

Redes Neuronales Convolutionales / Recurrentes



UNED



ETS de
Ingeniería
Informática

Dr. Manuel Castillo-Cara

www.manuelcastillo.eu

**Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingeniería Informática
Universidad Nacional de Educación a Distancia (UNED)**

Preliminar



- Improving Deep Learning by Exploiting Synthetic Images © 2024 by Manuel Castillo-Cara is licensed under Attribution-NonCommercial 4.0 International

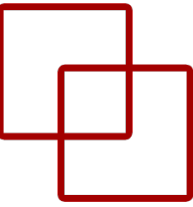


Attribution-NonCommercial 4.0
International (CC BY-NC 4.0)

UNED

ETS de
Ingeniería
Informática

Índice



- Redes Neuronales Convolucionales
 - Operación Convolución
 - Operación Pooling
 - DEMO
 - Hacia la estandarización
- Redes Neuronales Recurrentes
 - Transmisión de conocimiento en RNN
 - Redes Neuronales Recurrentes LSTM



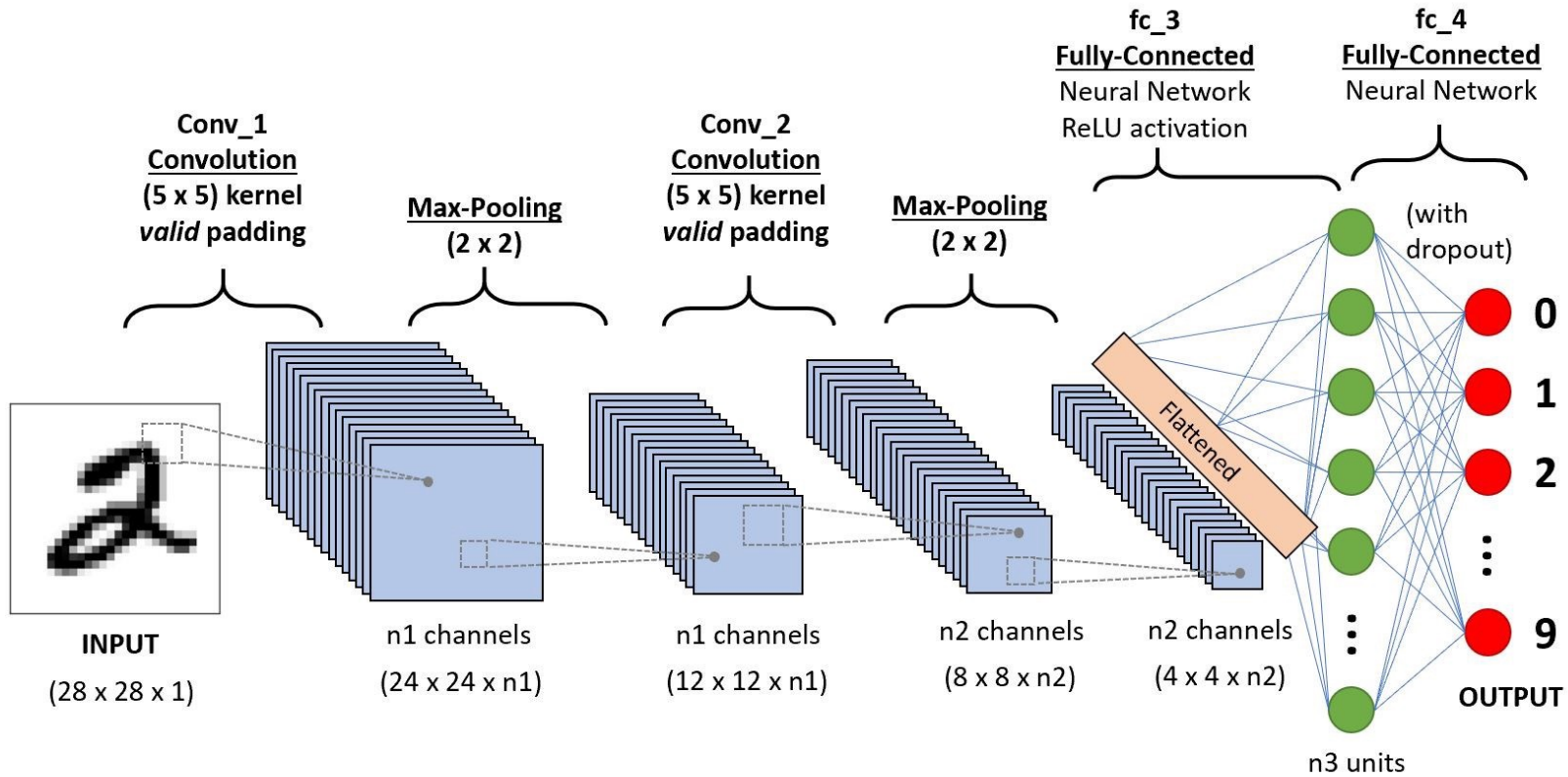
Redes Neuronales Convolucionales



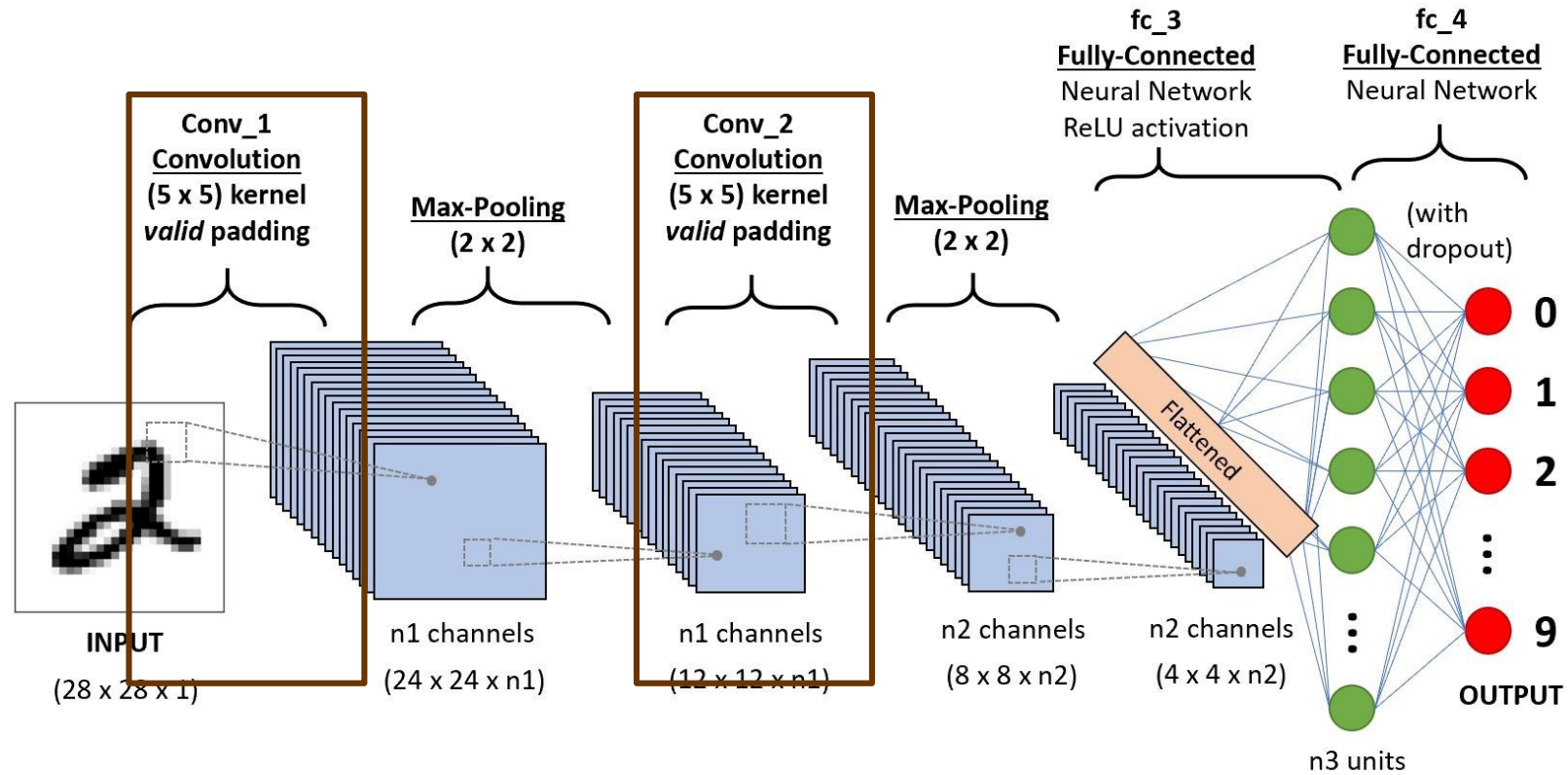
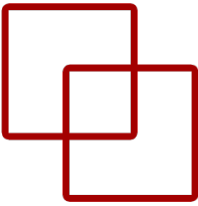
ETS de
Ingeniería
Informática

UNED

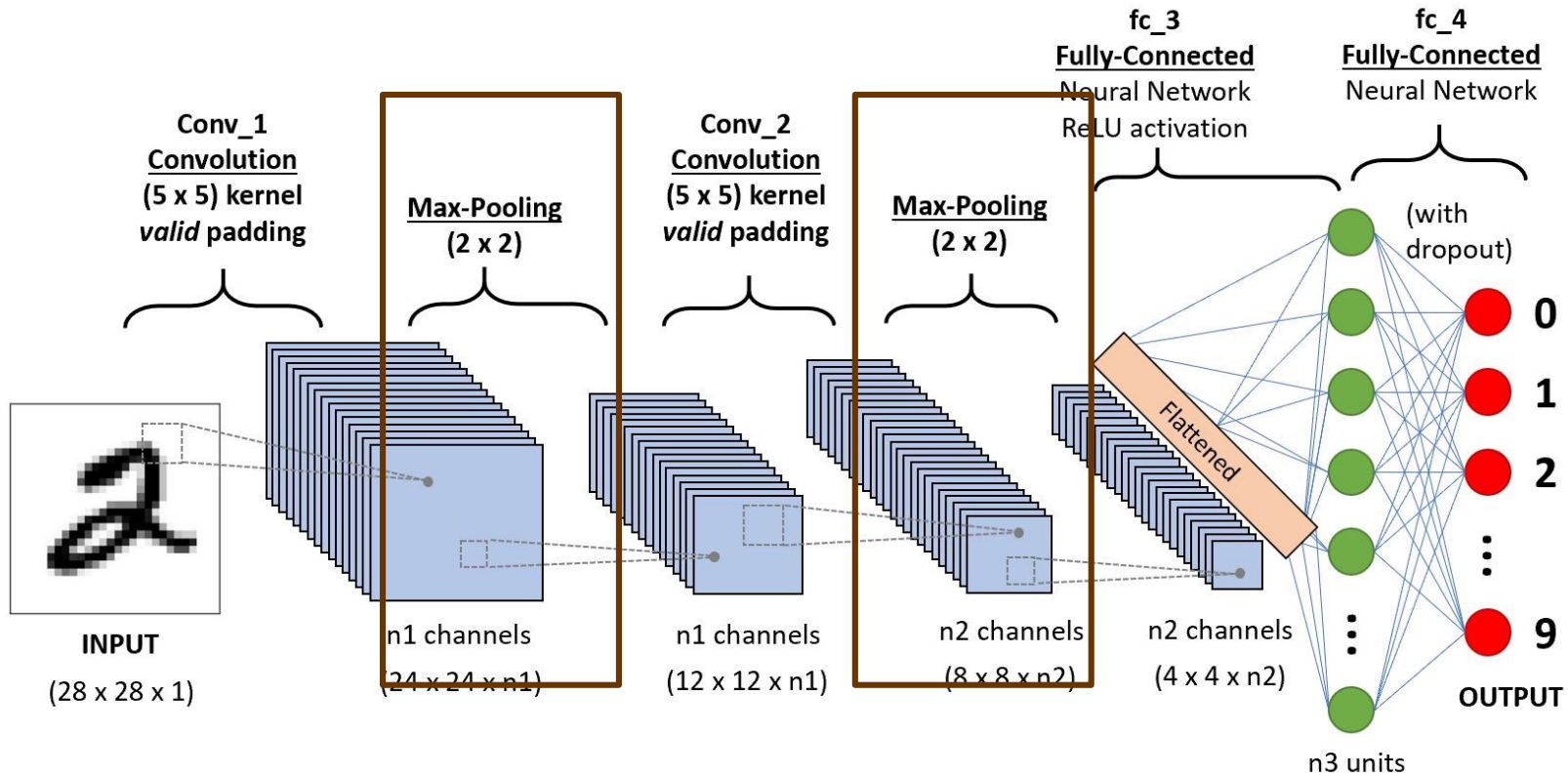
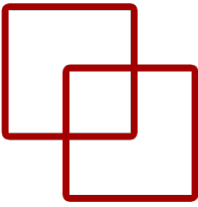
Background



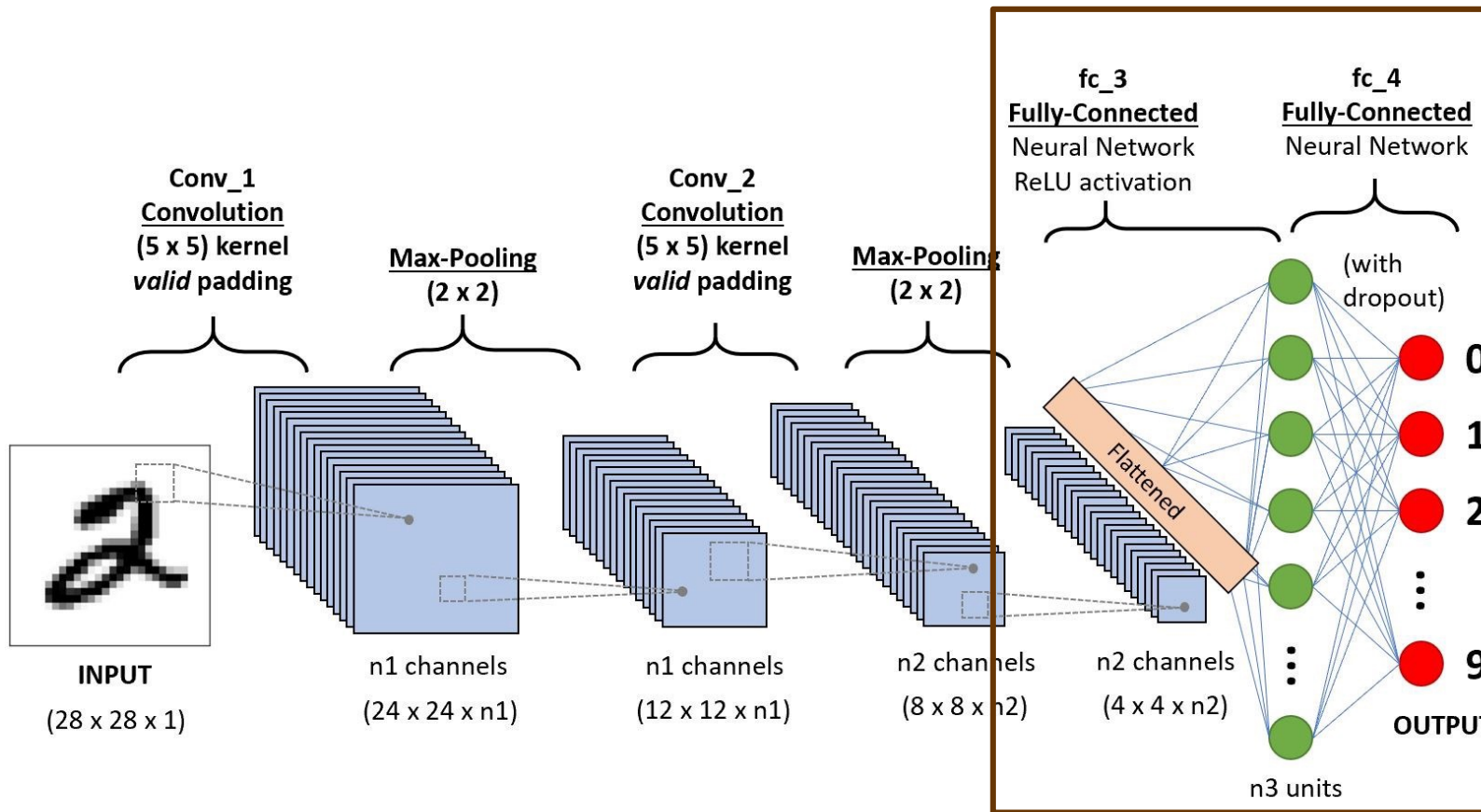
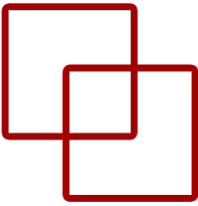
Capas convolucionales



Capas de agrupación (*pooling*)



Capas completamente conectadas





Operación Convolución



ETS de
Ingeniería
Informática

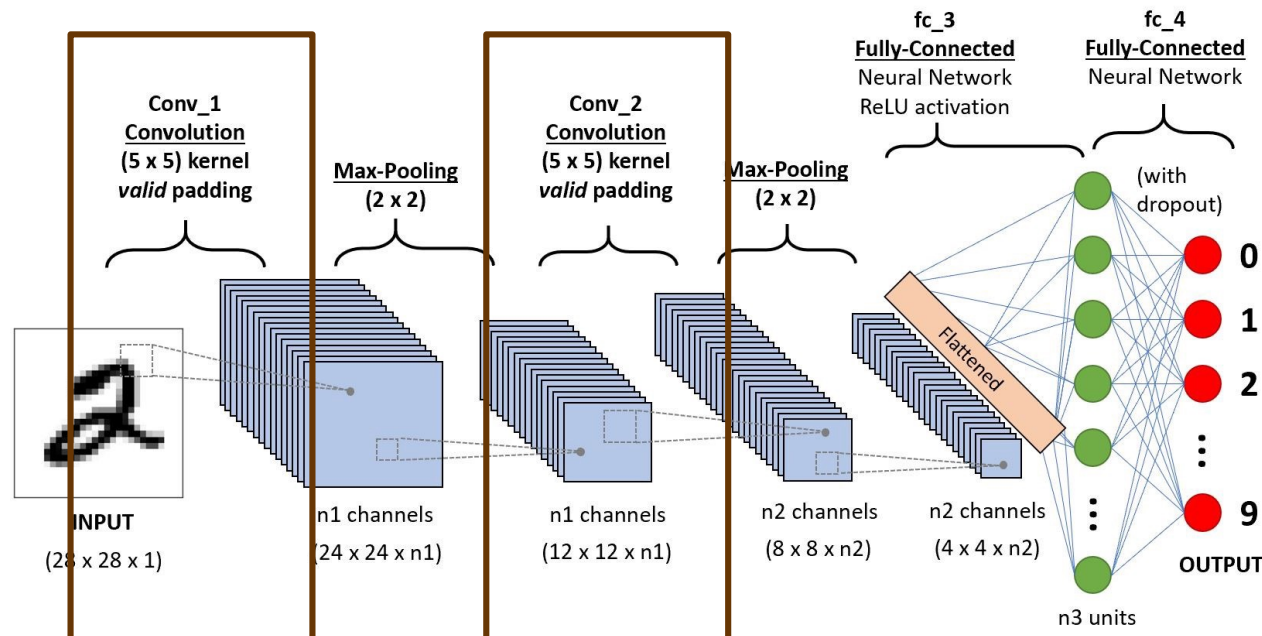


UNED

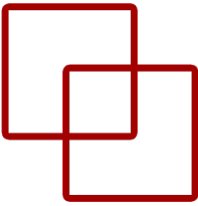
Operación convolución (I)



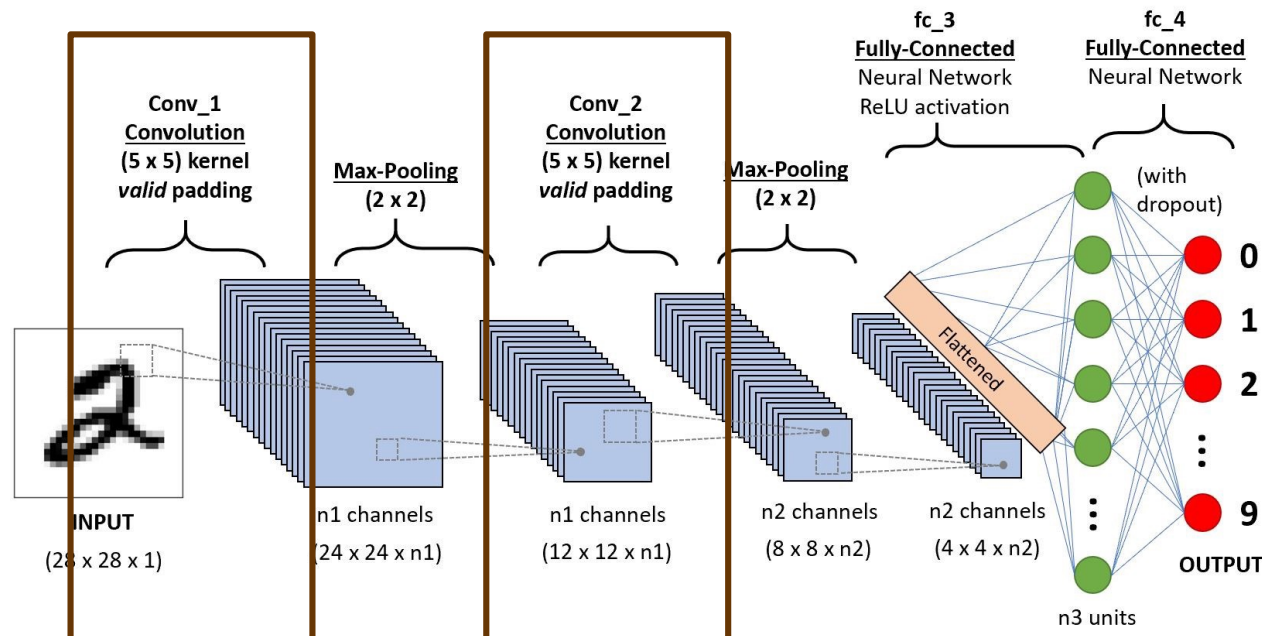
- La primera capa oculta de una CNN se suele corresponder con una **capa convolucional**, la cual puede entenderse como la codificación necesaria para la aplicación de filtros (conocidos como funciones *kernel*)
- Para una mejor comprensión de la operación convolución, pensemos en una capa de entrada que **codifica** una imagen binaria (B/N) y una convolución con la que queremos representar un filtro de **detección de bordes**.



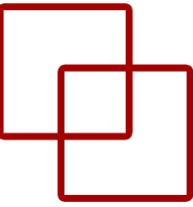
Operación convolución (II)



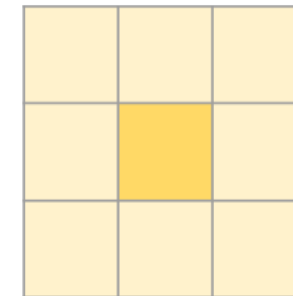
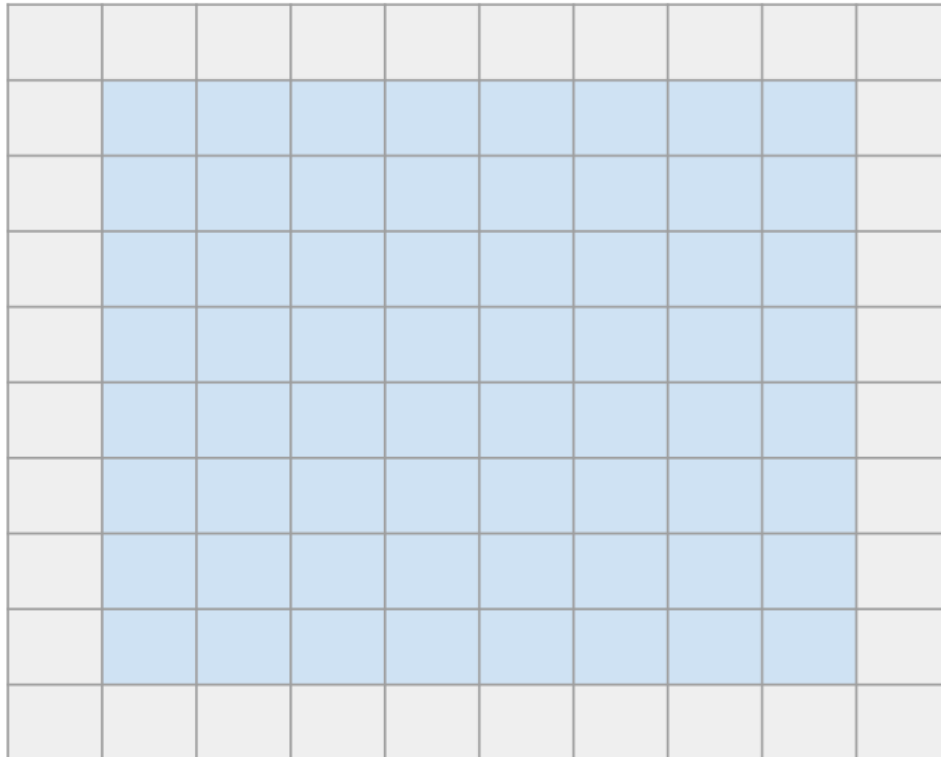
- Sin entrar en más detalles, la búsqueda de bordes se basa en buscar grandes diferencias entre un píxel y los píxeles que le rodean
- El **mismo tipo de operación** la queremos realizar sobre la imagen completa, por lo que aparece una primera cualidad: el uso de **pesos compartidos**
 - Esto significa que los parámetros / condiciones para detectar un borde en un píxel de la imagen deben ser iguales en todas las otras zonas de la imagen



Ejemplo



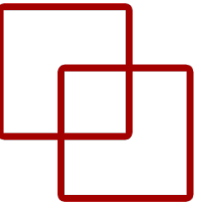
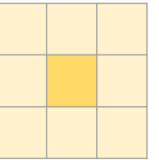
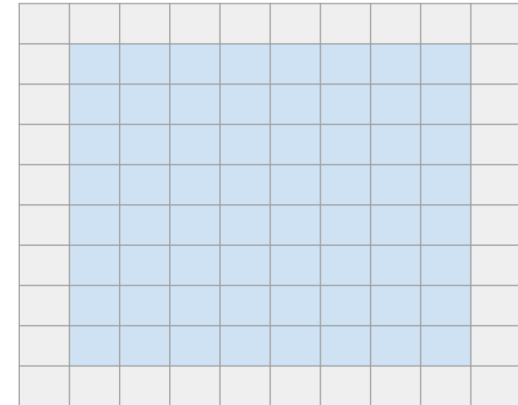
- Asumiendo una imagen de 10x10 píxeles, si la detección de un borde necesita trabajar con los 8 píxeles que le rodean tendríamos lo siguiente



Ejemplo

Operación convolución (I)

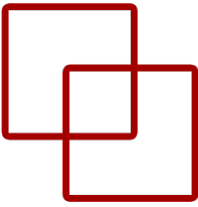
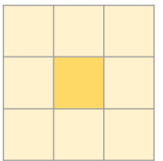
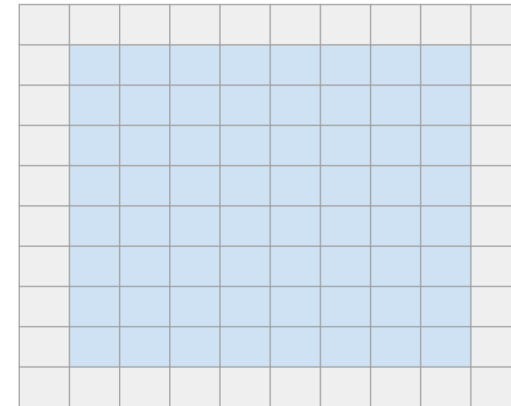
- En la imagen mostramos un filtro que se puede aplicar únicamente sobre las zonas resaltadas



Ejemplo

Operación convolución (I)

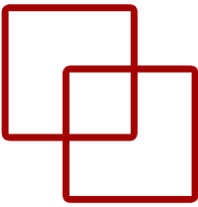
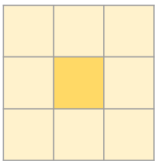
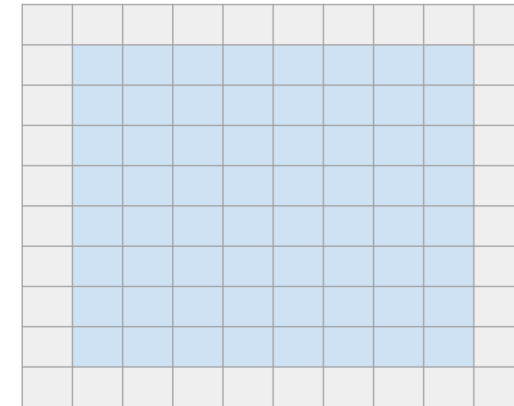
- En la imagen mostramos un filtro que se puede aplicar únicamente sobre las zonas resaltadas
- Trasladando el problema a resolver sobre la topología de una CNN, incluiríamos una capa inicial con 100 neuronas (puede mostrarse visualmente como una matriz de 10x10)



Ejemplo

Operación convolución (I)

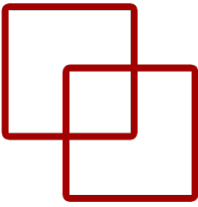
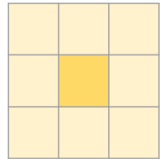
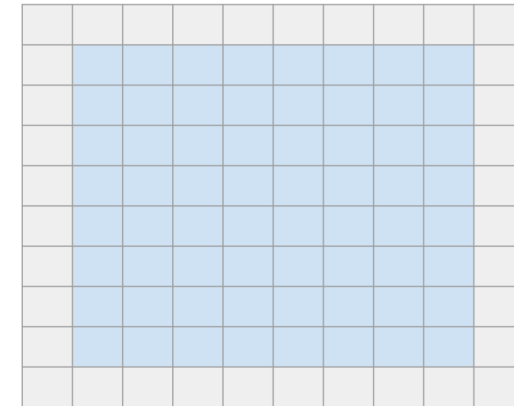
- En la imagen mostramos un filtro que se puede aplicar únicamente sobre las zonas resaltadas
- Trasladando el problema a resolver sobre la topología de una CNN, incluiríamos una capa inicial con 100 neuronas (puede mostrarse visualmente como una matriz de 10x10)
- La primera capa oculta contará con **una neurona por cada resultado** del filtro → tendrá 8x8 neuronas



Ejemplo

Operación convolución (II)

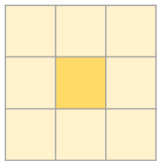
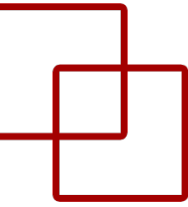
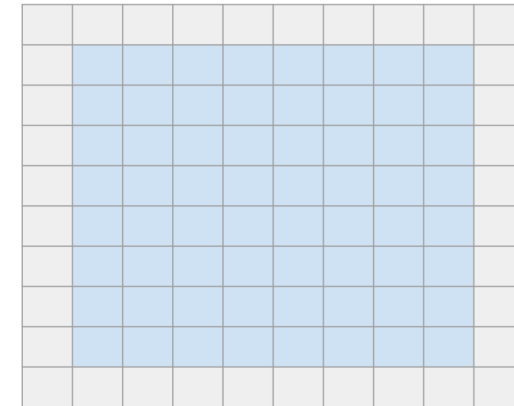
- Cada una de estas 64 neuronas estará conectada con 9 neuronas de la capa inicial, pero los **pesos** utilizados en esta interacción $w_1...w_9$ serán **compartidos** por todas las neuronas de esta capa oculta



Ejemplo

Operación convolución (II)

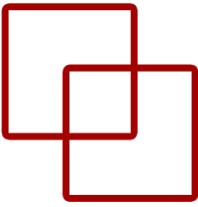
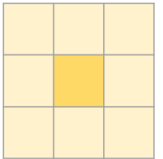
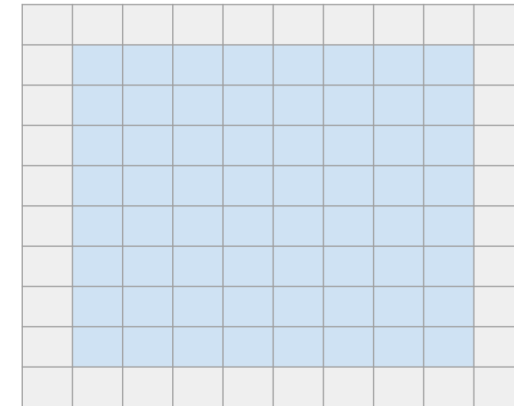
- Cada una de estas 64 neuronas estará conectada con 9 neuronas de la capa inicial, pero los **pesos** utilizados en esta interacción $w_1...w_9$ serán **compartidos** por todas las neuronas de esta capa oculta
- Esto **simplifica** el aprendizaje enormemente, además de dar **coherencia** al filtro a realizar



Ejemplo

Operación convolución (II)

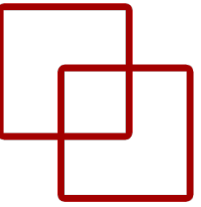
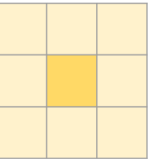
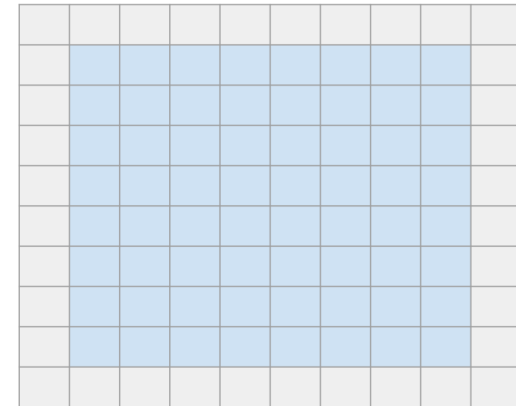
- Cada una de estas 64 neuronas estará conectada con 9 neuronas de la capa inicial, pero los **pesos** utilizados en esta interacción $w_1...w_9$ serán **compartidos** por todas las neuronas de esta capa oculta
- Esto **simplifica** el aprendizaje enormemente, además de dar **coherencia** al filtro a realizar
- Una vez aprendida, la función de activación (ReLU normalmente) nos permitirá saber qué píxel de la imagen se **corresponde o no con un borde**



Ejemplo

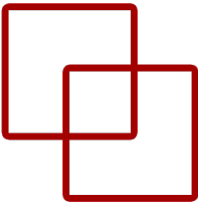
Operación convolución (III)

- En el ejemplo propuesto, se plantean convoluciones que involucren tanto al píxel bajo estudio, como los **8 píxeles a su alrededor (3x3)**

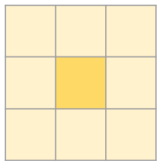
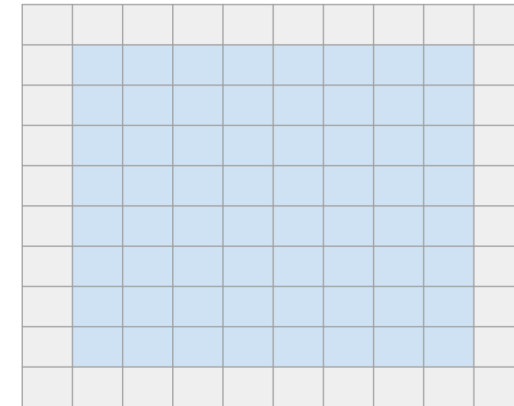


Ejemplo

Operación convolución (III)



- En el ejemplo propuesto, se plantean convoluciones que involucren tanto al píxel bajo estudio, como los **8 píxeles a su alrededor** (3x3)
- Además, se asume que esta operación se realiza uno a uno, pero sólo sobre los píxeles de la imagen que es posible (salvo las filas / columnas 0 y 9), obteniendo como resultado una **capa de tamaño inferior** (10x10 → 8x8)
 - 2 parámetros definen esta configuración: ***padding*** y ***stride***



Ejemplo

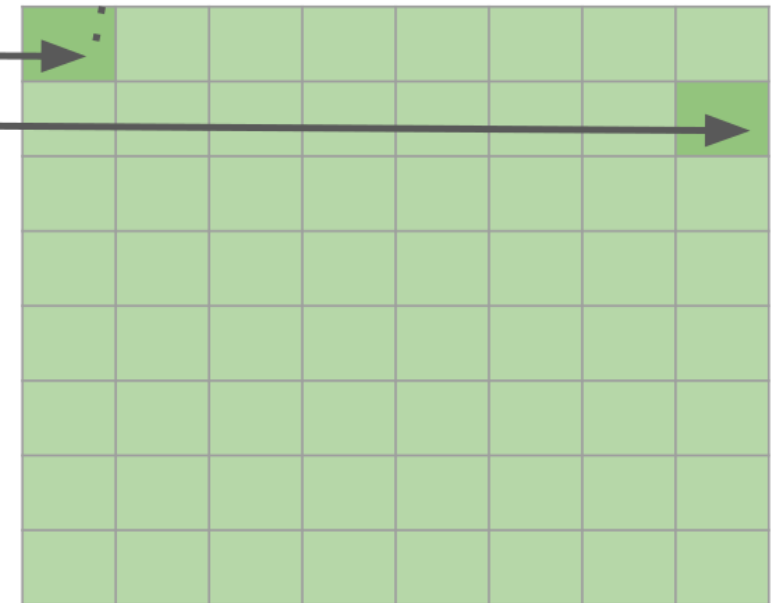
Operación convolución (IV)



1	3	2							
6	6	4							
5	7	6							

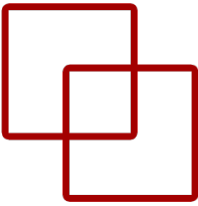
w_0	w_1	w_2
w_3	w_4	w_5
w_7	w_8	w_9

$$\begin{aligned} &= w_0 * 1 + w_1 * 3 + w_2 * 2 \\ &+ w_3 * 6 + w_4 * 6 + w_5 * 4 \\ &+ w_7 * 5 + w_8 * 7 + w_9 * 6 \end{aligned}$$

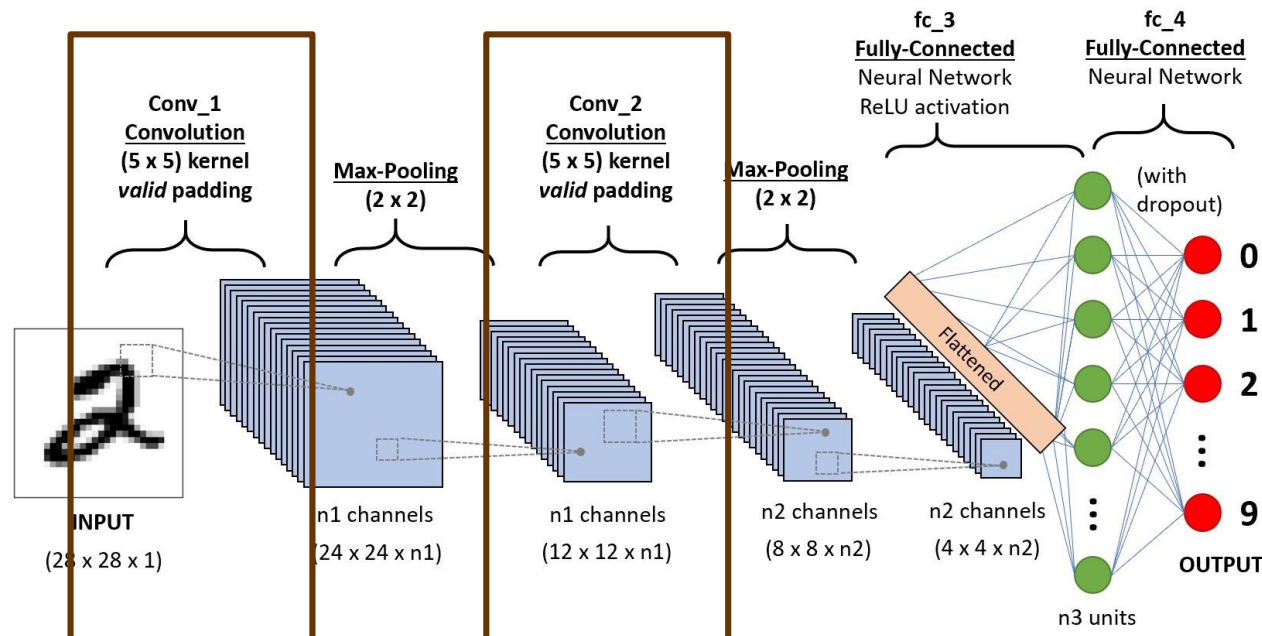


Padding

Definición

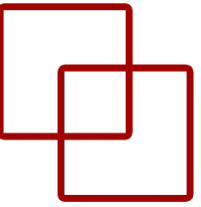


- El parámetro ***padding*** (opcional) define una serie de neuronas sintéticas usadas a modo de relleno, de forma que se permite realizar convoluciones sin que la capa resultado sea de un tamaño menor a la de entrada
- Estas neuronas deben tomar un **valor** para la realización de las operaciones, siendo cero el valor más usado.
 - En este caso nos referimos a este parámetro como *zero-padding N* (N = número de neuronas adicionales)

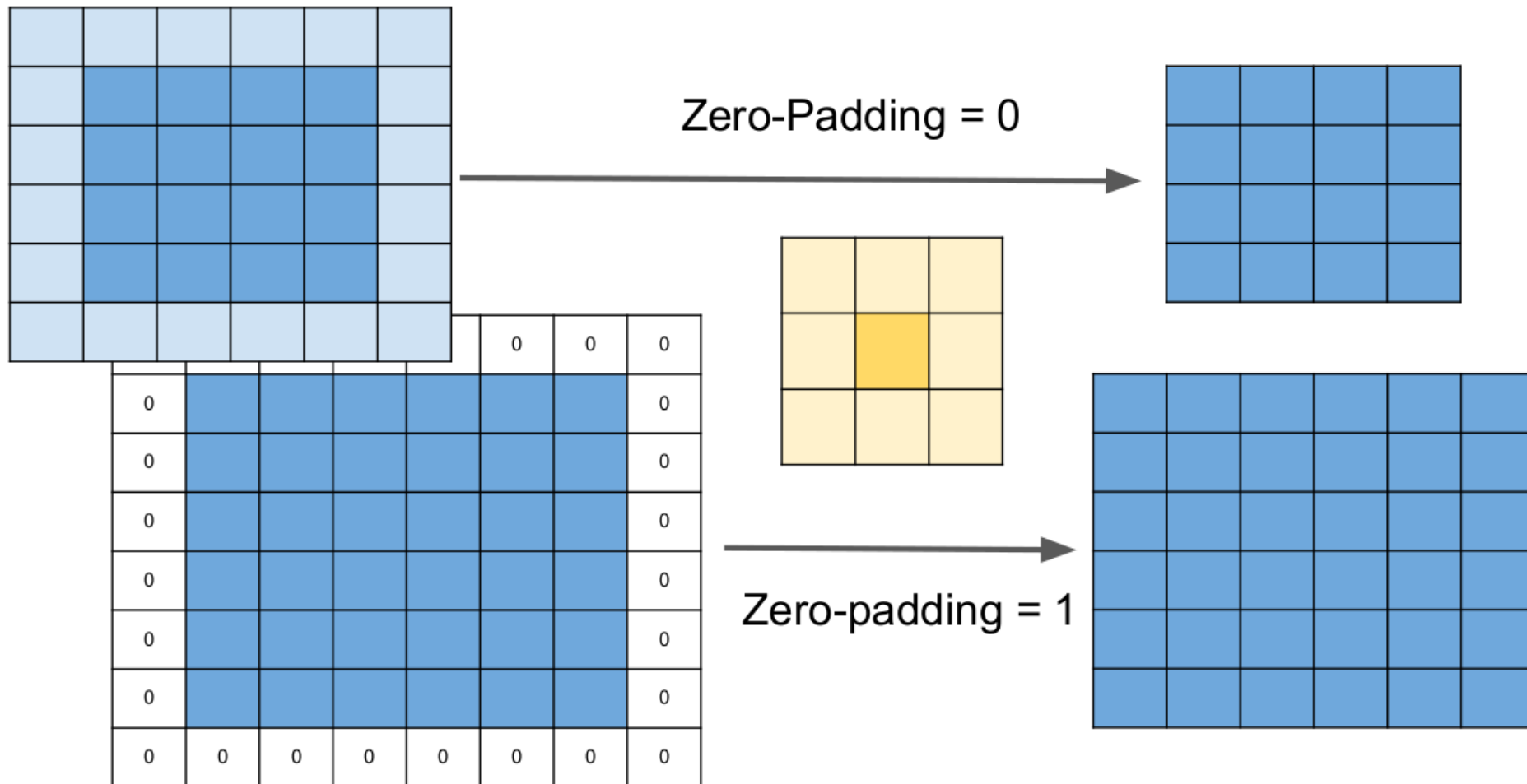


Padding

Ejemplo



- Veamos un ejemplo con imágenes de 6x6 y un filtro de 3x3





Operación *Pooling*

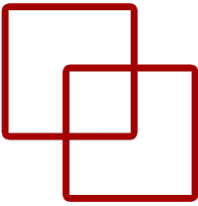


ETS de
Ingeniería
Informática

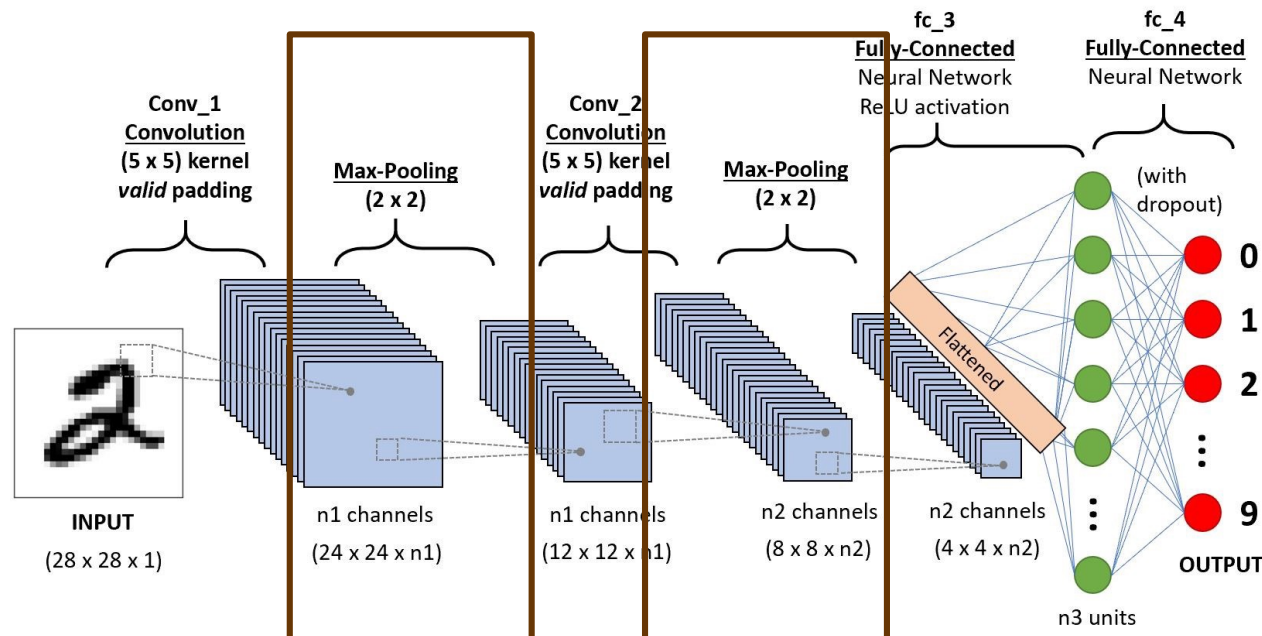


UNED

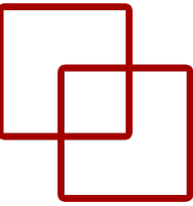
Definición



- La operación *pooling* tiene por objetivo **reducir la dimensionalidad** de las capas generadas tras la aplicación de operaciones de convolución
- Recordando que las capas disponen de tres dimensiones (*ancho x alto x profundidad*), la reducción se aplicaría únicamente sobre las **dimensiones de anchura y altura**, sin afectar a la profundidad de estas capas

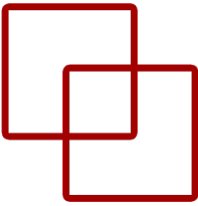


Ejemplo



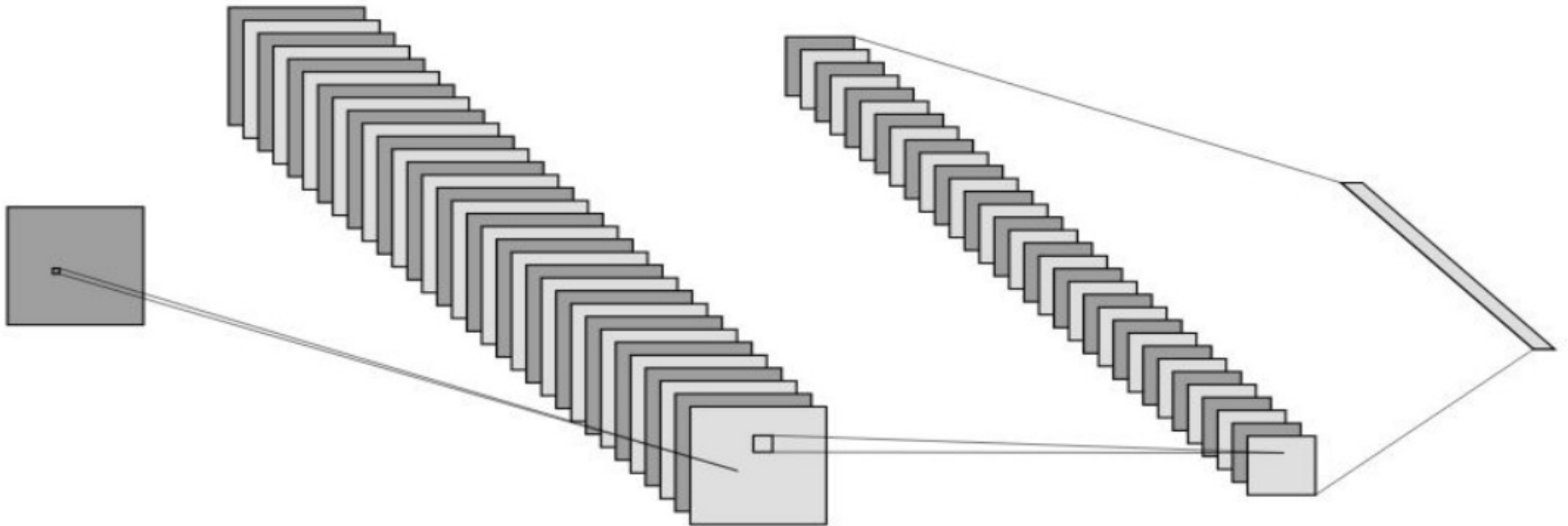
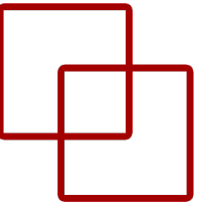
- Sobre un ejemplo simple, asumamos que trabajamos con imágenes **640x480** (B/N) y que la primera convolución aplica **32 filtros de 3x3** (*stride 1, zero-padding 1*)
 - El resultado será una capa intermedia de **640x480x32**

Ejemplo

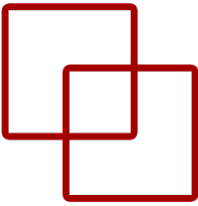


- Sobre un ejemplo simple, asumamos que trabajamos con imágenes **640x480** (B/N) y que la primera convolución aplica **32 filtros de 3x3** (*stride 1, zero-padding 1*)
 - El resultado será una capa intermedia de **640x480x32**
- Una operación *pooling* **reduciendo la dimensionalidad** a un 25% de la original obtendría una nueva capa intermedia de **320x240x32**

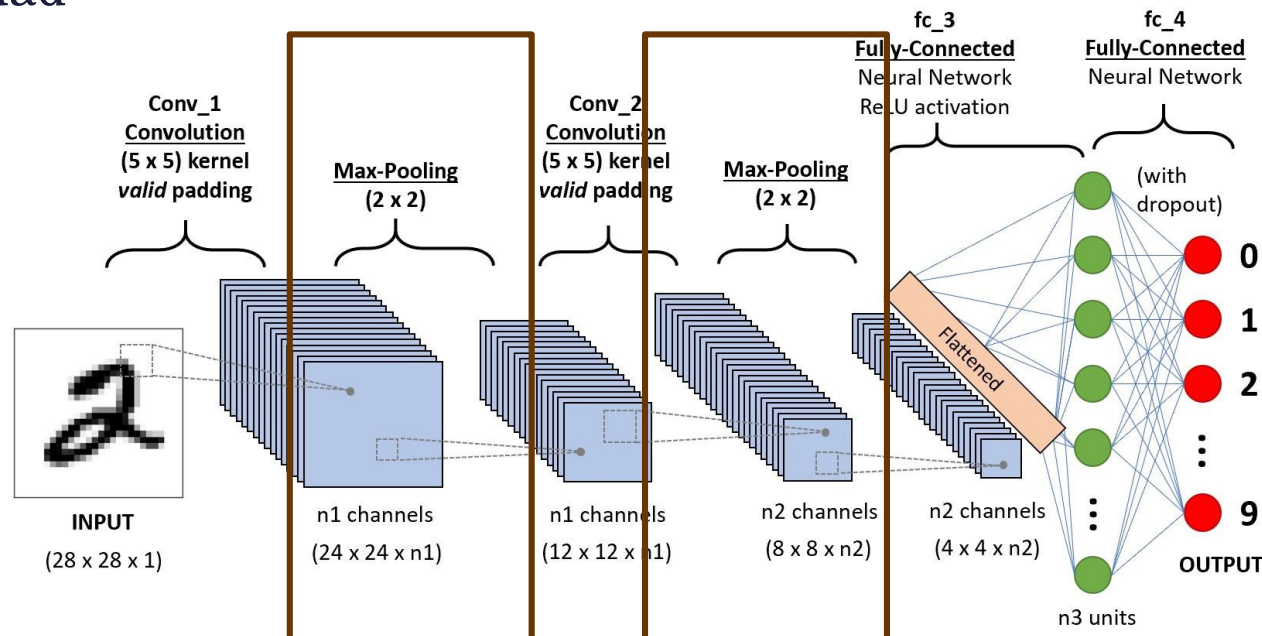
Ejemplo



Max-pooling

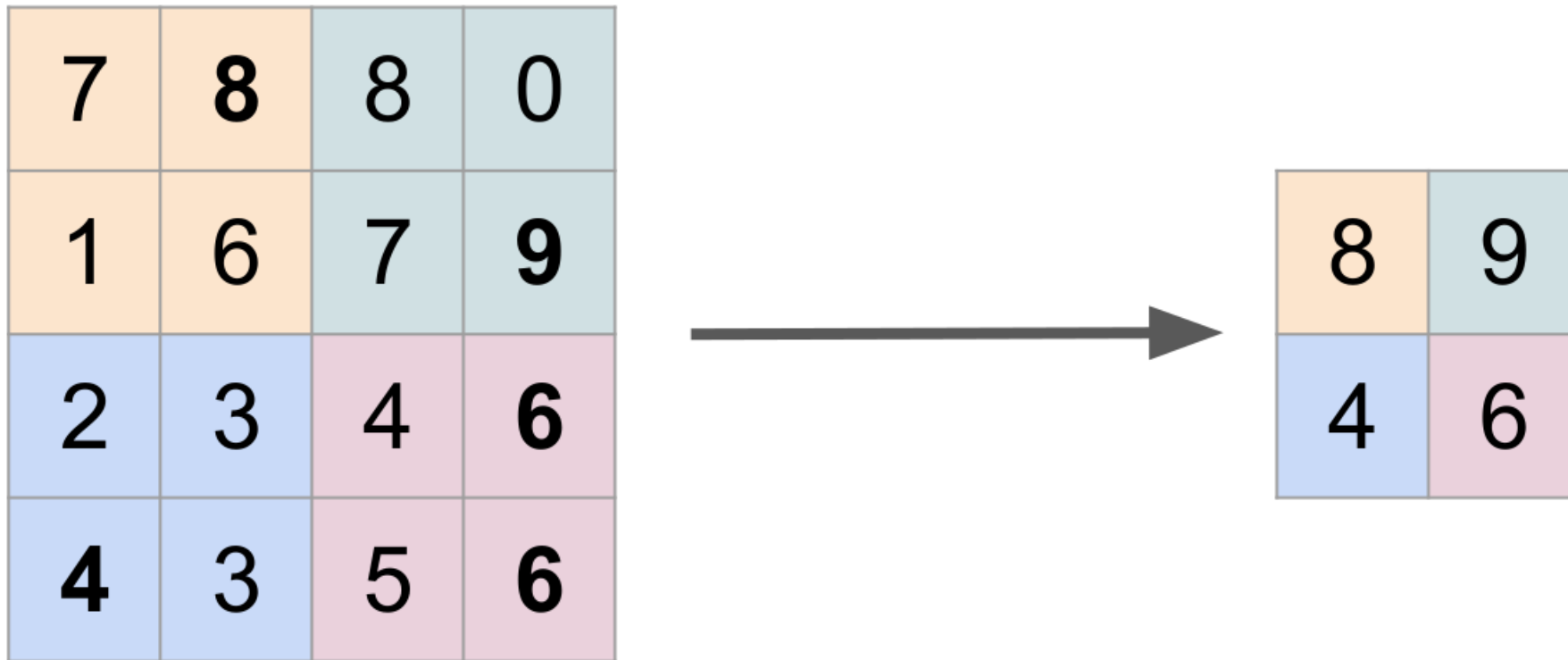
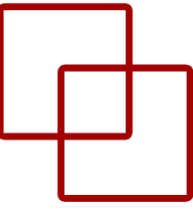


- La operación de *pooling* más común es *max-pooling*, donde cada operación computa el **máximo** para una serie de neuronas **conectadas espacialmente**
- Su aplicación se realiza a través de un filtro de tamaño $N \times M$, normalmente $M=N$ (se define como extensión espacial).
 - También se usa el **parámetro *stride*** que determinará el factor de reducción de dimensionalidad

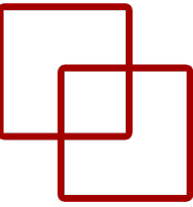


Max-pooling

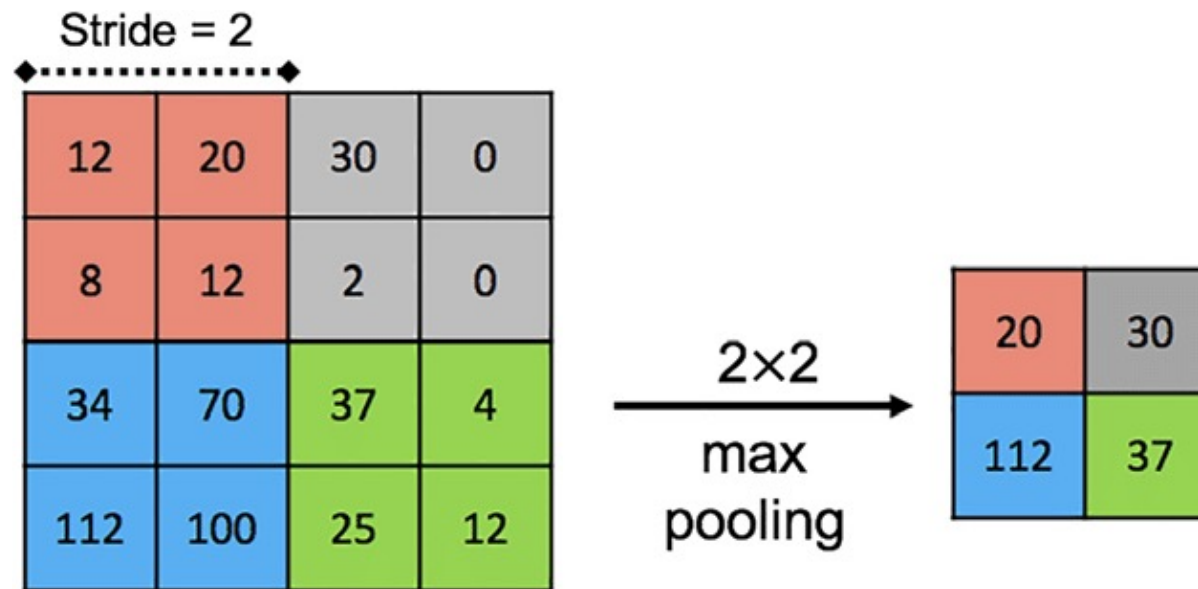
Ejemplo de max-pooling 2x2 con stride=2



Consideraciones



- A nivel práctico, el uso de filtros mayores a 3x3, o de un valor de *stride* superior a 2, suele generar malos resultados → desaconsejamos su uso
- Como **alternativas** al *max-pooling*, podemos reemplazar la función máximo por otras funciones como la media.
 - Sin embargo, la experiencia ha mostrado el mayor poder de la función máximo





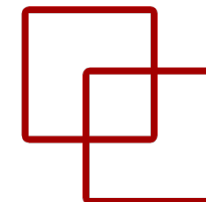
DEMO



ETS de
Ingeniería
Informática

UNED

ConvNetJS



Network Visualization

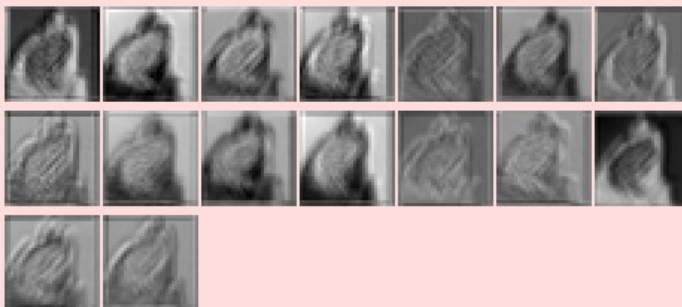
input (32x32x3)
max activation: 0.47254, min: -0.4451
max gradient: 0.02125, min: -0.01962

Activations:

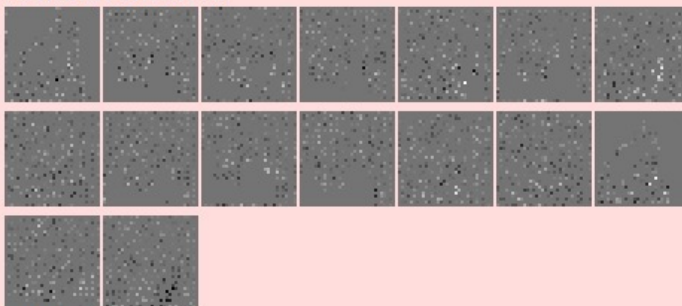


conv (32x32x16)
filter size 5x5x3, stride 1
max activation: 0.89, min: -0.52082
max gradient: 0.02142, min: -0.01816
parameters: $16 \times 5 \times 5 \times 3 + 16 = 1216$

Activations:



Activation Gradients:



Weights:



Weight Gradients:



ConvNetJS CIFAR-10 demo

Description

This demo trains a Convolutional Neural Network on the [CIFAR-10 dataset](#) in your browser, with nothing but Javascript. The state of the art on this dataset is about 90% accuracy and human performance is at about 94% (not perfect as the dataset can be a bit ambiguous). I used [this python script](#) to parse the [original files](#) (python version) into batches of images that can be easily loaded into page DOM with img tags.

This dataset is more difficult and it takes longer to train a network. Data augmentation includes random flipping and random image shifts by up to 2px horizontally and vertically.

By default, in this demo we're using Adadelta which is one of per-parameter adaptive step size methods, so we don't have to worry about changing learning rates or momentum over time. However, I still included the text fields for changing these if you'd like to play around with SGD+Momentum trainer.

Report questions/bugs/suggestions to [@karpathy](#).

Training Stats

Enlace: <https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html>



**Hacia la
estandarización**

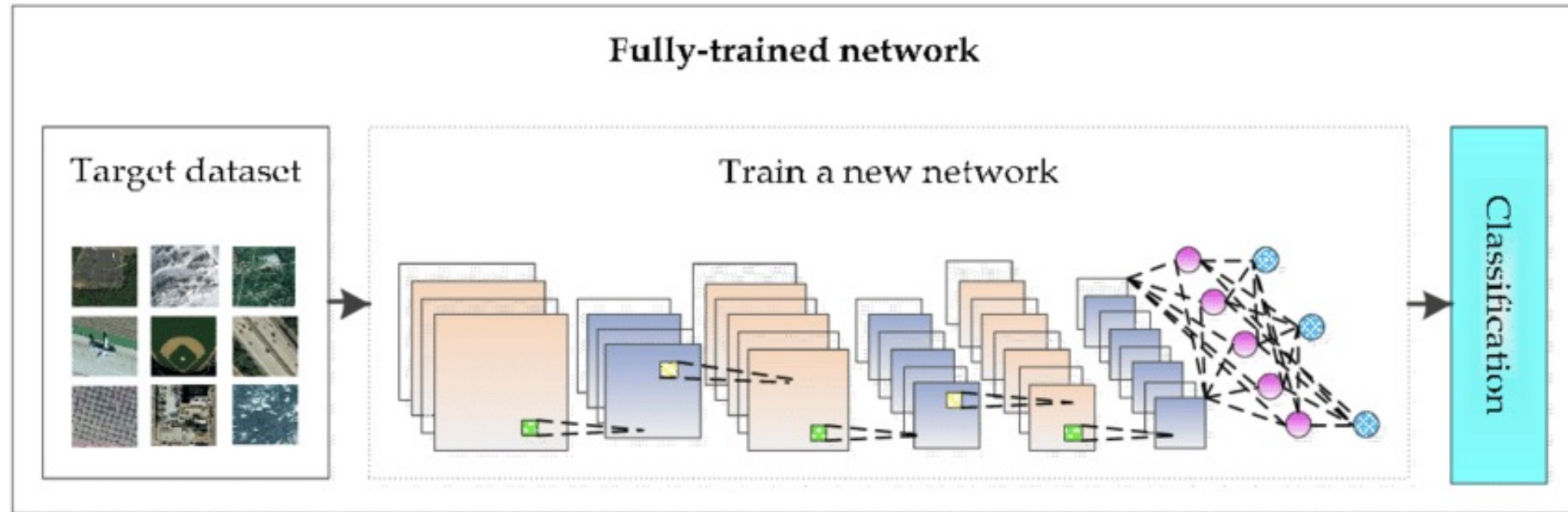
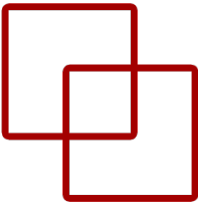


**ETS de
Ingeniería
Informática**

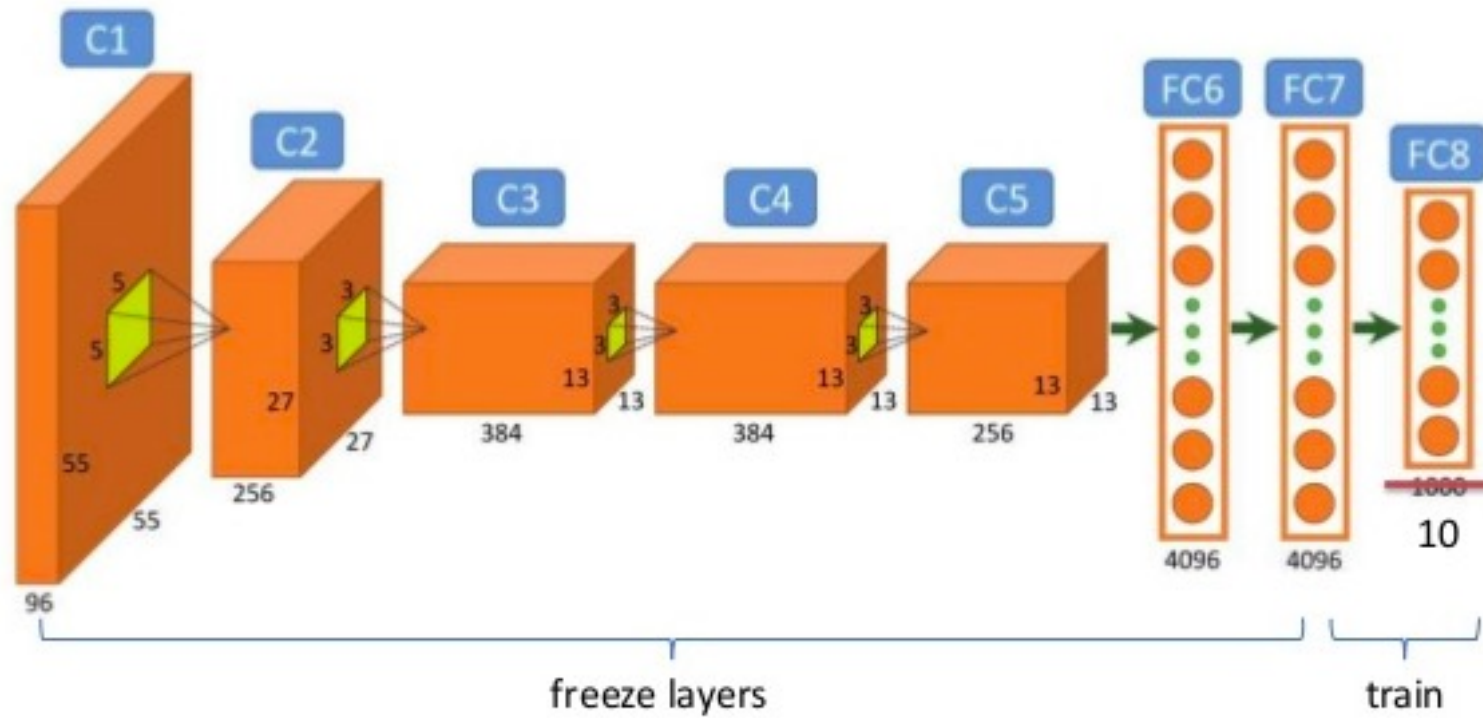
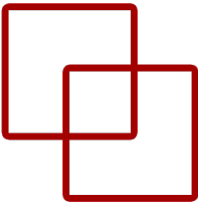


UNED

Transfer Learning



Finetuning





Redes Neuronales Recurrentes

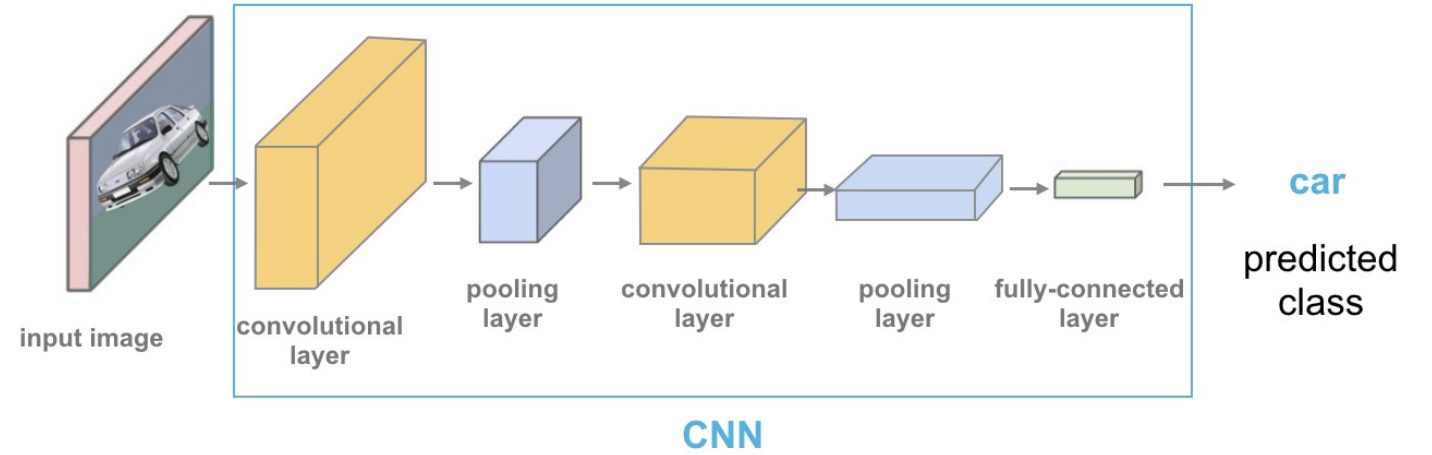
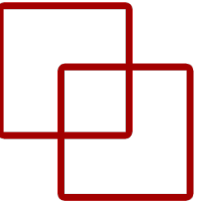


ETS de
Ingeniería
Informática

UNED

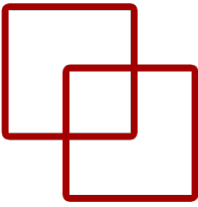
Background

CNN vs. RNN

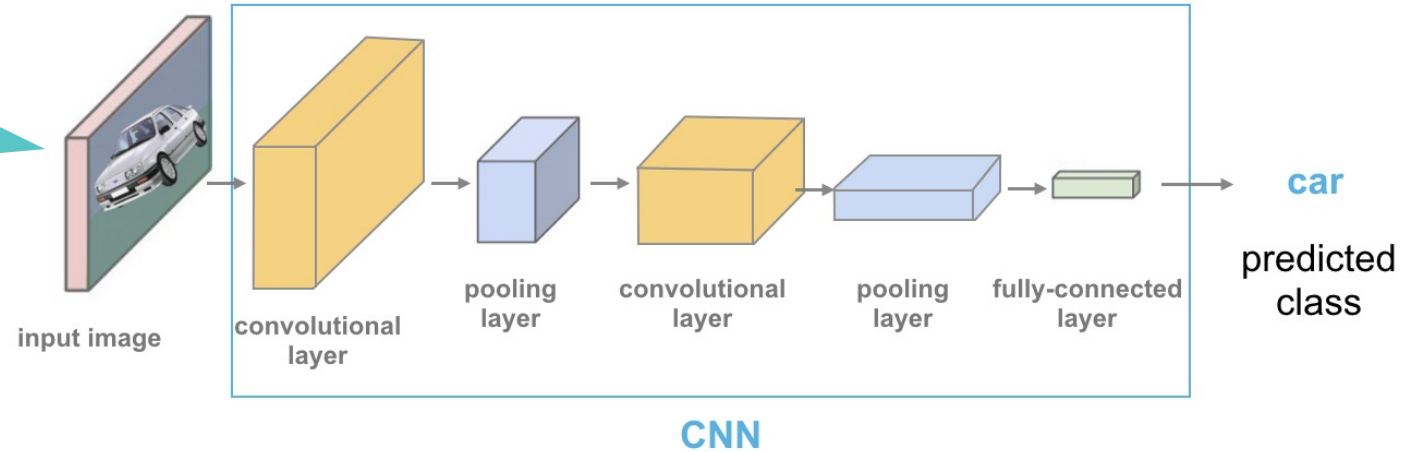


Background

CNN vs. RNN

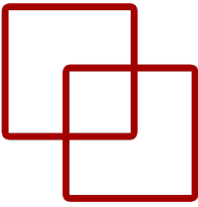


Dato (imagen, secuecnia, palabra, etc) a clasificar
¿Qué pasa si se introduce una secuencia?

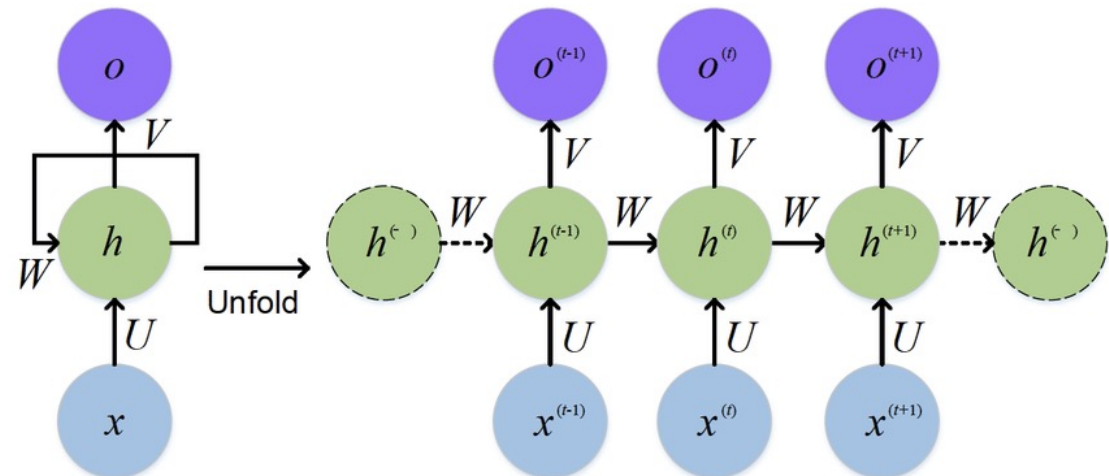
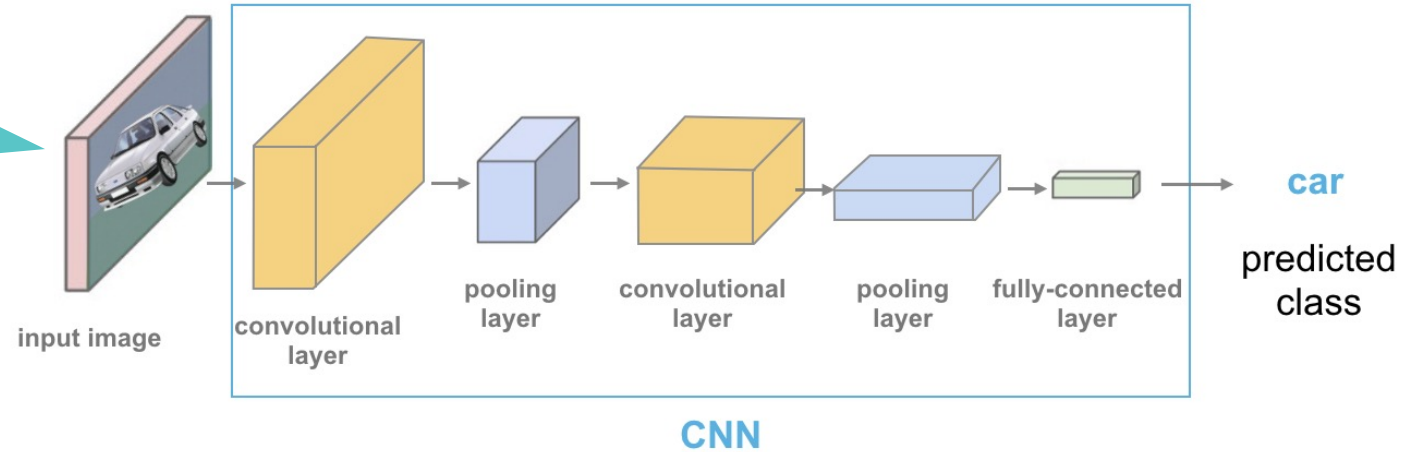


Background

CNN vs. RNN

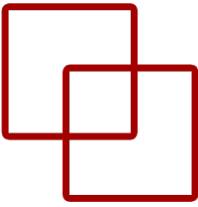


Dato (imagen, secuecnia, palabra, etc) a clasificar
¿Qué pasa si se introduce una secuencia?

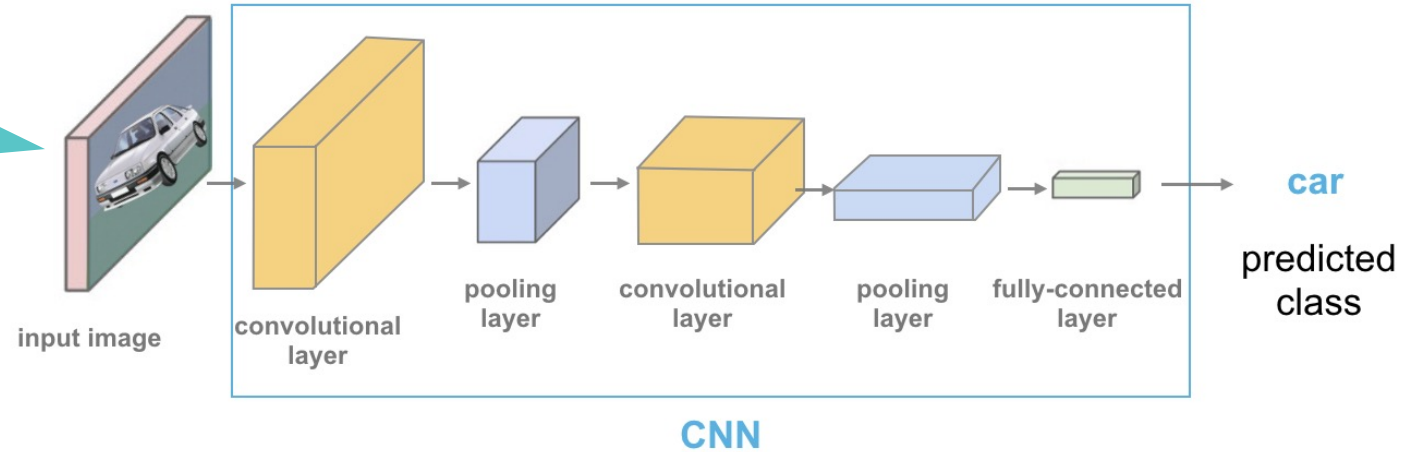


Background

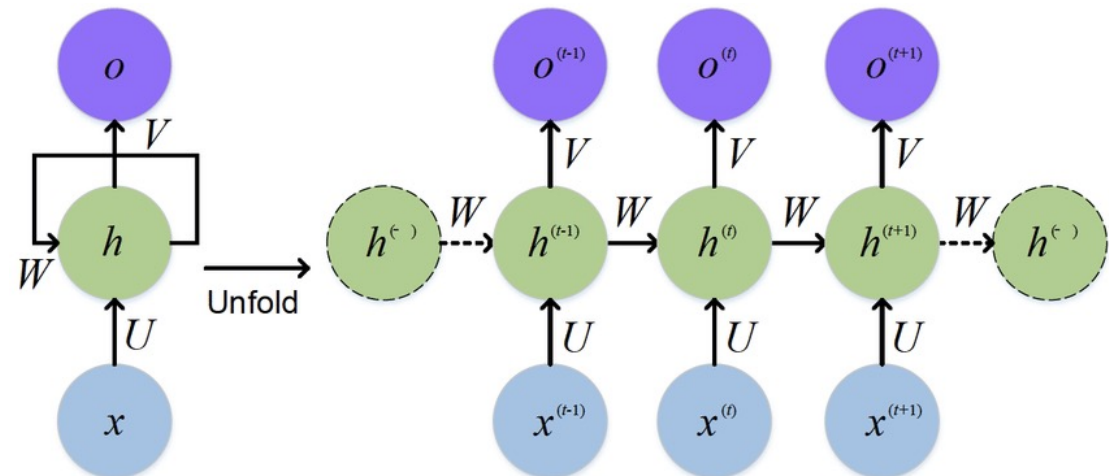
CNN vs. RNN



Dato (imagen, secuencia, palabra, etc) a clasificar
¿Qué pasa si se introduce una secuencia?

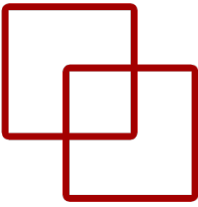


Secuencia (conversación, texto, vídeo) de datos con orden
Los datos están correlacionados dependiendo del texto anterior

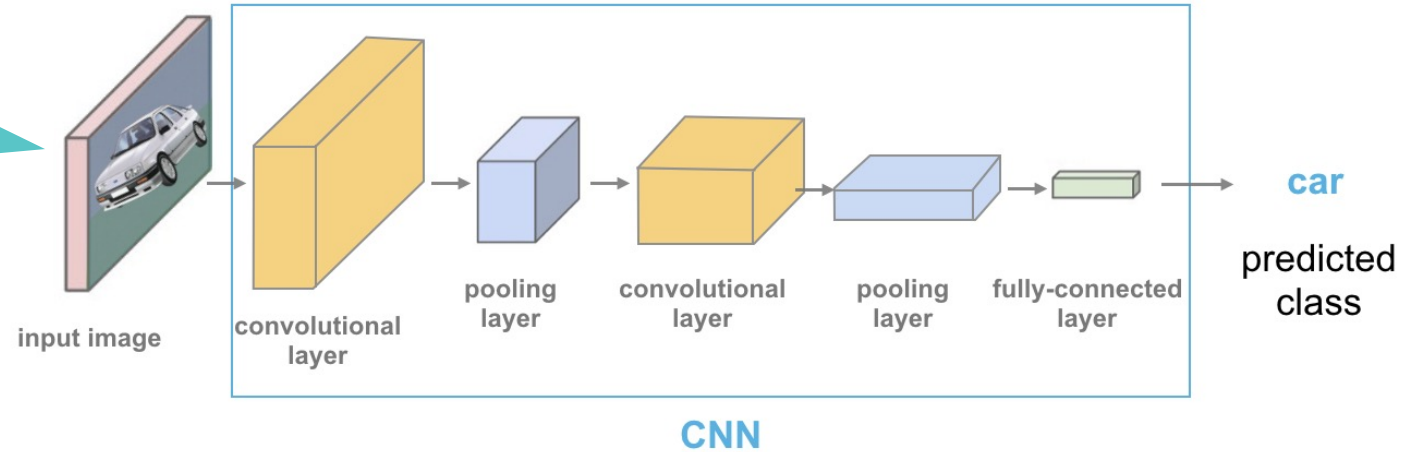


Background

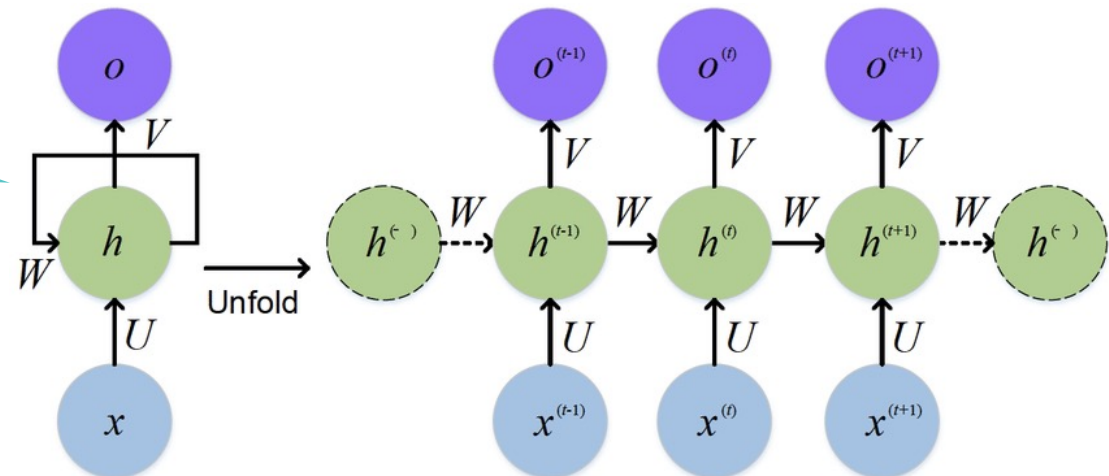
CNN vs. RNN



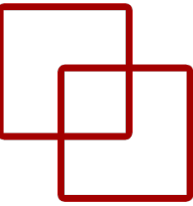
Dato (imagen, secuencia, palabra, etc) a clasificar
¿Qué pasa si se introduce una secuencia?



Para ello utiliza la activación de la **iteración anterior** para dar nuevos resultados, i.e., tiene memoria



Generación de texto



at first:

tyntd-iafhatawiaoighrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrqd t o idoe ns,smtt h ne etie h,hregtrs nigtkie,aoaenns lng

↓ train more

"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuw y fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."

↓ train more

Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.

↓ train more

"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftended him.
Pierre aking his soul came to the packs and drove up his father-in-law women.

Fuente: <http://cs231n.stanford.edu/index.html>



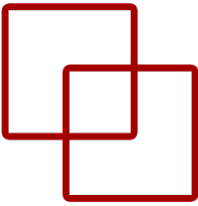
Transmisión de conocimiento en RNN



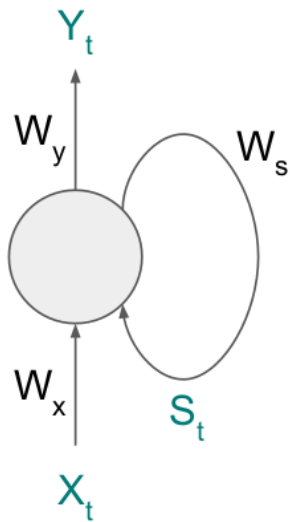
ETS de
Ingeniería
Informática

UNED

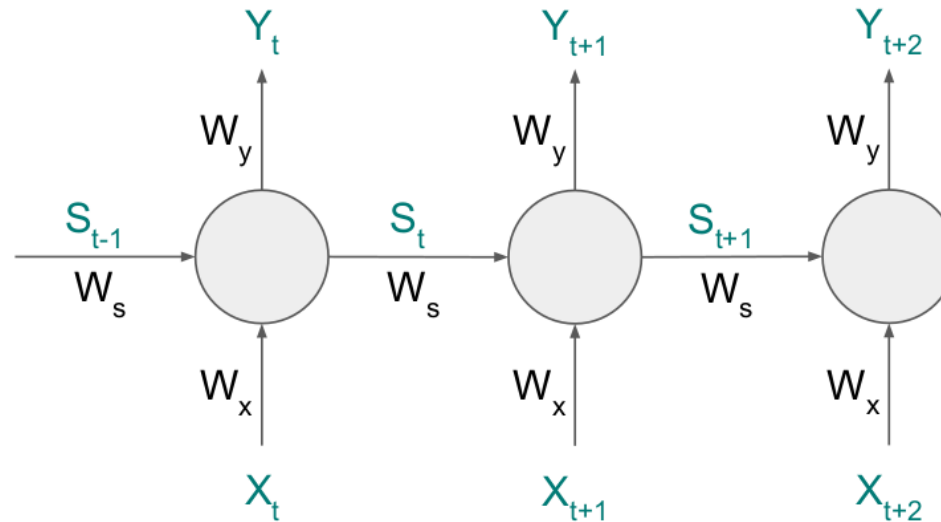
Funcionamiento (I)



- Un mismo modelo de RNN se pueden representar de forma **plegada o desplegada**.
 - Desplegado “en tiempo”.

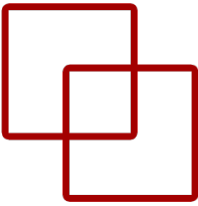


Modelo plegado

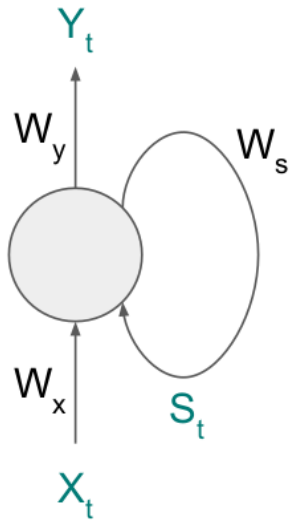


Modelo desplegado

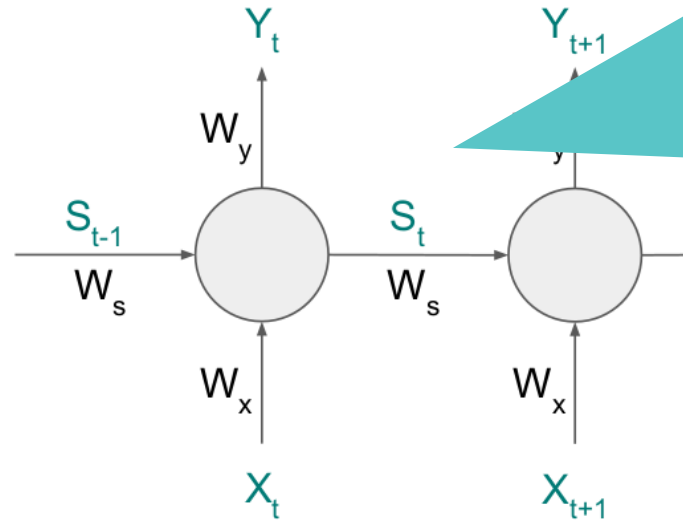
Funcionamiento (I)



- Un mismo modelo de RNN se pueden representar de forma **plegada o desplegada**.
 - Desplegado “en tiempo”.



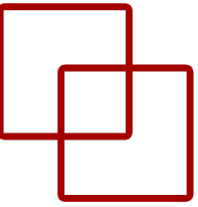
Modelo plegado



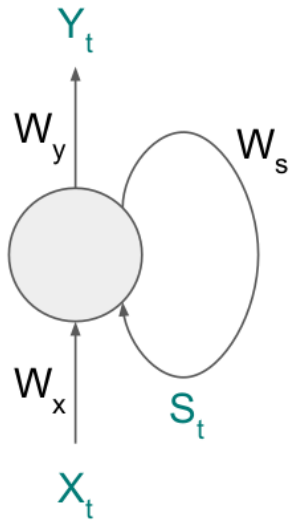
Modelo desplegado

Podemos observar que el estado actual viene dado de la activación anterior (S_{t-1}), el dato actual (X_t); mientras la salida es la predicción actual (Y_t), la activación actual (S_t) y los pesos (W)

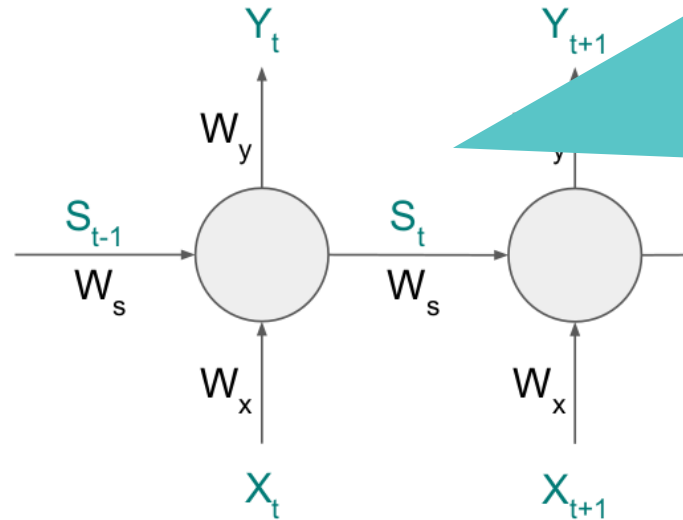
Funcionamiento (I)



- Un mismo modelo de RNN se pueden representar de forma **plegada o desplegada**.
 - Desplegado “en tiempo”.



Modelo plegado

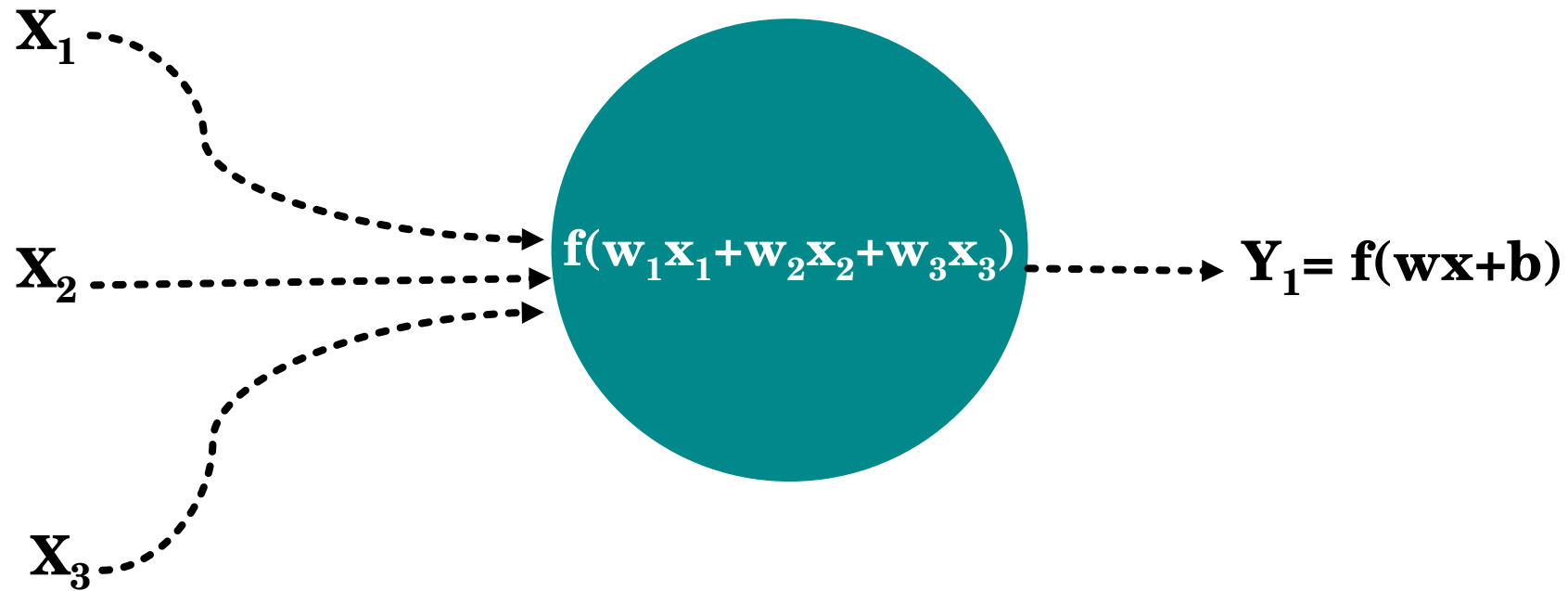
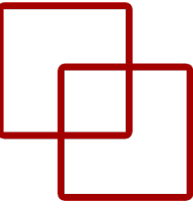


Modelo desplegado

La activación actual (S_t), llamadas estado oculto, son las que permiten preservar y compartir la información entre un estado y otro

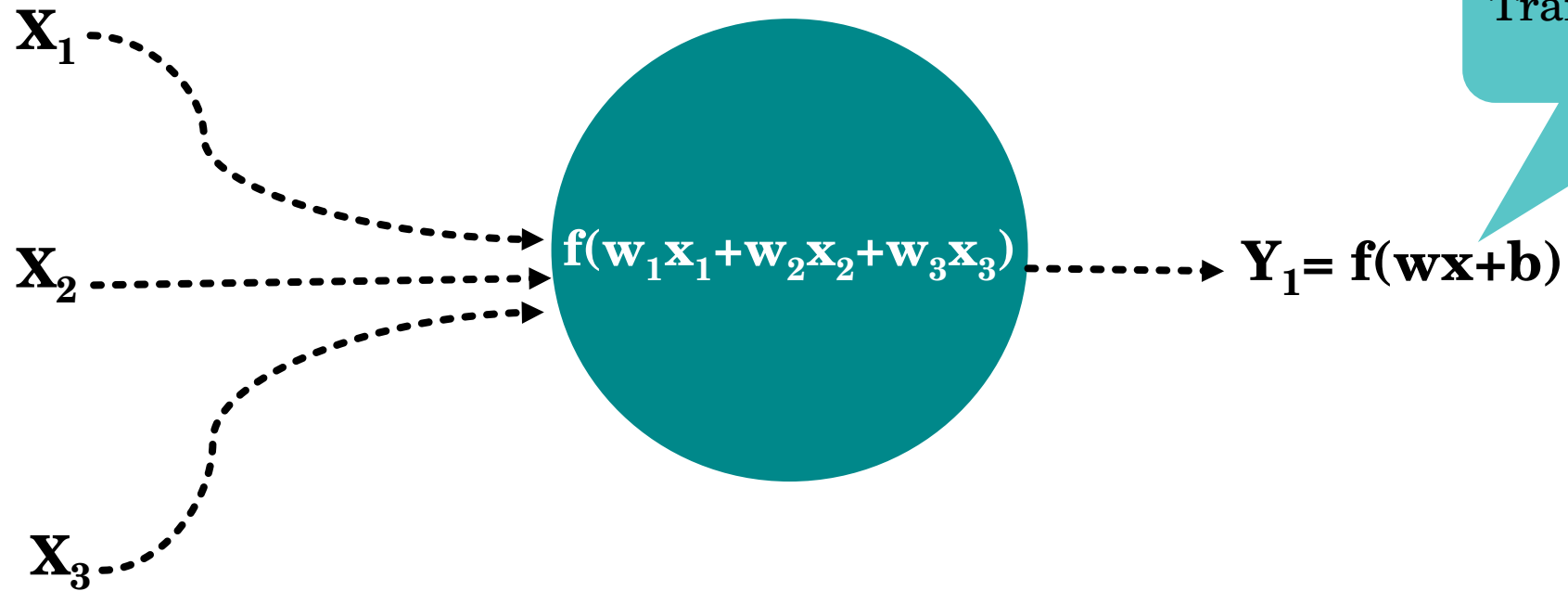
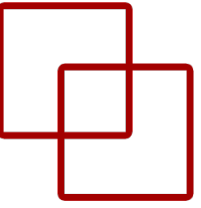
Funcionamiento (II)

La neurona tradicional



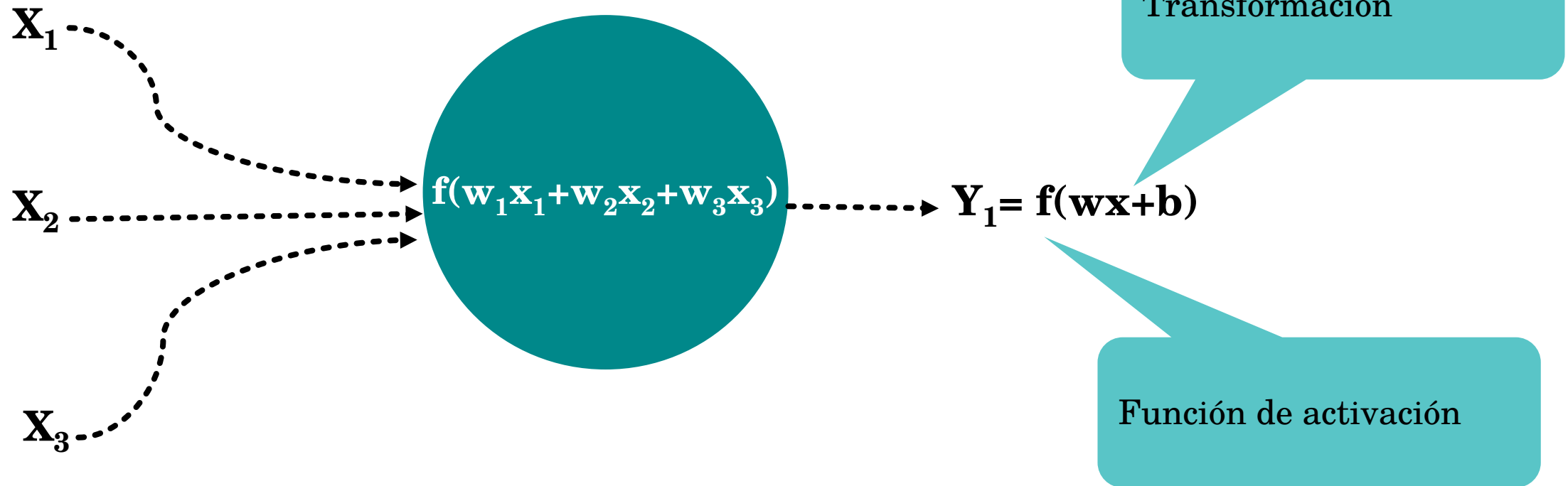
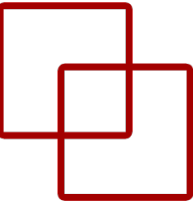
Funcionamiento (II)

La neurona tradicional



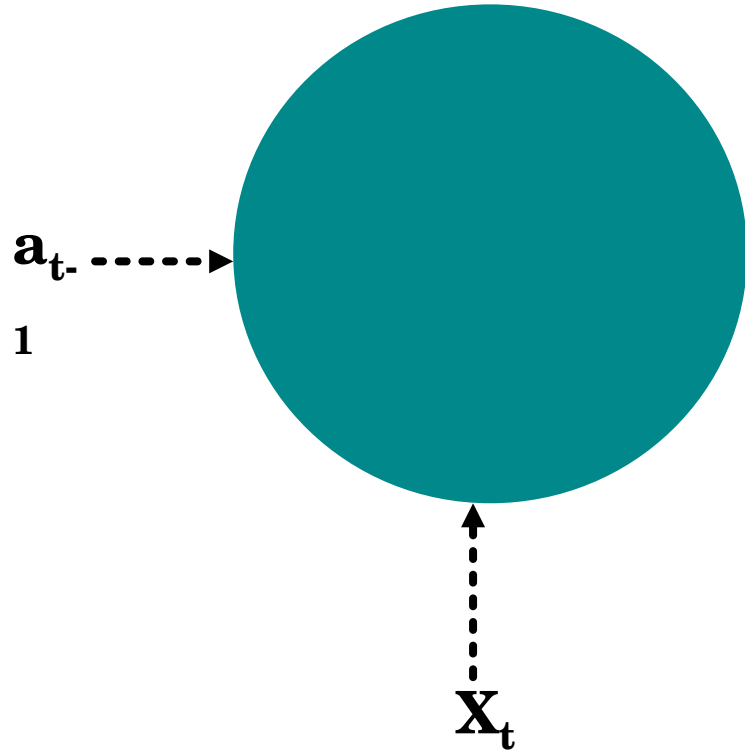
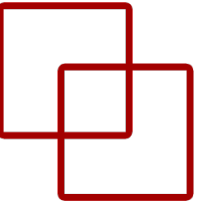
Funcionamiento (II)

La neurona tradicional



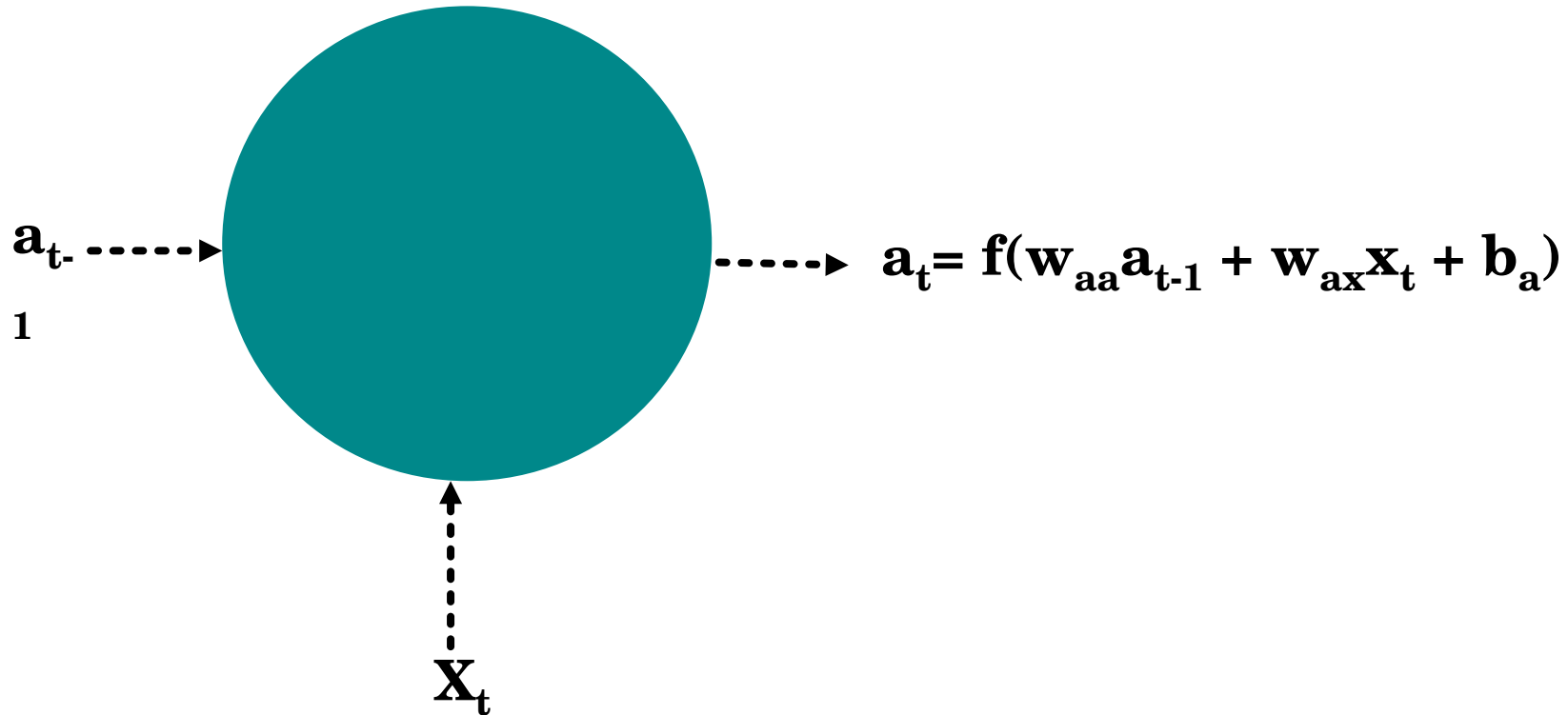
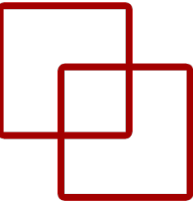
Funcionamiento (II)

RNN



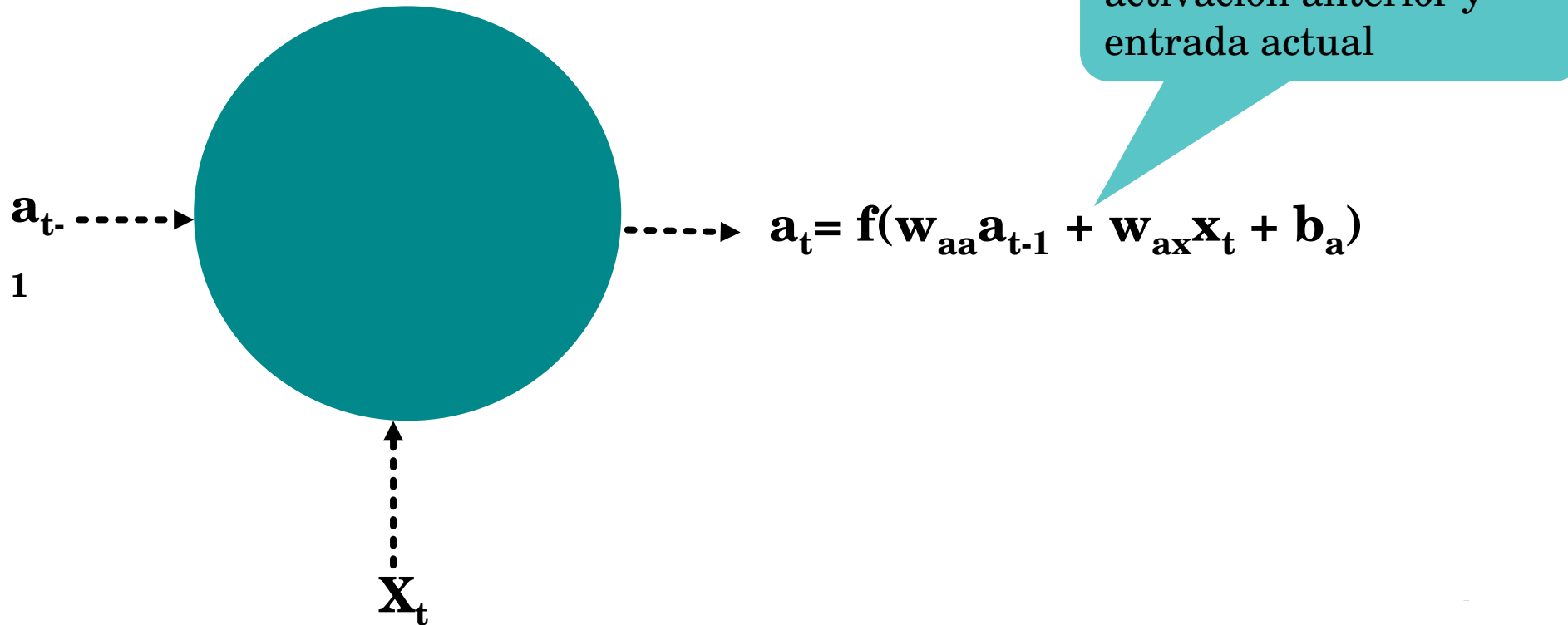
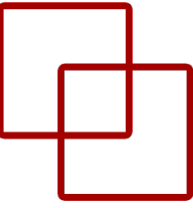
Funcionamiento (II)

RNN



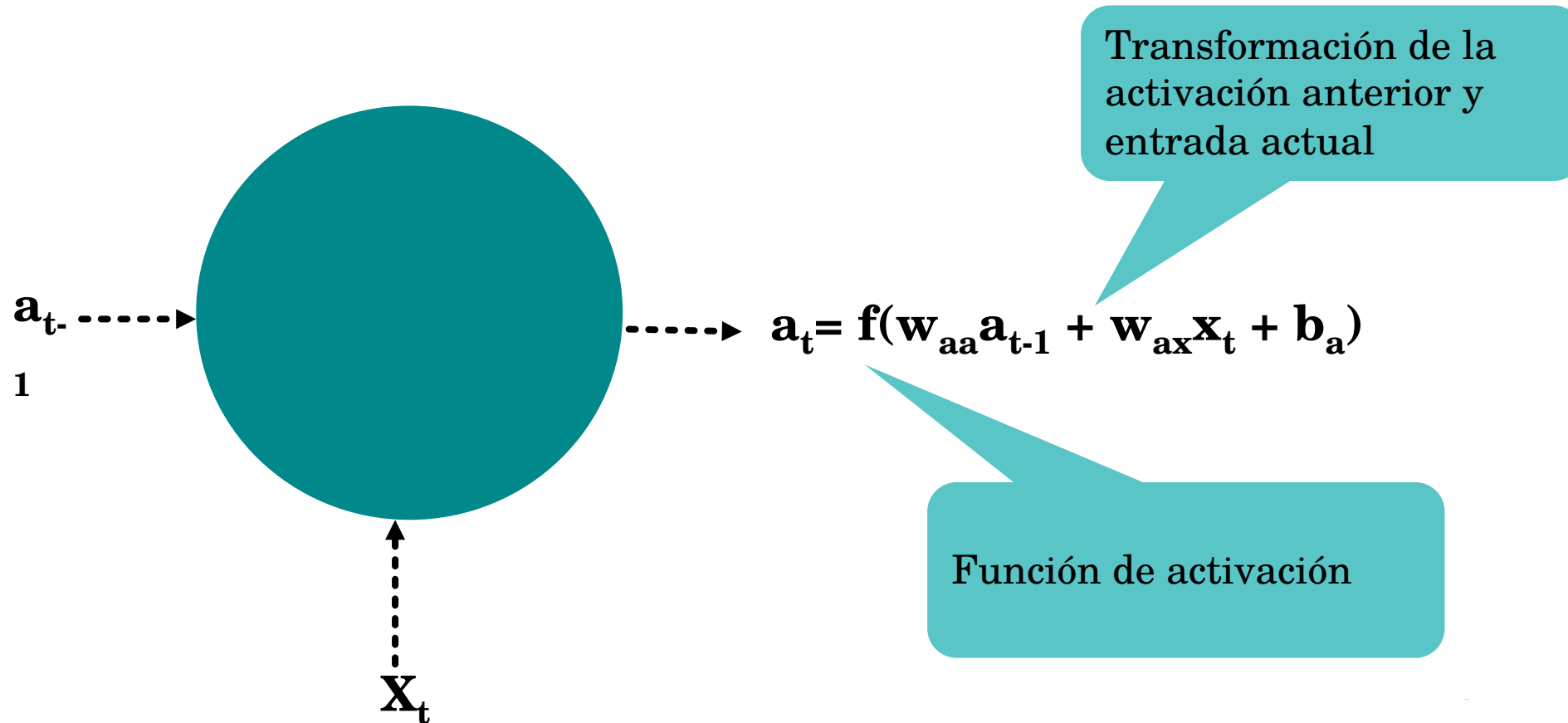
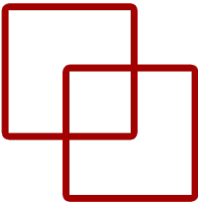
Funcionamiento (II)

RNN



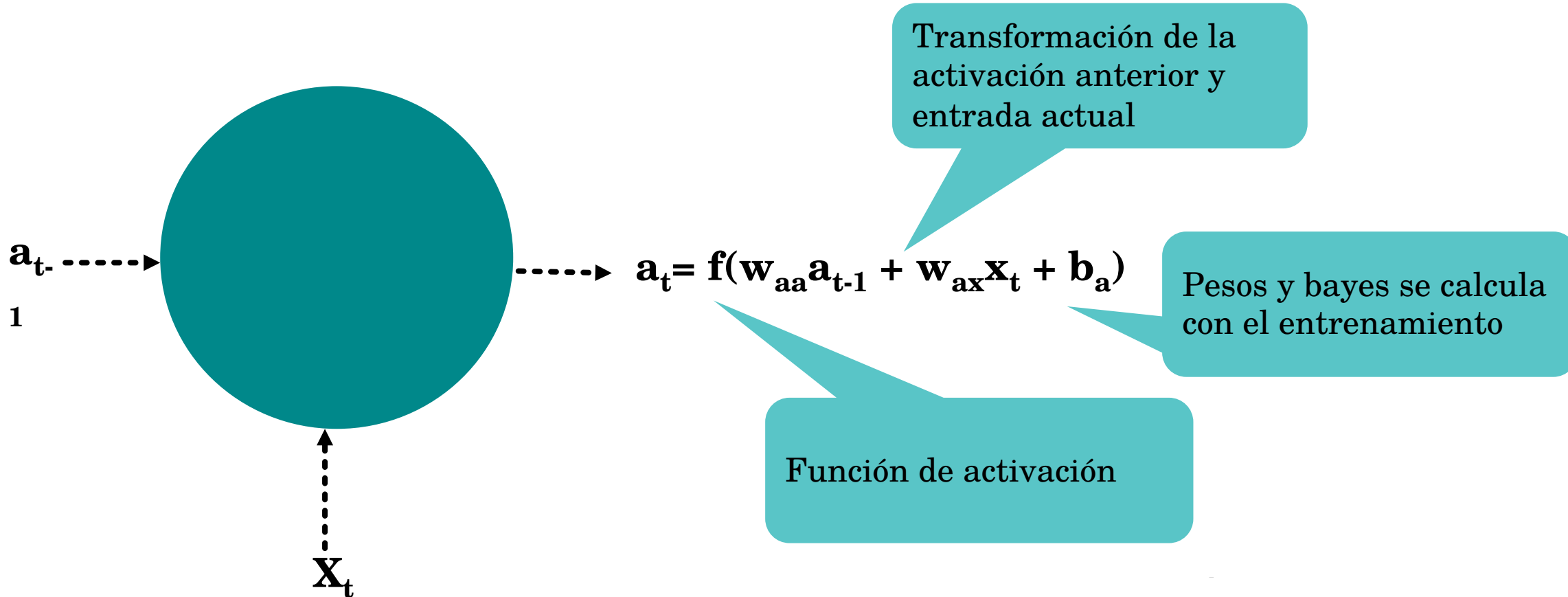
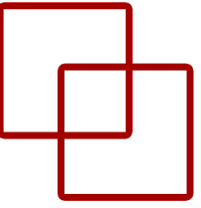
Funcionamiento (II)

RNN



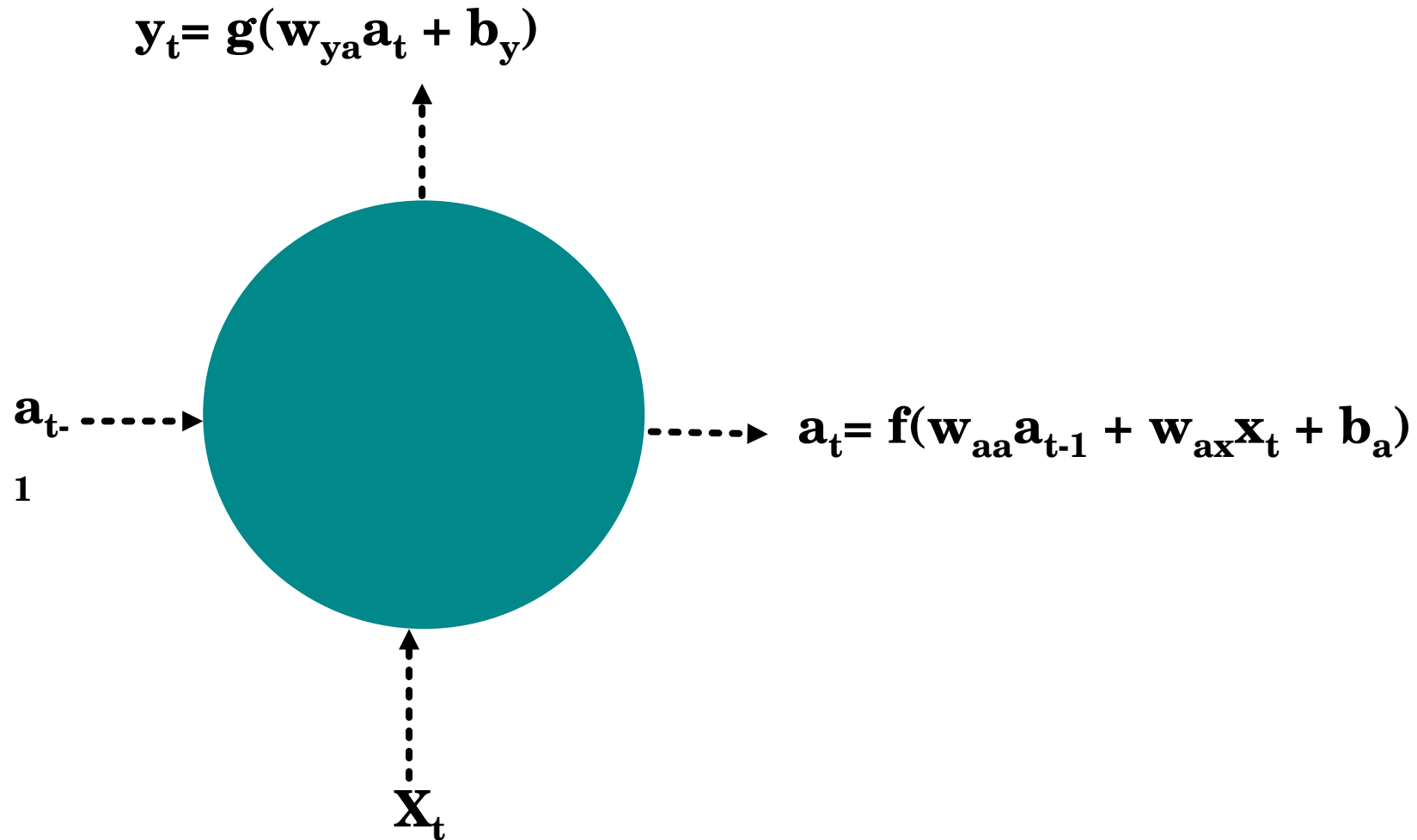
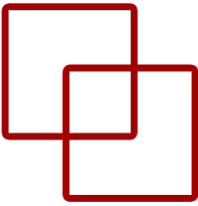
Funcionamiento (II)

RNN



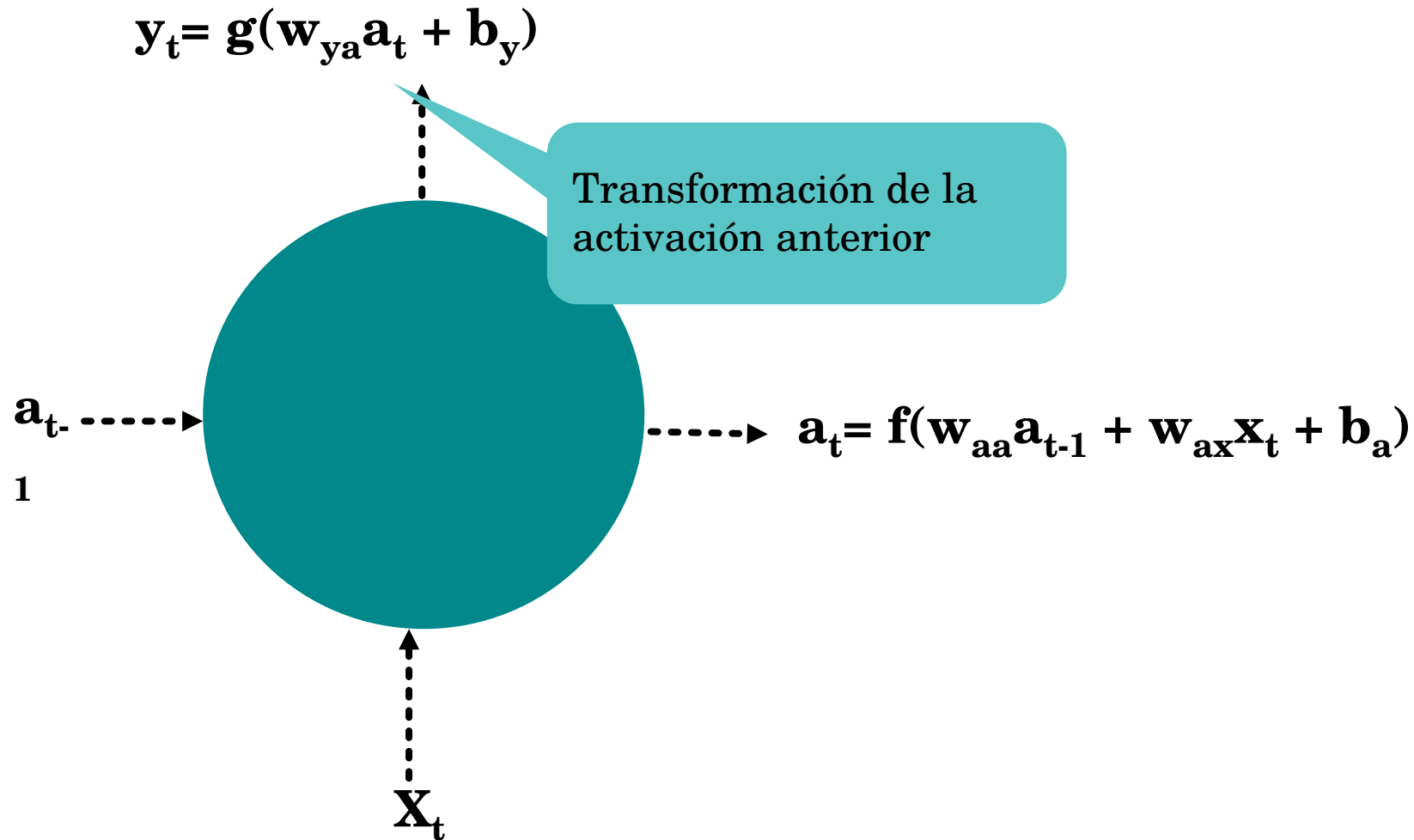
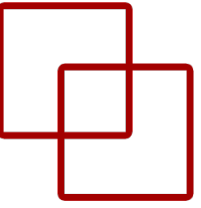
Funcionamiento (II)

RNN



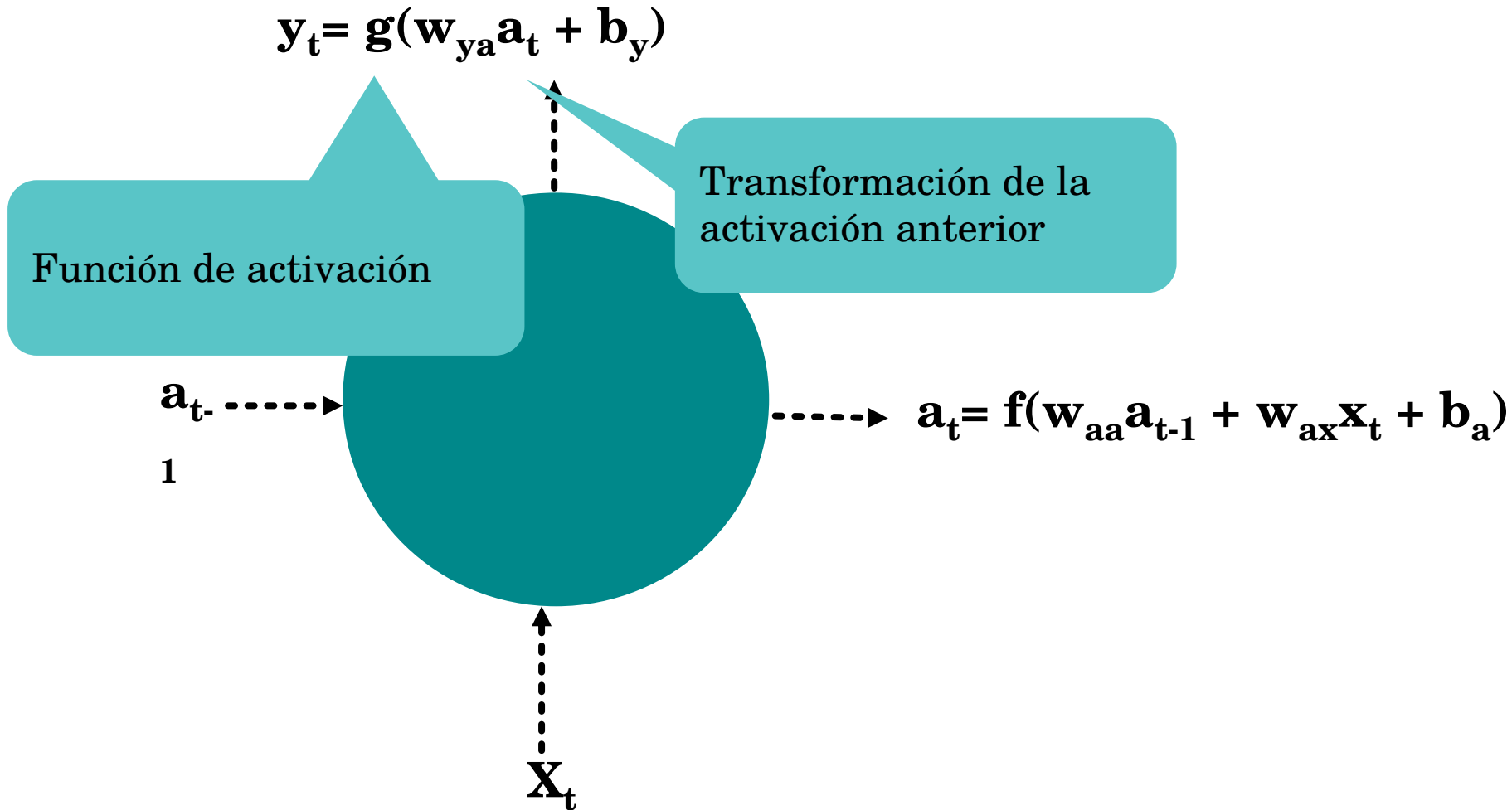
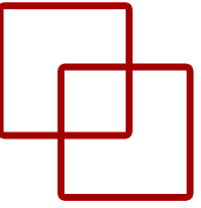
Funcionamiento (II)

RNN



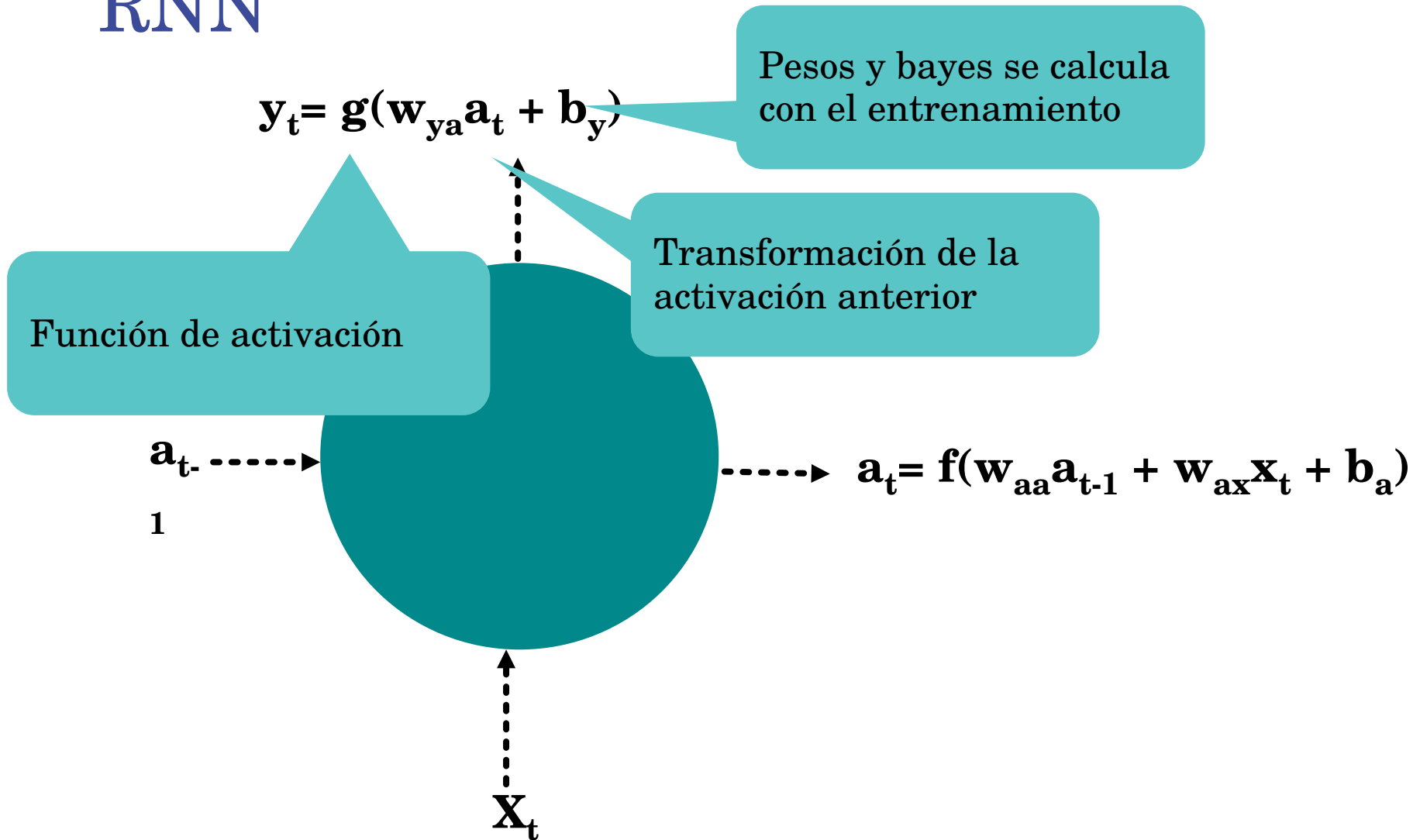
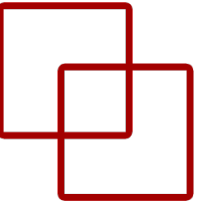
Funcionamiento (II)

RNN



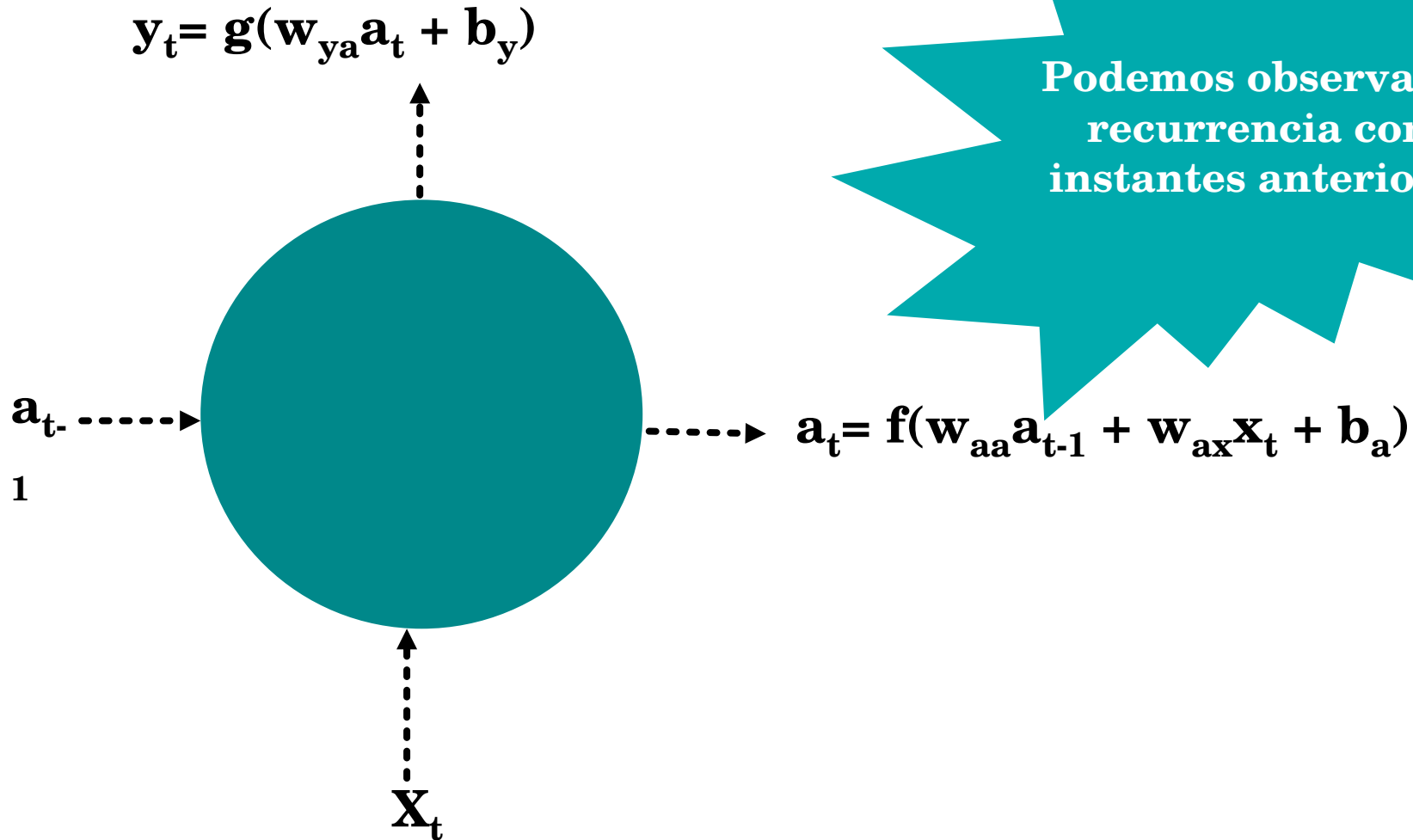
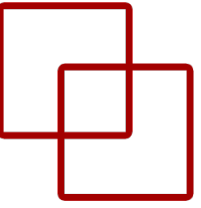
Funcionamiento (II)

RNN



Funcionamiento (II)

RNN

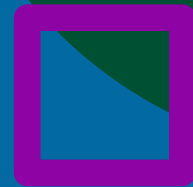


Podemos observar la
recurrencia con
instantes anteriores



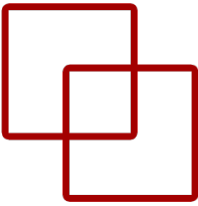
Redes LSTM

ETS de
Ingeniería
Informática

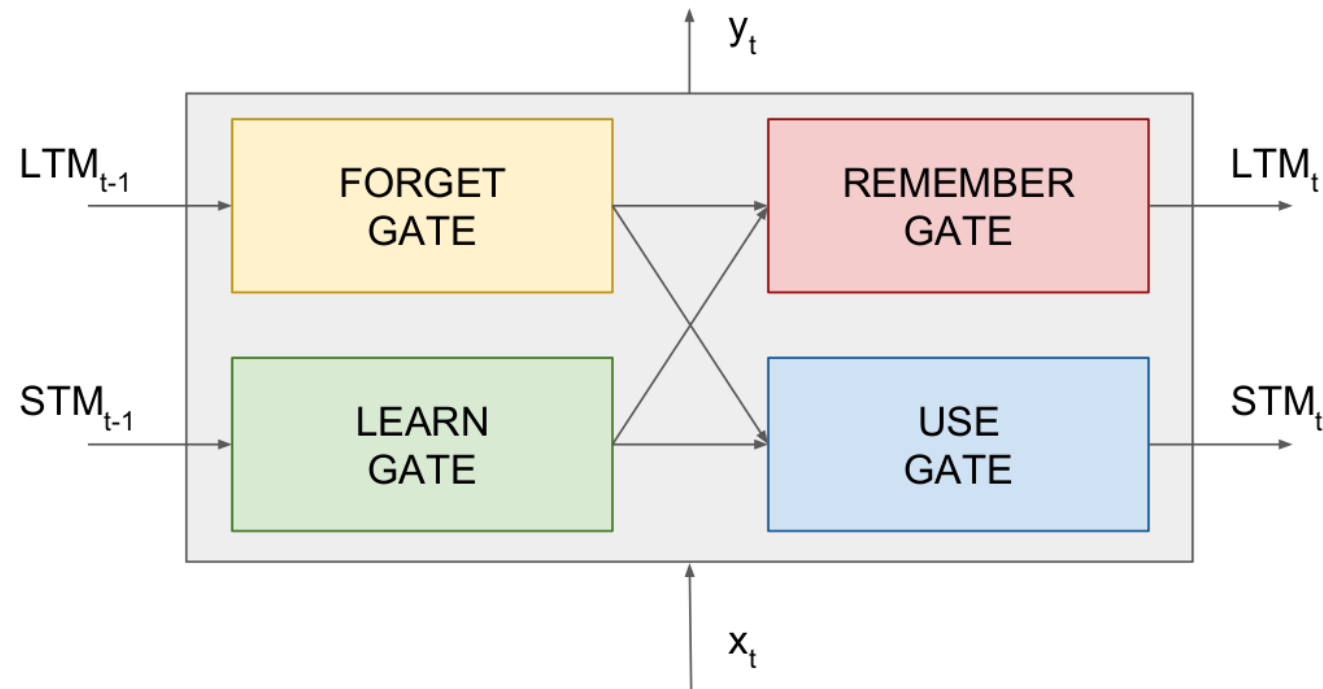


UNED

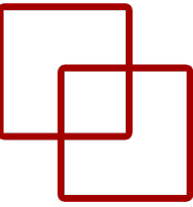
LSTM (I)



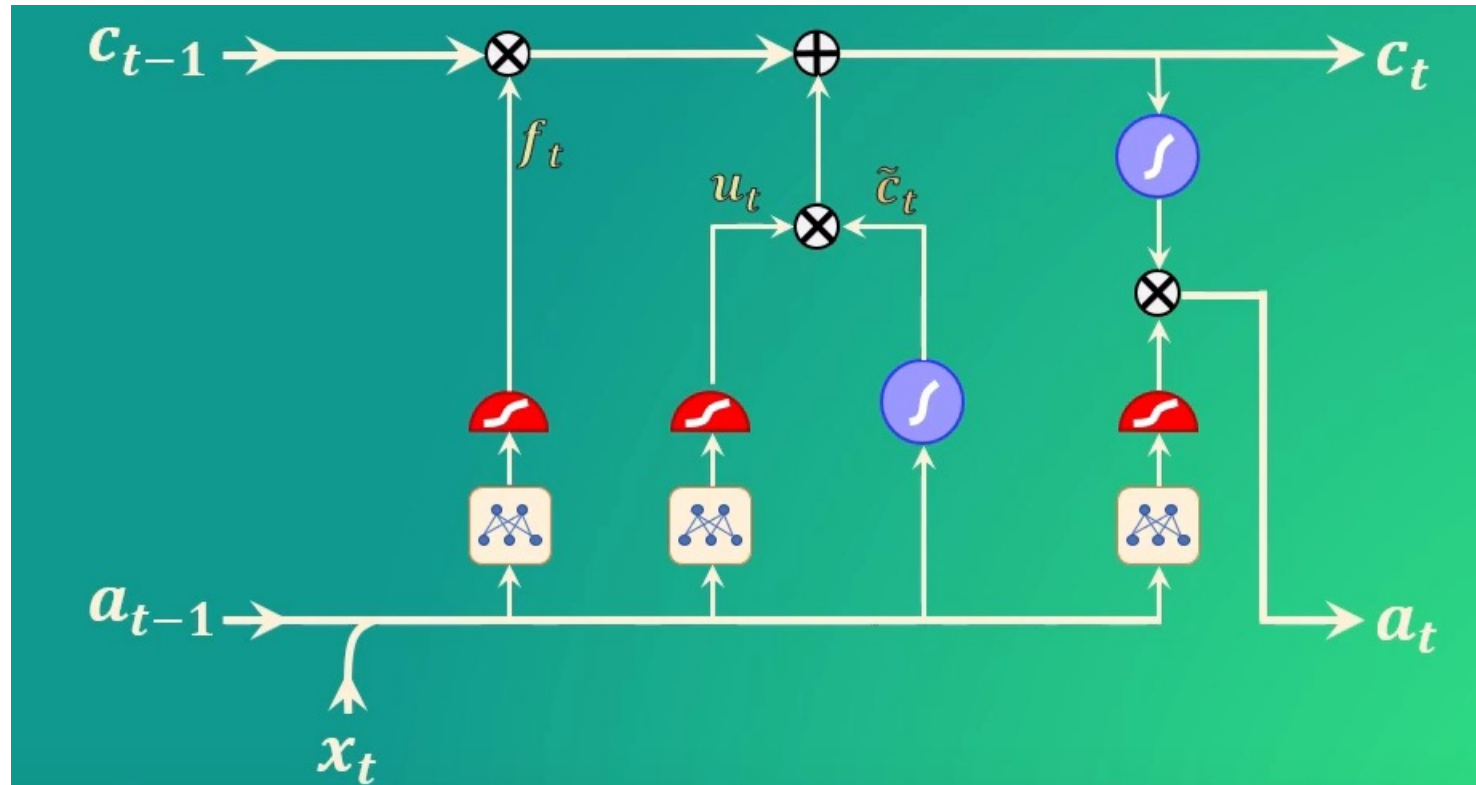
- Unidades más complejas para gestionar memoria a **corto plazo** y memoria a **largo plazo**.
 - Permite que el backpropagation pueda realizarse con una temporalidad mayor al descartar datos de corto/lago plazo
 - Consta de 4 **puertas** para controlar el flujo de información:



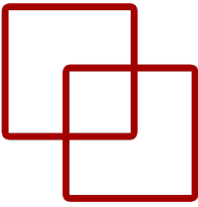
LSTM (II)



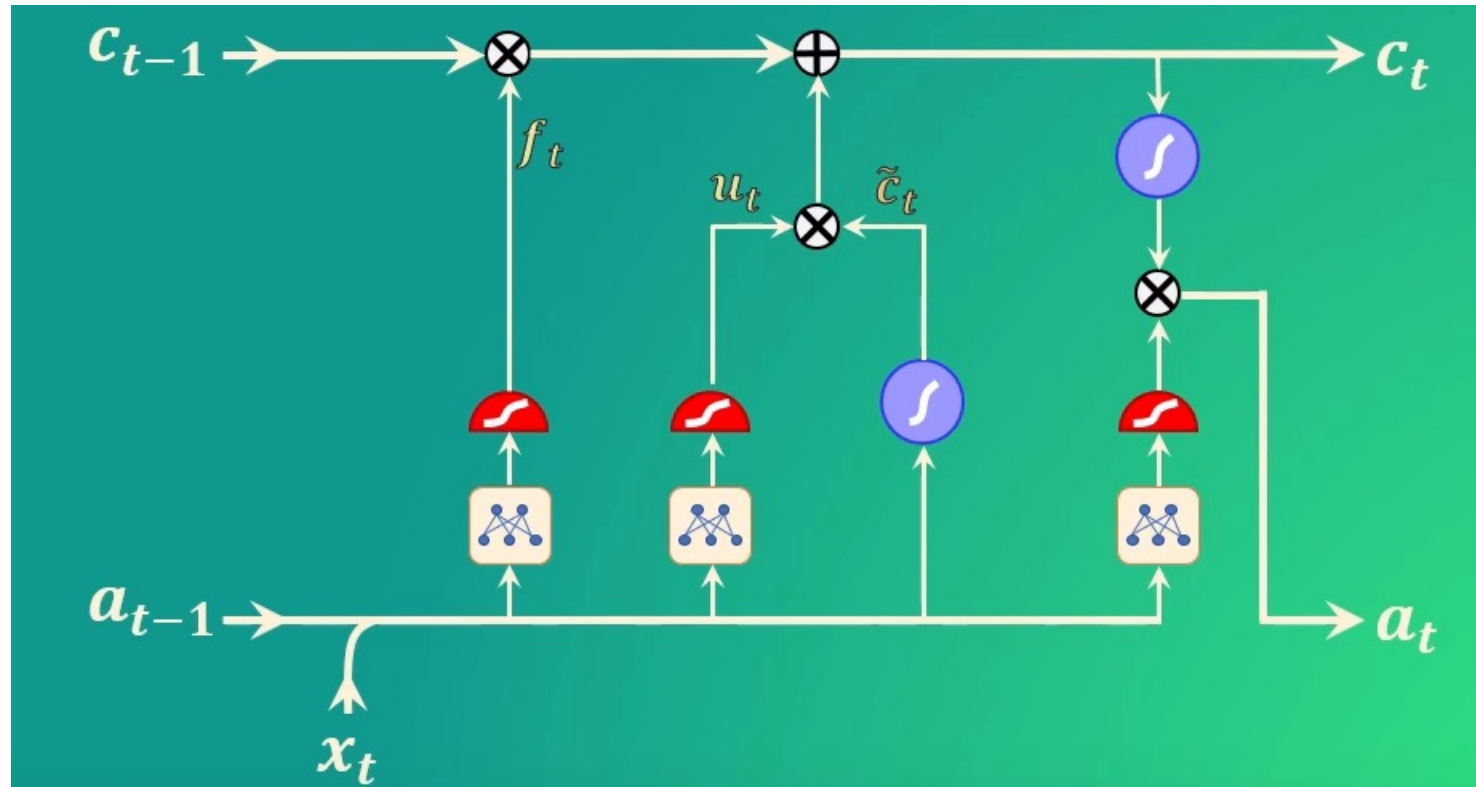
- Cada una de estas puertas están compuestas por una red neuronal, una función sigmoidal y un elemento multiplicador



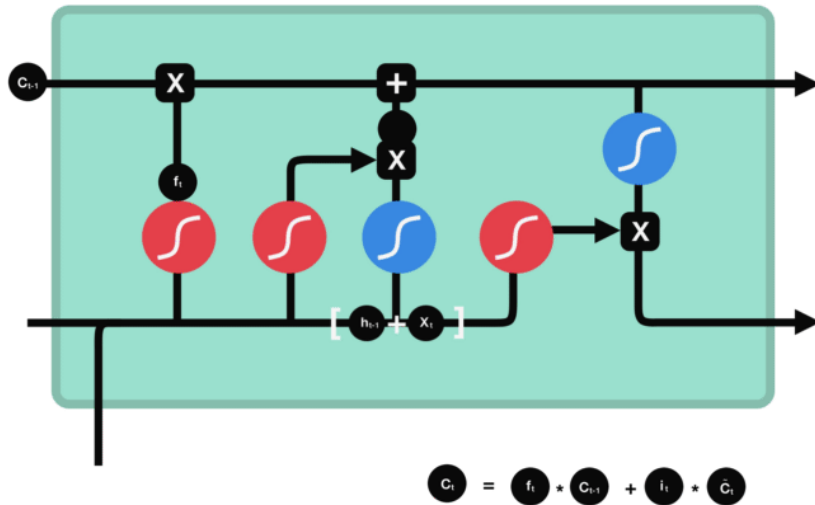
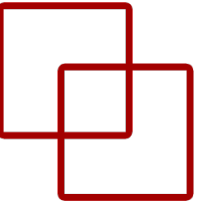
LSTM (II)



- Cada una de estas puertas están compuestas por una red neuronal, una función sigmoidal y un elemento multiplicador

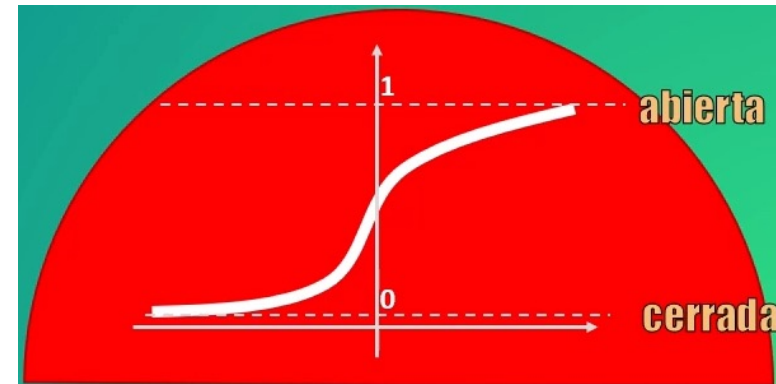


LSTM (III)



- c_{t-1} previous cell state
- f_t forget gate output
- i_t input gate output
- \hat{c}_t candidate
- c_t new cell state

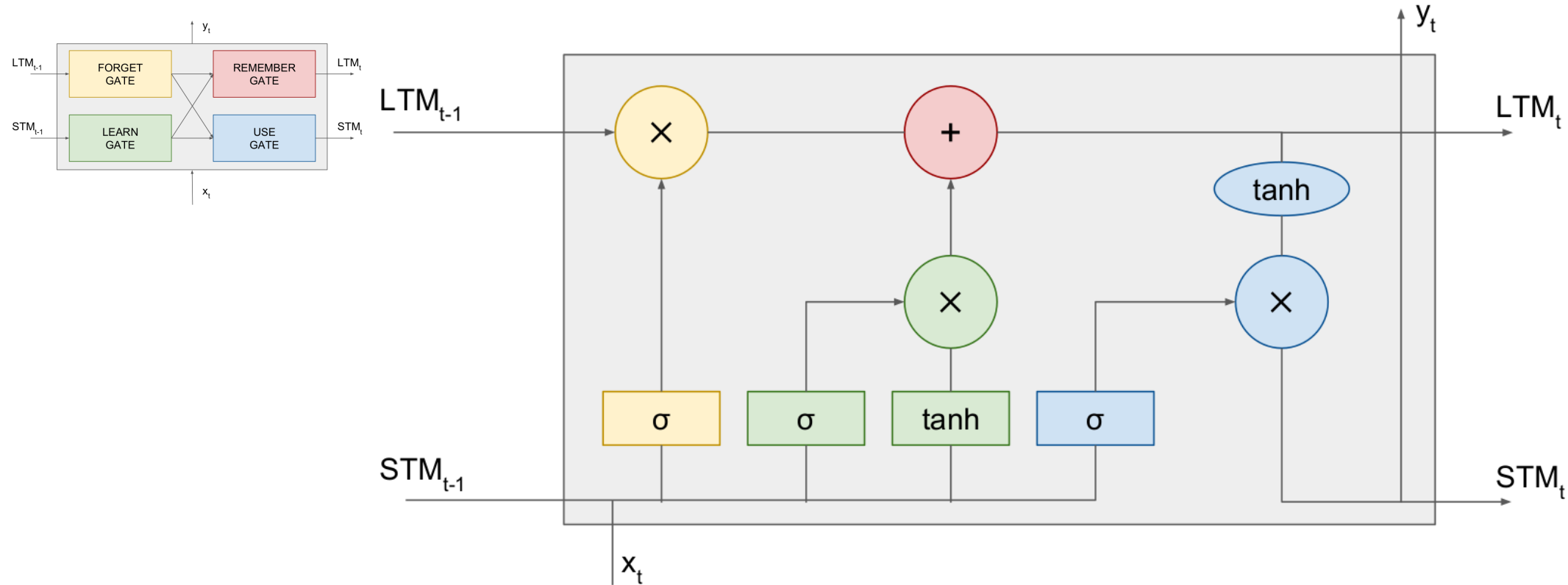
La función de activación le dará el comportamiento de válvula (si deja pasar el dato o no)



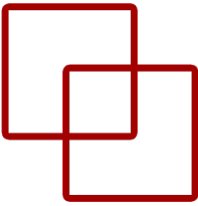
LSTM (IV)



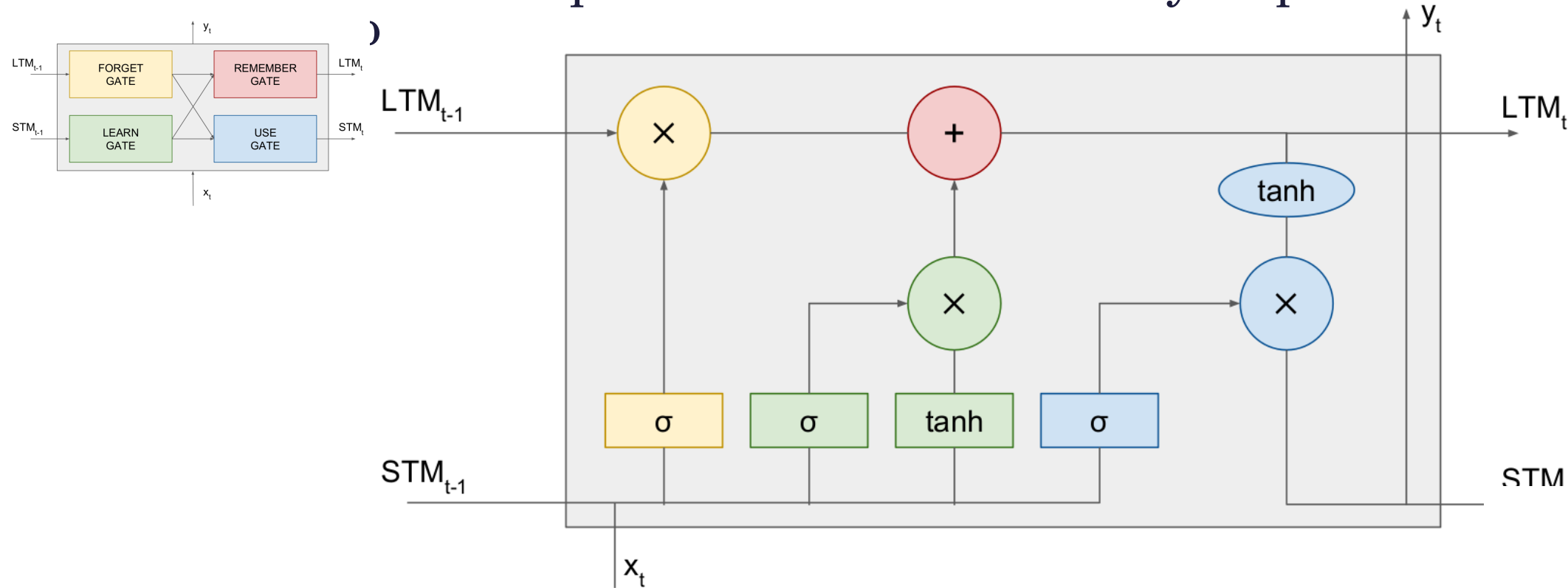
- **Learn gate:** Combina la entrada actual y la memoria a corto plazo.
 - La Sigmoide selecciona datos y establece cuales serán los que aprender



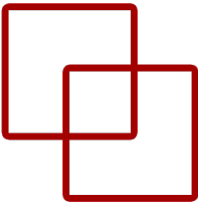
LSTM (V)



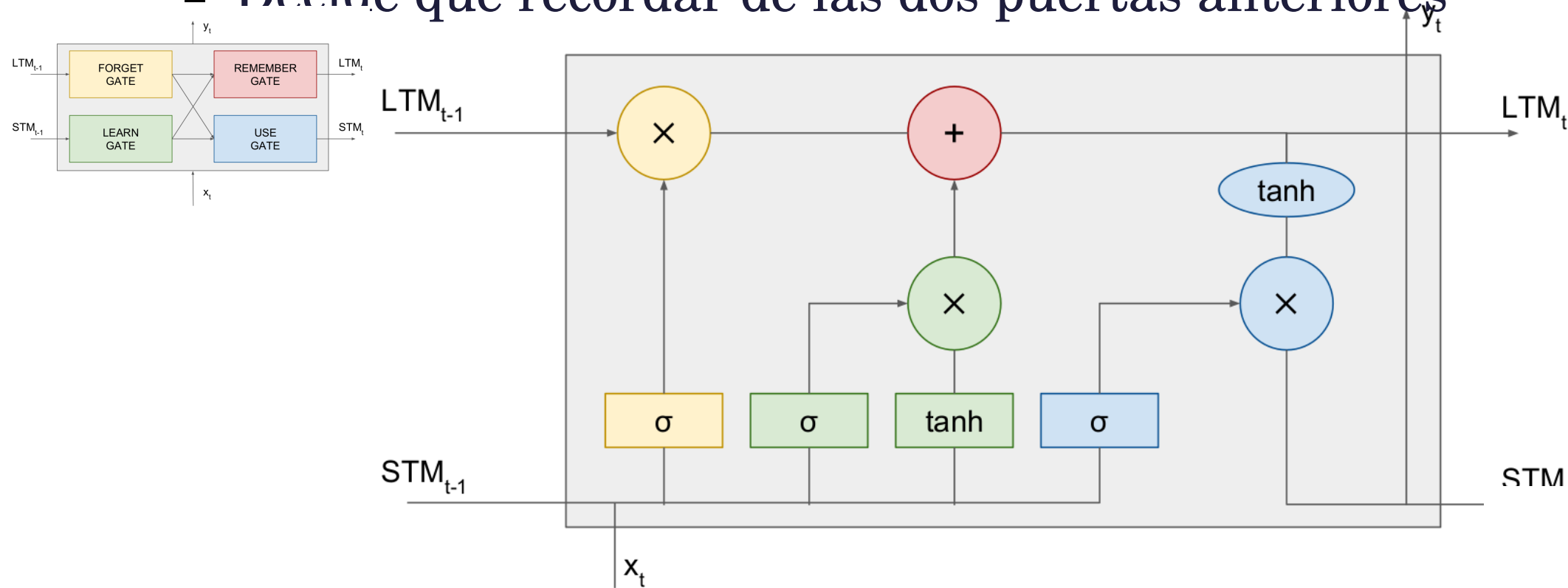
- **Forget gate:** Olvida parte de la entrada de la memoria a largo plazo (el “factor de olvido” depende de la entrada actual y la memoria a corto plazo).
 - Permite decir qué información descartar y no pasar a la celda de



LSTM (VI)



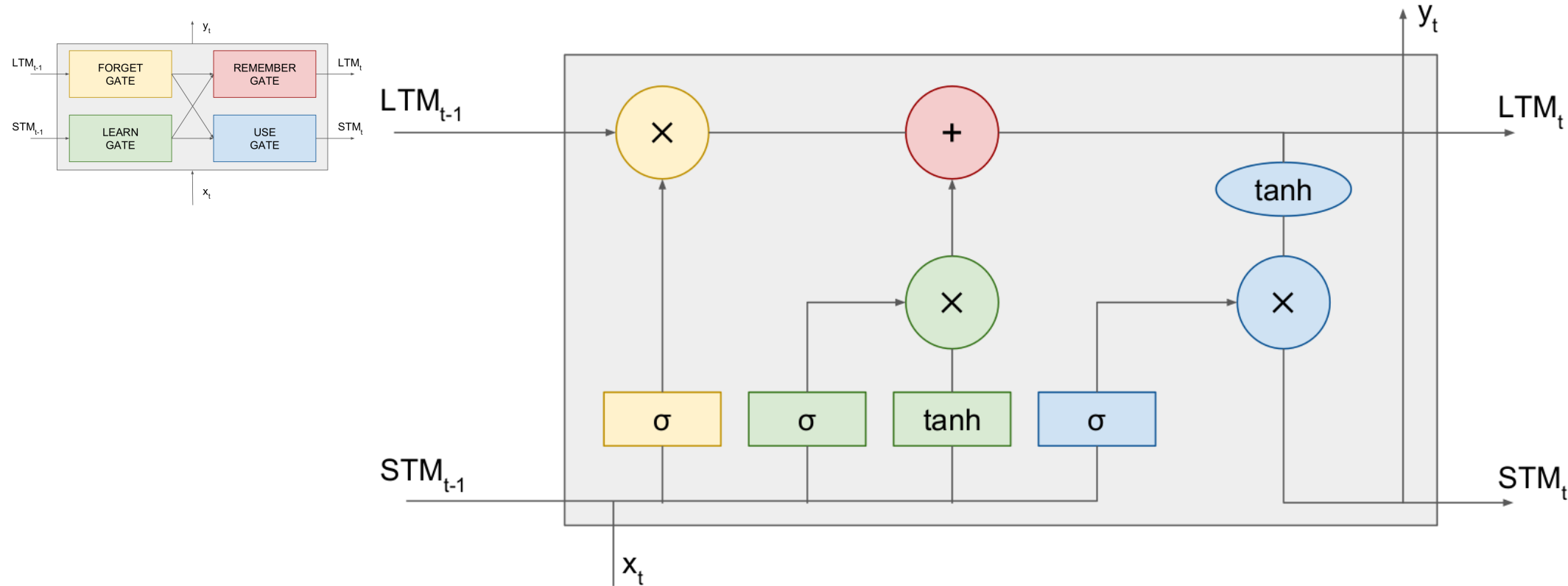
- **Remember gate:** Combina la memoria a corto plazo (salida de la *learn gate*) y la memoria a largo plazo (salida de la *forget gate*).
 - Decide qué recordar de las dos puertas anteriores



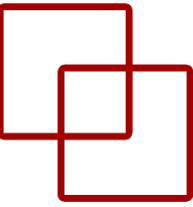
LSTM (VII)



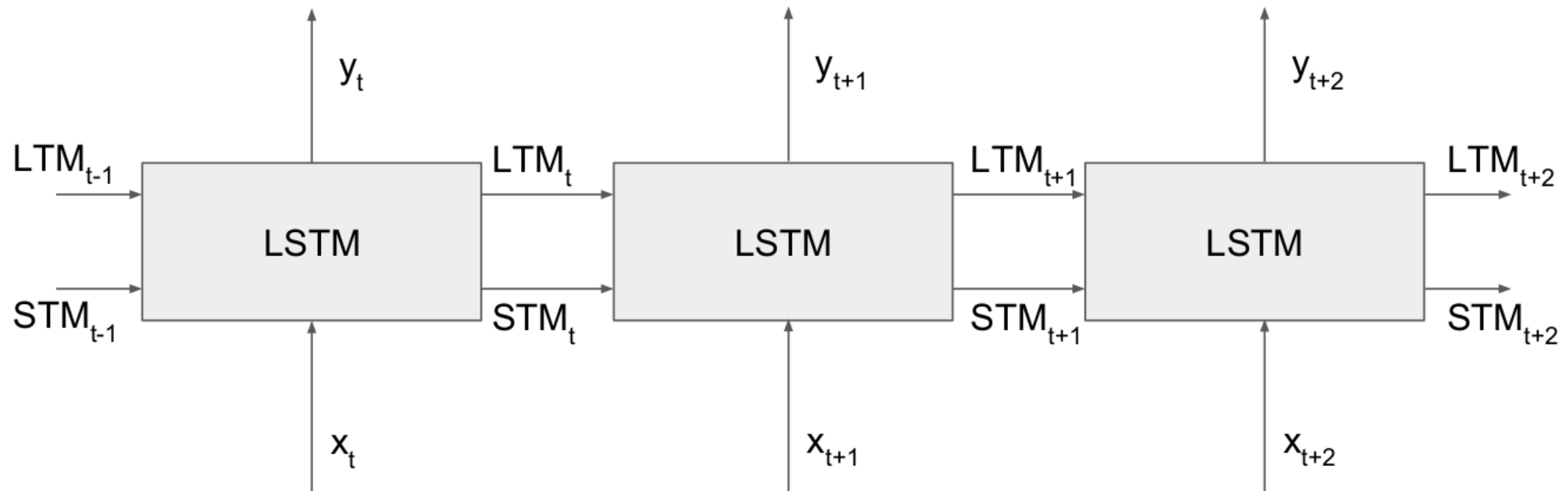
- **Use gate:** Combina la entrada actual, la memoria a corto plazo y la memoria a largo plazo (salida de *learn gate*) para generar la nueva memoria a corto plazo y la salida.



LSTM (VI)



- La salida de memoria a corto plazo y memoria a largo plazo sirve para realimentar la siguiente entrada.



¡GRACIAS!

The logo of the Universidad Nacional de Educación a Distancia (UNED), consisting of the letters 'UNED' in a white, stylized, sans-serif font on a dark green square background.

UNED

The logo of the Escuela Técnica Superior de Ingeniería Informática (ETS de Ingeniería Informática), consisting of the text 'ETS de Ingeniería Informática' in a white, sans-serif font on a dark green square background.

ETS de
Ingeniería
Informática

Dr. Manuel Castillo-Cara

www.manuelcastillo.eu

**Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingeniería Informática
Universidad Nacional de Educación a Distancia (UNED)**