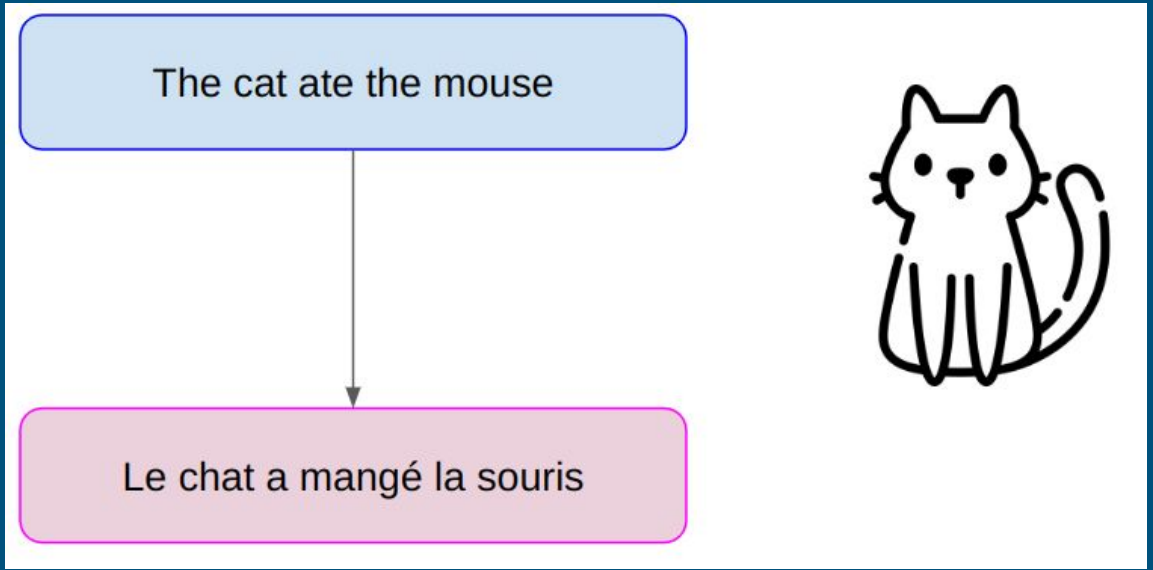
UNAM iimas INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS

Modelos basados en recurrencias

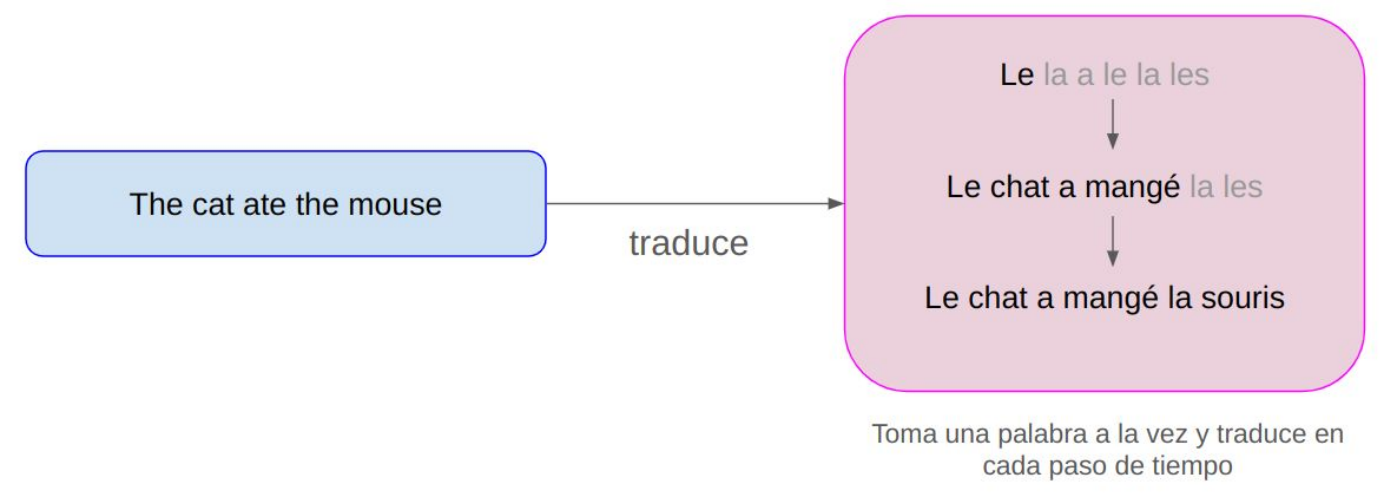
Materia: Procesamiento de lenguaje natural

Blanca Vázquez

Traducción de sentencias



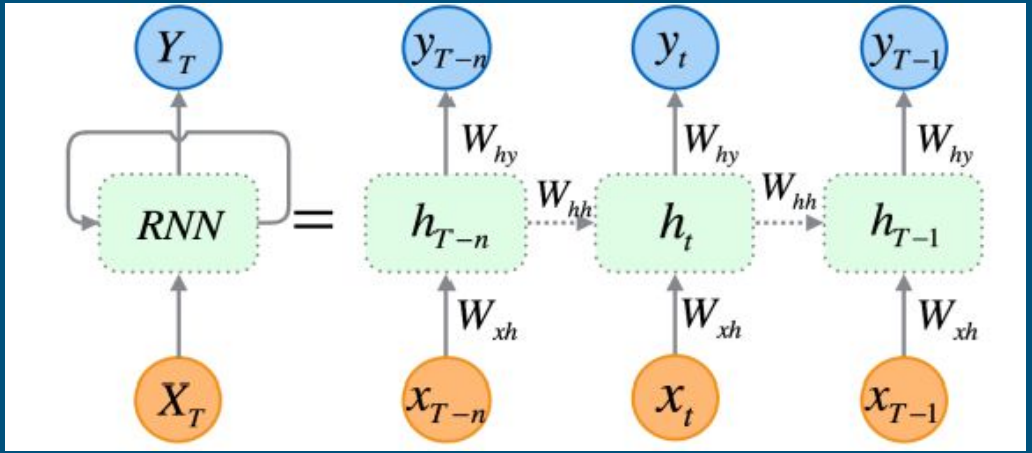
Traducción de sentencias (vista general)



Redes neuronales recurrentes

RNN, (por las siglas en inglés de *Recurrent Neural Network)* es un tipo de red neuronal artificial diseñada para procesar datos secuenciales.

A diferencia de las redes neuronales tradicionales, las RNN tienen conexiones que permiten que la información fluya en un ciclo, lo que les permite mantener un estado interno y procesar secuencias de longitud variable



En una RNN las conexiones entre nodos forman un grafo dirigido a lo largo de una secuencia temporal.

Imagen tomada de Image Caption Generation Methodologies, 2021.

Forward propagation

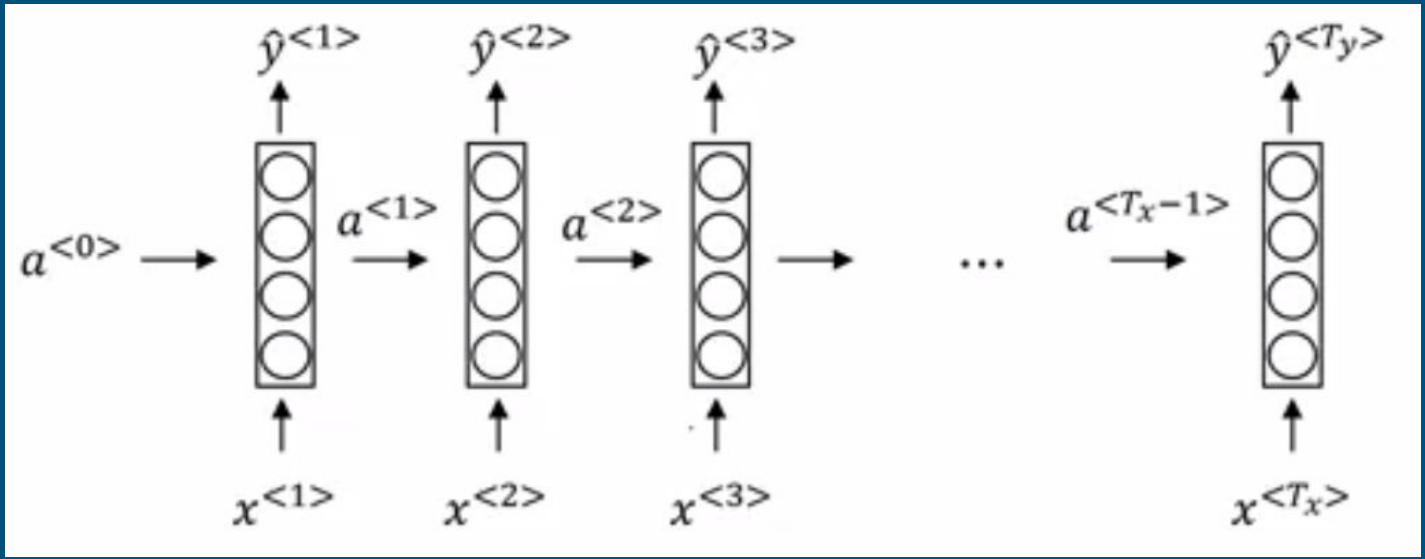


Imagen tomada de Andrew Ng.

Forward propagation

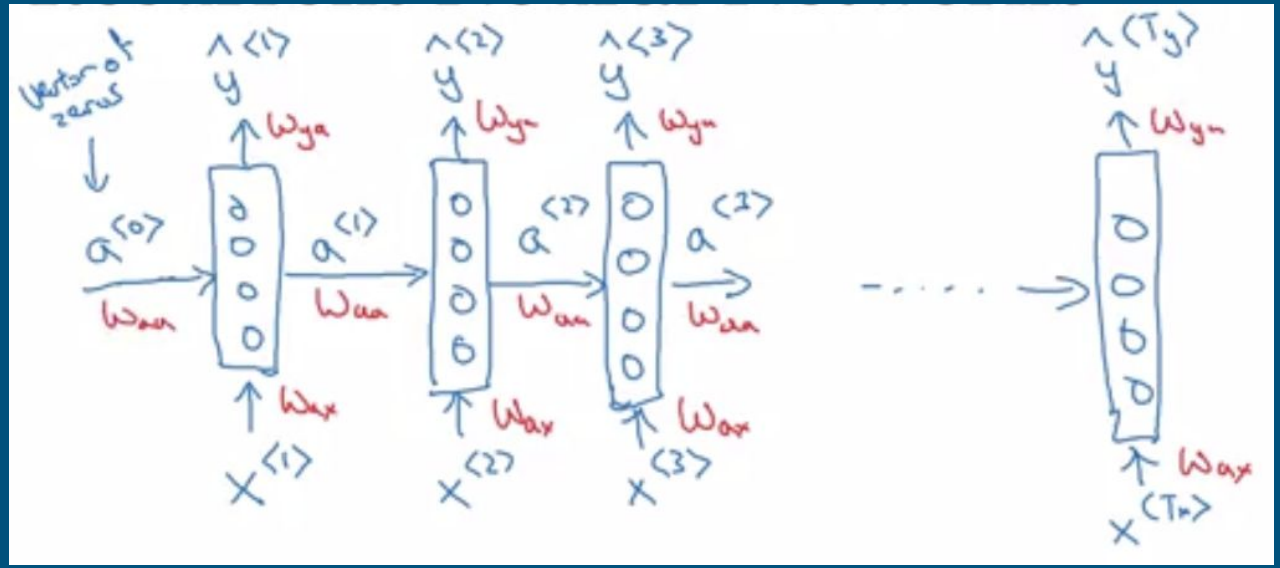
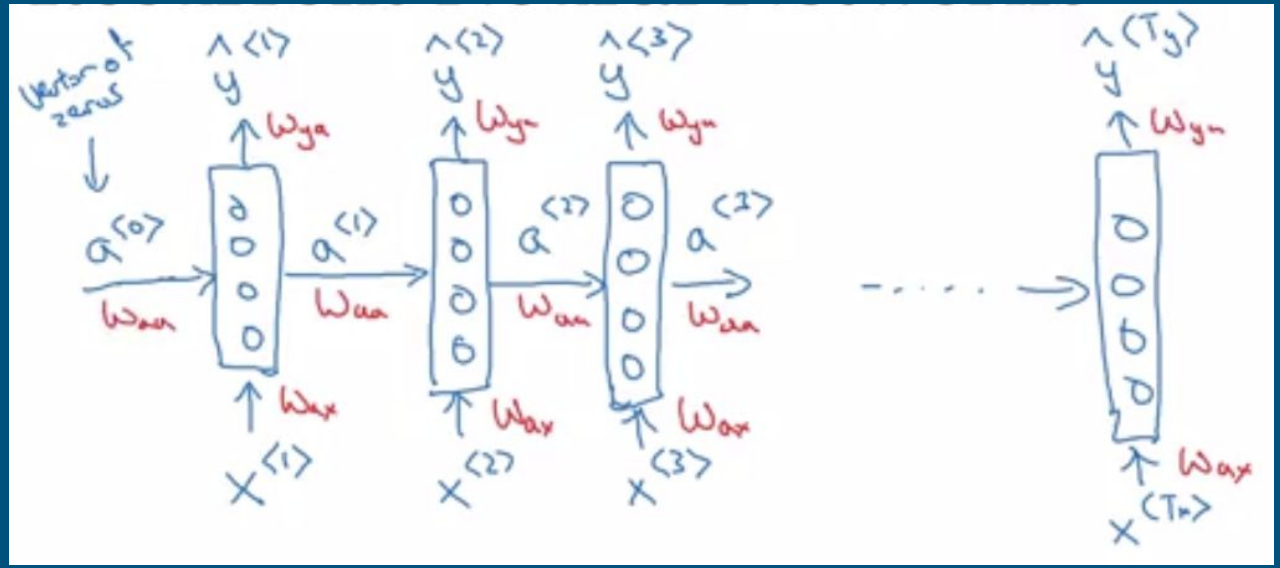


Imagen tomada de Andrew Ng.

Forward propagation



$$a^{<t>} = g(W_{aa}a^{<t>} + W_{ax}x^{<t>} + b_a)$$

Tanh / Relu

$$\hat{y}^{<t>} = g(W_{ya}a^{<t>} + b_y)$$

Sigmoid /softmax

Imagen tomada de Andrew Ng.

Backpropagation

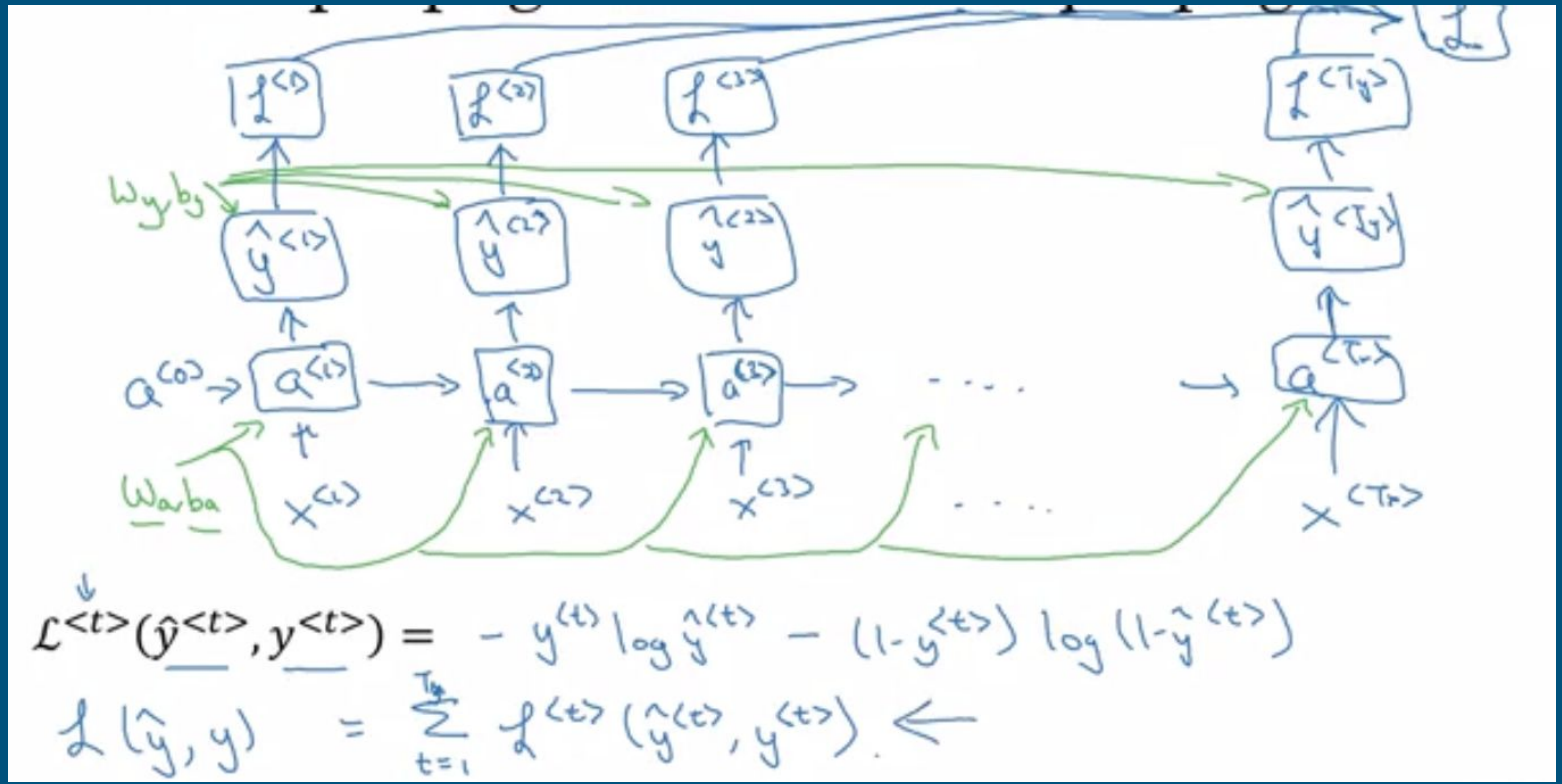


Imagen tomada de Andrew Ng.

Visualizando los estados ocultos en RNN

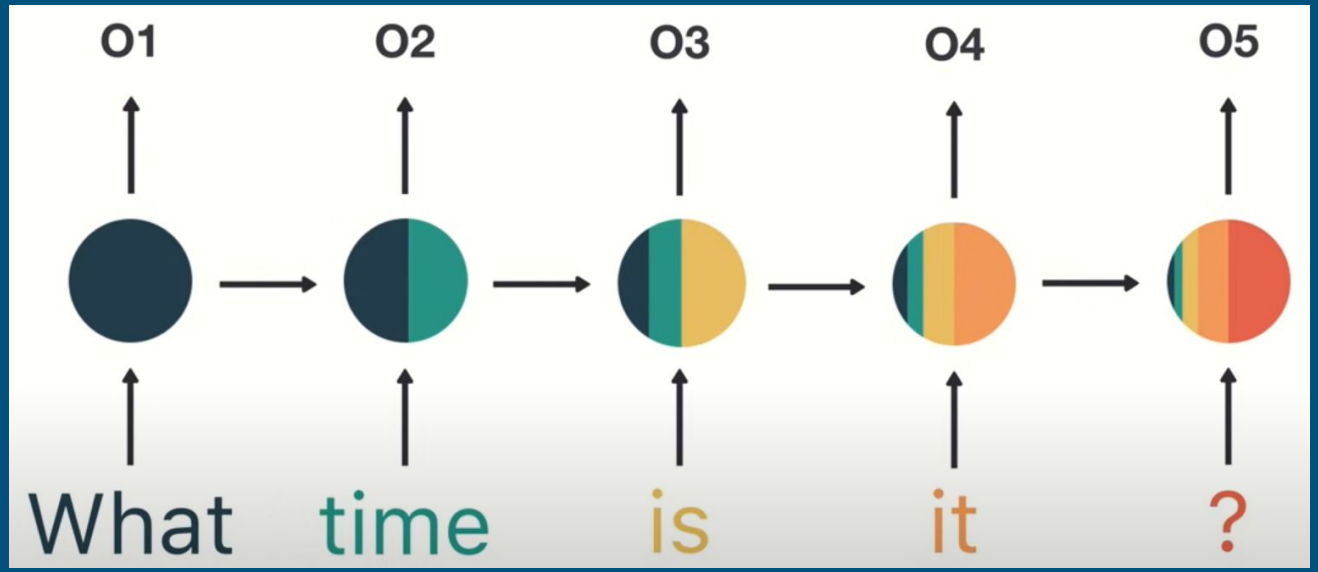
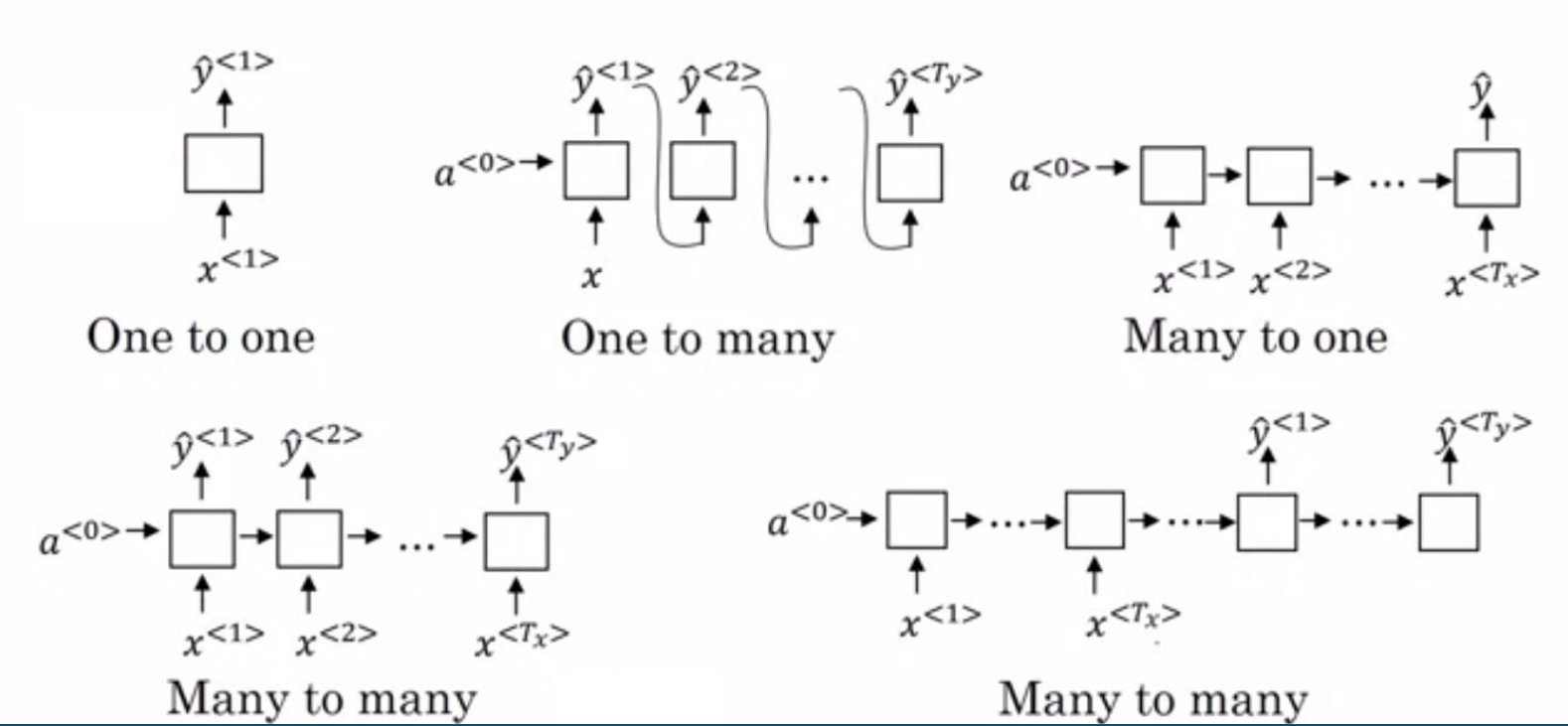
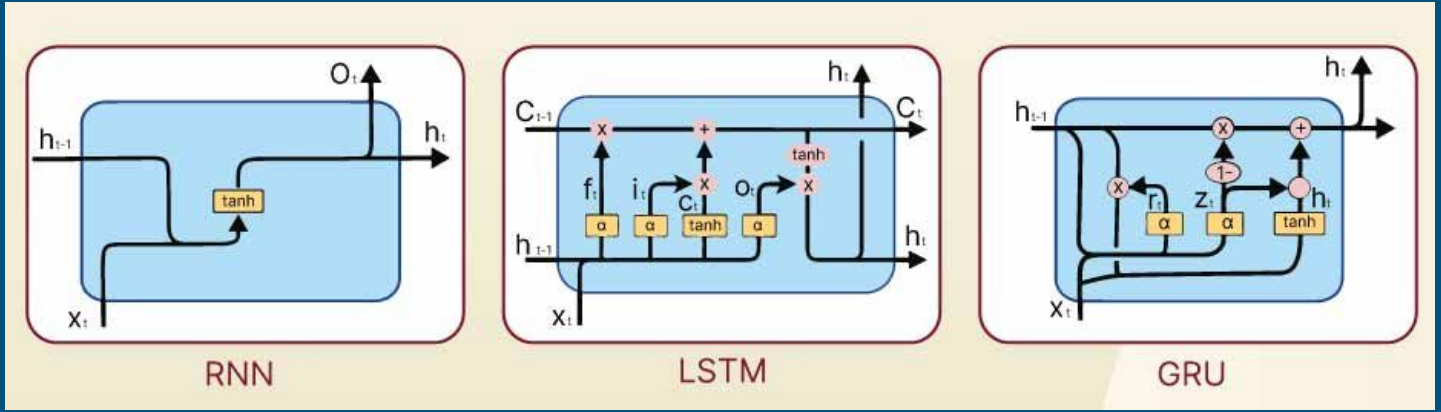


Imagen tomada de Will Koehrsen, 2018.

Tipos de RNN



RNN vs LSTM vs GRU



Tanto las redes LSTM y GRU están diseñadas para abordar el problema del desvanecimiento del gradiente.

Imagen tomada de Atul Harsha, 2023

LSTM (Long Short-Term Memory)

- LSTM fueron desarrolladas por Hochreiter y Schmidhuber en 1997.
- A diferencia de la RNN que tienen nodos recurrentes, las LSTM tienen celdas de memoria.
- Cada celda de memoria contiene un estado interno, es decir, un nodo con una arista recurrente auto conectada de peso fijo.

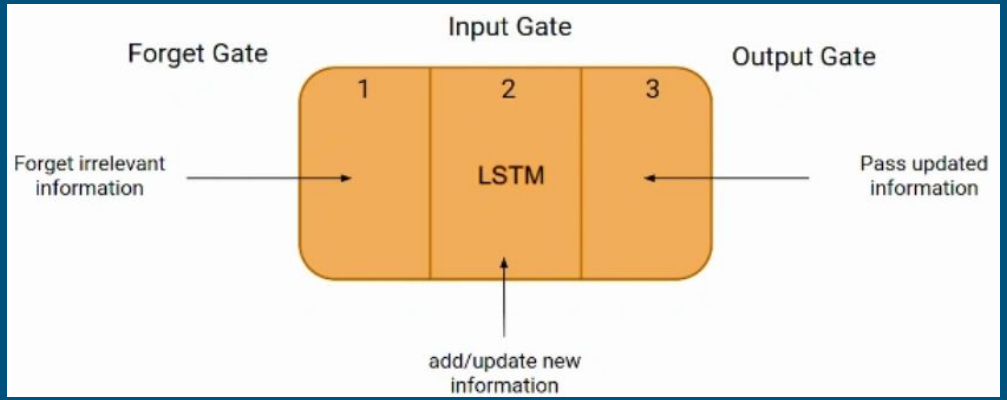


Imagen tomada de Shipra Saxena, 2025.

Puerta de olvido (Forget gate)

En una celda LSTM, el primer paso es decidir si se debe conservar la información del paso de tiempo anterior u olvidarla. La ecuación de la puerta de olvido es:

$$\mathbf{F}_t = \sigma(\mathbf{X}_t \mathbf{W}_{\text{xf}} + \mathbf{H}_{t-1} \mathbf{W}_{\text{hf}} + \mathbf{b}_{\text{f}}),$$

donde:
X_t es la entrada en el paso de tiempo actual
W_{xf} es el peso asociado a la entrada
H_{t-1} es el estado oculto del paso previo
W_{hf} es la matriz de pesos asociada al estado oculto
b_f es el sesgo

Puerta de olvido (Forget gate)

- Posteriormente, se le aplica una función sigmoide. Esto convierte a F_t en un número entre 0 y 1. Este F_t se multiplica posteriormente por el estado de la celda del paso anterior.

$$\mathbf{F}_t = \sigma(\mathbf{X}_t \mathbf{W}_{\text{xf}} + \mathbf{H}_{t-1} \mathbf{W}_{\text{hf}} + \mathbf{b}_{\text{f}}),$$

$C_{t-1} * F_t = 0$ … Si $F_t = 0$ (olvida todo)

$C_{t-1} * F_t = 0$ … Si $F_t = 1$ (no olvides nada)

Puerta de entrada (Input gate)

Supongamos la siguiente frase:

Bob sabe nadar. Me contó por teléfono que había servido en la

Marina durante cuatro largos años.

En ambas oraciones: ¿qué información deberíamos guardar?
¿Qué dato nos gustaría que el modelo guarde como dato futuro?

Puerta de entrada (Input gate)

La puerta de entrada se utiliza para cuantificar la importancia de la nueva información contenida en la entrada. Aquí está la ecuación de la puerta de entrada.

$$\mathbf{I}_t = \sigma(\mathbf{X}_t \mathbf{W}_{\text{xi}} + \mathbf{H}_{t-1} \mathbf{W}_{\text{hi}} + \mathbf{b}_{\text{i}}),$$

donde:
X_t es la entrada en el paso de tiempo actual
W_{xi} es el peso asociado a la entrada
H_{t-1} es el estado oculto del paso previo
W_{hi} es la matriz de pesos asociada al estado oculto
b_i es el sesgo

Puerta de entrada (Input gate)

Nuevamente, se aplica una función sigmoide. Como resultado, el valor de I en la marca de tiempo t estará entre 0 y 1.

$$\mathbf{I}_t = \sigma(\mathbf{X}_t \mathbf{W}_{\text{xi}} + \mathbf{H}_{t-1} \mathbf{W}_{\text{hi}} + \mathbf{b}_{\text{i}}),$$

- Ahora, la nueva información que debe pasarse al estado de la celda es una función de un estado oculto en la marca de tiempo anterior t-1 y la entrada X en la marca de tiempo t.
- La función de activación a usar es tanh.
- Debido a tanh, el valor de la nueva información estará entre -1 y 1. Si el valor de C_t es negativo, la información se resta del estado de la celda; si es positivo, se añade al estado de la celda en la marca de tiempo actual.

Nueva información

$$C_t = \tanh(X_t W_{xi} + H_{t-1} W_{hi} + b_i)$$

Sin embargo, el C_t no se añadirá directamente al estado de la celda.

Estado interno de la celda de memoria

La puerta de entrada I_t regula cuánto debemos tener en cuenta vía C_t y la compuerta de olvido F_t aborda qué tanto de la celda del estado anterior debe retenerse C_{t-1}. Utilizando el operador de producto Hadamard (elemento por elemento), llegamos a la siguiente ecuación de actualización:

$$C_t = F_t \odot C_{t-1} + I_t \odot N_t$$

- Si la puerta de olvido siempre es 1 y la puerta de entrada siempre es 0, el estado interno de la celda de memoria C_{t-1} se mantendrá constante indefinidamente, pasando sin cambios a cada paso de tiempo subsiguiente.

Estado interno de la celda de memoria

La puerta de entrada I_t regula cuánto debemos tener en cuenta vía C_t y la compuerta de olvido F_t aborda qué tanto de la celda del estado anterior debe retenerse C_{t-1}. Utilizando el operador de producto Hadamard (elemento por elemento), llegamos a la siguiente ecuación de actualización:

$$C_t = F_t \odot C_{t-1} + I_t \odot N_t$$

- Sin embargo, las puertas de entrada y las puertas de olvido otorgan al modelo la flexibilidad de aprender cuándo mantener este valor sin cambios y cuándo perturbarlo en respuesta a entradas posteriores.
- En la práctica, este diseño mitiga el problema del gradiente de desaparición, lo que resulta en modelos mucho más fáciles de entrenar, especialmente al trabajar con conjuntos de datos con secuencias de gran longitud.

Puerta de salida (Output gate)

Supongamos la siguiente frase:

Bob luchó solo contra el enemigo y murió por su país. Por sus contribuciones, valiente _____.

La palabra “valiente” debería ser influenciado más por “murió por su país” o “contribuciones”.
¿Qué palabra puede completar la segunda oración?

Puerta de salida (Output gate)

Predecir la palabra de salida es la función de la Output gate. La ecuación es:

$$\mathbf{O}_t = \sigma(\mathbf{X}_t \mathbf{W}_{\text{xo}} + \mathbf{H}_{t-1} \mathbf{W}_{\text{ho}} + \mathbf{b}_{\text{o}}),$$

donde:
X_t es la entrada en el paso de tiempo actual
W_{xo} es el peso asociado a la entrada
H_{t-1} es el estado oculto del paso previo
W_{ho} es la matriz de pesos asociada al estado oculto
b_o es el sesgo

Puerta de salida (Output gate)

Su valor también estará entre 0 y 1 debido a esta función sigmoidea. Para calcular el estado oculto actual, usaremos O_t y tanh del estado de celda actualizado, como se muestra a continuación.

$$\mathbf{H}_t = \mathbf{O}_t \odot \tanh(\mathbf{C}_t).$$

- Cuando la puerta de salida está cerca de 1, permitimos que el estado interno de la celda de memoria afecte a las capas subsiguientes.
- Cuando la puerta de salida es cercana a 0, impedimos que la memoria actual afecte a otras capas de la red en el intervalo de tiempo actual.

Resumen de puertas en una celda LSTM

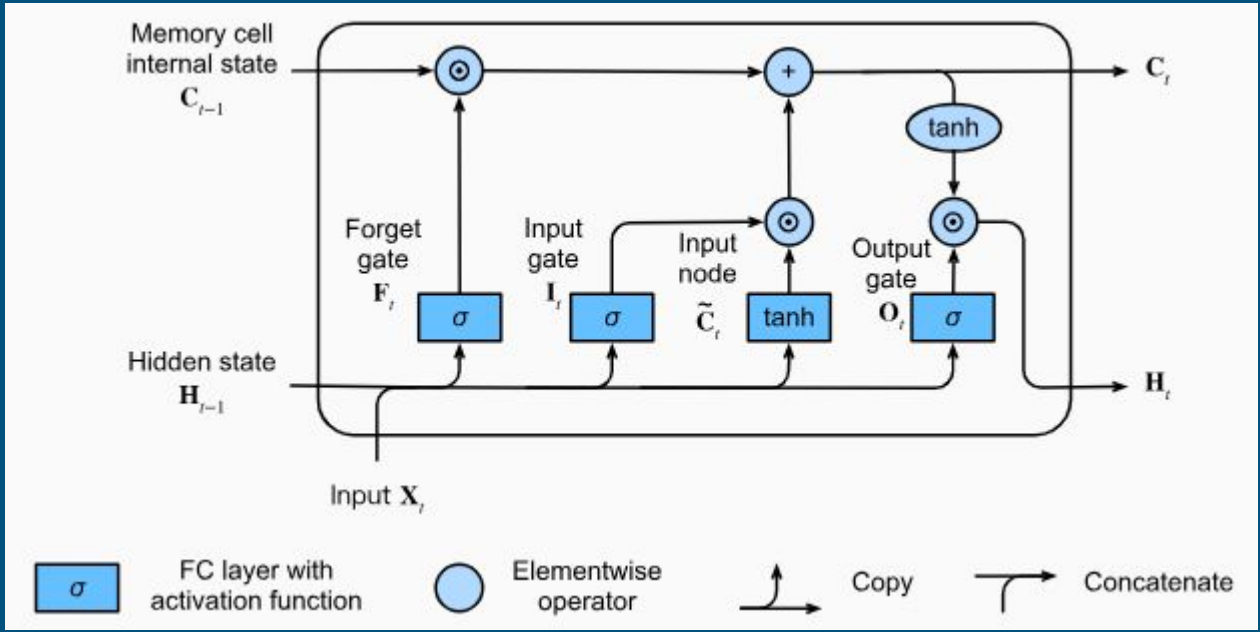


Imagen tomada de Dive into Deep Learning.

GRU (Gated Recurrent Unit)

- Fueron propuestas por Chung, Gulcehre, Cho y Bengio en 2014.
- Es una variación de las RNN.
- Incorpora mecanismos de puerta (gates) para controlar el flujo de información a través de la red.
- La puerta de actualización y puerta de reinicio, permiten a la GRU decidir qué información del pasado debe retenerse y cuál debe olvidarse
- Estas puertas ayudan a la red a capturar dependencias a largo plazo.

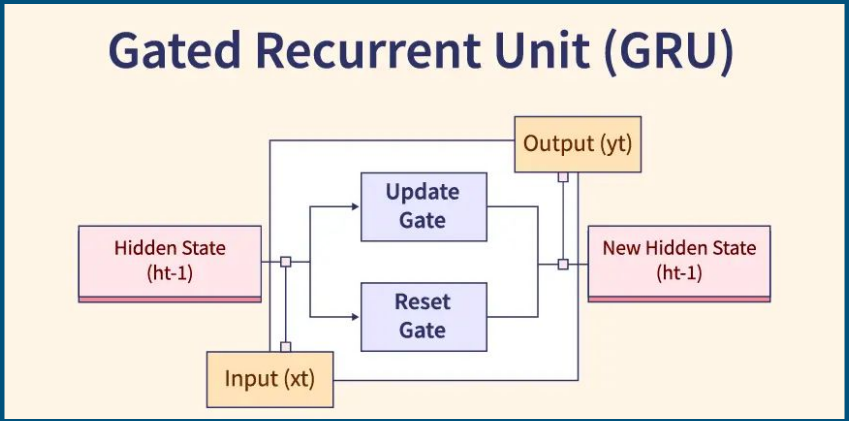
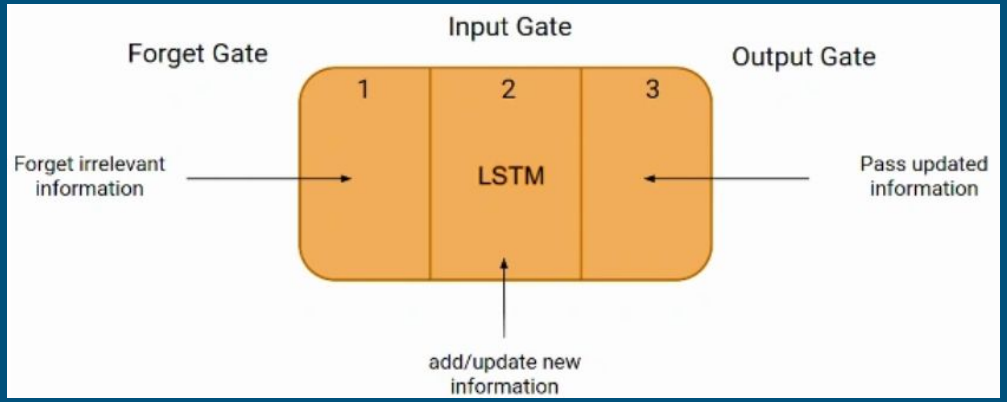
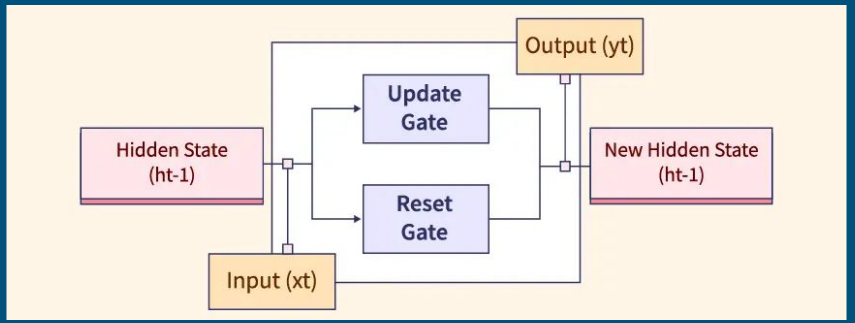


Imagen tomada de Cathrine Jeeva, 2023.

GRU vs LSTM

A diferencia de las redes LSTM que tienen 3 puertas (olvido, entrada y salida), la red GRU solo tiene 2 puertas (actualización y de reinicio), generando una arquitectura más simple y rápida de entrenar.





Vista general de la GRU

En cada paso de tiempo t, la celda toma como entrada: X_t y el estado oculto H_{t-1} (del paso previo t-1).

Como resultado, genera un nuevo estado oculto H_t que nuevamente pasa al siguiente paso de tiempo.

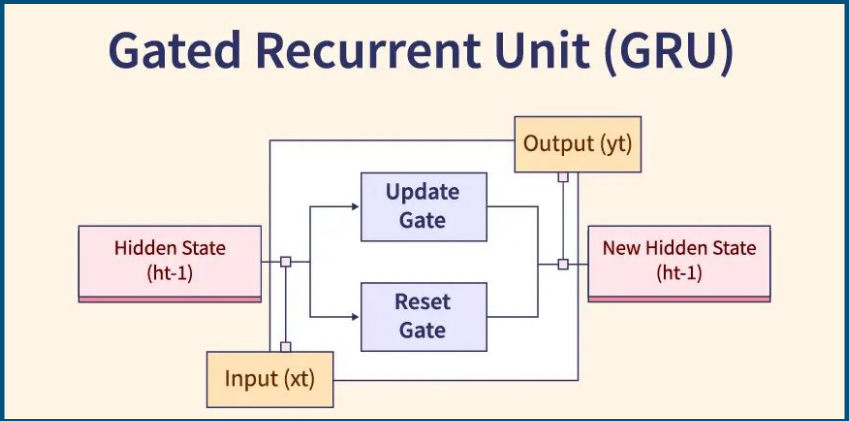


Imagen tomada de Cathrine Jeeva, 2023.

Puerta de reinicio (Reset gate)

La puerta de reinicio es responsable de la memoria a corto plazo de la red, es decir, del estado oculto (Ht) y está dada por la ecuación:

$$\mathbf{R}_t = \sigma(\mathbf{X}_t \mathbf{W}_{\text{xr}} + \mathbf{H}_{t-1} \mathbf{W}_{\text{hr}} + \mathbf{b}_{\text{r}}),$$

donde:
X_t es la entrada en el paso de tiempo actual
W_{xr} es el peso asociado a la entrada
H_{t-1} es el estado oculto del paso previo
W_{hr} es la matriz de pesos asociada al estado oculto
b_r es el sesgo

El valor obtenido de R_t tendrá un rango de 0 a 1 debido a una función sigmoide.

Puerta de actualización (Update gate)

La puerta de actualización es la responsable de la memoria a largo plazo y la ecuación de la puerta se muestra a continuación.

$$\mathbf{Z}_t = \sigma(\mathbf{X}_t \mathbf{W}_{\text{xz}} + \mathbf{H}_{t-1} \mathbf{W}_{\text{hz}} + \mathbf{b}_{\text{z}}),$$

donde:
X_t es la entrada en el paso de tiempo actual
W_{xz} es el peso asociado a la entrada
H_{t-1} es el estado oculto del paso previo
W_{hz} es la matriz de pesos asociada al estado oculto
b_z es el sesgo

El valor obtenido de Z_t tendrá un rango de 0 a 1 debido a una función sigmoide.

¿Cómo trabaja la GRU?

1) En cada paso de tiempo, GRU toma 2 entradas:
- X_t es la entrada en el paso de tiempo actual
- H_{t-1} es el estado oculto del paso previo

2) Se calculan los valores de cada puerta: R_t y Z_t

- Se aplican una función de activación (sigmoide) que ayudarán a controlar el flujo de la información

3) Estas puertas ayudarán a calcular el estado oculto candidato.

Estado oculto candidato (Candidate hidden state)

Para calcular el estado oculto candidato se emplea el resultado de la puerta de reinicio R_t

$$\tilde{\mathbf{H}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{\text{xh}} + (\mathbf{R}_t \odot \mathbf{H}_{t-1}) \mathbf{W}_{\text{hh}} + \mathbf{b}_{\text{h}}),$$

donde:
X_t es la entrada en el paso de tiempo actual
W_{xh} es el peso asociado a la entrada
R_t es el resultado de la puerta de reinicio
H_{t-1} es el estado oculto del paso previo
W_{hh} es la matriz de pesos asociada al estado oculto
b_h es el sesgo
$\odot$ producto Hadamard (elemento por elemento)
tanh es la función de activación

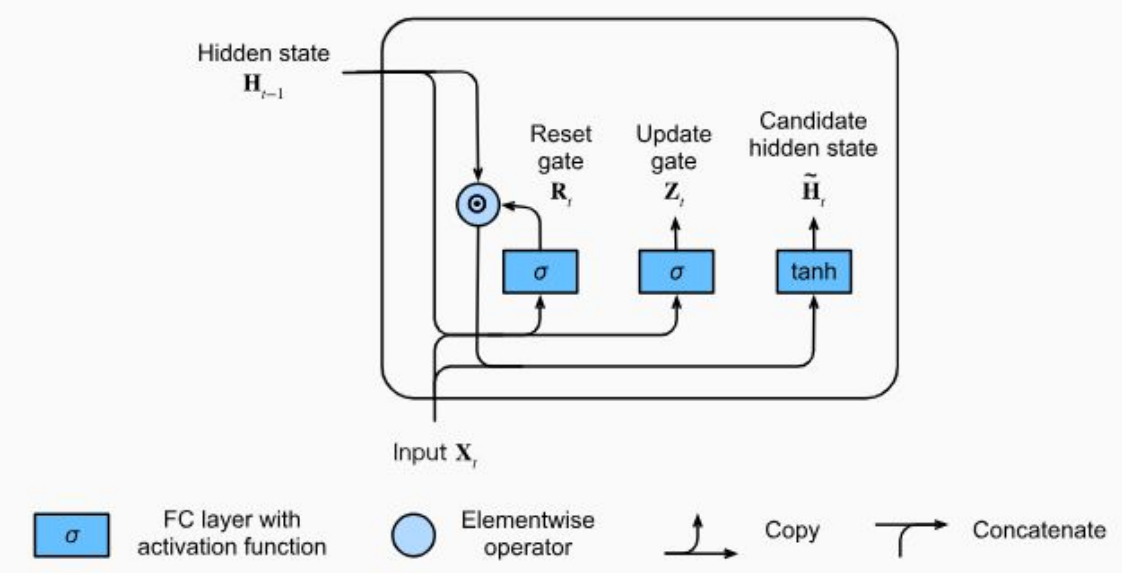
Estado oculto candidato (Candidate hidden state)

$$\tilde{\mathbf{H}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{\text{xh}} + (\mathbf{R}_t \odot \mathbf{H}_{t-1}) \mathbf{W}_{\text{hh}} + \mathbf{b}_{\text{h}}),$$

La parte más importante de esta ecuación es cómo usamos el valor de la puerta de reinicio para controlar la influencia del estado oculto previo en el estado candidato.

- Si el valor de R_t es igual a 1, entonces se considera toda la información del estado oculto del paso previo H_{t-1}.
- Si el valor de R_t es 0, significa que la información del estado oculto previo se ignora por completo.

Estado oculto candidato (Candidate hidden state)



Recibe el nombre de candidato debido a que aún es necesario incorporar la puerta de actualización.

Imagen tomada de Dive into Deep Learning.

Estado oculto (Hidden state)

- Una vez obtenido el estado candidato, este se utiliza para generar el estado oculto actual H_t.
- Aquí es donde entra en juego la puerta de actualización Z_t.

$$\mathbf{H}_t = \mathbf{Z}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{H}}_t.$$

Esta ecuación es muy interesante: en lugar de usar una puerta independiente como en la arquitectura LSTM, en la GRU se utiliza una única puerta de actualización para controlar tanto la información histórica (H_{t-1}) como la nueva información proveniente del estado candidato $\tilde{H}_t$.

Estado oculto (Hidden state)

El estado oculto está dado por la siguiente ecuación:

$$\mathbf{H}_t = \mathbf{Z}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{H}}_t.$$

donde:
Z_t es el resultado de la puerta de actualización
H_{t-1} es el estado oculto del paso previo
$\tilde{H}_t$ es el resultado del estado oculto candidato
$\odot$ producto Hadamard (elemento por elemento)

Estado oculto (Hidden state)

$$\mathbf{H}_t = \mathbf{Z}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{H}}_t.$$

- Si el valor de Z_t es cercano a 1, se retiene la información del estado anterior (H_{t-1}).
- Si el valor de Z_t es cercano a 0, el nuevo estado latente se aproxima al estado oculto candidato $\tilde{H}_t$.

Resumen de puertas en una celda GRU

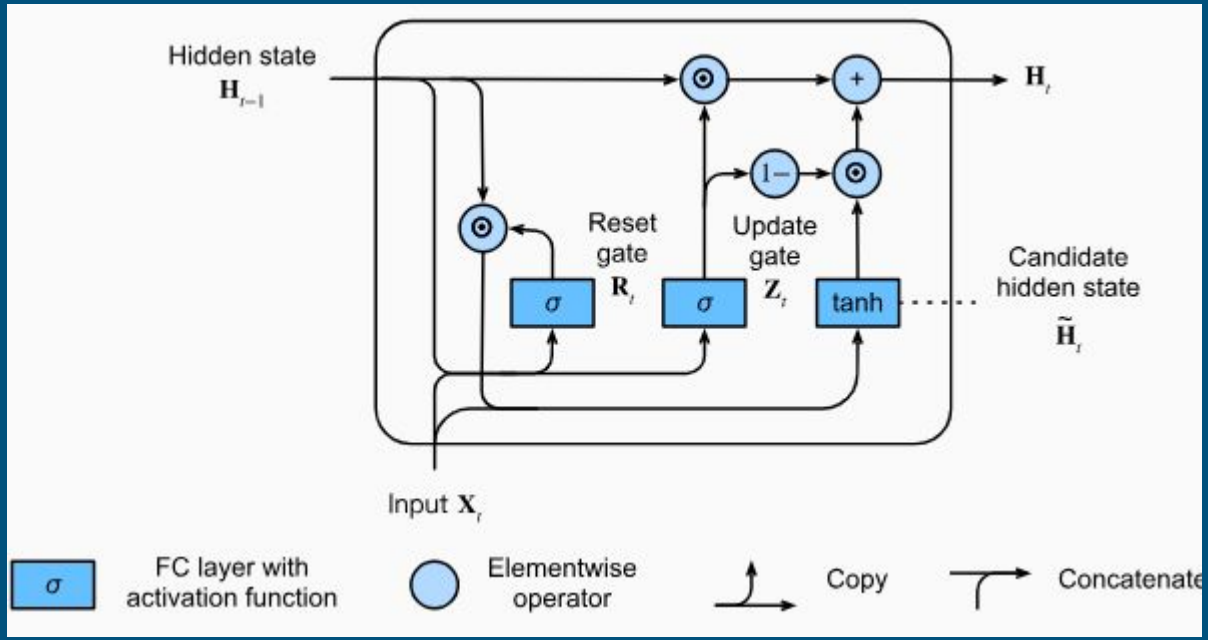


Imagen tomada de Dive into Deep Learning.

Comparativa

	GRU	LSTM
Estructura	Simple (2 puertas)	Compleja (3 puertas)
Parámetros	Pocos	Muchos
Entrenamiento	Rápida en entrenar	Lento en entrenar
Complejidad	En la mayoría de los casos, tiende a utilizar menos recursos de memoria debido a su estructura más simple y con menos parámetros.	Tiene una estructura más compleja y una mayor cantidad de parámetros, por lo que podría requerir más recursos de memoria.
Rendimiento	Es posible que no funcionen tan bien como las LSTM en tareas que requieren modelar dependencias a muy largo plazo o patrones secuenciales complejos.	LSTM tiene ventajas sobre GRU en tareas de comprensión del lenguaje natural y traducción automática.

Time to code