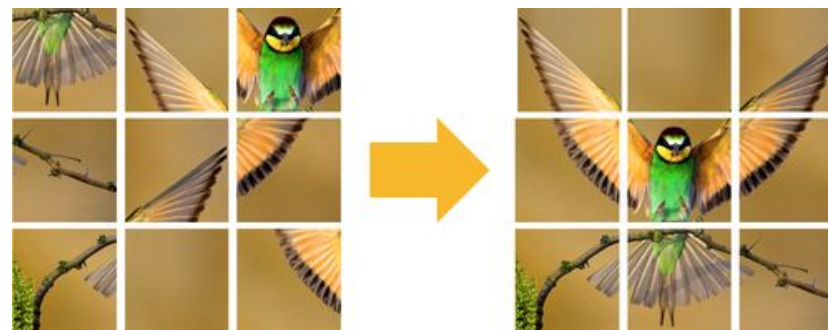
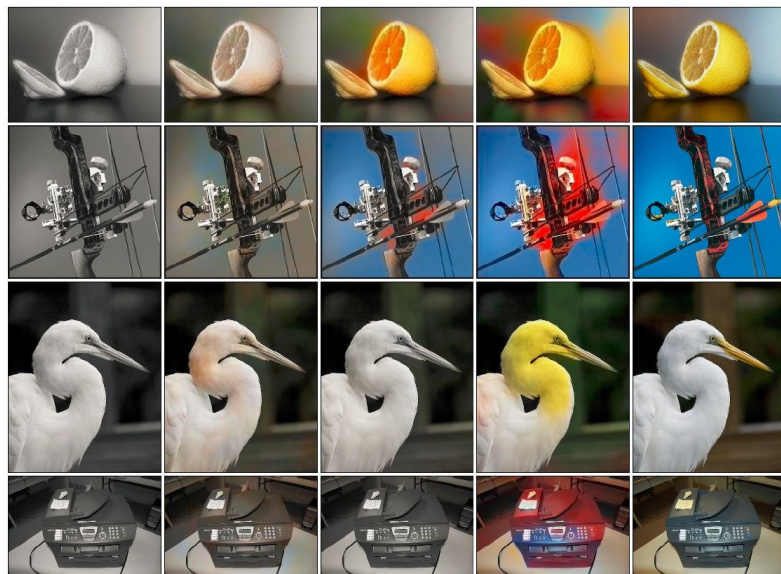
# Tareas de pretexto



el gato come pescado



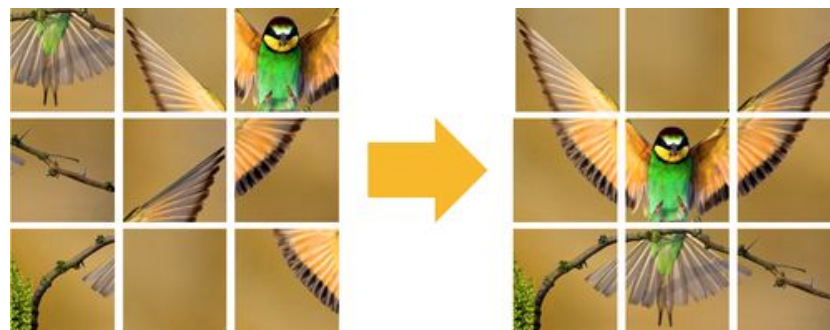
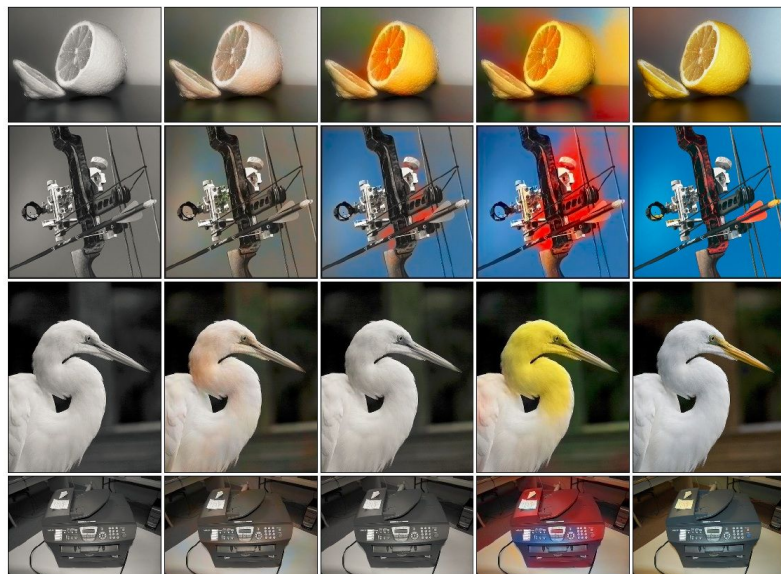
# Tareas de pretexto



el gato come pescado

__	gato	come	pescado	?
el	__	come	pescado	?
el	gato	__	pescado	?
el	gato	come	__	?

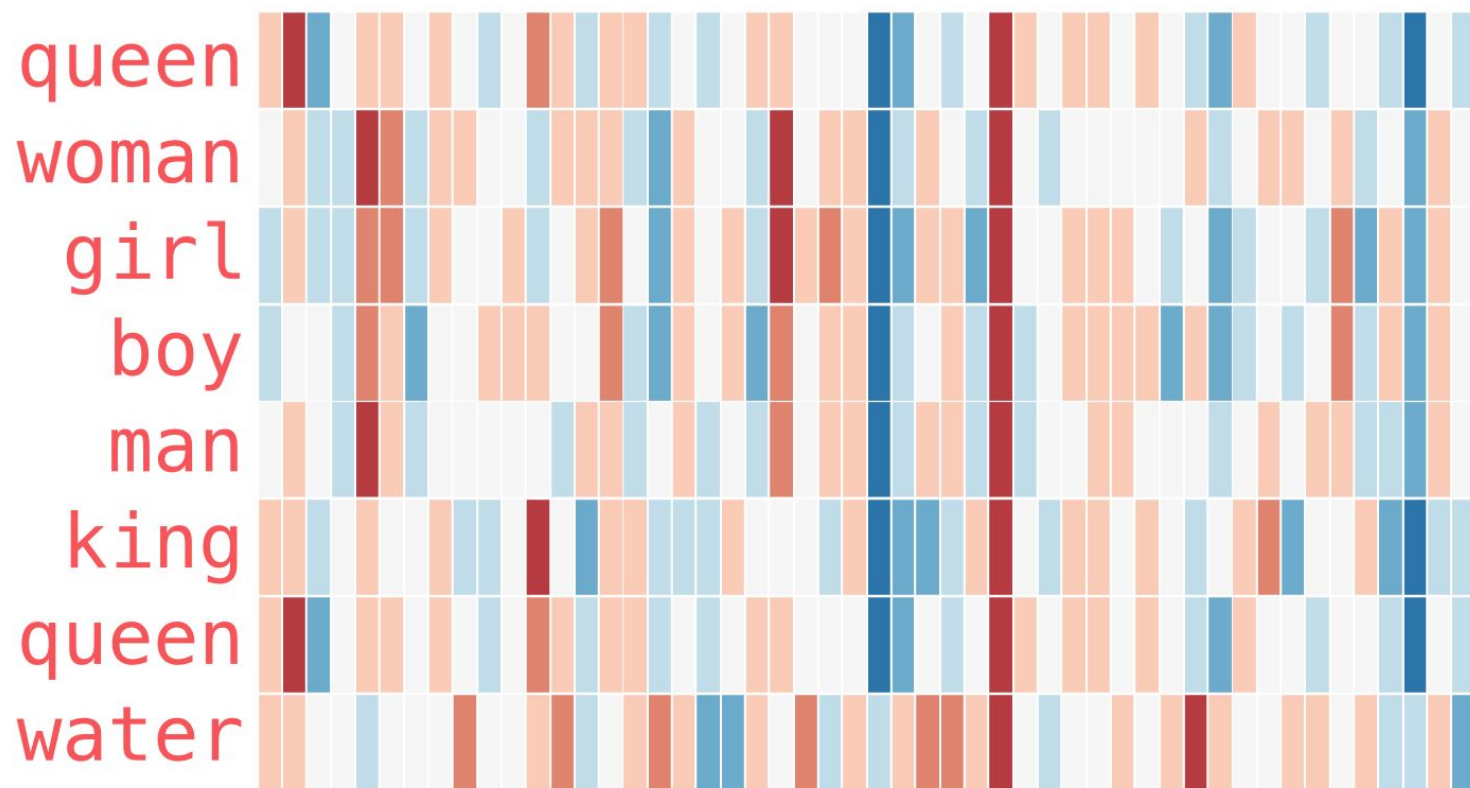
# mejores representaciones



el gato come pescado

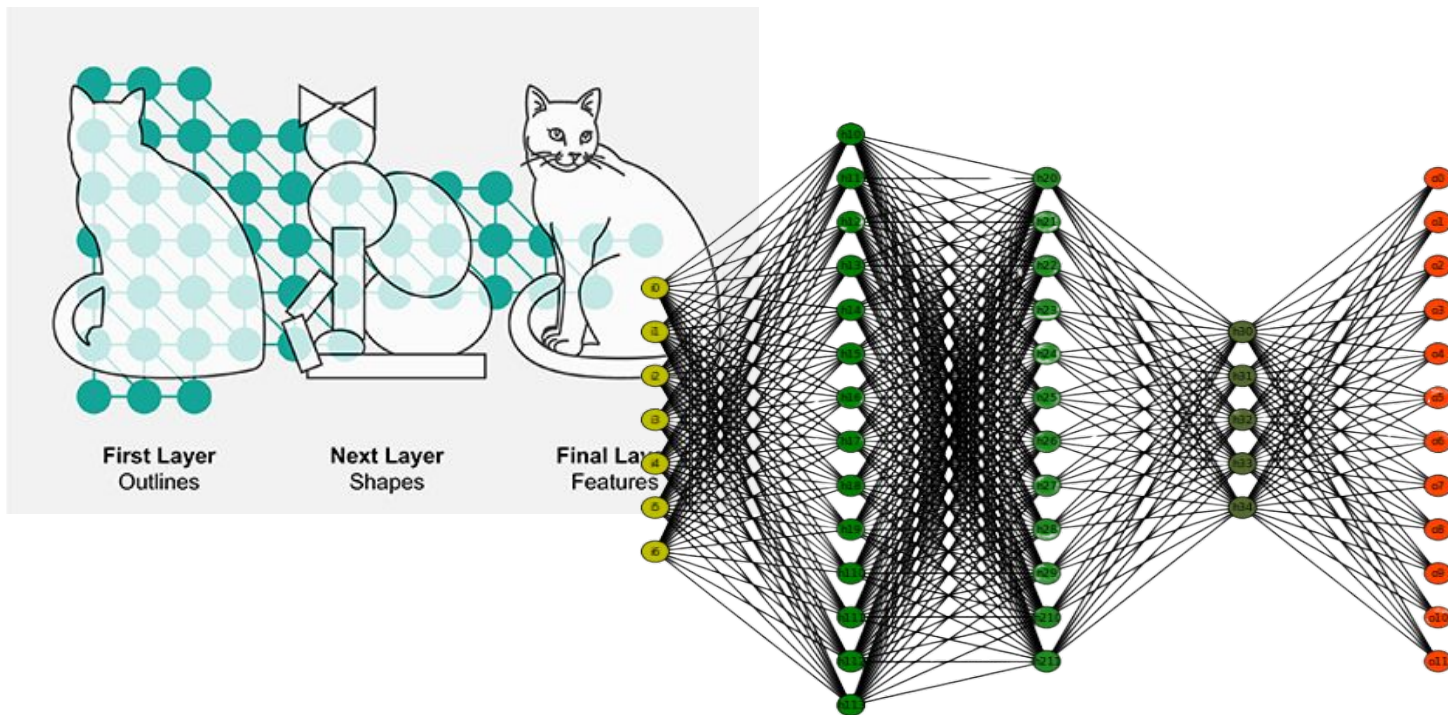
\_\_ gato come pescado                   ?   el  
 el \_\_\_\_ come pescado                   ?  
 gato  
 el gato \_\_\_\_ pescado                   ?  
 come  
 el gato come                               ?

# Nos acercamos a las causas latentes

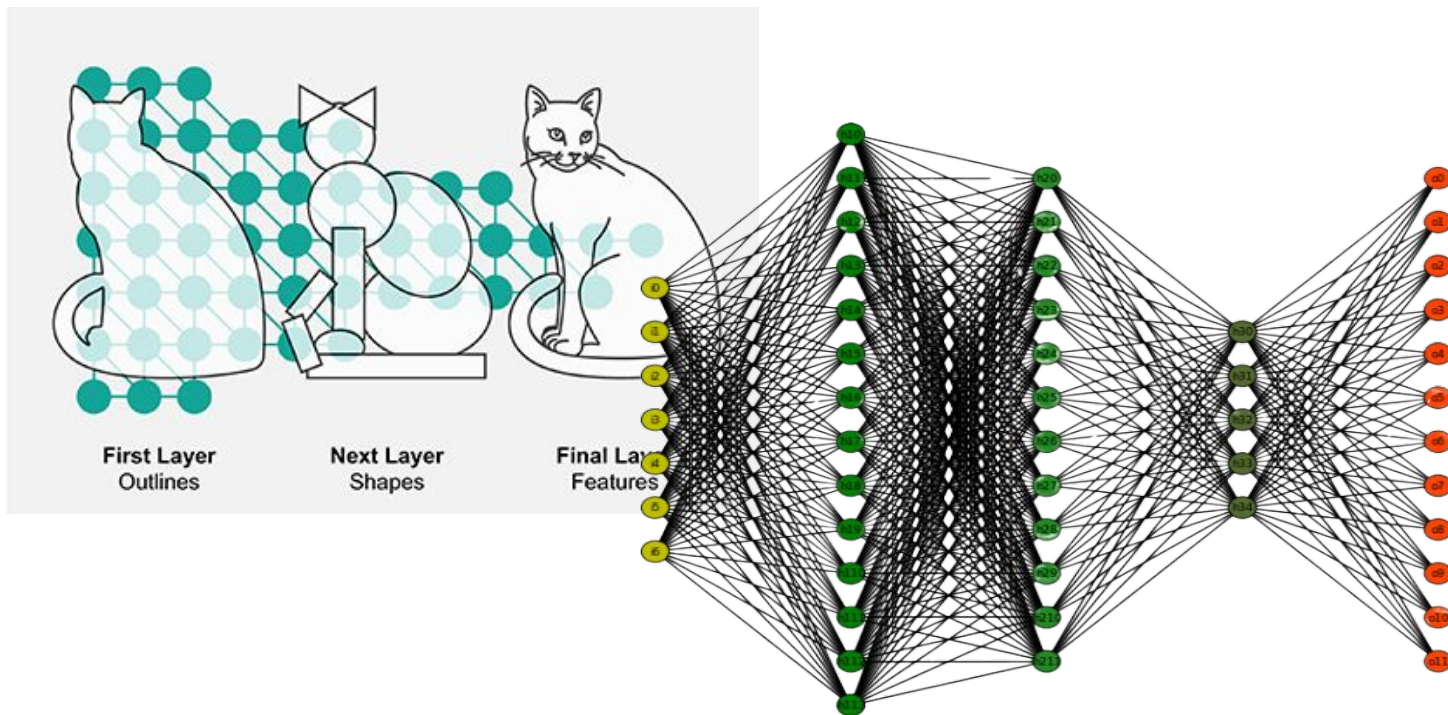




# Perdemos interpretabilidad



# Perdemos interpretabilidad



# Embeddings neuronales

Gensim (word2vec, doc2vec, y toda la familia)

Fasttext

Bert, GPT-2, GPT-3

T-sne

<https://shuaiw.github.io/2016/12/22/topic-modeling-and-tsne-visualization.html>

<https://distill.pub/2016/misread-tsne/>