Redes Neuronales Recurrentes

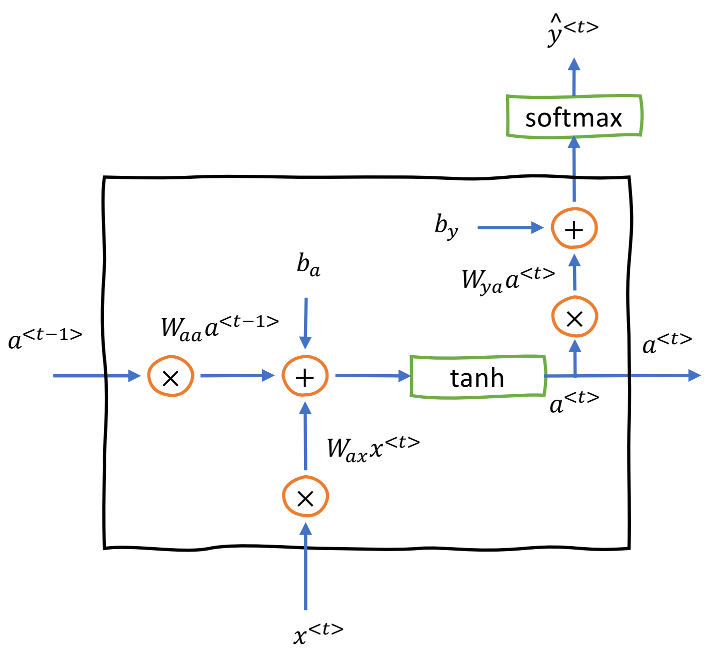
Francisco Cervantes

Octubre, 2019

HOY ...

- Modelo de una RNN
- Forward y backward propagation
- Diferentes tipos de RNNs
- Gated Recurrent Unit (GRU)
- Long Short Term Memory (LSTM)

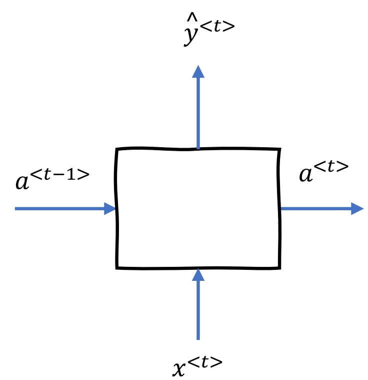
Modelo de una RNN



$$a^{<t>} = tanh(w_{aa}a^{<t-1>} + w_{ax}x^{<t>} + b_a)$$

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

De manera simplificada, la celda se puede representar cómo:



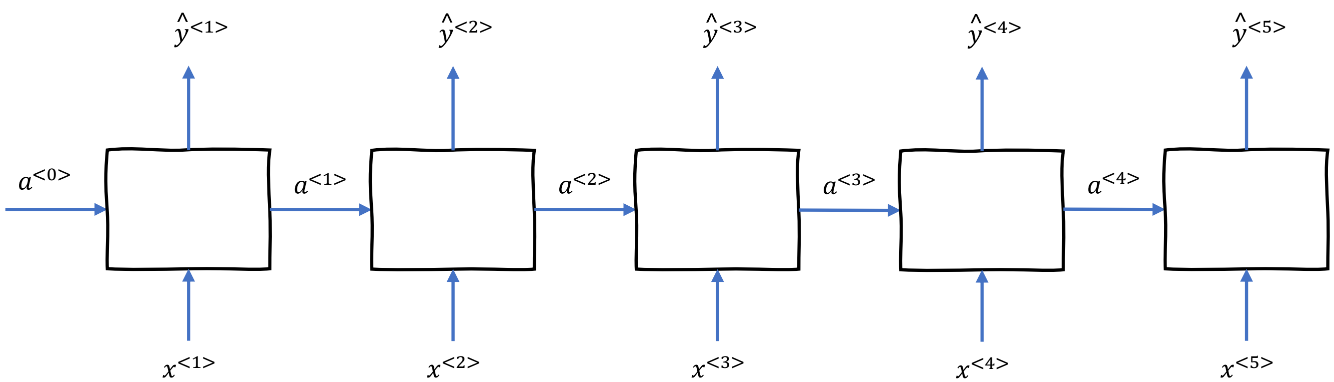
$$a^{<0>} = [\ 0 \ldots 0\ ]$$

$$a^{<1>} = g_1(w_{aa}a^{<0>} + w_{ax}x^{<1>} + b_a)$$

$$\hat{y}^{<1>} = g_2(w_{ya}a^{<1>} + b_y)$$

Modelo de una RNN

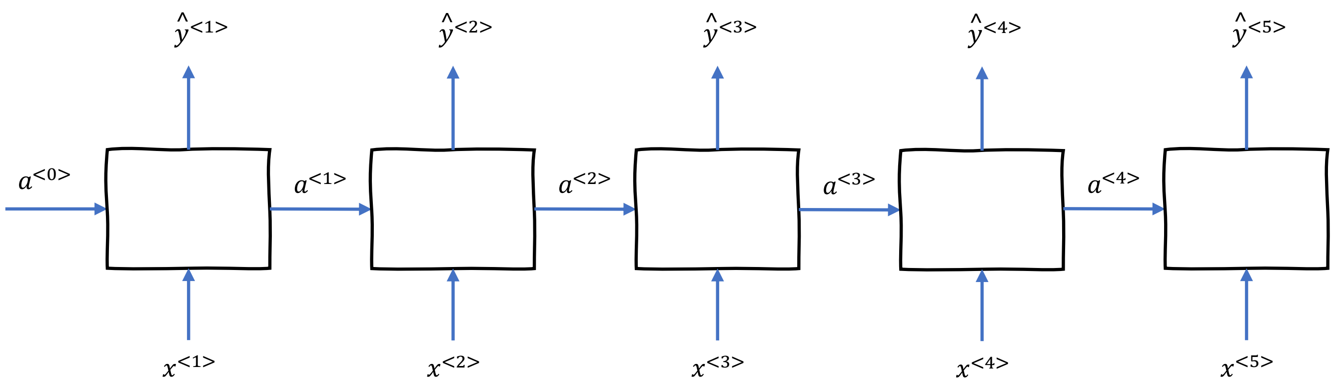
Dado el siguiente modelo con 5 celdas, en donde $T_x = T_y$, y donde $g_1 = tanh,\ g_2 = sigmoid$,



Para un ejemplo de entrenamiento x = $(x^{<1>}, x^{<2>}, x^{<3>}, x^{<4>}, x^{<5>})$, el forward propagation esta dado por:

Modelo de una RNN

Dado el siguiente modelo con 5 celdas, en donde $T_x = T_y$, y donde $g_1 = tanh,\ g_2 = sigmoid,$



Para un ejemplo de entrenamiento x = $(x^{<1>}, x^{<2>}, x^{<3>}, x^{<4>}, x^{<5>})$, el forward propagation esta dado por:

$a^{<1>} = tanh(w_{aa}a^{<0>} + w_{ax}x^{<1>} + b_a)$

$\hat{y}^{<1>} = sigmoid(w_{ya}a^{<1>} + b_y)$

$a^{<2>} = tanh(w_{aa}a^{<1>} + w_{ax}x^{<2>} + b_a)$

$\hat{y}^{<2>} = sigmoid(w_{ya}a^{<2>} + b_y)$

$a^{<3>} = tanh(w_{aa}a^{<2>} + w_{ax}x^{<3>} + b_a)$

$\hat{y}^{<3>} = sigmoid(w_{ya}a^{<3>} + b_y)$

$a^{<4>} = tanh(w_{aa}a^{<3>} + w_{ax}x^{<4>} + b_a)$

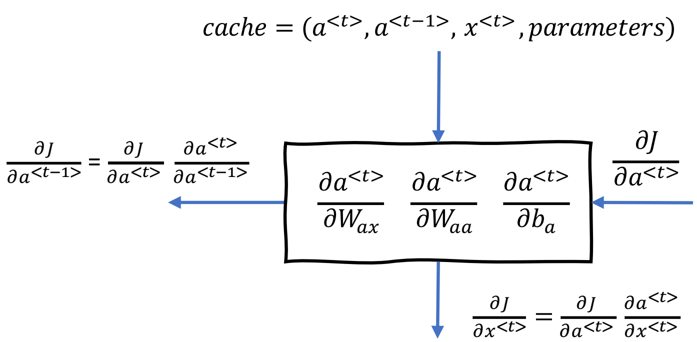
$\hat{y}^{<4>} = sigmoid(w_{ya}a^{<4>} + b_y)$

$a^{<5>} = tanh(w_{aa}a^{<4>} + w_{ax}x^{<5>} + b_a)$

$\hat{y}^{<5>} = sigmoid(w_{ya}a^{<5>} + b_y)$

Modelo de una RNN

El backward propagation sobre una celda de una red neuronal recurrente, esta dada por:



$$a^{<t>} = \tanh(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

$$\frac{\partial \tanh(x)}{\partial x} = 1 - \tanh(x)^2$$

$$\frac{\partial a^{<t>}}{\partial W_{ax}} = (1 - \tanh(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)^2) \cdot x^{<t>T}$$

$$\frac{\partial a^{<t>}}{\partial W_{aa}} = (1 - \tanh(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)^2) \cdot a^{<t-1>T}$$

$$\frac{\partial a^{<t>}}{\partial b_a} = \sum_{batch} (1 - \tanh(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)^2)$$

$$\frac{\partial a^{<t>}}{\partial x^{<t>}} = W_{ax}^T \cdot (1 - \tanh(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)^2)$$

$$\frac{\partial a^{<t>}}{\partial a^{<t-1>}} = W_{aa}^T \cdot (1 - \tanh(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)^2)$$

De igual forma que en una red neuronal clásica, la derivada de la función de costo se propaga hacía atrás de la RNN mediante el uso de la regla de la cadena. También se utiliza la regla de la cadena para calcular los gradientes de W_{aa}, W_{ax} y b_a para actualizar los parámetros.

Diferentes tipos de RNNs

- Reconocimiento del discurso

"El valor de una idea radica en el uso de la misma"

- Clasificación de sentimientos

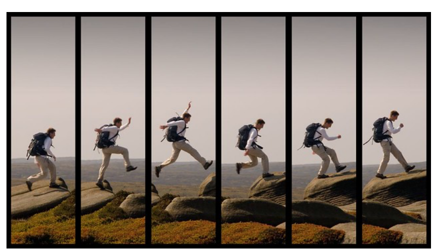
"La cinta consigue ser controversial en cada festival que se presenta y no deja a nadie indiferente"

- Traducción automática

¿Quieres cantar con migo?

Voulez-vous chanter avec moi?

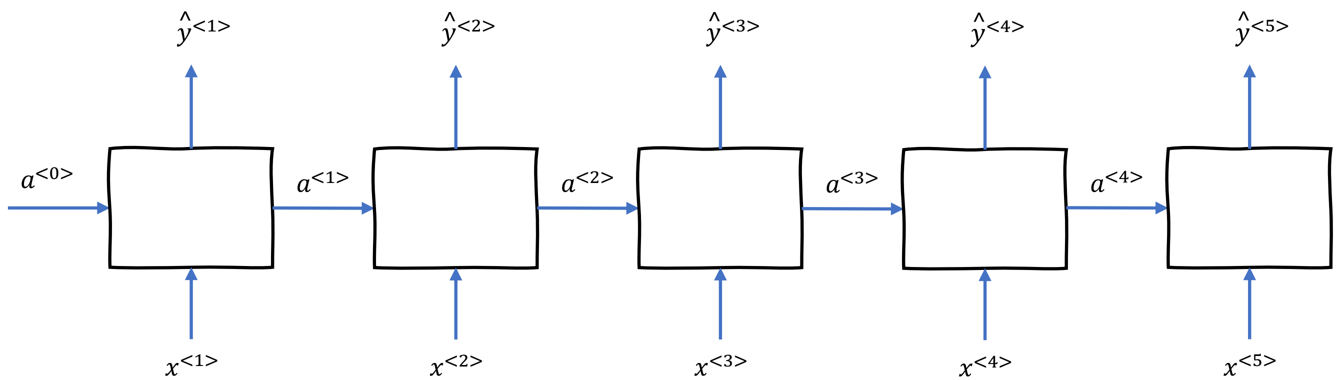
- Reconocimiento de acciones en video



Saltando

Ejemplos de arquitecturas RNN

Escenario: *Muchos a muchos*



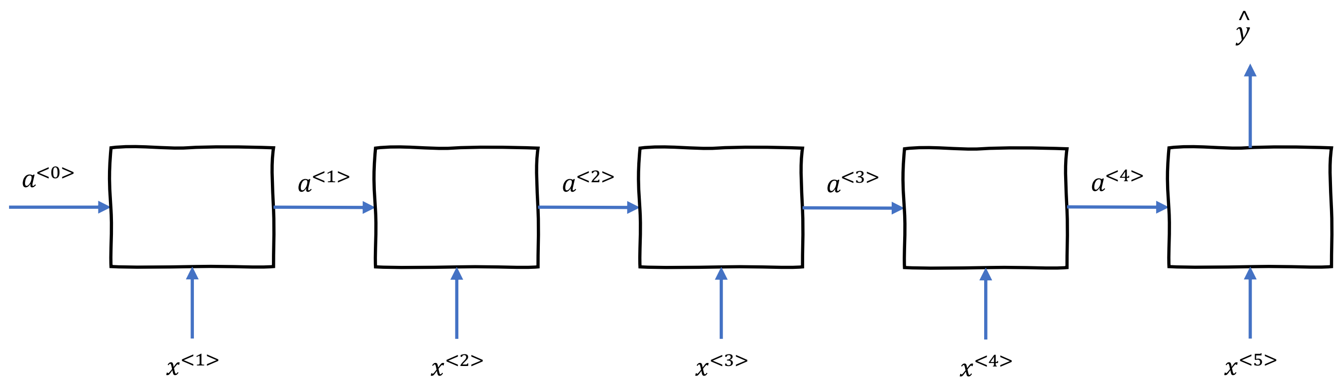
x: *Secuencia*

y: *Secuencia*

Caso: **named-entity**

Ejemplos de arquitecturas RNN

Escenario: *Muchos a uno*



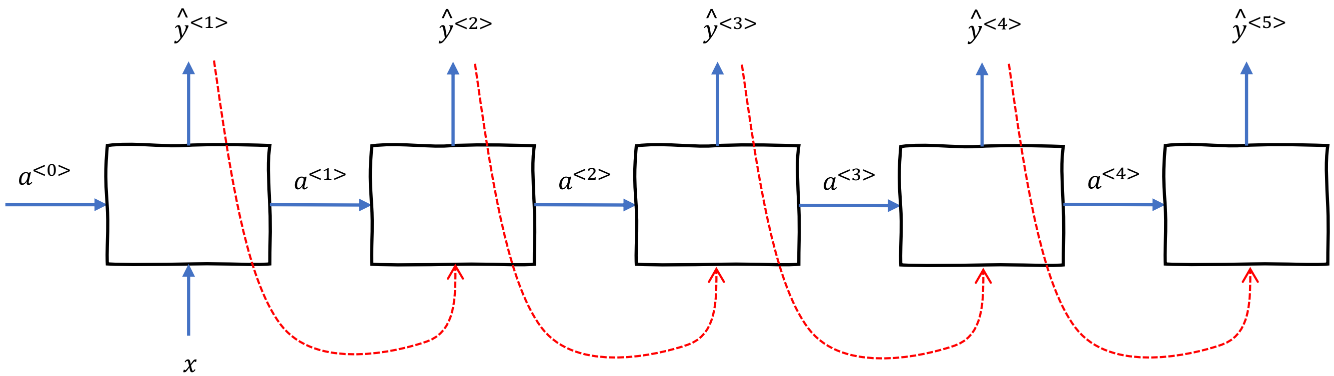
x: $Texto$

y: $\{c_1, c_2, \dots, ck\}$

Ejemplo de aplicación: **análisis de sentimientos**

Ejemplos de arquitecturas RNN

Escenario: *Uno a muchos*



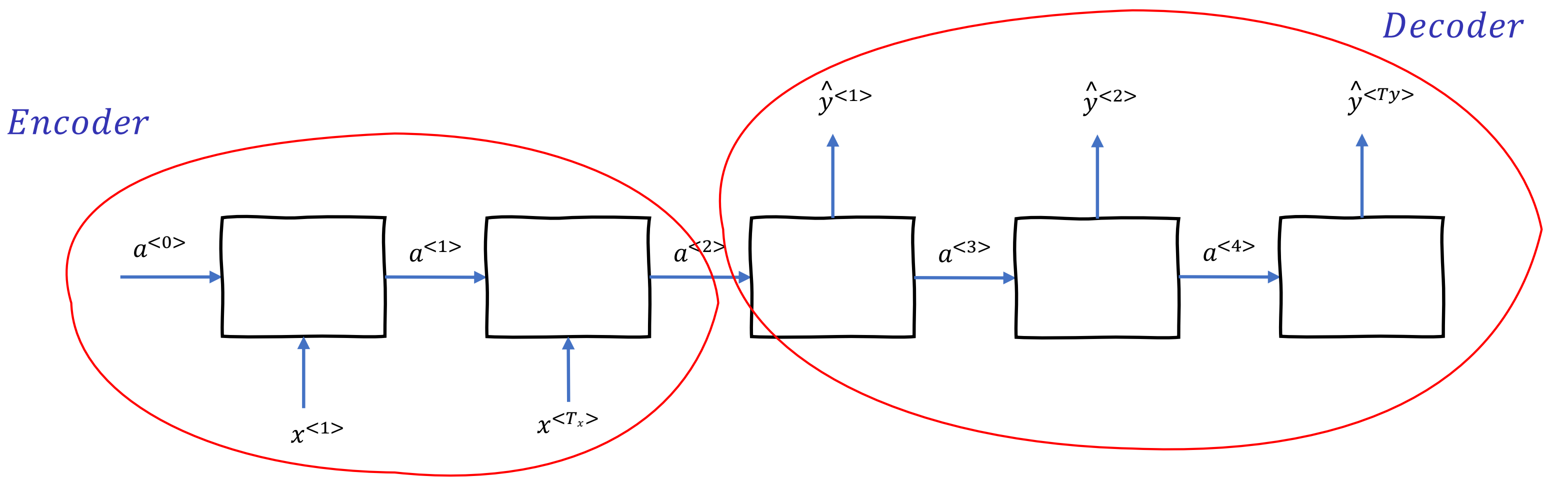
x: $\emptyset$

y: *secuencia de salida*

Ejemplo de aplicación: **generación de música**

Ejemplos de arquitecturas RNN

Escenario: *Muchos a muchos*



x: *secuencia de entrada*

y: *secuencia de salida*

Ejemplo de aplicación: **traducción automática**

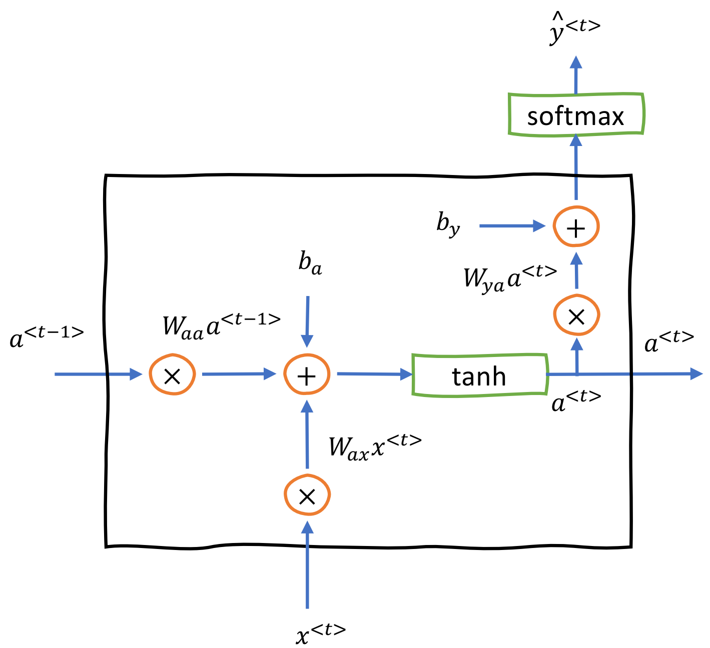
Gated Recurrent Unit (GRU)

$Antes\ de\ abordar\ el\ modelo\ GRU, recordemos\ la\ representación\ de\ una\ celda\ RNN$:

$a^{<0>}$ = [0 … 0]

$a^{<t>} = g_1(w_{aa}a^{<t-1>} + w_{ax}x^{<t>} + b_a)$ $\quad$ $a^{<t>} = g_1(w_a[a^{<t-1>}, x^{<t>}] + b_a)$

$\hat{y}^{<t>} = g_2(w_{ya}a^{<t>} + b_y)$



Unidad GRU (simplificada)

En el modelo GRU, se introduce la variable **c** que significa celda de memoria.

$$c^{<t>} = a^{<t>}$$

$$\tilde{c}^{<t>} = tanh\,(w_c[c^{<t-1>}, x^{<t>}] + b_c)$$

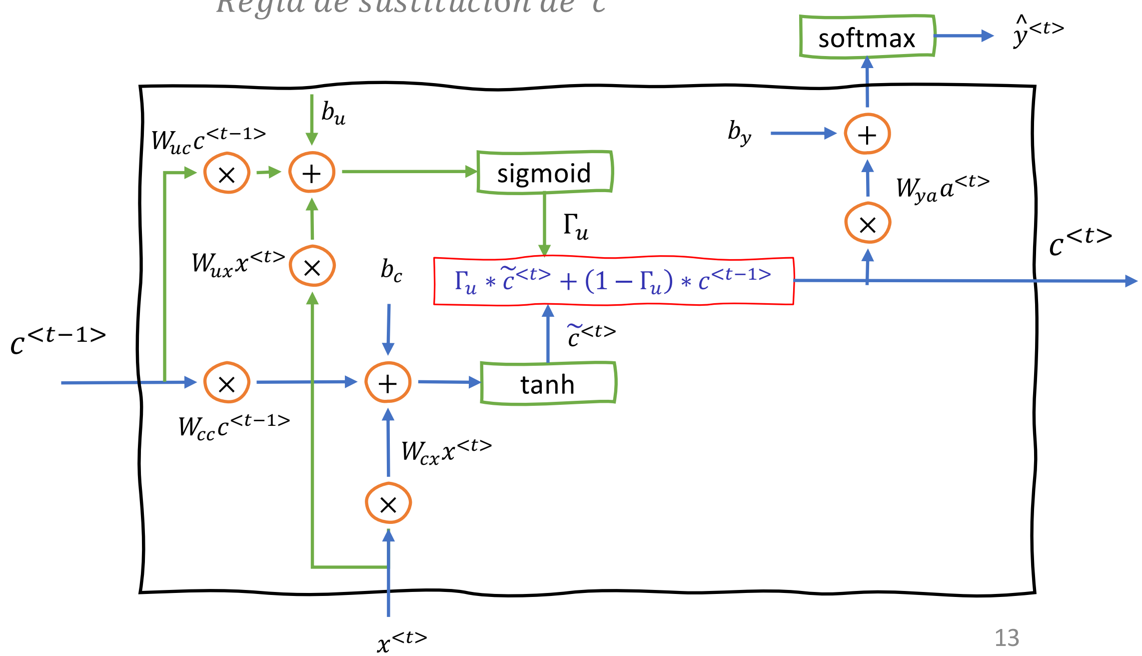
$$\Gamma_u = sigmoid(w_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

Comparando GRU con una celda RNN

Podría sustituir el valor de $c^{<t>}$

Regla de sustitución de $c^{<t>}$



Unidad GRU (completa)

En el modelo GRU, se introduce la variable **c** que significa celda de memoria.

$c^{<t>} = a^{<t>}$

$\Gamma_r = sigmoid(w_r[c^{<t-1>}, x^{<t>}] + b_r)$ — $Relevancia\ de\ c^{<t-1>}$ para calcular $c^{<t>}$

$\tilde{c}^{<t>} = tanh\,(w_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$ — $Podría\ sustituir\ el\ valor\ de\ c^{<t>}$

$\Gamma_u = sigmoid(w_u[c^{<t-1>}, x^{<t>}] + b_u)$ — $Regla\ de\ sustitución\ de\ c^{<t>}$

$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$

$Otra\ versión\ común\ para\ las\ celdas\ de\ una\ red\ neuronal\ recurrente\ es: LSTM\ (Long\ Short\ Term\ Memory)$

GRU y LSTM

GRU

$$\tilde{c}^{<t>} = tanh\,(w_c[\Gamma_r * \; c^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = sigmoid(w_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_r = sigmoid(w_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

$$a^{<t>} = c^{<t>}$$

LSTM

$$\tilde{c}^{<t>} = tanh\,(w_c[a^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = sigmoid(w_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = sigmoid(w_f[a^{<t-1>}, x^{<t>}] + b_f)$$

$$\Gamma_o = sigmoid(w_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

$$a^{<t>} = \Gamma_o * c^{<t>}$$

Celda LSTM

LSTM

$$\tilde{c}^{<t>} = tanh\,(w_c[a^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = sigmoid(w_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = sigmoid(w_f[a^{<t-1>}, x^{<t>}] + b_f)$$

$$\Gamma_o = sigmoid(w_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

$$a^{<t>} = \Gamma_o * c^{<t>}$$