

# Tema 6.1

# Modelos de secuencias y texto

Deep Learning

Máster Oficial en Ingeniería Informática

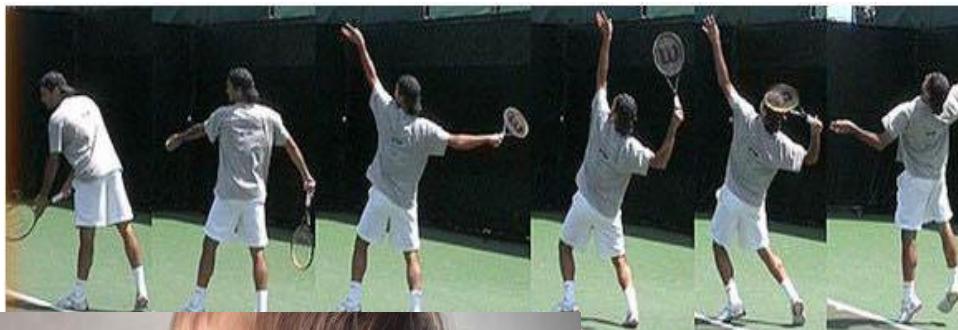
Universidad de Sevilla

# Contenido

- Modelos de secuencias
- Modelos para texto

# Modelos de secuencias

- Los datos suelen tener naturaleza de secuencia



STOCK	BID	OFFER	LAT.	FARM PRIDE	0.100	0.140
EUR GROUP	0.060	0.070	0.000	0 FARM PRIDE	0.026	0.030
EUROGOLD	0.098	0.140	0.000	0 FE LIMITED	0.120	0.130
EURO GAS	0.325	0.335	0.335	77T FEDAX	0.024	0.033
EUROZ	1.000	1.020	1.000	4T FERROWEST	0.052	0.057
EVOLUTION	1.935	1.940	1.935	2M FERRUM	0.800	0.810
FV2 LTD	0.041	0.050	0.050	5T FIDUCIAN	0.110	0.125
EXALT RES	0.000	0.000	0.000	0 FEAX	0.175	0.180
EXCAX	0.040	0.049	0.040	50T FINBAR	0.200	0.220
EXCALIBUR	0.001	0.002	0.000	0 FINDERS	0.008	0.009
EXCELA	0.010	0.090	0.000	0 FIRESTONE	0.014	0.015
EXCENSOR	0.190	0.195	0.190	30T FIRSTFOLIO	0.020	0.035
EXCO RES	0.260	0.265	0.260	5HT FISSION EN		
EXOMA ENER	0.072	0.075	0.072	35T FITZROYRES		
EZAAX	0.430	0.490	0.000			



WIKIPEDIA

The Free Encyclopedia

Article Talk

Deep learning

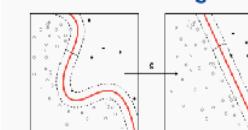
From Wikipedia, the free encyclopedia

For deep versus shallow learning in educational psychology, see [Student approaches to learning](#).

Deep learning (also known as [deep structured learning](#), [hierarchical learning](#) or [deep machine learning](#)) is a class of [machine learning](#) algorithms that:<sup>[1]</sup>(pp 199–200)

- use a cascade of many layers of nonlinear processing units for [feature extraction](#) and transformation. Each successive layer uses the output from the previous layer as input. The algorithms may be [supervised](#) or [unsupervised](#) and applications include pattern analysis (unsupervised) and classification (supervised).
- are based on the (unsupervised) learning of multiple levels of features or representations of the data. Higher level features are derived from lower level features to form a hierarchical representation.
- are part of the broader machine learning field of learning representations

## Machine learning and data mining



### Problems

[Supervised learning \(classification · regression\)](#)[Clustering](#)[Dimensionality reduction](#)

[show]

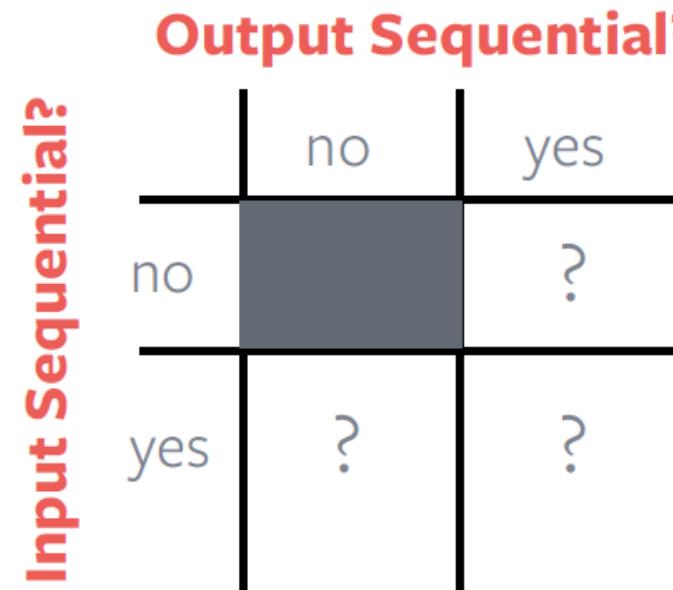
[show]

[show]

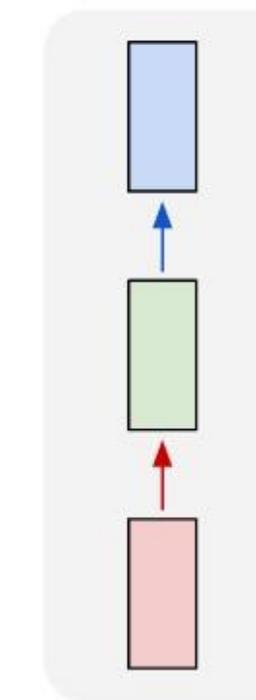
[show]

# Modelos de secuencias

- Escenarios



one to one

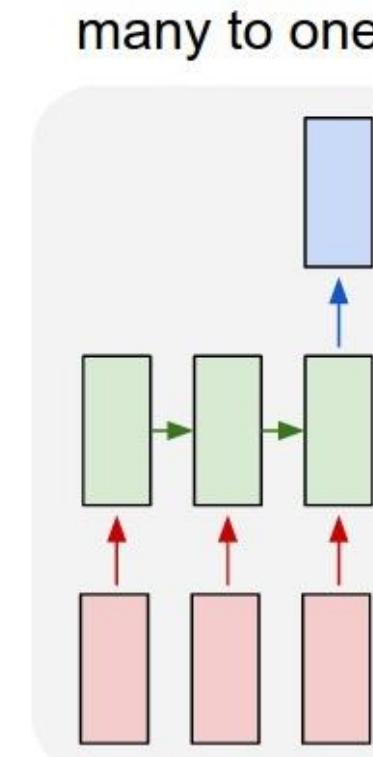
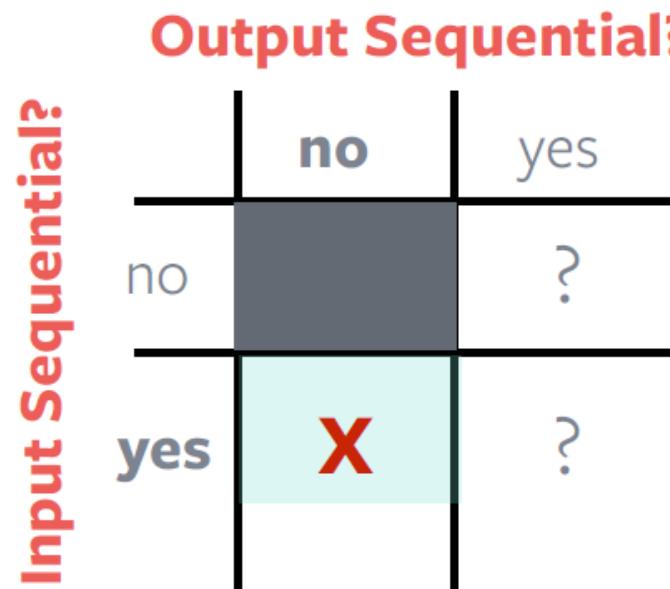


Ejemplos:

- Clasificación de imágenes

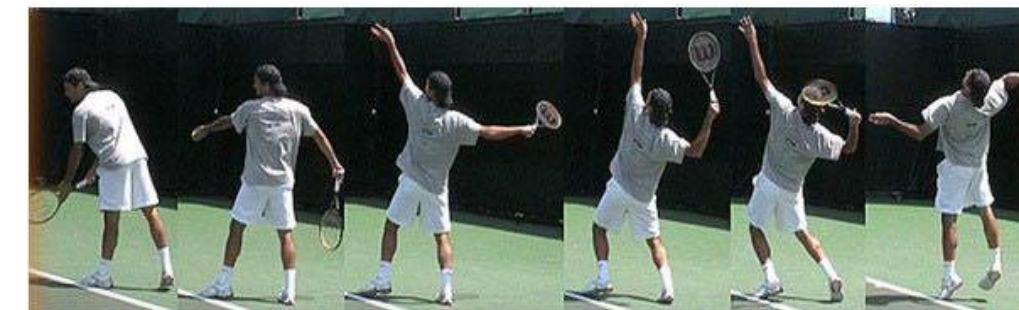
# Modelos de secuencias

- Escenarios



Ejemplos:

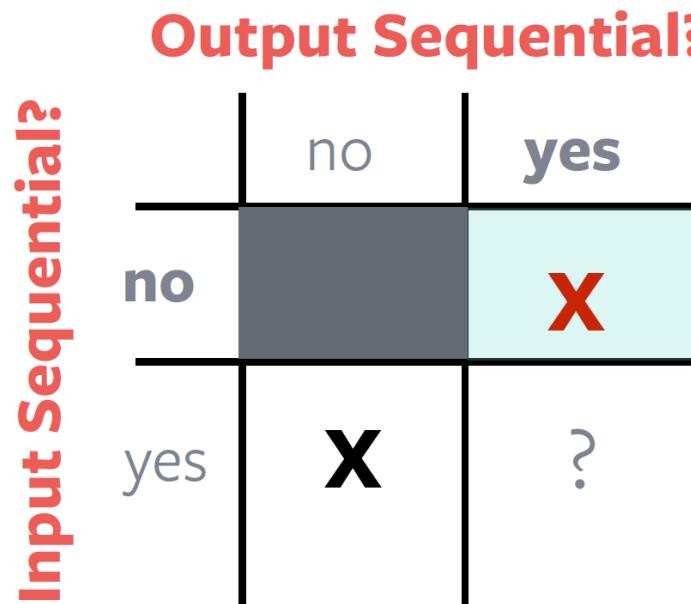
- Clasificación de texto
- Modelado de lenguaje
- Reconocimiento de acciones
- Clasificación de género de música



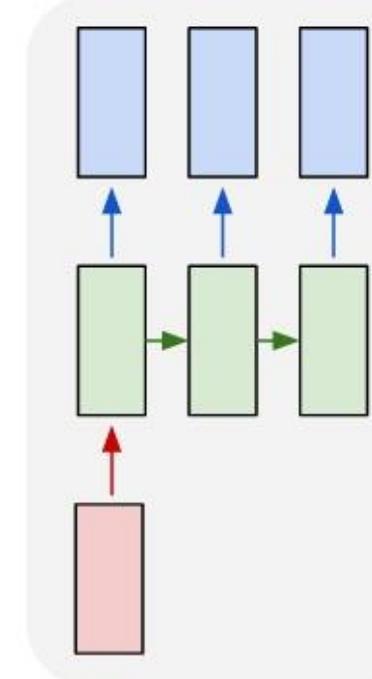
Jugando al tenis

# Modelos de secuencias

- Escenarios



one to many



Ejemplos:

- Image captioning

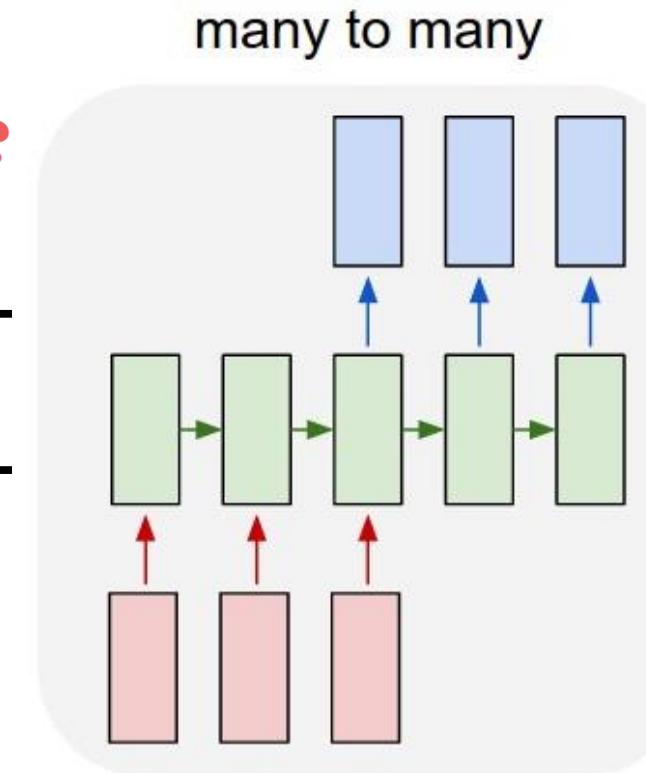


A square with a fountain and tall buildings in the background, with some trees and a few people hanging out.

# Modelos de secuencias

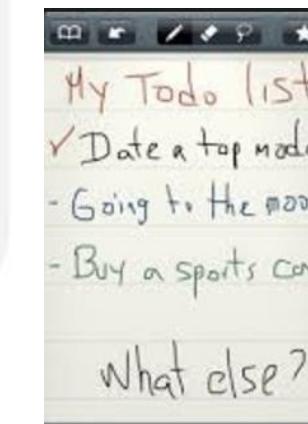
- Escenarios

Output Sequential?	
no	yes
no	X
yes	X



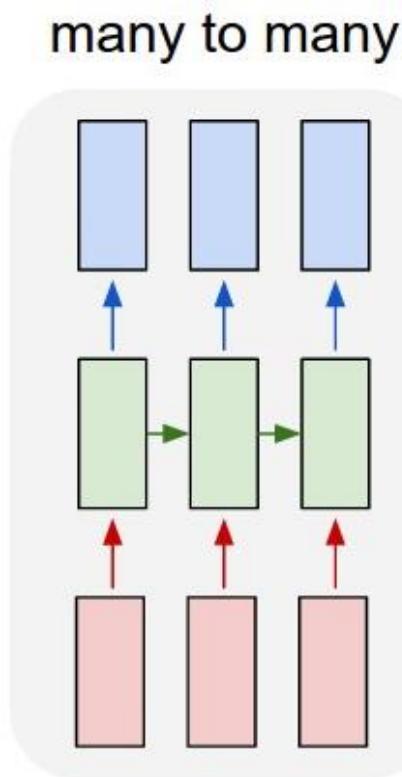
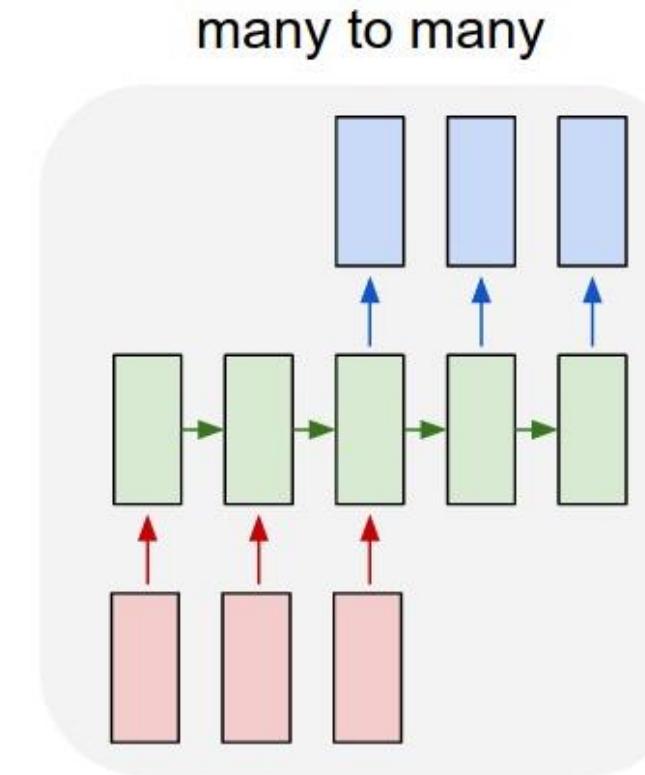
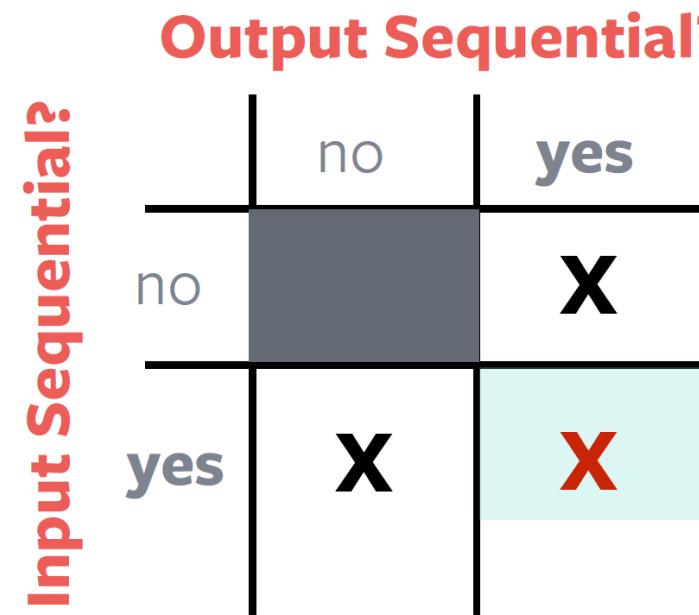
Ejemplos:

- Traducción máquina
- Resúmenes
- Reconocimiento del habla
- OCR
- Predicción de fotogramas en video



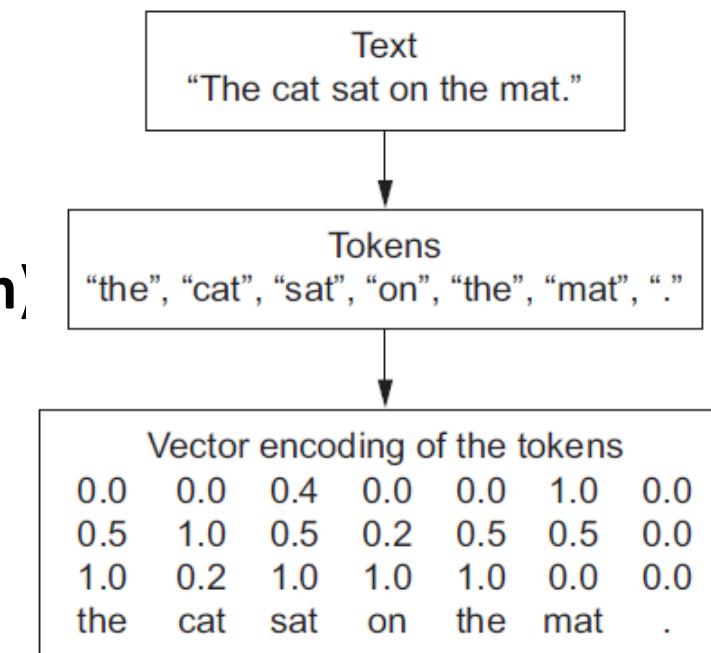
# Modelos de secuencias

- Escenarios



# Modelos para texto

- Texto: secuencia de caracteres o palabras (mejor palabras)
- Deep Learning no permite entender el texto, sino **reconocer patrones** en las palabras, frases y párrafos
- Primer paso: introducir texto en una red neuronal (**vectorización**)
  - Romper el texto en tokens (**tokenization**)
  - Traducir cada token a un vector



# Modelos para texto

- Un **token** puede ser:
  - Una palabra
  - Un carácter
  - Un **N-grama (bag-of-words)**
    - N-grama: grupo de n palabras o caracteres consecutivos en una frase que se pueden solapar (siendo  $n < N$ )
    - Ejemplo de 2-grama: “La asignatura me gusta mucho” → {"La", “La asignatura”, “asignatura”, “asignatura me”, “me”, “me gusta”, “gusta”, “gusta mucho”, “mucho”}
    - El conjunto anterior de 2-gramas se denomina bag-of-2words.
    - Bag-of-words: es un conjunto de tokens, sin ordenación (el orden está dentro del token).
    - Usado en técnicas clásicas, pero no en Deep Learning

# Modelos para texto

- Codificación de tokens en vectores:

- **One-hot encoding**

- Cada palabra corresponde a un índice en un vector de tamaño igual al vocabulario
    - One-hot hashing trick: usar una tabla hash en vez de un vector, para reducir tamaño
    - Desventajas:
      - Requiere mucha memoria
      - Demasiada dimensionalidad
      - No existe una función que calcule la distancia entre dos palabras
      - Mucho ruido en la red: palabras cercanas pueden tener representación lejana

Hombre (5391)	Mujer (9853)	Rey (4914)	Reina (7157)	Manzana (456)	Naranja (6257)
0	0	0	0	0	0
0	0	0	0	...	0
0	0	...	0	1	0
0	0	1	0	...	0
...	0	...	0	0	0
1	...	0	...	0	...
...	1	0	1	0	1
0	...	0	...	0	...
0	0	0	0	0	0

# Modelos para texto

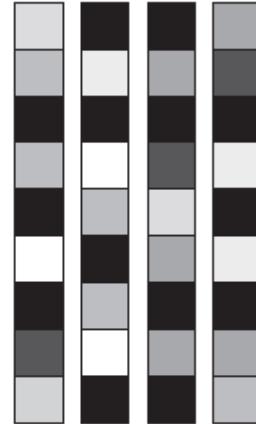
- Codificación de tokens en vectores:
  - **Token embedding (Word-embedding para palabras, o inmersión de palabras)**
  - Vectores de números reales (vs binario)
  - Menor dimensionalidad (256 vs 20.000)
  - Es posible calcular la distancia entre dos palabras.

	<b>Hombre (5391)</b>	<b>Mujer (9853)</b>	<b>Rey (4914)</b>	<b>Reina (7157)</b>	<b>Manzana (456)</b>	<b>Naranja (6257)</b>
Sexo	-1	1	-0,95	0,97	0,00	0,01
Realeza	0,01	0,02	0,93	0,95	-0,01	0,00
Edad	0,03	0,01	0,7	0,69	0,03	-0,02
Comida	0,04	0,03	0,02	0,01	0,95	0,97
Tamaño...						



One-hot word vectors:

- Sparse
- High-dimensional
- Hardcoded

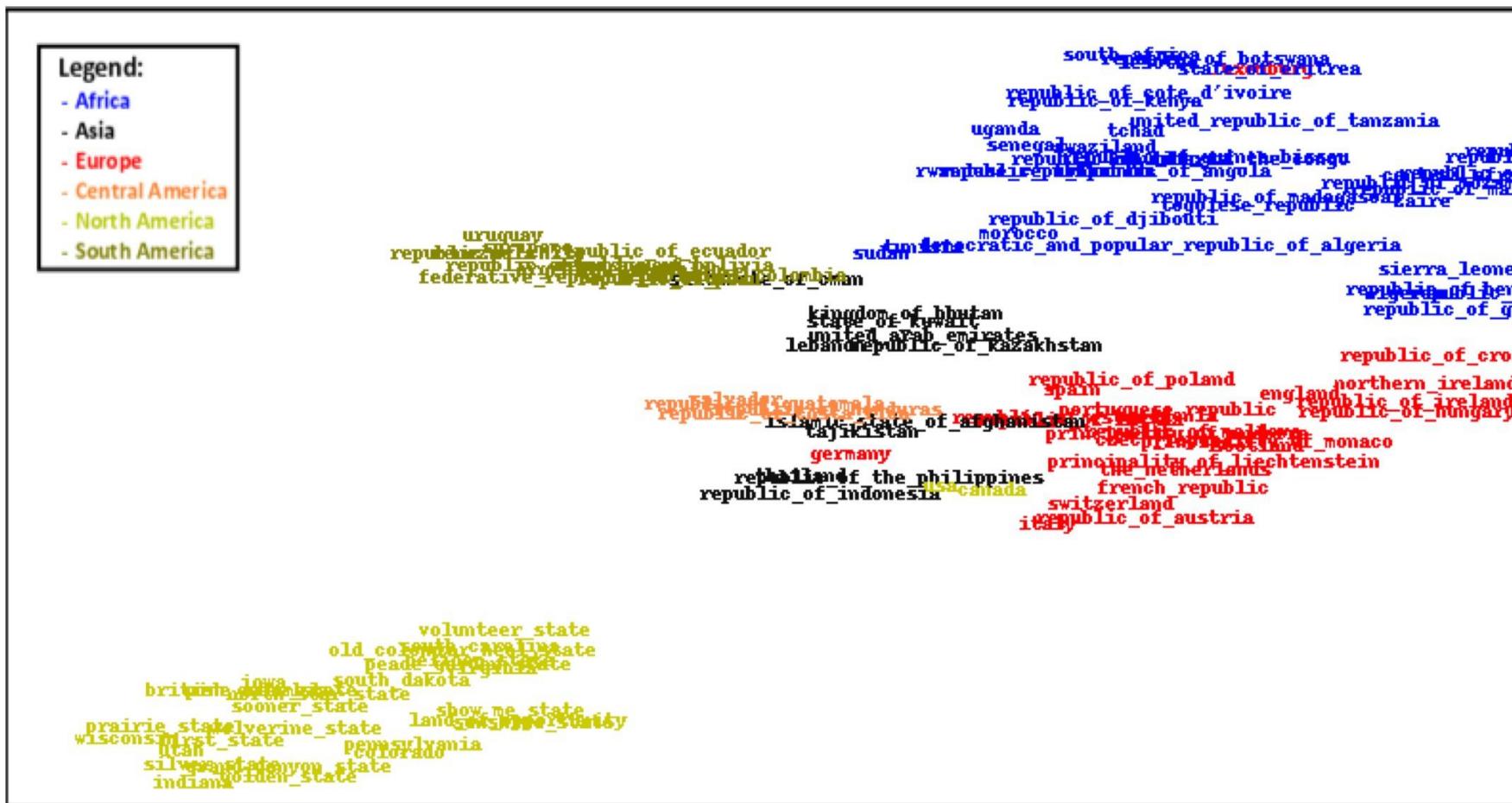


Word embeddings:

- Dense
- Lower-dimensional
- Learned from data

# Modelos para texto

- Ejemplo de Word-embedding [Bordes et al 2011]

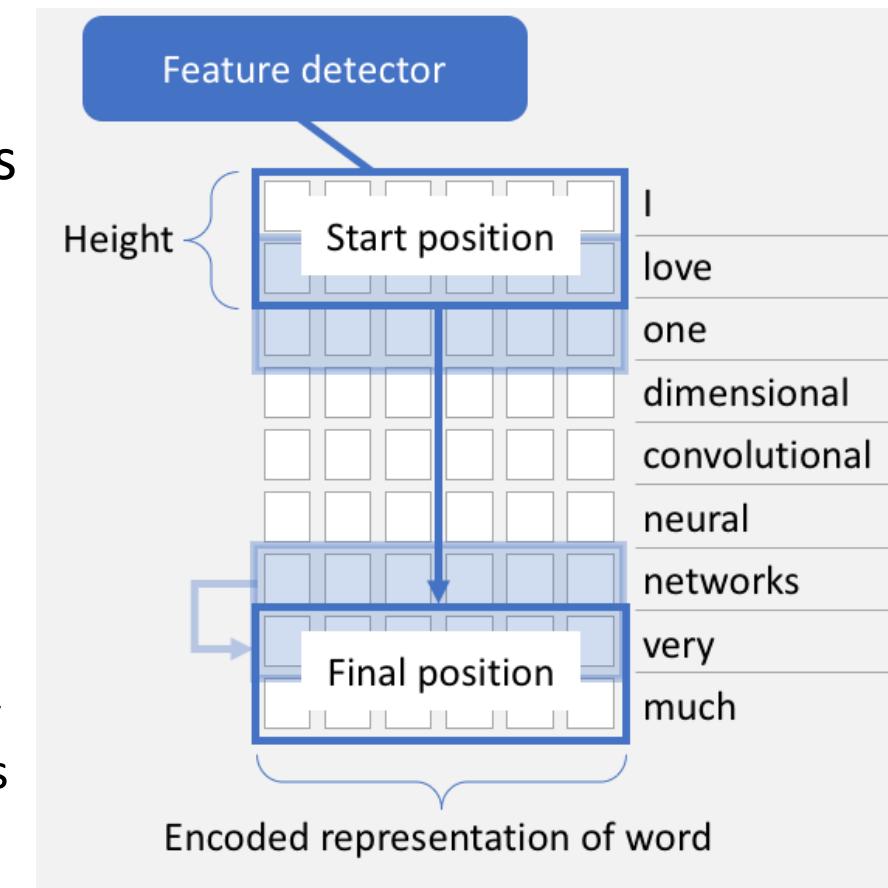


# Modelos para texto

- Obtención del Word-embedding:
  - **Entrenamiento** sobre datos:
    - Uso de una capa especial (embedding layer)
    - Queremos que palabras cercanas tengan un embedding cercano
    - Un Word-embedding puede diferir según lenguaje (Inglés, Español), contexto (twitter, legal), etc.
  - Usar **Word-embedding pre-entrenados** (transfer learning):
    - Cuando hay pocos datos para poder entrenar un Word-embedding propio
    - Entrenados como pesos de una red
    - **Word2vec** (Mikolov et al 2013, Google), basado en bag-of-words
    - **GloVe** (Pennington et al 2014, Standford), basado en contexto global
    - **FastText** (Mikolov et al 2018, Google), basado en Word2Vec mejorando precisión

# Modelos de secuencias

- En Deep Learning:
  - **Redes recurrentes neuronales (RNN)**: las veremos en el siguiente tema.
  - **Convoluciones 1D**: haremos una práctica al final del módulo. Son efectivas cuando:
    - Derivar características de segmentos cortos y de longitud fija.
    - La localización de la característica dentro del segmento no es relevante.
    - En el ejemplo: height es el tamaño del kernel (en 1D), y cada palabra se codifica con un vector (cuya longitud es la profundidad del kernel).



# Recapitulando

- Modelos para trabajar con secuencias:
  - **One-to-many, many-to-one, many-to-many**
- Modelos para texto:
  - Vectorización de texto mediante técnicas de **tokenization**
  - **One-hot-encoding vs Word-embedding**
- Veremos dos modelos para trabajar con secuencias en DL:
  - Las RNN
  - Convoluciones 1D.