ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE MADRID

Embeddings

Eduardo Fernández Carrión

Embeddings space

Es un **espacio vectorial** utilizado en NLP para representar palabras o frases (entidades).

- Los embeddings transforman entidades a vectores de alta dimensionalidad, donde **cada dimensión representa una característica latente** que puede o no tener un significado interpretable.
- La cercanía o lejanía entre dos vectores en el espacio de embeddings puede interpretarse como **similitud o disimilitud** entre las entidades que representan.
- Los embeddings **suelen ser aprendidos de grandes corpus** de texto.

Puede considerarse uno de los principales avances del Deep Learning en problemas complejos de NLP.

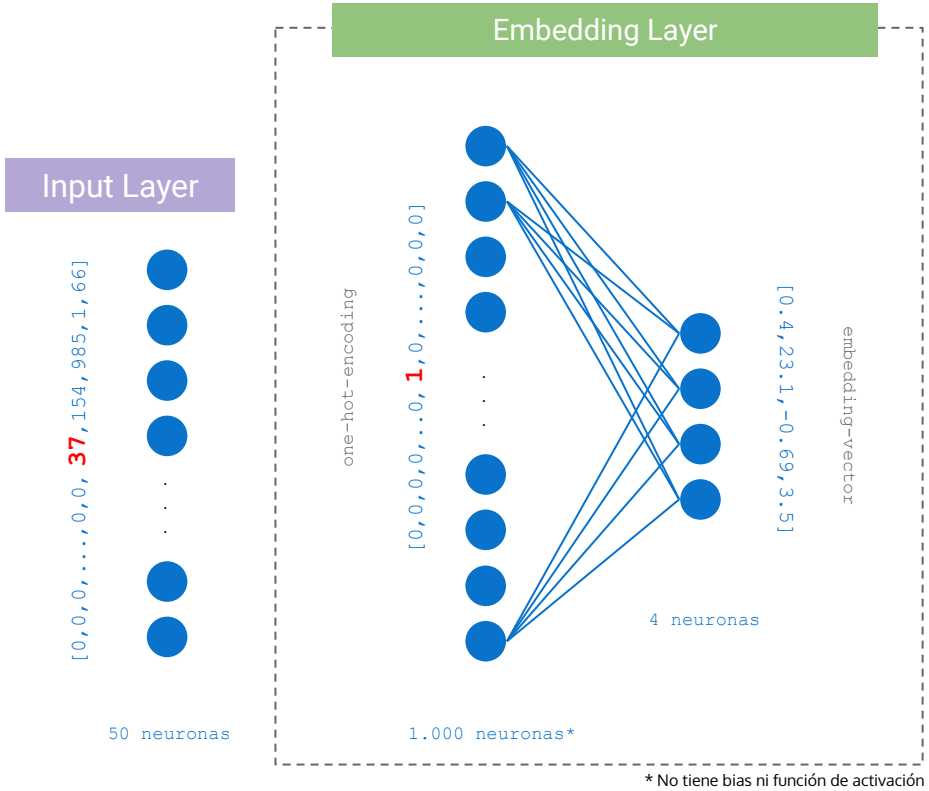
Word embeddings

Vocabulary size = 1.000

Truncating / Padding → sequences 50

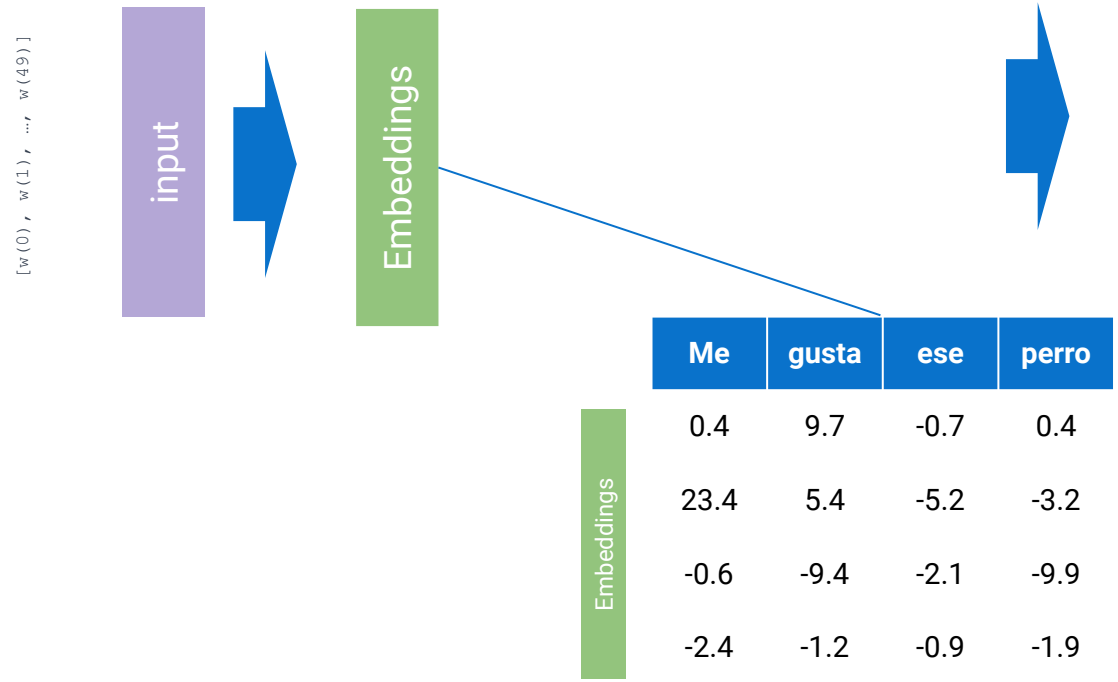
4-dimensional Embedding Layer

```
sample_sequence
[0,0,0,...,0,0,37,154,985,1,66]
```



* No tiene bias ni función de activación

Word embeddings



resto de la arquitectura de la red neuronal

(clasificación, regressión, serie temporal...)

Word embeddings

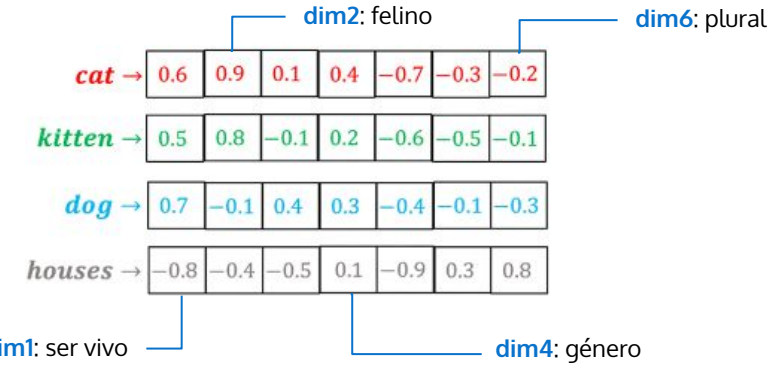
Sin embargo, el concepto clave de *Word Embeddings* es localizar palabras que comparten contextos comunes en el corpus de entrenamiento, muy próximas en un espacio vectorial en comparación con otras.

> *Word Embedding* es capaz de capturar el **contexto**, la **similitud semántica** y **sintáctica** (género, sinónimos, etc.) de una palabra **reduciendo su dimensión***

Esto podría ser un vector de un espacio de *Word Embeddings* de dimensión 7 que una red neuronal ha entrenado.

cat →	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
kitten →	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
dog →	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses →	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8

La red neuronal (sin entender qué podría significar cada dimensión) ha encontrado un ajuste que incluso podríamos interpretar.



*si hubiésemos creado un espacio vectorial con OneHotEncoder de cada palabra del diccionario, tendríamos un espacio vectorial enorme

Keras layer

Embedding layer

`Embedding` class [source]

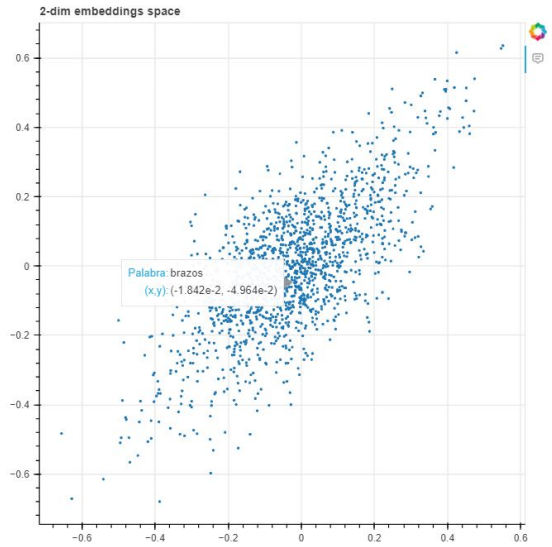
```
keras.layers.Embedding(
    input_dim,
    output_dim,
    embeddings_initializer="uniform",
    embeddings_regularizer=None,
    embeddings_constraint=None,
    mask_zero=False,
    lora_rank=None,
    **kwargs
)
```

Example

```
>>> model = keras.Sequential()
>>> model.add(keras.layers.Embedding(1000, 64, input_length=10))
>>> # The model will take as input an integer matrix of size (batch,
>>> # input_length), and the largest integer (i.e. word index) in the input
>>> # should be no larger than 999 (vocabulary size).
>>> # Now model.output_shape is (None, 10, 64), where `None` is the batch
>>> # dimension.
>>> input_array = np.random.randint(1000, size=(32, 10))
>>> model.compile('rmsprop', 'mse')
>>> output_array = model.predict(input_array)
>>> print(output_array.shape)
(32, 10, 64)
```

Word embeddings

Cada **palabra se asigna a un vector** y **los valores del vector se aprenden** mediante redes neuronales que tienden a agrupar palabras de significado similar en dicho espacio de dimensiones predefinidas.



Paso 1:

- Creación del diccionario de palabras
- Creación del espacio de embeddings *(decidimos nosotros el número de dimensiones latentes)*

Paso 2:

- Inicializamos (*de forma aleatoria)* la proyección del diccionario en el espacio de embeddings

Paso 3:

- Creamos una red neuronal que involucre el espacio de embeddings como una de sus capas

Paso 4:

- Entrenamos la red neuronal *(debe resolver un problema supervisado o no supervisado)*

Distancia del coseno

La distancia del coseno es una medida de similitud entre dos vectores no nulos en un espacio con producto interior que se utiliza a menudo para comparar documentos en el procesamiento de texto.

$$\text{Similitud del coseno}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

La similitud del coseno varía entre -1 y 1:

- Un valor de 1 indica que los dos vectores son idénticos en orientación.
- Un valor de 0 indica que los vectores son ortogonales (independientes).
- Un valor de -1 indica que los vectores son diametralmente opuestos.

La distancia del coseno es una métrica que pondera la **cercanía de dos puntos en un espacio vectorial** → La usamos en NLP para ver cuán similares son dos palabras

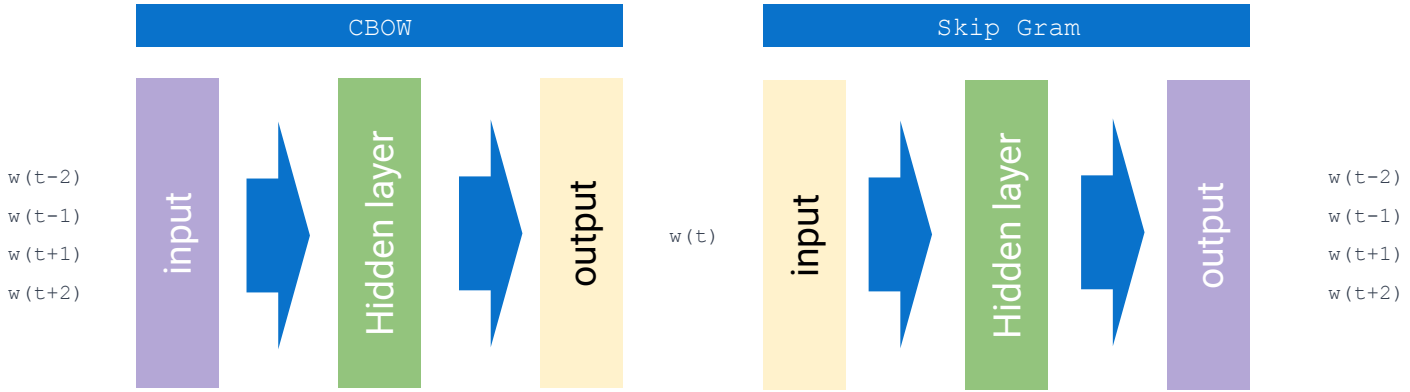
Esta fórmula viene del teorema de coseno:

$$c^2 = a^2 + b^2 - 2ab \cdot \cos(C) \quad \Rightarrow \quad A \cdot B = \|A\|\|B\| \cos(\theta)$$

Word2Vec algorithm (2013)

Es una técnica para crear *Word Embeddings*.

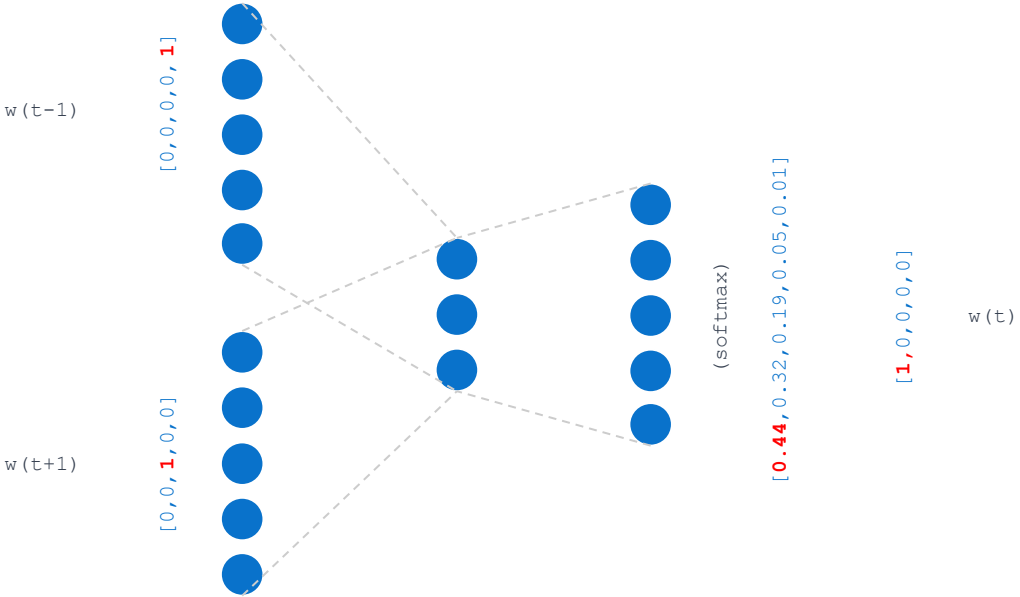
Su arquitectura está basada en dos redes neuronales concatenadas: **CBOW (Continuous Bag of Words)** y **Skip-Gram**.



CBOW predice una palabra objetivo basándose en las palabras de contexto

Skip Gram predice las palabras de contexto a partir de una palabra objetivo

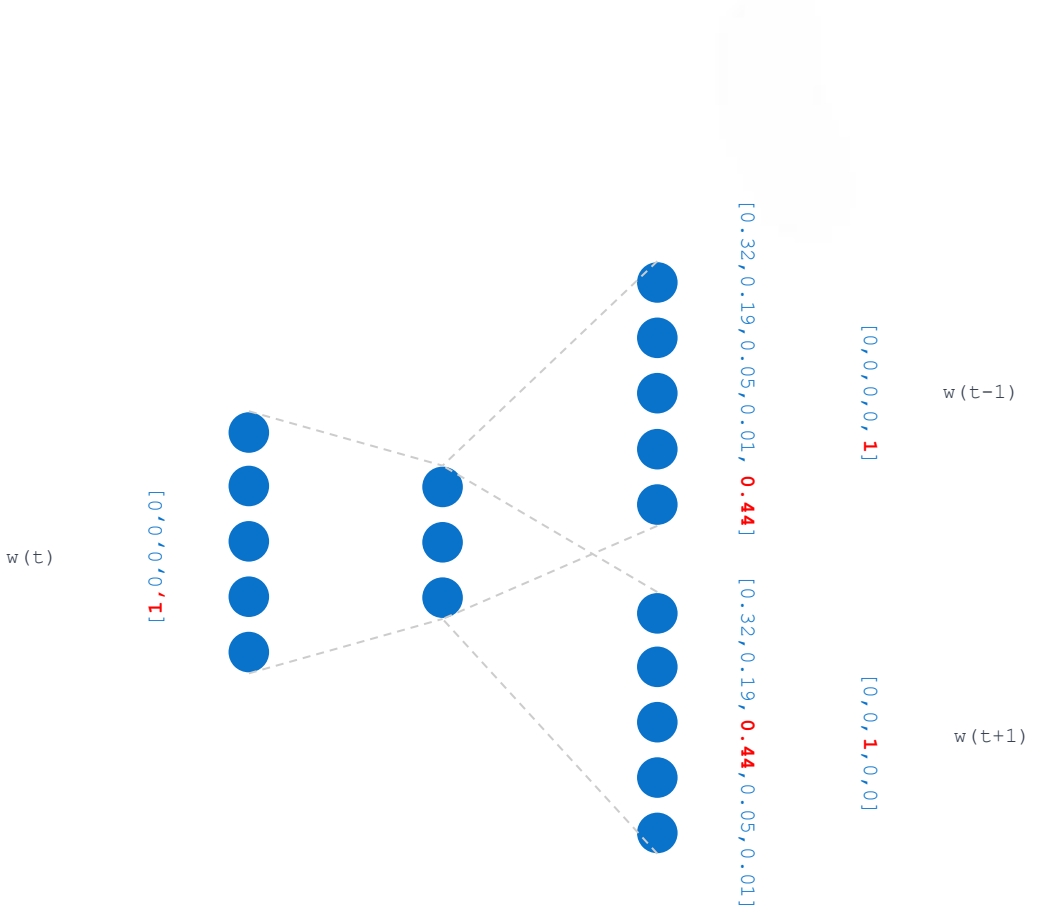
Word2Vec



CBOW

Skip Gram

Word2Vec



CBOW

Skip Gram

Ejemplo

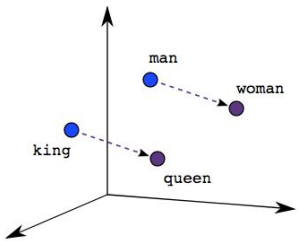
"I love NLP and I will learn NLP in a month"

Text chunk	target	context
I love NLP	I	love, NLP
I love NLP and	love	I, NLP, and
I love NLP and I will learn	NLP	I, love, and, I, will

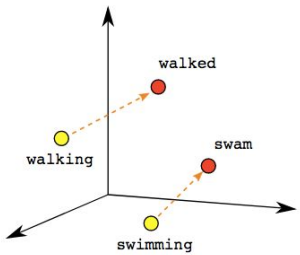
```
model = Word2Vec(
    sentences=word_sentences,  # corpus
    vector_size=20,            # embedding dimmensions --> tamaño de las hidden layers
    window=3,                  # número de palabras antes/después a tener en cuenta
    min_count=3,               # frecuencia mínima de la palabra para que aparezca en el diccionario
)
```

https://radimrehurek.com/gensim/models/word2vec.html

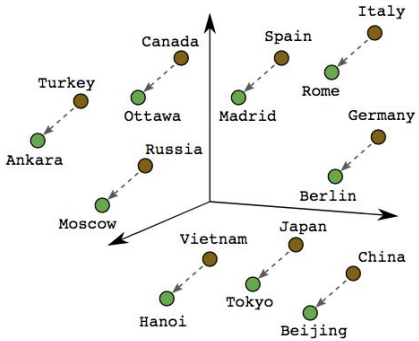
Ilustración de relaciones de palabras



Male-Female



Verb Tense



Country-Capital

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (Skip-gram model trained on 783M words with 300 dimensionality).*

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

sentence-transformers

- Desarrollada por UKPLab, `sentence-transformers` extiende la funcionalidad de la popular librería `transformers` de Hugging Face
- `sentence-transformers` está optimizada para trabajar a nivel de oraciones o textos más largos, generando embeddings que capturan el significado completo del texto introducido.
- Soporte para múltiples modelos preentrenados específicamente diseñados para la generación de embeddings de oraciones.
- Funcionalidades para realizar tareas de NLP como agrupación semántica, búsqueda semántica, y clasificación de texto, aprovechando los embeddings generados.

https://www.sbert.net

Further Readings

Este tema se puede completar con el algoritmo **Global Vectors for Word Representation** (**GloVe**, 2014). Es un algoritmo de aprendizaje no supervisado para obtener representaciones vectoriales de palabras. Este modelo es utilizado para capturar tanto la estadística global de la co-ocurrencia de palabras en un corpus como las relaciones locales entre palabras .

El caso de uso es muy similar al visto de **Word2Vec**, pero el enfoque matemático es distinto.

Se propone al estudiante que investigue más sobre este algoritmo.

ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE MADRID