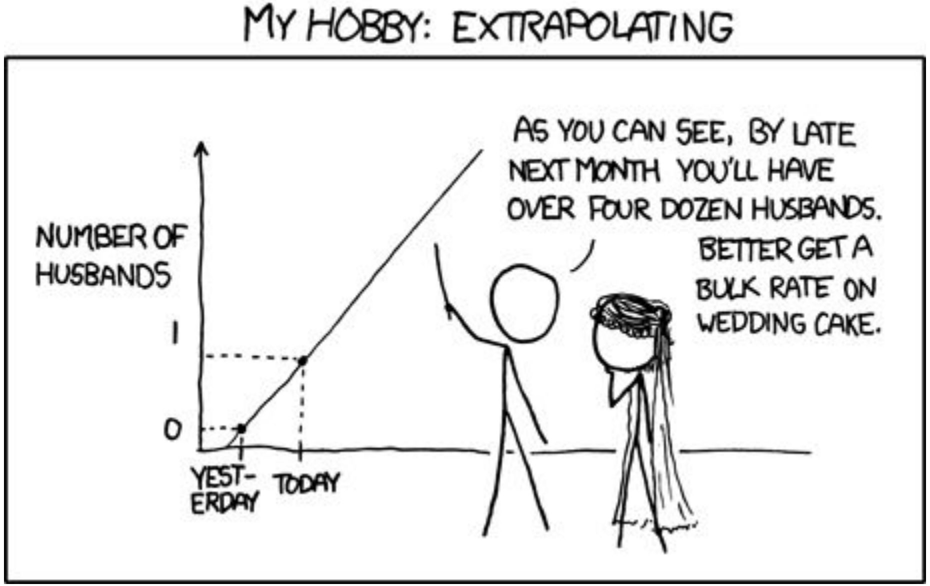
Series Temporales: Forecasting parte 2

Federico Albanese
(falbanese@dc.uba.ar)
@f__albanese



https://xkcd.com/605/

¿ Qué vamos a ver hoy?

¿ Qué vamos a ver hoy?

- Deep Learning:
  - Neural Networks
  - CNN: Convolutional Neural Network (LeCun, 1989)
  - RNN: Recurrent Neural Network (Rumelhart et al., 1986a)
    - LSTM: Long Short-Term Memory (Hochreiter and Schmidhuber, 1997).

¿ Qué vamos a ver hoy?

- Deep Learning:
  - Neural Networks
  - <u>CNN:</u> Convolutional Neural Network (LeCun, 1989)
  - <u>RNN:</u> Recurrent Neural Network (Rumelhart et al., 1986a)
    - <u>LSTM:</u> Long Short-Term Memory (Hochreiter and Schmidhuber, 1997).

- Prophet (Taylor et al., 2018)

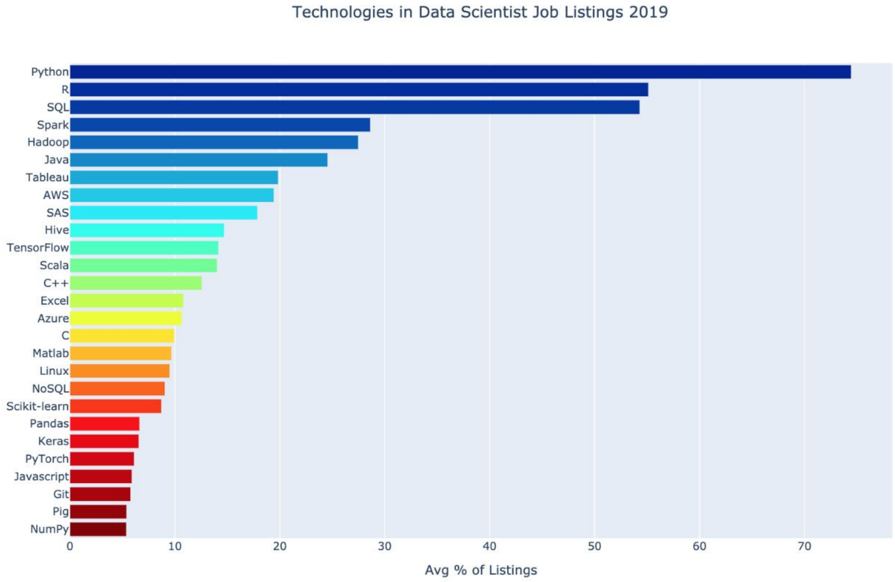
¿ Qué vamos a ver hoy?

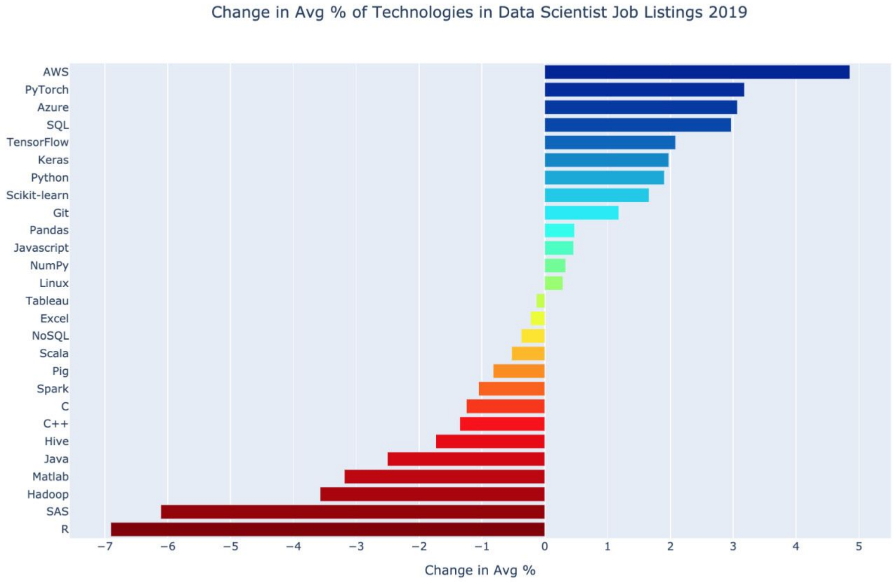
- Deep Learning:
    - Neural Networks
    - CNN: Convolutional Neural Network (LeCun, 1989)
    - RNN: Recurrent Neural Network (Rumelhart et al., 1986a)
        - LSTM: Long Short-Term Memory (Hochreiter and Schmidhuber, 1997).

- Prophet (Taylor et al., 2018)

¿ Qué herramientas vamos a usar?

PyTorch





Fuente: https://towardsdatascience.com/the-most-in-demand-tech-skills-for-data-scientists-d716d10c191d

¿ Qué vamos a ver hoy?

- Deep Learning:
  - Neural Networks
  - CNN: Convolutional Neural Network (LeCun, 1989)
  - RNN: Recurrent Neural Network (Rumelhart et al., 1986a)
    - LSTM: Long Short-Term Memory (Hochreiter and Schmidhuber, 1997).

- Prophet (Taylor et al., 2018)

¿ Qué herramientas vamos a usar?

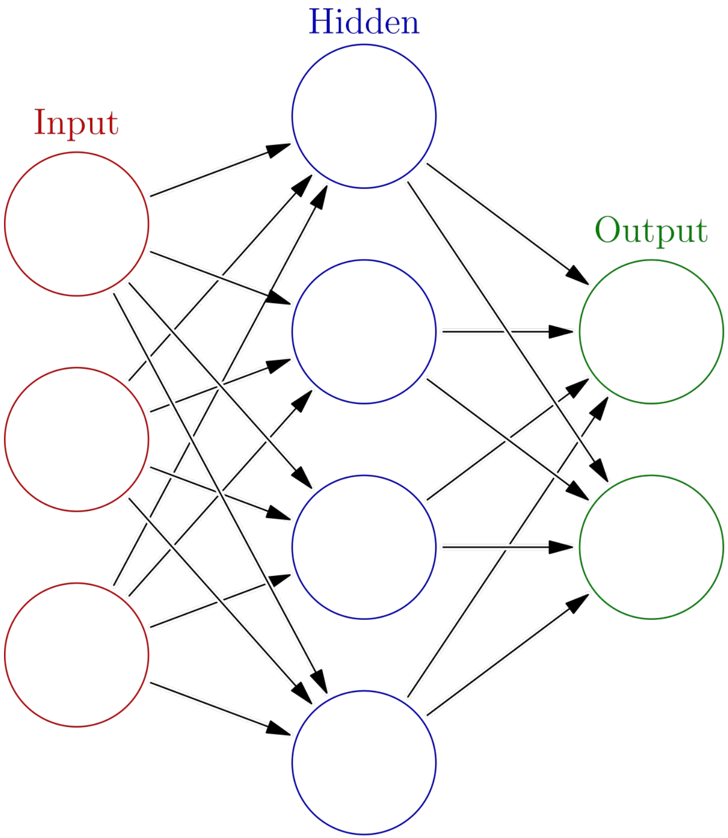
PyTorch


Does everyone understand?
(silent)
Good, I'll move on then


ONE DOES NOT SIMPLY
USE MEMES IN THE CLASSROOM

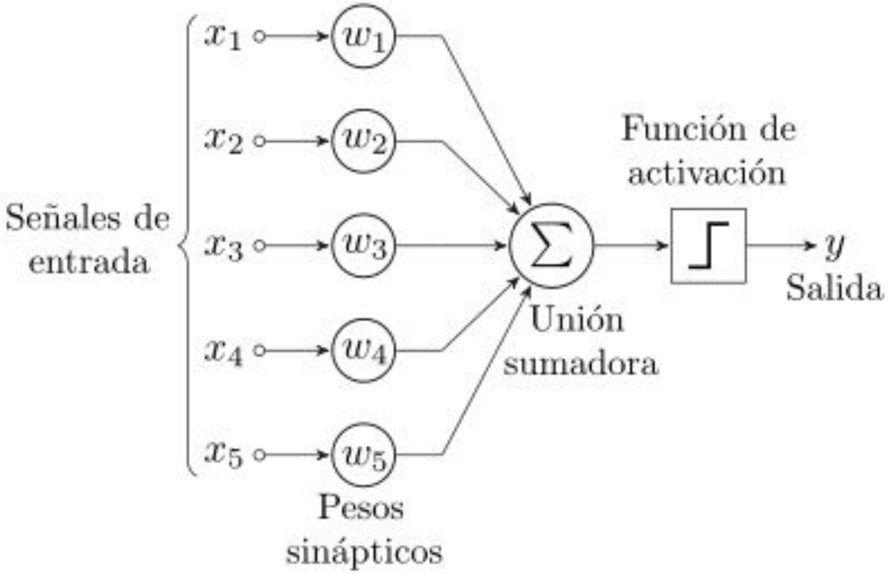
Neural Networks

Red Neuronal

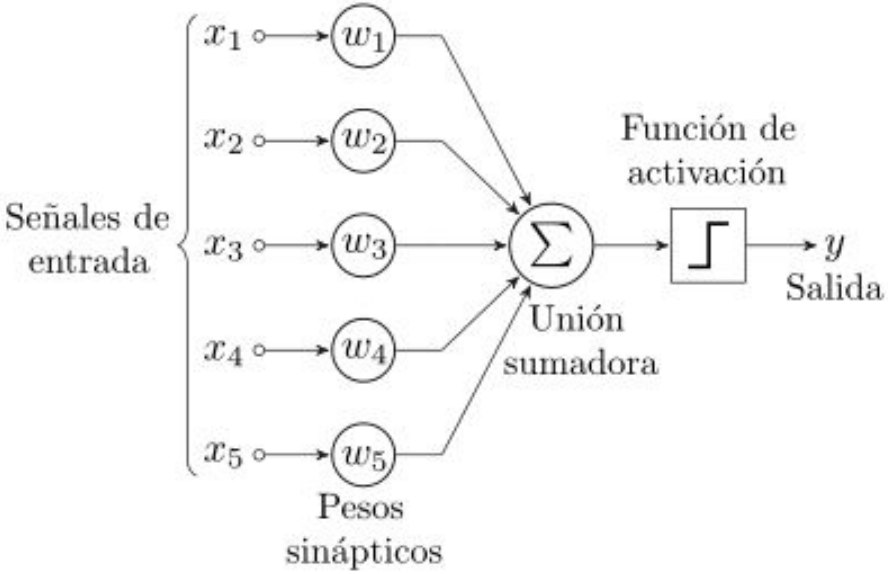


“Las **redes neuronales** son un modelo computacional. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida.” (wikipedia)

Red Neuronal: perceptrón simple

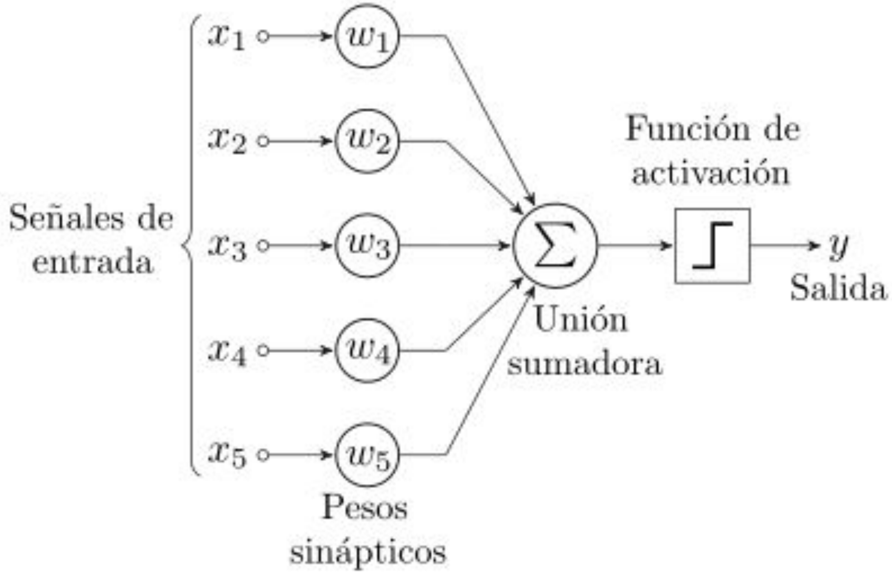


Red Neuronal: perceptrón simple

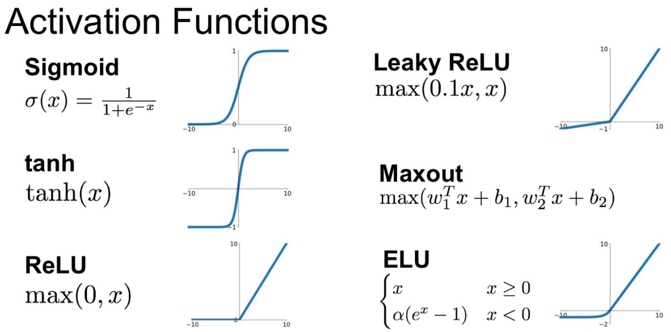


$$Y = \sigma(\Sigma W_i * X_i + b)$$

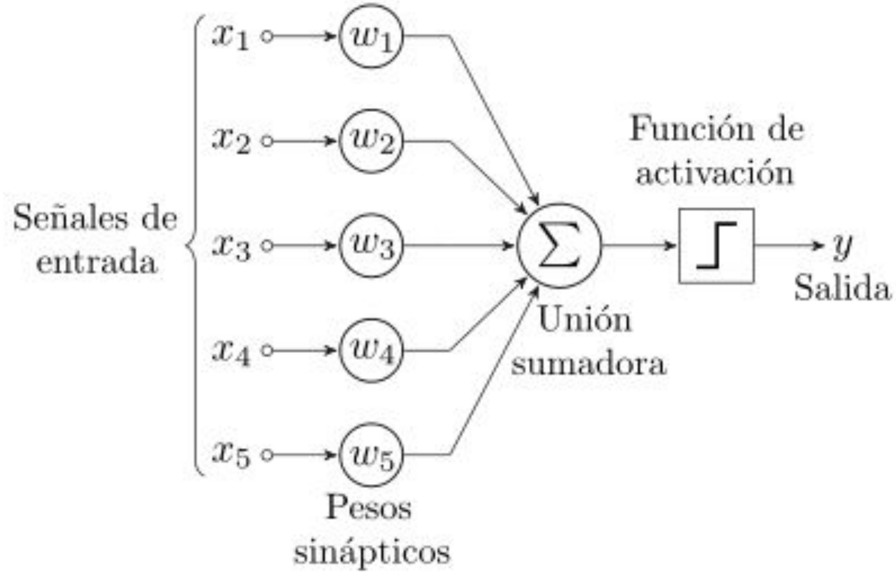
Red Neuronal: perceptrón simple



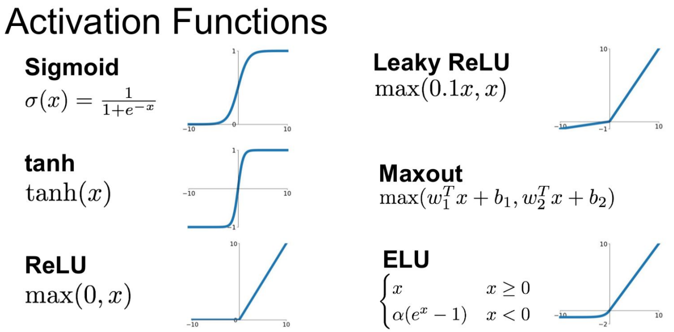
$$Y = \sigma(\Sigma W_i * X_i + b)$$



Red Neuronal: perceptrón simple

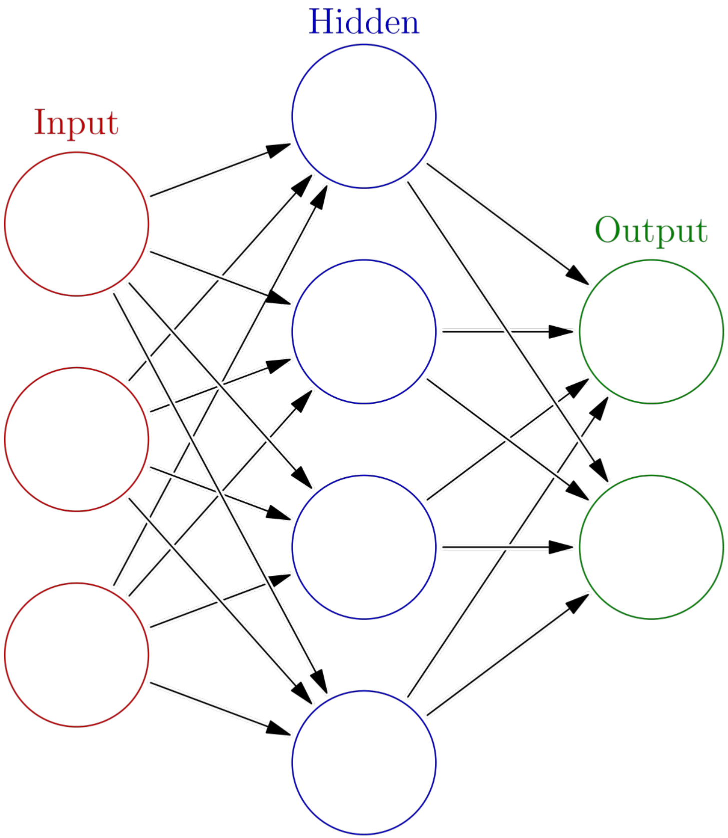


$$Y = \sigma(\Sigma W_i * X_i + b)$$



¿Cuánto valen los pesos y los bias?

Red Neuronal

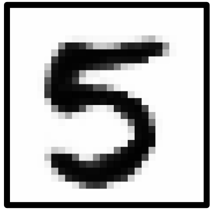


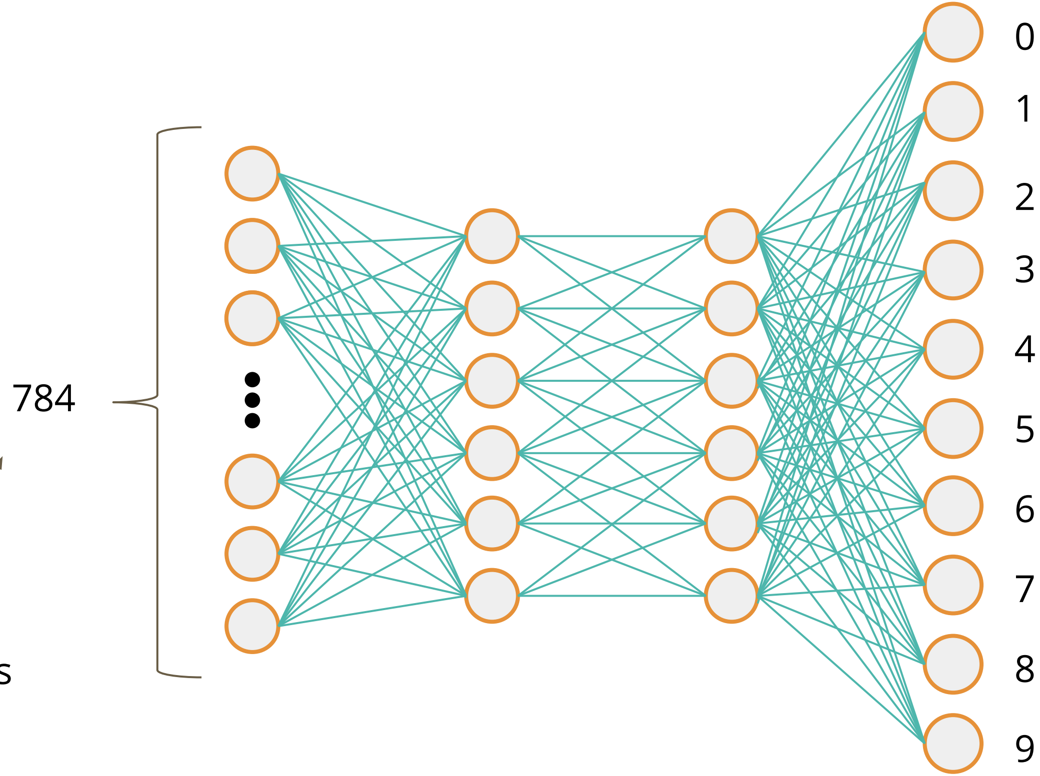
Una Red Neuronal solo son muchos perceptrones simples conectados entre sí.

Red Neuronal

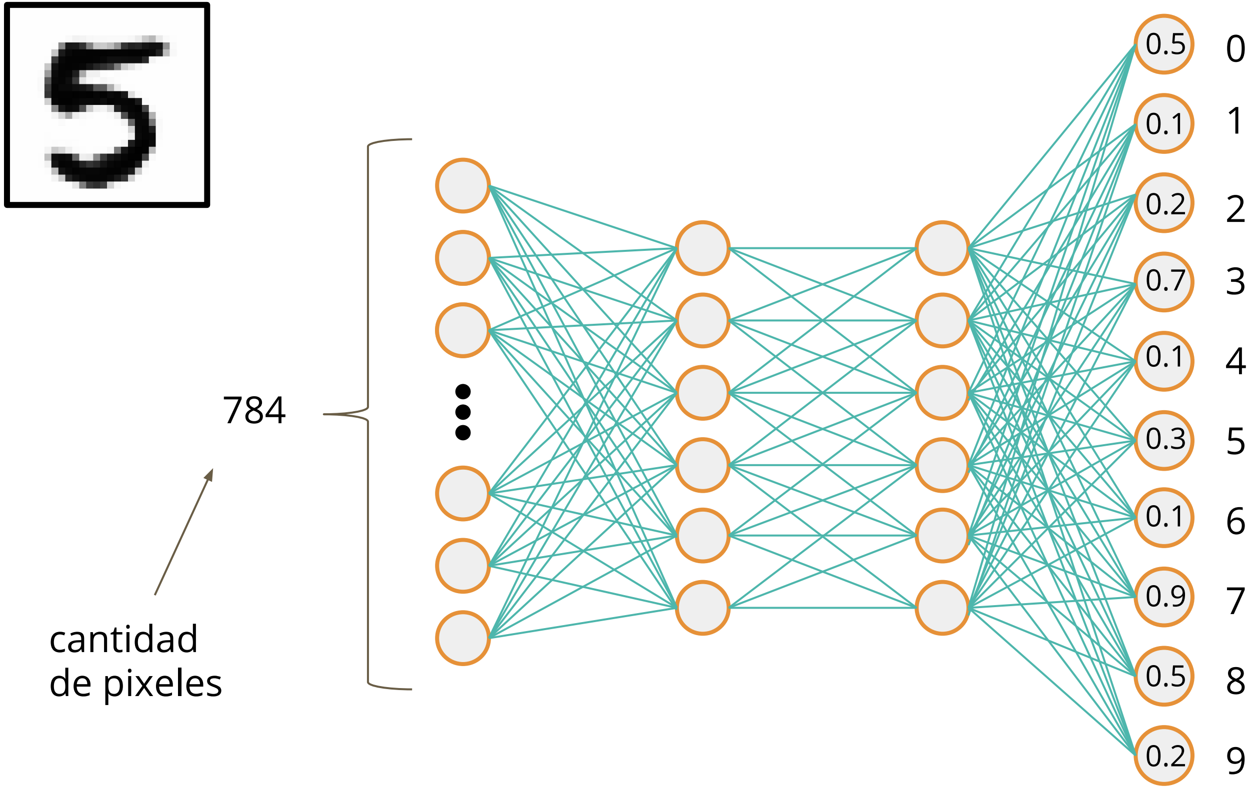
Ejemplo: mnist

Red Neuronal: Backpropagation o cómo calcular w y b

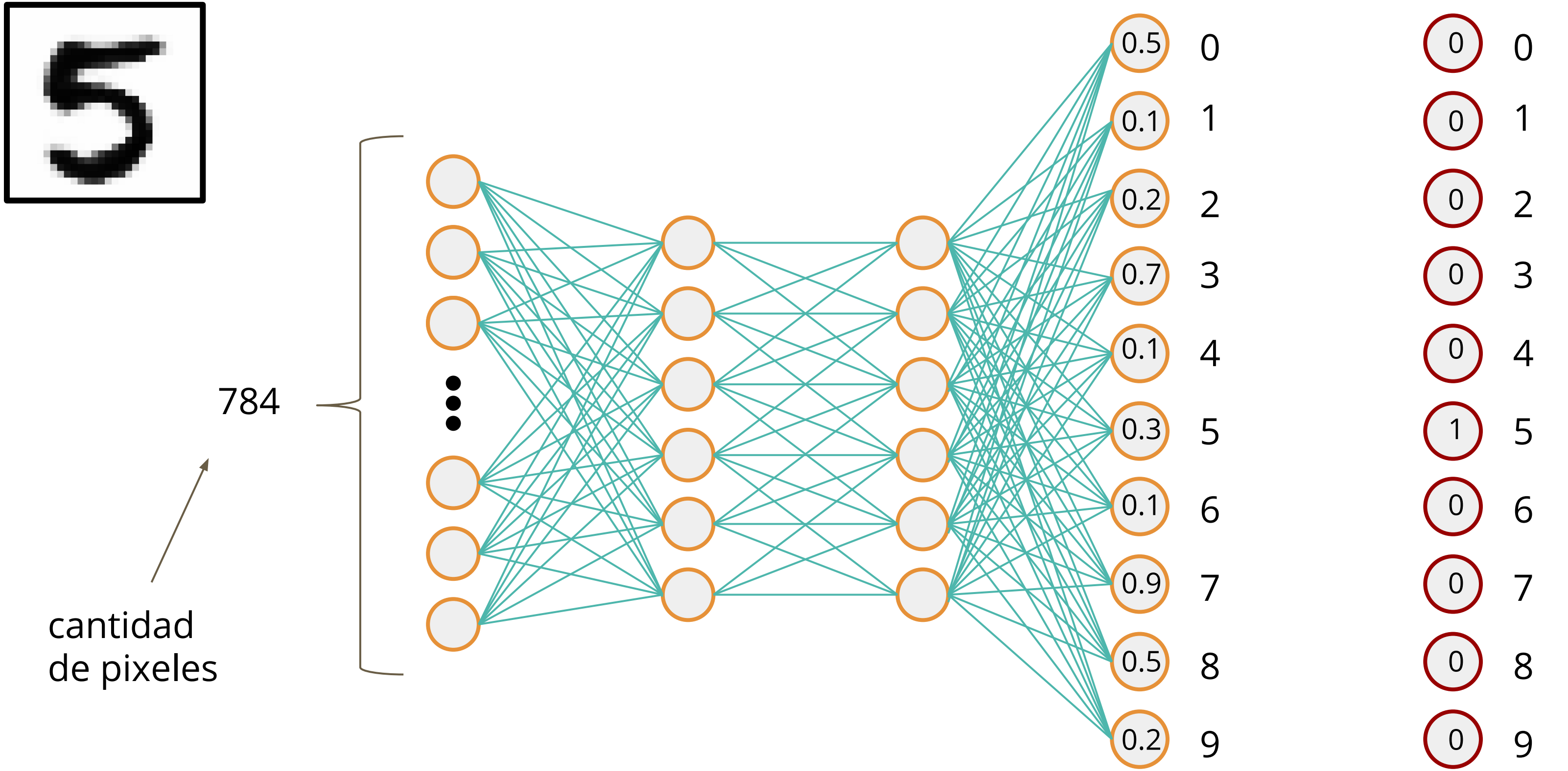




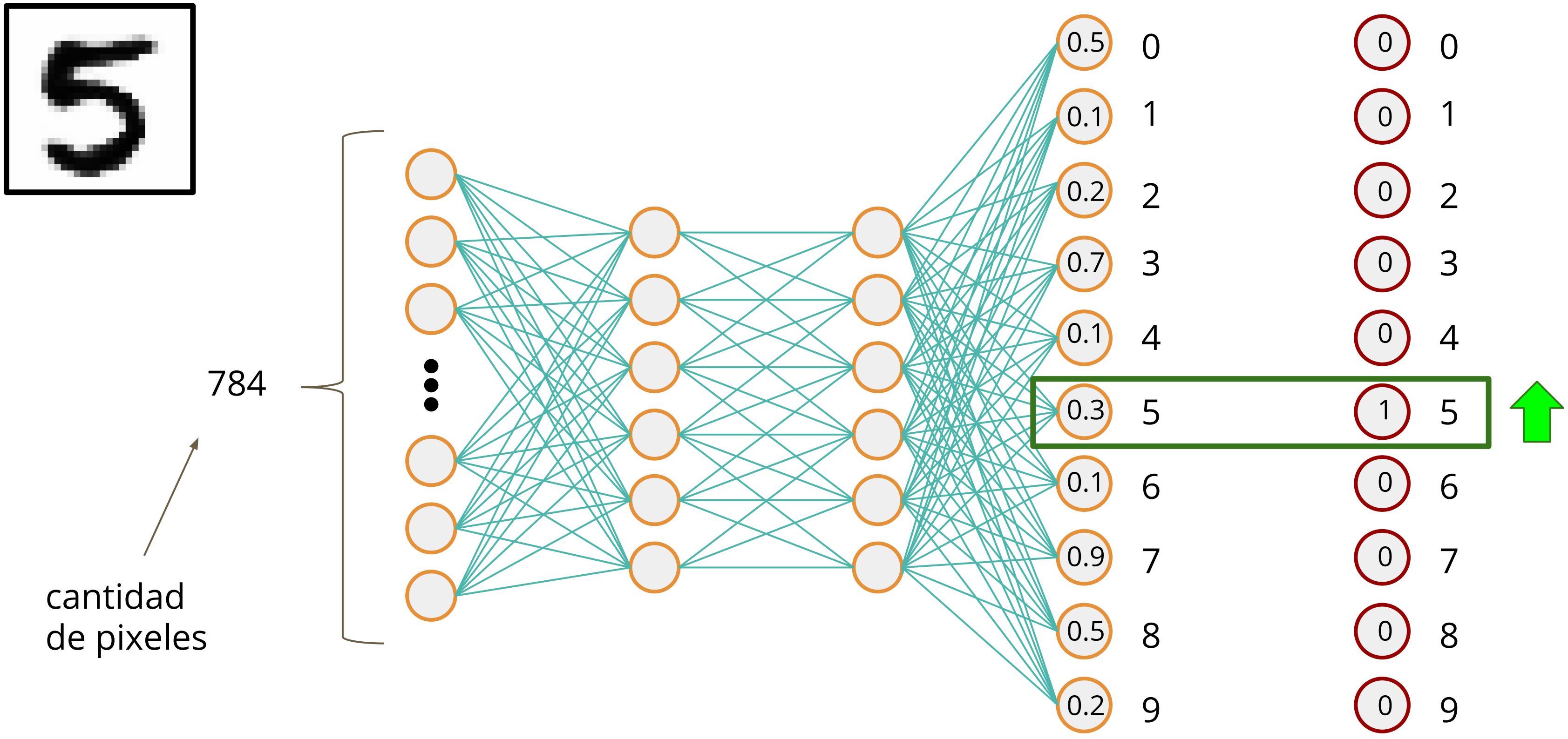
Red Neuronal: Backpropagation o cómo calcular w y b



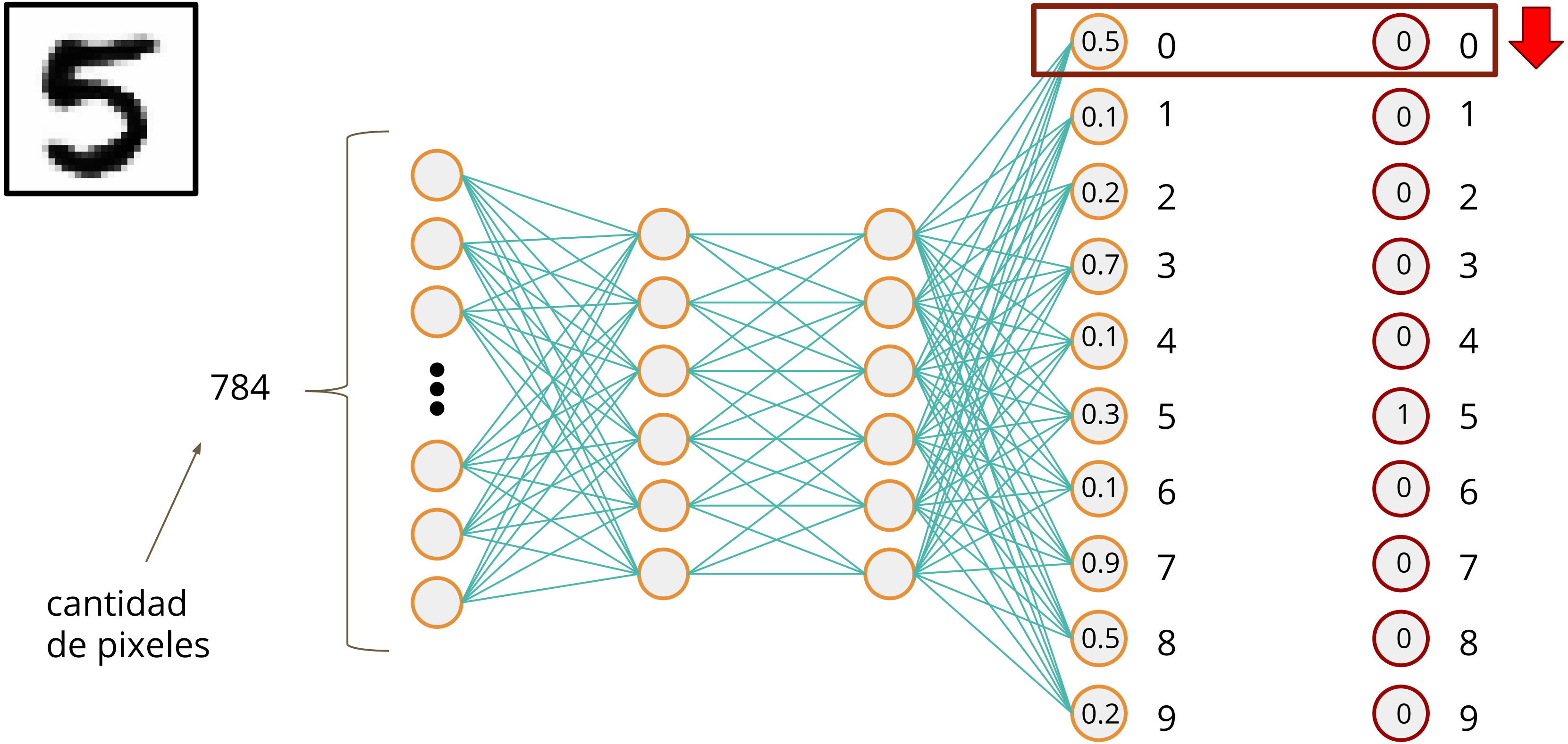
Red Neuronal: Backpropagation o cómo calcular w y b



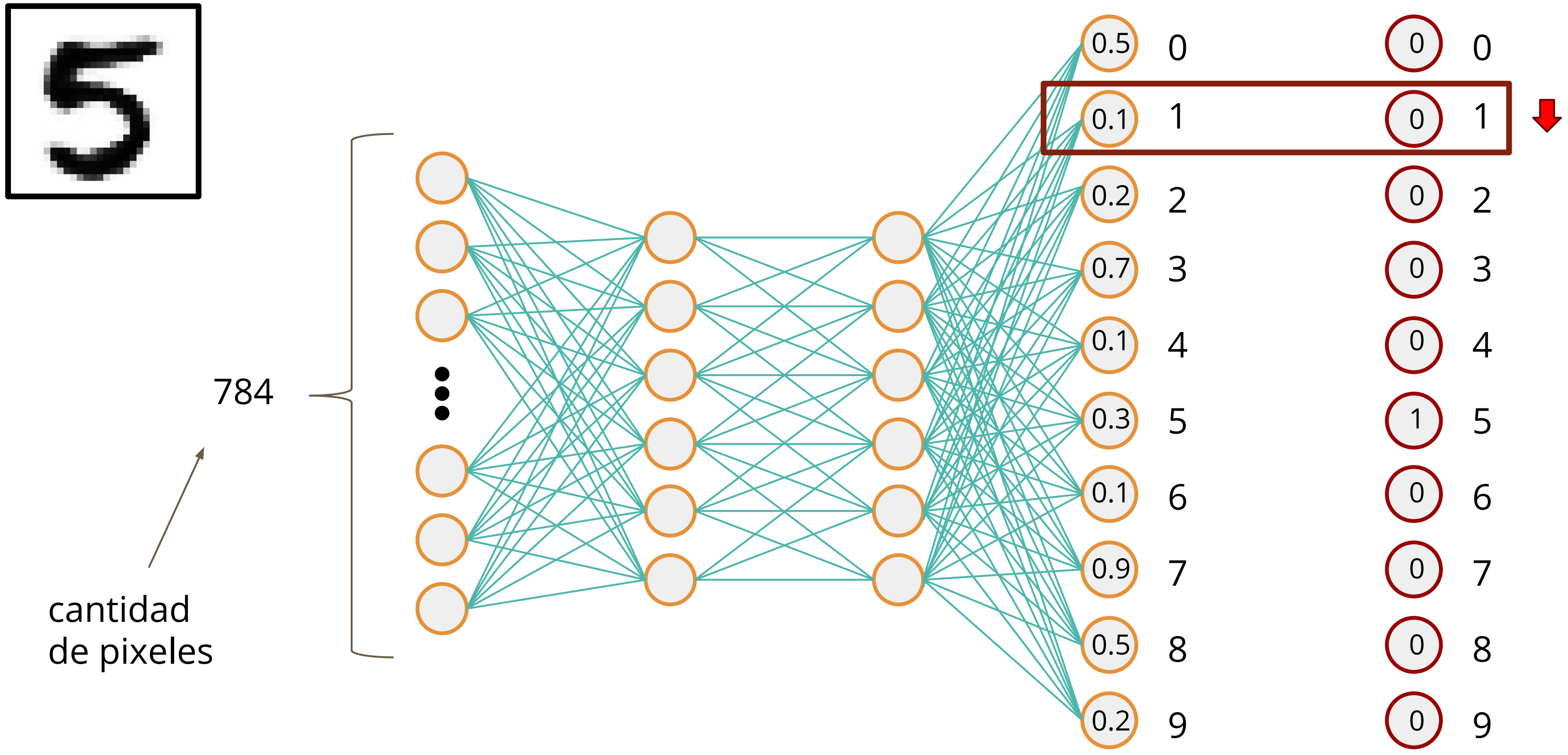
Red Neuronal: Backpropagation o cómo calcular w y b



Red Neuronal: Backpropagation o cómo calcular w y b



Red Neuronal: Backpropagation o cómo calcular w y b



Red Neuronal: Backpropagation

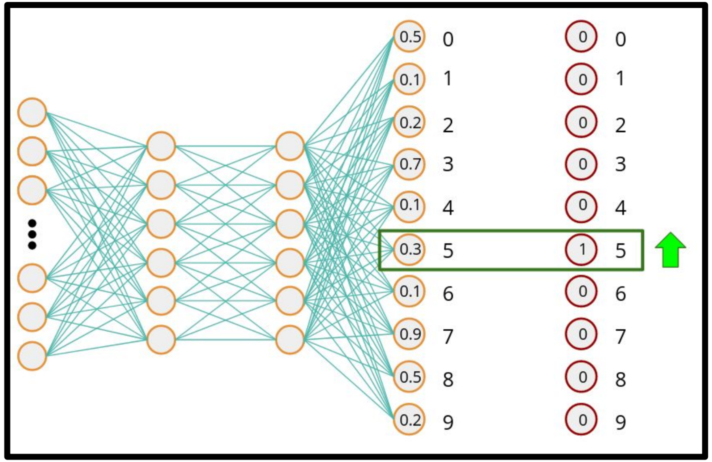
Se define una función de costo que determina mi error a partir del valor predicho y el valor real.
Por ejemplo la función de pérdida cuadrática:

$$Cost = (f(x) - Y)^2$$

Donde el valor predicho f(x) es:

$$f(x) = \sigma(z^l)$$

$$z^l = \sum_i w_i^l * a_i^{l-1} + b^l$$



Donde a_i^{l-1} corresponde al valor que toma la neurona i de la capa l-1.

Red Neuronal: Backpropagation

Para saber cuánto tengo que cambiar los valores, calculo cómo variaciones en w modifican la función de costo:

$$\frac{\partial C_0}{\partial w^l} = \frac{\partial z^l}{\partial w^l} * \frac{\partial a^l}{\partial z^l} * \frac{\partial C_0}{\partial a^l} = a^{l-1} * \sigma'(z^l) * 2(a^l - y)$$

Luego:

$$\frac{\partial C}{\partial w^l} = \frac{1}{n} \sum_{k=0}^{n-1} \frac{\partial C_k}{\partial w^l}$$

Equivalente para el bias.

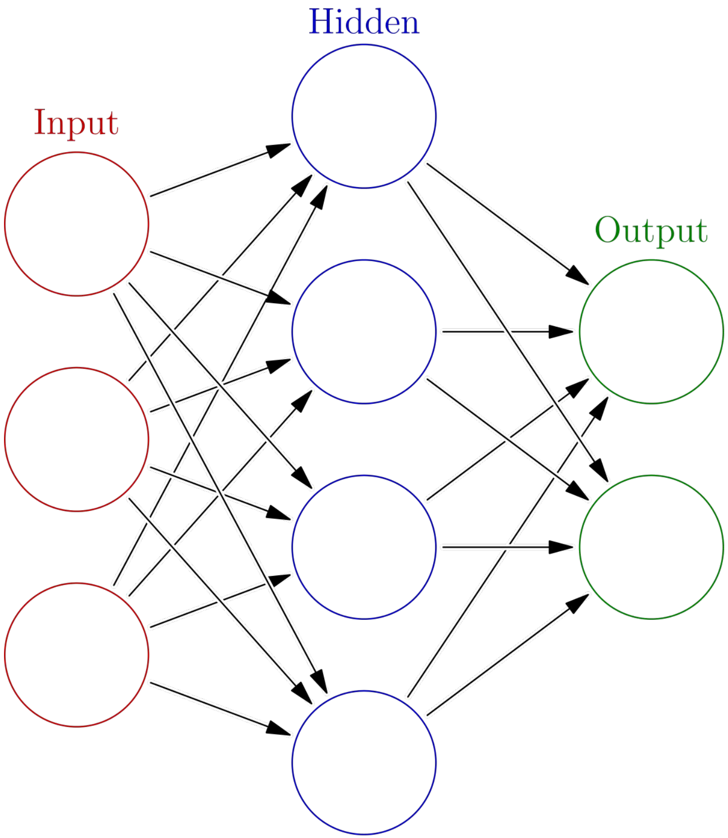
Red Neuronal: Backpropagation

Finalmente, para actualizar los valores de w hago:

$$W_i^+ = W_i - \eta * \frac{\partial C}{\partial w_i}$$

donde η es el “learning rate”.

Red Neuronal

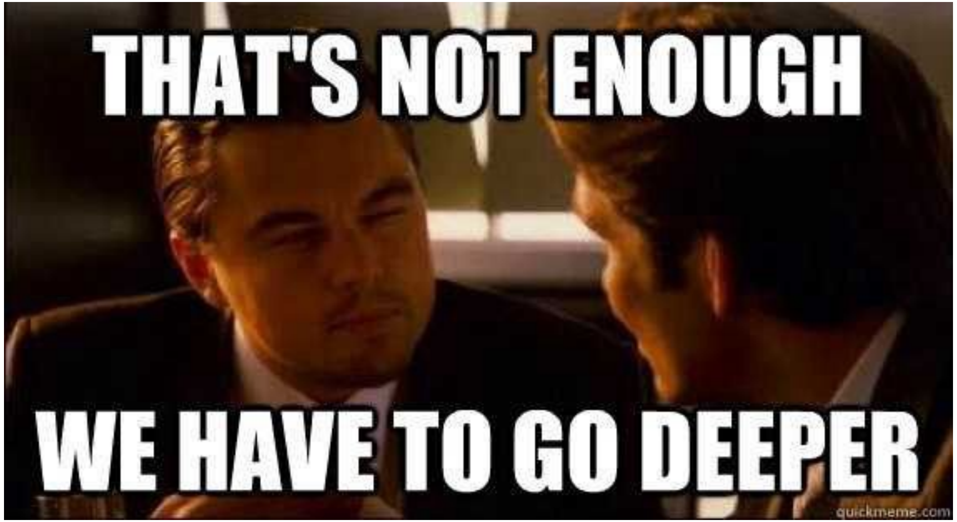


Un perceptrón simple es equivalente a una regresión logística.

[Link]

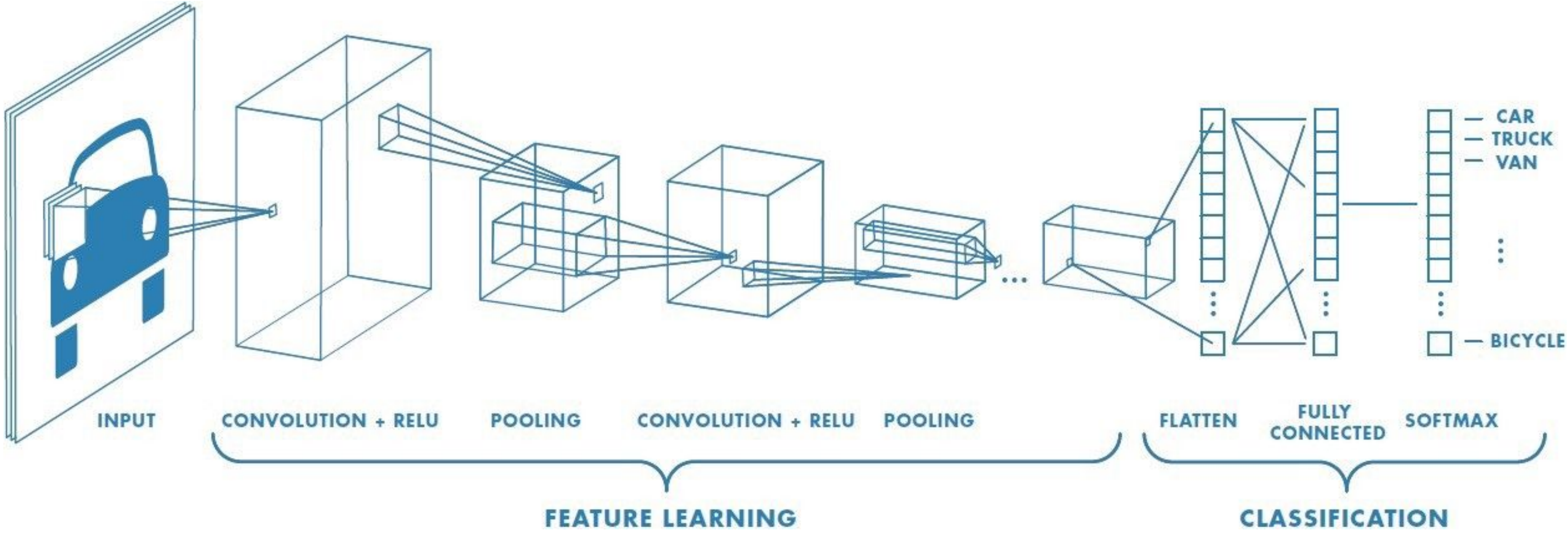
FIN ?

FIN ?



Convolutional Neural Networks

Convolutional Neural Networks: Estructura

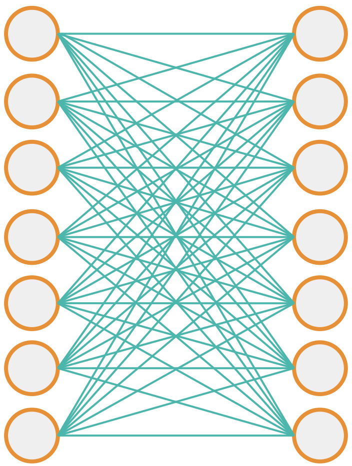


fuente: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolutional Neural Networks:

¿Qué son las CNN? Parte1: Convolución

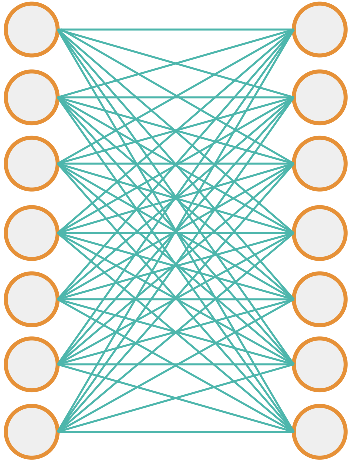
Antes (capa densa)



Convolutional Neural Networks:

¿Qué son las CNN? Parte1: Convolución

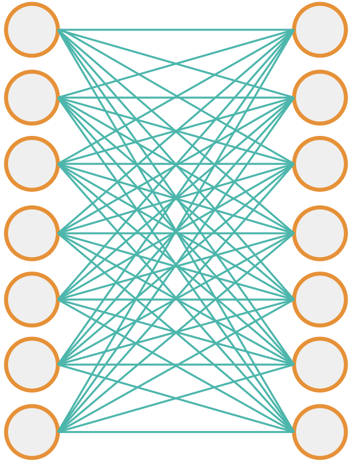
<u>Antes (capa densa)</u>



Convolutional Neural Networks:

¿Qué son las CNN? Parte1: Convolución

Antes (capa densa)

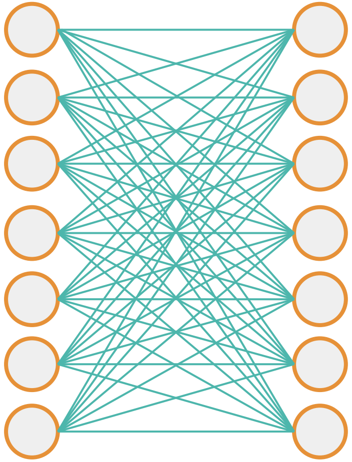


¡Muchos parámetros!

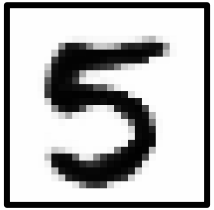
Convolutional Neural Networks:

¿Qué son las CNN? Parte1: Convolución

Antes (capa densa)



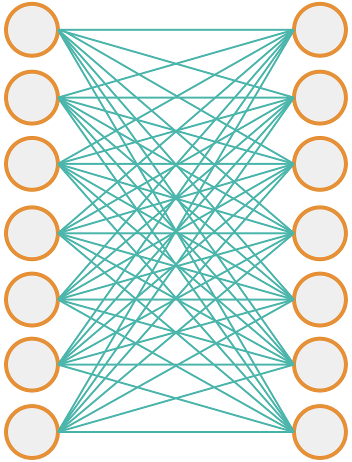
¡Muchos parámetros!



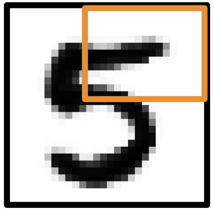
Convolutional Neural Networks:

¿Qué son las CNN? Parte1: Convolución

<u>Antes (capa densa)</u>



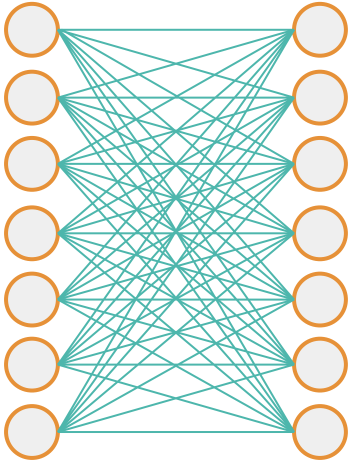
¡Muchos parámetros!

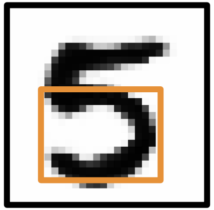


Convolutional Neural Networks:

¿Qué son las CNN? Parte1: Convolución

<u>Antes (capa densa)</u>



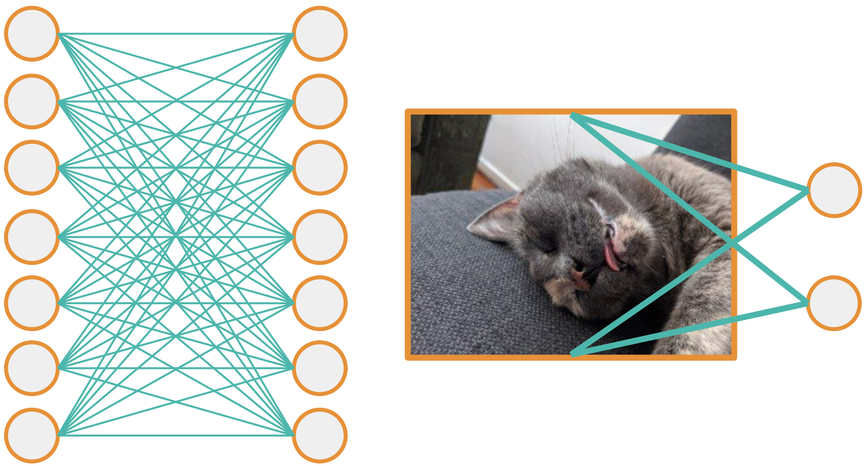


¡Muchos parámetros!

Convolutional Neural Networks:

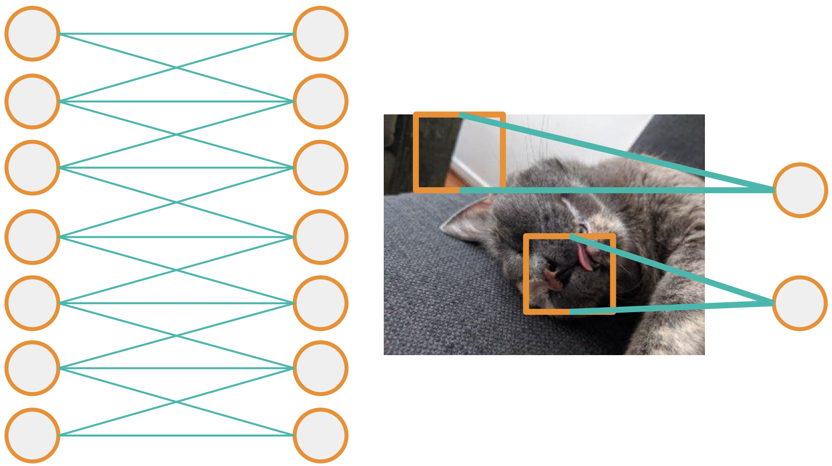
¿Qué son las CNN? Parte1: Convolución

Antes (capa densa)



¡Muchos parámetros!

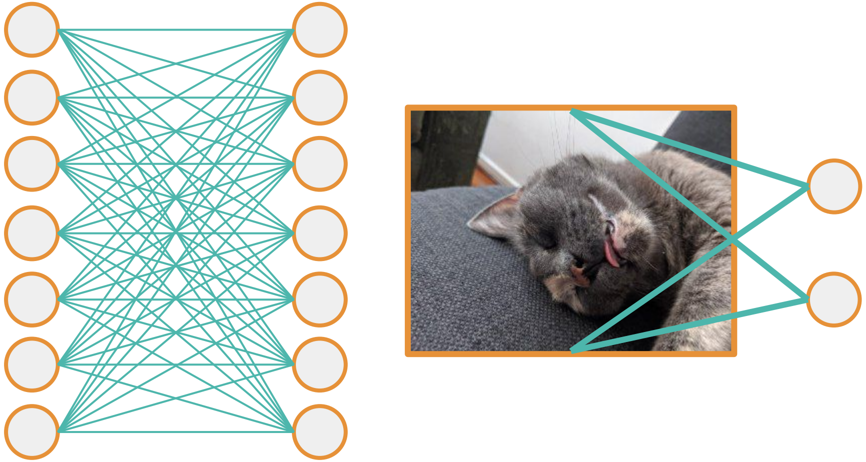
Convolución



Convolutional Neural Networks:

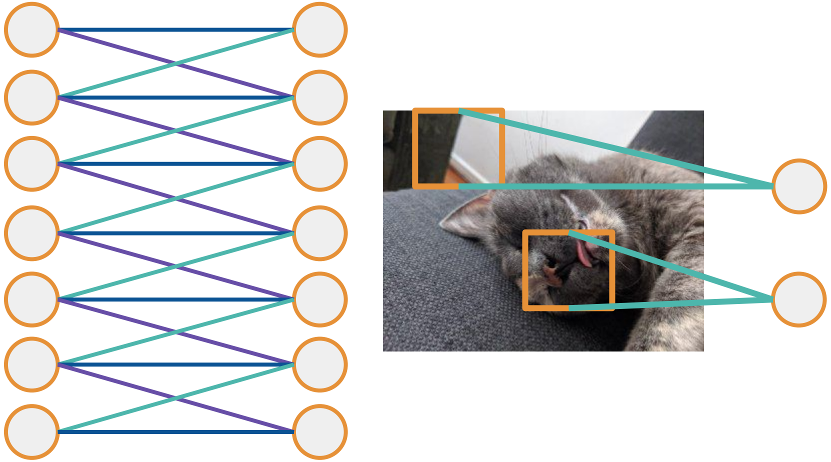
¿Qué son las CNN? Parte1: Convolución

Antes (capa densa)



¡Muchos parámetros!

Convolución

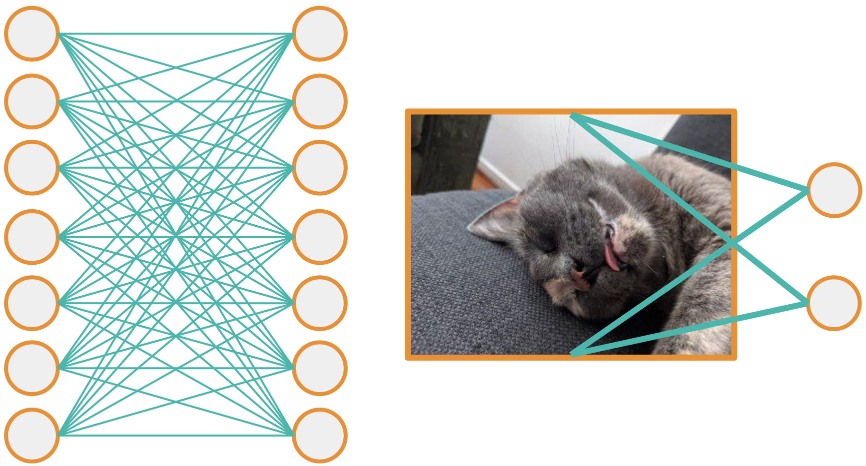


Menos parámetros → Generaliza mejor

Convolutional Neural Networks:

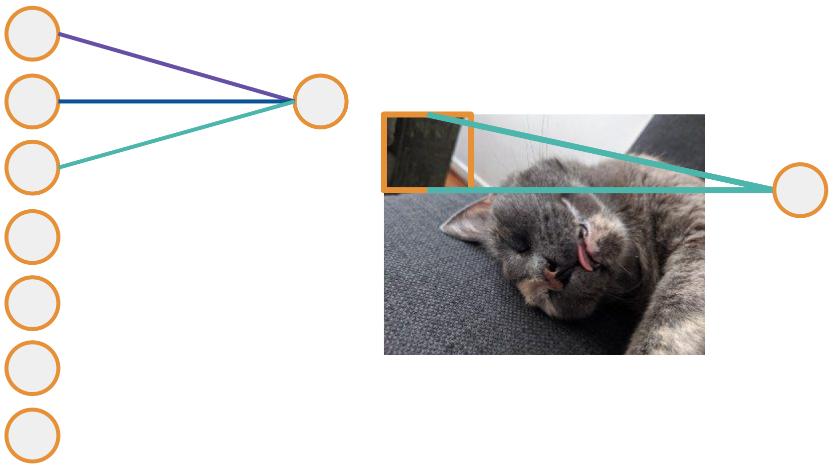
¿Qué son las CNN? Parte1: Convolución

Antes (capa densa)



¡Muchos parámetros!

Convolución

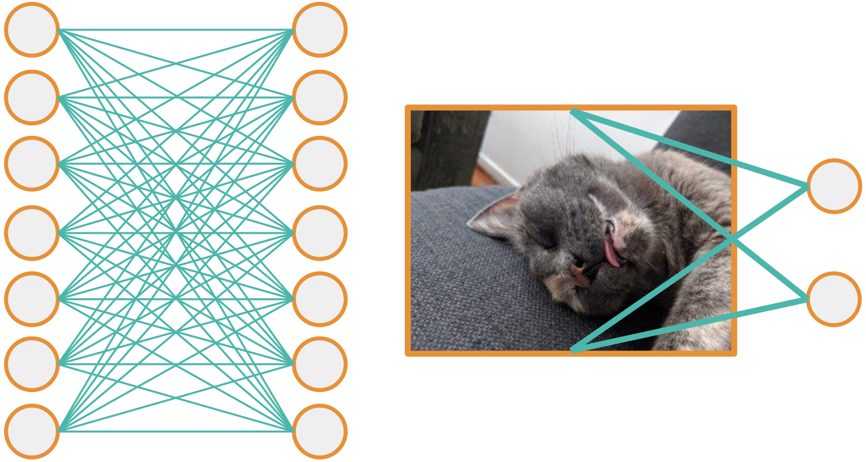


Menos parámetros → Generaliza mejor

Convolutional Neural Networks:

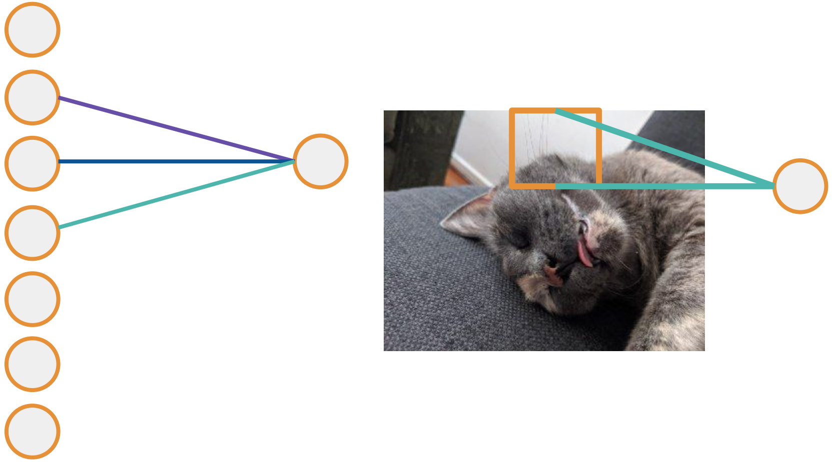
¿Qué son las CNN? Parte1: Convolución

Antes (capa densa)



¡Muchos parámetros!

Convolución

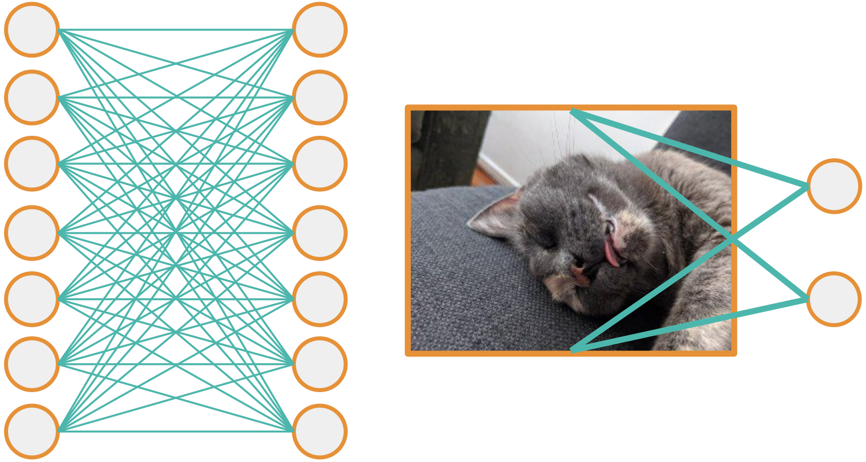


Menos parámetros → Generaliza mejor

Convolutional Neural Networks:

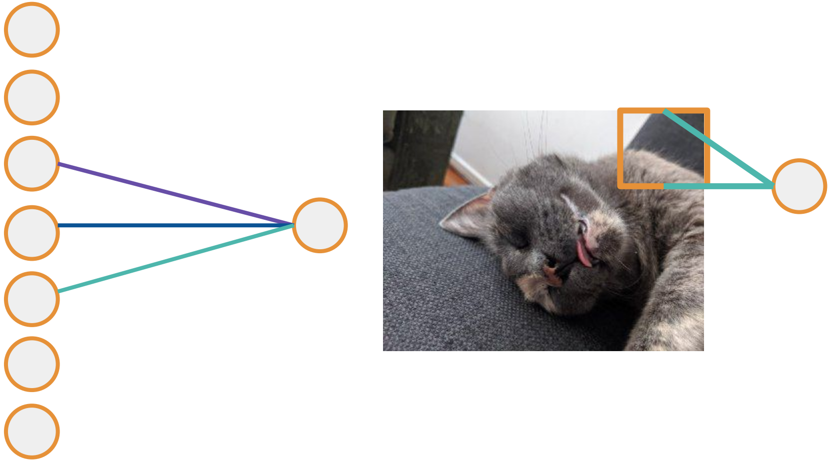
¿Qué son las CNN? Parte1: Convolución

<u>Antes (capa densa)</u>



¡Muchos parámetros!

<u>Convolución</u>

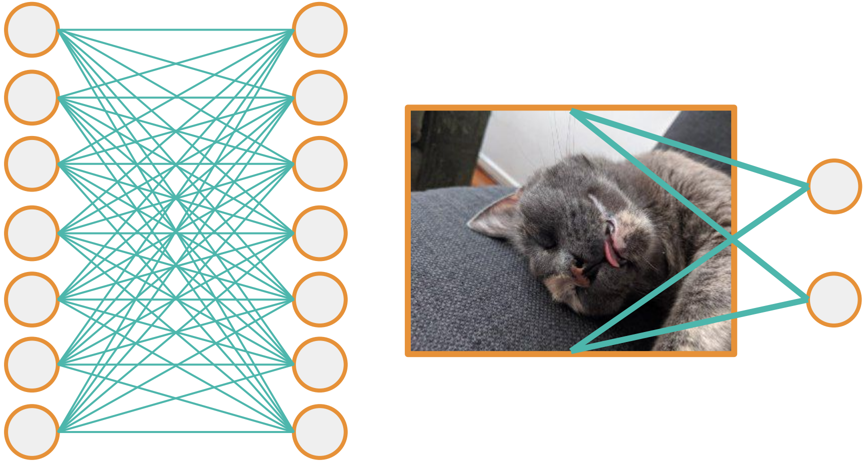


Menos parámetros → Generaliza mejor

Convolutional Neural Networks:

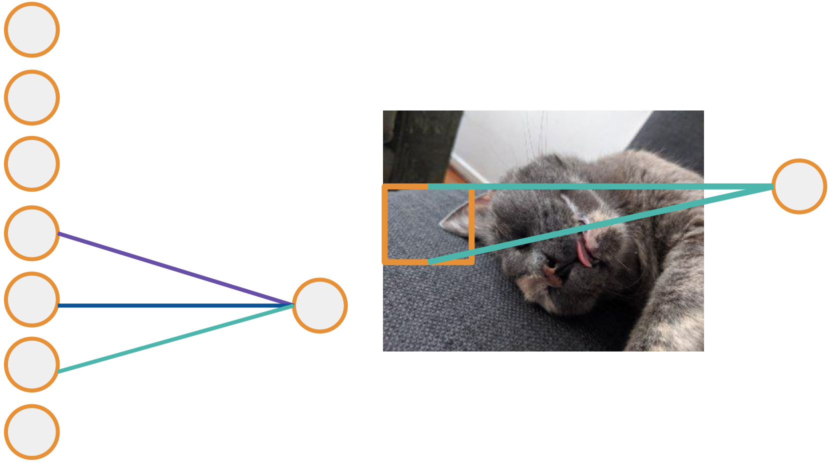
¿Qué son las CNN? Parte1: Convolución

Antes (capa densa)



¡Muchos parámetros!

Convolución

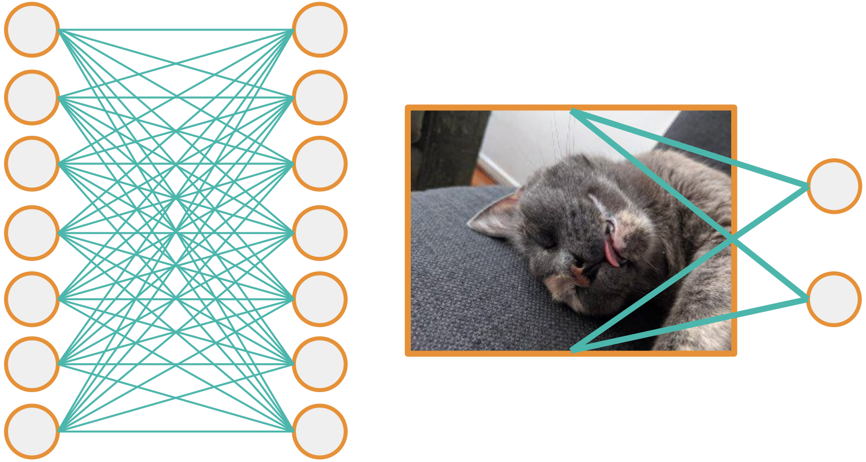


Menos parámetros → Generaliza mejor

Convolutional Neural Networks:

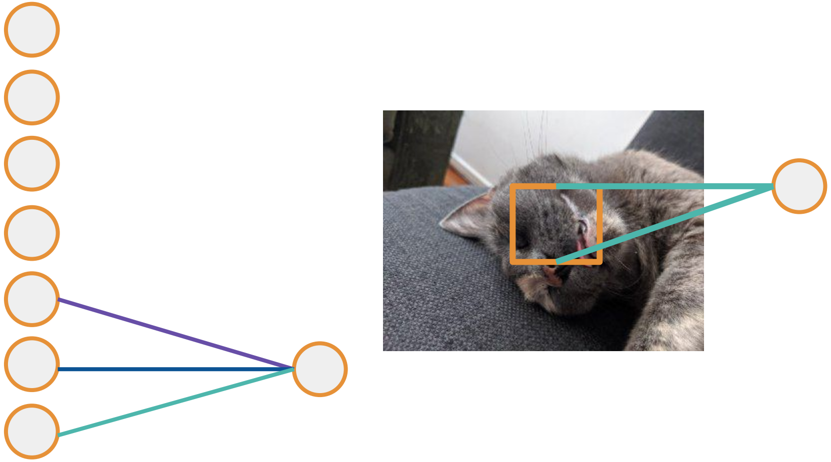
¿Qué son las CNN? Parte1: Convolución

<u>Antes (capa densa)</u>



¡Muchos parámetros!

<u>Convolución</u>

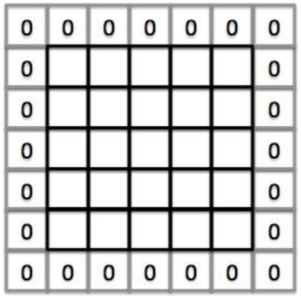


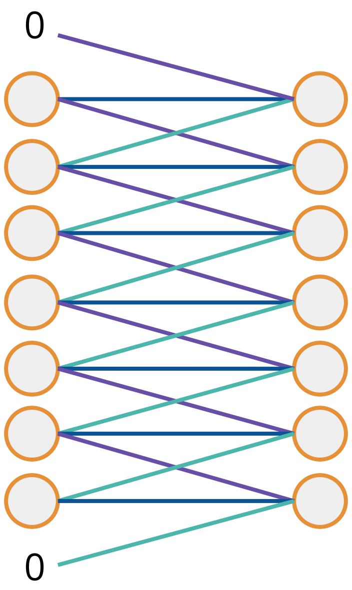
Menos parámetros → Generaliza mejor

Convolutional Neural Networks:

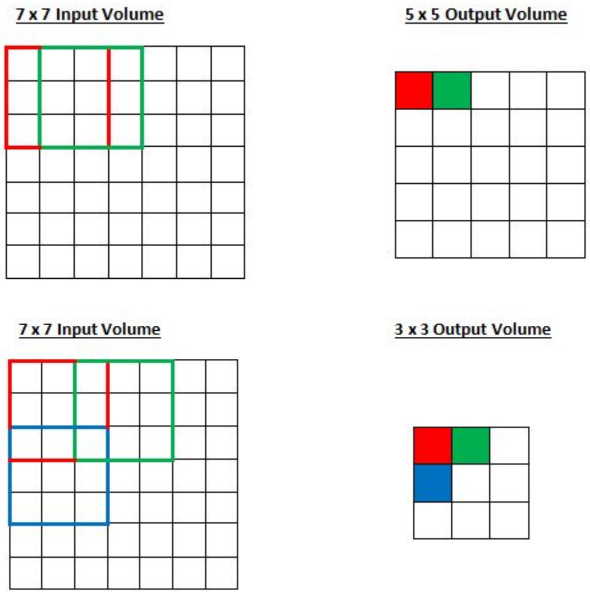
¿Qué son las CNN? Parte1: Convolución

Padding: Agregar a los costados algo (generalmente ceros).





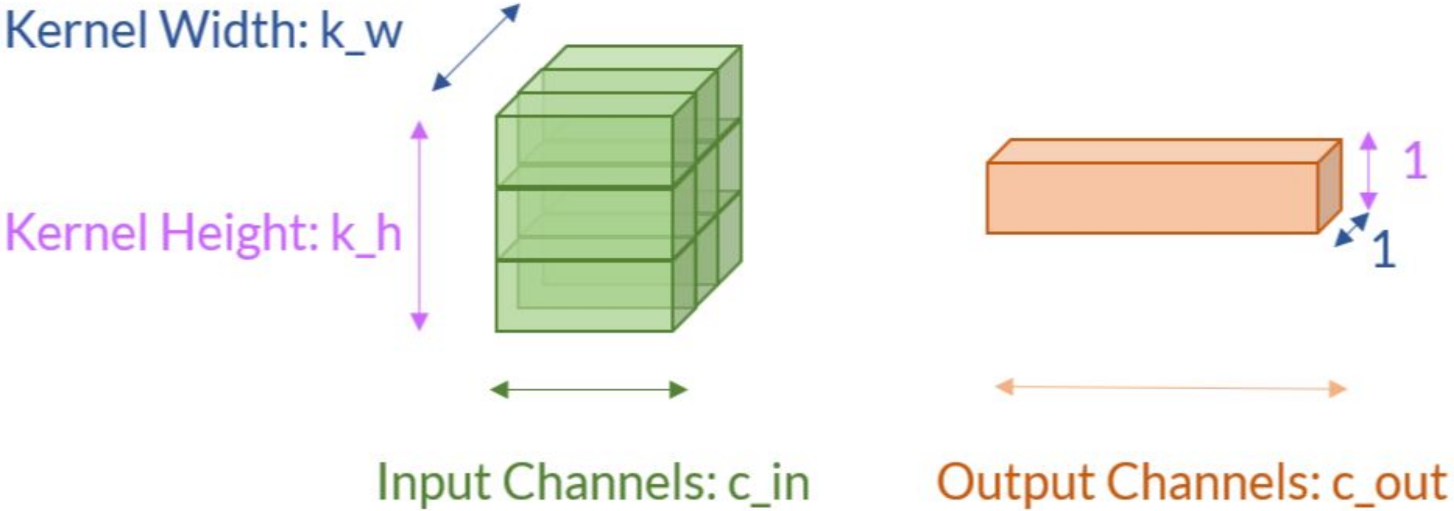
Stride: El número de pixels que se mueve la convolucional sobre la imagen..



Convolutional Neural Networks:

¿Qué son las CNN? Parte1: Convolución

Kernel, Input Channels y Output Channels:



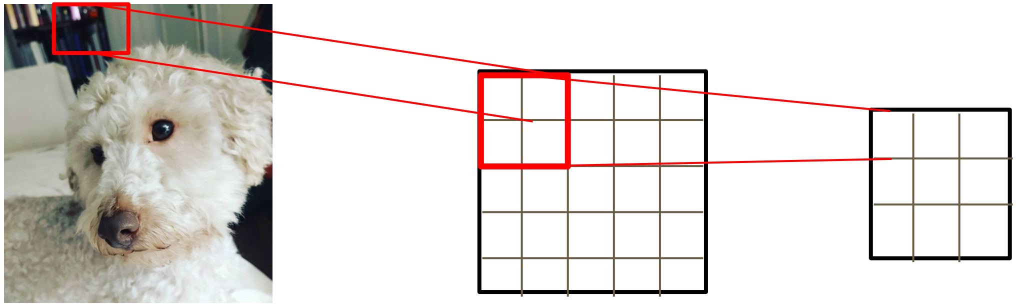
Convolutional Neural Networks:

¿Qué son las CNN? Parte 2: Max Pooling

1	4	2	7
2	6	8	5
3	4	0	7
1	2	3	1

max pooling con filtros 2x2 y “stride” 2

6	8
4	7



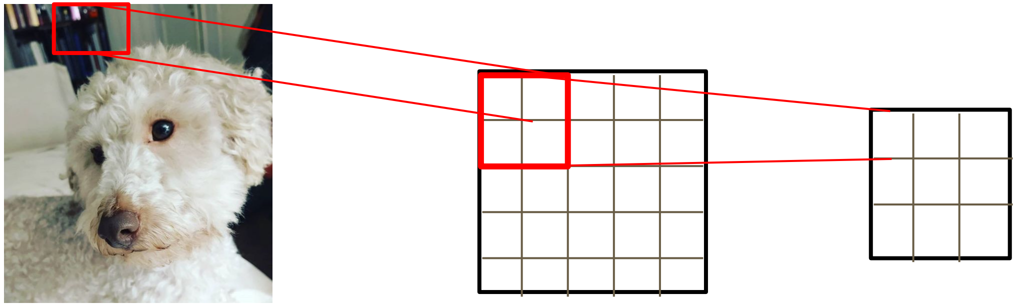
Convolutional Neural Networks:

¿Qué son las CNN? Parte 2: Max Pooling

1	4	2	7
2	6	8	5
3	4	0	7
1	2	3	1

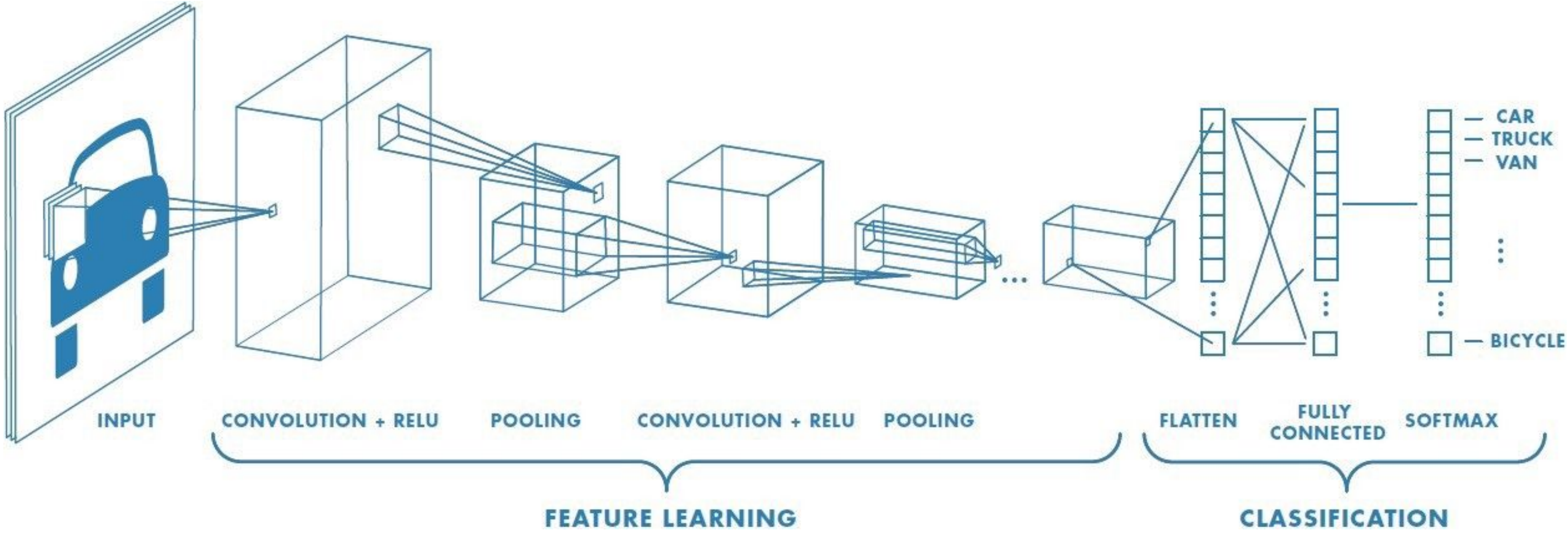
max pooling con filtros 2x2 y “stride” 2

6	8
4	7



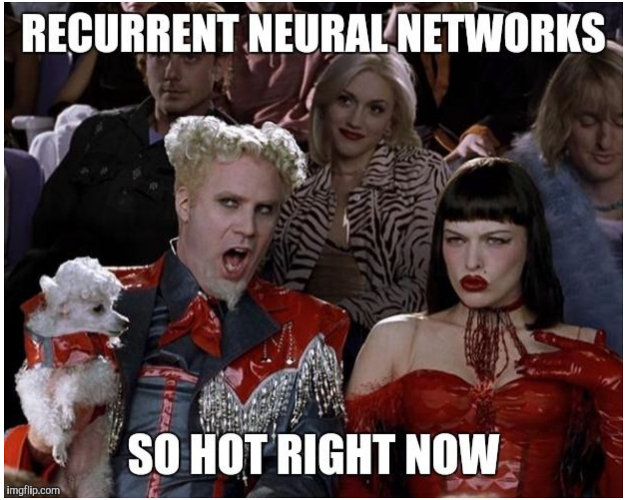
No necesariamente se tiene que quedar con el máximo, sino puede ser con el **promedio** u otras cosas más complicadas.

Convolutional Neural Networks:



fuente: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

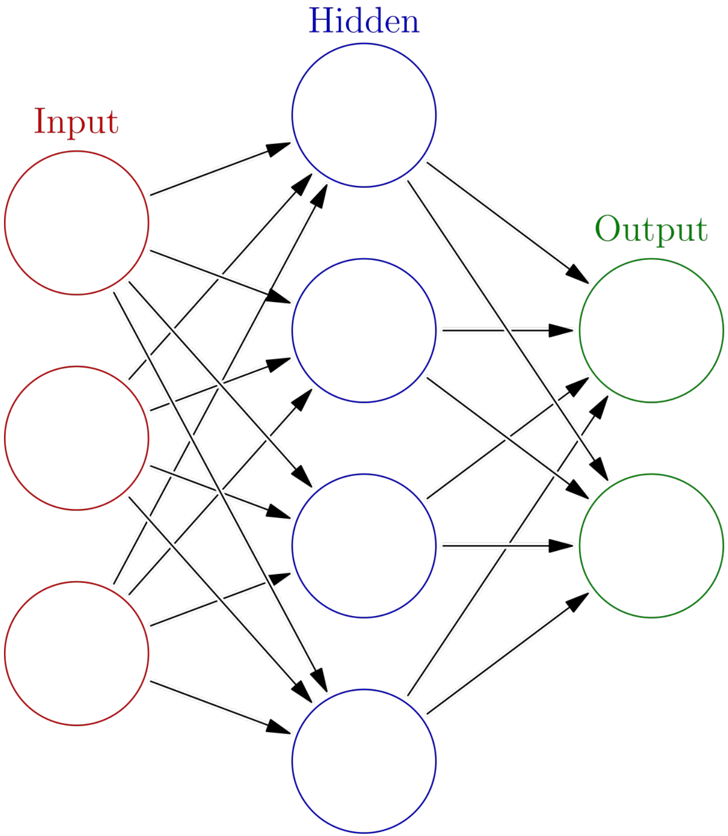
Recurrent Neural Networks


RECURRENT NEURAL NETWORKS
SO HOT RIGHT NOW
imgflip.com

Recurrent Neural Networks

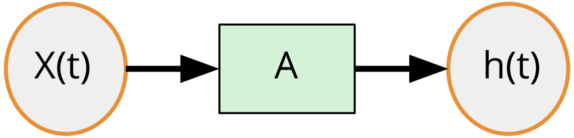
Antes

$$h(t) = \sigma(W \cdot x(t) + b)$$



Recurrent Neural Networks

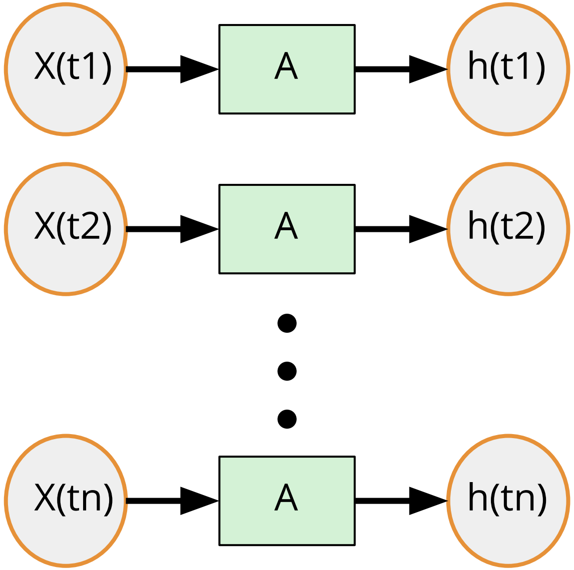
Antes



$$h(t) = \sigma(W \cdot x(t) + b)$$

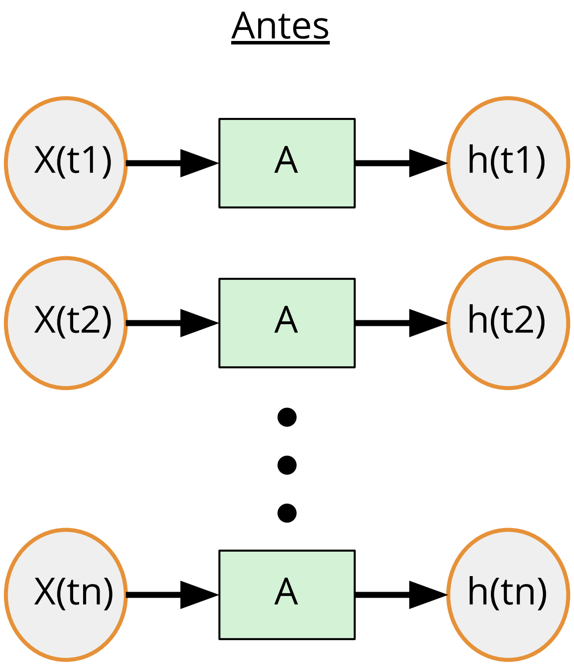
Recurrent Neural Networks

Antes

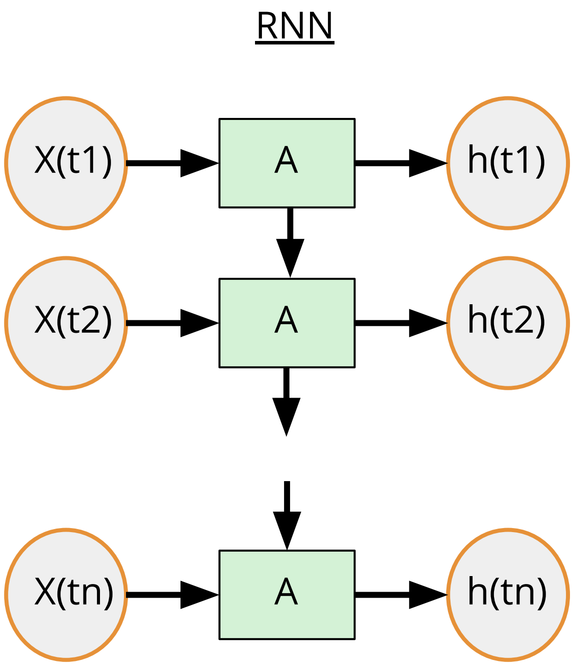


$$h(t) = \sigma(W \cdot x(t) + b)$$

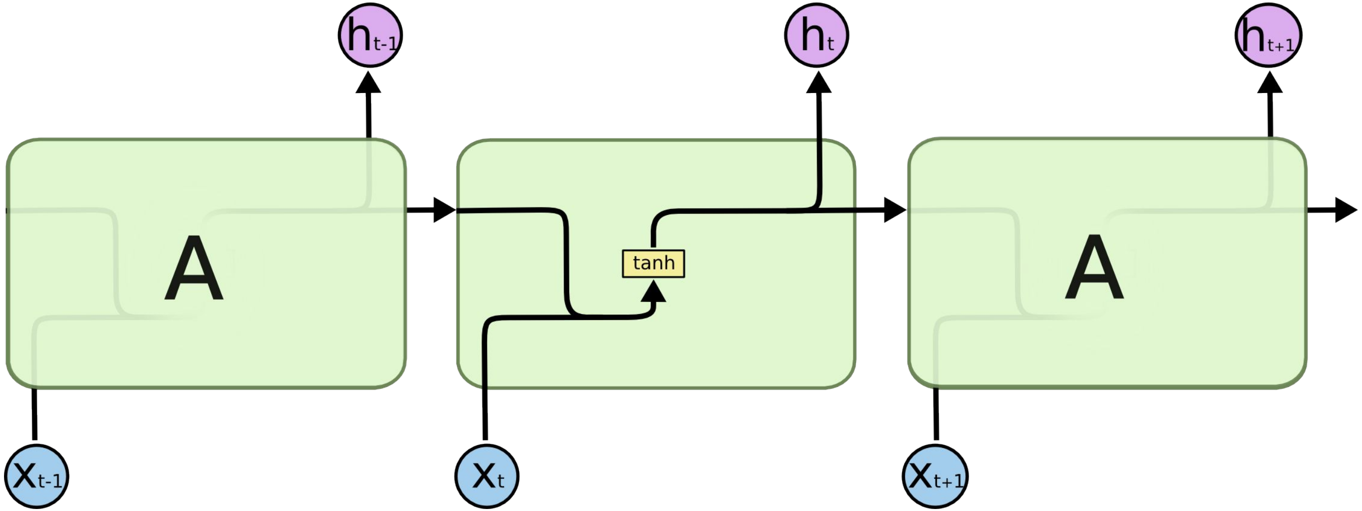
Recurrent Neural Networks

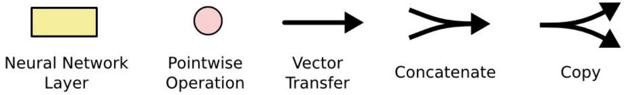


$$h(t) = \sigma(W \cdot x(t) + b)$$



Recurrent Neural Networks

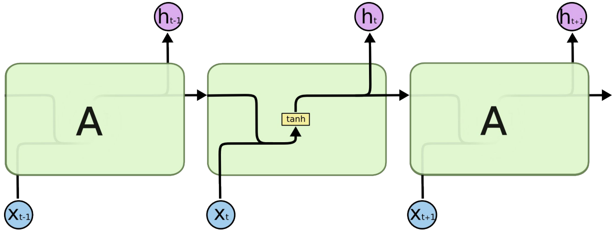




http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Recurrent Neural Networks

Problemas con este simple approach:

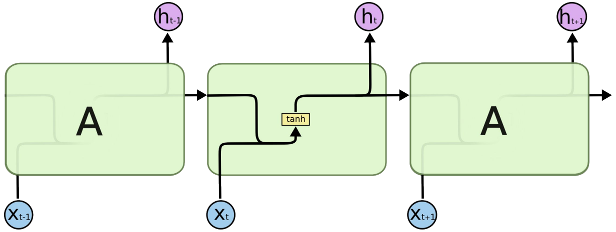


Recurrent Neural Networks

Problemas con este simple approach:

- Vanishing Gradients

- Exploding Gradients



Recurrent Neural Networks

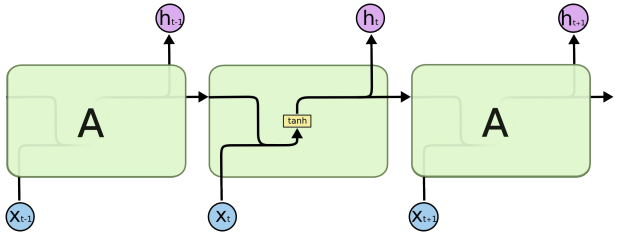
Problemas con este simple approach:

- Vanishing Gradients

$$0.99^{1000} = 0.000043...$$

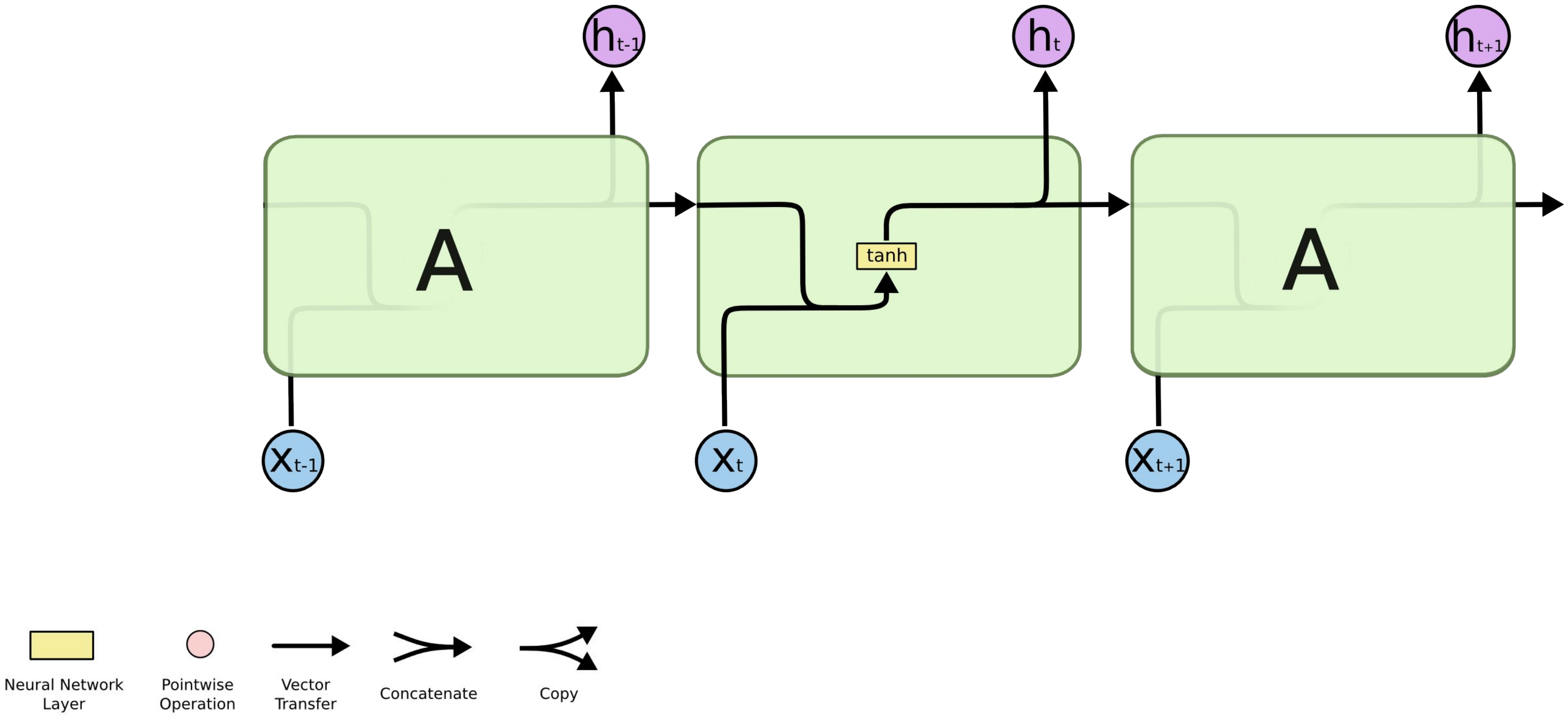
- Exploding Gradients

$$1.01^{1000} = 20959.15...$$

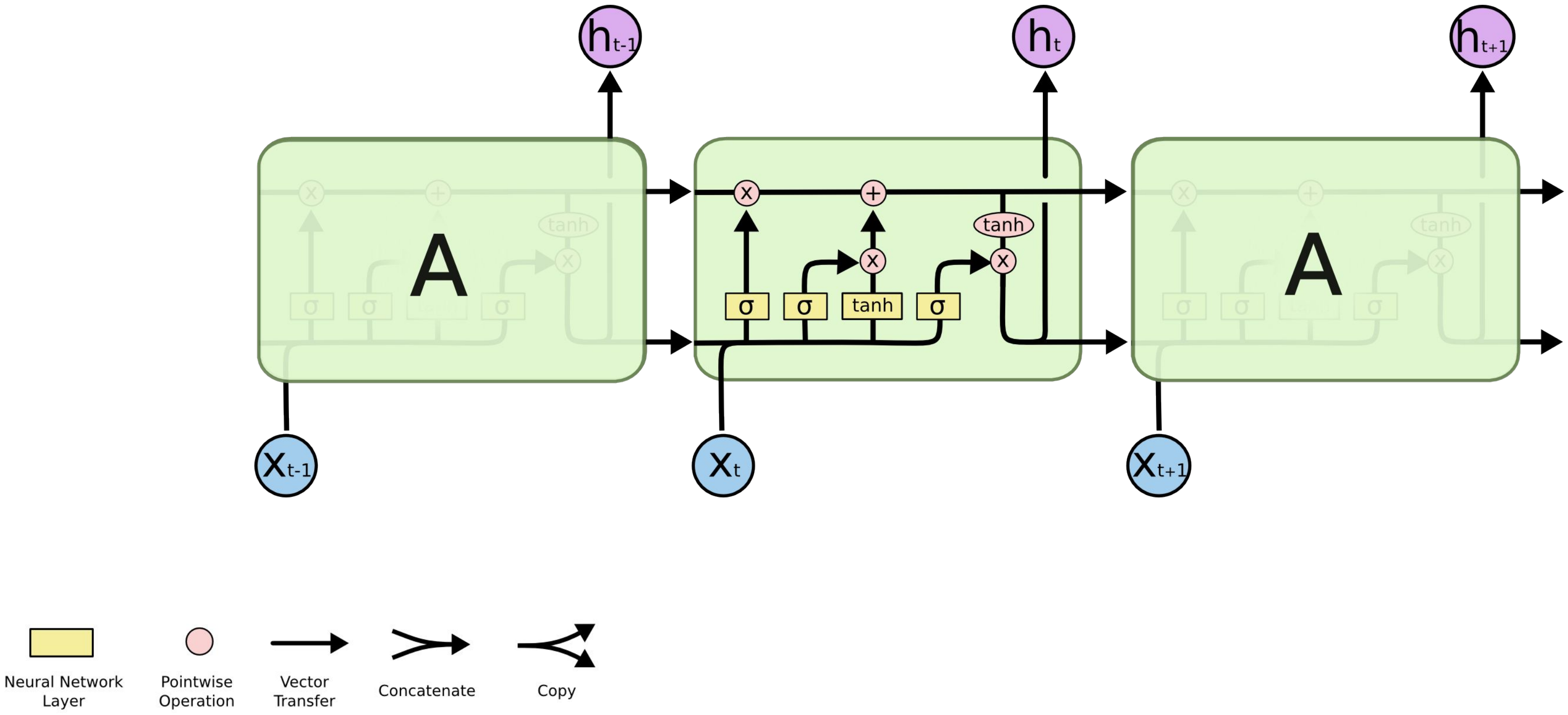


LSTM

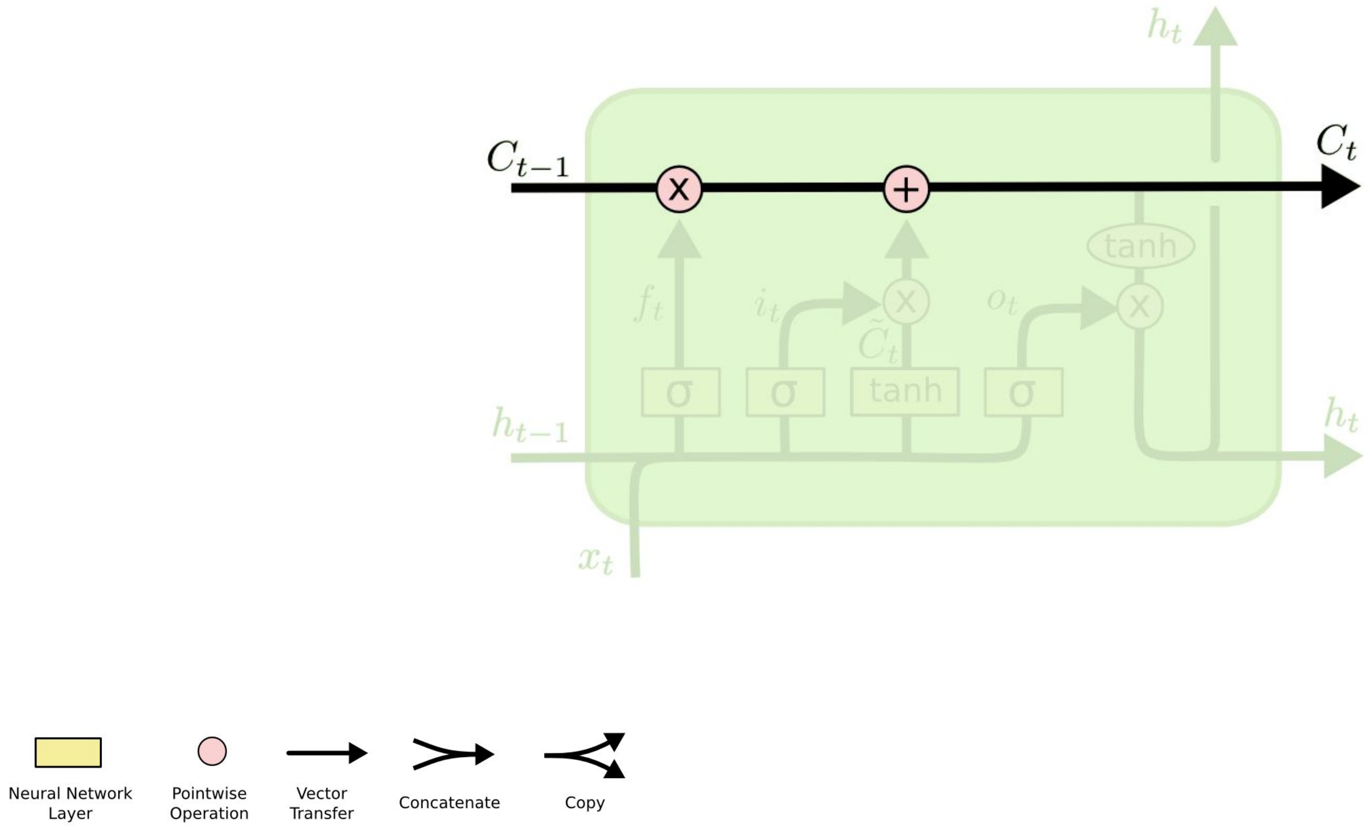
Recurrent Neural Networks



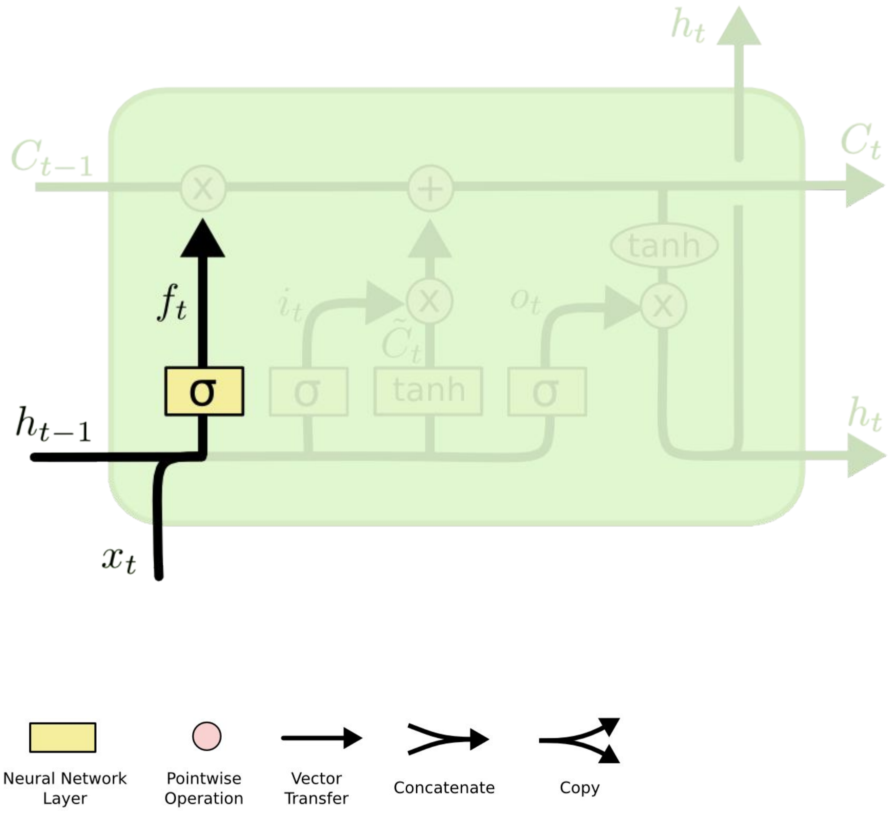
Long Short Term Memory networks



Long Short Term Memory networks

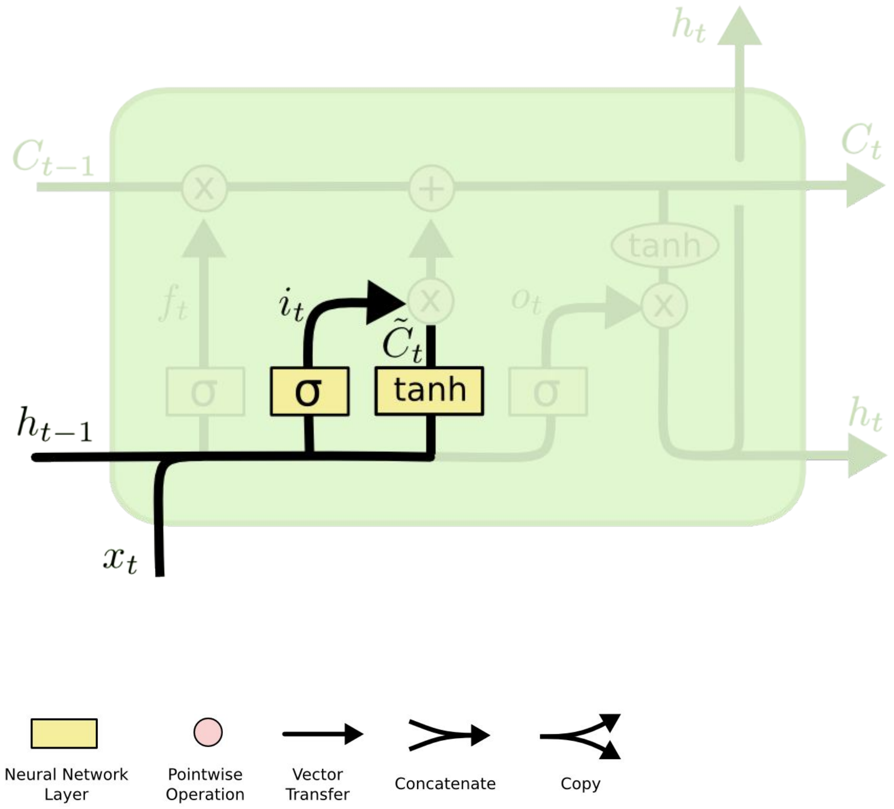


Long Short Term Memory networks



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

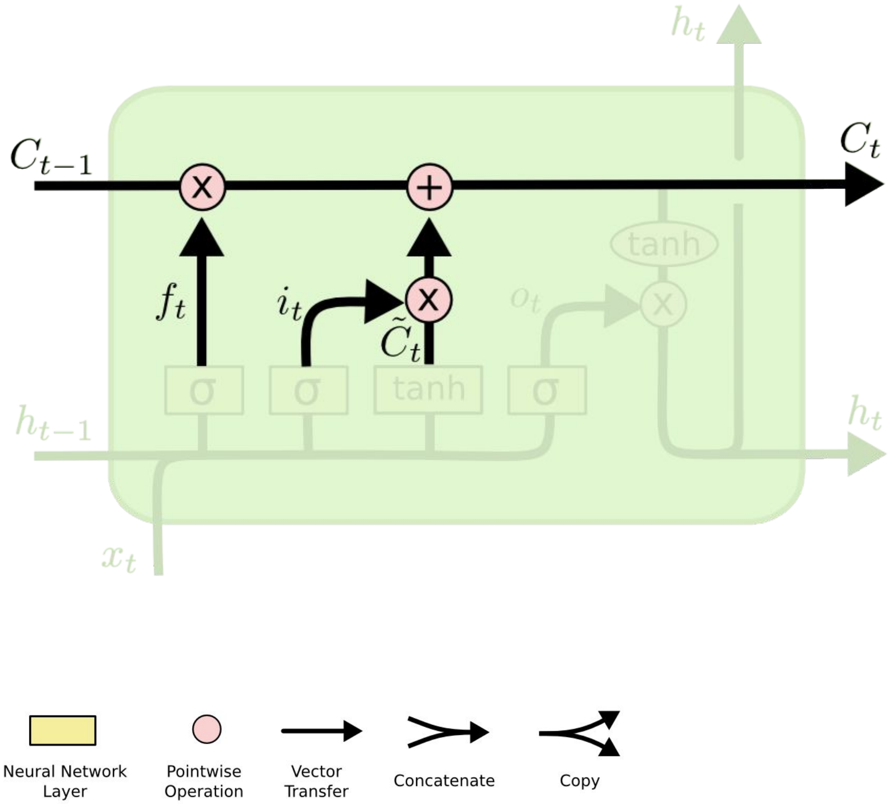
Long Short Term Memory networks



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

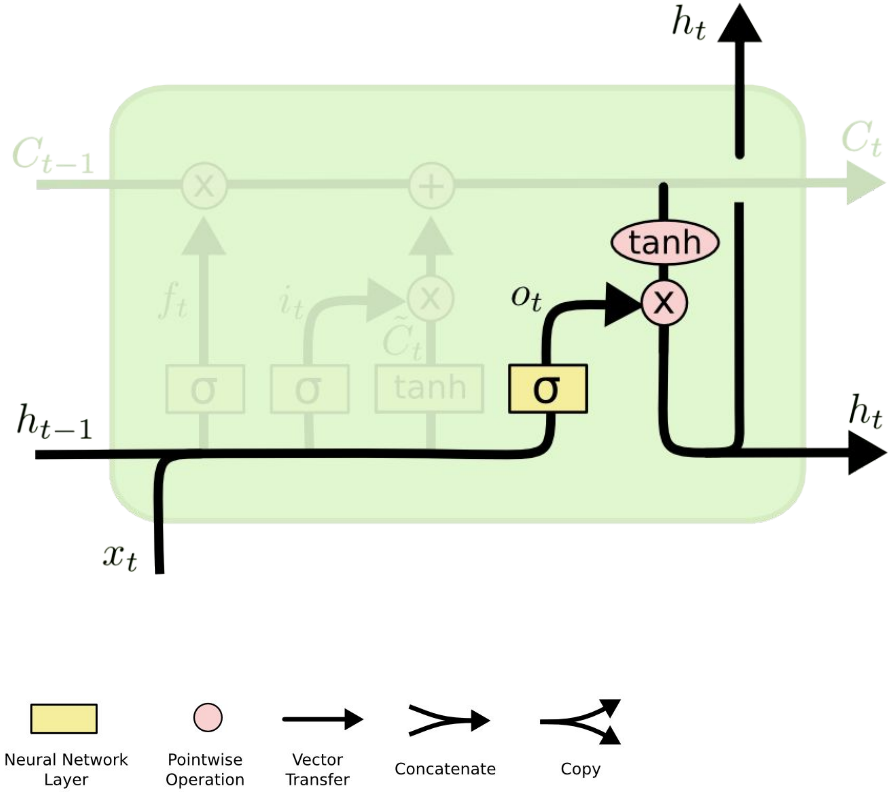
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Long Short Term Memory networks



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Long Short Term Memory networks



$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Long Short Term Memory networks

- Conclusión simples:
    - Las LSTM tienen en cuenta la estructura ordenada de la serie de tiempo.
    - Las LSTM son buenas para aprender dependencias distantes en el tiempo.

Prophet

Prophet

Descompone en 3 a la serie de tiempo:

$$y(t) = g(t) + s(t) + h(t) + e_t$$

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, *72*(1), 37-45.

Prophet

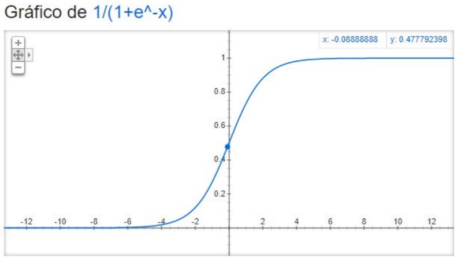
Descompone en 3 a la serie de tiempo:

$$y(t) = g(t) + s(t) + h(t) + e_t$$

- g(t): la tendencia de la serie temporal (trend). Modela cambios no periódicos.

$$g(t) = \frac{C}{1+exp(-k(t-m))}$$

(la fórmula de g(t) una forma simplificada de lo que usan en verdad)



Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, *72*(1), 37-45.

Prophet

Descompone en 3 a la serie de tiempo:

$$y(t) = g(t) + s(t) + h(t) + e_t$$

- g(t): la tendencia de la serie temporal (trend). Modela cambios no periódicos.
- s(t): la periodicidad de la serie temporal (seasonality)

$$s(t) = \Sigma(a_n cos(\frac{2\pi nt}{P}) + b_n sin(\frac{2\pi nt}{P}))$$

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, *72*(1), 37-45.

Prophet

Descompone en 3 a la serie de tiempo:

$$y(t) = g(t) + s(t) + h(t) + e_t$$

- g(t): la tendencia de la serie temporal (trend). Modela cambios no periódicos.
- s(t): la periodicidad de la serie temporal (seasonality)
- h(t): las irregularidades de la serie temporal(holidays)

$$h(t) = \kappa[1(t \in D_1), \ldots, 1(t \in D_L)]$$

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, *72*(1), 37-45.

Prophet

Descompone en 3 a la serie de tiempo:

$$y(t) = g(t) + s(t) + h(t) + e_t$$

- g(t): la tendencia de la serie temporal (trend). Modela cambios no periódicos.
- s(t): la periodicidad de la serie temporal (seasonality)
- h(t): las irregularidades de la serie temporal(holidays)
- e_t: el error (que se asume gaussiano)

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, *72*(1), 37-45.

Prophet

Descompone en 3 a la serie de tiempo:

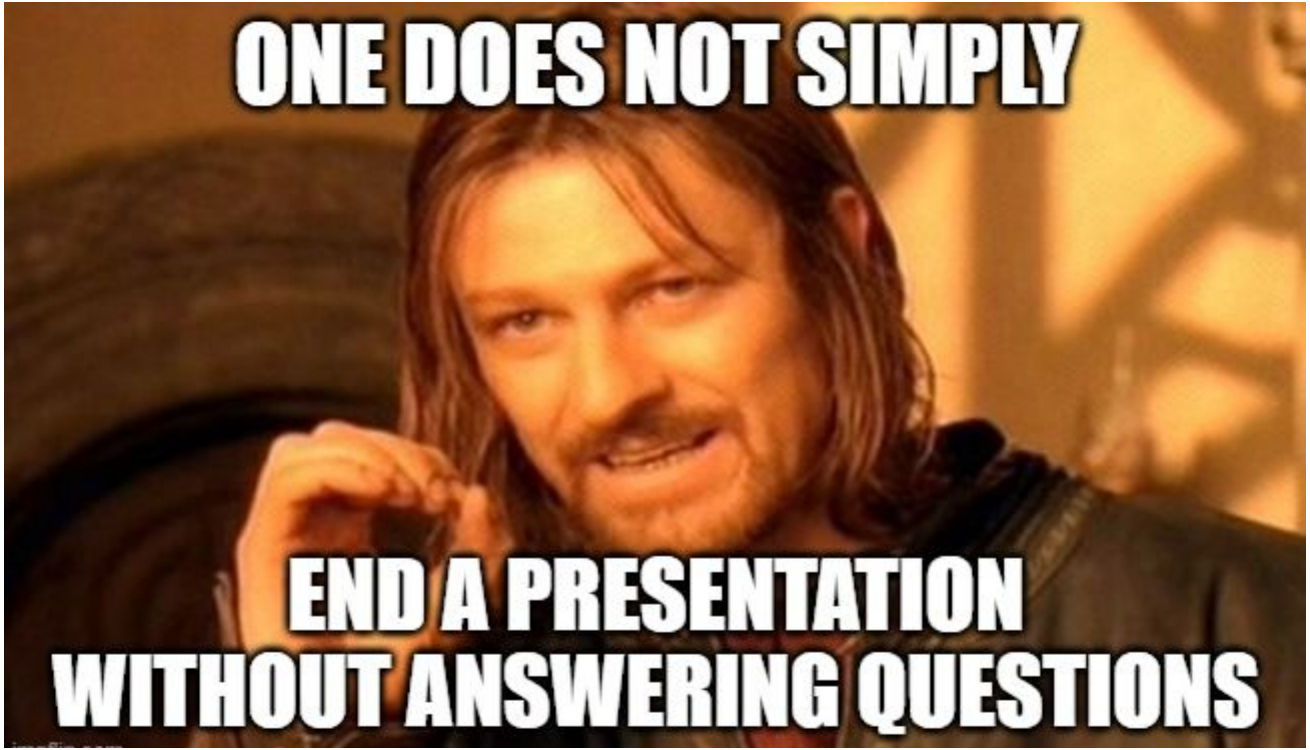
$$y(t) = g(t) + s(t) + h(t) + e_t$$

Luego usa L-BFGS (un algoritmo de optimización de la familia del método de Newton) para calcular las distribuciones a posteriori de los parámetros.

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, *72*(1), 37-45.


TALKING ABOUT
TIME SERIES?
MY FAVORITE MOVIE
IS BACK TO THE FUTURE 2

Preguntas


ONE DOES NOT SIMPLY
END A PRESENTATION
WITHOUT ANSWERING QUESTIONS