

# Tema 6.1

## Modelos de secuencias y texto

Deep Learning

Máster Oficial en Ingeniería Informática

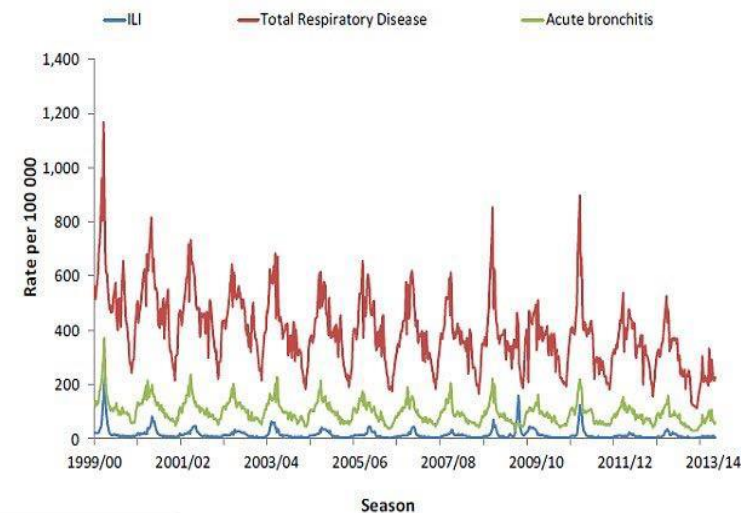
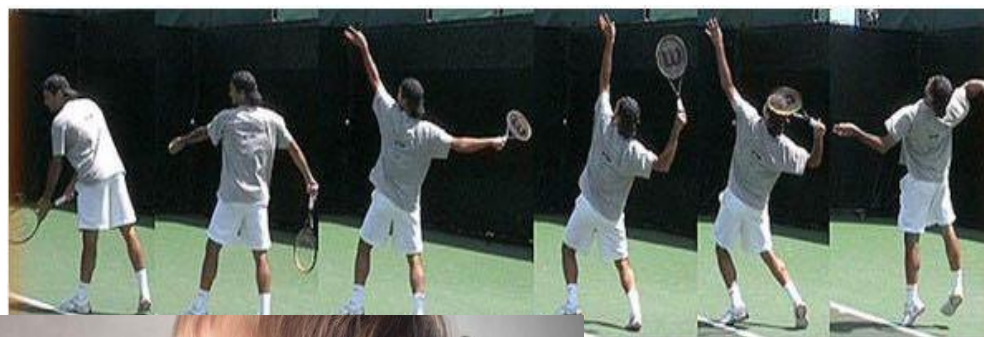
Universidad de Sevilla

# Contenido

- Modelos de secuencias
- Modelos para texto

# Modelos de secuencias

- Los datos suelen tener naturaleza de secuencia



WIKIPEDIA  
The Free Encyclopedia

Main page  
Contents  
Featured content  
Current events  
Random article  
Donate to Wikipedia  
Wikipedia store  
Interaction  
Help  
About Wikipedia  
Community portal  
Recent changes  
Contact page  
Tools

Article Talk

## Deep learning

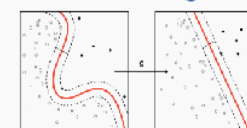
From Wikipedia, the free encyclopedia

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:

- 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 [\[show\]](#)

**Supervised learning** [\[show\]](#)  
(classification · regression)

**Clustering** [\[show\]](#)

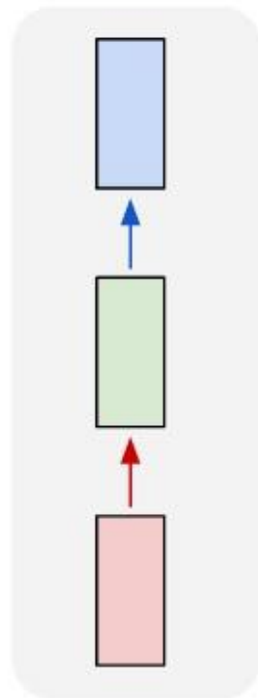
**Dimensionality reduction** [\[show\]](#)

# Modelos de secuencias

- Escenarios

		Output Sequential?	
Input Sequential?	no	no	yes
	no		?
yes	?	?	?

one to one



Ejemplos:

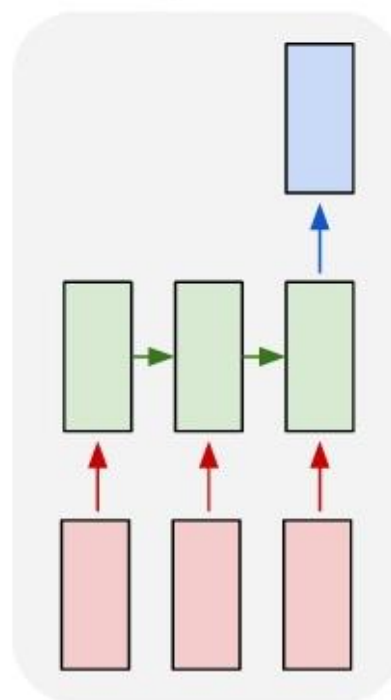
- Clasificación de imágenes

# Modelos de secuencias

- Escenarios

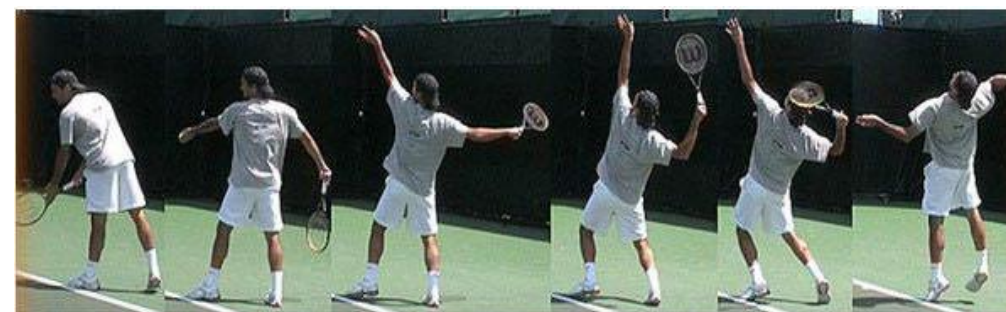
		Output Sequential?	
		no	yes
Input Sequential?	no		?
	yes	X	?

many to one



Ejemplos:

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

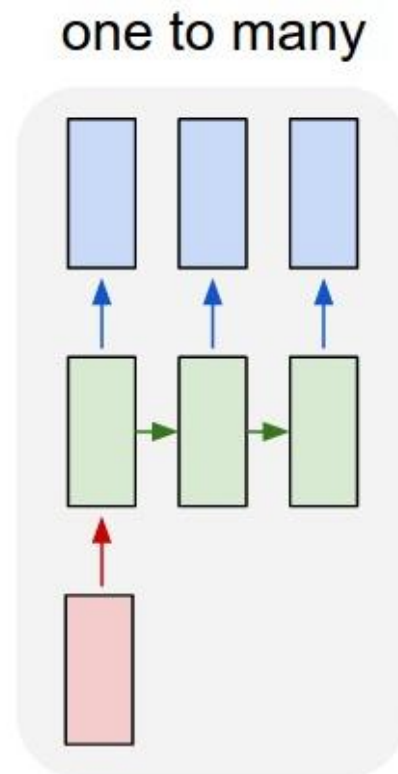


Jugando al tenis

# Modelos de secuencias

- Escenarios

		Output Sequential?	
		no	yes
Input Sequential?	no		X
	yes	X	?



Ejemplos:

- Image captioning



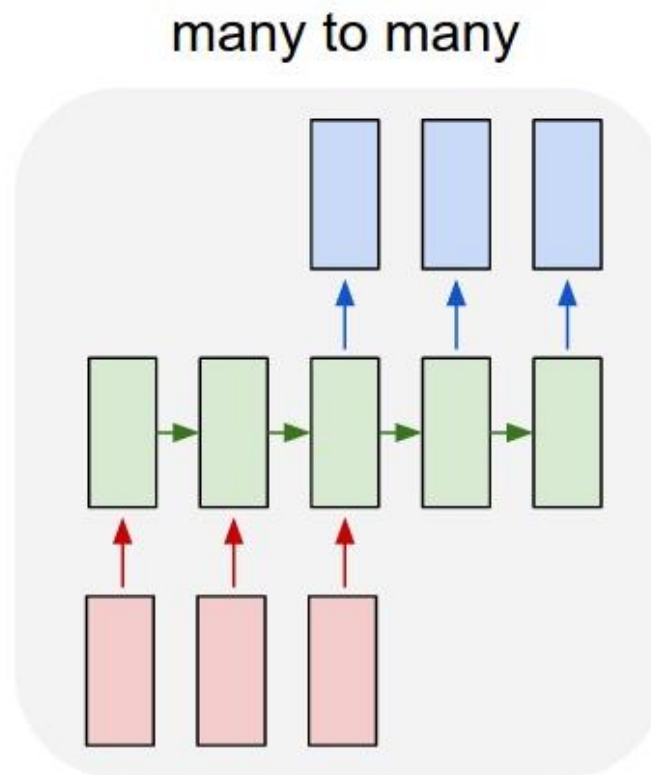
A square with a fountain and tall buildings in the background, with some trees and a few people hanging out.



# Modelos de secuencias

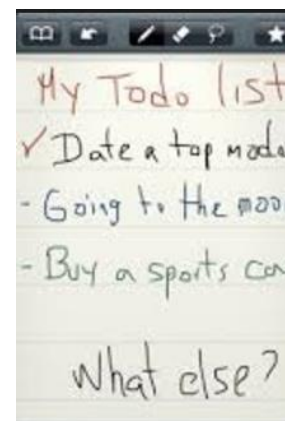
## • Escenarios

		Output Sequential?	
		no	yes
Input Sequential?	no		<b>X</b>
	yes	<b>X</b>	<b>X</b>



Ejemplos:

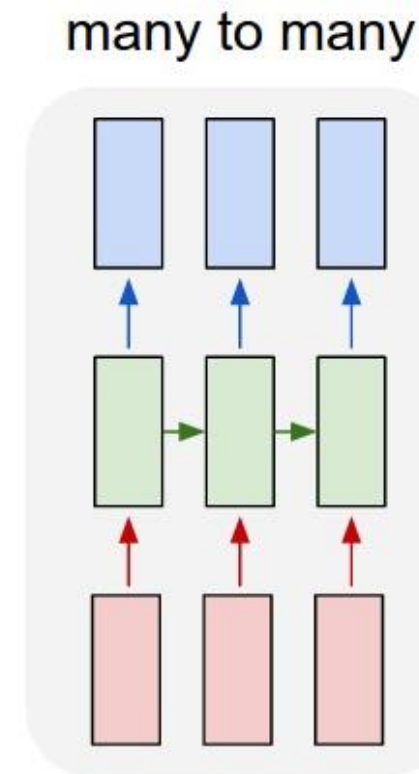
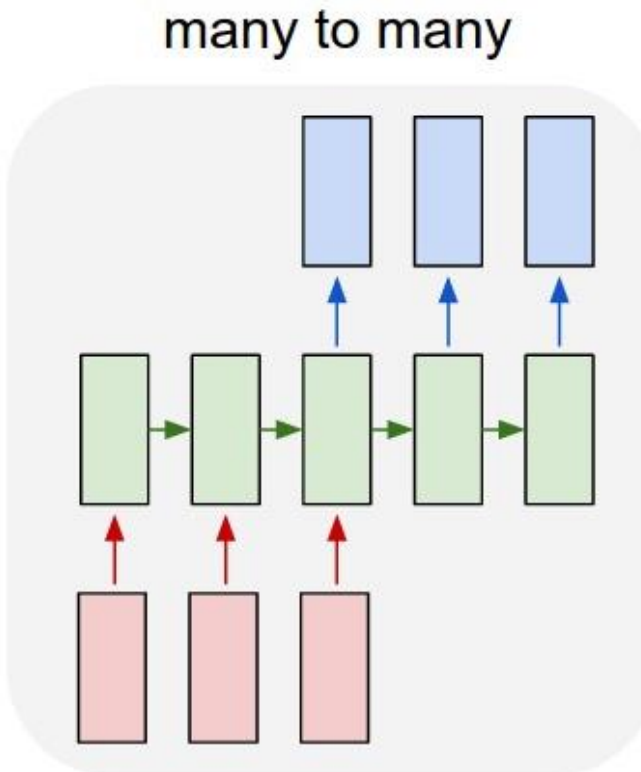
- Traducción máquina
- Resúmenes
- Reconocimiento del habla
- OCR
- Predicción de fotogramas en video



# Modelos de secuencias

- Escenarios

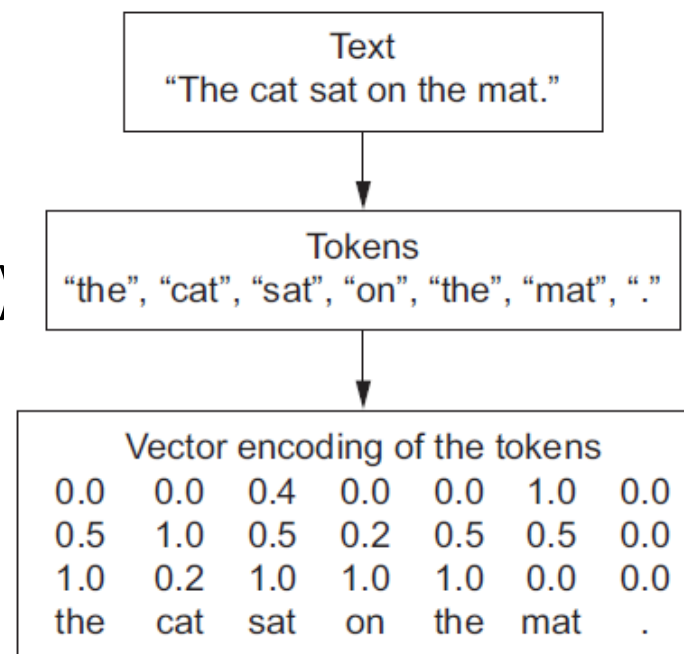
		Output Sequential?	
		no	yes
Input Sequential?	no		<b>X</b>
	yes	<b>X</b>	<b>X</b>





# 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”  $\rightarrow$  {“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

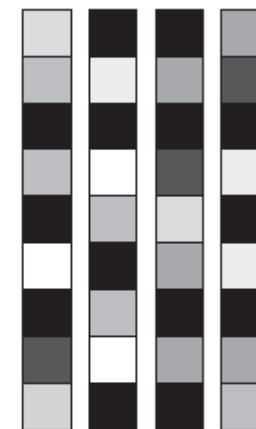
- Codificación de tokens en vectores:
  - **Token embedding (Word-embedding** para palabras, o **inmersión de palabras**)
    - Vectores de números reales (vs binario)
    - Menor dimensionalidad (256 vs 20.000)
    - Es posible calcular la distancia entre dos palabras.

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

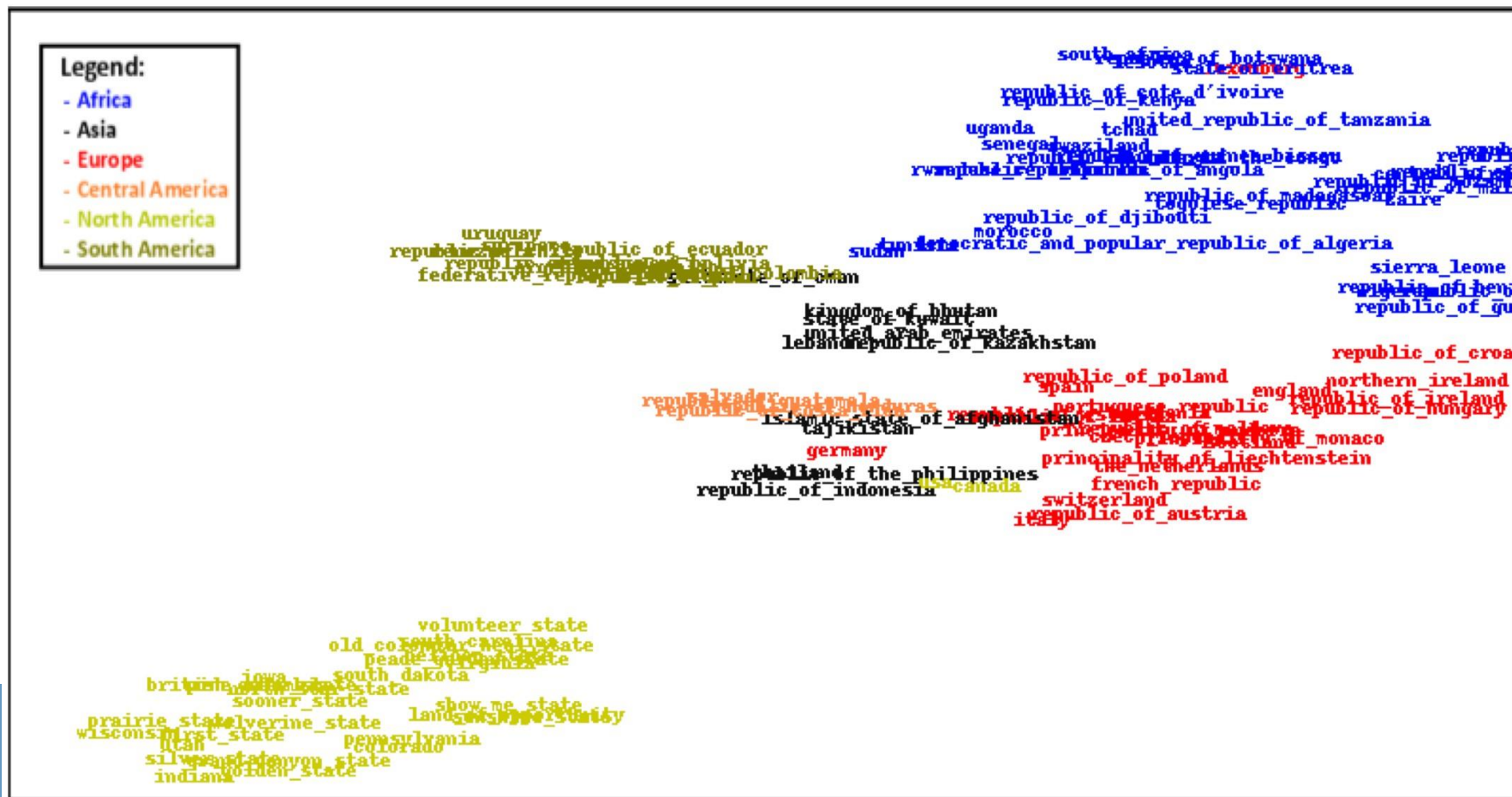


Word embeddings:

- Dense
- Lower-dimensional
- Learned from data

# Modelos para texto

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



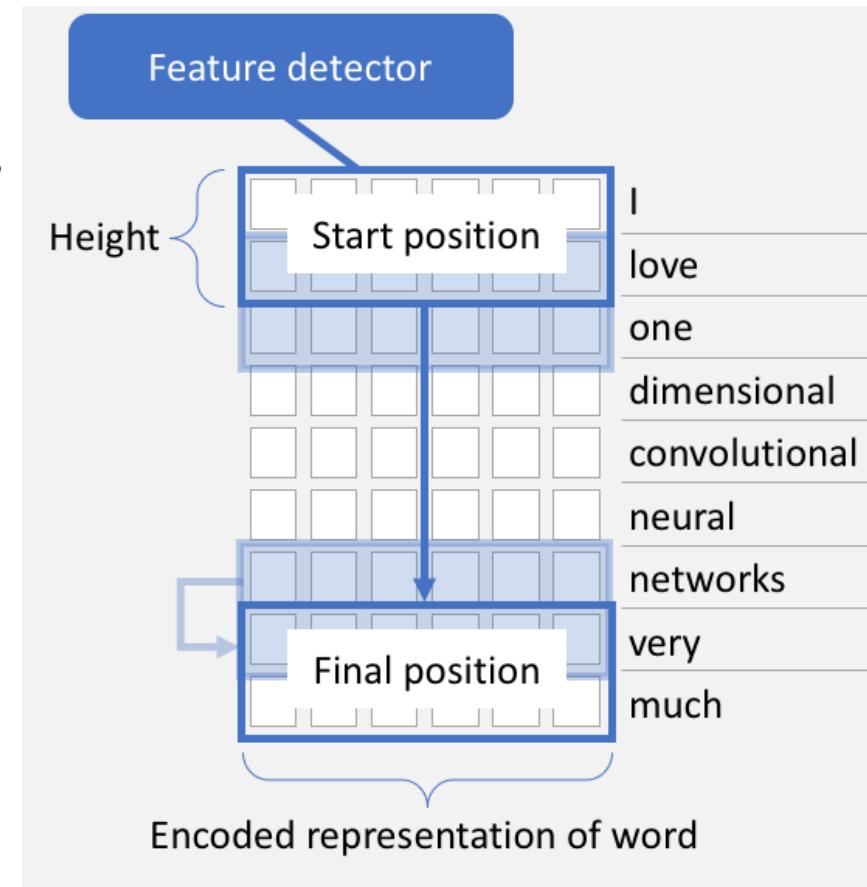
# Modelos para texto

- 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, Stanford), 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.