Escuela de verano

Universidad Autónoma del Estado de Morelos

Centro de Investigación en Ciencias

Laboratorio de Semántica Computacional

Presentación

David Torres Moreno

Licenciatura en Ciencias área terminal Computación

Maestría en Ciencias Cognitivas

Estudiante del Doctorado en Ciencias

Contenido

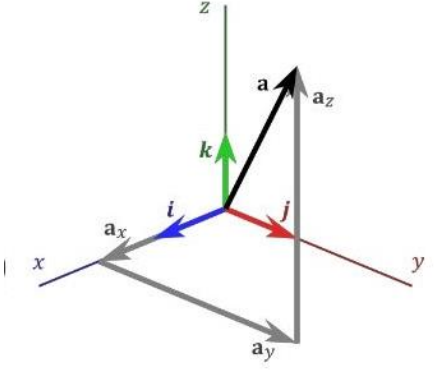
1. Words embeddings
2. Word2Vec

Modelo vectorial – Bag of words

El vector de características representa diferentes aspectos de la palabra: cada palabra se asocia a un punto en un espacio vectorial.

Un espacio vectorial es una estructura matemática creada a partir de un conjunto no vacío, con una operación suma interna al conjunto y una operación producto externa entre dicho conjunto y un cuerpo, cumpliendo una serie de propiedades o requisitos iniciales.

A los elementos de un espacio vectorial se les llamará vectores y a los elementos del cuerpo se les llamará escalares.

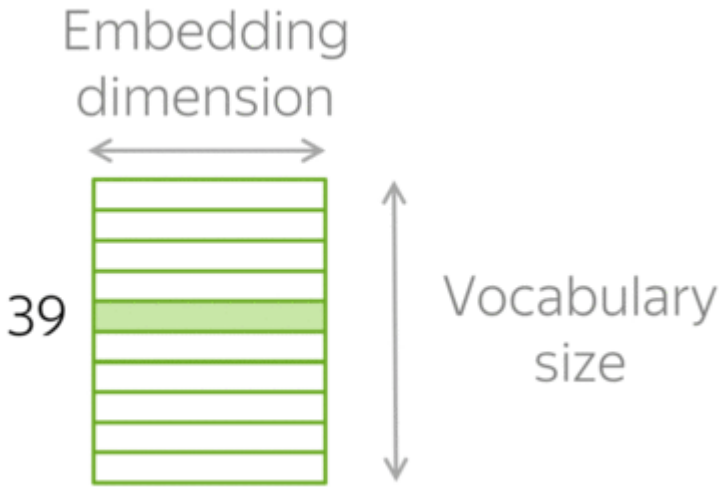


Modelo vectorial – Bag of words

Centro de Investigación en Ciencias

Queremos asignar un vector numérico a cada palabra del vocabulario de un texto.

El vector de características representa diferentes aspectos de la palabra: cada palabra se asocia a un punto en un espacio vectorial.



LABSEMCO Data Science

Modelo vectorial – Bag of words

Centro de Investigación en Ciencias

Codificaremos el número en un vector binario (vector de 0 y 1) de tamaño |V| donde V es el vocabulario de palabras en un texto dado. Este vector tendrá ceros en todas partes excepto en el índice correspondiente al número que le asignamos a la palabra donde pondremos 1.

```
I think therefore I am.
```

```
I   think   therefore   am
1     2         3        4
```

```
              1  2  3  4
I            [1, 0, 0, 0]
think        [0, 1, 0, 0]
therefore    [0, 0, 1, 0]
am           [0, 0, 0, 1]
```

LABSEMCO Data Science

Word embeddings

- Los vectores de características asociados a cada palabra podrían inicializarse utilizando un conocimiento previo de las características semánticas.

1. ¿Es un ser humano?
2. Es masculino?
3. Es mujer?
4. ¿Es un gobernante estatal?
5. ¿Es inanimado?

```
word/context   1.    2.    3.    4.    5.
king         [ 0.9   1     0     1     0 ]
queen        [ 0.9   0     1     1     0 ]
man          [ 1     1     0     0.5   0 ]
woman        [ 1     0     1     0.5   0 ]
boy          [ 1     1     0     0.5   0 ]
girl         [ 1     0     1     0.5   0 ]
book         [ 0     0     0     0     1 ]
paper        [ 0     0     0     0     1 ]
```

$$king - man + woman = \begin{bmatrix} 0.9 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0.5 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = queen$$

Word embeddings

El contexto juega un papel fundamental en la determinación de la similitud semántica. Vectores resultantes serían mucho más representativos de sus palabras.

Se espera que las palabras "similares" se encuentren cercanas (cos(α)).

Word embeddings

Centro de Investigación en Ciencias

Diversos algoritmos proporcionan representaciones vectoriales de las palabras en las que estos vectores conservan la relación lingüística subyacente entre las palabras de acuerdo a la coocurrencia con su contexto.

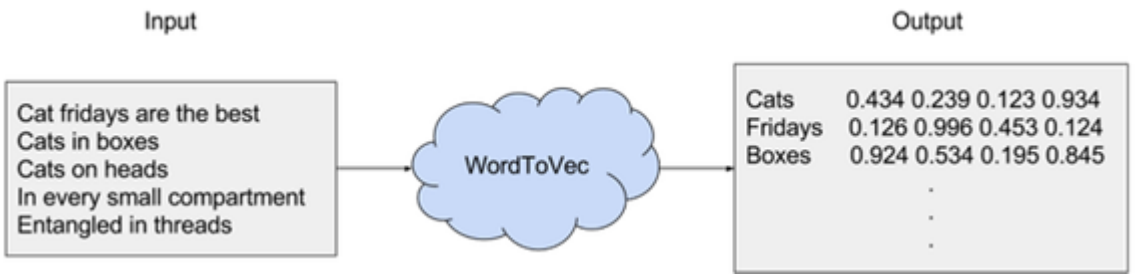
Estos vectores se calculan utilizando diferentes enfoques como las redes neuronales, la matriz de co-ocurrencia de palabras o las representaciones en términos del contexto en el que aparece la palabra. Algunas de los words embeddings pre-entrenadas más utilizadas son: word2vec, GloVe, fastText, BERT, etc.

LABSEMCO Data Science

Word2vec

Centro de Investigación en Ciencias

- El algoritmo word2vec utiliza un modelo de red neuronal para aprender asociaciones de palabras a partir de un gran corpus de texto.
- Puede detectar palabras sinónimas y discriminar palabras de acuerdo en el contexto qu se desenvuelven
- Los resultados del entrenamiento de Word2vec pueden ser sensibles a la parametrización.



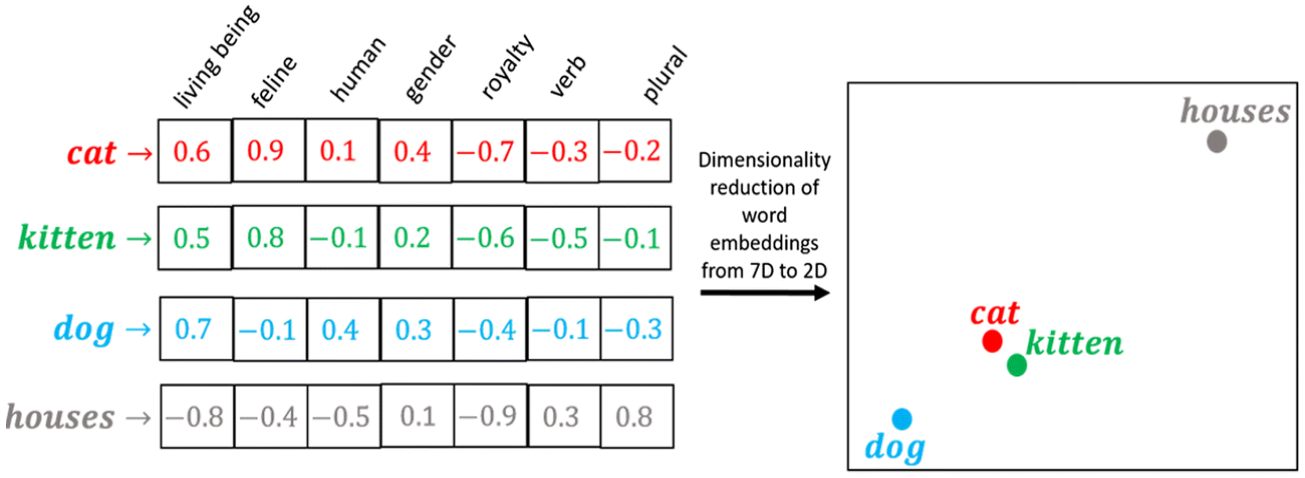
Word2vec

Centro de Investigación en Ciencias

Word2vec: Coocurrencia de primer orden:

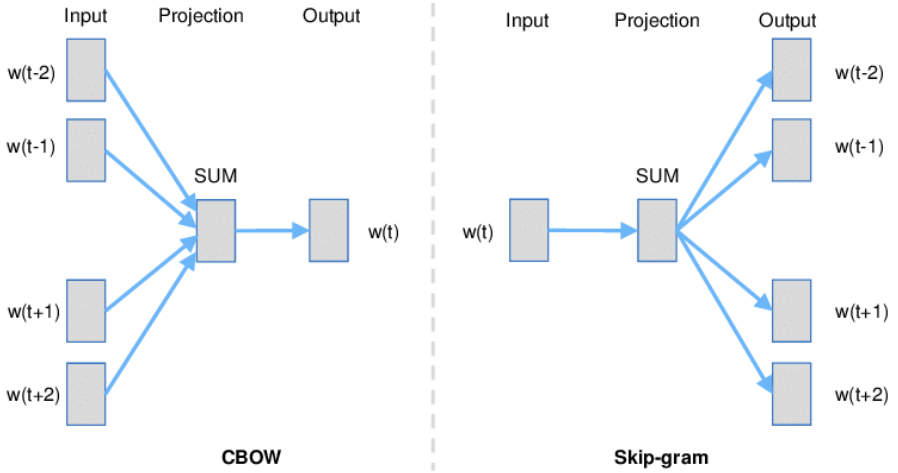
Relaciona palabras que comparten contextos similares.

Word2vec es capaz de capturar múltiples grados diferentes de similitud entre palabras.



Word2vec

- Desarrollado desde el conjunto de datos de noticias de Google, que contiene aproximadamente 3 millones de representaciones de vectores de palabras y frases, $word2vec$ es un modelo de red neuronal utilizado para producir representaciones de palabras basadas en un corpus.
- Existen dos modelos diferentes de $word2vec$ propuestos: la bolsa continua de palabras (cbow) y el modelo de skip-gram.



Antecedentes

La red neuronal de Bengio (2003) era profunda y muy ineficiente.

- 1. asociar a cada palabra del vocabulario un vector de características de palabra distribuido (un vector de valor real en Rm),
- 2. expresar la función de probabilidad conjunta de secuencias de palabras en términos de los vectores de características de estas palabras en la secuencia, y
- 3. aprender simultáneamente los vectores de características de las palabras y los parámetros de esa función de probabilidad.

Antecedentes

Semántica y sintáctica

- The cat is walking in the bedroom
- A dog was running in a room
- The cat is running in a room
- A dog is walking in a bedroom
- The dog was walking in the room.

Contexto

- **¡Dime quienes son tus amigos y te diré quien eres!**

- Las palabras que ocurren en contextos similares deberán ser cercanas.

- The **kid** said he would grow up to be superman
- The **child** said he would grow up to be superman

Contexto

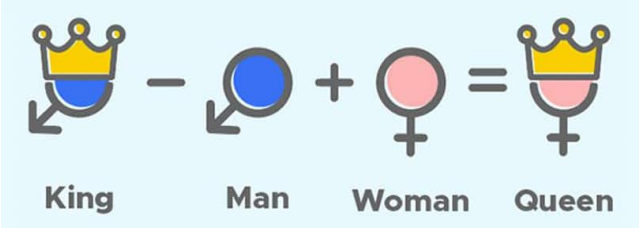
- **¡Dime quienes son tus amigos y te diré quien eres!**

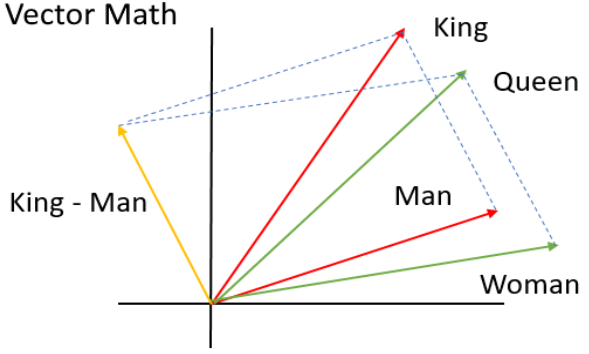
- Las palabras que ocurren en contextos similares deberán ser cercanas.

- The **kid** said he would grow up to be superman
- The **child** said he would grow up to be superman
- The **child** runs for the ball

Tomas Mikolov, Kai Chen, Greg Corrado & Jeffrey Dean (2013)

- Introducir nuevas y mejores técnicas para aprender vectores de palabras de grandes corpus (billones de palabras) y con vocabulario con millones de palabras.
- Ninguna propuesta hasta ese momento tuvo éxito con esto. (Bengio)
- Se espera que las palabras similares sean cercanas entre ellas, pero que además se vean varios grados de similitud, más allá de simples regularidades sintácticas.





Arquitectura

apple	0
drink	0
eat	1
juice	0
milk	0
orange	0
rice	0
water	0

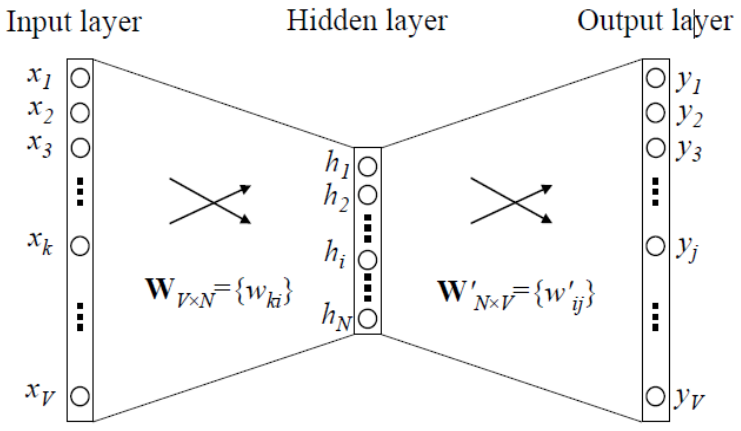


Figure 1: A simple CBOW model with only one word in the context

apple	1
drink	0
eat	0
juice	0
milk	0
orange	0
rice	0
water	0

- El tamaño del vocabulario es V y el tamaño de la capa oculta es N.
- Entrada es un vector de tamaño V; $\{x_1; ; x_V\}$, x_i será 1 si es la palabra y todas las demás son 0.

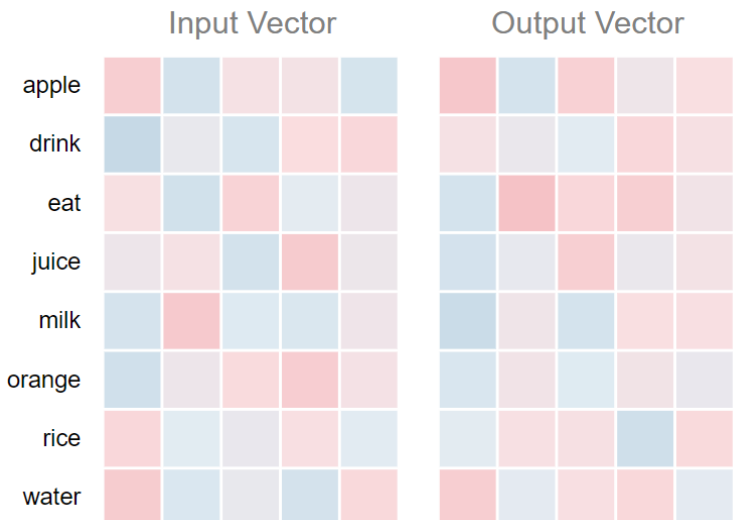
Arquitectura

Los pesos entre la capa de entrada y la de salida pueden representarse mediante una matriz W = V x N

Cada fila de W es la representación vectorial de dimensión N; v_w de la palabra asociada de la capa de entrada.

v_{wi} es la representación vectorial de la palabra de entrada w_i .

Esto implica que la función de activación de las unidades de la capa oculta es simplemente lineal (es decir, que pasa directamente su suma ponderada de entradas a la siguiente capa).



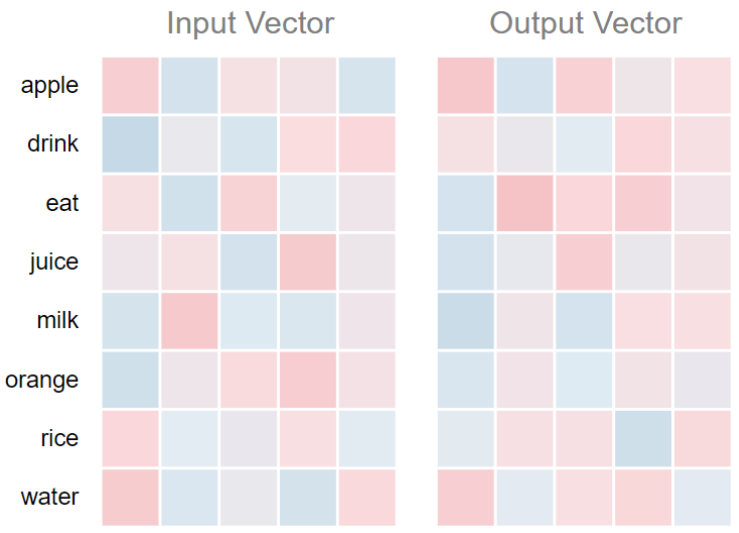
$$\mathbf{h} = \mathbf{W}^T\mathbf{x} = \mathbf{W}^T_{(k,\cdot)} := \mathbf{v}^T_{w_I},$$

Arquitectura

Desde la capa oculta hasta la capa de salida, hay una matriz de pesos diferente W = N x V

Utilizando estos pesos, podemos calcular una puntuación u_j para cada palabra del vocabulario.

Ahora, con softmax, podemos obtener la distribución de las palabras.



$$u_j = \mathbf{v}'_{w_j}{}^T \mathbf{h},$$

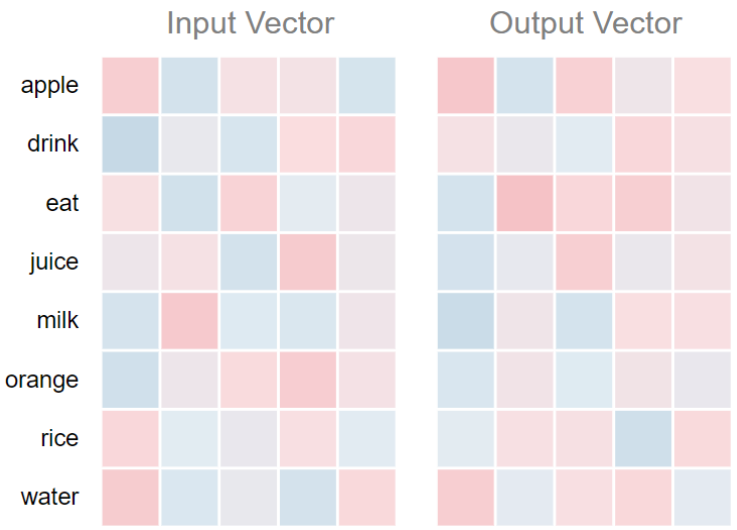
Arquitectura

donde y_j es la salida de la unidad j de la capa de salida.

$$p(w_j|w_I) = y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u_{j'})},$$

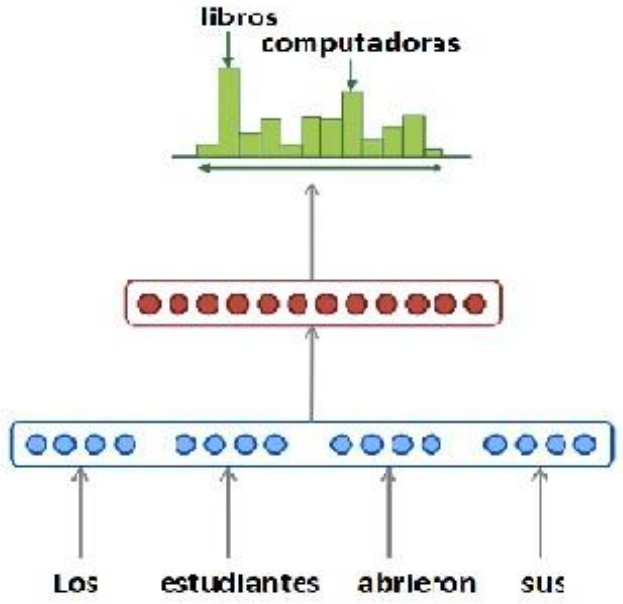
Llamamos a v_w como el "vector de entrada", y v'_w como el "vector de salida" de la palabra w.

$$p(w_j|w_I) = \frac{\exp\left({\mathbf{v}'_{w_j}}^T \mathbf{v}_{w_I}\right)}{\sum_{j'=1}^{V} \exp\left({\mathbf{v}'_{w_{j'}}}^T \mathbf{v}_{w_I}\right)}$$



$y = \{ y_1 , y_2 , \ldots , y_v \}$

Arquitectura



$$\sigma : \mathbb{R}^K \to [0,1]^K$$

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{para } j = 1, \ldots, K.$$

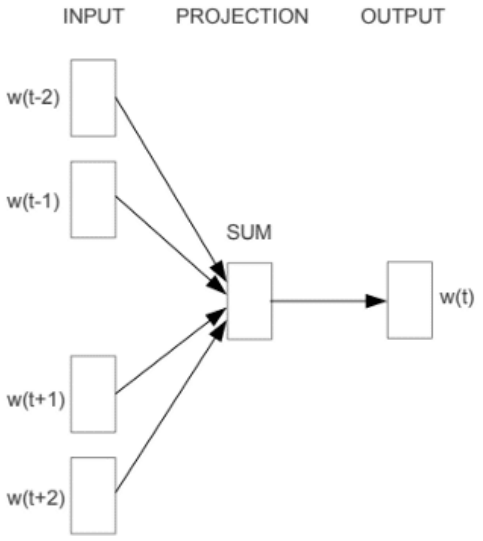
- Yj es el valor obtenido de uj al aplicarle softmax

Contexto

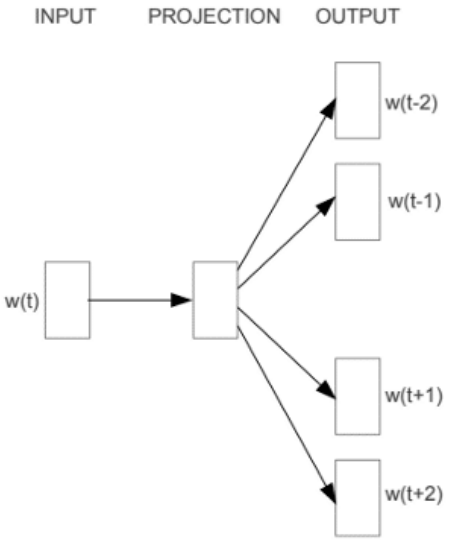
Source Texte

The	quick	brown	fox	jumps	over	the	lazy	dog.
The	quick	brown	fox	jumps	over	the	lazy	dog.
The	quick	brown	fox	jumps	over	the	lazy	dog.

Ventana=5

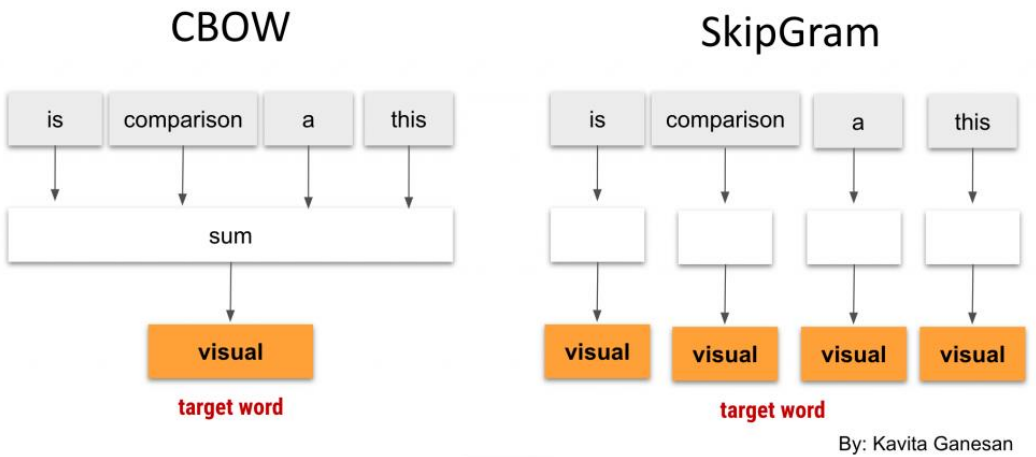


CBOW



Skip-gram

El modelo CBOW aprende a predecir una palabra objetivo con la suma de los vectores de contexto.
El modelo SkipGram, por su parte, aprende a predecir una palabra basándose en una palabra vecina. En pocas palabras, dada una palabra, aprende a predecir otra palabra en su contexto.



Backpropagation

El objetivo de entrenamiento es maximizar

$$p(w_j|w_I) = \frac{\exp\left({\mathbf{v}'_{w_j}}^T \mathbf{v}_{w_I}\right)}{\sum_{j'=1}^{V} \exp\left({\mathbf{v}'_{w_{j'}}}^T \mathbf{v}_{w_I}\right)}$$

Es así que:

$$\begin{aligned} \max p(w_O|w_I) &= \max y_{j^*} \\ &= \max \log y_{j^*} \\ &= u_{j^*} - \log \sum_{j'=1}^{V} \exp(u_{j'}) := -E, \end{aligned}$$

J* es el índice de la palabra de salida real en la capa de salida

Sea la definición de la función de perdida

$$E = -\log p(w_O|w_I)$$

Calcular la derivada es simplemente el error de predicción e_j de la capa de salida.

$$\frac{\partial E}{\partial u_j} = y_j - t_j := e_j$$

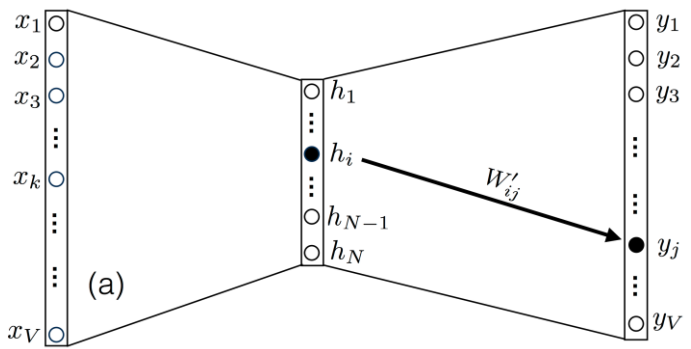
Donde tj= {0, 0, 1_j, 0, 0} cuando j=j*

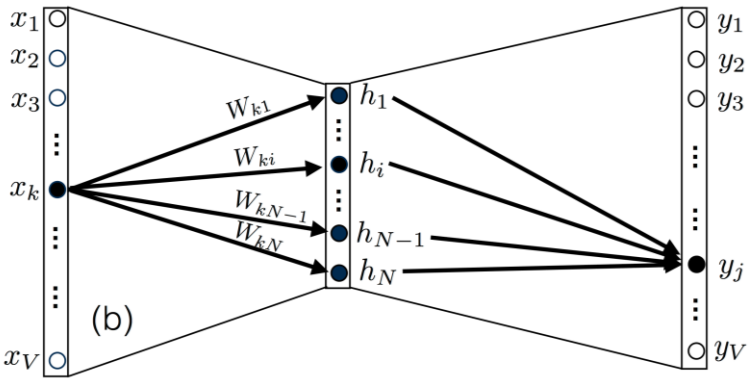
J* es el índice de la salida de la palabra w

Actualización de la ecuación para los pesos de salida

Ahora se calcula la derivada sobre w'_{ij} para obtener el gradiente de la salida de los pesos ocultos

$$\frac{\partial E}{\partial w'_{ij}} = \frac{\partial E}{\partial u_j} \cdot \frac{\partial u_j}{\partial w'_{ij}} = e_j \cdot h_i$$





Actualización de la ecuación para los pesos de salida

Ahora se calcula la derivada sobre w'_{ij} para obtener el gradiente de la salida de los pesos ocultos

$$\frac{\partial E}{\partial w'_{ij}} = \frac{\partial E}{\partial u_j} \cdot \frac{\partial u_j}{\partial w'_{ij}} = e_j \cdot h_i$$

Usando el gradiente descendente

$$w'^{(\text{new})}_{ij} = w'^{(\text{old})}_{ij} - \eta \cdot e_j \cdot h_i.$$

$$\mathbf{v}'^{(\text{new})}_{w_j} = \mathbf{v}'^{(\text{old})}_{w_j} - \eta \cdot e_j \cdot \mathbf{h} \qquad \text{for } j = 1, 2, \cdots, V.$$

Etha > 0, es la tasa de aprendizaje, h_i es la unidad de la capa oculta y v'_{mj} es el vector de salida de w_j

Esta ecuación de actualización implica revisar todas las palabras del vocabulario, verificar su probabilidad de salida y_j y comparar y_j con su salida esperada t_j

Actualización de la ecuación para los pesos de entrada

Una vez actualizado los pesos de la capa de salida, se actualiza la matriz W.

Se calcular la derivada de E sobre la salida de la capa oculta:

$$\frac{\partial E}{\partial h_i} = \sum_{j=1}^{V} \frac{\partial E}{\partial u_j} \cdot \frac{\partial u_j}{\partial h_i} = \sum_{j=1}^{V} e_j \cdot w'_{ij} := \mathrm{EH}_i$$

Para poder actualizar cada uno de los pesos en la capa de entrada.

En caso de que existieran más capas se realiza lo mismo.

- Para tener resultados efectivos, si el tamaño de datos incrementa también lo debe de hacer la dimensión de los vectores de representación de las palabras. Pero esto aumenta la complejidad computacional.
- Tiempo de entrenamiento en un CPU: CBOW un día y Skip-gram tres días

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

Word embeddings

Centro de Investigación en Ciencias

Uno de los principales desafíos que enfrentan al despliegue los embeddings de palabras para medir la similitud es la deficiencia en la confluencia de significados.

Denota que los embeddings de palabras no atribuyen a los diferentes significados de una palabra, contaminando el espacio semántico con ruido al acercar las palabras irrelevantes entre sí.

Por ejemplo, las palabras ‘finance' y 'River' pueden aparecer en el mismo espacio semántico ya que la palabra ‘bank' tiene dos significados diferentes.

LABSEMCO Data Science

Word embeddings

Centro de Investigación en Ciencias

Es fundamental entender que los embeddings de palabras explotan la hipótesis distributiva para la construcción de vectores y dependen de grandes corpus, por lo tanto, se clasifican según los métodos de similitud semántica basados en Corpus.

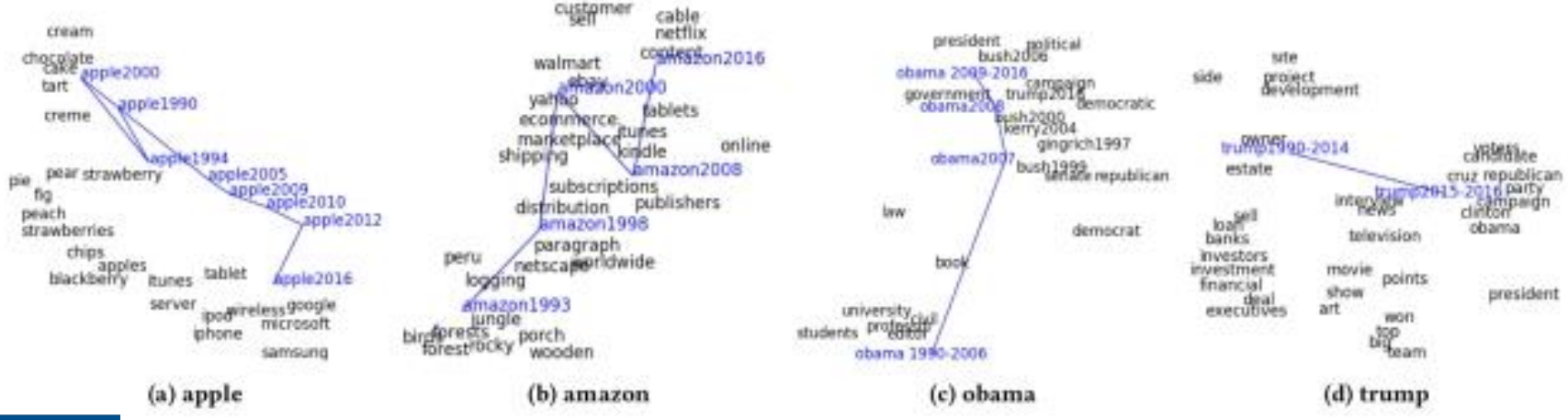


Figure 1: Trajectories of brand names and people through time: apple, amazon, obama, and trump.

LABSEMCO Data Science

Metodos de reducción de dimensionalidad

Centro de Investigación en Ciencias

La técnica lineal principal para la reducción de dimensionalidad, análisis de componentes principals PCA, realiza un mapeo lineal de los datos a un espacio inferior-dimensional de tal manera que la varianza de los datos en la representación de pocas dimensiones se maximiza.

Los componentes se ordenan por la cantidad de varianza original que describen, por lo que la técnica es útil para reducir la dimensionalidad de un conjunto de datos.

LABSEMCO Data Science

Metodos de reducción de dimensionalidad

Centro de Investigación en Ciencias

Reduce el espacio de tiempo y almacenamiento requerido.

La eliminación de multicolinealidad mejora el rendimiento del modelo de aprendizaje automático.

Se hace más fácil de visualizar los datos cuando se reduce a dimensiones muy bajas tales como 2D o 3D.

LABSEMCO Data Science

Word embeddings

Centro de Investigación en Ciencias

Gracias

LABSEMCO Data Science