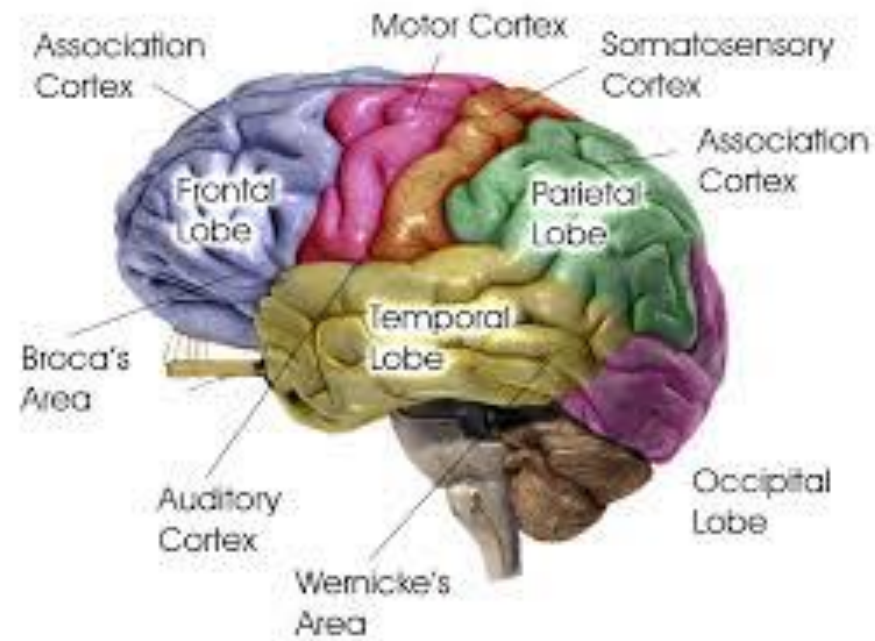


# | Inteligencia Artificial Redes neuronales

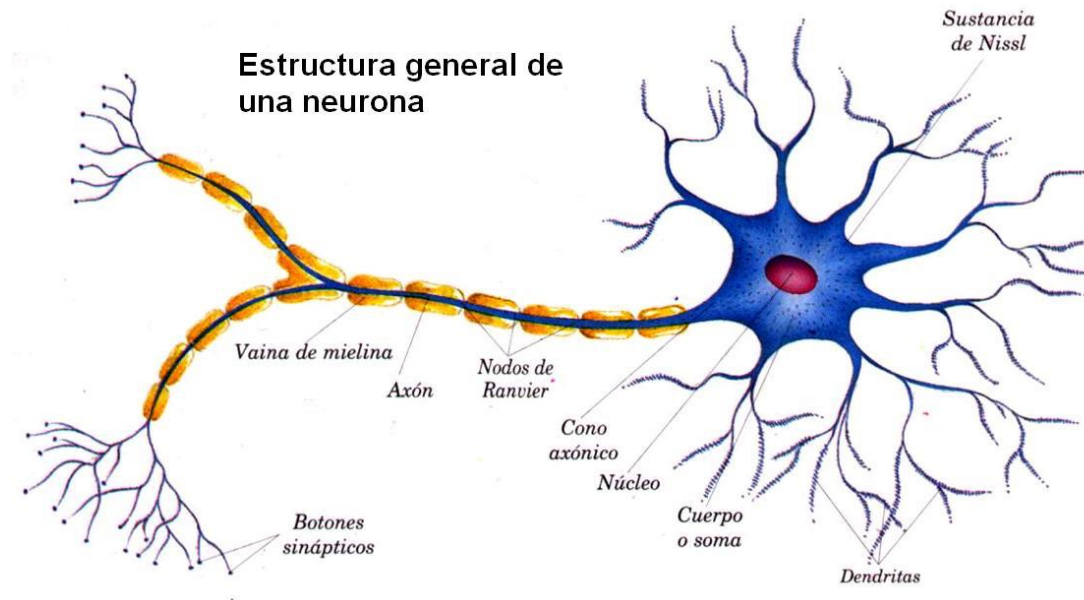
# Ingeniería inversa

- Unos de los proyectos de ingeniería inversa más retadores es entender como funciona el cerebro



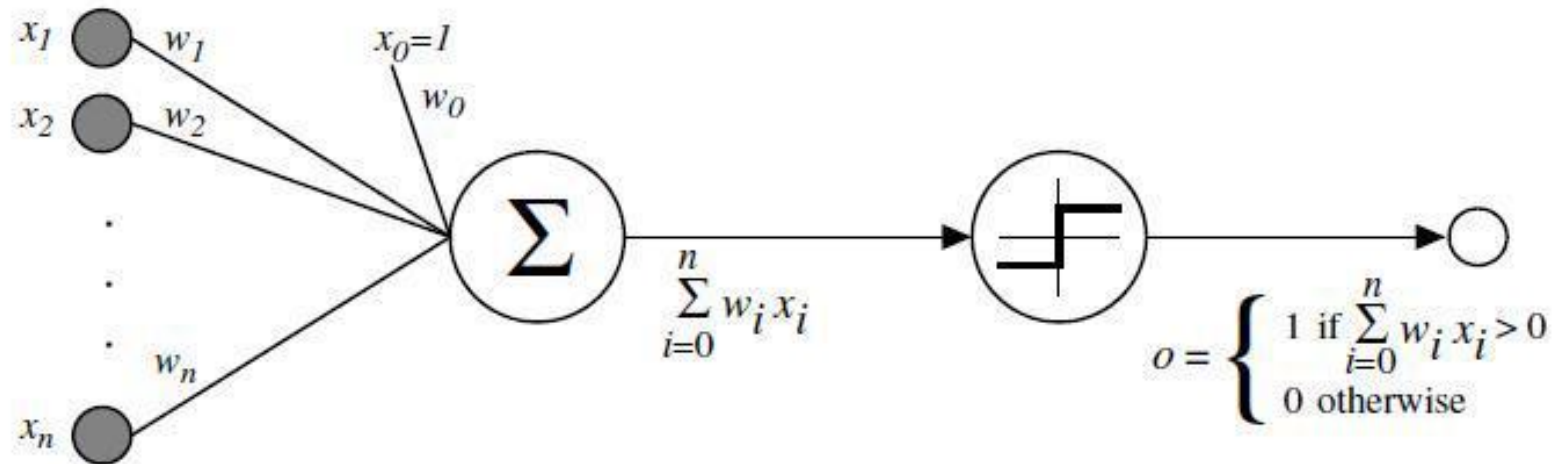
# Neurona

- Las células básicas con las que procesa la información el cerebro son las neuronas
  - El cerebro humano tiene 81 mil millones de neuronas y  $1.5 \times 10^{14}$  sinapsis

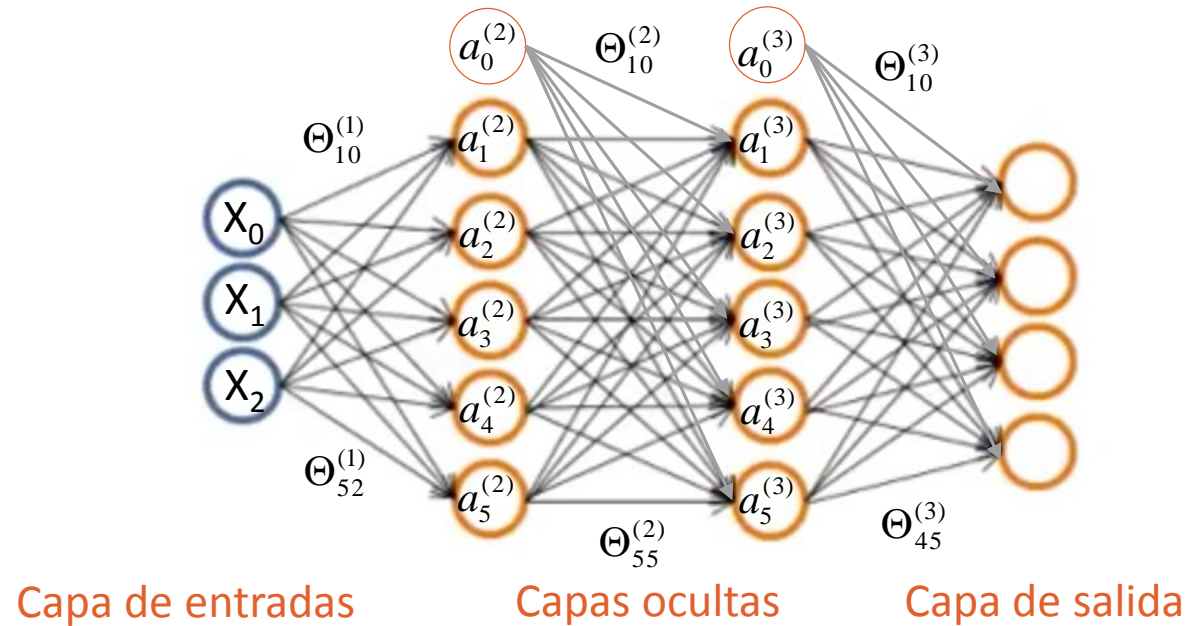


# Redes neuronales

- Perceptrón



# Redes Neuronales



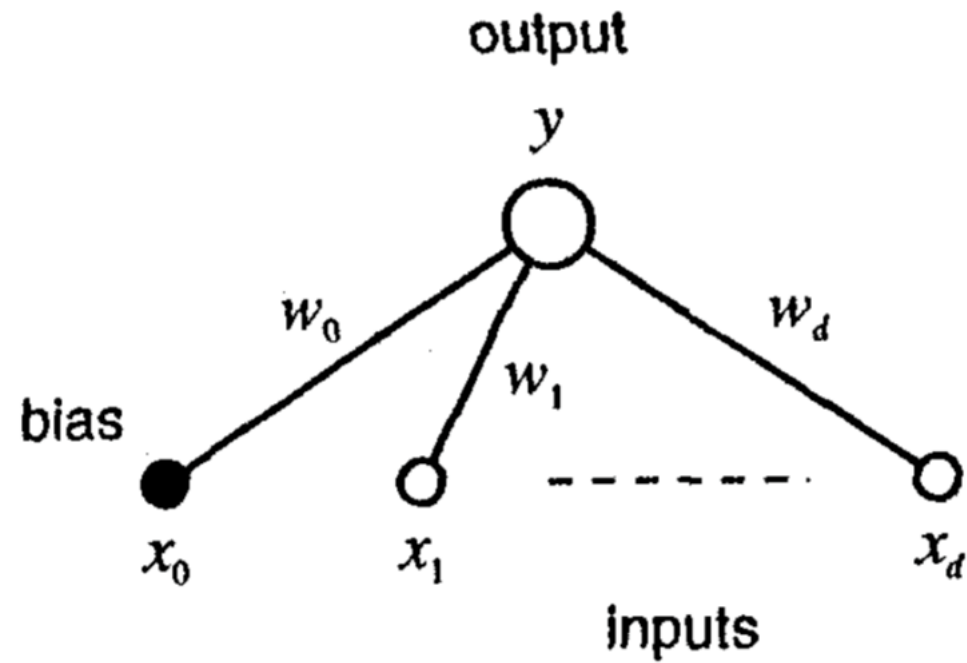
$$a^{(1)} = x$$

$$a^{(2)} = g(\Theta^{(1)} a^{(1)})$$

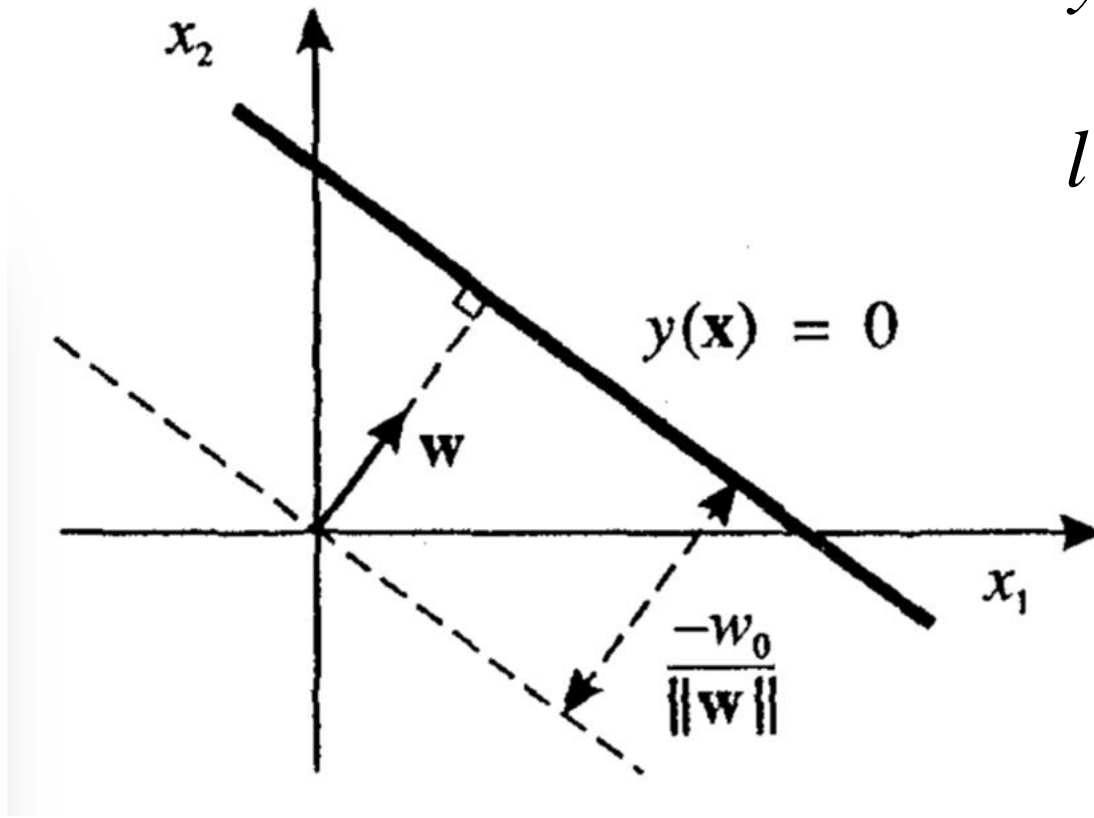
$$a^{(3)} = g(\Theta^{(2)} a^{(2)})$$

$$h_{\Theta}(x) = a^{(4)} = g(\Theta^{(3)} a^{(3)})$$

# Redes de una capa



# Redes de una capa



$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

$$l = \frac{\mathbf{w}^T \mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}$$

# Categorizador logístico

$$p(x | C_k) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{\left\{ -\frac{1}{2} (x - \mu_k)^T \Sigma^{-1} (x - \mu_k) \right\}}$$

$$P(C_1 | x) = \frac{p(x | C_1)P(C_1)}{p(x | C_1)P(C_1) + p(x | C_2)P(C_2)}$$

$$= \frac{1}{1 + e^{-a}}$$

$$= g(a)$$

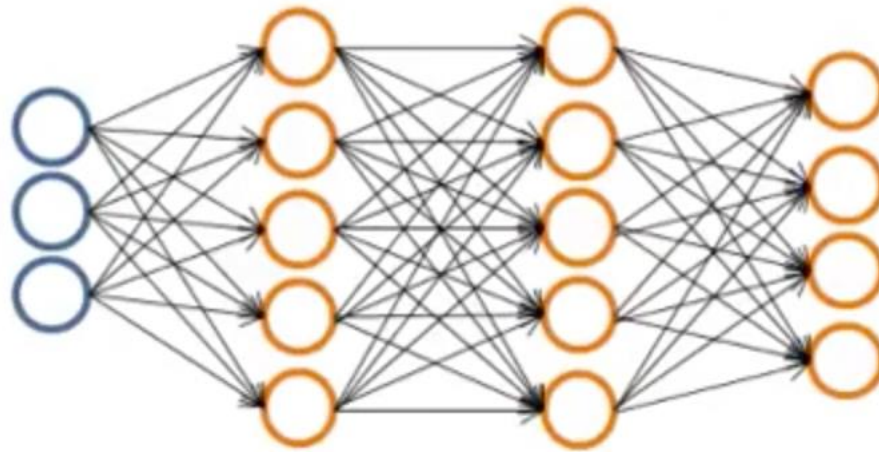
$$a = \ln \frac{p(x | C_1)P(C_1)}{p(x | C_2)P(C_2)}$$

$$a = w^T x + w_0$$



# Clasificación multiclase

- Softmax



$$h_{\Theta}(x) \in \mathbb{R}^4$$

$$\sigma(x^T w)_j = P(y = j | x) = \frac{e^{x^T w_j}}{\sum_{i=1}^K e^{x^T w_i}} \quad \text{for } j = 1, \dots, K$$

$$\frac{\partial \mathcal{E}^n}{\partial a_k} = y_k - t_k$$

# Racional detrás del back propagation

- Consideremos la expresión:

$$e=(a+b)*(b+1)$$

- Podemos reescribirla como

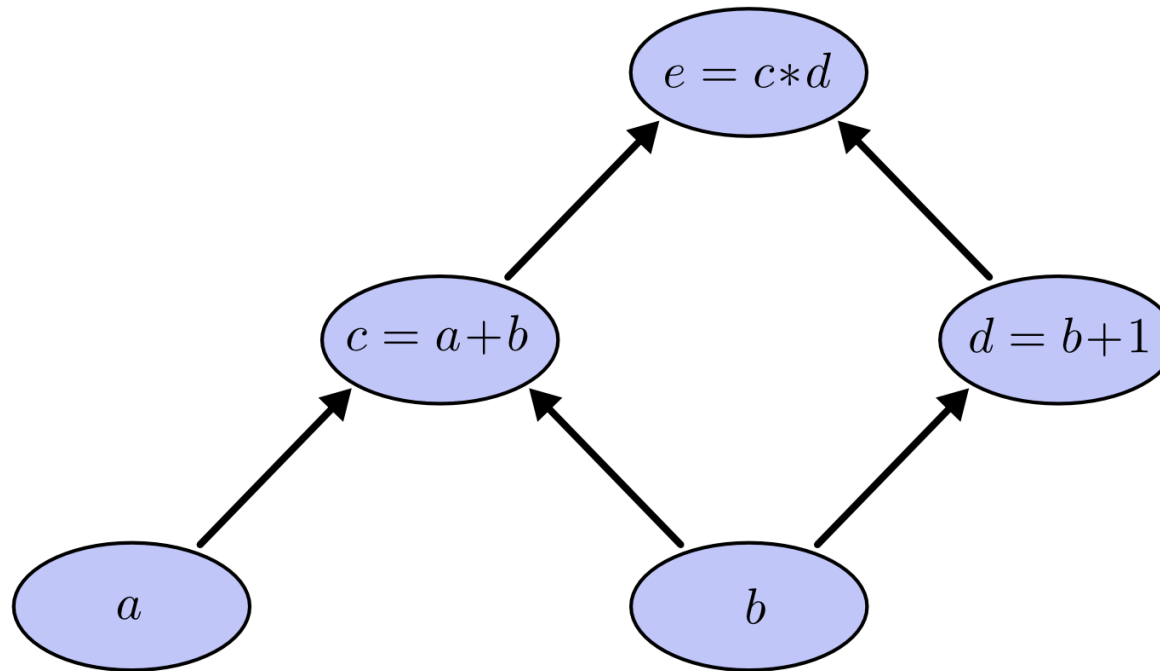
$$c=a+b$$

$$d=b+1$$

$$e=c*d$$

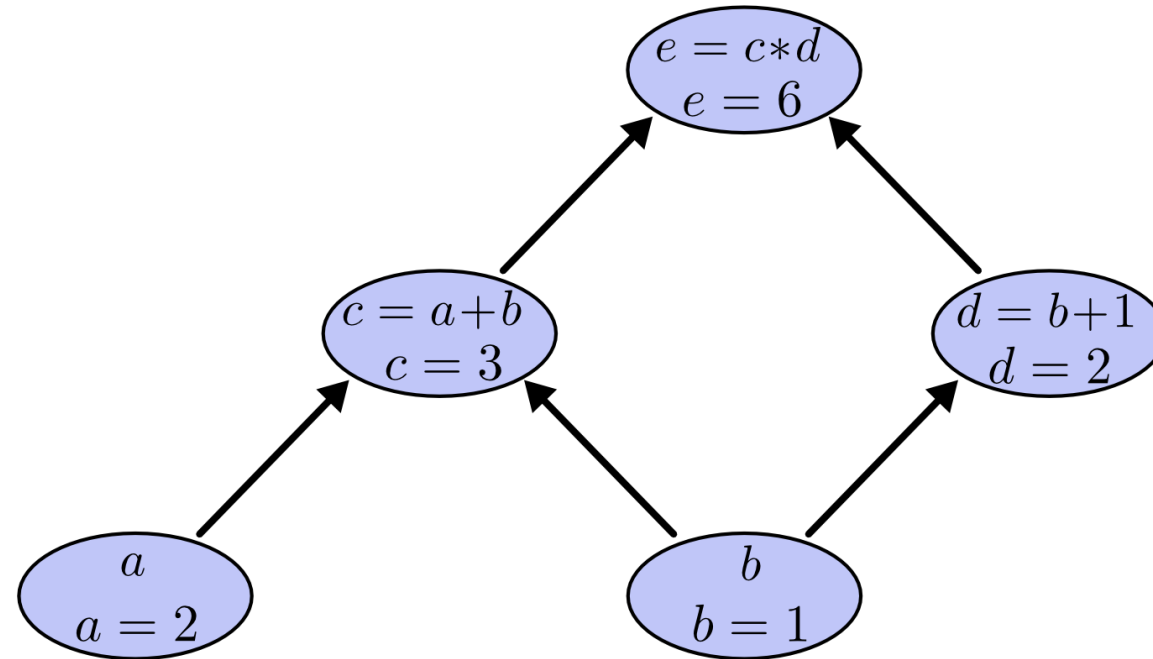
# Racional detrás del back propagation

- Podemos representar la expresión mediante el siguiente grafo computacional:



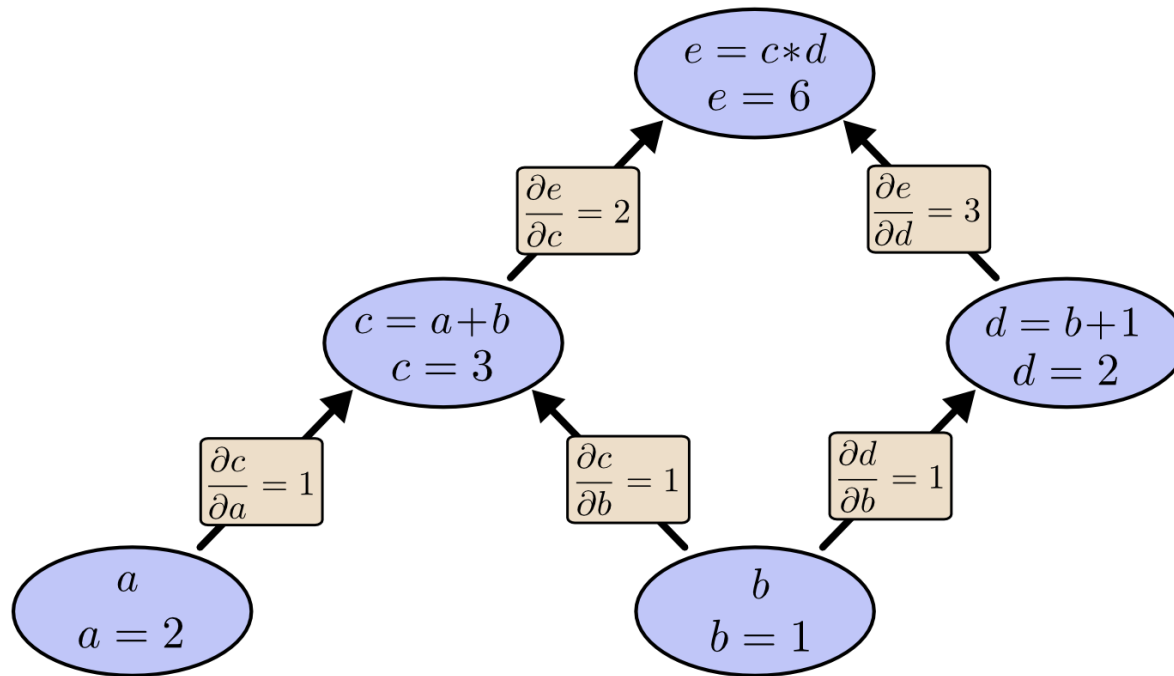
# Racional detrás del back propagation

- Podemos evaluar la expresión asignando valores a las variables (por ejemplo  $a=2$  y  $b=1$ ):



# Racional detrás del back propagation

- Para entender las derivadas en el grafo debemos analizar las derivadas en las aristas:

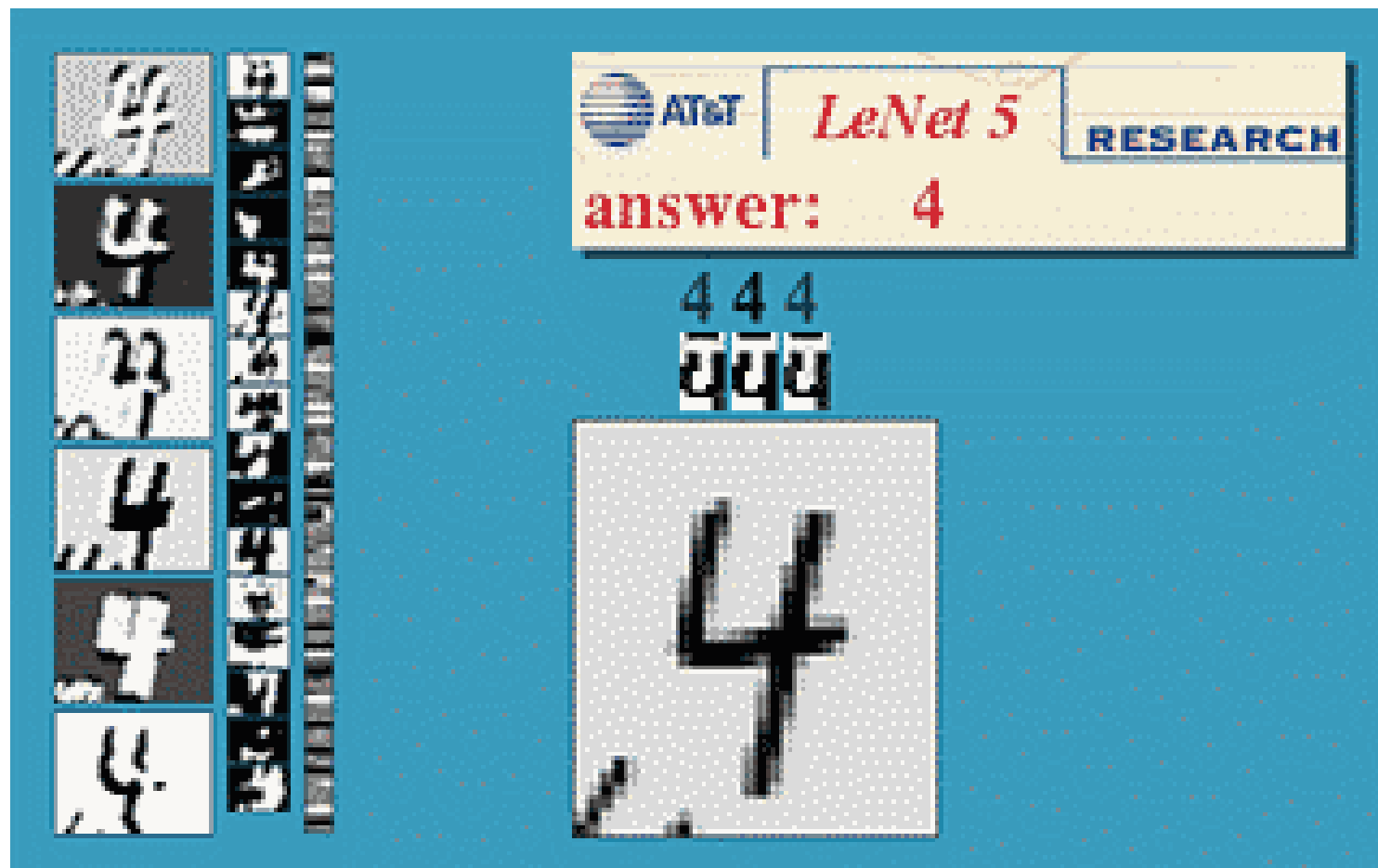


Reglas de la suma y el producto:

$$\frac{\partial}{\partial a}(a + b) = \frac{\partial a}{\partial a} + \frac{\partial b}{\partial a}$$

$$\frac{\partial}{\partial u}uv = u \frac{\partial v}{\partial u} + v \frac{\partial u}{\partial u}$$

# Ejemplo



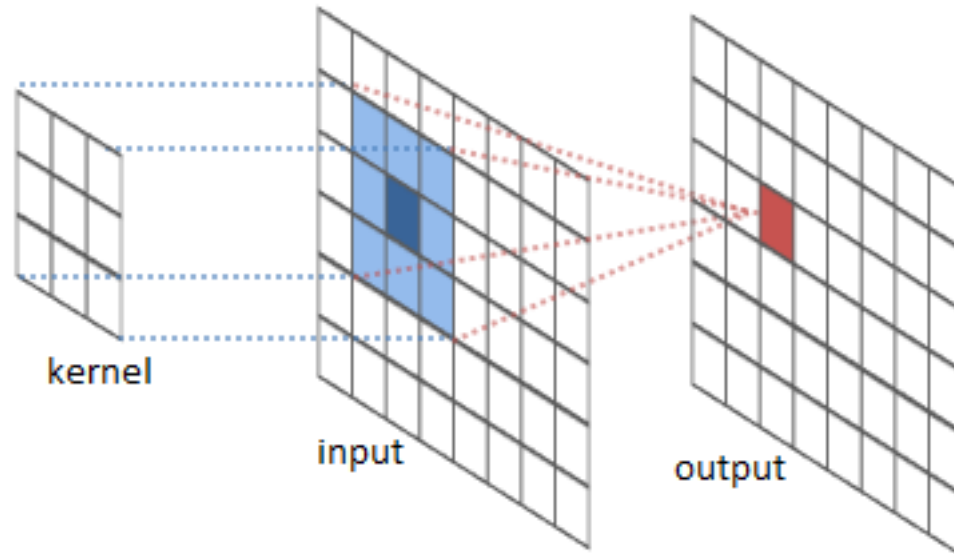
# Redes neuronales convolucionales

- La convolución es una integral que expresa la cantidad de superposición de una función  $g$  mientras se desplaza sobre otra función  $f$ .
  - Por lo tanto, "mezcla" una función con otra.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

# Redes neuronales convolucionales

- Podemos pensar que las imágenes son funciones bidimensionales
  - Varias transformaciones de imágenes son convoluciones de una imagen con un kernel



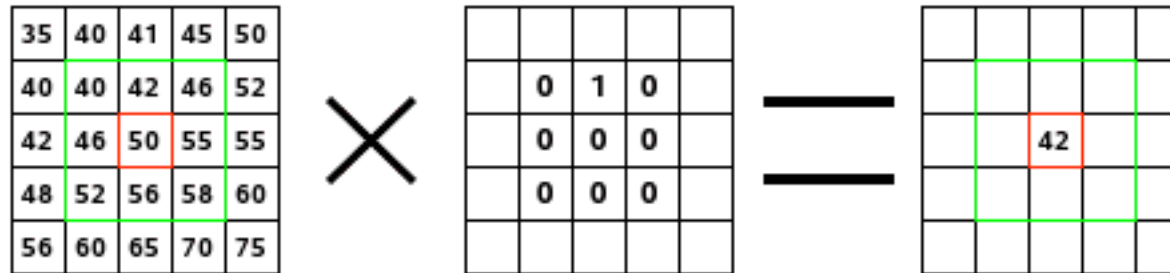


# Redes neuronales convolucionales

- Un ejemplo simple:

$$(40*0)+(42*1)+(46*0) + (46*0)+(50*0)+(55*0) + (52*0)+(56*0)+(58*0) = 42$$

- Sólo movió el pixel superior un reglón abajo



# Redes neuronales convolucionales

- Difuminar (promediando pixeles)

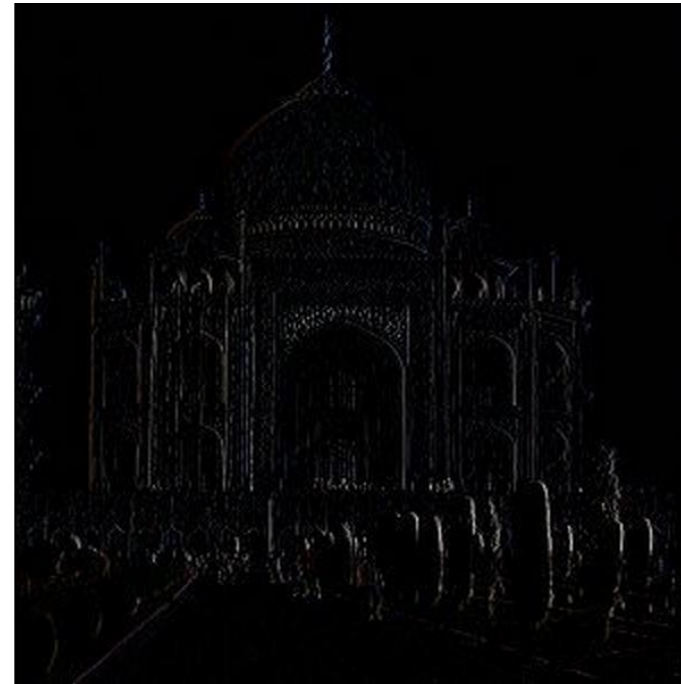
0	0	0	0	0
0	1/9	1/9	1/9	0
0	1/9	1/9	1/9	0
0	1/9	1/9	1/9	0
0	0	0	0	0



# Redes neuronales convolucionales

- Detección de bordes (llevando a valores cercanos a cero pixeles vecinos similares)

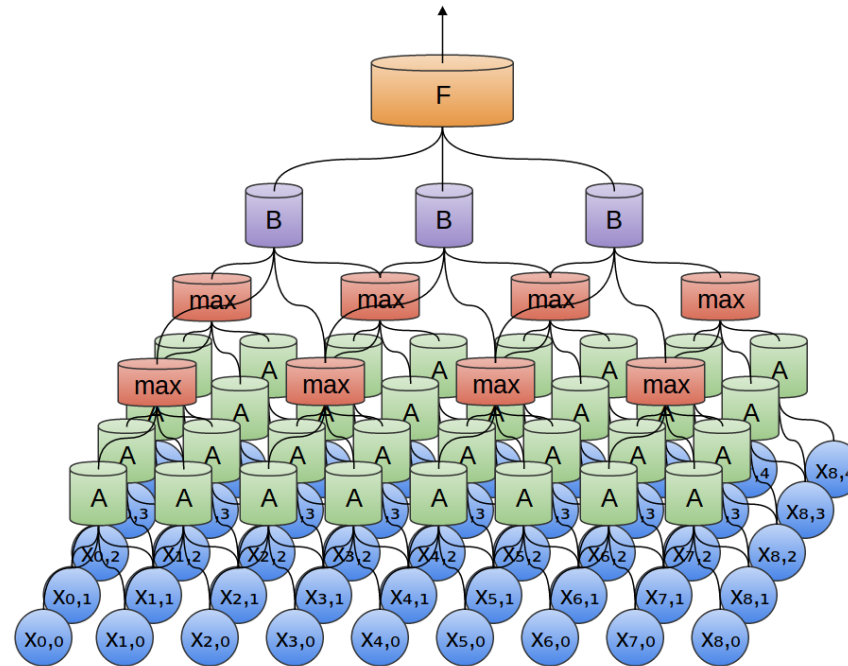
0	0	0	0	0
0	0	0	0	0
0	-1	1	0	0
0	0	0	0	0
0	0	0	0	0



# Redes neuronales convolucionales

- Reducción de resolución mediante max-pooling para resumir características sobre una región

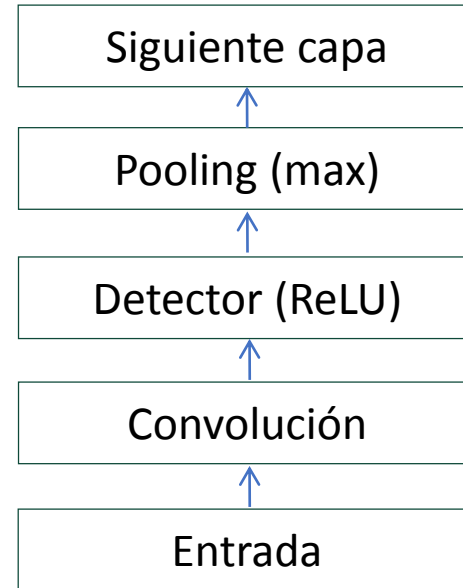
Es más importante  
conocer si se dio una  
característica que  
justo el lugar en  
donde se dio



Hace la arquitectura  
más tolerante a  
pequeñas  
translaciones

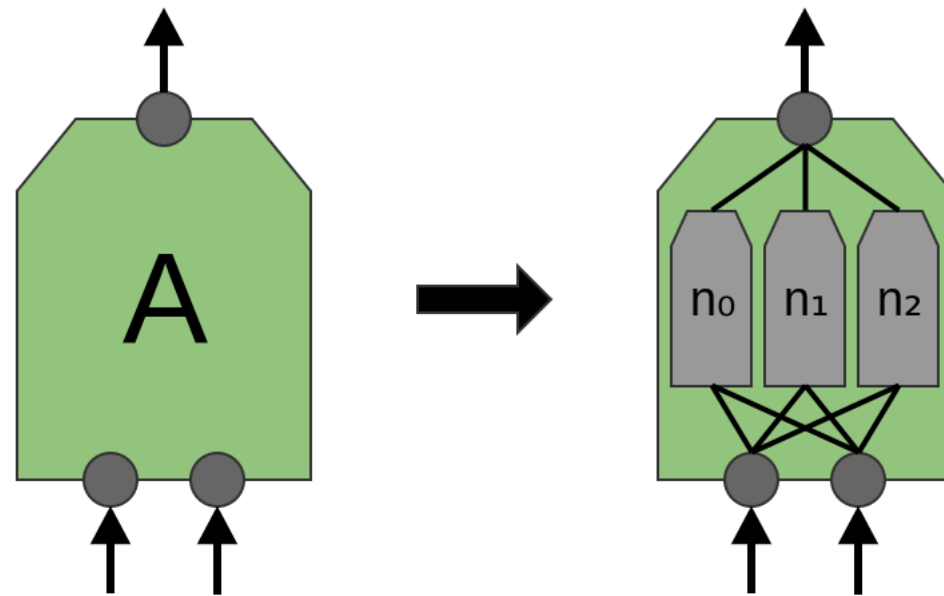
# Redes neuronales convolucionales

- Arquitectura común



# Redes neuronales convolucionales

- Normalmente  $A$  es una serie de neuronas en paralelo en donde todas tienen las mismas entradas y obtienen diferentes características



# Redes neuronales recurrentes

- Son una familia de redes neuronales que procesan datos secuenciales
- Al igual que las redes convolucionales tienen parámetros compartidos lo que les permite generalizar:
  - distintos tamaños de secuencia
  - posiciones distintas de la secuencia

# Redes neuronales recurrentes

- Son una familia de redes neuronales que procesan datos secuenciales
- Al igual que las redes convolucionales tienen parámetros compartidos lo que les permite generalizar:
  - distintos tamaños de secuencia
  - posiciones distintas de la secuencia



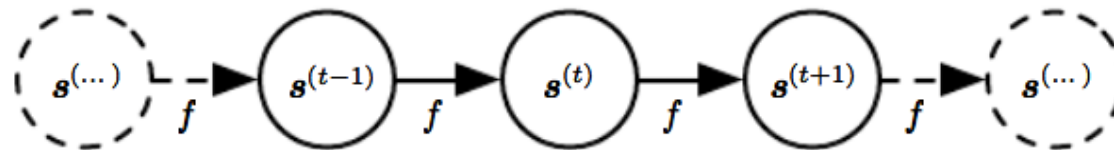
# Redes neuronales recurrentes

- Grafo computacional del despliegue del sistema dinámico:

$$\mathbf{s}^{(t)} = f(\mathbf{s}^{(t-1)}; \boldsymbol{\theta})$$

Para  $\tau = 3$

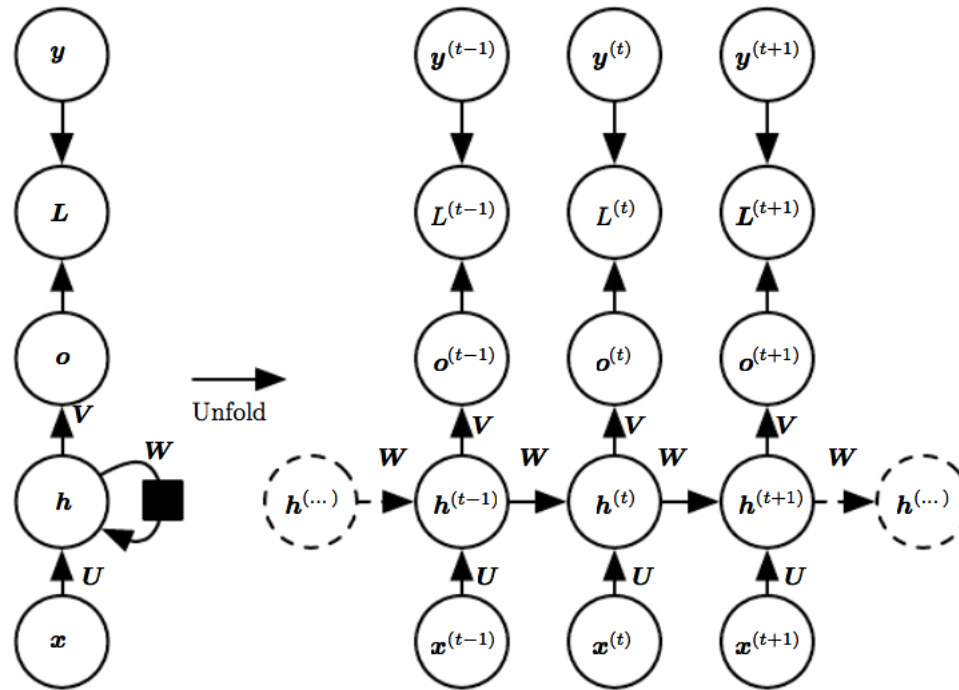
$$\begin{aligned}\mathbf{s}^{(3)} &= f(\mathbf{s}^{(2)}; \boldsymbol{\theta}) \\ &= f(f(\mathbf{s}^{(1)}; \boldsymbol{\theta}); \boldsymbol{\theta})\end{aligned}$$



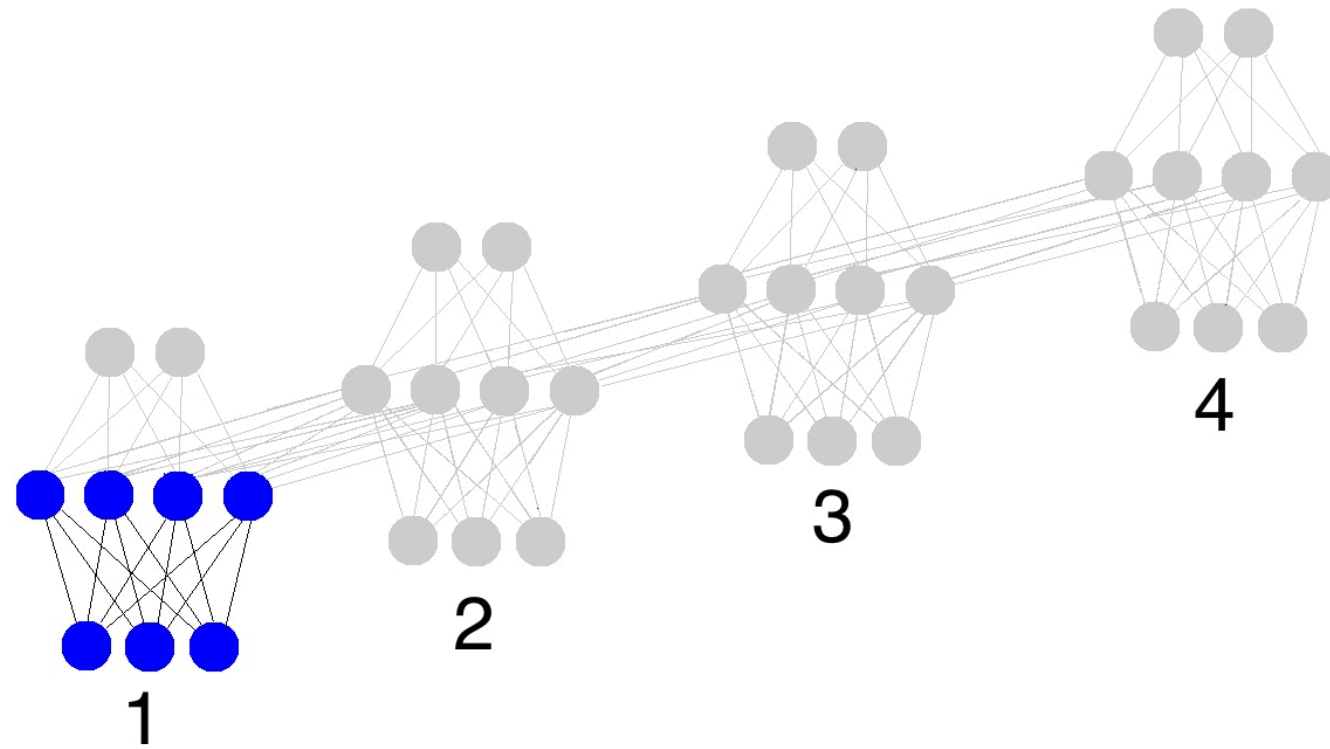
$\mathbf{s}^{(t)}$  = estado en tiempo  $t$

# Redes neuronales recurrentes

- Una salida en cada paso y conexiones recurrentes en la capa oculta

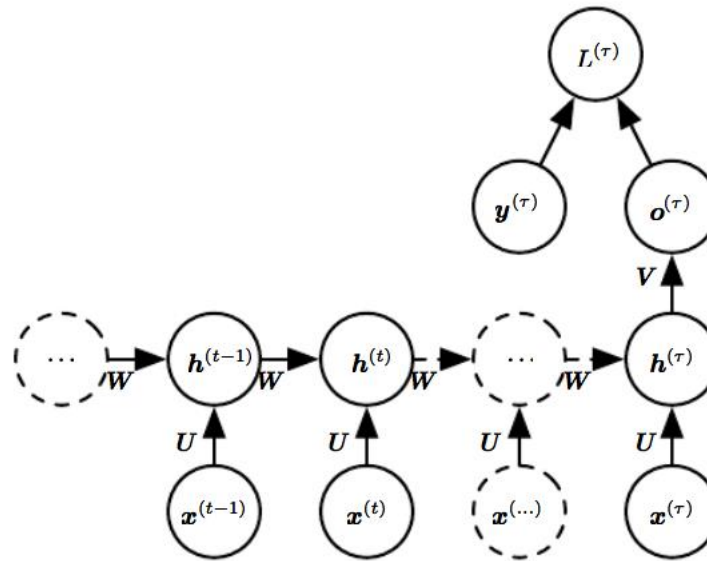


# Redes neuronales recurrentes



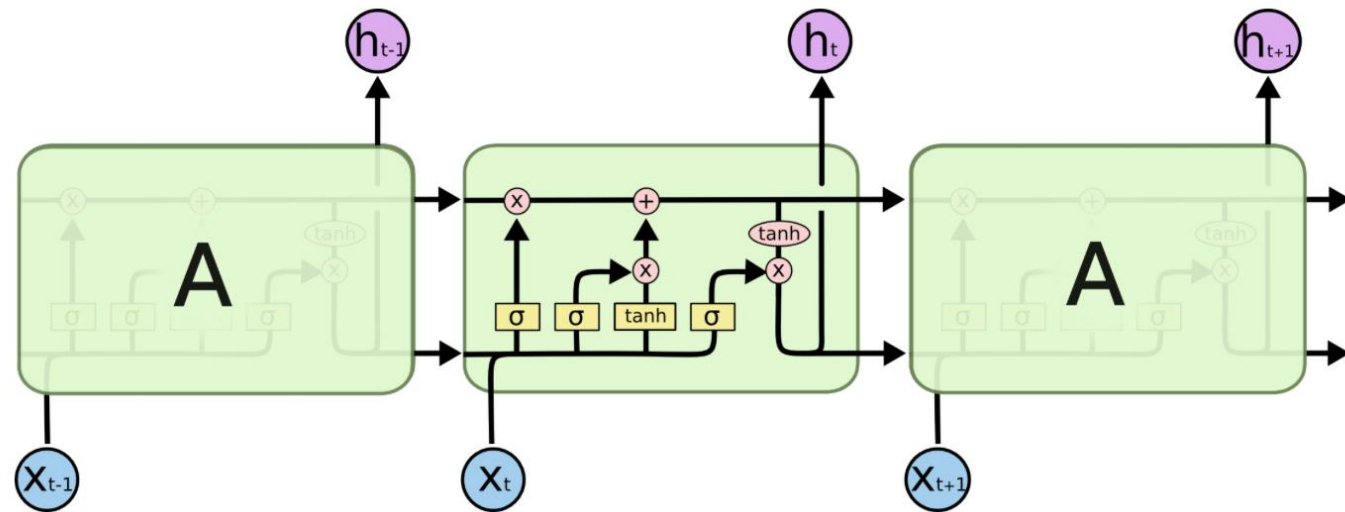
# Redes neuronales recurrentes

- Una salida al final de la secuencia y conexiones recurrentes en la capa oculta



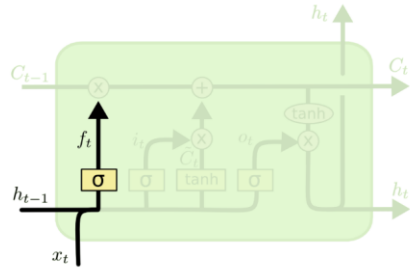
# Long Short Term Memory (LSTM)

- Un tipo de red neuronal recurrente capaz de aprender dependencia de largo plazo



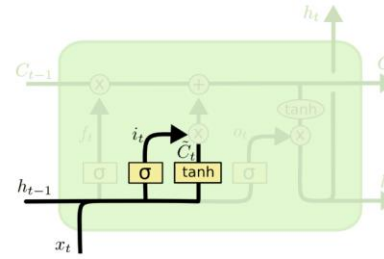
# Long Short Term Memory (LSTM)

Capa de compuertas para olvidar



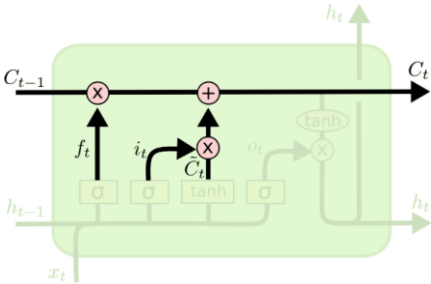
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Capa de compuertas de entrada



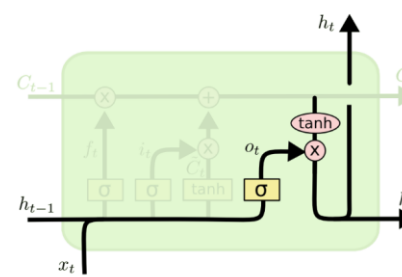
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Cálculo del estado de la celda



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

Cálculo de la salida



$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

# TensorFlow

- TensorFlow es una biblioteca de software de código abierto para el cálculo numérico usando diagramas de flujo de datos.
- Desarrollada originalmente por Google Brain para conducir investigaciones de aprendizaje máquina y, en específico, deep neural networks.



# Keras

- Keras es una biblioteca de redes neuronales minimalista, altamente modular, escrita en Python capaz de ejecutarse en utilizando CNTK, TensorFlow o Theano.
- Fue desarrollado con el objetivo de facilitar la experimentación rápida.
  - Ser capaz de ir de la idea al resultado con el menor retraso posible es clave para hacer una buena investigación
- `pip install keras --user`





# Word Embedding

- Método de aprendizaje no supervisado que transforma un conjunto de palabras en vectores de altas dimensiones
- Basado en el modelo distribucional
  - Palabras que aparecen en los mismos contextos comparten significado semántico
- Los vectores tiene la propiedad de que conceptos similares se agrupan

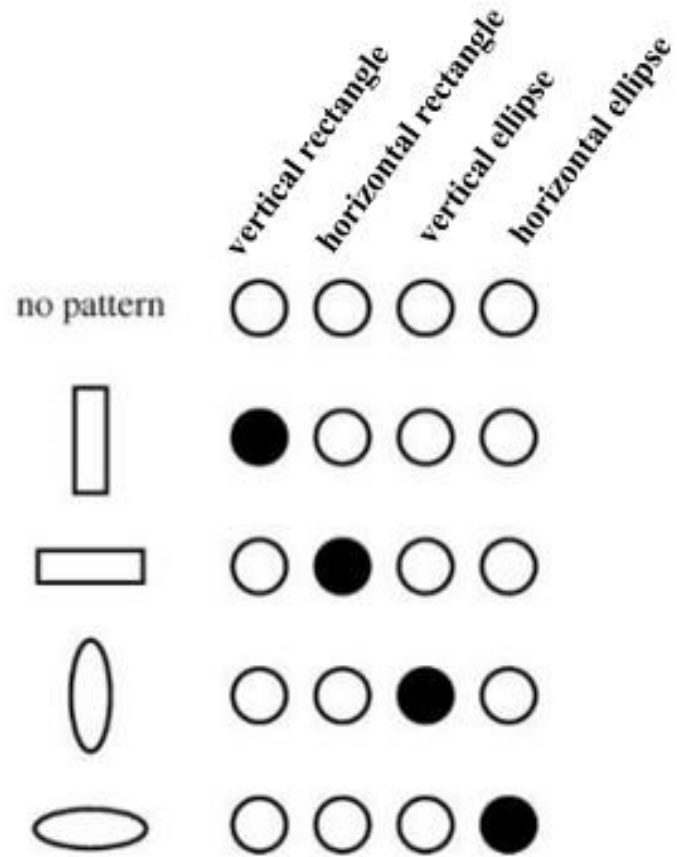
Similitud entre Hombre y Mujer: 0.74
- Se pueden realizar operaciones entre los vectores

$\text{Rey} - \text{Hombre} + \text{Mujer} = \text{Reina}$

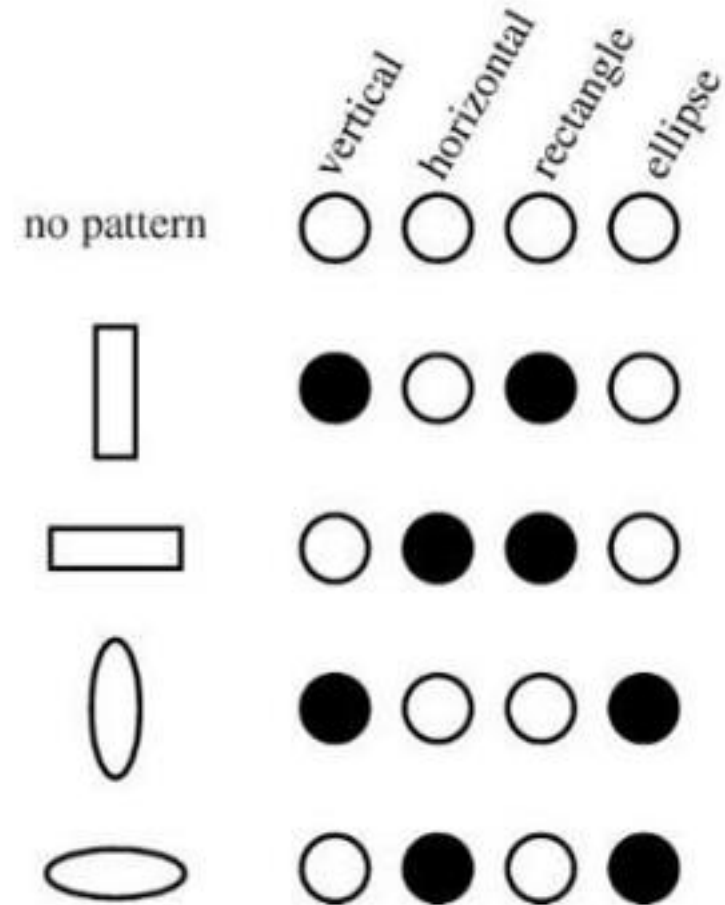
# IMDB corpus

- 25,000 opiniones de películas de IMDB etiquetadas con su sentimiento asociado (positivo con calificación  $\geq 7$  / negativo con calificación  $\leq 4$ ).
- Cada revisión se codifica como una secuencia de índices de palabras (enteros).
- Por conveniencia, las palabras son indexados por frecuencia global en el conjunto de datos.
  - Se puede filtrar la información rápidamente

# Representación distribuida

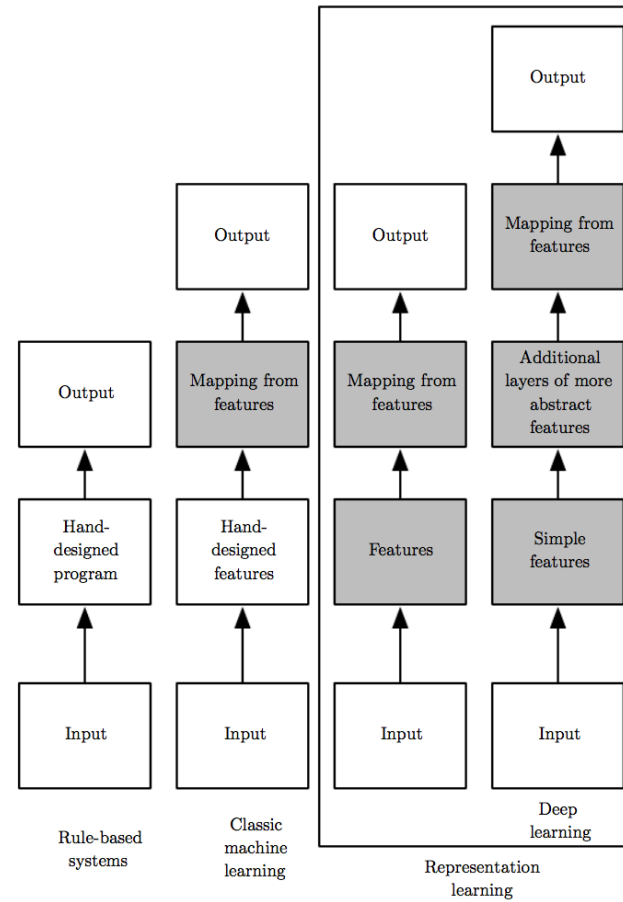


localista

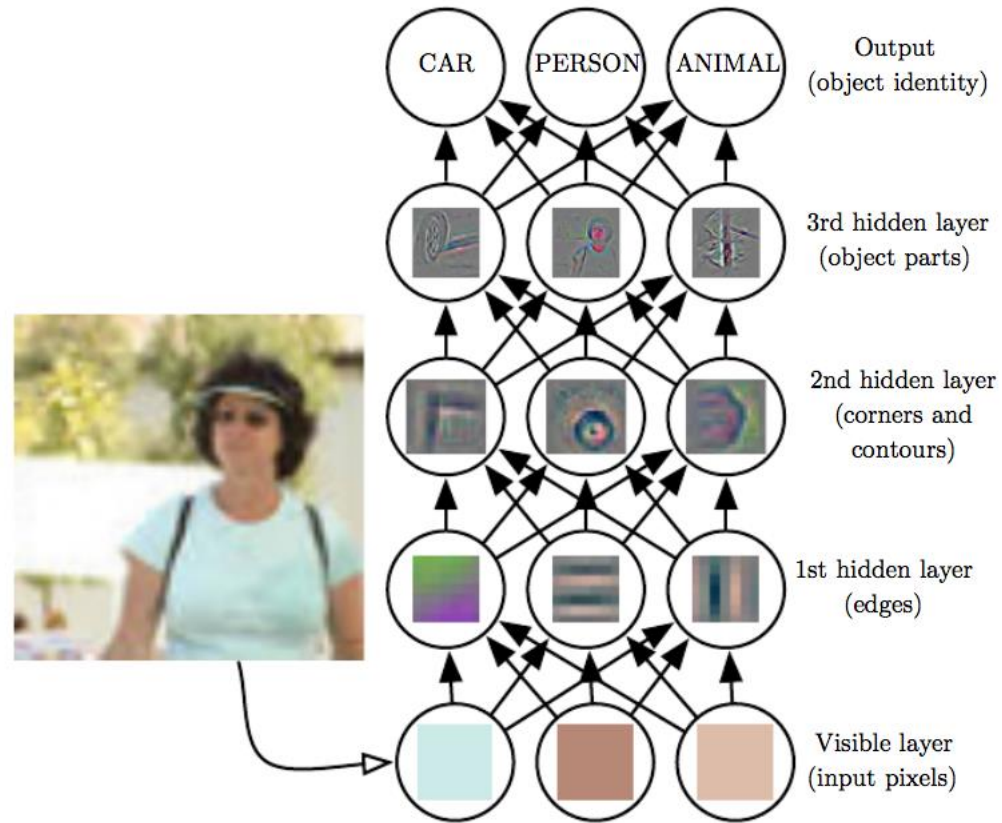


distribuida

# Disciplinas en inteligencia artificial



# Deep learning





# Microsoft

©2014 Microsoft Corporation. All rights reserved. Microsoft, Windows, Office, Azure, System Center, Dynamics and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.