



Maestría en Inteligencia Artificial Aplicada (MNA)

# Modelos Secuenciales, Series de Tiempo y RNN-LSTM

Inteligencia Artificial y Aprendizaje Automático  
Luis Eduardo Falcón Morales

# Modelos Secuenciales y Series de Tiempo

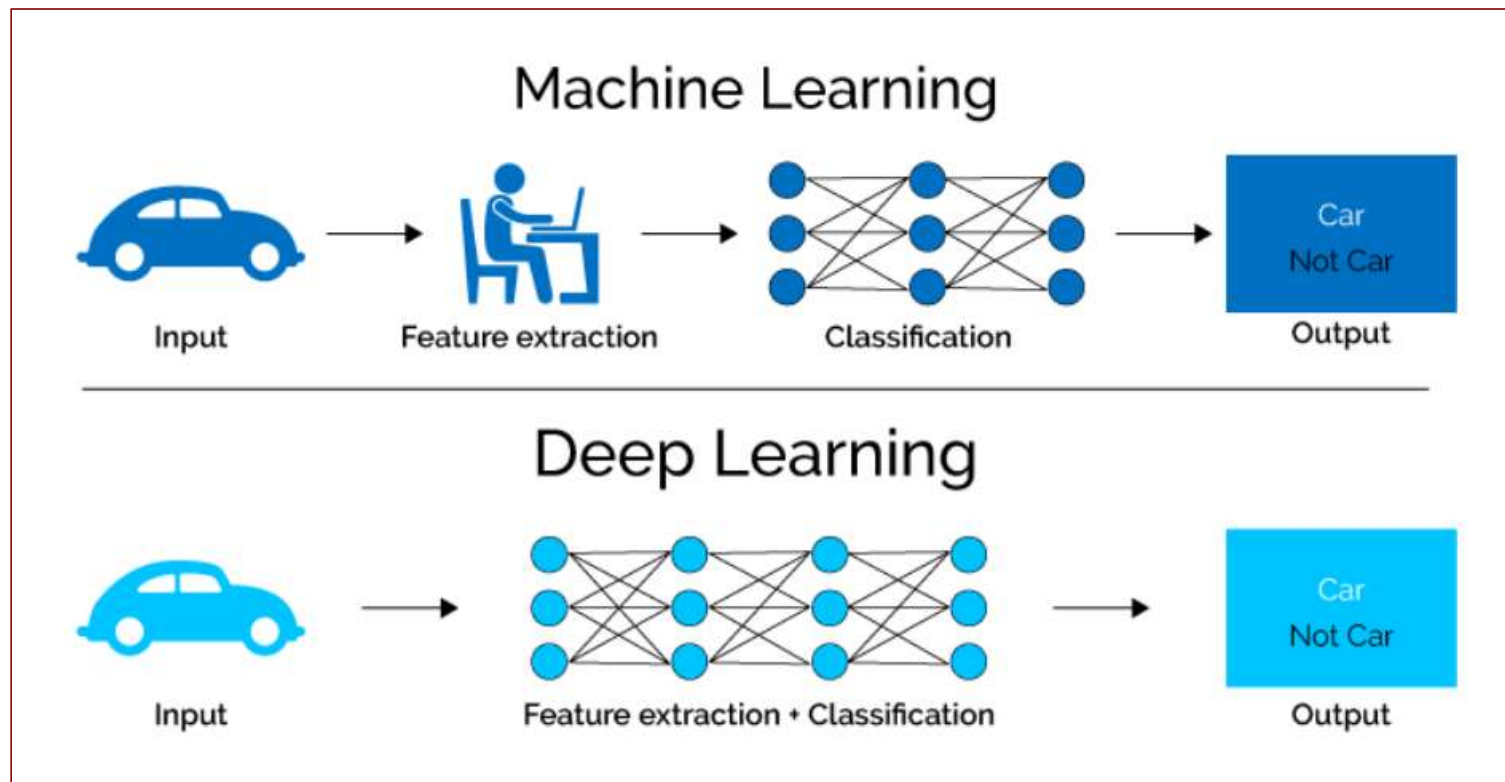


Mucha de la complejidad que involucran los problemas de análisis de textos radica en la dependencia o relación que guarda una palabra con las anteriores y posteriores en un documento.

Este tipo de relación también lo encontramos de manera similar en los problemas relacionados con variables temporales, como las series de tiempo. Por ejemplo, el tratar de predecir un índice financiero a lo largo del tiempo.

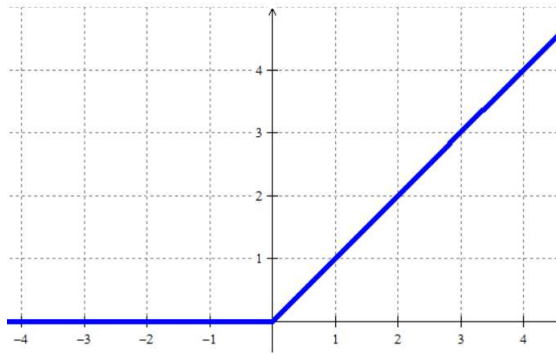
Los llamados modelos secuenciales surgen para ayudar a explicar este tipo de problemas temporales y aunque gran parte de estas temáticas están dentro del área conocida como Aprendizaje Profundo (Deep Learning) y este no es un curso al respecto, no se puede negar que varios de estos temas están teniendo un gran impacto en particular en el área de series temporales, por lo que veremos lo necesario en esta semana para que tengas una primera idea de cómo se está aplicando.

## Aprendizaje Automático vs Aprendizaje Profundo (Machine Learning vs Deep Learning)



Unidad Lineal Rectificada  
(Rectified Linear Unit)  
**Función de Activación**

**ReLU**



$$f(x) = \max(0, x)$$

A la salida de cada neurona (filtro o kernel) se aplica una función de activación. En las redes profundas la función de activación que usualmente se aplica es la ReLU o alguna de sus variantes.

## Batch Normalization (BN)

- Se recomienda aplicarlo después de la función de activación ReLU y antes de la siguiente capa de la red, por ejemplo, una Pool u otra CONV.
- Sin embargo, algunos autores recomiendan aplicar BN después de la función de activación (ReLU).
- En general, a la fecha se han obtenido mejores resultados aplicando BN **después** de la ReLU, pero el debate sigue vigente.
- Generalmente se normaliza con la siguiente ecuación:

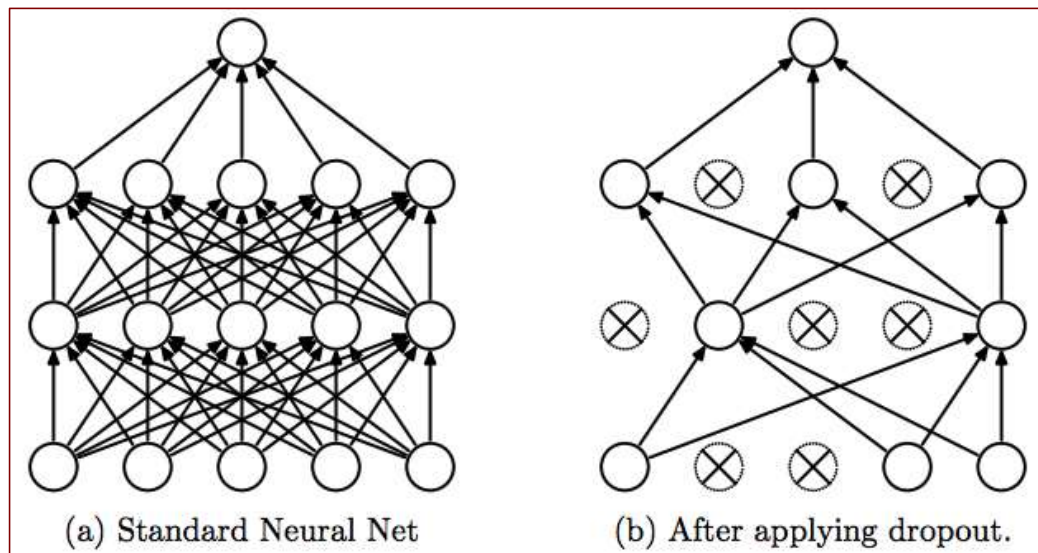
$$\hat{x} = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + eps}}$$

El resultado son valores de media cero y varianza 1.

- Durante el proceso de entrenamiento se calculan la media  $\mu_\beta$  y la varianza  $\sigma_\beta^2$  de cada mini-batch  $\beta$ .
- $eps \sim 10^{-7}$ , se utiliza para evitar una división por cero.
- Aunque hace más lento el proceso, puede ayudar a disminuir el *overfitting* y a obtener mejores resultados en menos *epochs*.

## Dropout

Como método para prevenir el sobre-entrenamiento la conectividad de cada neurona es activada y desactivada de manera aleatoria en cada proceso de entrenamiento mini-batch.

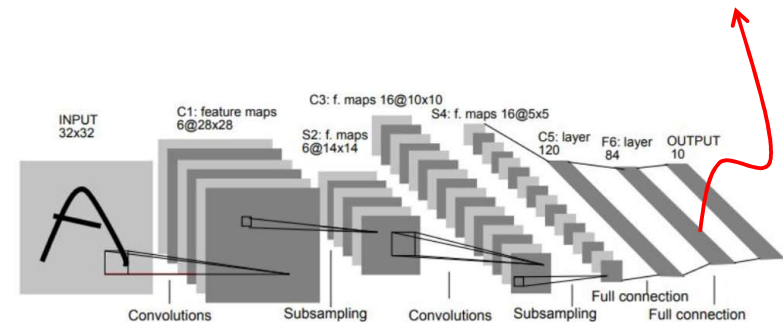
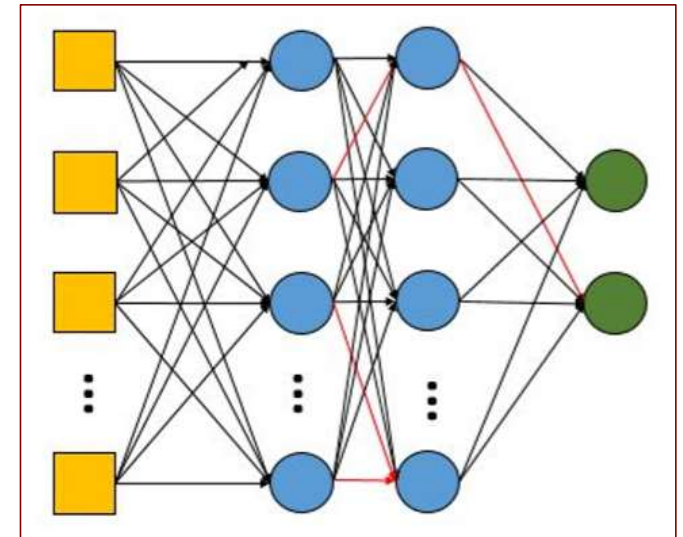


Este método puede ser aplicable a cualquier tipo de red neuronal, y suelen colocarse entre capas CONV o FC.

Apoya en la generalización del modelo, es decir, podemos decir que funciona como un método de regularización.

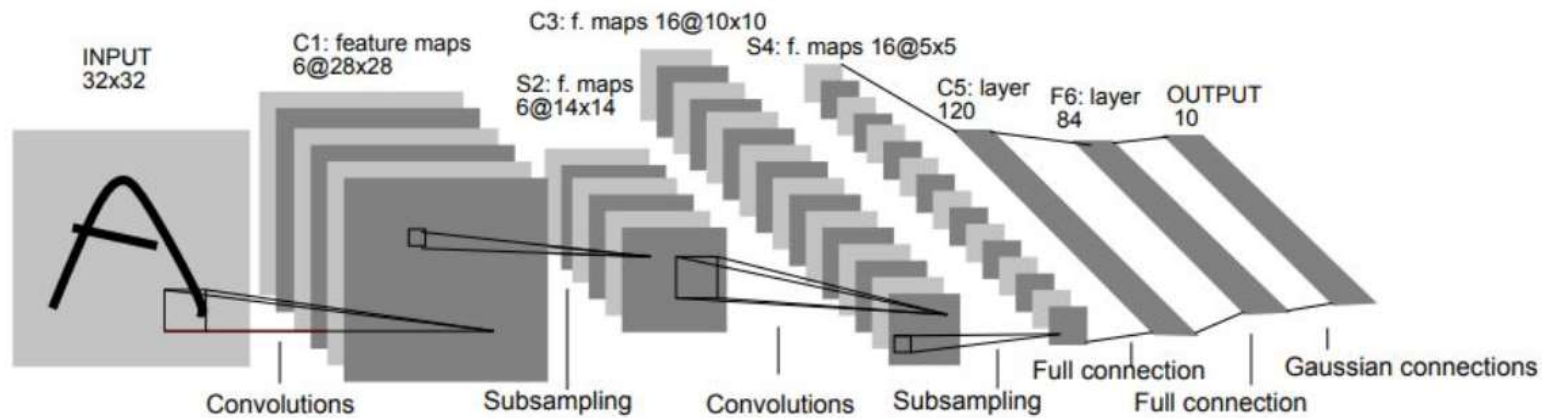
## Capas Totalmente Conectadas (Fully Connected (FC) / Dense layer)

- Las capas FC siempre se aplican al final de la red.
- Después de una FC no se vuelve a aplicar una capa convolucional CONV.
- Se suelen aplicar 1 o 2 capas FC antes de la última capa de salida.
- Cuando se aplican varias capas FC, en ocasiones el tamaño de ellas puede ir disminuyendo o amentando.
- También se les llama *dense layer*.
- Una capa FC es equivalente a verla como un kernel del tamaño de la imagen (*feature map*) 2D de entrada.





# LeNet-5 Architecture (1998)



PROC. OF THE IEEE, NOVEMBER 1998

1

## Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

*Abstract—*

Multilayer Neural Networks trained with the backpropagation algorithm constitute the best example of a successful Gradient-Based Learning technique. Given an appropriate network architecture, Gradient-Based Learning algorithms

### I. INTRODUCTION

Over the last several years, machine learning techniques, particularly when applied to neural networks, have played an increasingly important role in the design of pattern

<http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf>



# ImageNet Classification with Deep Convolutional Neural Networks

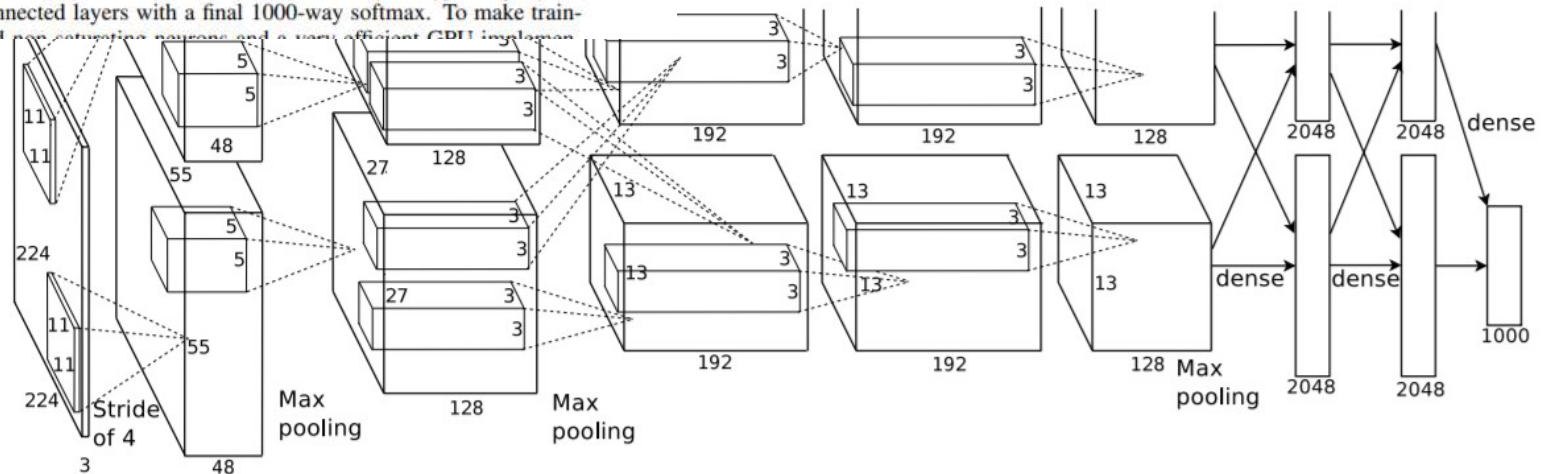
Alex Krizhevsky  
University of Toronto  
kriz@cs.utoronto.ca

Ilya Sutskever  
University of Toronto  
ilya@cs.utoronto.ca

Geoffrey E. Hinton  
University of Toronto  
hinton@cs.utoronto.ca

## Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training feasible, we used non-saturating activation functions and a very efficient GPU implementation.



[ 2012 - modelo AlexNet ]

Primer modelo en el cual se utilizan los GPUs (hardware diseñado inicialmente para los videojuegos) para acelerar el entrenamiento de estos modelos y que da inicio a la revolución del aprendizaje profundo (deep learning).

## Lenguaje Natural y Series de Tiempo

El análisis que requiere entender un enunciado, es similar al que se requiere para analizar las series temporales.

Enunciado:

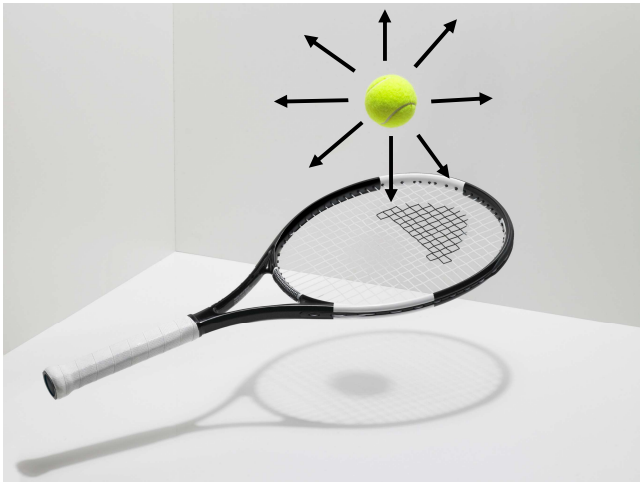
1	2	3	4	5	6	7	8	9	10	11
La	mucha	luz	es	como	la	mucha	sombra,	no	deja	ver.

Octavio Paz

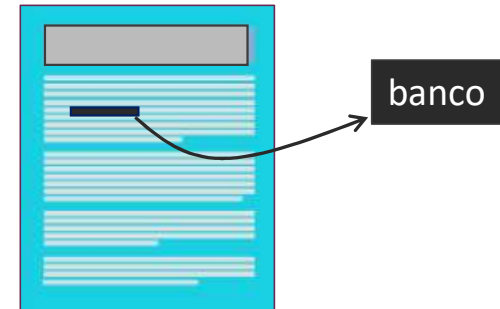
Serie de tiempo:

1	2	3	4	5	...
$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	...

## Modelos Secuenciales / Recurrentes



Si tenemos la imagen congelada de una pelota que estaba en movimiento, ¿cómo poder predecir la dirección de su movimiento con solo dicha información?



Si conocemos solo una palabra de un texto, ¿cómo saber el significado de dicho texto?

Requerimos conocer las palabras que le rodean, es decir su posición en el texto o en el tiempo de lectura del documento.

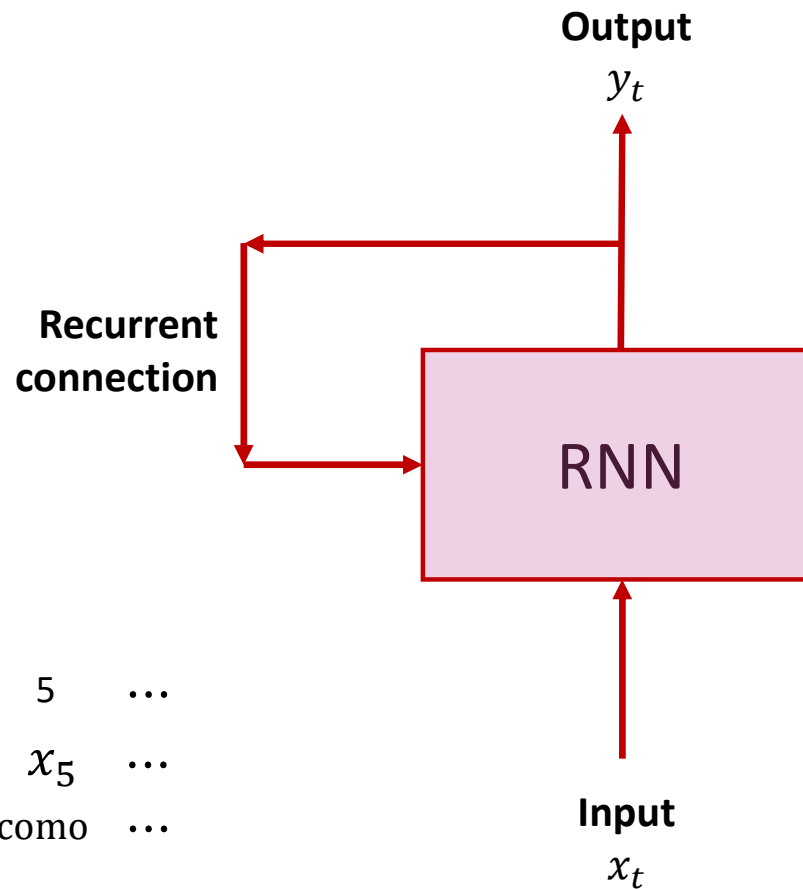
# Modelo de Aprendizaje Profundo (Deep Learning)

## Red Neuronal Recurrente

### Recurrent Neural Network (RNN)

(1982 – 1986)

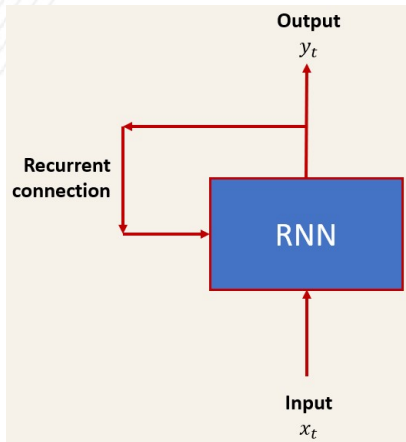
- Las RNN pueden procesar la información considerando el pasado.
- Podemos decir entonces que las RNN tienen memoria.
- Propagan información a lo largo del tiempo.



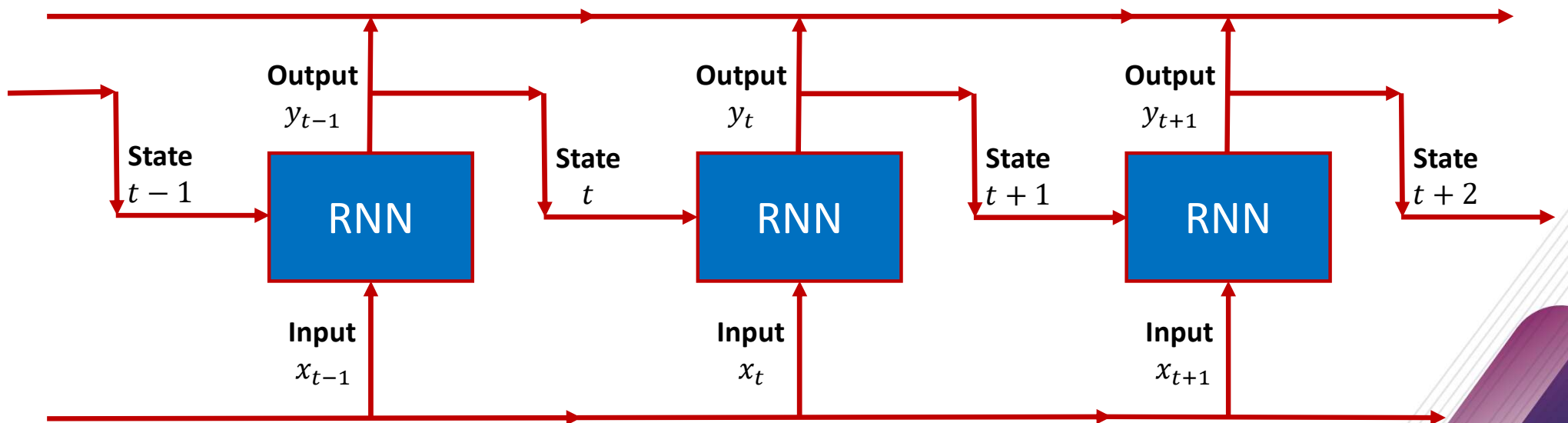
Serie de tiempo:

1	2	3	4	5	...
$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	...
La	mucha	luz	es	como	...

## RNN Simple

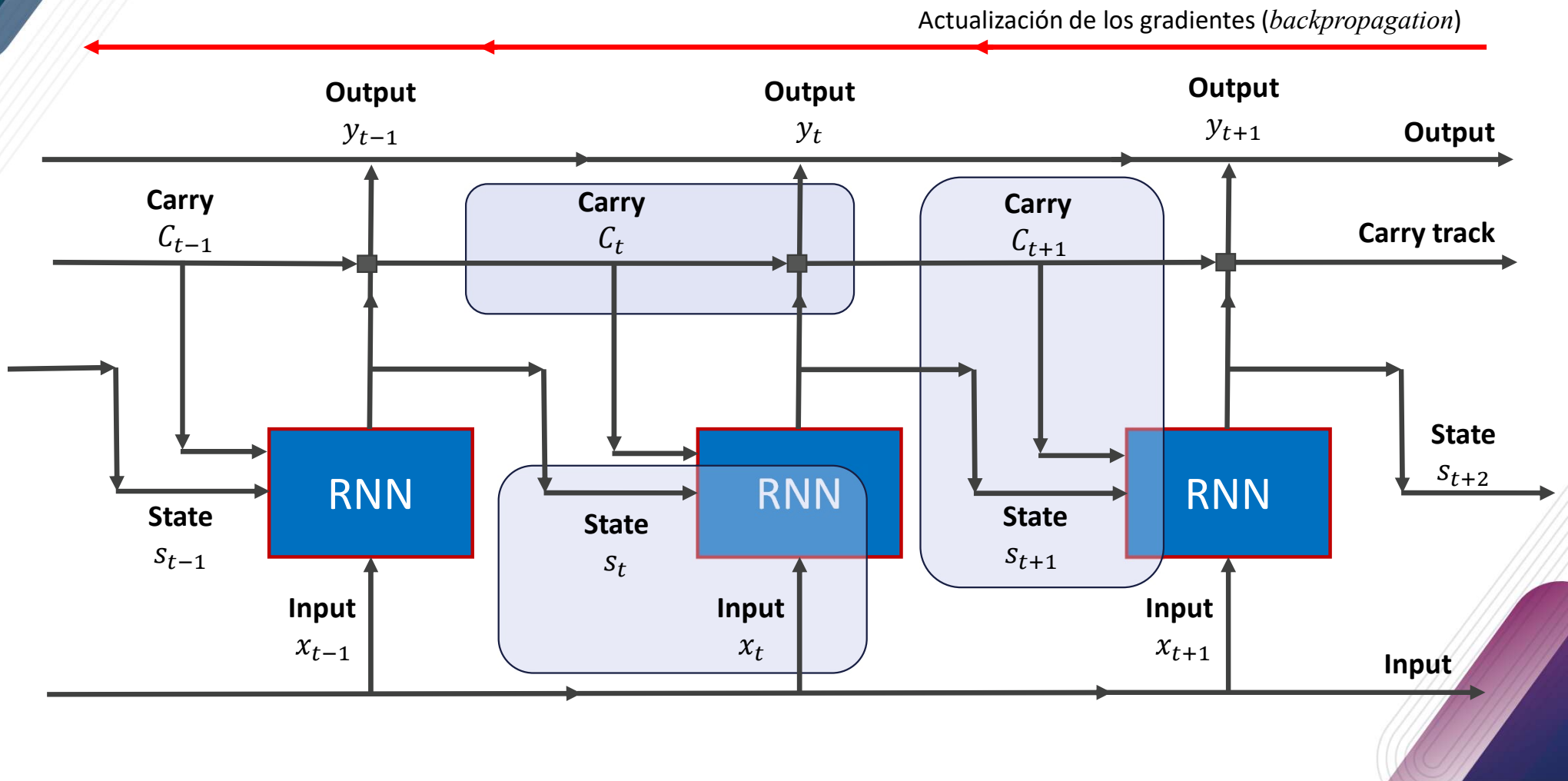


Una RNN puede desdoblarse a lo largo del tiempo y visualizarse como sigue:



# LSTM : Long Short Term Memory

(1995 – 1997)



## LSTM : Long Short Term Memory & GRU : Gated Recurrent Unit

- En la práctica, el problema de la disminución del gradiente hace que no se puedan entrenar redes con demasiadas capas.
- Para evitar dicho problema se conserva información del pasado a lo largo del tiempo y se va incrustando en los bloques convolucionales para preservar la información.
- En el artículo de Hochreiter & Schmidhuber, 1997, se define la arquitectura LSTM para atacar por primera vez el problema de la memoria y la disminución del gradiente.
- Se utiliza además la función ReLU.
- Los modelos LSTM y GRU son los más conocidos que preservan información a lo largo del tiempo. Podemos decir que todos son de la familia de modelos secuenciales o recurrentes.
- GRU es un modelo más simple que LSTM, en particular LSTM conserva memoria a corto y largo plazo de manera independiente, mientras que GRU tiene un solo tipo de memoria.

Paper: Long short-term memory  
S. Hochreiter & J. Schmidhuber. 1997

<https://ieeexplore.ieee.org/abstract/document/6795963>





D.R.© Tecnológico de Monterrey, México, 2022.  
Prohibida la reproducción total o parcial  
de esta obra sin expresa autorización del  
Tecnológico de Monterrey.