NLP

Modelos de lenguaje con redes LSTM

Docentes:
Dr. Rodrigo Cardenas Szigety
Dr. Nicolás Vattuone

emails: rodrigo.cardenas.sz@gmail.com
nicolas.vattuone@gmail.com

Redes Neuronales Recurrentes (RNNs)

Es un tipo de neurona con un estado interno (o memoria) de manera que la información del pasado influye en los resultados futuros.

Se utiliza principalmente para resolver problemas de secuencia, en donde el valor anterior está relacionado con el valor futuro.

Permite construir modelos cuyos tamaños de secuencia sean potencialmente arbitrarios.

Implementa modelos de lenguaje de la forma:

$$\prod_{i=1}^{i=m} P(w_i | w_{i-(n-1)}, \ldots, w_{i-1})$$

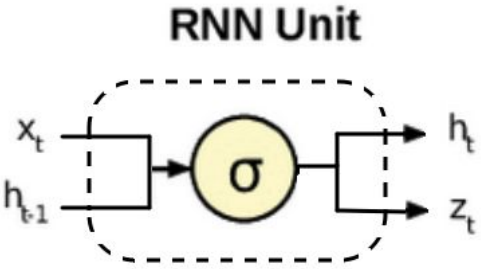
"Hoy el día está hermoso y despejado, se puede ver un hermoso cielo... azul"

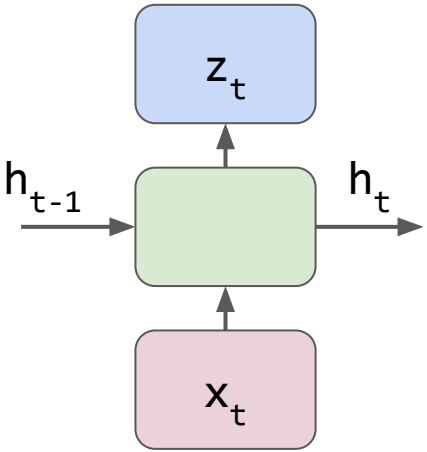
Celda RNN básica (Elman)

LINK API KERAS

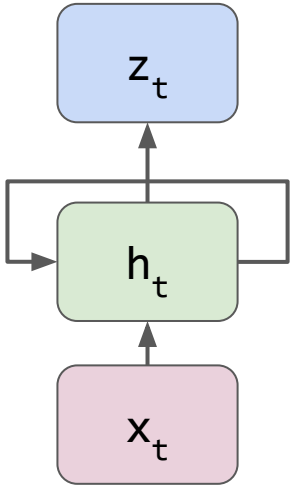
En Keras: SimpleRNN

$$h_t = \sigma(W_{hh} * h_{t-1} + W_{hx} * x + b_h)$$
$$z_t = h_t$$





Unidad básica



Representación compacta

Long short term memory (LSTM) LINK

Se introduce este tipo de celda neuronal con mayor persistencia de memoria para lograr capturar relaciones de palabras a largo plazo.

Se crearon en 1997. Se adoptó como la layer principal para problemas de secuencia en 2014 hasta la aparición de los transformers en 2017.

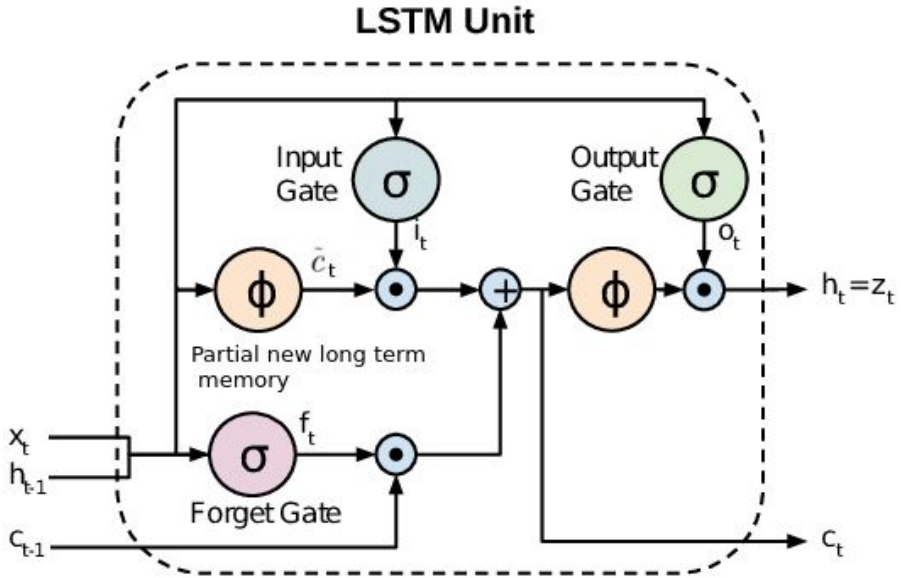
Desplazaron completamente a las capas RNN simples (Elman), ya que el costo adicional de las LSTM es marginal respecto al beneficio que otorgan

Se basan en el principio de ponderar la importancia de una palabra respecto al contexto futuro/pasado (key words).

¿Comprarías este producto?

"Increible! El producto es lo que venden, hace lo que tiene que hacer y me ayudó mucho a resolver los problemas que tenía. Lo volvería a comprar sin dudas"

LSTM approach - Idea de la arquitectura



C: 'Memoria a largo plazo'

Input (i): Dado el último estado (h_{t-1}) evalúa cuánto de la nueva entrada (x) se incluirá en la memoria de largo plazo (C_t).

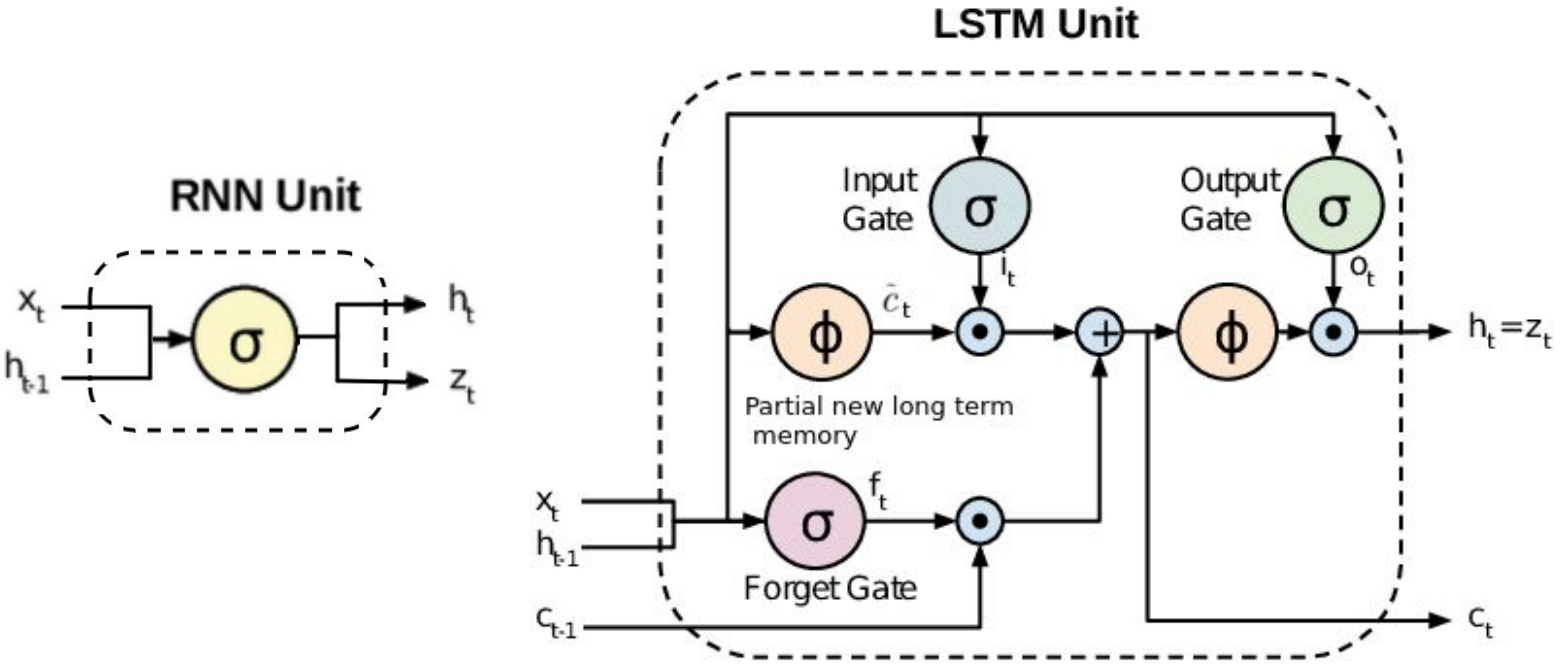
Forget (f): Dada la entrada (X) cuánto del estado de memoria anterior (C_{t-1}) tiene importancia en el nuevo estado.

Output (o): Qué parte del estado de memoria (C_t) pasa al próximo estado de memoria h_t.

•: Producto elemento a elemento
ϕ: Función Tanh
+: Suma vectorial
-> : Conexiones

LSTM vs RNN

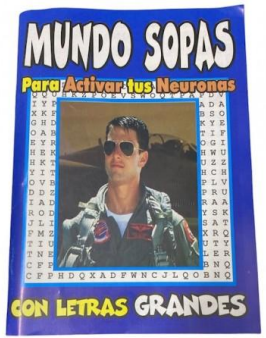
La memoria de largo plazo $\mathbf{c}_t$ permite propagar gradiente eficientemente a mayor “profundidad temporal”.

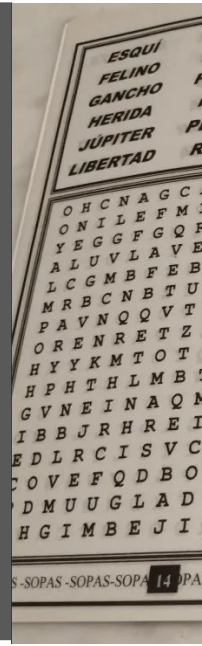


Ejemplo Many-to-one: Sopa de letras

Link al Colab

LINK



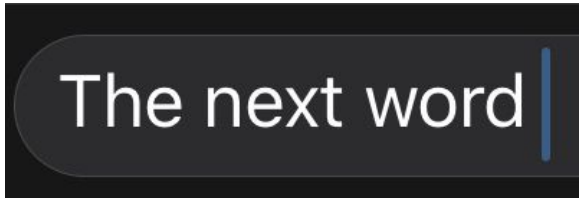


Predicción de texto - Modelos de lenguaje

Objetivo: Entender la estructura estadística del lenguaje aprovechando su estructura secuencial.

$$\prod_{i=1}^{i=m} P(w_i | w_{i-(n-1)}, \ldots, w_{i-1})$$

Tarea: Aprender a predecir la siguiente palabra.



$(X_1,\ldots,X_n) \longrightarrow X_{n+1}$

Pasos a seguir:

Segmentación → Tokenización → Truncado → Aumento de datos → Padding

Segmentación y Tokenización

Documentos:

```
'Yesterday, all my troubles seemed so far away'
```

Segmentación:

```
['yesterday', 'all', 'my', 'troubles', 'seemed', 'so', 'far', 'away']
```

Tokenizacion:

→ [34, 189, 200, 1345, 8, 69, 12,753]

Ventana de contexto

La **ventana de contexto** en un modelo de lenguaje es la cantidad máxima de tokens (palabras o subpalabras) que el modelo puede considerar a la vez como entrada. Define el alcance del "pasado" que el modelo puede usar para predecir la siguiente palabra o generar texto.

Si la secuencia es más **CORTA** que la ventana de contexto → Padding (relleno)

Si la secuencia es más **LARGA** que la ventana de contexto → Truncado

Truncado de secuencias:

Si tenemos una secuencia de tamaño M y una ventana de contexto de tamaño N tal que M > N, entonces podemos generar 1 + M-N secuencias de tamaño N:

Ejemplo (M= 6, N=3)

[“Todas”, “las”,”hojas”, “son”,”del”,”viento”] se convierte en:

["Todas", "las", "hojas"]
["las", "hojas", "son"]
["hojas", "son", "del"]
["son", "del", "viento"]

Data Augmentation y padding

Queremos que el modelo sea capaz de predecir cada elemento de la secuencia, no solo el último:

[“Todas”, “las”,”hojas”, “son”,”del”,”viento”] → **[1, 2, 3, 4, 5, 6]**

Ejemplo (M= 6, N=3)

["Todas", "las", "hojas"]
["las", "hojas", "son"]
["hojas", "son", "del"]
["son", "del", "viento"]

→

["Todas"] -> [0 , 0 ,1]
[“Todas”,”las”] -> [0, 1, 2]
[“Todas”, “las”,”hojas”] -> [1, 2, 3]

La biblioteca de Babel

“No hay en la vasta Biblioteca, dos libros idénticos. De esas premisas incontrovertibles dedujo que la Biblioteca es total y que sus anaqueles registran todas las posibles combinaciones de los veintitantos símbolos ortográficos (número, aunque vastísimo, no infinito) o sea todo lo que es dable expresar: en todos los idiomas. Todo[…]”

Probabilidad de una secuencia

$$P_{LM}\left(x^{T+1}, x^{T}, ..., x^{1}\right) = P_{LM}\left(x^{T+1}|x^{T}, ..., x^{1}\right) P_{LM}\left(x^{T}, x^{T-1}, ..., x^{1}\right)$$

$$= P_{LM}\left(x^{T+1}|x^{T}, ..., x^{1}\right) P_{LM}(x^{T}|x^{T-1}, ..., x^{1}) P_{LM}\left(x^{T-1}, ..., x^{1}\right)$$

$$= \prod_{t=1}^{T} P_{LM}\left(x^{t+1}|x^{t}, ..., x^{1}\right)$$

Ejemplo:

P(“La hojarasca crepitar”) = P(“crepitar” | “La hojarasca”) P(“hojarasca” | “la”) P(“la”)

Perplejidad: “Cantidad media de palabras posibles, en promedio.”

$$\text{perplexity} = \prod_{t=1}^{T} \left(\frac{1}{P_{\text{LM}}(\boldsymbol{x}^{(t+1)} | \boldsymbol{x}^{(t)}, \ldots, \boldsymbol{x}^{(1)})} \right)^{1/T}$$

Normalized by number of words

Inverse probability of corpus, according to Language Model

The

[ ]

Guarda! Es el promedio geométrico en verdad: **16,49**…

Midiendo desempeño en modelos de lenguaje: perplexity

$$\text{perplexity} = \prod_{t=1}^{T} \left(\frac{1}{P_{\text{LM}}(\boldsymbol{x}^{(t+1)} | \boldsymbol{x}^{(t)}, \ldots, \boldsymbol{x}^{(1)})} \right)^{1/T}$$

Normalized by number of words

Inverse probability of corpus, according to Language Model

Por cuestiones de estabilidad numérica conviene operar sobre los logaritmos de las probabilidades.

$$\text{PPL}(X) = \exp\left\{ -\frac{1}{t} \sum_{i}^{t} \log p_\theta(x_i | x_{<i}) \right\}$$

Cross-entropy:

$$L = -\frac{1}{m} \sum_{i=1}^{m} y_i \cdot \log(\hat{y}_i)$$

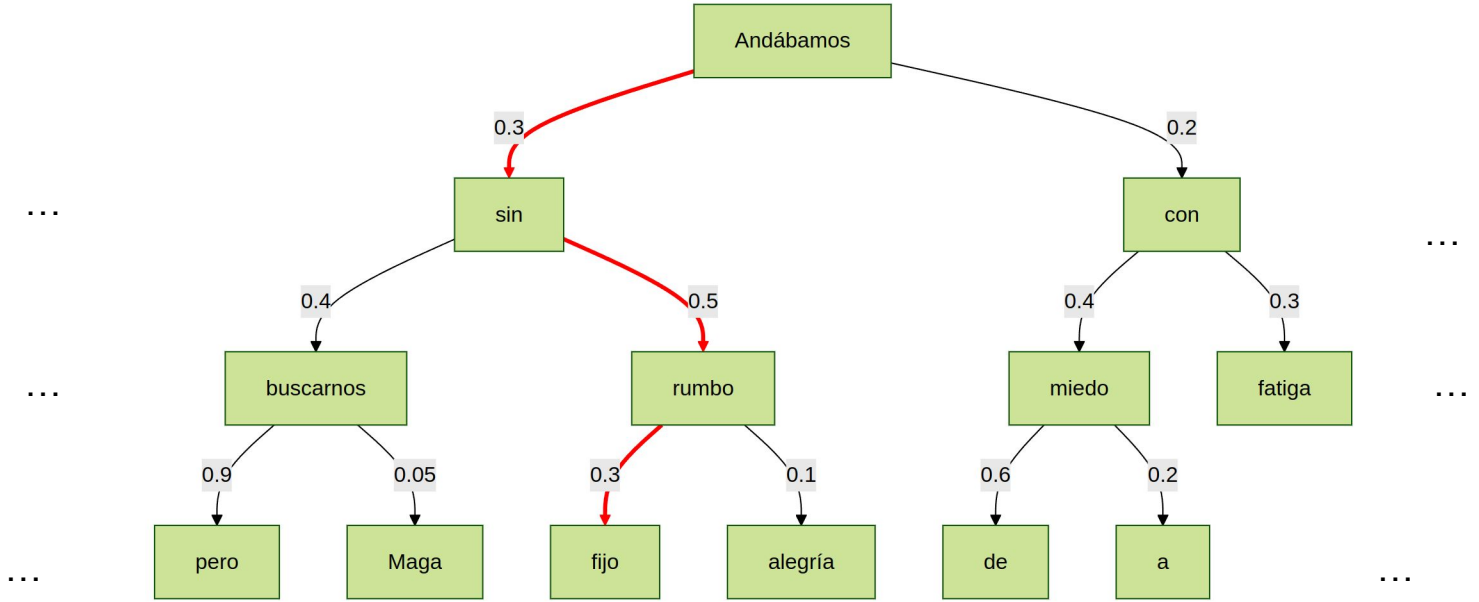
Modelo equiprobable: perplejidad V

Text prediction

Link al Colab

LINK

Generación de texto en modelos de lenguaje: Greedy search

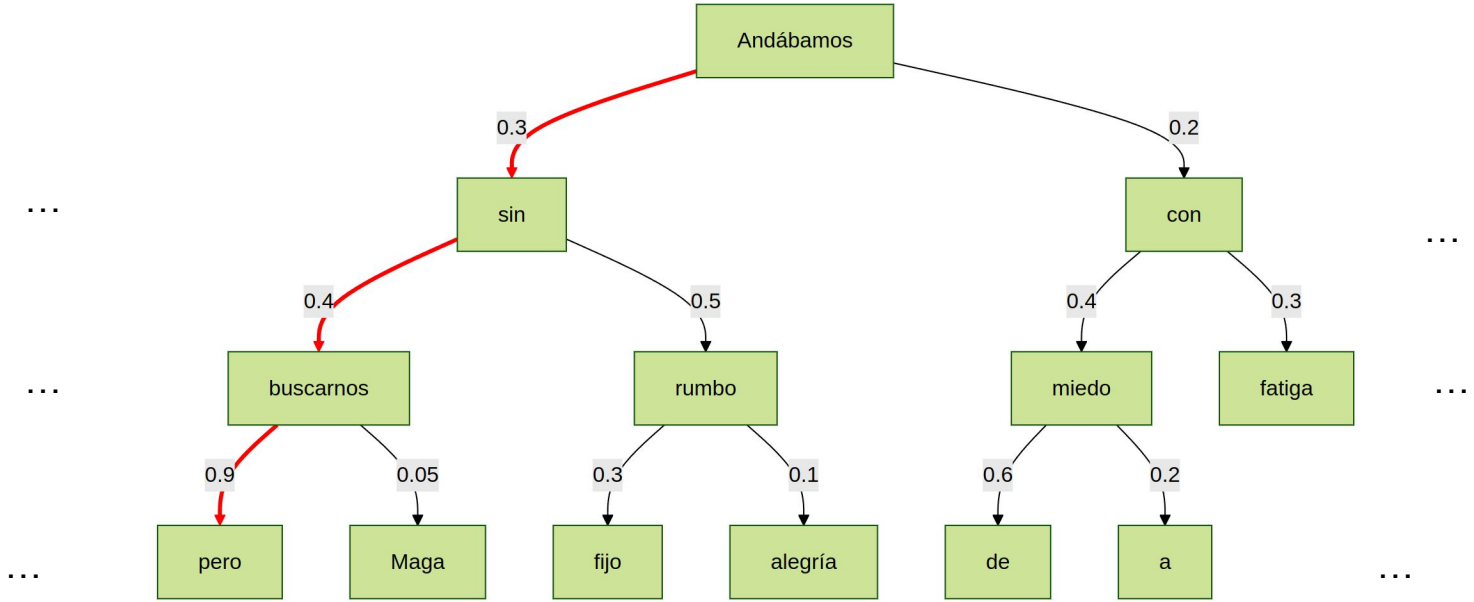


$$\prod_{t=1}^{T} P_{LM}\left(x^{t+1}|x^{t},...,x^{1}\right)$$

P(“Andábamos sin rumbo fijo”) =

0.3 x 0.045 x 0.3 = **0.045**

Generación de texto en modelos de lenguaje: Beam search



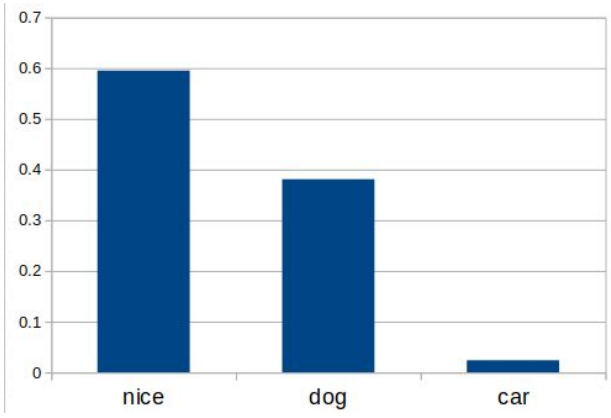
$$\prod_{t=1}^{T} P_{LM}\left(x^{t+1}|x^{t},...,x^{1}\right)$$

P(“Andábamos sin buscarnos, pero”) =

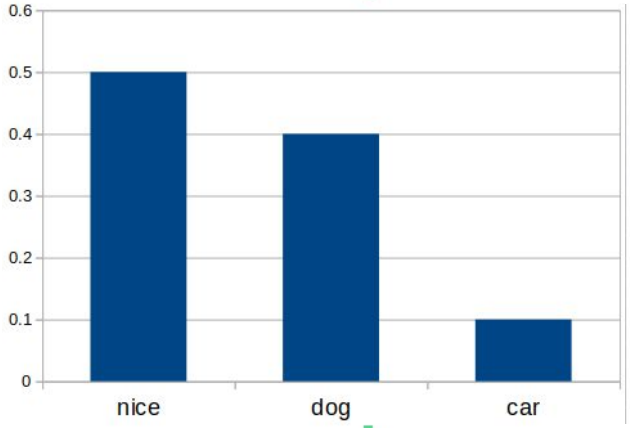
0.3 x 0.4 x 0.9 = **0.108**

Generación de texto en modelos de lenguaje: Muestreo aleatorio con temperatura

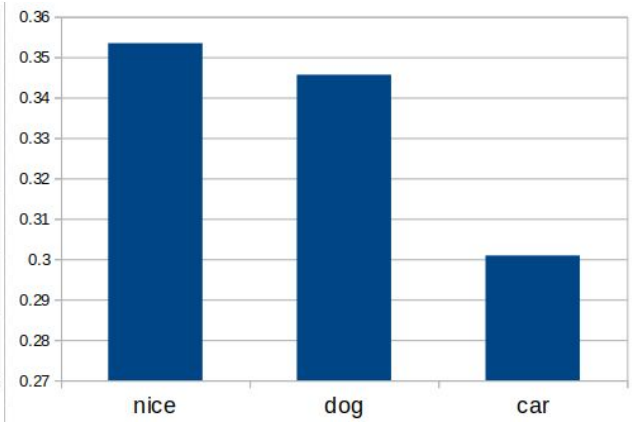
$$P_i = \frac{e^{\frac{y_i}{T}}}{\sum_{k=1}^{n} e^{\frac{y_k}{T}}}$$



Temperatura = 0.5



Temperatura = 1



Temperatura = 10

Desafio

Utilizar otro dataset y poner en práctica la generación de secuencias con las estrategias presentadas.