Deep Learning y Redes Recurrentes

Fabián Villena

Introducción

La utilización de redes neuronales recurrentes para el análisis de datos no estructurados alcanza el estado de arte.

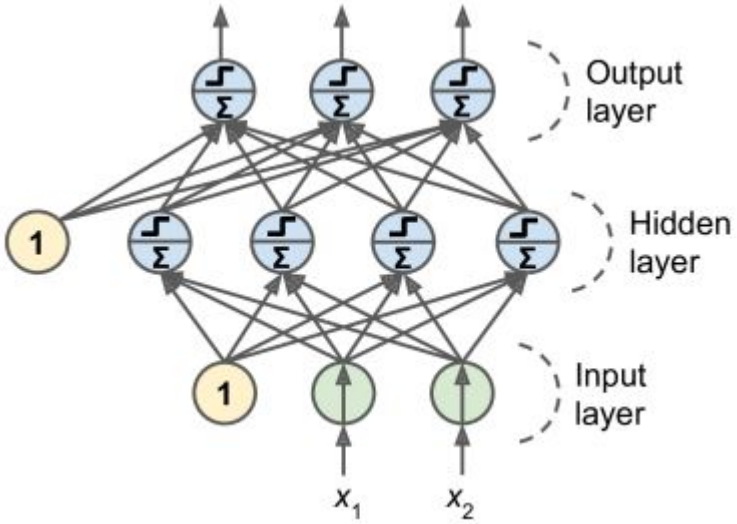
Existe un tipo de red neuronal llamada recurrente la cual toma en cuenta el orden de los elementos de una secuencia y es natural utilizarla en análisis de texto.

Si bien actualmente las redes neuronales recurrentes no obtienen el estado del arte en tareas de NLP, es importante analizarlas.

Redes neuronales artificiales

Las redes neuronales son una herramienta fundamental para el procesamiento de lenguaje natural.

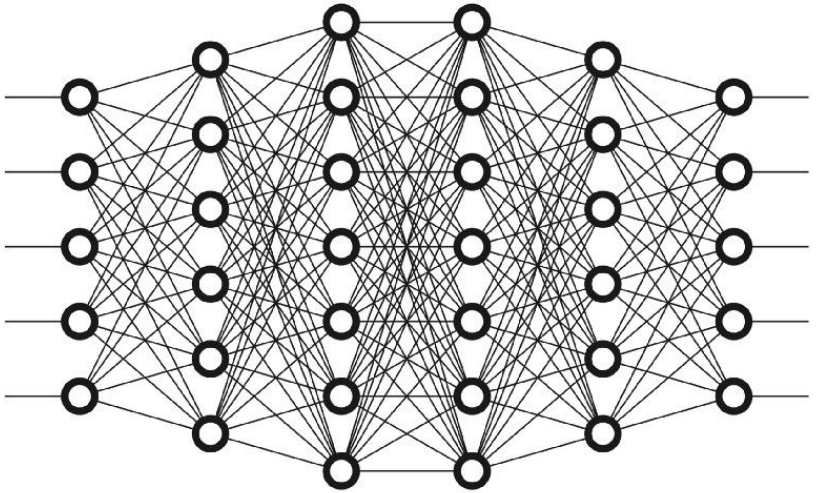
Las redes neuronales artificiales modernas son una red de pequeñas unidades de computación, donde cada una toma un vector de valores de entrada y produce un escalar de salida.



Deep learning

El uso de redes neuronales modernas típicamente se llama Deep Learning dado que muchas de las redes actuales tienen múltiples capas ocultas, lo que las hace profundas.

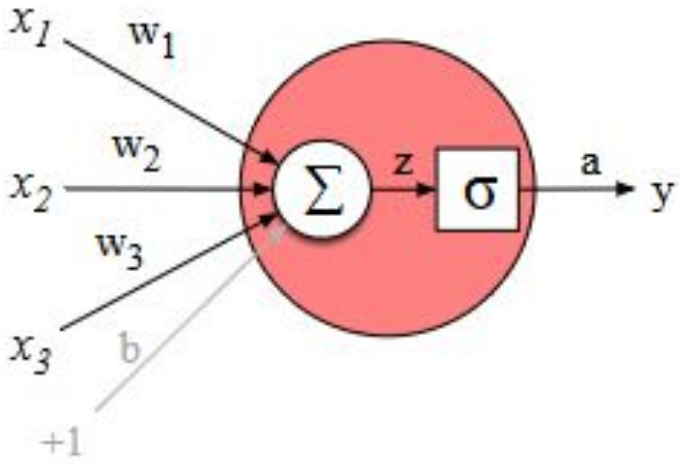
Cuando trabajamos con redes neuronales, normalmente no generamos características a mano, sino que dejamos que la red aprenda a representar características.



Neurona o unidad

La unidad mínima de computación en una red neuronal artificial es la neurona o la unidad.

Una neurona realiza una suma ponderada de sus entradas y tiene un parámetro adicional que se suma al final llamado sesgo.



Función de activación

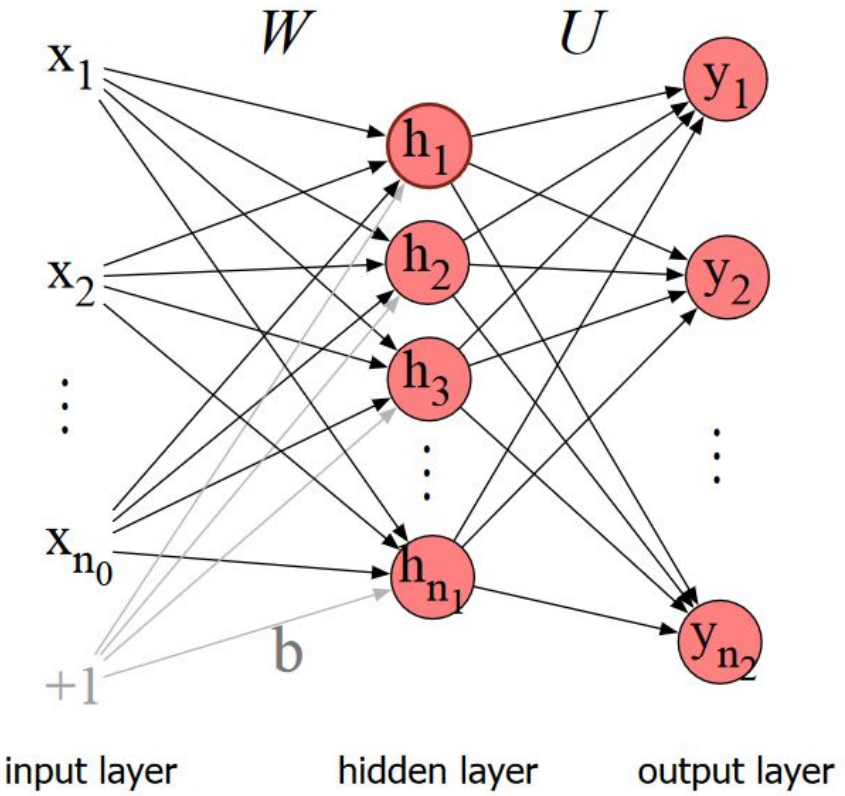
Después de realizar las sumas ponderadas, las neuronas aplican una función no lineal a la salida. Esta función es llamada función de activación y al valor retornado por esta función se le llamará activación.

La función sigmoidea mapea la saluda hacia un rango entre 0 y 1. La función tangente hiperbólica es similar a la sigmoidea, pero mapea hacia valores entre -1 y +1.

La función más simple y la más utilizada es la ReLU, que retorna el mismo valor cuando es mayor a 0 y retorna 0 cuando el valor de entrada es menor a 0.

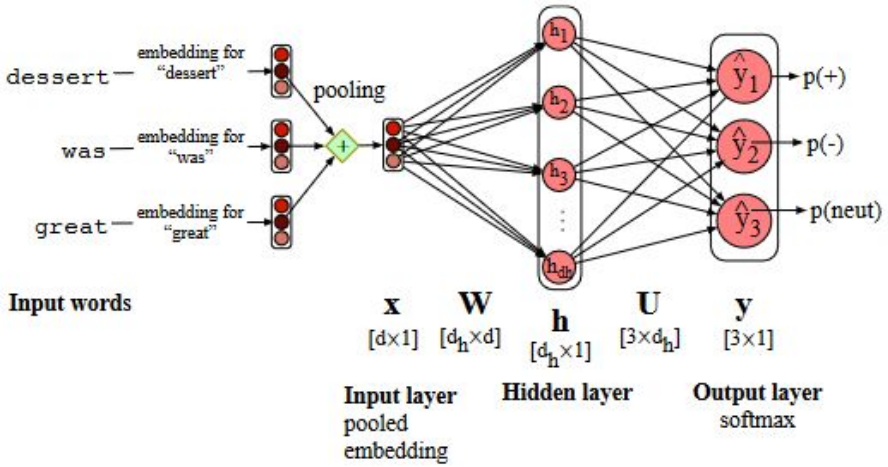
Red neuronal totalmente conectada

Las redes totalmente conectadas son una red multicapa en donde las unidades están conectadas sin ciclos; las salidas de las unidades en cada capa son pasados hacia unidades en la siguiente capa.



Clasificación en NLP con una red totalmente conectada

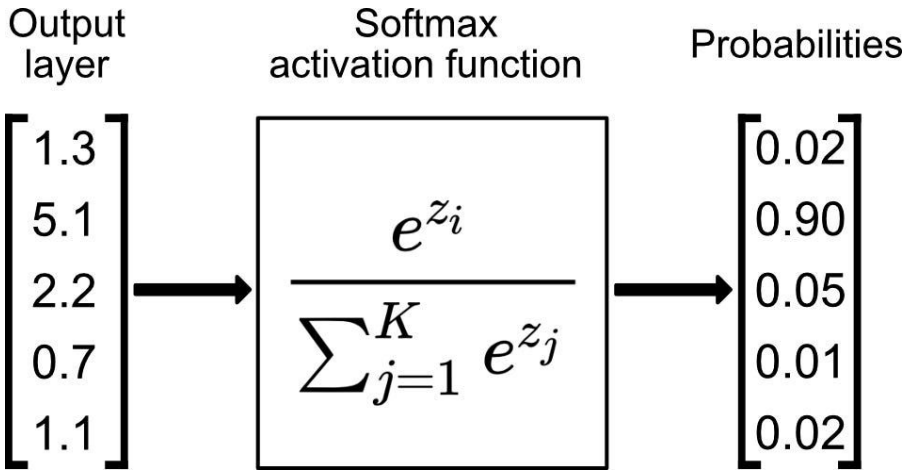
Lo más simple para utilizar una red totalmente conectada es juntar los vectores asociados a cada una de las palabras de la secuencia y después pasar este vector único por la red totalmente conectada.



Softmax

Cuando queremos predecir sobre un espacio de más de dos etiquetas, la predicción que sale de la red debe ser representada como un vector y a su vez, la etiqueta de los ejemplos también se representa como un vector (one-hot).

La función softmax convierte el vector de números de la capa de salida en un vector de probabilidades, en donde las probabilidades de cada valor son relativas a la escala de cada valor en el vector.



Función de pérdida

Debemos tener una métrica que nos diga qué tan bien predecimos un ejemplo del conjunto de entrenamiento. Si tenemos más de dos clases en el espacio de las etiquetas, debemos representar la salida de la red y la etiqueta del ejemplo como vectores.

Para comparar la diferencia entre estos dos vectores calculamos la suma negativa de los logaritmos de las clases de salida, cada una ponderada por su probabilidad. Esto se llama entropía cruzada.

$$L(\hat{y}, y) = -\sum_{k}^{K} y^{(k)} \log \hat{y}^{(k)}$$

Calculando el gradiente

Debemos calcular cuánto aporta al error cada uno de los parámetros de la red a través de derivadas parciales de la función de pérdida respecto a cada parámetro utilizando la regla de la cadena.

Este proceso se realiza a través del algoritmo de Backpropagation.

Backpropagation

El algoritmo de Backpropagation está diseñado para calcular los gradientes de la red y actualizar los parámetros para minimizar la función de pérdida.

Esto se realiza iterativamente para cada uno de los paquetes de datos que se pasan por la red.

Deep Learning en NLP

Un componente muy importante de las redes neuronales para lenguaje es el uso de una capa de embedding, un mapeo de símbolos hacia vectores continuos en un espacio vectorial de bajas dimensiones.

Las redes totalmente conectadas no toman en cuenta el orden de la entrada, por lo que sería lo mismo que utilizar un modelo lineal de clasificación. Pero las redes neuronales recurrentes son modelos especializados para datos secuenciales. Estas redes toman como entrada una secuencia de elementos y los resumen en un vector único de salida.

Redes neuronales recurrentes

El lenguaje es un fenómeno inherentemente temporal. El lenguaje es una secuencia de eventos a lo largo del tiempo y nosotros producimos lenguaje como un flujo continuo.

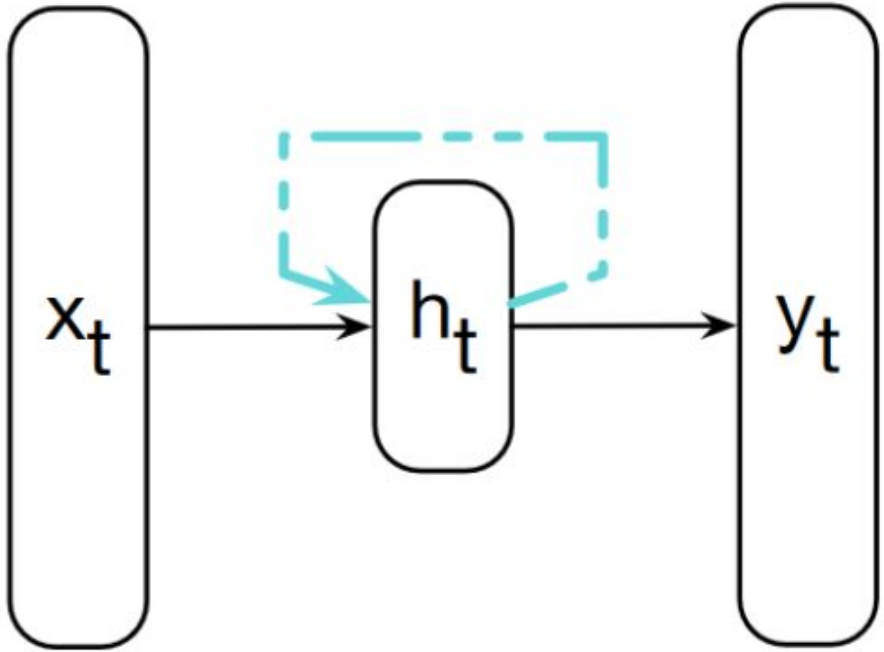
Las redes neuronales recurrentes tienen un mecanismo que maneja directamente la naturaleza temporal del lenguaje sin utilizar ventanas de tamaño fijo.

Las redes recurrentes ofrecen una nueva manera de representar el contexto anterior en sus conexiones recurrentes, permitiendo al modelo depender de información del pasado.

Redes Elman

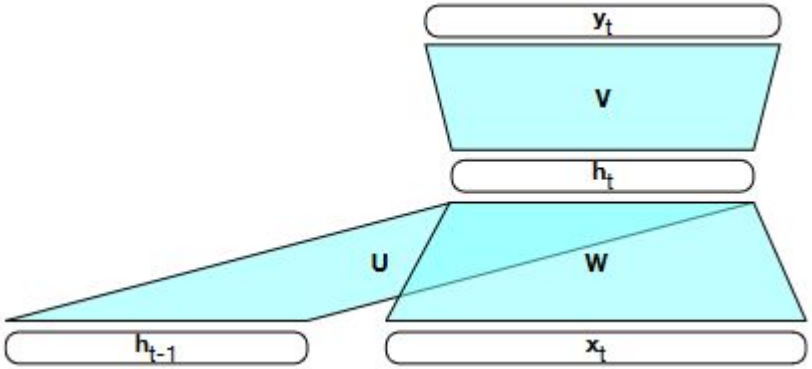
Una red recurrente es cualquier red que contiene un ciclo dentro de sus conexiones de red. Esto significa que el valor de una unidad es dependiente de su propia salida en el pasado.

Las redes Elman son las redes neuronales recurrentes más simples. Este tipo de red es la base de las redes más complejas.



Inferencia con RNN

La inferencia con RNN es casi idéntica a las redes totalmente conectadas. Para calcular la salida para una entrada, necesitamos el valor de activación para cada capa oculta. Multiplicamos la entrada **x** con la matriz de pesos **W**, y la capa de salida desde el tiempo anterior con la matriz **U** de pesos. Se suman estos valores y se pasan a través de una función de activación.



$$\mathbf{h}_t = g(\mathbf{U}\mathbf{h}_{t-1} + \mathbf{W}\mathbf{x}_t)$$
$$\mathbf{y}_t = f(\mathbf{V}\mathbf{h}_t)$$

Entrenamiento

Como en las redes totalmente conectadas, se utiliza un conjunto de entrenamiento, una función de pérdida y el algoritmo de Backpropagation para obtener los gradientes.

Ahora tenemos 3 conjuntos de pesos que actualizar: $\mathbf{W}$, los pesos para pasar de la capa de entrada hacia la capa oculta, $\mathbf{U}$, los pesos para pasar desde la capa oculta anterior a la capa oculta actual y $\mathbf{V}$, los pesos de la capa oculta hacia la capa de salida.

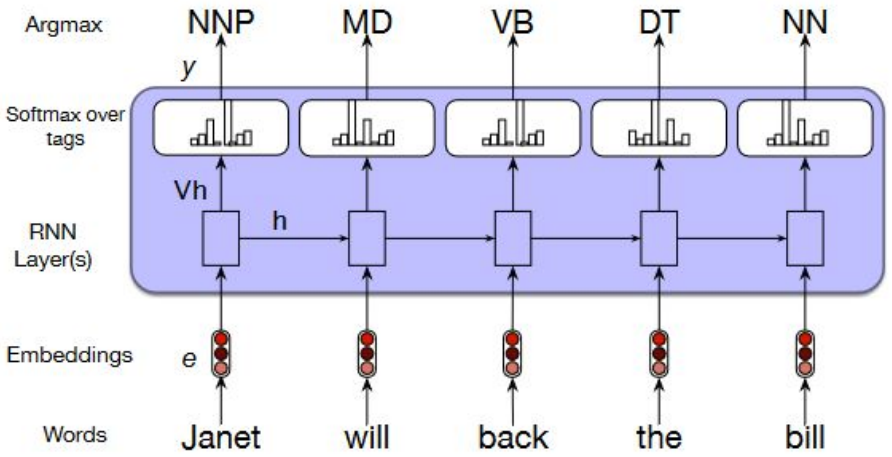
Backpropagation through time

Para calcular la función de pérdida para la salida en el tiempo t, necesitamos la capa oculta para el tiempo $t - 1$. La capa oculta en el tiempo t influencia la salida en el tiempo t y la capa oculta en el tiempo $t + 1$. Para calcular el error respecto a h_t, necesitamos saber su influencia en la salida actual como también en las salidas que siguen.

Existe una adaptación del algoritmo llamada Backpropagation through time que lidia con los problemas específicos de las redes neuronales recurrentes.

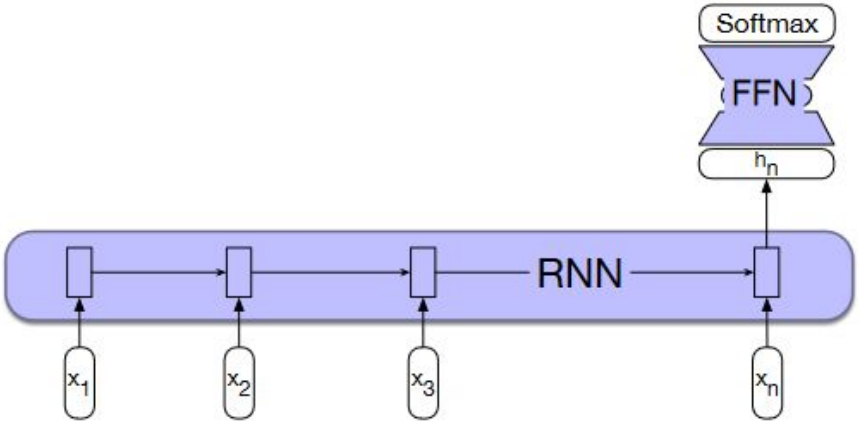
RNN para clasificación de tokens

Para clasificar cada uno de los tokens de una secuencia podemos utilizar una red neuronal recurrente que en cada paso de tiempo genere una salida en el espacio de las etiquetas.



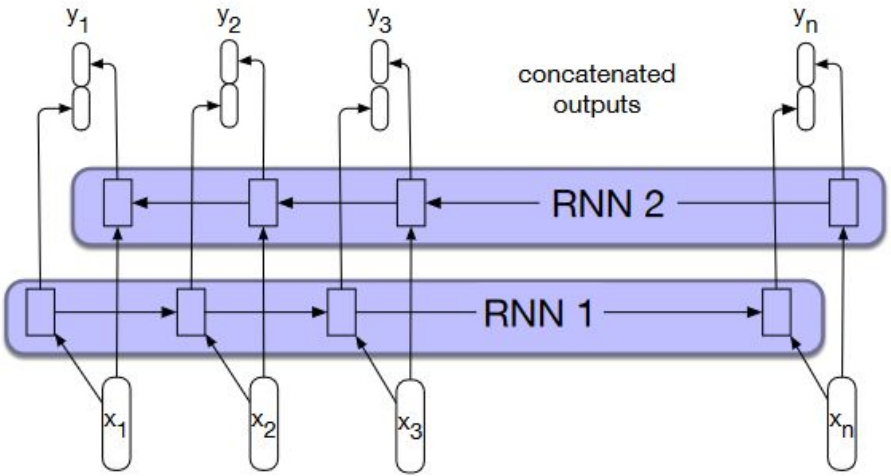
RNN para clasificación de secuencias

Para clasificar un documento completo, podemos pasar la capa oculta del último tiempo por una red totalmente conectada que haga la predicción en el espacio de las etiquetas.



RNN bidireccionales

Las redes neuronales recurrentes utilizan información desde el contexto de la izquierda para realizar sus predicciones en el tiempo **t**. Pero en muchas aplicaciones tenemos acceso a la secuencia completa de entrada y nos gustaría utilizar palabras del contexto de la derecha del tiempo **t**.



Desvanecimiento de gradientes

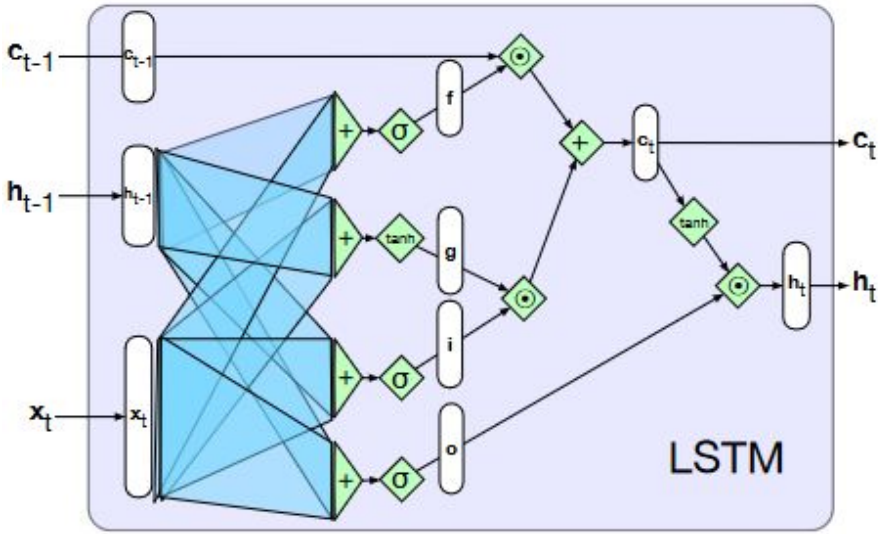
Un problema con el entrenamiento de las redes neuronales recurrentes es que necesitamos propagar los errores a través del tiempo. Como resultado, en el algoritmo de BAckpropagation las capas ocultas están sujetas a repetidas multiplicaciones determinadas por el largo de la secuencia.

El problema de esto es que los gradientes son eventualmente llevados a cero, una situación que se llama el problema del desvanecimiento de gradientes.

Long Short-term Memory

Con las redes Elman es difícil usar información distante, debido a que la información contenida en los estados ocultos tiende a ser muy local.

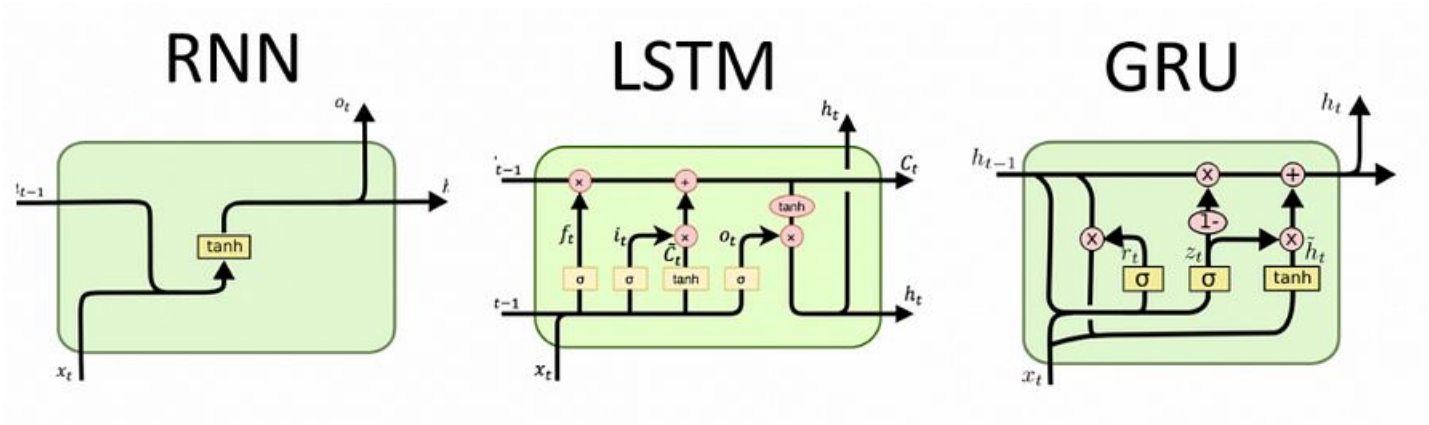
Las LSTM agregan explícitamente una capa de contexto y además controlan el flujo de información a través de parámetros adicionales.



Gated recurrent unit

La arquitectura LSTM es muy efectiva, pero es muy compleja. Las GRU son una alternativa a las LSTM.

Las GRU tienen muchas menos compuertas y no tienen un componente de memoria separado.



Encoder-Decoder

Este tipo de modelos también llamados redes secuencia-a-secuencia son capaces de generar secuencias de tamaño arbitrario, contextualmente apropiadas dada una secuencia de entrada.