Redes Neuronales Recurrentes

Aprendizaje Automático
Aplicado

Agenda

- Redes Neuronales Recurrentes
- Arquitectura LSTM
- Embeddings
- Modelo BERT

Redes Neuronales Recurrentes

Introducción

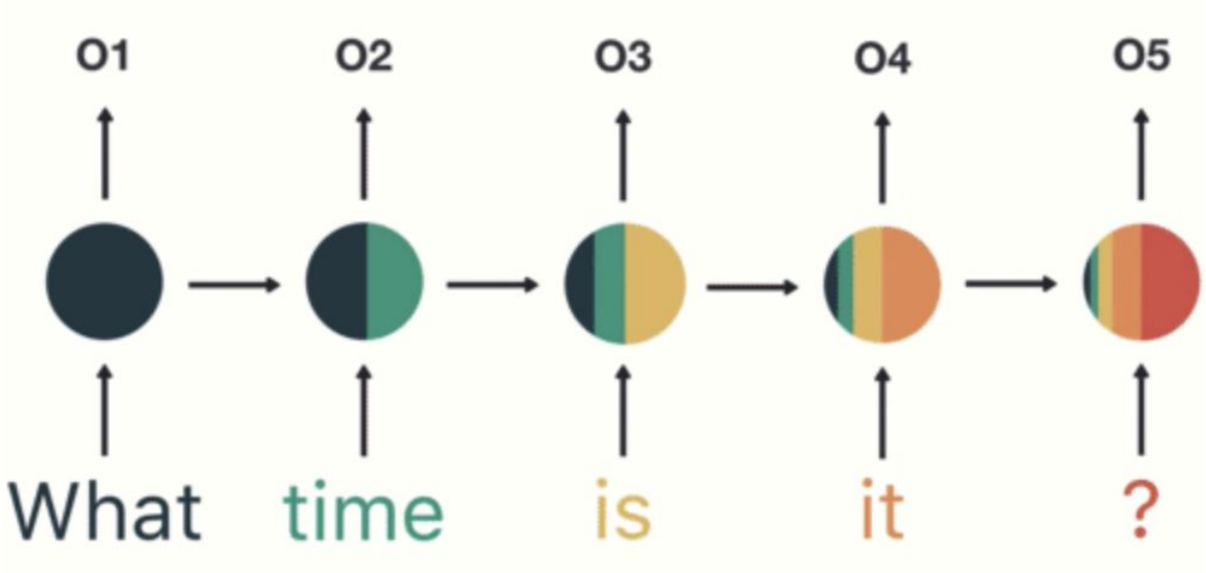
Redes neuronales tradicionales:

- Todas las entradas son independientes
- Carecen de memoria

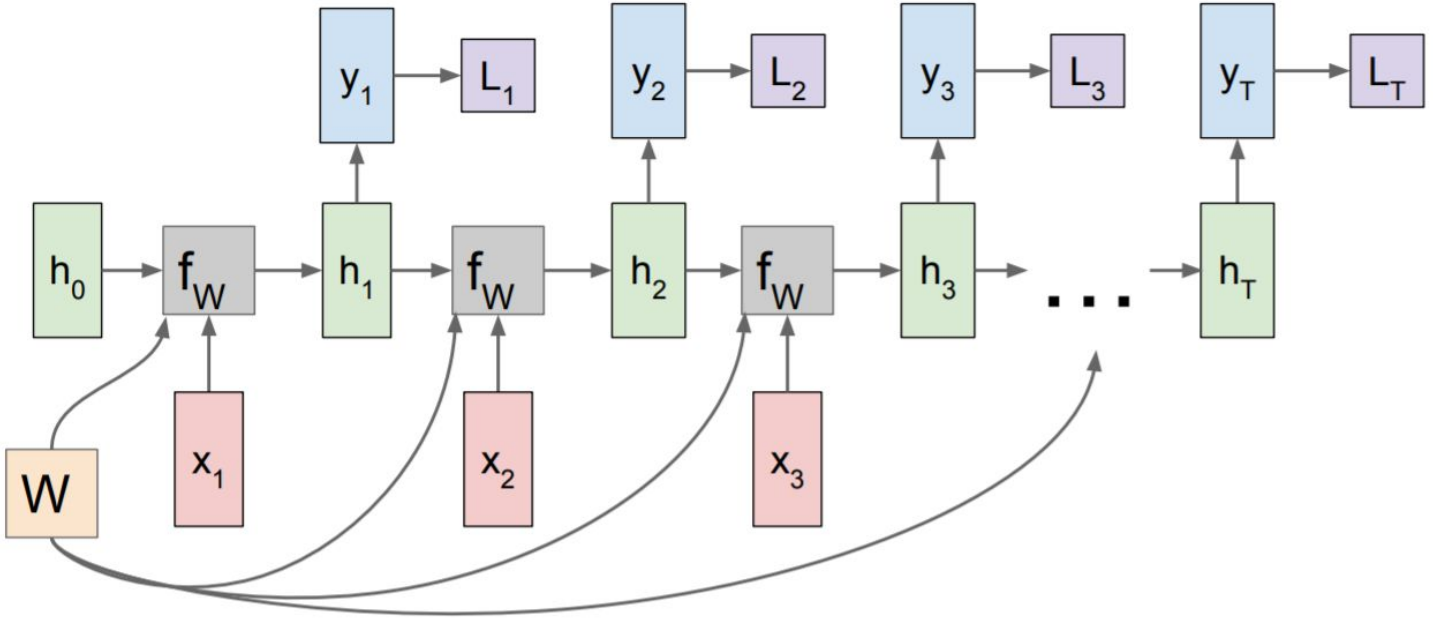
RNN:

- Entrada de tamaño variable
- Tamaño del modelo no se incrementa con la entrada
- Toma en cuenta información histórica (tienen memoria)

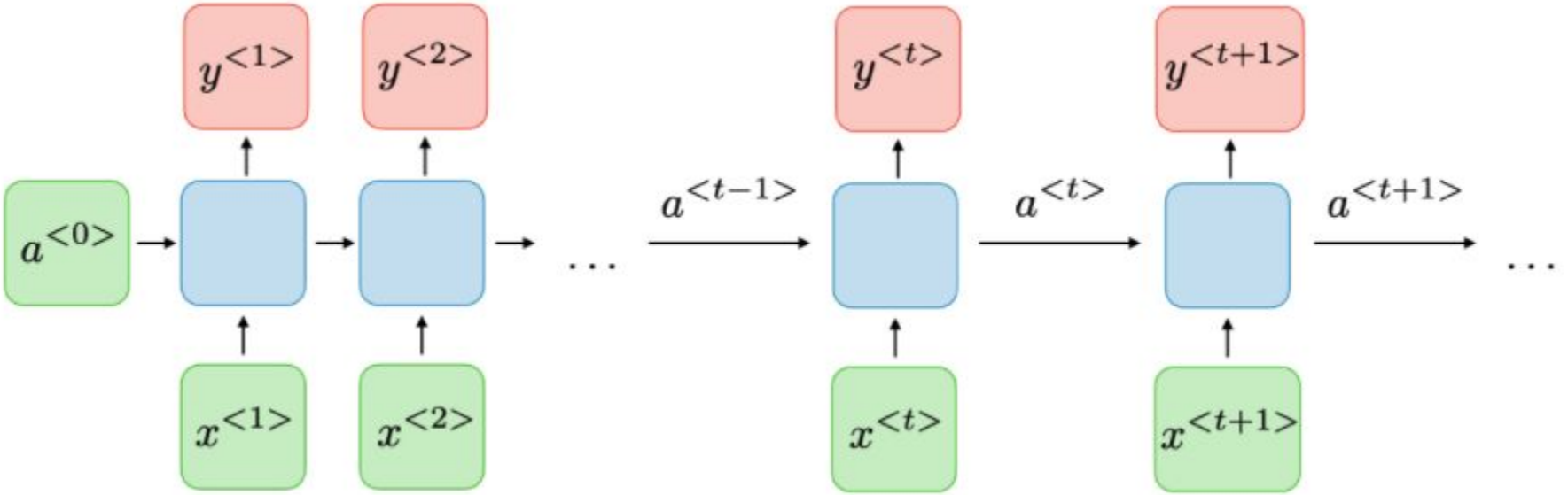
Introducción



Arquitectura

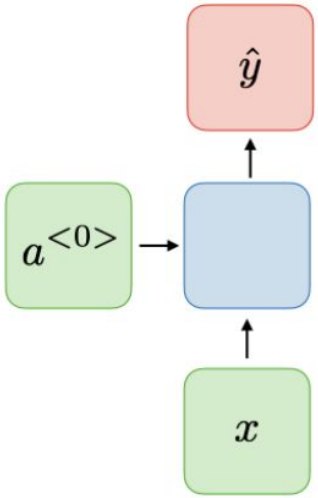


Arquitectura



Arquitectura

One to one



Arquitectura

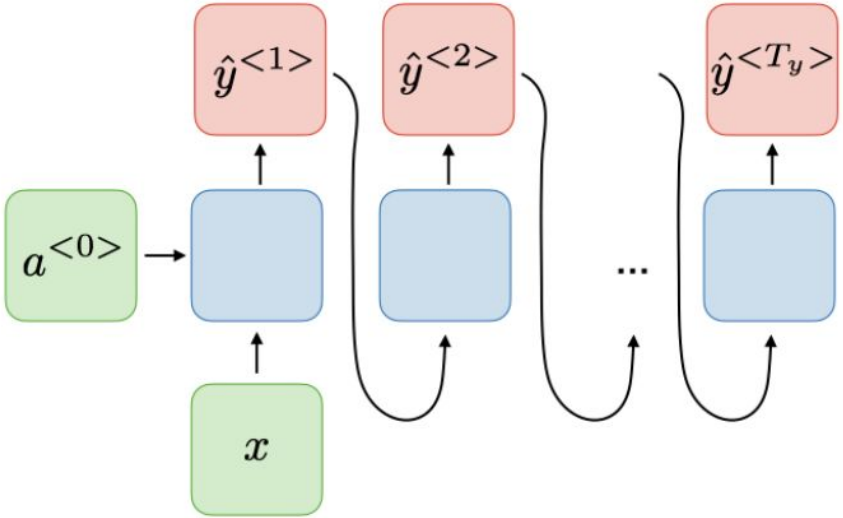
One to one

- Una entrada
- Una salida

Es la red neuronal neuronal tradicional!

Arquitectura

One to many



Arquitectura

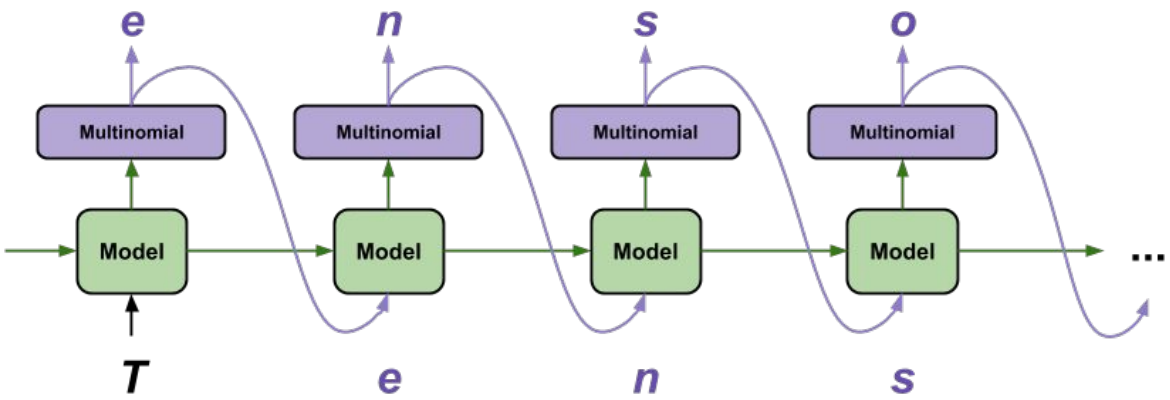
One to many

- Una entrada
- Múltiples salidas

Ejemplo: generación de texto

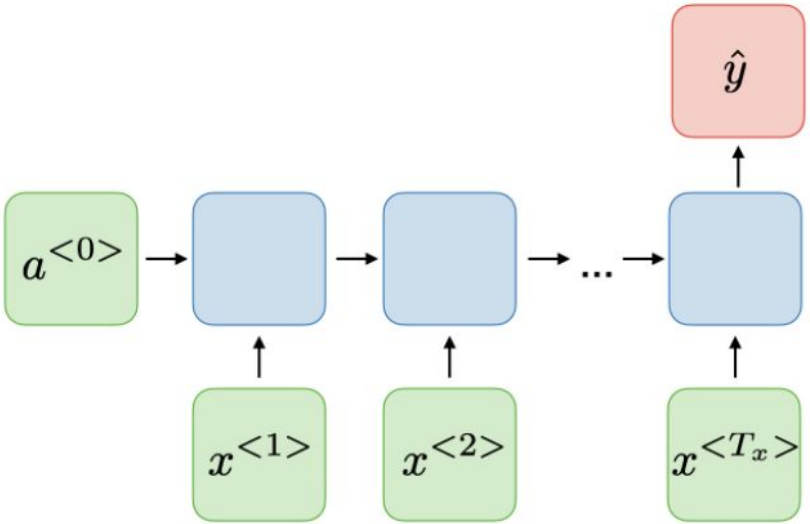
Arquitectura

Ejemplo: generación de texto



Arquitectura

Many to one



Arquitectura

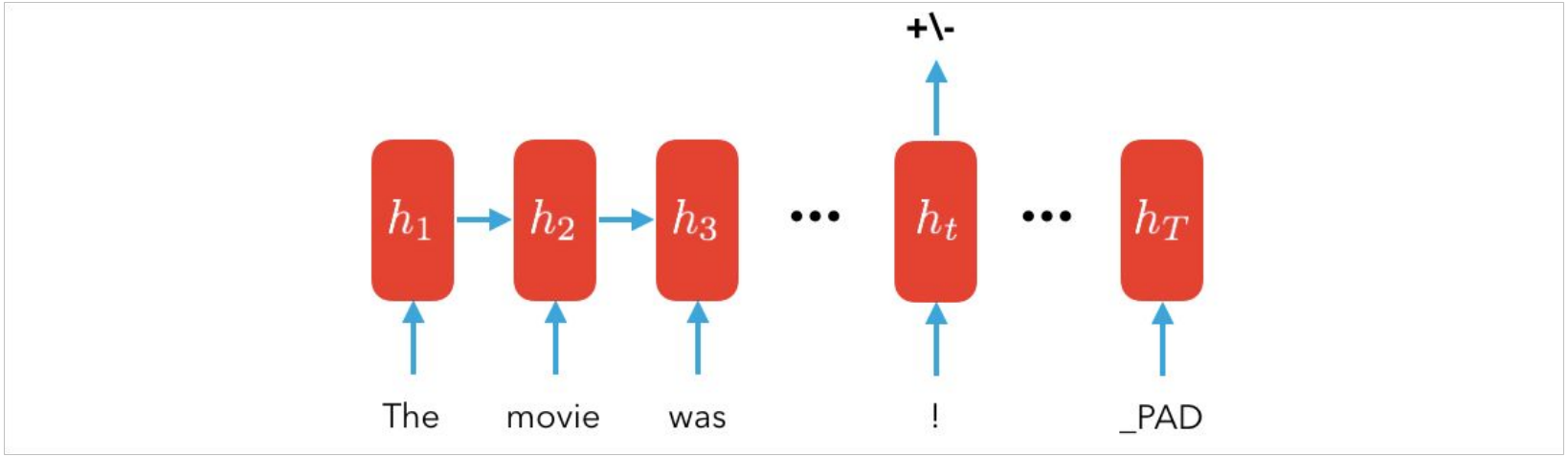
Many to one

- Múltiples entradas
- Una salida

Ejemplo: análisis de sentimiento

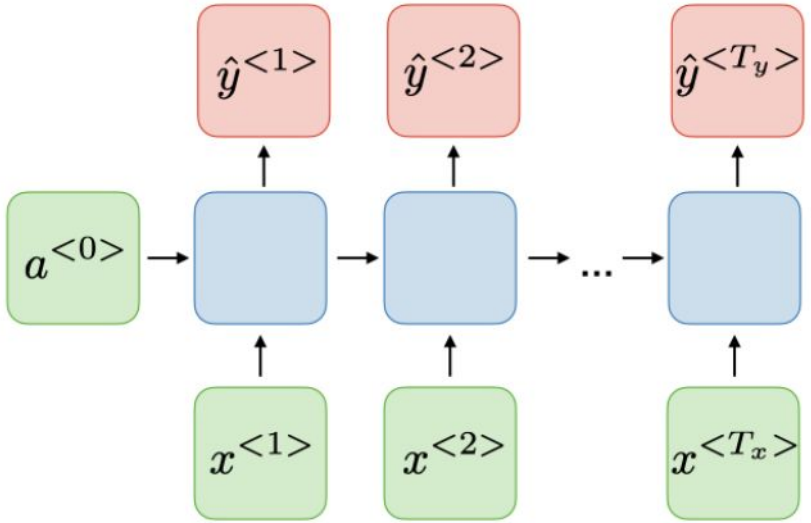
Arquitectura

Ejemplo: análisis de sentimiento



Arquitectura

Many to many (x=y)



Arquitectura

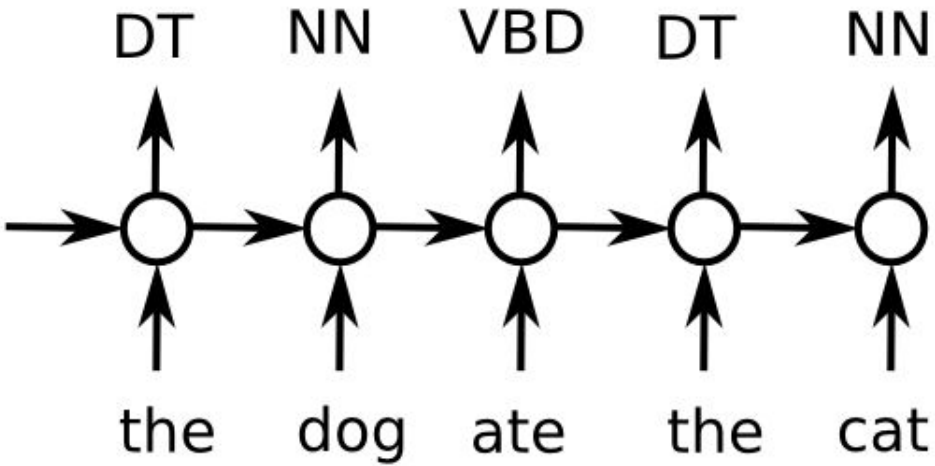
Many to many (x=y)

- Múltiples entradas
- Múltiples salidas (misma cantidad que entradas)

Ejemplo: etiquetado gramatical

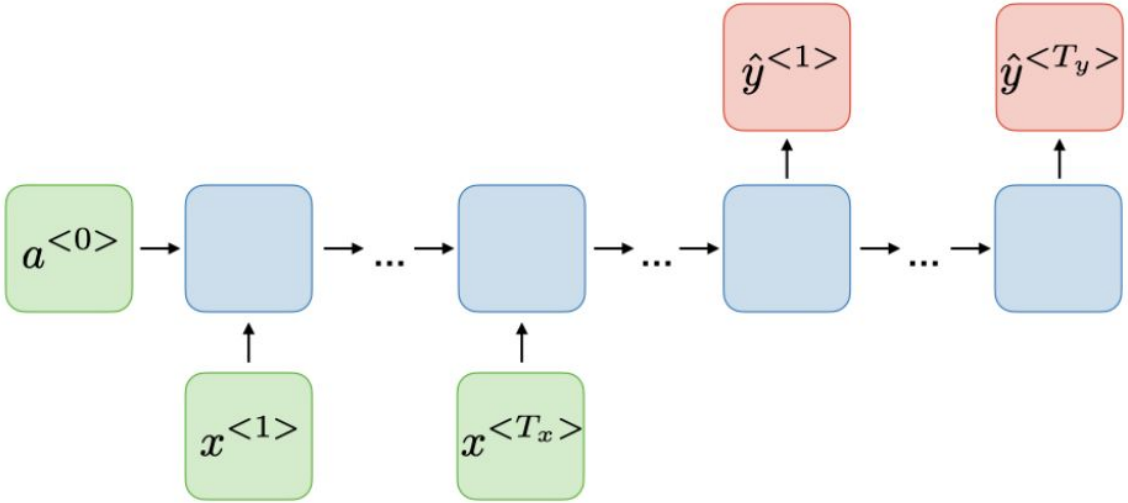
Arquitectura

Ejemplo: etiquetado gramatical



Arquitectura

Many to many (x≠y)



Arquitectura

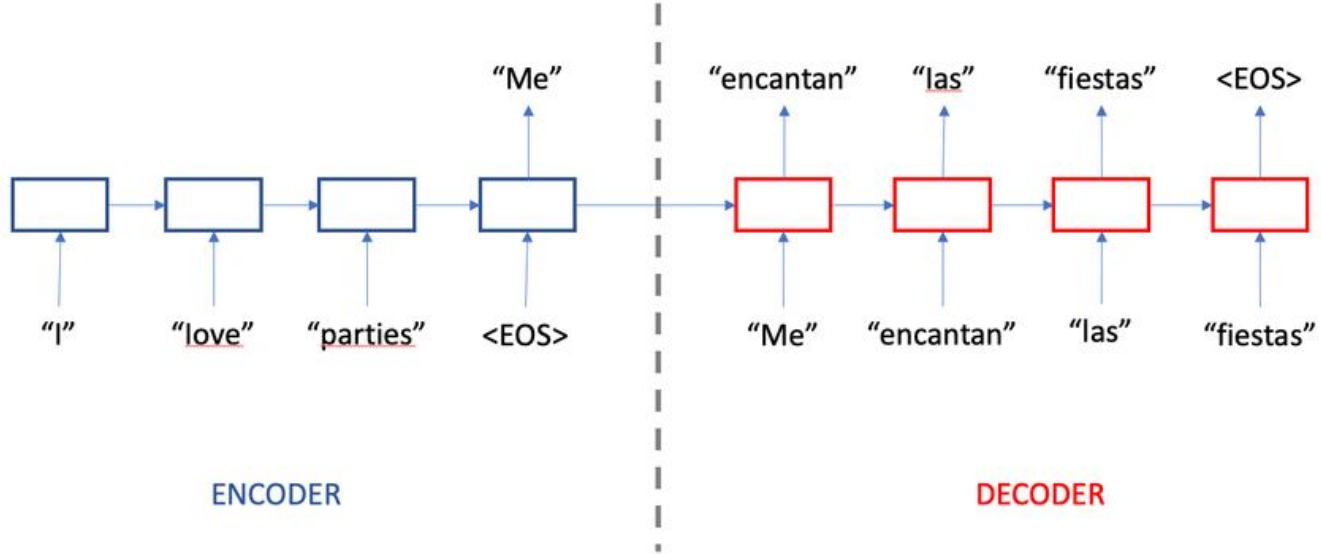
Many to many (x≠y)

- Múltiples entradas
- Múltiples salidas (no necesariamente la misma cantidad)

Ejemplo: traducción automática

Arquitectura

Ejemplo: traducción automática



Arquitectura LSTM

Introducción

- Es una RNN
- Trata los *vanishing gradients*
- Controlada por 3 *Gates*
- Mejor capacidad de memoria

Introducción

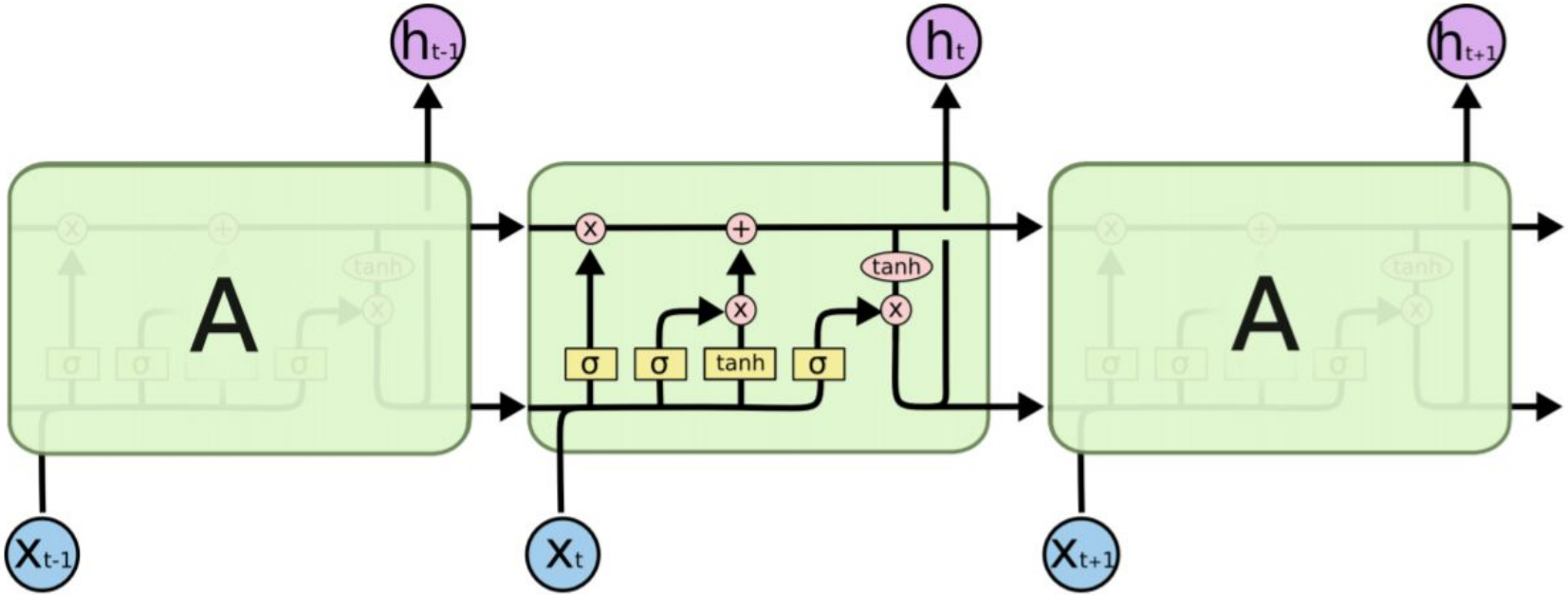
Cada estado oculto tiene 3 entradas

- Memoria actual
- Salida del paso anterior
- Entrada del paso actual

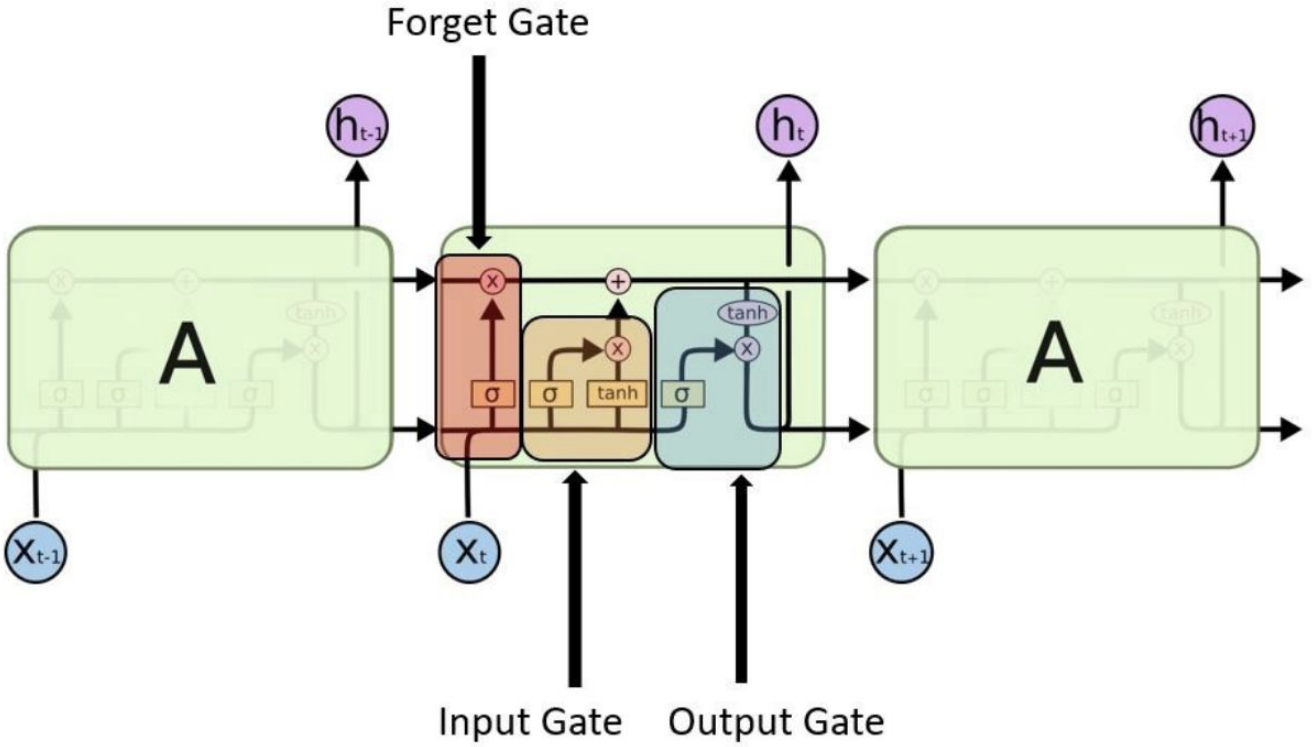
Y 2 salidas

- Memoria actualizada
- Salida del paso actual

Arquitectura



Arquitectura



Arquitectura

Forget Gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Decide qué datos serán olvidados

Arquitectura

Input Gate

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Decide cuál valor de la entrada será tenido en cuenta en memoria

Arquitectura

Output Gate

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Calcula la salida a partir de la entrada y la memoria

Arquitectura

Parámetros a aprender

- $W_f \rightarrow$
- $W_i \rightarrow$
- $W_C \rightarrow$
- $W_o \rightarrow$

- $b_f \rightarrow$
- $b_i \rightarrow$
- $b_C \rightarrow$
- $b_o \rightarrow$

Primer quiz!

Arquitectura

Parámetros a aprender

- $W_f \rightarrow (d_h+d_x)*d_h$
- $W_i \rightarrow (d_h+d_x)*d_h$
- $W_C \rightarrow (d_h+d_x)*d_h$
- $W_o \rightarrow (d_h+d_x)*d_h$

- $b_f \rightarrow d_h$
- $b_i \rightarrow d_h$
- $b_C \rightarrow d_h$
- $b_o \rightarrow d_h$

$4 * (d_h + d_x + 1) * d_h$

Quiz time!

1- ¿Qué aportan las RNN a las redes neuronales tradicionales?

2- ¿Qué aportan las LSTM a las RNN?

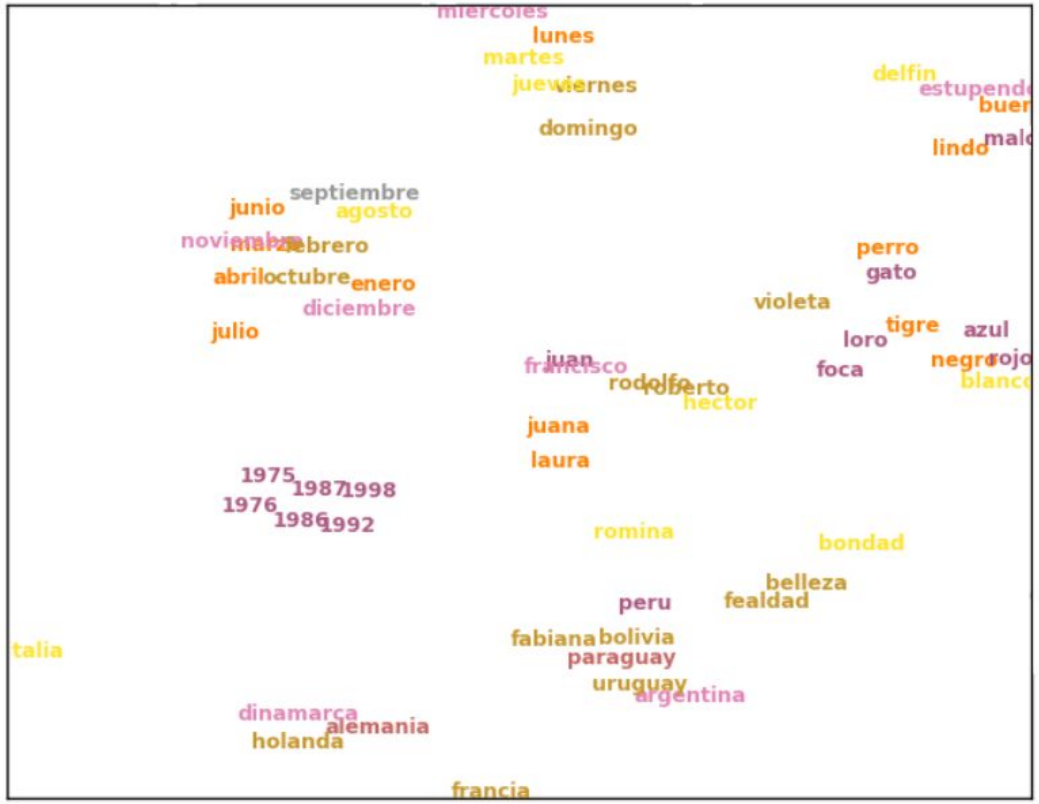
3- ¿Qué modelo utilizarían para reconocimiento de escritura?

Embeddings

Introducción

- Representación vectorial de palabras
- Orientada por similaridad
- Palabras similares ocurren en contextos similares
- Los contextos en los que ocurre una palabra son informativos respecto a su significado

Introducción



Ejemplo

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

Ejemplo

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

Ejemplo

...dig a [hole. The	**car**	drove away] leaving behind ...
...to directly [drive the	**car**	wheel angle] 3. Force ...
...celebrity status, [drove fast	**cars**	and partied] with some ...
...but there [are police	**cars**	that chase] you. Each ...
...world of [money, fast	**cars**	and excitement] and, under ...
...to pet [the family's	**cat**	and dog,] who tended ...
...and then [wanted a	**cat**	to eat] the many ...
...murmur is [detectable. The	**cat**	often eats] and drinks ...
...behaviour of [a domestic	**cat**	playing with] a caught ...
...have never [seen a	**cat**	eat so] little and ...
...bank, children [playing with	**dogs**	and a] man leading. ...
...sure you [encourage your	**dog**	to play] appropriate chase ...
...Truth, Lord: [yet the	**dogs**	eat of] the crumbs ...
...vegetable material [and enzymes.	**Dogs**	also eat] fruit, berries ...
...hubby once [ate the	**dog**	food and] asked for ...
...were back [at the	**van**	and drove] down to ...
...go down [as the	**van**	drove off.] As he ...
...heavy objects, [driving transit	**vans**	, wiring plugs] and talking ...
...of the [fast food	**van**	being located] outside their ...
...each of [the six	**van**	wheels , and] also under ...

(Ejemplo tomado de la charla de Alessandro Lenci en la Global WordNet Conference (GWC 2014).)

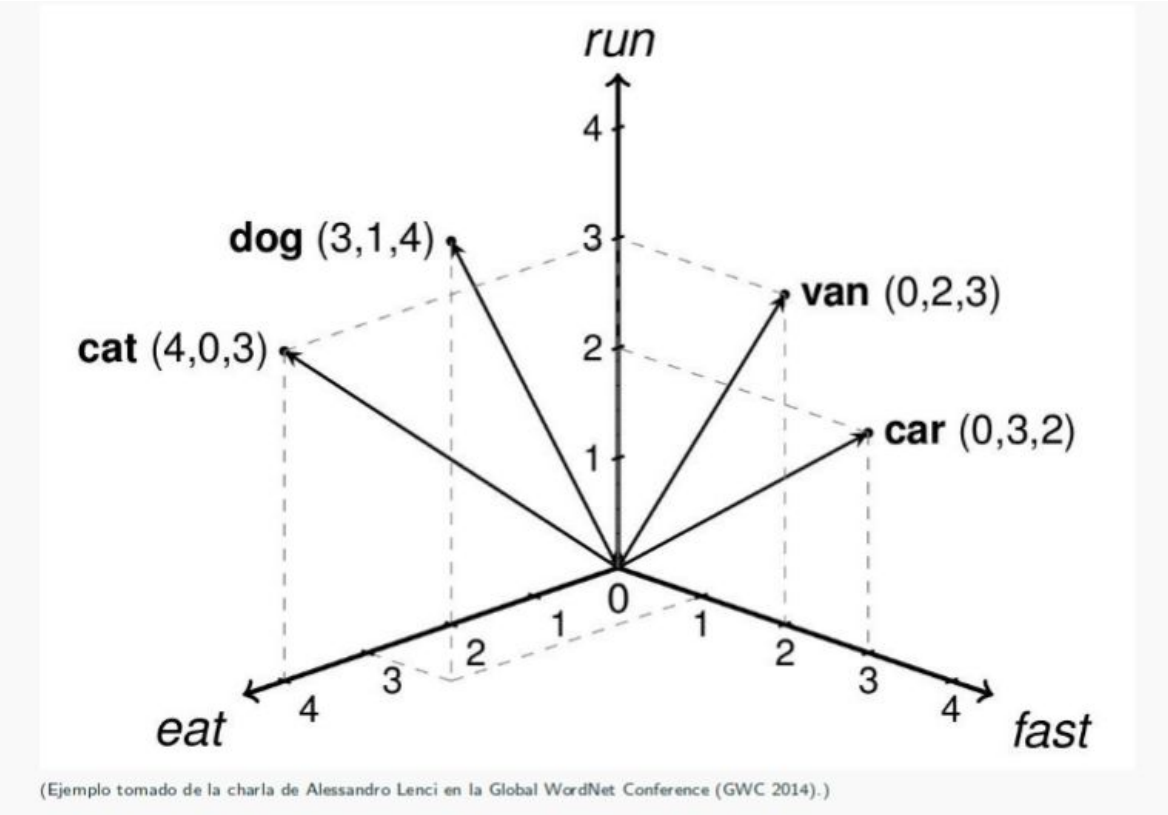
Ejemplo

	dog	*drive*	*eat*	*fast*	*play*	...	*the*	*wheel*
car	0	3	0	2	0	⋮	2	1
cat	1	0	3	0	1	⋮	2	0
dog	0	0	3	0	2	⋮	2	0
van	0	3	0	1	0	⋮	3	1

co-occurrence matrix

(Ejemplo tomado de la charla de Alessandro Lenci en la Global WordNet Conference (GWC 2014).)

Ejemplo



(Ejemplo tomado de la charla de Alessandro Lenci en la Global WordNet Conference (GWC 2014).)

tf-idf

tf: Frecuencia del término

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(t,d) : t \in d\}}$$

idf: Frecuencia inversa en documentos

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

tf-idf

Para comparar dos palabras:

- tf-idf de matriz palabra-palabra
- coseno de los vectores

PMI

Pointwise Mutual Information

$$\mathrm{PMI}(X,Y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- es una medida de la asociación entre dos palabras.
- se usa PPMI

Vectores dispersos

- Casi todos lo valores son ceros
- Demasiado grandes: dimensión de 20.000 a 50.000

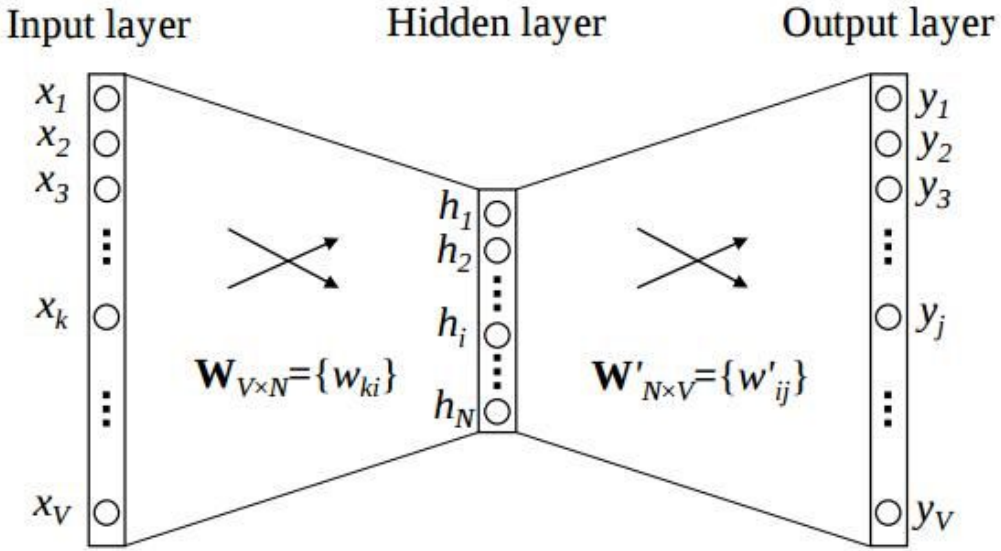
Vectores densos

- Dimensión entre 50 y 1000
- La mayoría de sus valores son distintos de cero
- Menos pesos para entrenar
- Generalizan mejor

Word2Vec

- Entrenamiento rápido
- Disponible en la web
- Predecir en lugar de contar
- Probabilidad de que una palabra esté en el contexto de la otra

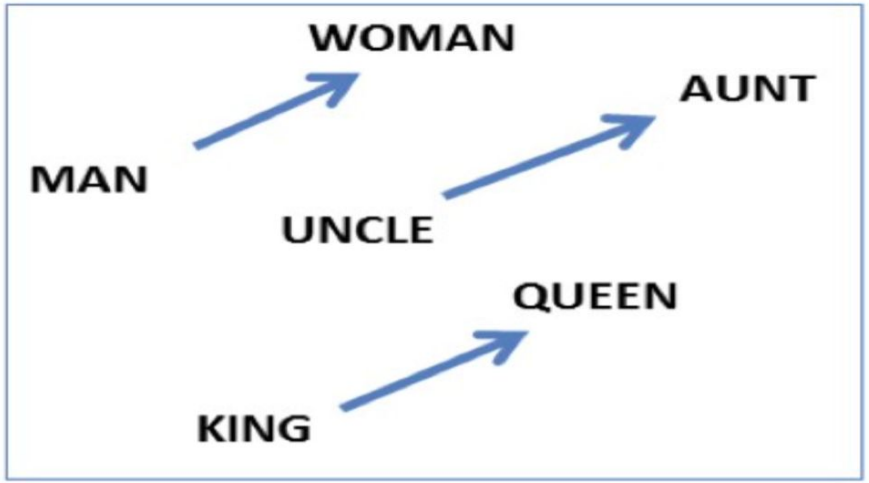
Word2Vec

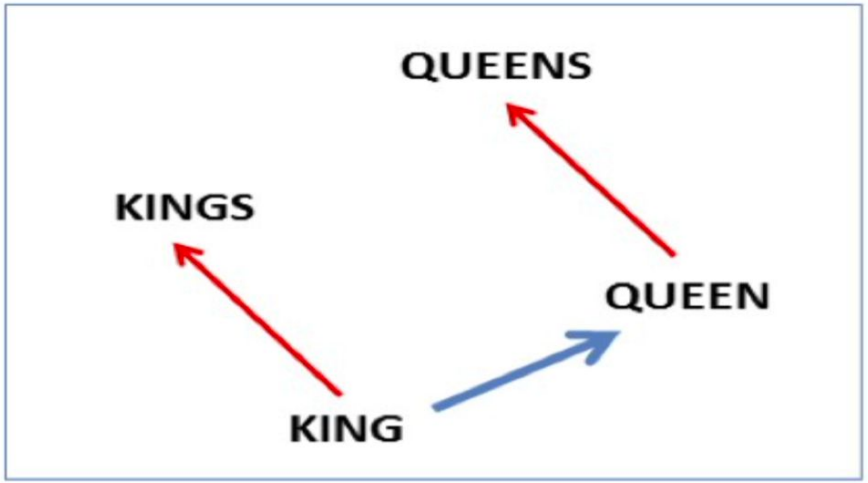

Input layer
Hidden layer
Output layer
x_1
x_2
x_3
x_k
x_V
h_1
h_2
h_i
h_N
y_1
y_2
y_3
y_j
y_V
$\mathbf{W}_{V\times N}=\{w_{ki}\}$
$\mathbf{W}'_{N\times V}=\{w'_{ij}\}$

Skip-gram

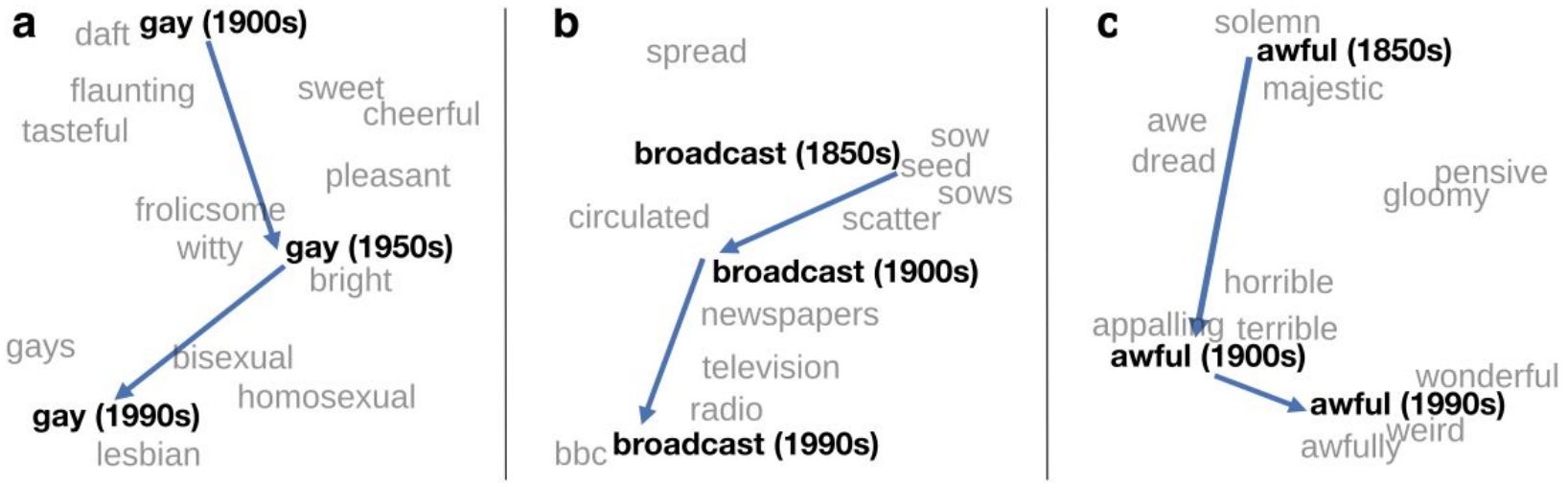
- Variante de Word2Vec
- Contexto: ventana de tamaño 2 a 10
- Ejemplos negativos: se generan aleatoriamente
- Clasificador binario: regresión logística
- Los pesos calculados en la capa oculta son los elementos del vector

Analogías





Tiempo


a
daft
gay (1900s)
flaunting
sweet
cheerful
tasteful
pleasant
frolicsome
witty
gay (1950s)
bright
gays
bisexual
homosexual
gay (1990s)
lesbian
b
spread
broadcast (1850s)
sow
seed
sows
circulated
scatter
broadcast (1900s)
newspapers
television
radio
bbc
broadcast (1990s)
c
solemn
awful (1850s)
majestic
awe
dread
pensive
gloomy
horrible
appalling
terrible
awful (1900s)
wonderful
awful (1990s)
weird
awfully

Quiz time!

1- ¿Qué diferencias hay entre los vectores densos y dispersos?

2- ¿Cómo se obtienen los vectores densos?

3- Se quiere calcular los word embeddings (vectores dispersos) a partir de 1.000 documentos con un vocabulario de 20.000 palabras. ¿Qué dimensión tendrá el vector correspondiente a una palabra?

Modelo BERT

Introducción

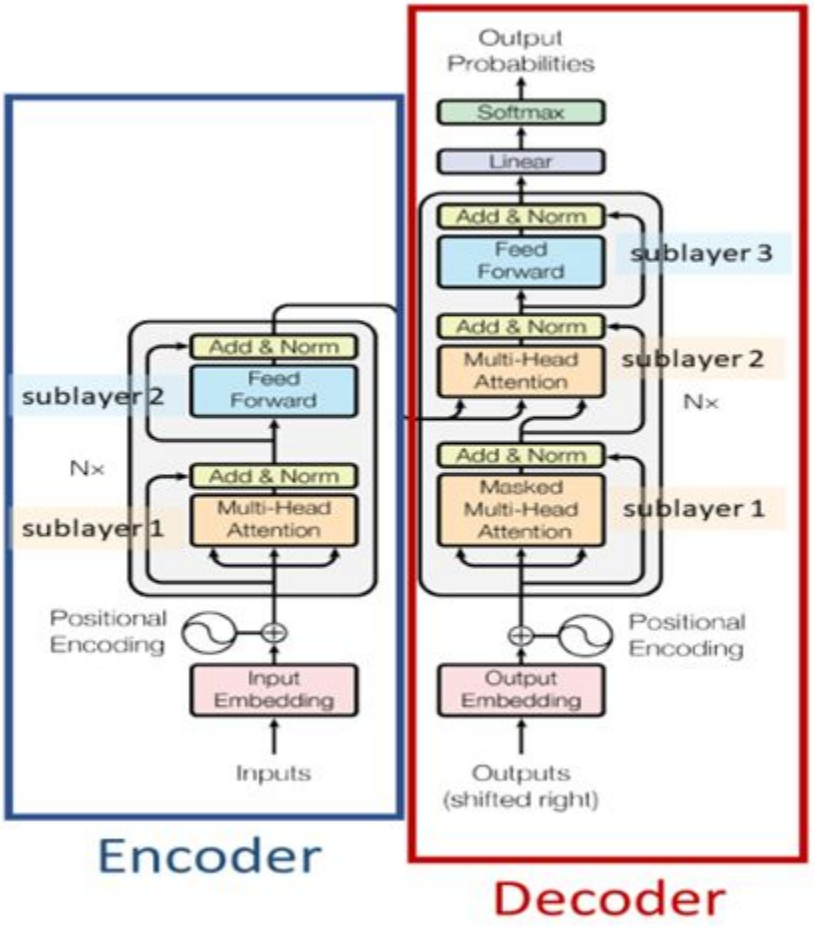
Modelos del lenguaje

- Mucha cantidad de texto disponible
- Tareas específicas de NLP requieren texto anotado
- Cantidad de texto anotado es mucho menor
- Se entrena un modelo del lenguaje y se realiza *fine tuning*

Transformers

- Arquitectura Encoder-Decoder
- Embeddings de posición
- Mecanismo de Attention

Transformers



BERT

Bidirectional Encoder Representations from Transformers

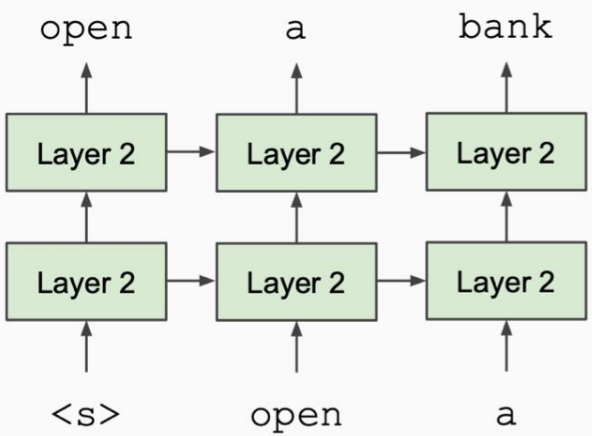
- Lee la entrada de forma bidireccional
- Utiliza sólo el Encoder de Transformers

Objetivo: Generar un modelo del lenguaje

BERT

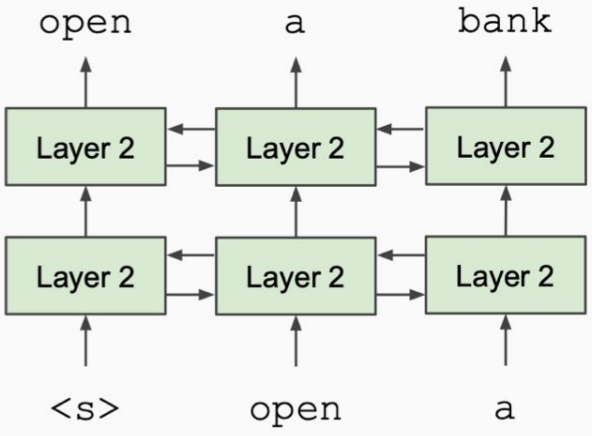
Unidirectional context

Build representation incrementally



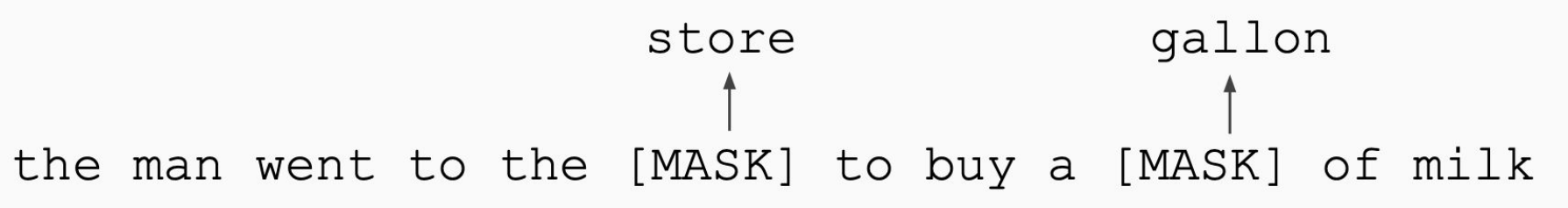
Bidirectional context

Words can “see themselves”



Masked LM

Se enmascaran palabras de la entrada (normalmente 15%) y se entrena para predecirlas.



Next Sentence Prediction

Se entrena para predecir cuando una oración procede a otra

Sentence A = `The man went to the store.`
Sentence B = `He bought a gallon of milk.`
Label = `IsNextSentence`

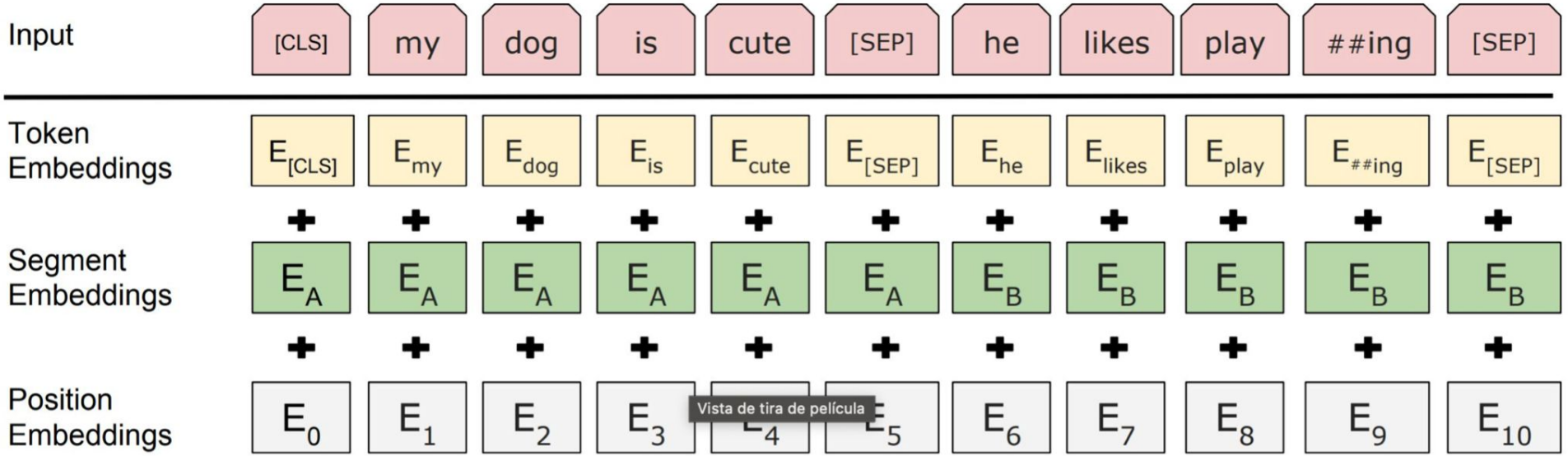
Sentence A = `The man went to the store.`
Sentence B = `Penguins are flightless.`
Label = `NotNextSentence`

Representación

Entrada:

- Embedding del token
- Embedding de la oración
- Embedding de posición

Representación



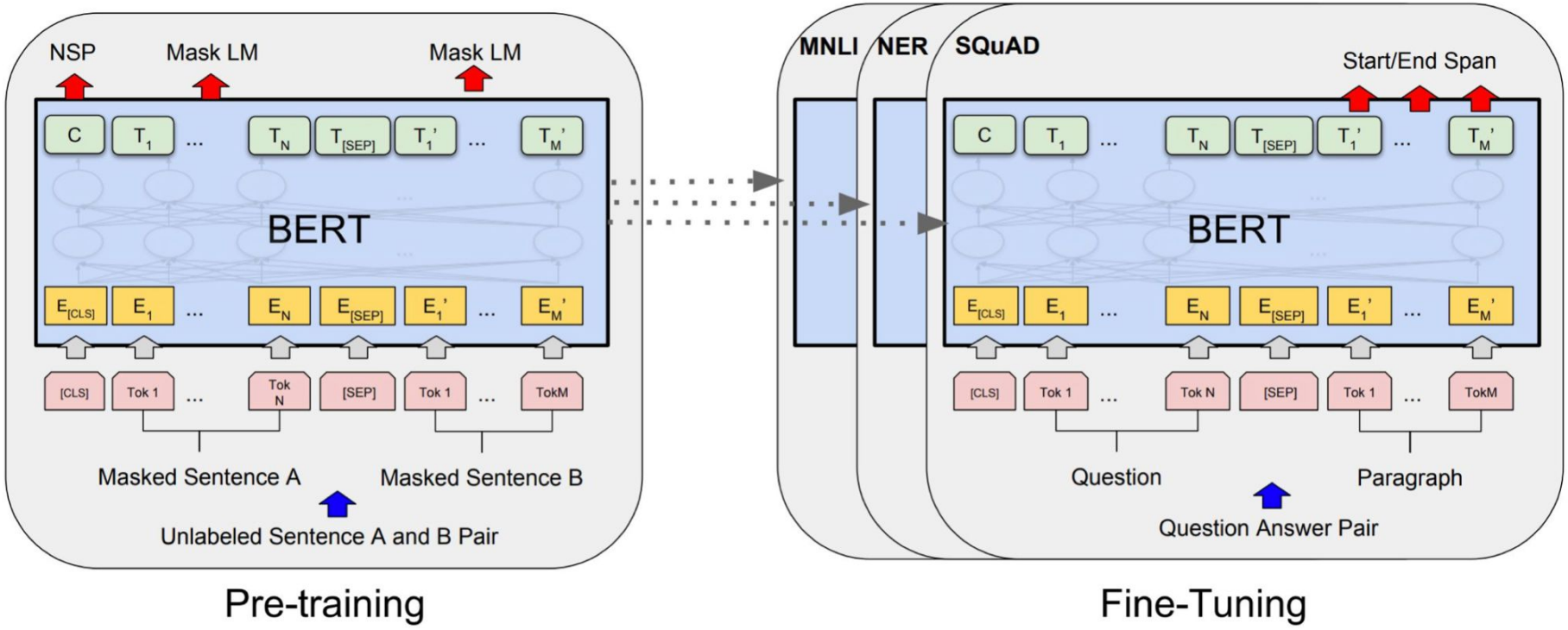
Fine tuning

BERT puede utilizarse para muchas tareas distintas.

Para estas tareas específicas se de debe agregar una capa extra encima del modelo.

La mayoría de los hiperparámetros se mantienen iguales.

Fine tuning


NSP
Mask LM
Mask LM
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair
Pre-training
MNLI
NER
SQuAD
Start/End Span
Question
Paragraph
Question Answer Pair
Fine-Tuning

Aplicaciones

- Búsqueda de respuestas
- Resúmen automático
- Generación de respuestas (conversaciones)
- Desambiguación de palabras
- Reconocimiento de entidades con nombre

Laboratorio