

Deep Learning en NLP



KEEPCODING
Tech School

Recordamos: Word2Vec

CBOW: Durante el entrenamiento, el modelo tratará de **predecir la palabra actual** dado el contexto en el que se encuentre. La capa de entrada contendrá las palabras-contexto y la de salida será la palabra actual (o palabra a predecir).

Skip-Gram: Durante el entrenamiento, el modelo tratará de predecir **el contexto (palabras-contexto)** a una palabra dada. La capa de entrada contendrá la palabra actual y la de salida serán las palabras contexto.

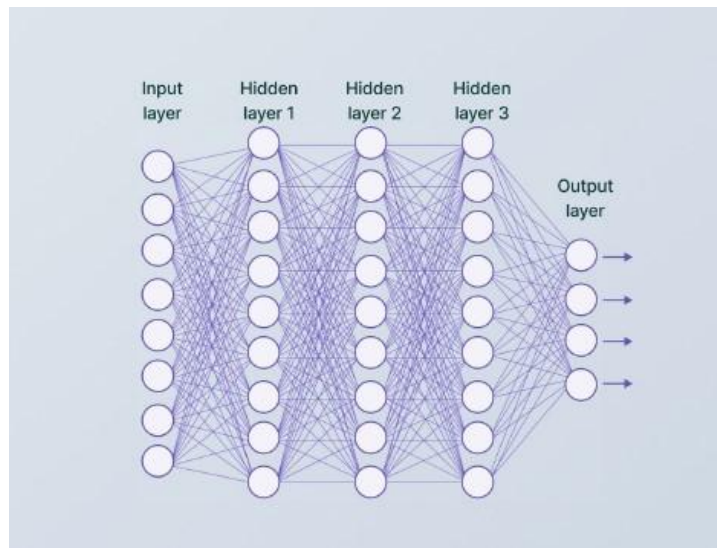


Introducción

- Necesidad de entender cada palabra en base a lo entendido en palabra previas
- El entendimiento tiene persistencia por lo que se tiene que capturar ese valor del texto

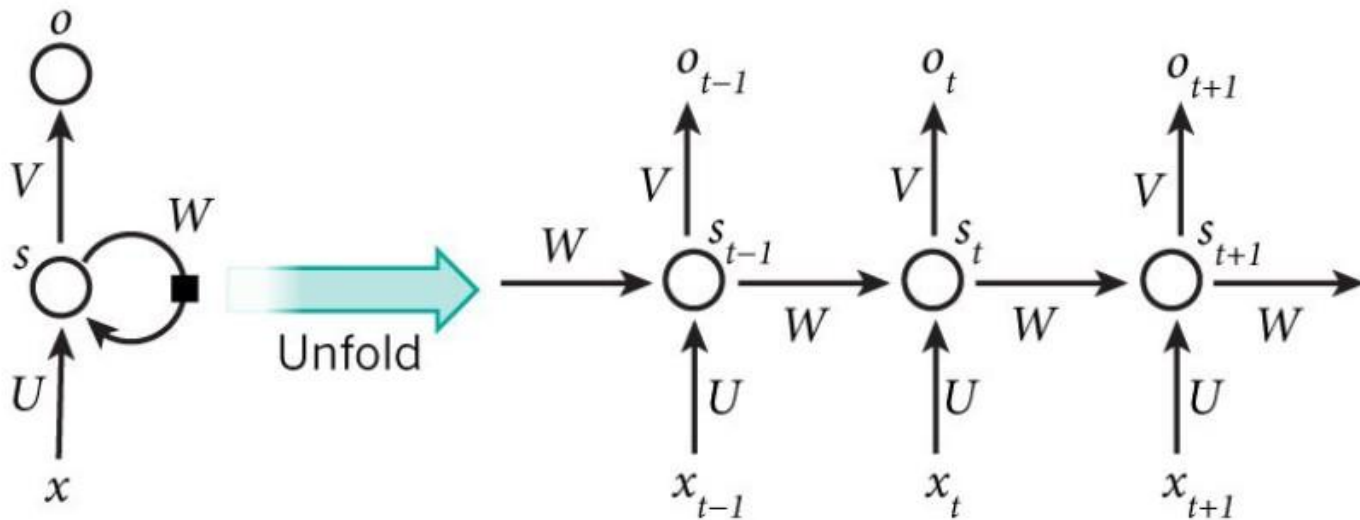
CNN vs RNN

- En redes neuronales tradicionales todas las entradas y las salidas son independientes entre sí. mal planteamiento para NLP



CNN vs RNN

- Las RNNs procesan secuencias y retienen información en la 'memoria'



RNN

- Unfolding es escribir la NN para la secuencia completa. Por ejemplo, si una secuencia tiene 4 palabras la red hará unfolding a 4 capas de la NN.
- S en el tiempo t es la memoria de la red en capa que se encarga de capturar lo que ha pasado en pasos previos.
- Una NN tradicional usa diferentes parámetros para capa capa de la red. En las RNN todos los parámetros son compartidos por todas las capas.

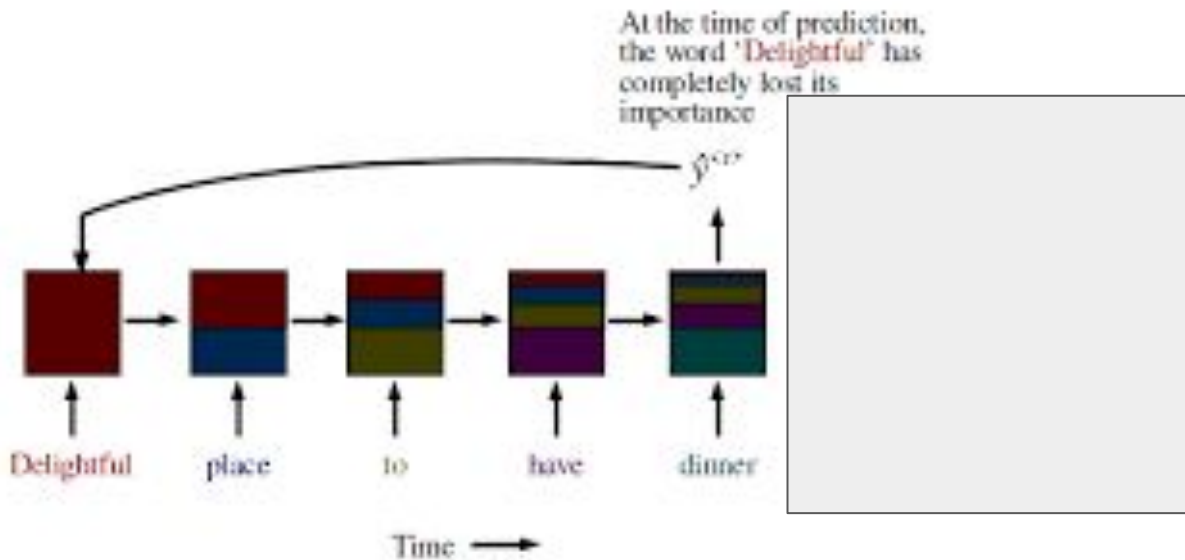


RNN

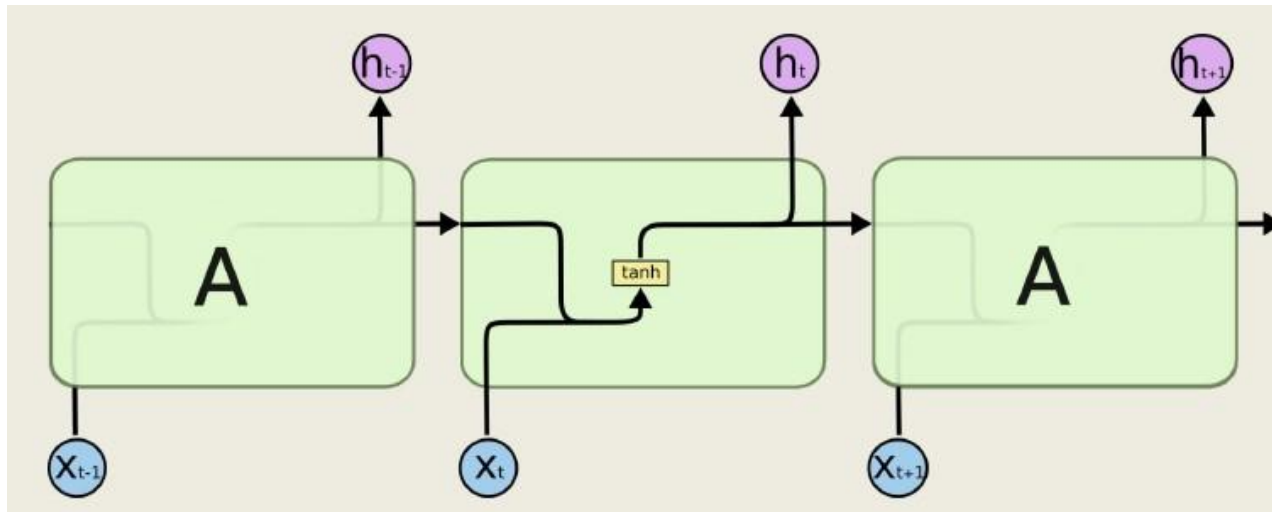


- Compartir los parámetros en todas las capas hace que realicemos la misma tarea con diferentes entradas, reduciendo el número de parámetros a aprender.
- Para usar RNN es necesario preprocesar el texto en una forma entendible para la red. Tokenización junto con One Hot Encoding / Word Embeddings.

RNN

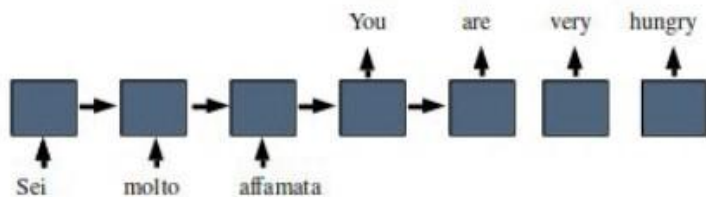


RNN



Tipos de RNN

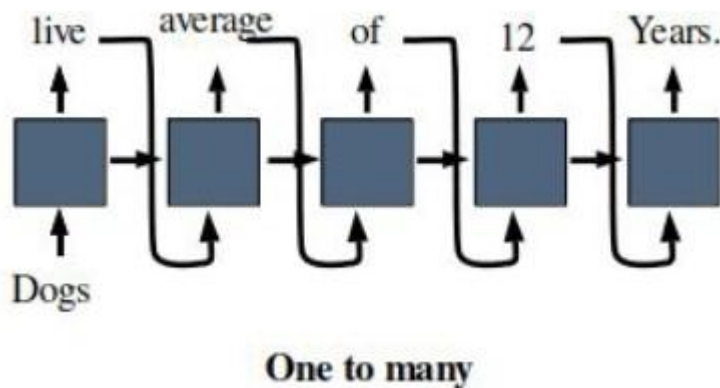
- Propagación de las activaciones por toda la red antes de generar una predicción por capa de salida
- Número diferente de salidas al de entrada o viceversa
- Muy útil para generación de traductores de idioma



Many to many (unequal input-output)

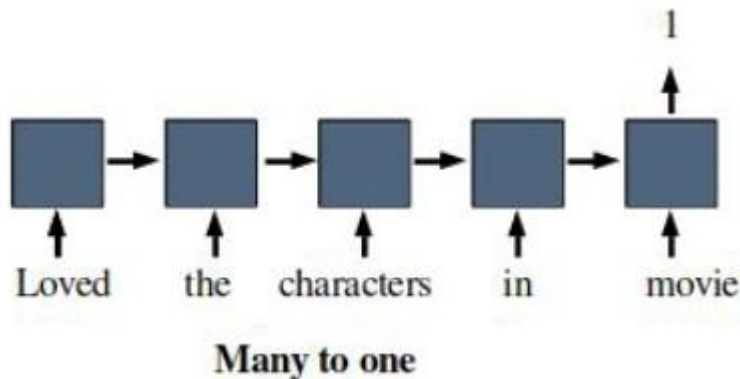
Tipos de RNN

- En cada propagación por una capa genera una predicción y su activación se pasa a la siguiente
- Muy útil para NLG (Natural language Generation), escribir el pie de foto de una imagen



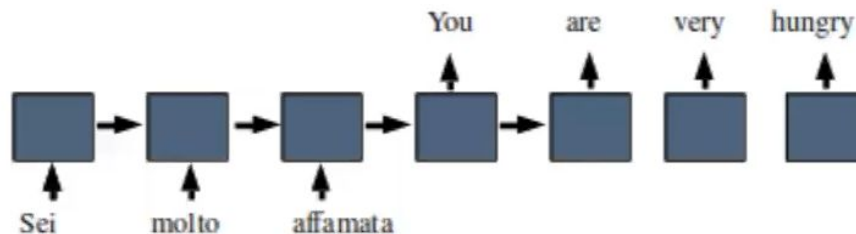
Tipos de RNN

- Propagación de las activaciones por toda la red antes de generar una única predicción
- Muy útil para análisis de sentimiento



Tipos de RNN

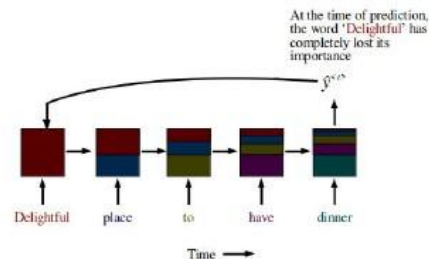
- Propagación de las activaciones por toda la red antes de generar una única predicción
- Muy útil para análisis de sentimiento
-



Many to many (unequal input-output)

Vanishing and Exploding Gradient Problem

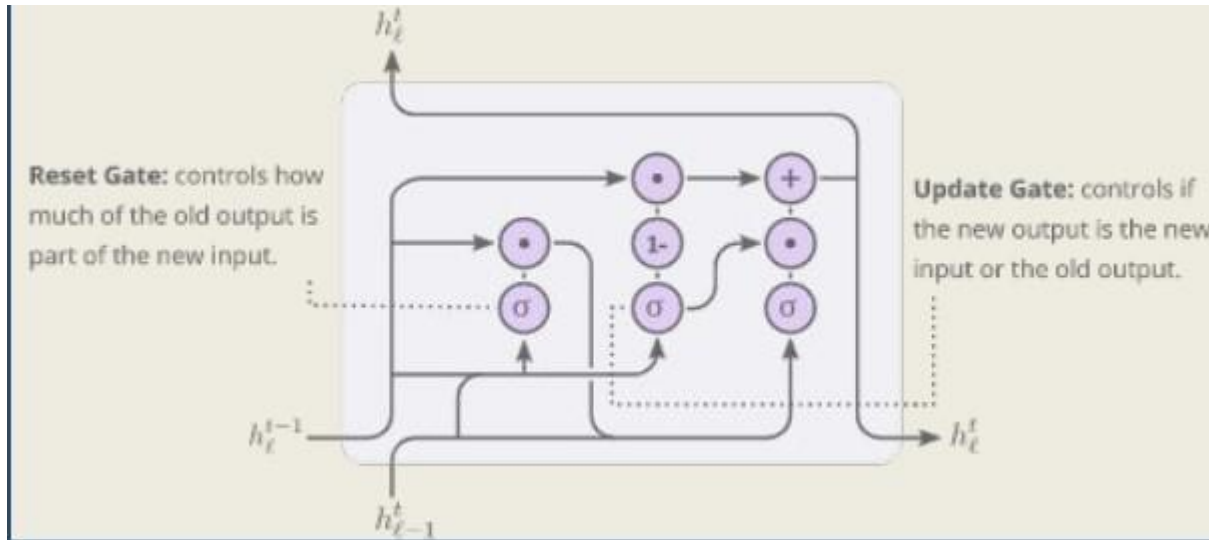
- Las redes RNN tienen solo memoria a corto plazo
- Si usamos una gran cantidad de capas en un RNN podemos sufrir ese problema
- El gradiente indica el ajuste a realizar en los pesos con respecto a la variación en el error.
 - 'Gradientes explosivos': el algoritmo asigna una importancia exageradamente alta a los pesos. En este caso el problema se puede resolver trucando los gradientes
 - 'Gradientes Desaparecidos' : los valores de los gradientes son demasiados pequeños y el modelo deja de aprender o aprende muy despacio.
- Si usamos una gran cantidad de capas en un RNN podemos sufrir ese problema
- Los gradientes se desvanecen (valores de casi 0) cuando se hace Backpropagation hacia las capas iniciales



GRU Networks

- Incorporación de Update Gate y Reset Gate para resolver Vanishing Gradient problem.
- Update Gate: ayuda a decidir cuánta información de capas anteriores debe pasarse a capas futuras.
- Reset Gate: ayuda a decidir cuánta información de capas anteriores olvidar

GRU Networks

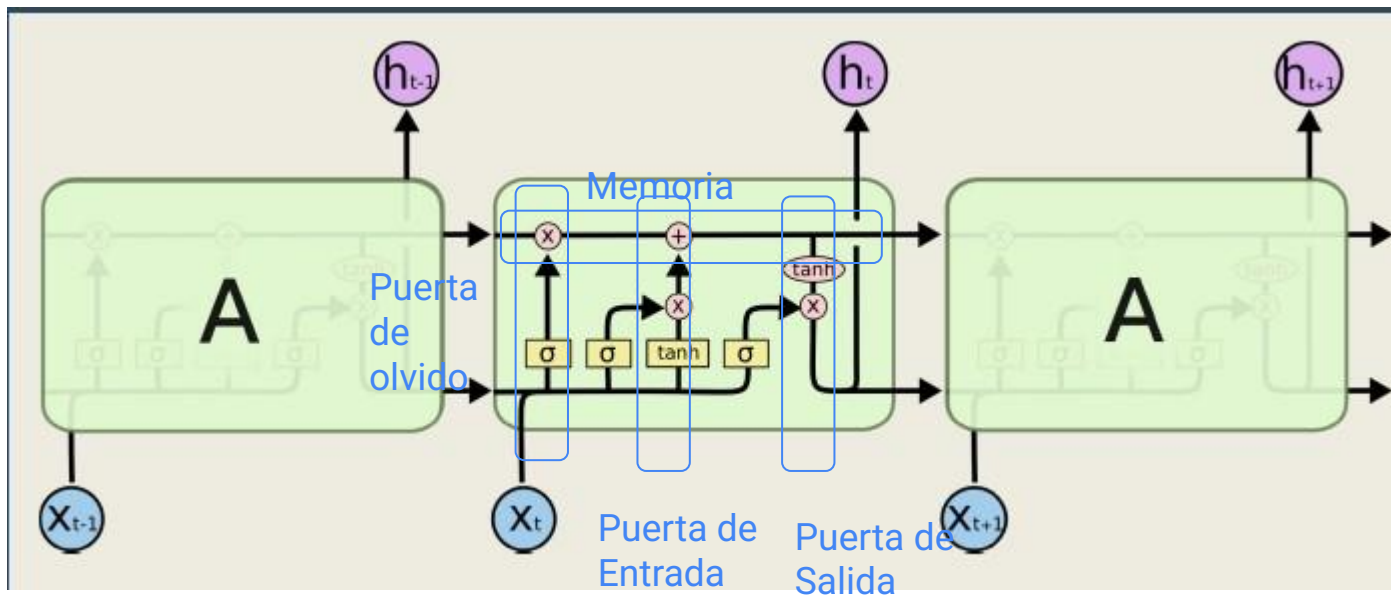




LSTM Networks

- Hace el uso de Gates como en las GRU.
- No propaga la información a posteriores capas sino que la almacena en una unidad de memoria integrada en cada capa.
- Tienen un diseño más complejo pues hace uso de más Gates.

LSTM Networks



GRU vs LSTM

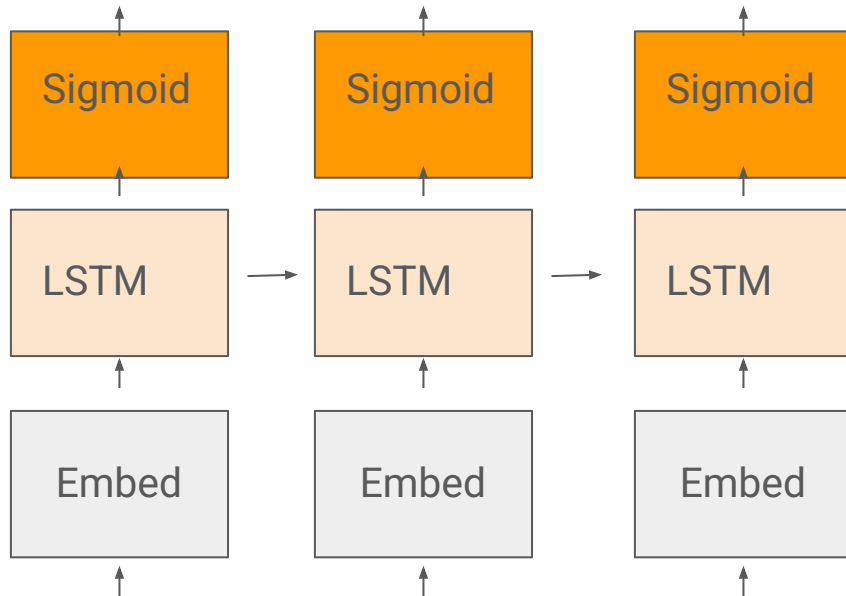
- Las LSTM son capaces de almacenar información a más largo plazo que GRU.
- Las GRU entrenan más rápido que las LSTM y tienen un rendimiento más alto con menor cantidad de datos.
- En tareas dependientes de largas relaciones temporales las LSTM tienen mejor rendimiento.



RNNs, LSTMs y GRUs

- RNNs: problemas relacionados con 'Gradientes Desaparecidos'
- LSTMs: mantienen los valores de los gradientes suficientemente altos y, por lo tanto, el entrenamiento es más rápido y la precisión mejora
- GRUs: rendimiento similar, computacionalmente más eficientes

Lab: Análisis del sentimiento



- Primero pasaremos palabras a una capa de embeddings
- Después que las palabras se hayan pasado a una capa de embedding, el nuevo embedding será pasado a una LSTM. Ayudando a crear una conexión recurrente e incluyendo información acerca de la secuencia de las palabras en las reviews.
- Finalmente, la salida del LSTM irá a una capa sigmoid output. En nuestros lab, los resultados que tendremos serán 1 o 0. Nos va a predecir el sentimiento de la review entre 0 y 1.



KEEPCODING

Tech School

Madrid | Barcelona | Bogotá

Datos de contacto