

Modulo 1 (II)

Introducción al Deep Learning Redes Neuronales Artificiales (ANN)

Mercedes Riveira-Martin
Instituto de Investigación Sanitaria Galicia Sur (IISGS)

INTELIGENCIA ARTIFICIAL

Roadmap

1. Qué es el Deep Learning
 1. Sus orígenes
2. El pilar principal del DL: Redes Neuronales Artificiales
 1. Elementos principales de una red neuronal
 2. El Perceptrón
 3. Multi-layer perceptron (MLP)
 4. DNN (Deep Neural Network)
 5. ¿Qué es entrenar una red neuronal?
 6. El algoritmo que lo cambió todo: backpropagation
 7. Parámetros que debemos modificar
3. Tipos de redes neuronales
 3. Arquitectura de una red neuronal
 4. Redes neuronales más conocidas:
 5. Arquitecturas más conocidas
4. Convolutional Neural Networks (CNNs) (más en profundidad)
 3. Cómo funciona el cortex visual
 4. Capas de una CNN:
 5. Arquitecturas
 6. Problemas que resuelven
5. Ejemplos de aplicación
 3. Sanidad
 4. RT

¿Qué es el Deep Learning (DL)?

Artificial Intelligence

Cualquier
técnica que
permite a los
ordenadores
imitar la
inteligencia
humana.
Incluye
machine
learning

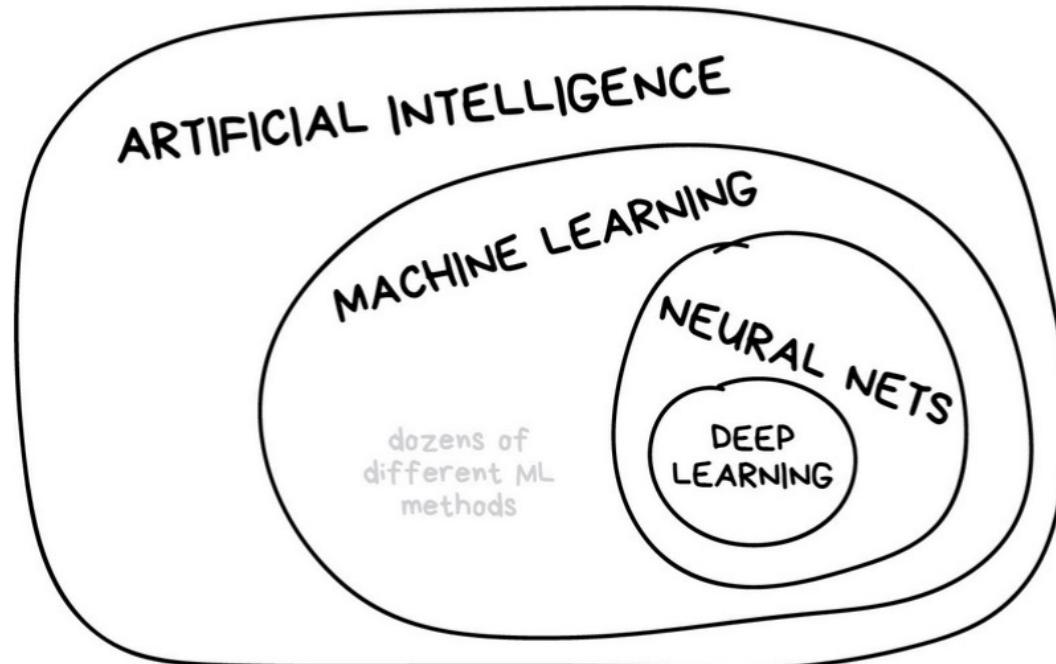
Machine Learning

Subconjunto de
IA cuyas técnicas
permiten a las
máquinas
aprender sin ser
programadas.
Incluye Deep
Learning

Deep Learning

Subconjunto del ML basado en redes
neuronales que permite a una máquina
entrenarse para realizar una tarea
aprendiendo con ejemplos

¿Qué es el Deep Learning (DL)?



Inteligencia Artificial

¿Qué es el Deep Learning (DL)?

“the effort to automate intellectual tasks normally performed by humans”

François Chollet (creador de Keras)

Machine Learning

“field of study that gives computers the ability to learn without being explicitly programmed”

Arthur Samuel

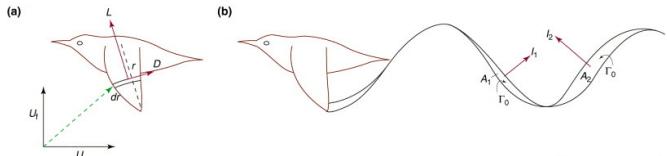
Deep Learning

“[...] is a branch of machine learning where neural networks – algorithms inspired by the human brain – learn from large amounts of data”

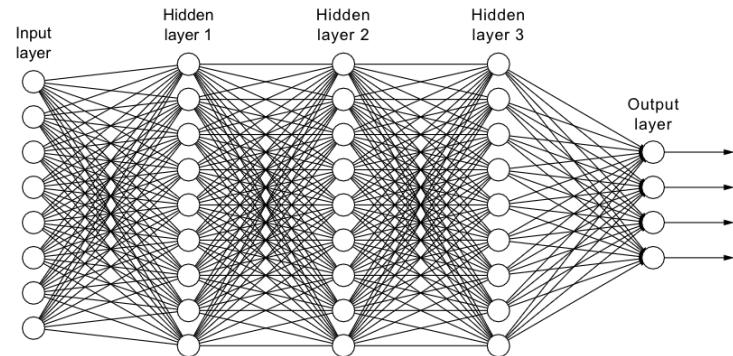
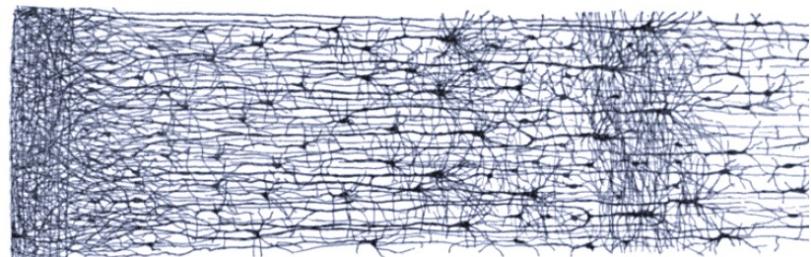
Bill Brock

¿Qué es el Deep Learning (DL)?

1



TRENDS in Ecology & Evolution



¹Dibujo de una lámina cortical, por Santiago Ramón y Cajal (dominio público)

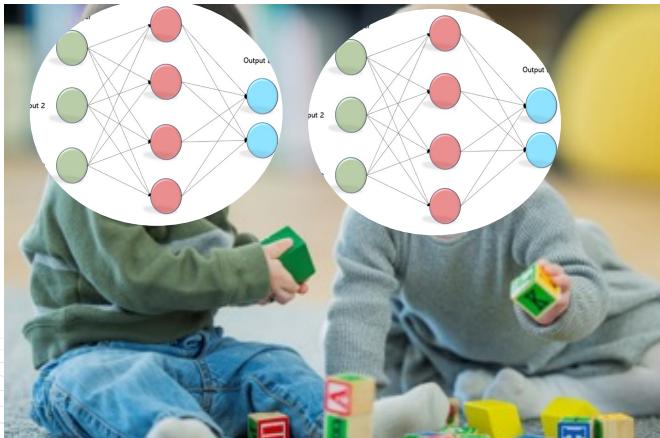
¿Qué es el Deep Learning (DL)?

Aprender con ejemplos



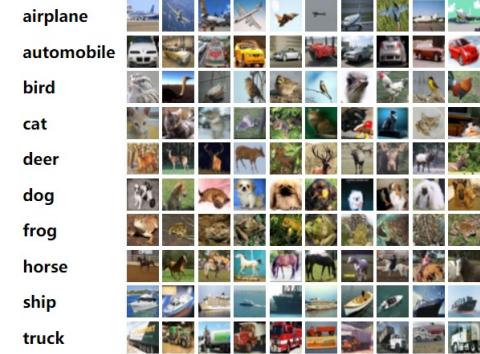
¿Qué es el Deep Learning (DL)?

Aprender con ejemplos



Datos
etiquetados

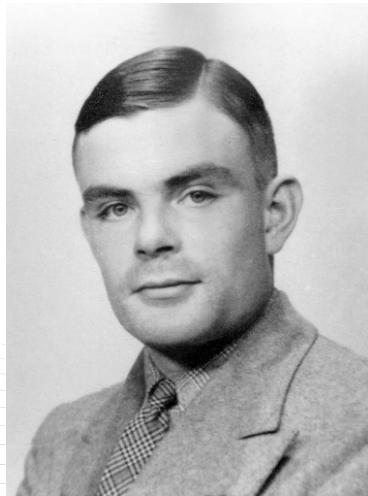
Capacidad
computacional



¿Qué es el Deep Learning (DL)?

Sus orígenes

1935



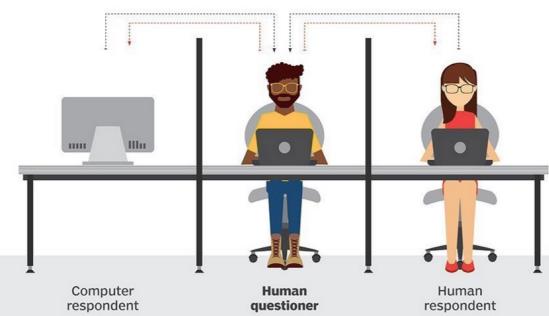
Alan M. Turing

Describió una máquina de computación abstracta que se "autoprogramaba" y mejoraba a medida que iba haciendo iteraciones (Máquina de Turing)

Test de Turing: Can computers really think?



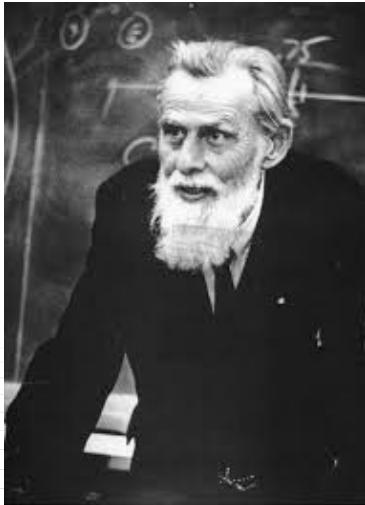
■ QUESTION TO RESPONDENTS ■ ANSWERS TO QUESTIONER



¿Qué es el Deep Learning (DL)?

Sus orígenes

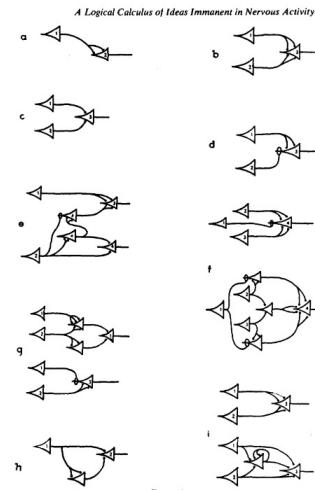
1943



Warren McCulloch
Neuroólogo



Walter Pitts
Matemático / Lógica



Bulletin of Mathematical Biology Vol. 52, No. 1/2, pp. 99-115, 1990.
Printed in Great Britain.

0092-8240/90\$3.00 + 0.00
Pergamon Press plc
Society for Mathematical Biology

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY*

■ WARREN S. McCULLOCH AND WALTER PITTS
University of Illinois, College of Medicine,
Department of Psychiatry at the Illinois Neuropsychiatric Institute,
University of Chicago, Chicago, U.S.A.

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

Entonces... ¿por qué ahora?

¿Qué es el Deep Learning (DL)?

Sus orígenes

Los inviernos de la IA

- 1970's: Primer **Invierno** de la IA
Falta de poder de computación
- 1985's: Segundo **Invierno** de la IA
Se creyó que la IA había "tocado fondo"
Se detuvieron las inversiones en investigación

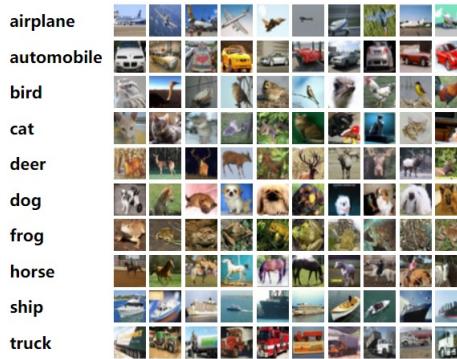


¿Qué es el Deep Learning (DL)?

Sus orígenes

90's (~) – Actualidad...

- Gran cantidad de **datos etiquetados**
- Mejora en la **capacidad computacional** desde los años 90 (Cloud, GPUs, TPUs...)
- Introducción de **mejoras** en los algoritmos y en los algoritmos de entrenamiento (ej. *Backpropagation, 1986*)



El pilar principal del DL: Redes Neuronales Artificiales (ANN)

1. Elementos principales de una Red Neuronal
2. El Perceptrón
3. Multi-layer perceptron (MLP)
4. DNN (Deep Neural Network)
5. ¿Qué es entrenar una red neuronal?
6. El algoritmo que lo cambió todo: *backpropagation*

El pilar principal del DL: Redes Neuronales Artificiales (ANN)

1. Elementos principales de una Red Neuronal
2. El Perceptrón
3. Multi-layer perceptron (MLP)
4. DNN (Deep Neural Network)
5. ¿Qué es entrenar una red neuronal?
6. El algoritmo que lo cambió todo: *backpropagation*

El pilar principal del DL: Redes Neuronales Artificiales (ANN)

1. Elementos principales de una Red Neuronal

Una red neuronal (ANN) es un modelo de Machine Learning que está formado por un conjunto de **neuronas (o unidades)**, dispuestas en capas y conectadas entre sí (entendemos conectadas en el sentido de que la información de salida de una neurona es la información de entrada de otra). Estas conexiones están moderadas por unos valores (pesos) que modelan el error del valor de salida de la última neurona (es decir, la predicción).

El objetivo de estos modelos es **aprender de manera autónoma** (modificando sus parámetros automáticamente).

Como los problemas complejos empiezan a resolverse cuando se unen muchas capas de neuronas (Deep neural network) estos algoritmos son el “core” del Deep Learning

El pilar principal del DL: Redes Neuronales Artificiales (ANN)

1. Elementos principales de una Red Neuronal

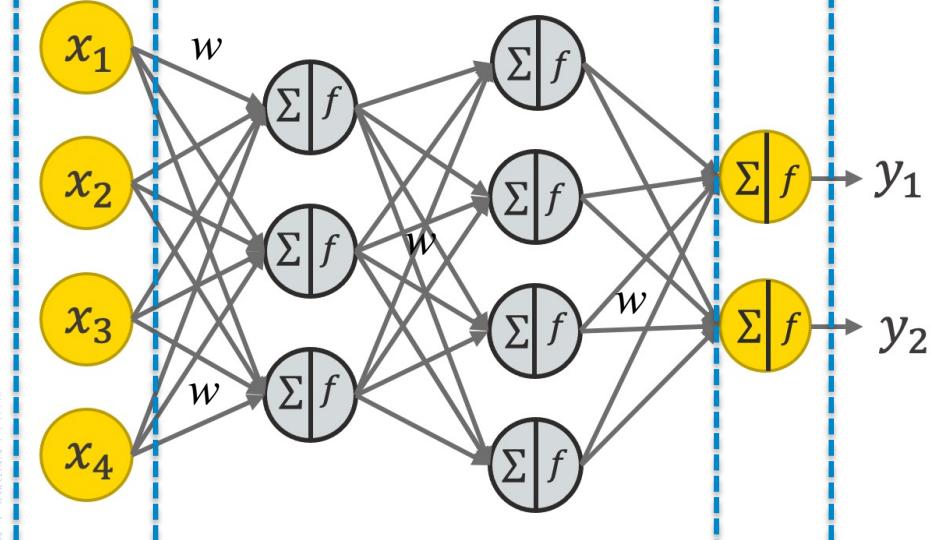
Una red neuronal (ANN) es un modelo de Machine Learning que está formado por un conjunto de **neuronas (o unidades)**, dispuestas en capas y conectadas entre sí

Estas conexiones están moderadas por unos valores (pesos) que modelan el error del valor de salida de la última neurona

Capa de entrada (input layer)

Capa de salida (output layer)

Capas ocultas (hidden layers)



Objetivo: **aprender de manera autónoma**
(modificando sus parámetros automáticamente)

Los problemas complejos empiezan a resolverse cuando se unen muchas capas de neuronas (Deep neural network) por lo que estos algoritmos son el **“core” del Deep Learning**

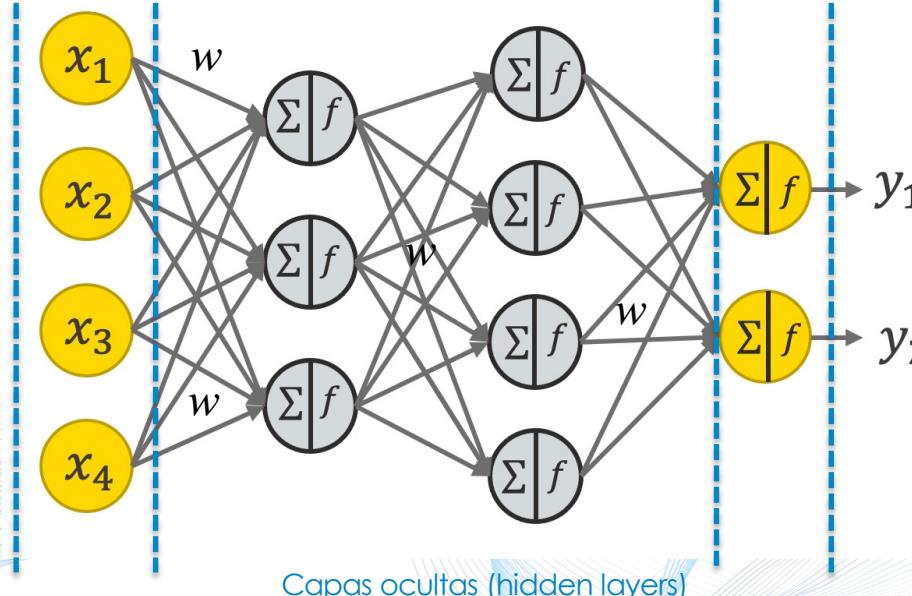
El pilar principal del DL: Redes Neuronales Artificiales (ANN)

1. Elementos principales de una Red Neuronal

La disposición de las capas y cómo estén conectadas las neuronas entre sí determina la **arquitectura de la red**

Capa de entrada (input layer)

Capa de salida (output layer)



Capas ocultas (hidden layers)

Vamos a ver qué son:

- **Pesos (w_{ij})**
- **Función de activación**

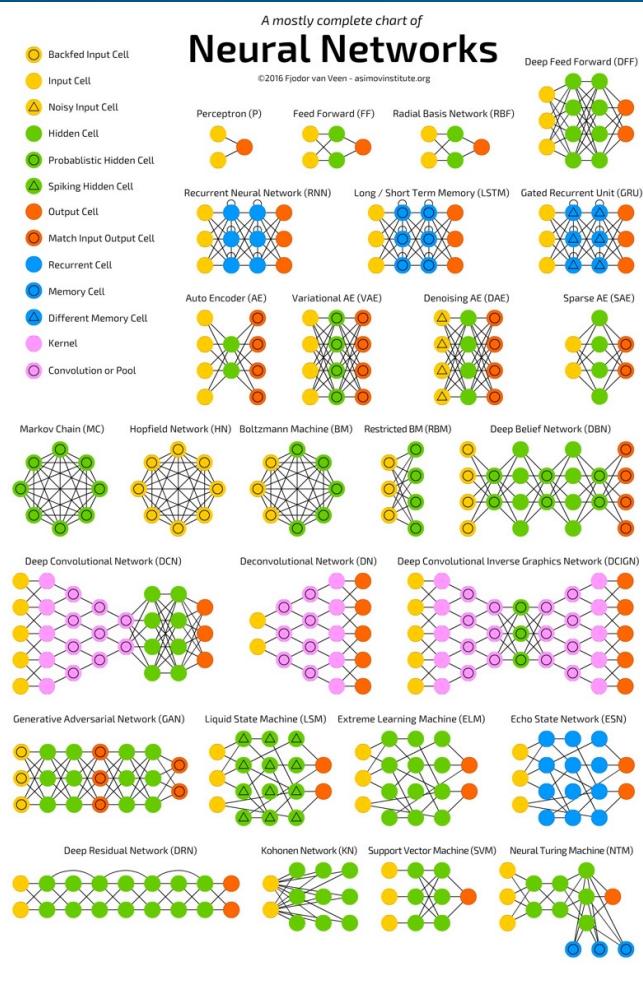
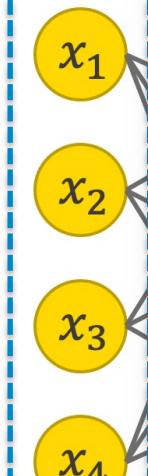
El pilar principal de la IA

1. Elementos principales

Capa de entrada (input layer)

La disposición de las capas y cómo estén conectadas las neuronas entre sí determina la **arquitectura de la red**

Empezemos mejor por el principio...



Artificiales (ANN)

Vamos a ver qué son:

- **Pesos (w_{ij})**
- **Función de activación**

El pilar principal del DL: Redes Neuronales

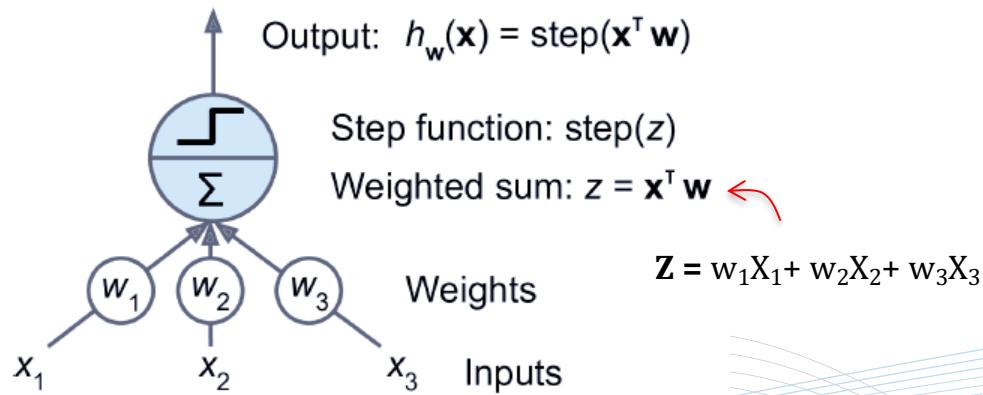
2. El Perceptrón

1957



Frank Rosenblatt

El perceptrón es la red más simple. Está basado en unidades llamadas **threshold logic unit (TLU)**



Ejemplo de función escalón: función de Heaviside, sigmoide, etc.

F. ROSENBLATT

Cornell Aeronautical Laboratory

El pilar principal del DL: Redes Neuronales

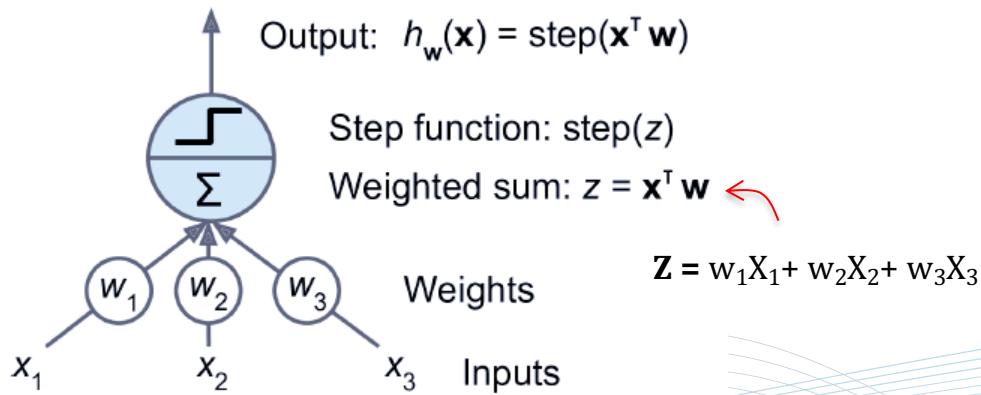
2. El Perceptrón

1957



Frank Rosenblatt

El perceptrón es la red más simple. Está basado en unidades llamadas **threshold logic unit (TLU)**



Ejemplo de función escalón: función
de Heaviside, sigmoide, etc.

El pilar principal del DL: Redes Neuronales Artificiales (ANN)

2. El Perceptrón

- El perceptrón está compuesto por un TLU o una capa de múltiples TLUs, con cada TLU conectada a todos los inputs
- **Fully Connected Layer (o dense layer):** todas las TLUs de una capa se conectan a todas las TLUs de la siguiente capa
- Outputs:

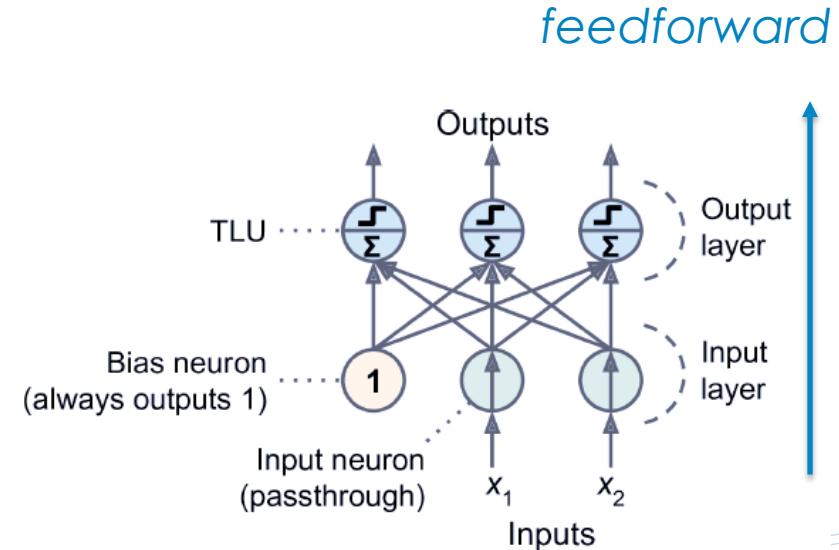
$$h_{W, b}(X) = \phi(XW + b)$$

X: matriz de inputs

W: matriz de pesos

b: bias

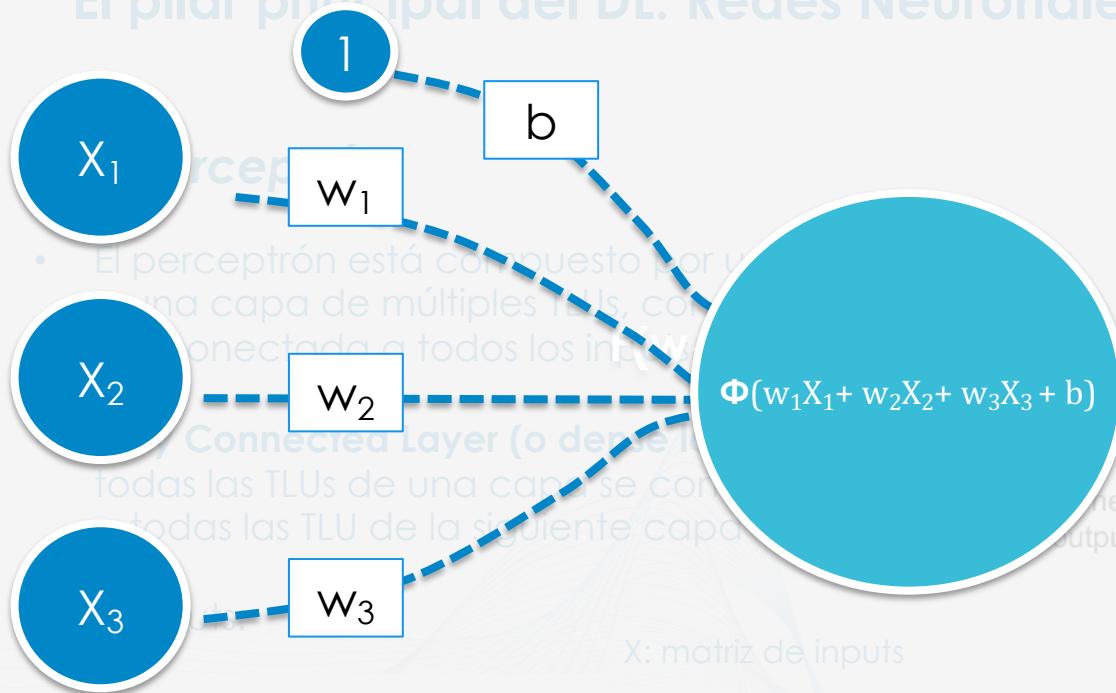
ϕ : función de activación



$$\text{salida} = \begin{cases} 1 & \text{si } \sum_i w_i \cdot x_i + b \geq 0 \\ 0 & \text{si } \sum_i w_i \cdot x_i + b < 0 \end{cases}$$

Φ: función de activación

El pilar principal del DL: Redes Neuronales Artificiales (ANN)



La función de activación añade una “deformación” no lineal al valor de salida, que depende del tipo de función de activación:

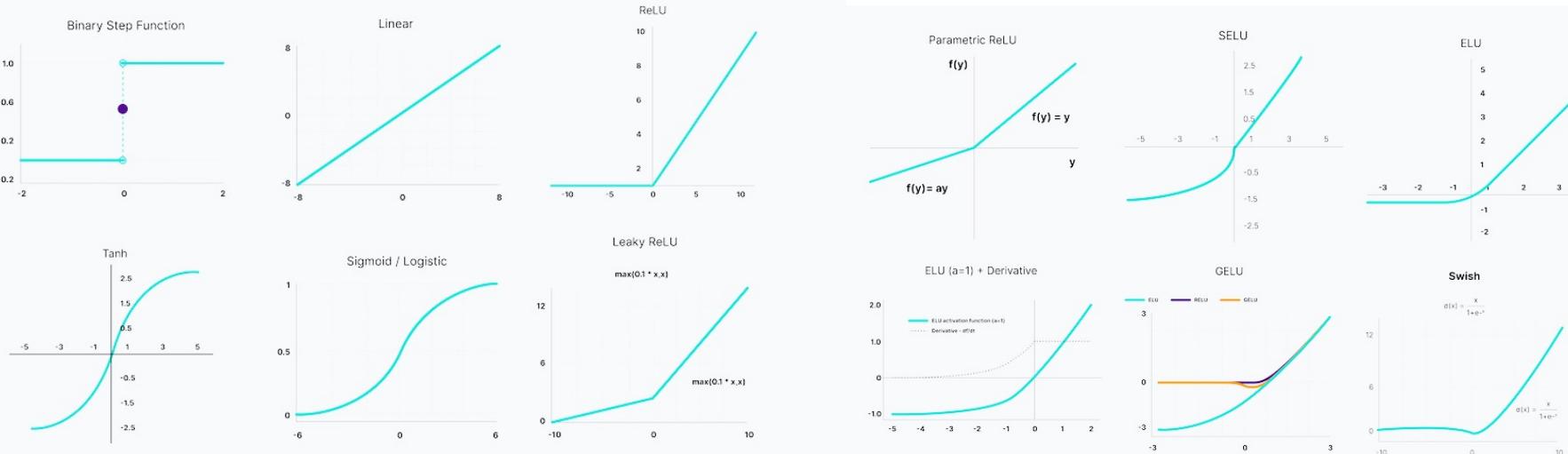
$$\begin{cases} 1 & \text{si } \sum_i w_i \cdot x_i + b \geq 0 \\ 0 & \text{si } \sum_i w_i \cdot x_i + b < 0 \end{cases}$$



Φ : función de activación

El pilar principal del DL: Redes Neuronales Artificiales (ANN)

Neural Network Activation Functions



$h_{w, b}(X) = \phi(XW + b)$
b: bias
Regresión, clasificación binaria, clasificación múltiple... (igual que en ML)

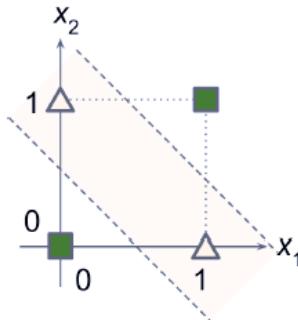
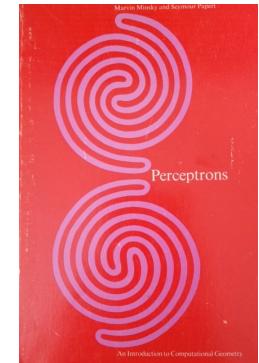
Φ : función de activación

$$\text{salida} = \begin{cases} 1 & \text{si } \sum_i w_i x_i + b \geq 0 \\ 0 & \text{si } \sum_i w_i x_i + b < 0 \end{cases}$$

El pilar principal del DL: Redes Neuronales Artificiales (ANN)

2. El Perceptrón

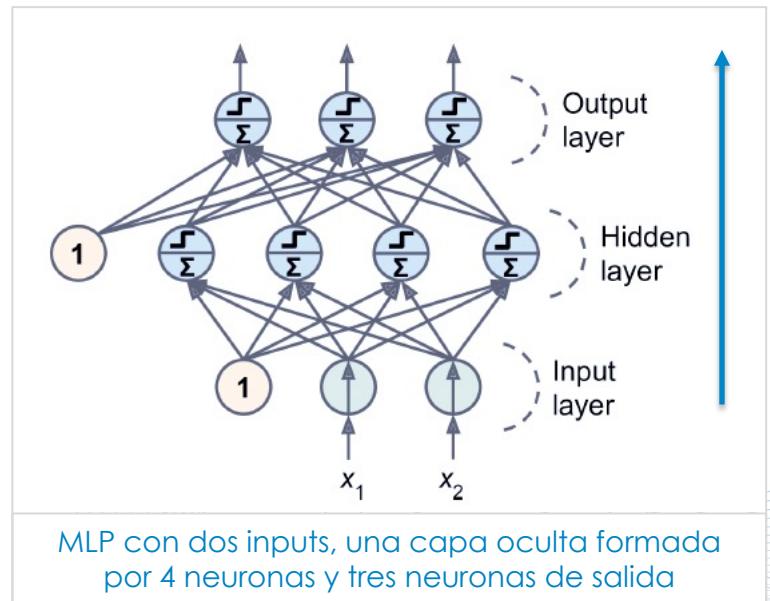
- 1969: Perceptrons, Marvin Minsky y Seymour Papert
- Problema: el perceptrón tiene serias limitaciones, incapaz de resolver problemas un poco complejos (ej.: puertas XOR)
- Comienza el primer **invierno de la Inteligencia Artificial** (hasta 1986 con la introducción del algoritmo **backpropagation**)



El pilar principal del DL: Redes Neuronales Artificiales (ANN)

3. Multi-Layer Perceptron (MLP)

- Un perceptrón formado por una capa de TLUs solo puede resolver problemas simples
- Uniendo varias capas formadas por múltiples TLUs se forma una **MLP** (“perceptrón” multicapa) (OJO, herencia histórica)
- Capas:
 - Input
 - Capas ocultas
 - Output (# neuronas en función del problema)

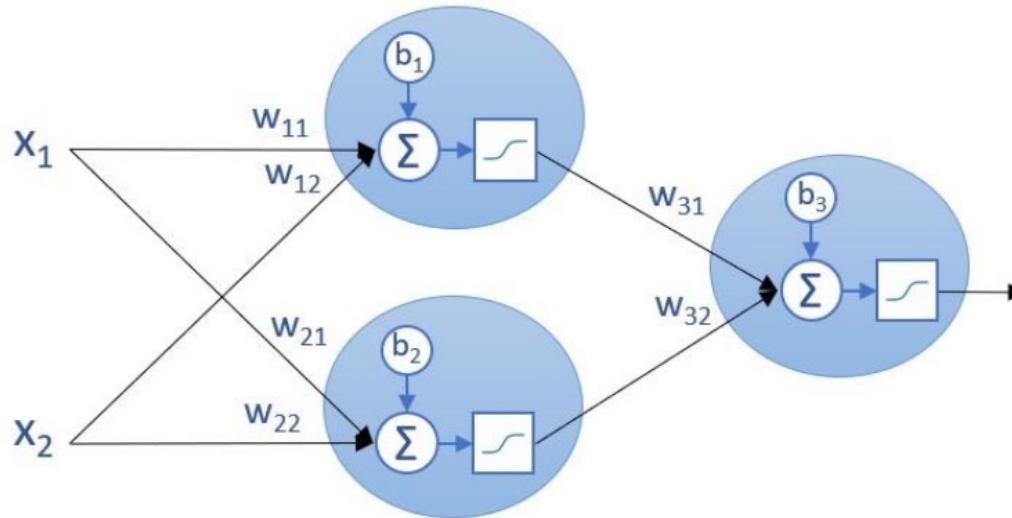


El pilar principal del DL: Redes Neuronales Artificiales (ANN)

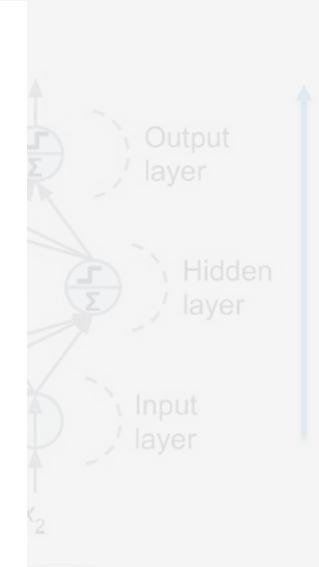
3. Multi-Layer

- El perceptrón
- Uniendo
TLUs se forma
(OJO, he
- Capas:

- Input
- Capas ocultas
- Output (# neuronas en función del problema)



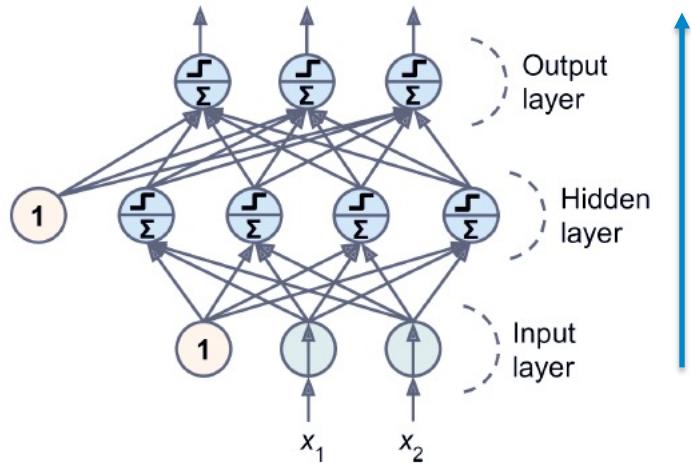
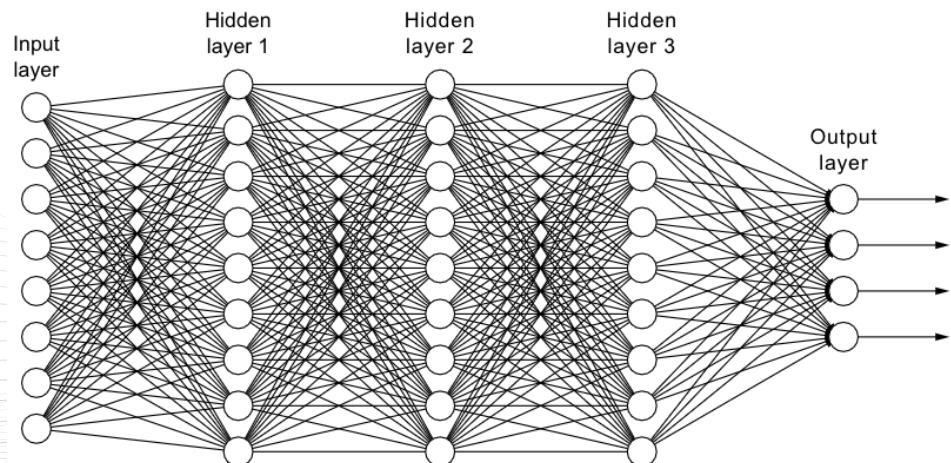
MLP con dos inputs, una capa oculta formada por 4 neuronas y tres neuronas de salida



El pilar principal del DL: Redes Neuronales Artificiales (ANN)

4. Deep Neural Network (DNN)

- Deep = “muchas” capas ocultas

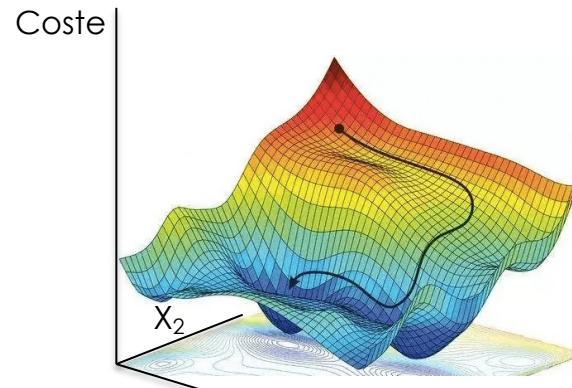


En los años 90' una red con dos capas ocultas era "Deep". Hoy en día es común ver redes de 12, 50 y hasta cientos de capas ocultas

El pilar principal del DL: Redes Neuronales Artificiales (ANN)

5. ¿Qué es entrenar una red neuronal?

- Buscar los parámetros de tu función (pesos (W) y bias (b)) que **minimizan la función de coste**
- La función de coste (C) es la función que determina el error entre el valor estimado y el valor real (al igual que en ML)
- El **descenso del gradiente** es un algoritmo iterativo que optimiza la función de coste = encuentra los mínimos de la función (valles) buscando el gradiente (que se obtiene por medio de derivadas)



x_1

x_2

Fundamentos de un modelo de Machine Learning

4. La función de coste

Descenso del gradiente (Gradient Descent)

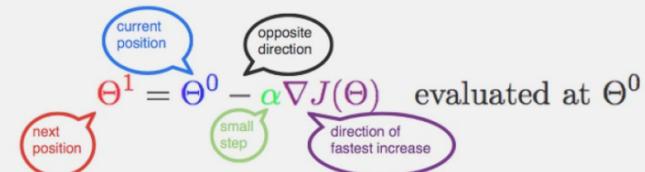
- Ejemplo: descenso del gradiente en regresión lineal simple

$$Y' = mX + b$$

Coste =

$$J_{m,b} = \frac{1}{N} \sum_{i=1}^N (Y'_i - Y_i)^2$$

$$J_{m,b} = \frac{1}{N} \sum_{i=1}^N (\text{Error}_i)^2$$



$\alpha = \text{learning rate}$

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Fórmula para el
descenso del
gradiente

$$\frac{\partial J}{\partial m} = 2 \cdot \text{Error} \cdot \frac{\partial}{\partial m} \text{Error}$$

$$\frac{\partial J}{\partial b} = 2 \cdot \text{Error} \cdot \frac{\partial}{\partial b} \text{Error}$$

$$\frac{\partial}{\partial m} \text{Error} = \frac{\partial}{\partial m} (Y' - Y) = x$$

$$\frac{\partial}{\partial b} \text{Error} = \frac{\partial}{\partial b} (Y' - Y) = 1$$

$$\frac{\partial J}{\partial m} = 2 \cdot \text{Error} \cdot X$$

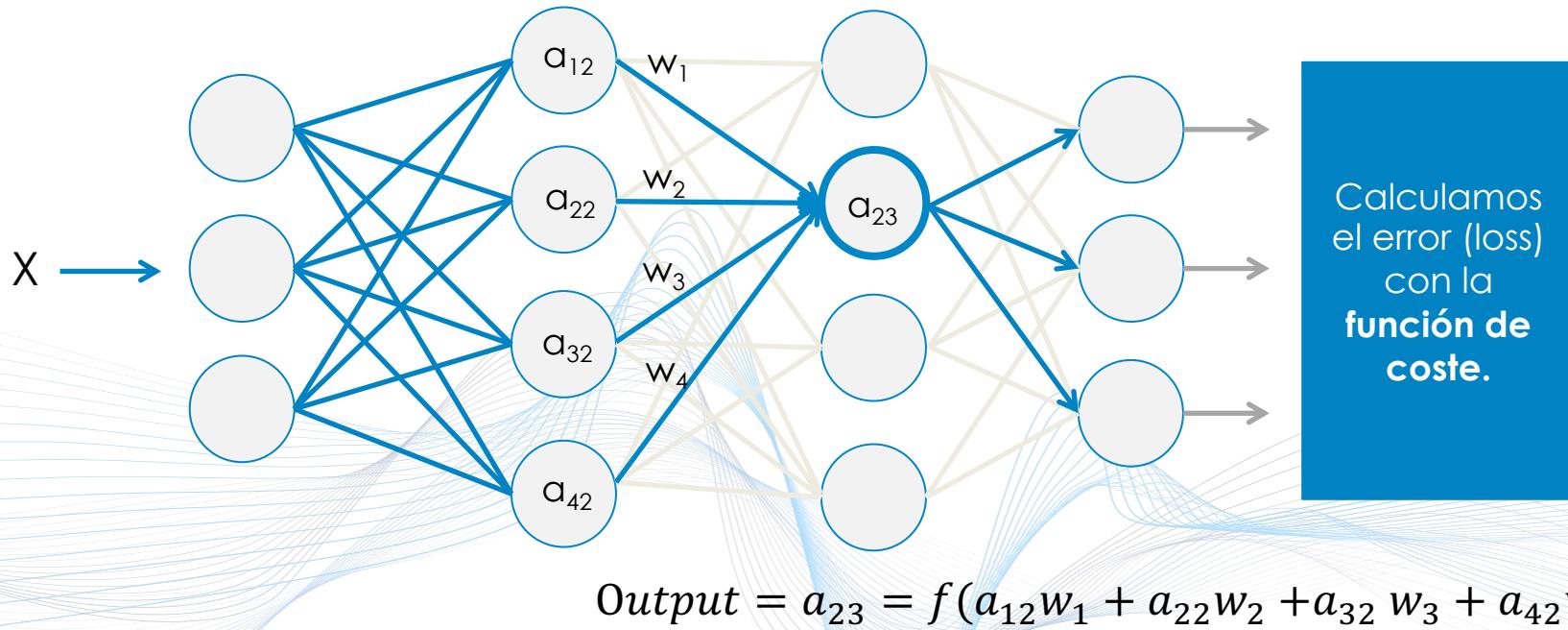
$$\frac{\partial J}{\partial b} = 2 \cdot \text{Error}$$

$$m' = m_0 - \alpha \cdot 2 \cdot (Y'_i - Y) \cdot X$$

$$b' = b_0 - \alpha \cdot 2 \cdot (Y'_i - Y)$$

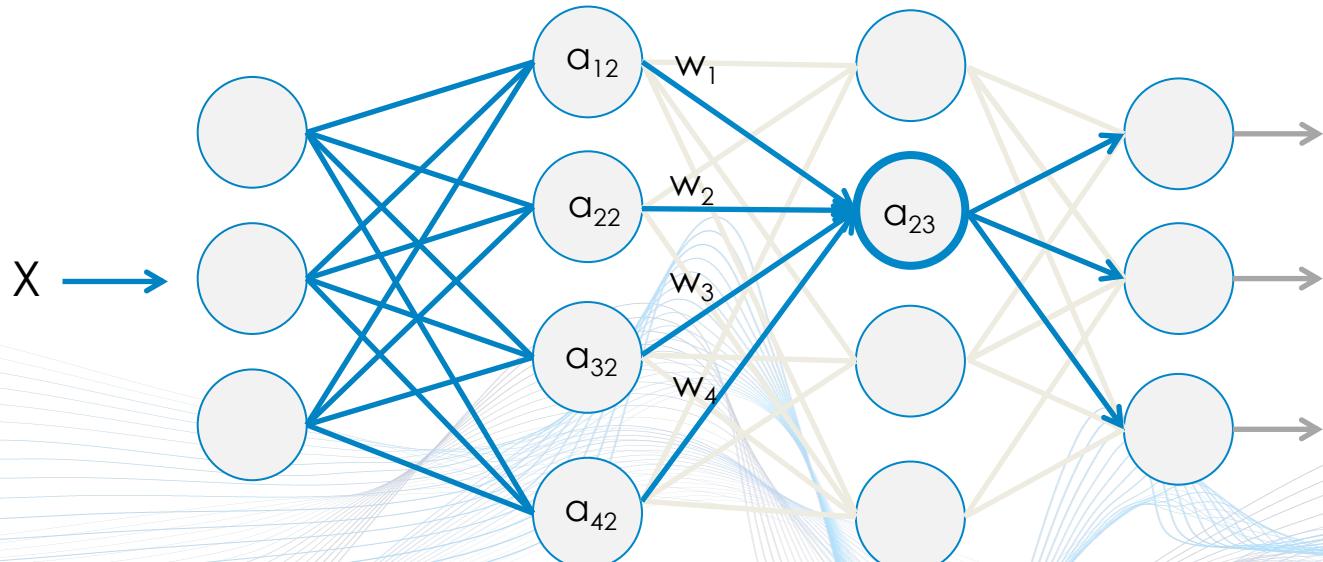
El pilar principal del DL: Redes Neuronales Artificiales (ANN)

5. ¿Qué es entrenar una red neuronal?



El pilar principal del DL: Redes Neuronales Artificiales (ANN)

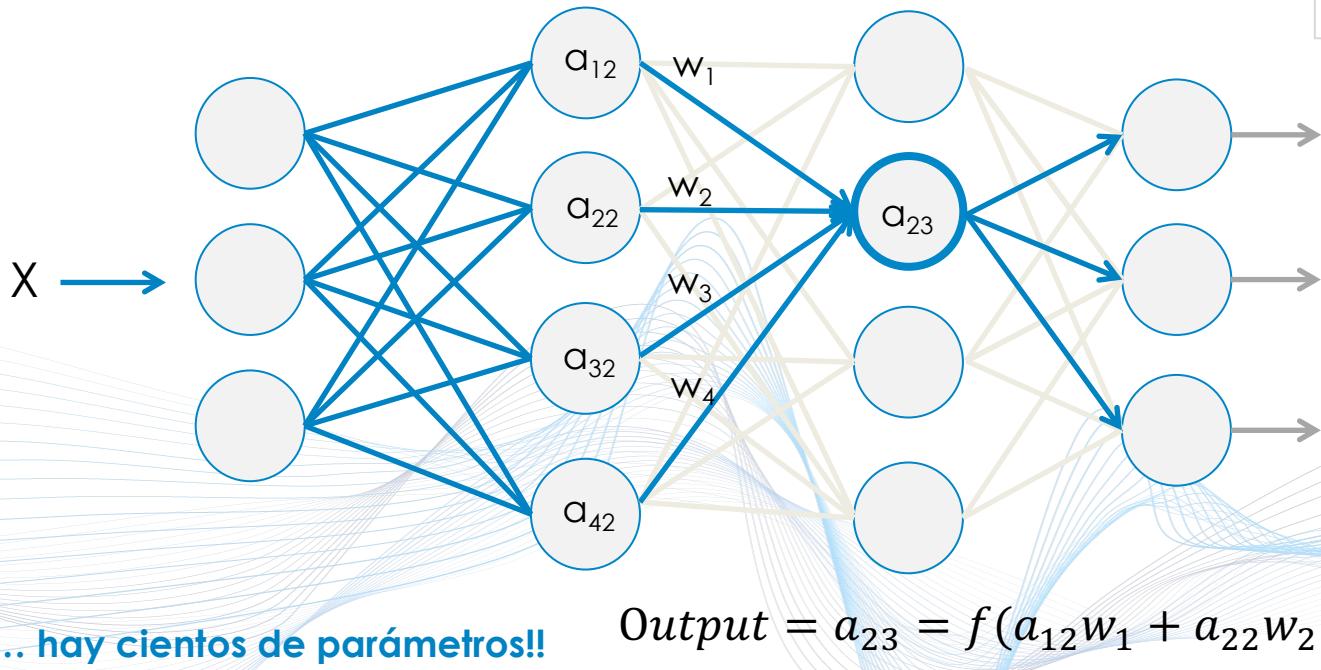
5. ¿Qué es entrenar una red neuronal?



Queremos minimizar ese error (la función de coste). ¿Cómo varia la f.d.c al variar los parámetros?

El pilar principal del DL: Redes Neuronales Artificiales (ANN)

5. ¿Qué es entrenar una red neuronal?



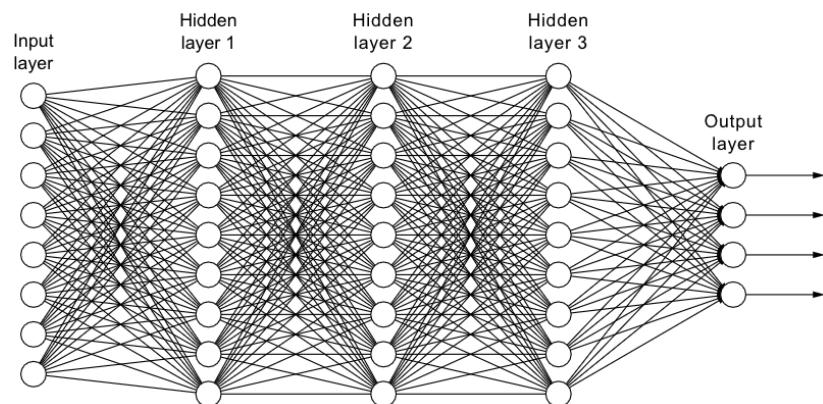
$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Calculamos sus derivadas con respecto a los pesos y modificamos los pesos con el descenso del gradiente

El pilar principal del DL: Redes Neuronales Artificiales (ANN)

5. ¿Qué es entrenar una red neuronal?

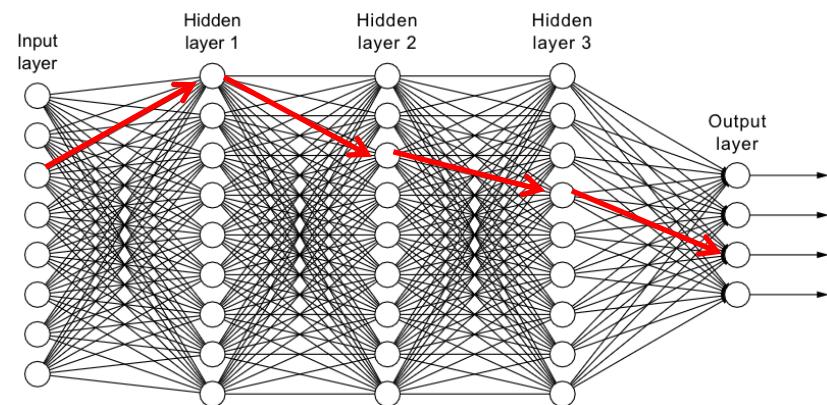
- Necesitamos un **método** que nos indique como cambia la **función de coste** (queremos minimizarla) cuando se varían los **pesos w** de la red.
- Antiguamente, este cálculo se hacía por fuerza bruta: los pesos se modificaban desde el input hasta el output buscando la combinación que diese el valor más bajo de la función de coste
- Se requería una capacidad computacional muy alta, por lo que el interés por las redes neuronales se vino abajo.



El pilar principal del DL: Redes Neuronales Artificiales (ANN)

5. ¿Qué es entrenar una red neuronal?

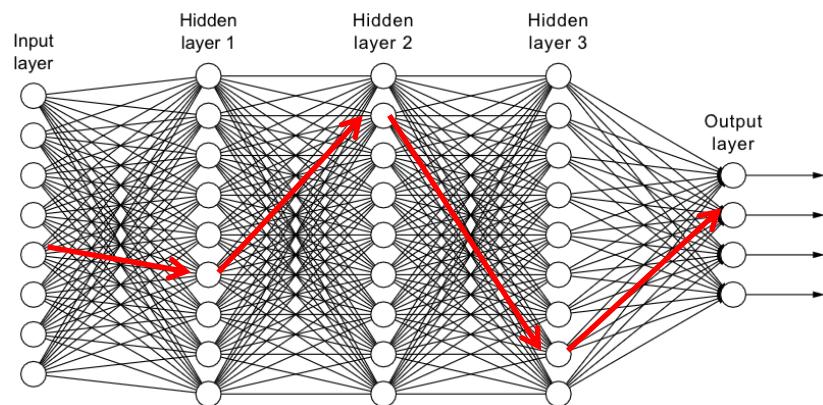
- Necesitamos un **método** que nos indique como cambia la **función de coste** (queremos minimizarla) cuando se varían los **pesos w** de la red.
- Antiguamente, este cálculo se hacía por fuerza bruta: los pesos se modificaban desde el input hasta el output buscando la combinación que diese el valor más bajo de la función de coste
- Se requería una capacidad computacional muy alta, por lo que el interés por las redes neuronales se vino abajo.



El pilar principal del DL: Redes Neuronales Artificiales (ANN)

5. ¿Qué es entrenar una red neuronal?

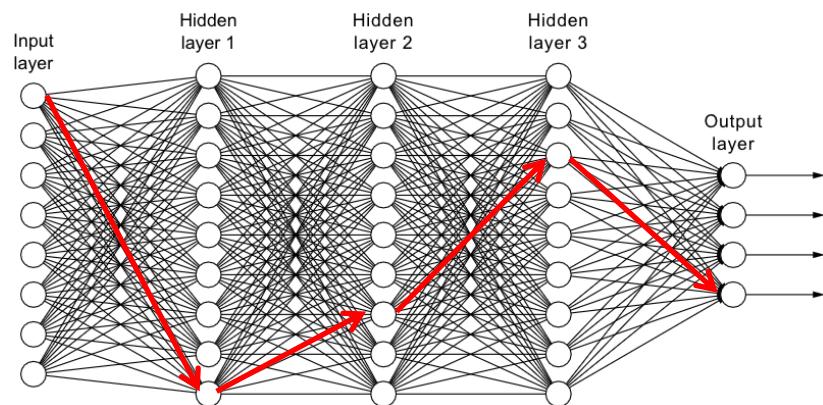
- Necesitamos un **método** que nos indique como cambia la **función de coste** (queremos minimizarla) cuando se varían los **pesos w** de la red.
- Antiguamente, este cálculo se hacía por fuerza bruta: los pesos se modificaban desde el input hasta el output buscando la combinación que diese el valor más bajo de la función de coste
- Se requería una capacidad computacional muy alta, por lo que el interés por las redes neuronales se vino abajo.



El pilar principal del DL: Redes Neuronales Artificiales (ANN)

5. ¿Qué es entrenar una red neuronal?

- Necesitamos un **método** que nos indique como cambia la **función de coste** (queremos minimizarla) cuando se varían los **pesos w** de la red.
- Antiguamente, este cálculo se hacía por fuerza bruta: los pesos se modificaban desde el input hasta el output buscando la combinación que diese el valor más bajo de la función de coste
- Se requería una capacidad computacional muy alta, por lo que el interés por las redes neuronales se vino abajo.



El pilar principal del DL: Redes Neuronales Artificiales (ANN)

6. El algoritmo que lo cambió todo: *backpropagation*

1986



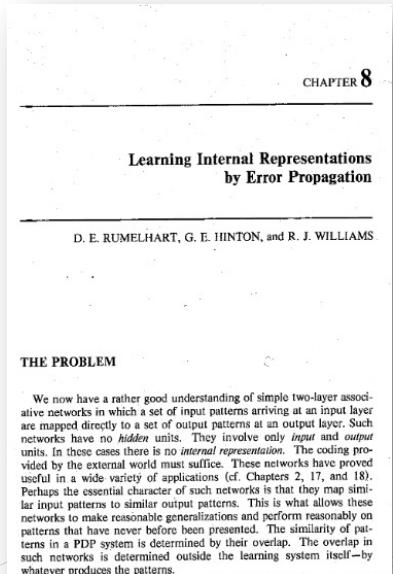
David Rumelhart



Geoffrey Hinton



Ronald Williams

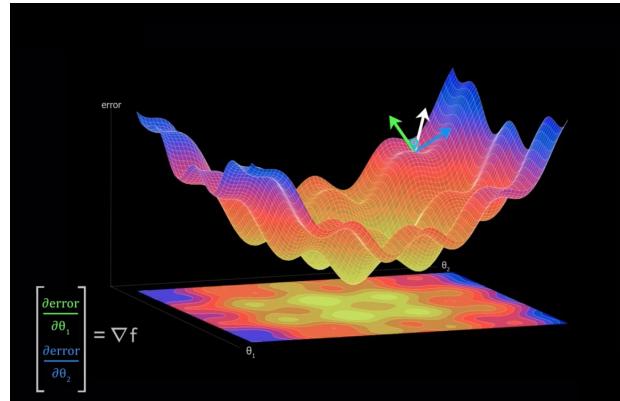


El DL empieza a utilizarse de verdad con la introducción del algoritmo de entrenamiento “backpropagation” (retropropagación)

El pilar principal del DL: Redes Neuronales Artificiales (ANN)

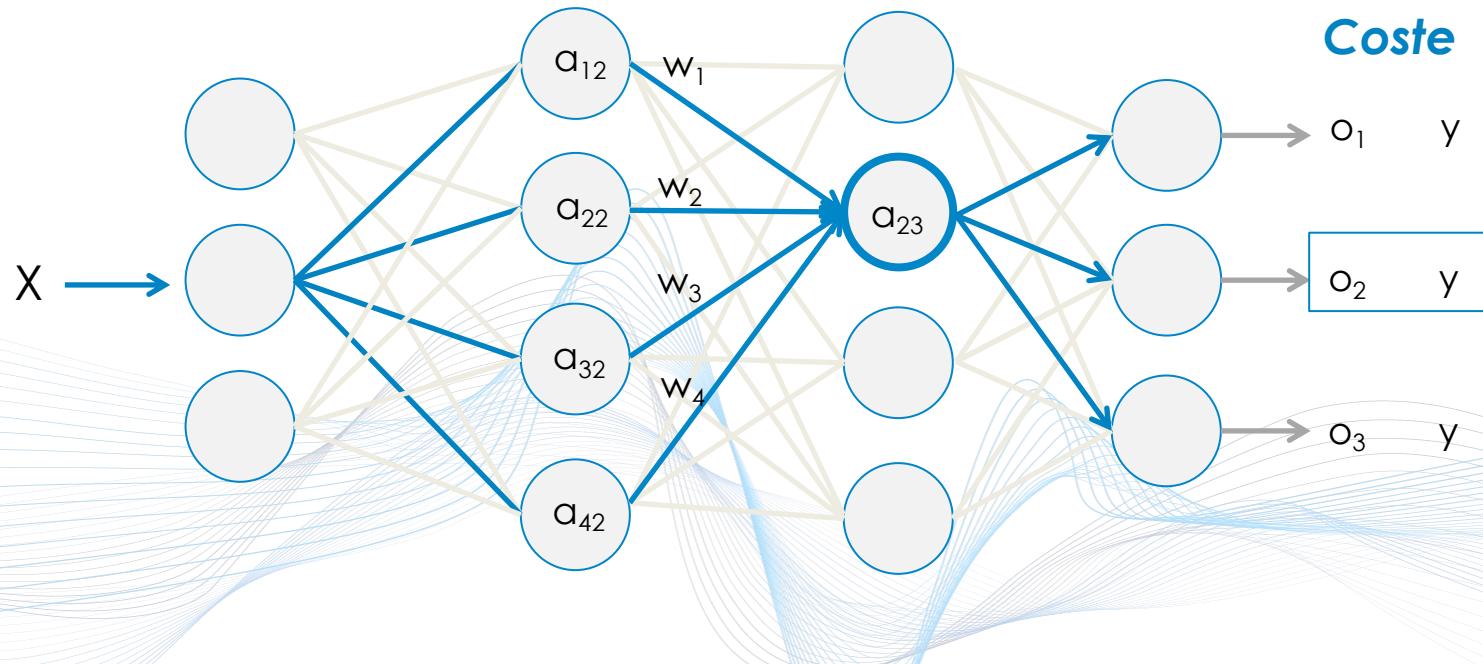
6. El algoritmo que lo cambió todo: *backpropagation*

- Con el algoritmo de *backpropagation* **calculamos el gradiente** de la función de coste, el cual necesitamos para encontrar los pesos que la hagan mínima, utilizando el descenso del gradiente
- Pero a diferencia del cálculo a fuerza bruta, no se empieza desde la primera capa hasta las últimas, si no de la última capa, hasta las primeras.
- Cuando se obtiene una predicción, el error se propaga desde las últimas capas hasta las primeras ("hacia atrás"), modificando los pesos y bias



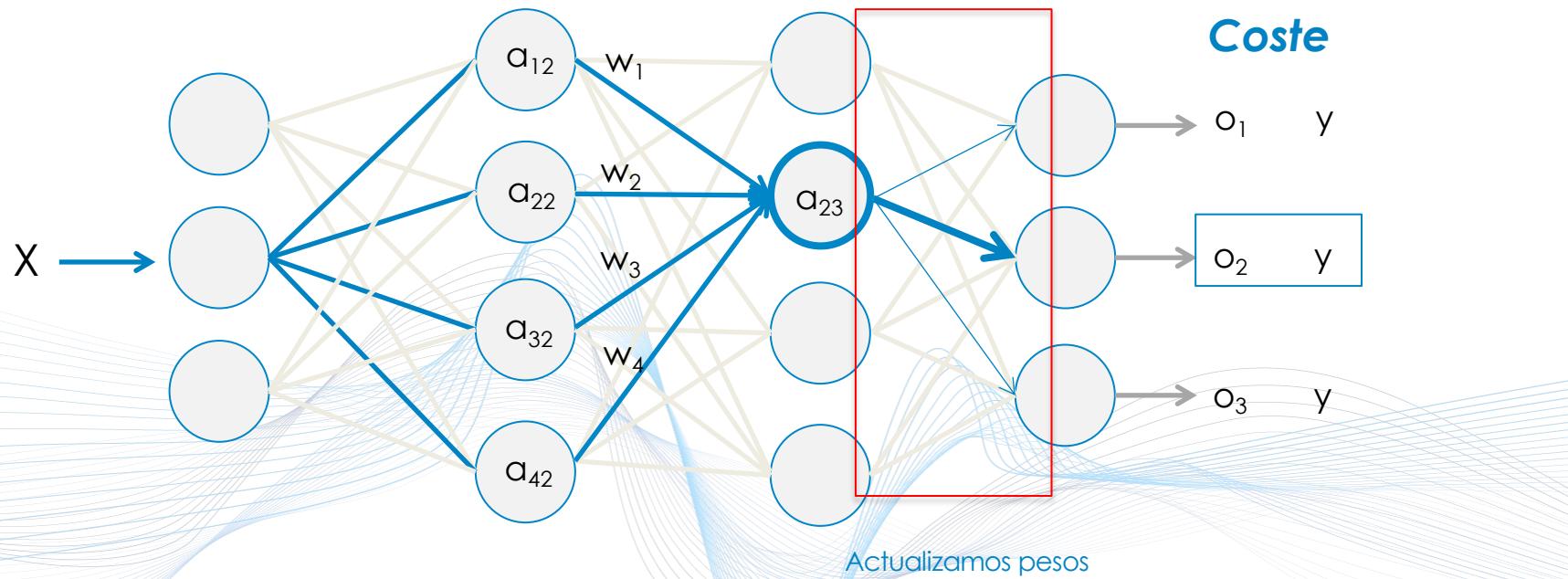
El pilar principal del DL: Redes Neuronales Artificiales (ANN)

6. El algoritmo que lo cambió todo: backpropagation



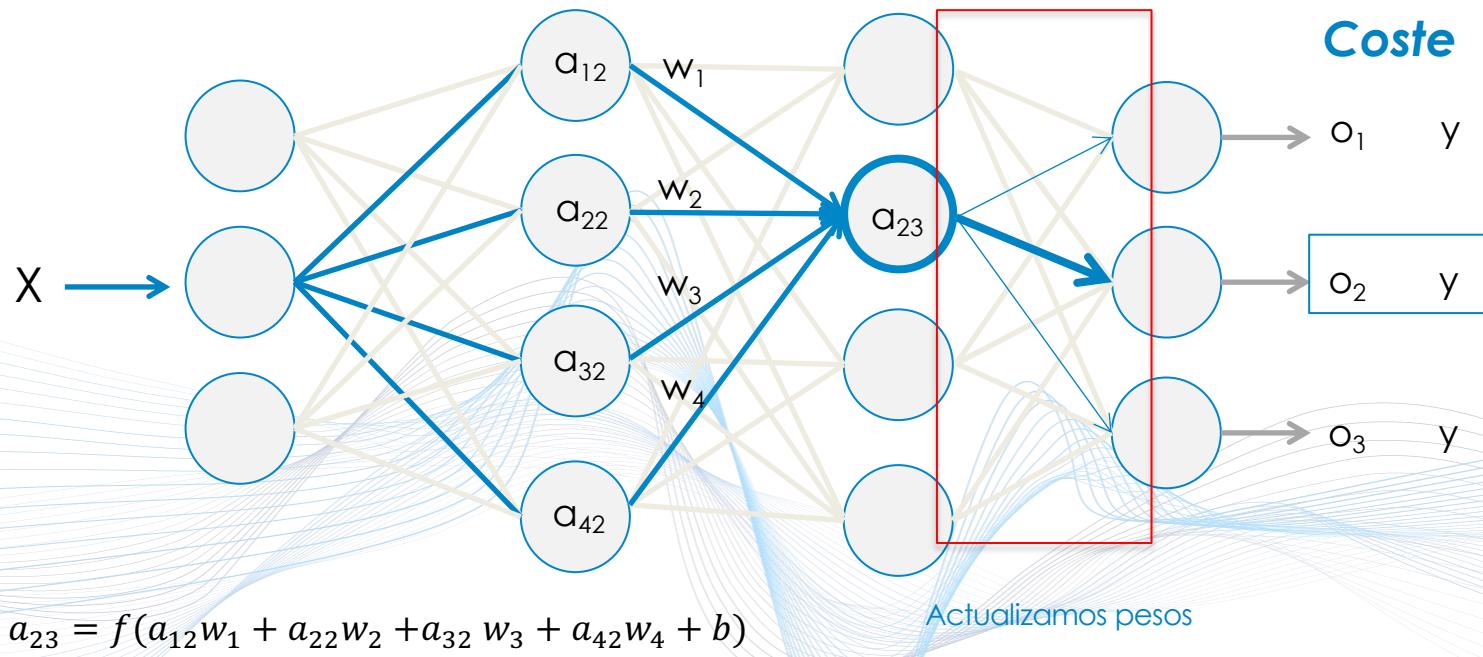
El pilar principal del DL: Redes Neuronales Artificiales (ANN)

6. El algoritmo que lo cambió todo: backpropagation



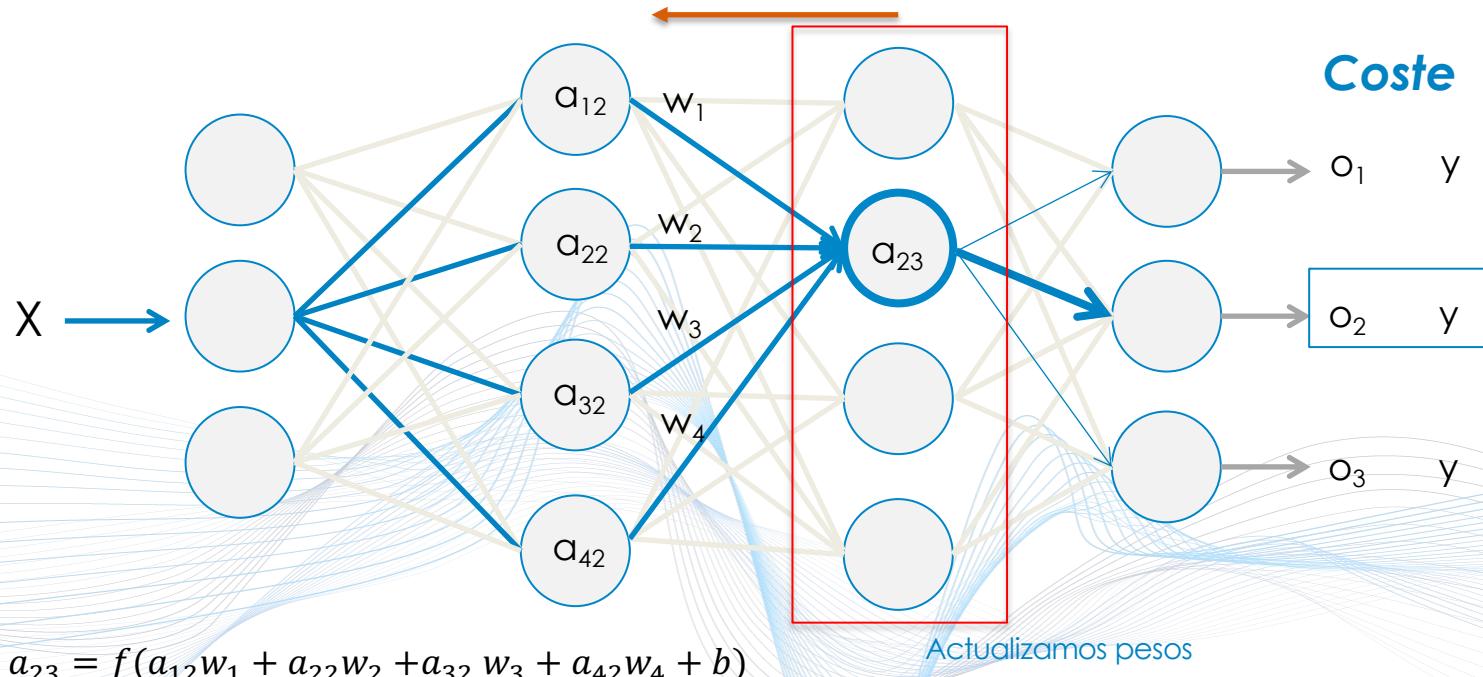
El pilar principal del DL: Redes Neuronales Artificiales (ANN)

6. El algoritmo que lo cambió todo: backpropagation



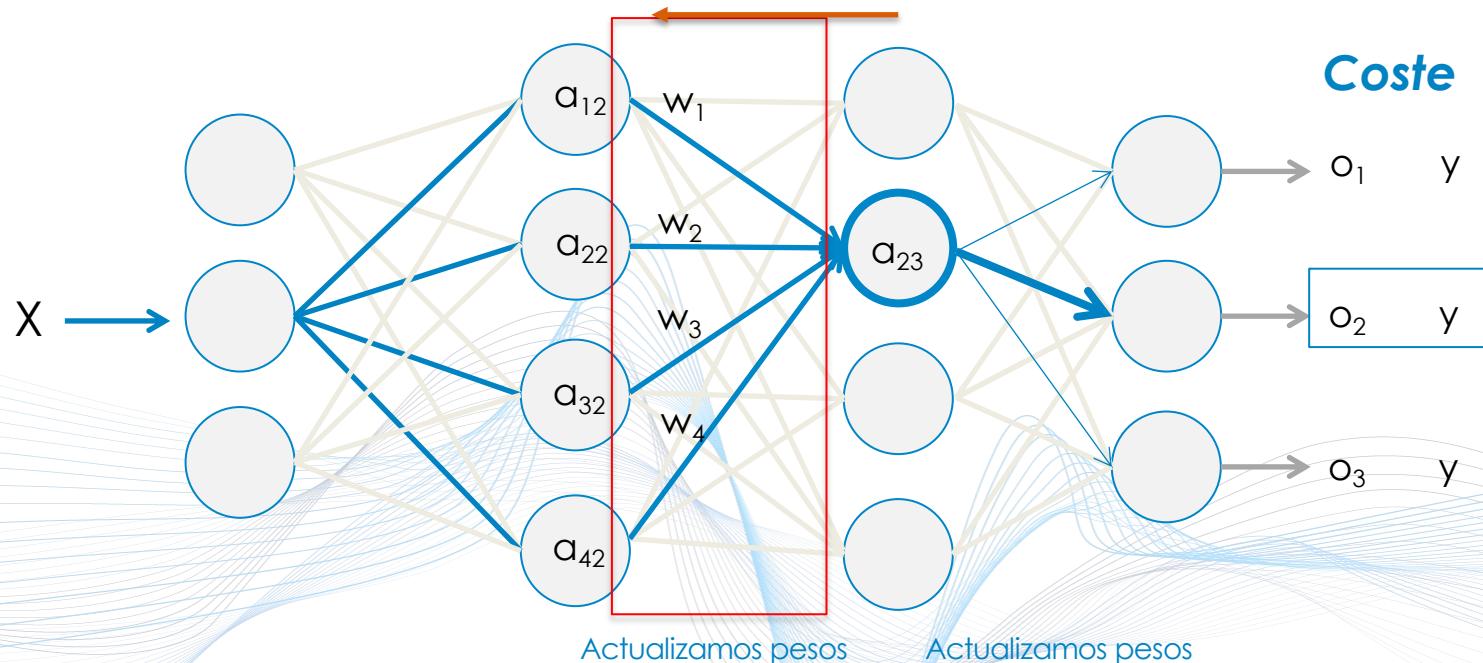
El pilar principal del DL: Redes Neuronales Artificiales (ANN)

6. El algoritmo que lo cambió todo: backpropagation



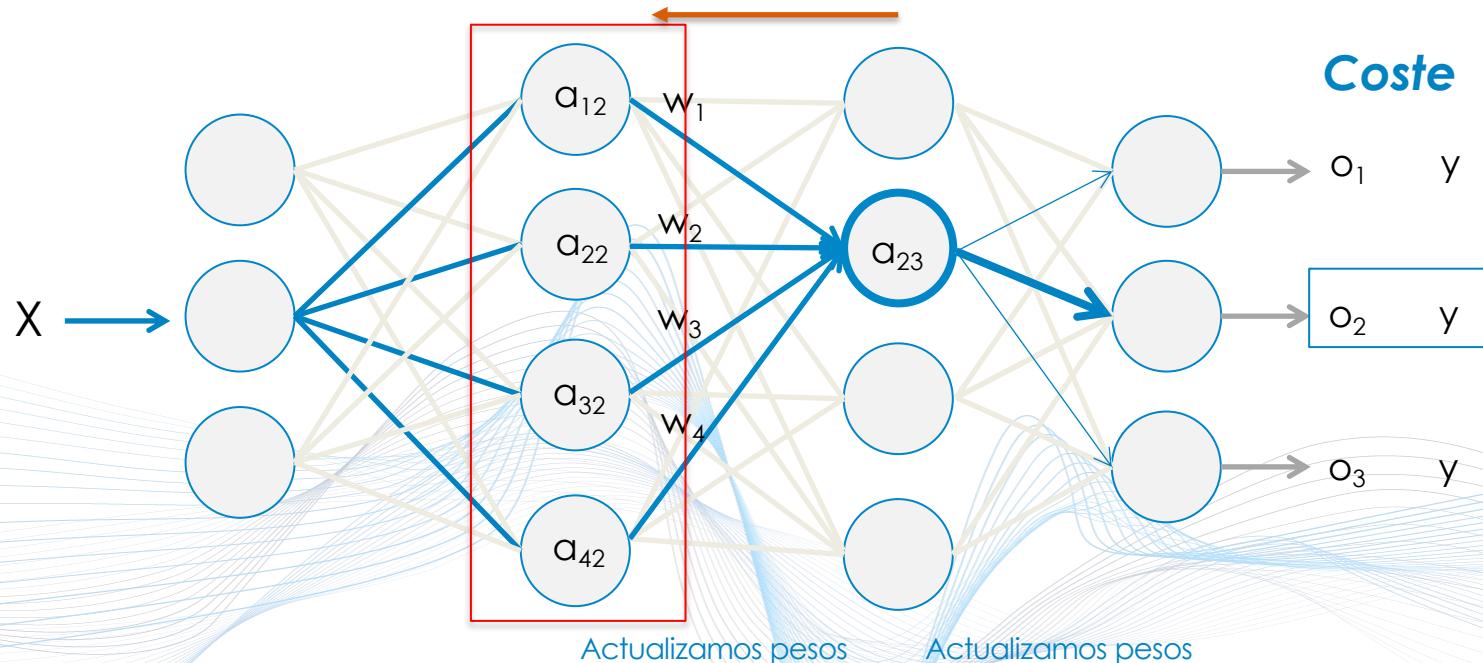
El pilar principal del DL: Redes Neuronales Artificiales (ANN)

6. El algoritmo que lo cambió todo: backpropagation



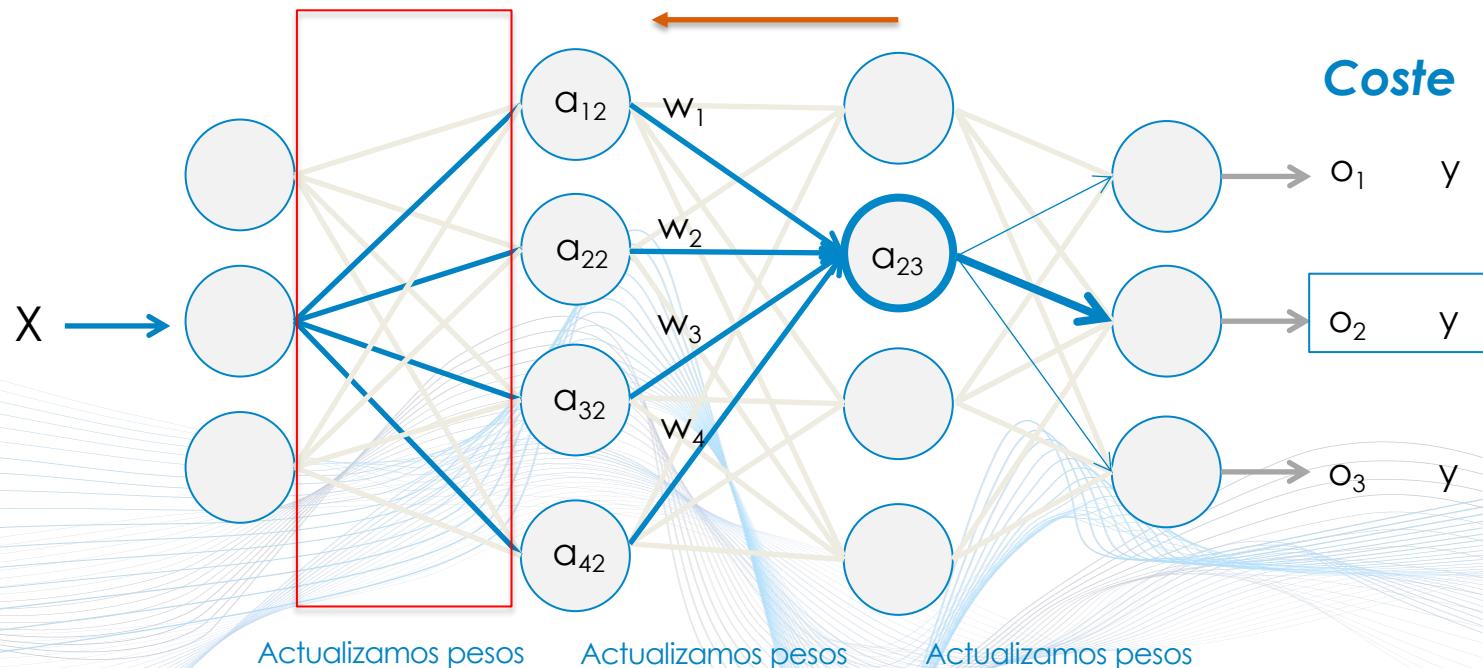
El pilar principal del DL: Redes Neuronales Artificiales (ANN)

6. El algoritmo que lo cambió todo: backpropagation



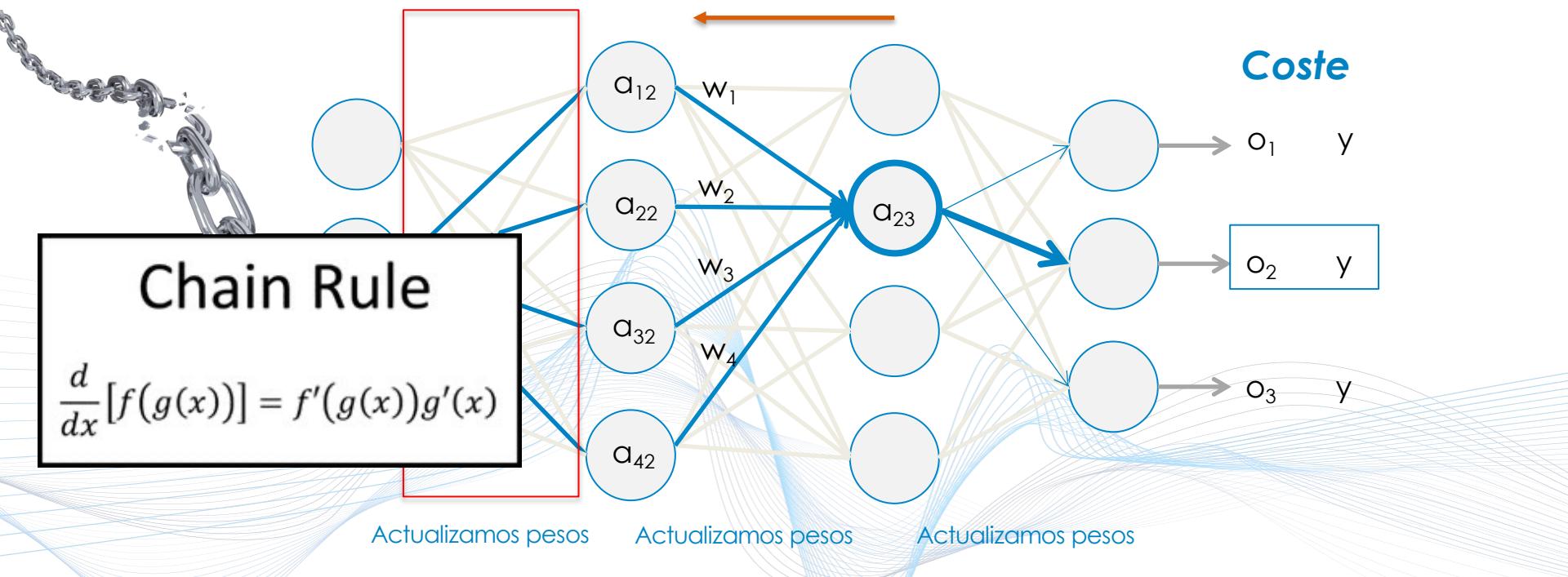
El pilar principal del DL: Redes Neuronales Artificiales (ANN)

6. El algoritmo que lo cambió todo: backpropagation



El pilar principal del DL: Redes Neuronales Artificiales (ANN)

6. El algoritmo que lo cambió todo: backpropagation



El pilar principal del DL: Redes Neuronales Artificiales (ANN)

6. El algoritmo que lo cambió todo: *backpropagation*

Ventajas:

- Rápido, sencillo y fácil de analizar y programar
- Aparte del número de entradas, no requiere ningún parámetro de ajuste
- Método flexible
- Método estandarizado y que funciona de forma muy eficiente.

Desventajas

- Hace que la red sea muy sensible al ruido.
- No se puede utilizar con “mini-batches”

El pilar principal del DL: Redes Neuronales Artificiales (ANN)

7. Parámetros que debemos modificar

- Número de **capas ocultas** (se debe utilizar *transfer learning*)
- Número de **neuronas en cada capa oculta**, ya que el de las de entrada y salida viene determinado por el problema (ej. MNIST requiere $28 \times 28 = 748$ neuronas de entrada y 10 de salida (0-9))
- **Función de activación** (sigmoide, ReLu, tanH...)
- Otros parámetros:
 - **Batch**: numero de datos de entrada extraído del training set
 - **Epoch**: el numero de veces que el algoritmo evalúa el training set
 - **Learning rate**: ratio al que itera la función de coste
 - **Optimizador**: función que optimiza la función de coste (gradient descent, SGD, Adam, Nadam...)
 - **Pesos** (w) iniciales

Tipos de redes neuronales

- 1. Arquitectura de una red neuronal**
- 2. Redes neuronales más conocidas:**
 - 2.1. Feedforward neural network (ANN)**
 - 2.2. Redes neuronales recurrentes (RNN)**
 - 2.3. Redes Neuronales Convolucionales (CNN)**
- 3. Arquitecturas más conocidas**

Tipos de redes neuronales

- 1. Arquitectura de una red neuronal**
- 2. Redes neuronales más conocidas:**
 - 2.1. Feedforward neural network (ANN)**
 - 2.2. Redes neuronales recurrentes (RNN)**
 - 2.3. Redes Neuronales Convolucionales (CNN)**
- 3. Arquitecturas más conocidas**

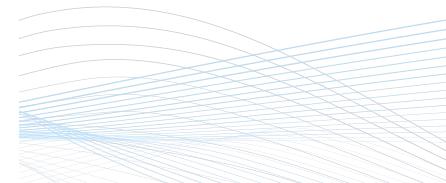
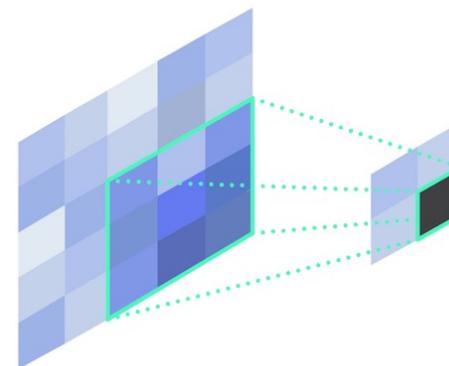
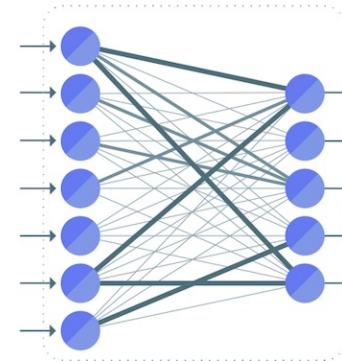
Tipos de redes neuronales

1. Arquitectura de una red neuronal

La **arquitectura de la red neuronal** está definida por la disposición de las capas y por cómo se conectan entre sí

Tipos de capas:

- Fully connected layer
- Convolution layer
- Deconvolution layer
- Recurrent layer
- ...

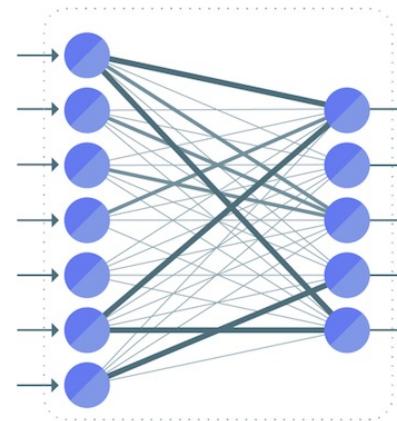


Tipos de redes neuronales

2. Redes neuronales más conocidas

2.1. Feedforward neural network (ANN)

- La información fluye en una sola dirección
- Ventajas
 - Son sencillas
 - Buenas para analizar datos **tabulares**
- Desventajas
 - Con imágenes el número de parámetros que hay que modificar se incrementa con el tamaño de la imagen
 - Sufren del problema *Vanishing and Exploding Gradient*

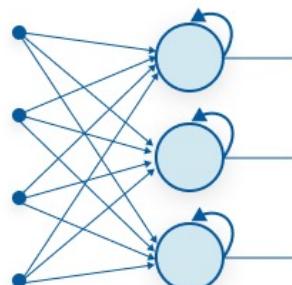


Tipos de redes neuronales

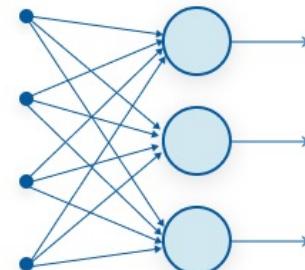
2. Redes neuronales más conocidas

2.2. Redes Neuronales Recurrentes (RNN)

- El output de una neurona puede volver a ser el input de una neurona anterior (o de la misma neurona)
- Ventajas
 - Permiten analizar datos **secuenciales**
 - Son redes con “memoria”
 - Muy utilizadas en el Procesamiento del Lenguaje Natural (NLP)
- Desventajas
 - Alta capacidad de computo
 - También sufren de “vanishing gradient”



Recurrent Neural Network



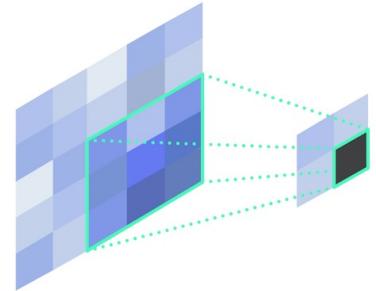
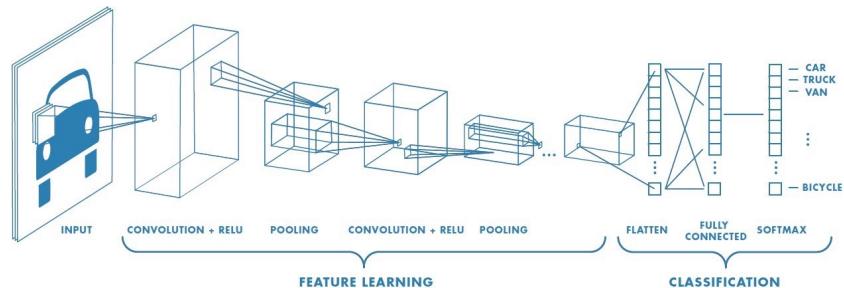
Feed-Forward Neural Network

Tipos de redes neuronales

2. Redes neuronales más conocidas

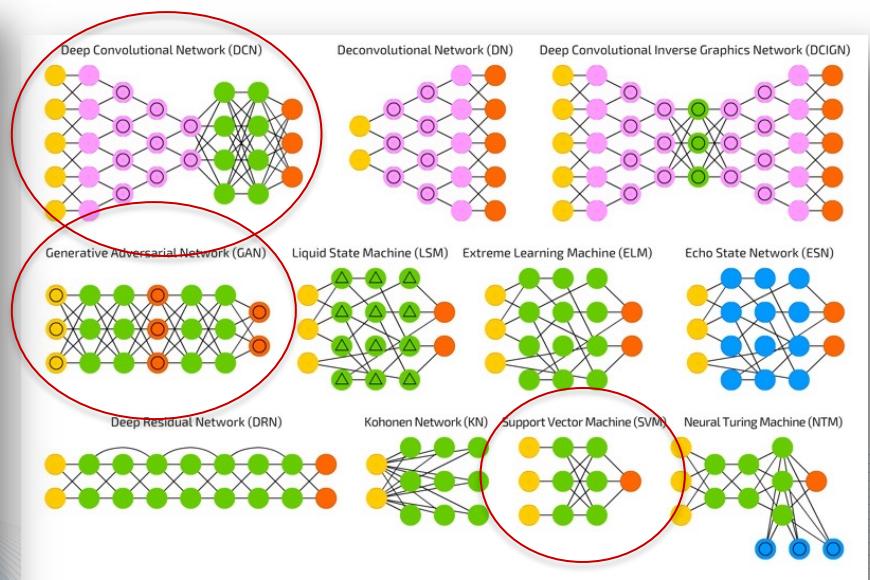
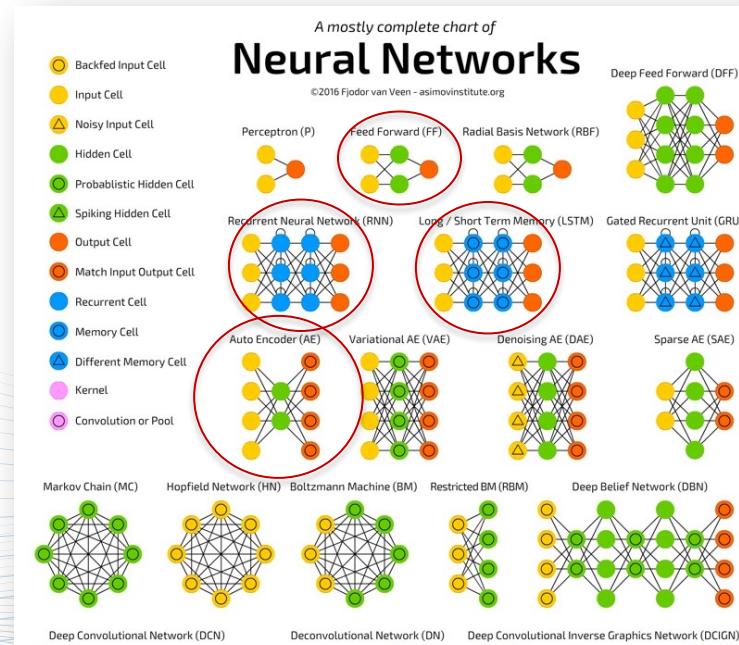
2.3. Redes Neuronales Convolucionales (CNN)

- Las capas se unen por operaciones de convolución
- Los “pesos” de estas funciones son **filtros**
- Ventajas
 - Permiten analizar **imágenes**
 - Buenas en **NLP**
- Desventajas
 - Necesitan muchos datos para funcionar bien
 - Necesitan capacidades de computación altas (GPU)



Tipos de redes neuronales

3. Arquitecturas conocidas



Convolutional Neural Networks (CNNs)

- 1. Cómo funciona el cortex visual**
- 2. Capas de una CNN:**
 2. 1. Convolución
 2. 2. Pooling
- 3. Arquitecturas: ResNet, LeNet-5, AlexNet**
- 4. Problemas que resuelven:**
 4. 1. Clasificación y localización
 4. 2. Detección de objetos (YOLO)
 4. 3. Segmentación semántica

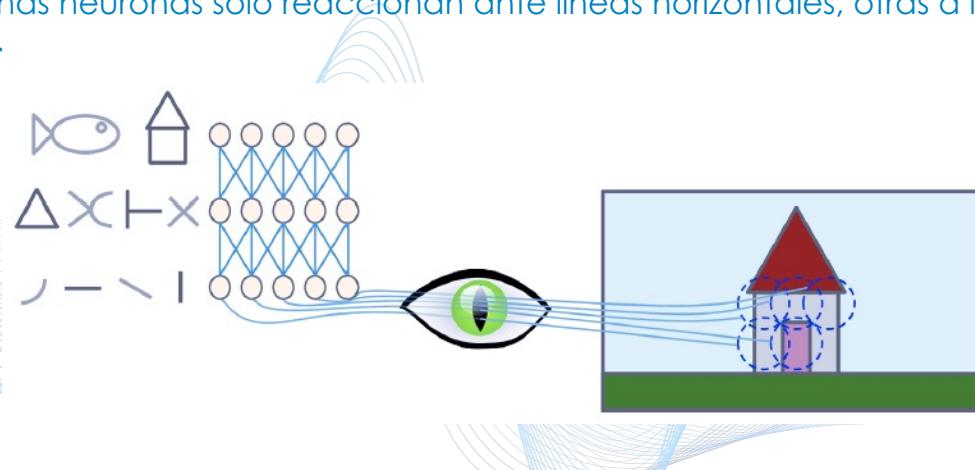
Convolutional Neural Networks (CNNs)

- 1. Cómo funciona el cortex visual**
- 2. Capas de una CNN:**
 2. 1. Convolución
 2. 2. Pooling
- 3. Arquitecturas: ResNet, LeNet-5, AlexNet**
- 4. Problemas que resuelven:**
 4. 1. Clasificación y localización
 4. 2. Detección de objetos (YOLO)
 4. 3. Segmentación semántica

Convolutional Neural Networks (CNNs)

1. Cómo funciona el cortex visual

- Las CNNs están inspiradas en el funcionamiento del córtex visual
- Las neuronas del córtex visual solo reaccionan a estímulos visuales dentro de una región concreta del campo de visión (llamado **campo receptivo local**)
- Además, algunas neuronas solo reaccionan ante líneas horizontales, otras a líneas verticales, otras a círculos, etc.



Convolutional Neural Networks (CNNs)

1. Cómo funciona el cortex visual

1980



Kunihiko Fukushima

El comportamiento del córtex visual inspiró el **neocognitrón** (1980)

Biol. Cybernetics 36, 193–202 (1980)

Biological
Cybernetics
© by Springer-Verlag 1980

**Neocognitron: A Self-organizing Neural Network Model
for a Mechanism of Pattern Recognition
Unaffected by Shift in Position**

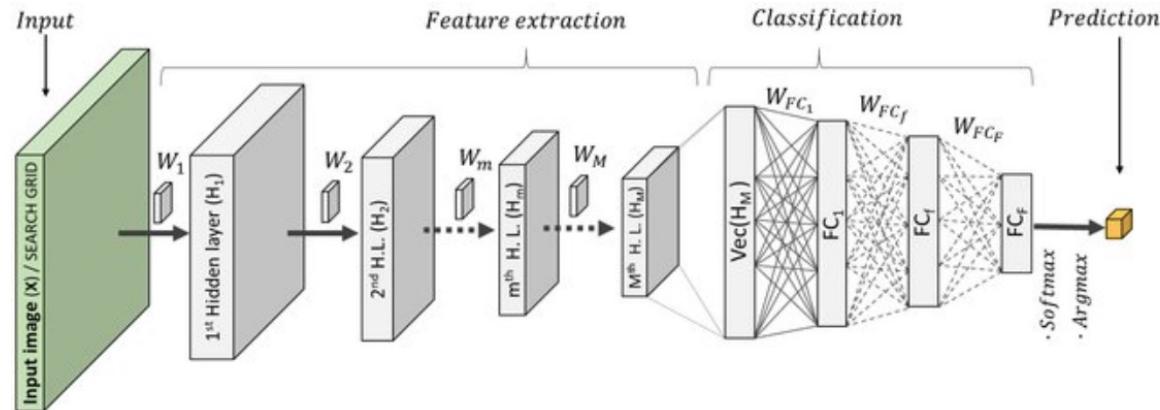
Kunihiko Fukushima

NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan

Convolutional Neural Networks (CNNs)

2. Capas de una CNN

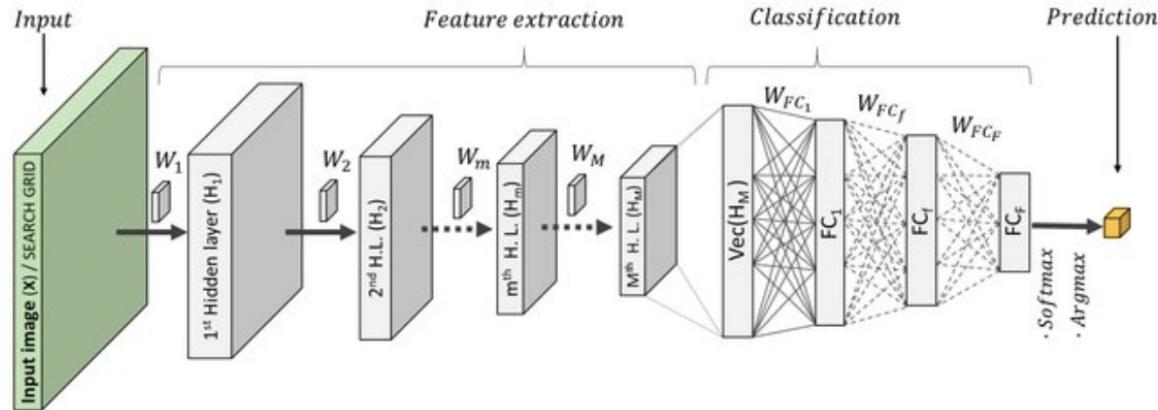
Una CNN es una red neuronal en la que las neuronas de una capa se conectan con las neuronas de la siguiente capa por medio de operaciones de **convolución** y de "**pooling**".



Convolutional Neural Networks (CNNs)

2. Capas de una CNN

La primera fase es la de extracción de características (**feature extraction**), seguida de una fase de **clasificación**, en la que las capas son densas (*fully-connected*).



Convolutional Neural Networks (CNNs)

2. Capas de una CNN

2.1. Capas de Convolución



0	2	15	0	0	11	10	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0
0	10	16	110	238	255	244	245	243	250	249	255	222	103	10
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124
1	2	98	255	228	255	251	254	211	141	116	122	215	251	238
13	217	243	255	155	33	226	52	2	0	10	18	232	255	36
16	229	252	254	49	12	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137
0	87	252	250	248	215	60	0	1	21	252	255	248	144	6
0	13	113	255	255	245	255	182	181	148	252	242	208	36	0
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1
0	0	4	91	255	255	248	254	252	255	244	255	182	10	0
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56
0	218	251	250	173	7	11	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52
0	18	146	250	255	247	255	255	256	249	255	240	255	129	0
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4
0	0	5	5	5	5	5	0	0	14	1	0	6	6	0

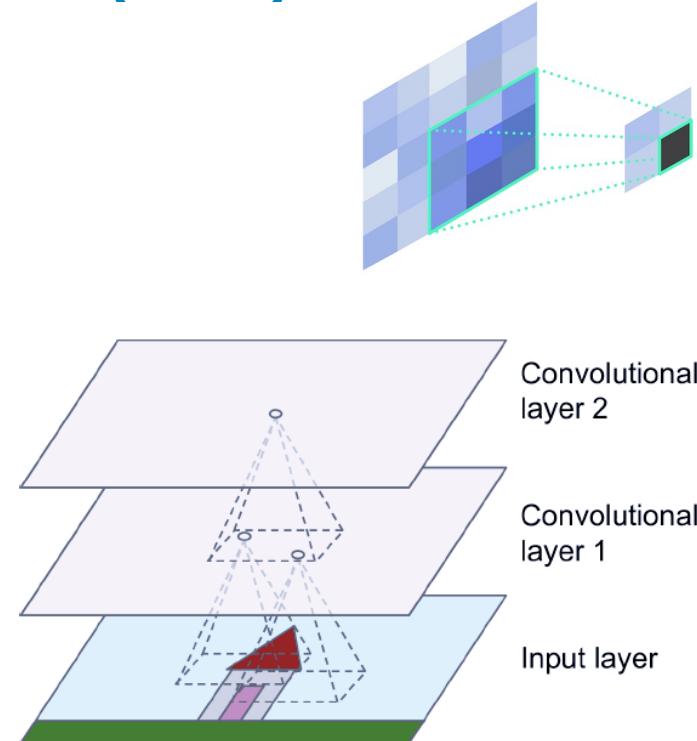
Una imagen no es más que un conjunto de números (a los que llamamos pixel)

Convolutional Neural Networks (CNNs)

2. Capas de una CNN

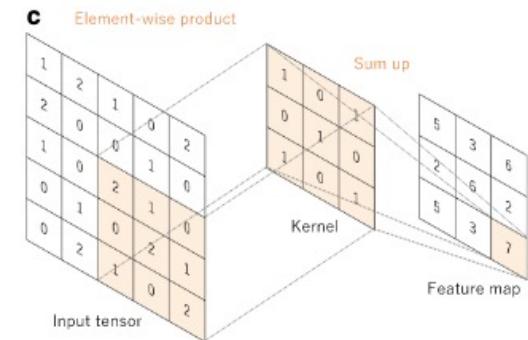
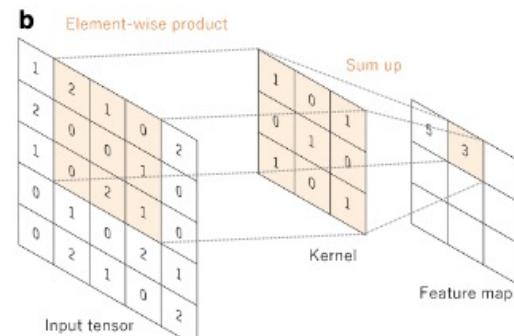
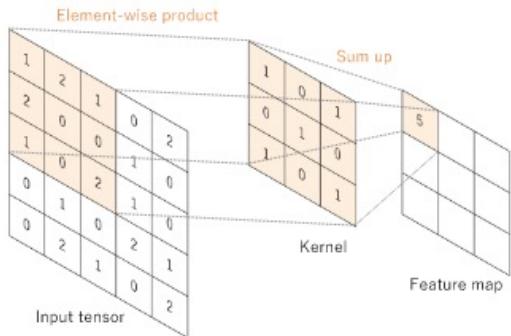
2.1. Capas de Convolución

- Son el **elemento principal** de una CNN
- **Clave:** las neuronas de la primera capa de convolución no están conectadas a cada pixel de entrada (inputs), si no a los píxeles de su campo receptivo local (sucesivamente)



Convolutional Neural Networks (CNNs)

2. Capas de una CNN



Esquema de convolución

Insight into Imaging (2018) 9:631–629
<https://doi.org/10.1007/s13244-018-0639-9>

REVIEW

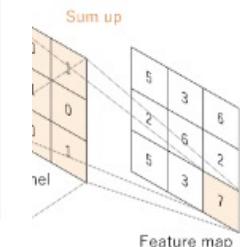
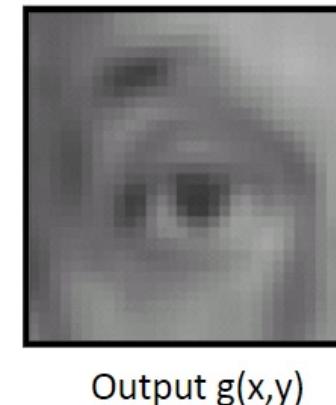
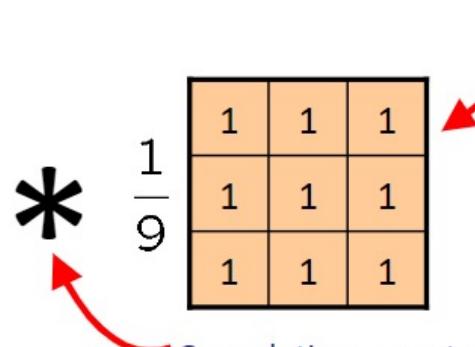
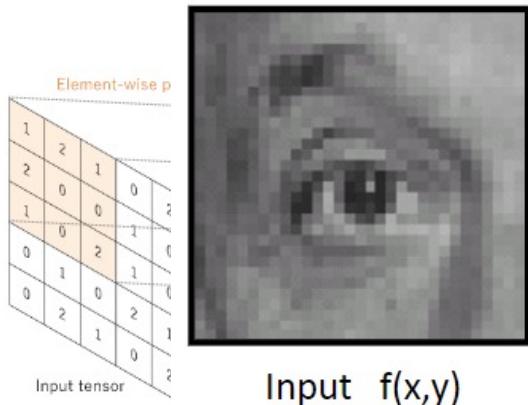


Convolutional neural networks: an overview
and application in radiology

Rikiya Yamashita^{1,2} · Mizuho Nishio^{1,3} · Richard Kinh Gian Do² · Kaori Togashi¹

Convolutional Neural Networks (CNNs)

2. Capas de una CNN



Esquema de convolución

Insight into Imaging (2018) 9:631–629
<https://doi.org/10.1007/s13244-018-0639-9>

REVIEW



Convolutional neural networks: an overview
and application in radiology

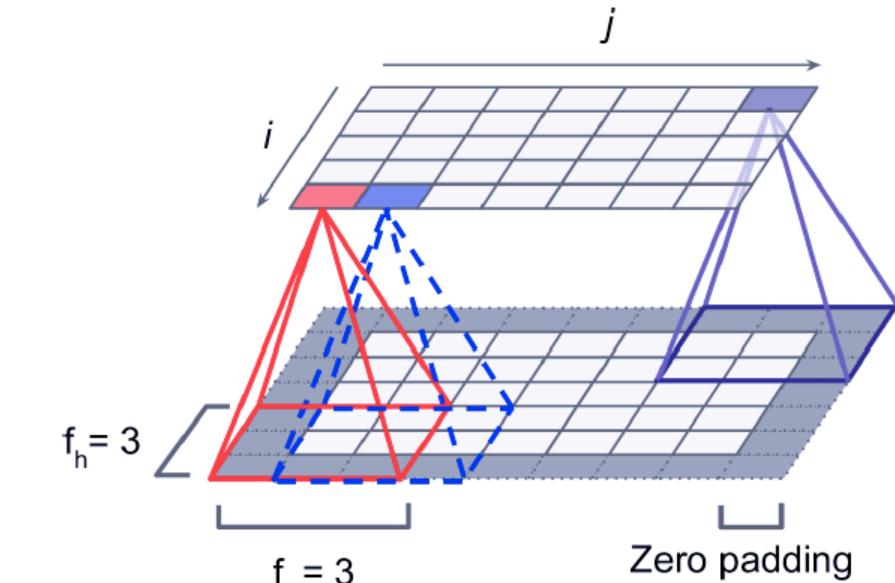
Rikiya Yamashita^{1,2} · Mizuho Nishio^{1,3} · Richard Kinh Gian Do² · Kaori Togashi¹

Convolutional Neural Networks (CNNs)

2. Capas de una CNN

2.1. Capas de Convolución

- Neurona de la fila i , columna j de una capa a se conecta a los outputs de las neuronas de la capa anterior situadas en las filas i a $i + f_h - 1$ y columnas j a $j + f_w - 1$
- Zero padding: filas y columnas de ceros para cuadrar dimensiones entre capas

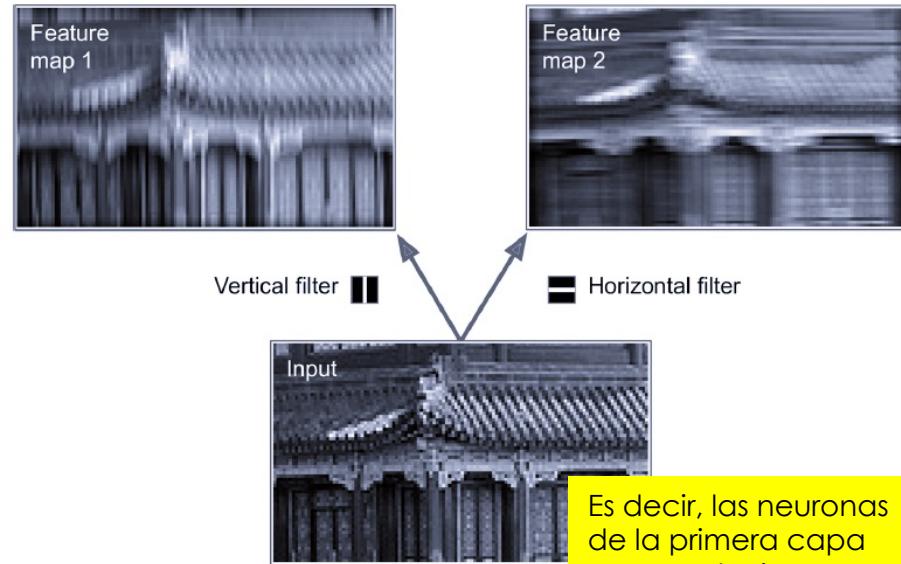


Convolutional Neural Networks (CNNs)

2. Capas de una CNN

2.1. Capas de Convolución

- Los “pesos” de las neuronas de una CNN se representan con “**filtros**” o “kernels de convolución”.
- Ej.: Las neuronas asociadas al filtro vertical omitirán toda la información de su campo receptivo que no sea líneas verticales
- Si todas las neuronas de una capa usan un mismo filtro, se obtiene un “**feature map**”



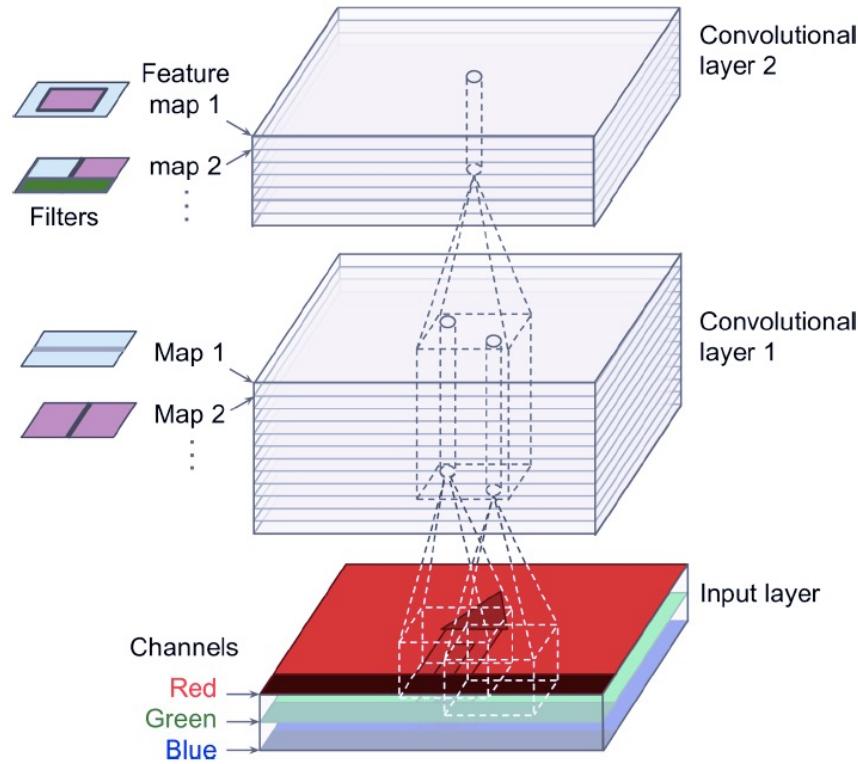
Es decir, las neuronas de la primera capa se convolucionan con un filtro, que da como resultado la siguiente neurona

Convolutional Neural Networks (CNNs)

2. Capas de una CNN

2.1. Capas de Convolución

- En la realidad tenemos múltiples capas de convolución con múltiples feature maps cada una
- Cada feature map utiliza el mismo filtro, lo que **reduce considerablemente los parámetros a entrenar**
- **Los filtros son los valores que la CNN irá aprendiendo**

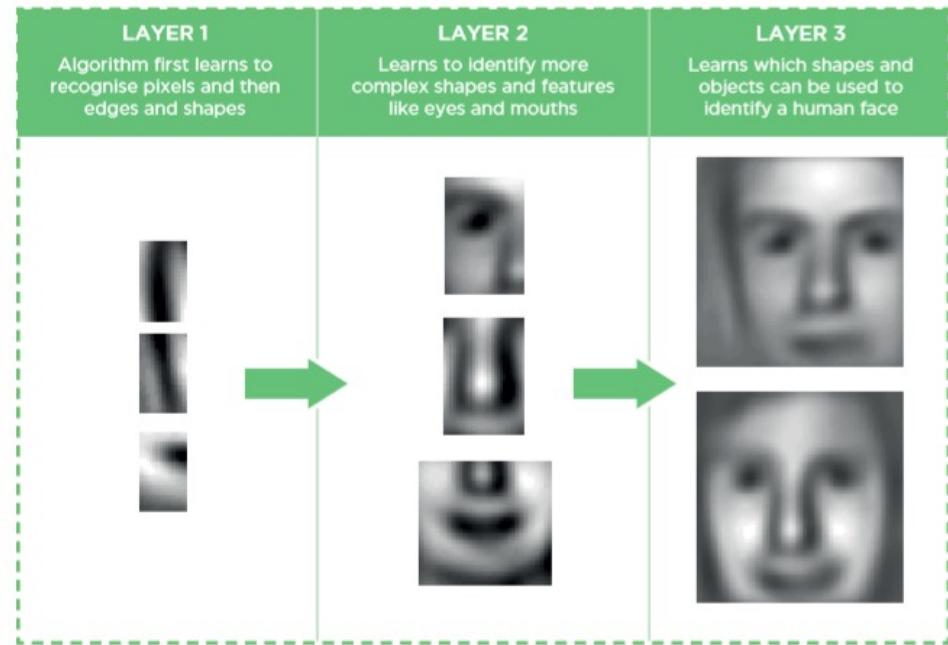


Convolutional Neural Networks (CNNs)

2. Capas de una CNN

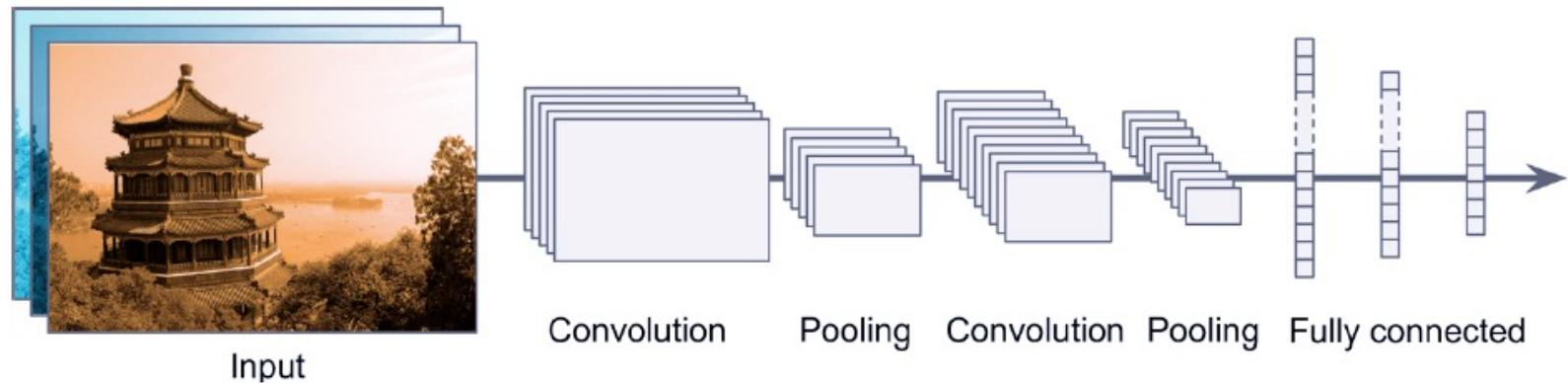
2.1. Capas Pooling

- Son capas que reducen la dimensionalidad de la imagen (no utilizan filtros de convolución)
- Pueden coger el máximo valor (**MaxPooling**) el mínimo o el valor medio (**AvgPooling**)



Convolutional Neural Networks (CNNs)

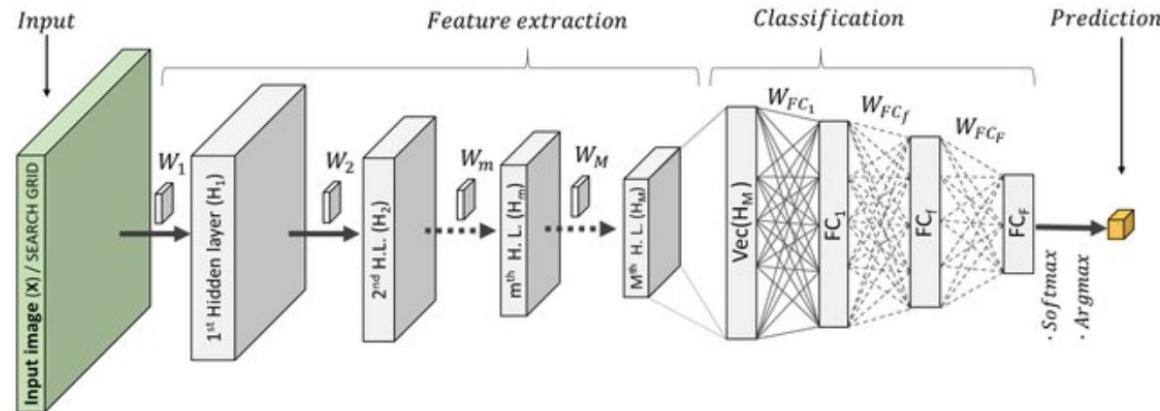
2. Capas de una CNN



Arquitectura típica de una CNN

Convolutional Neural Networks (CNNs)

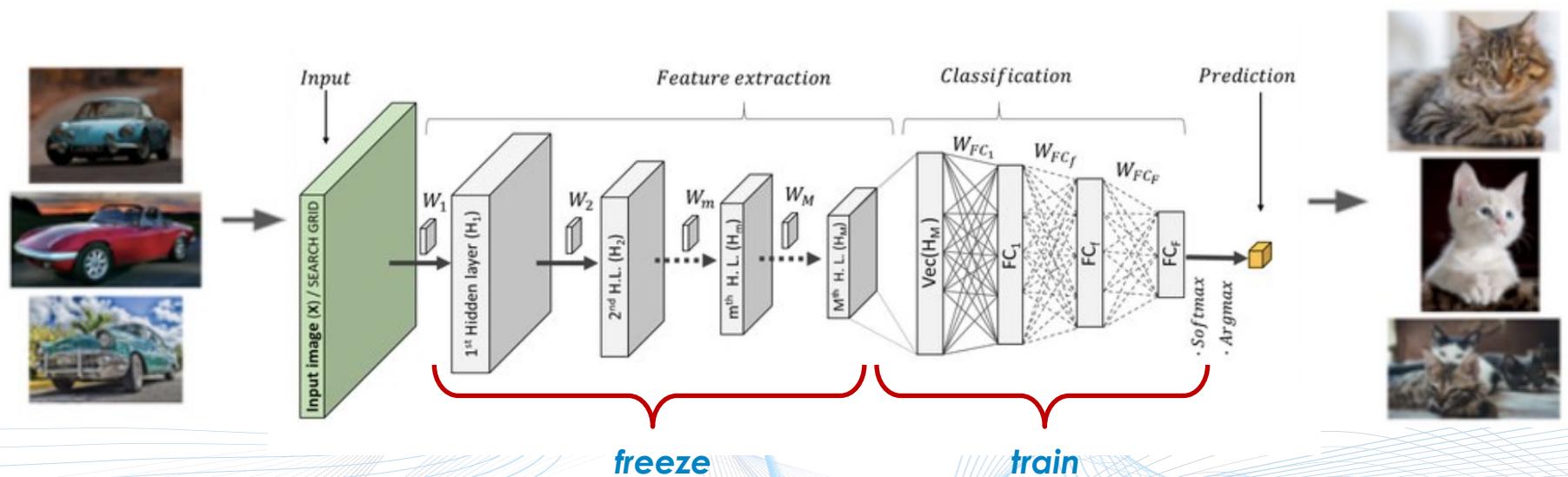
2. Capas de una CNN



Arquitectura típica de una CNN

Convolutional Neural Networks (CNNs)

2. Capas de una CNN

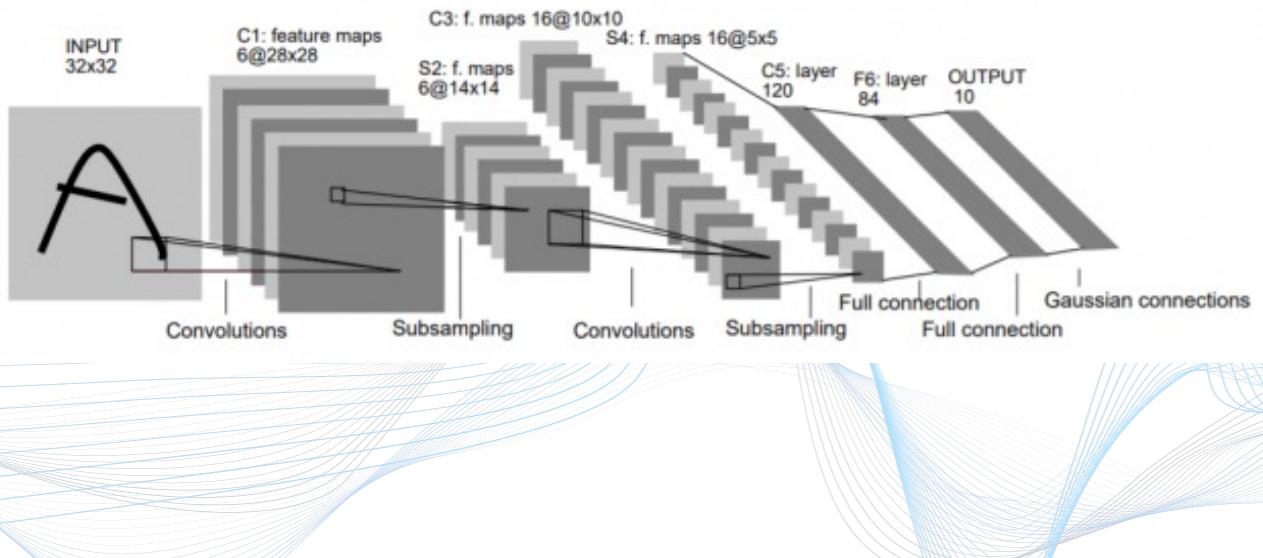


Esto es **transfer learning**

Convolutional Neural Networks (CNNs)

3. Arquitecturas de CNN

LeNet-5



1998

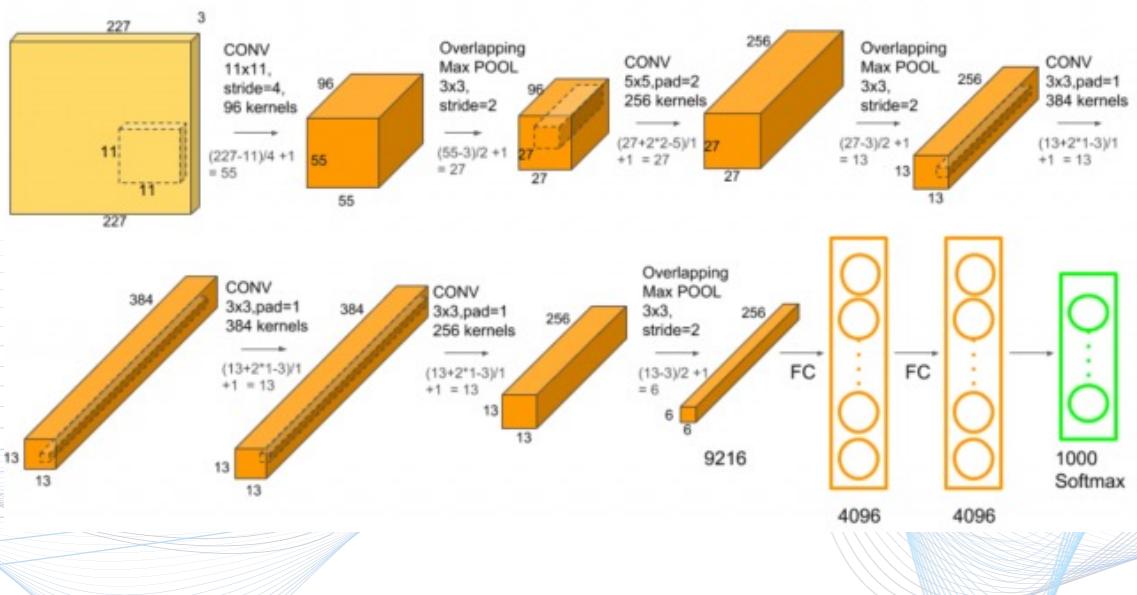
Yann LeCun

Layer	Type	Maps	Size	Kernel size	Stride	Activation
Out	Fully connected	–	10	–	–	RBF
F6	Fully connected	–	84	–	–	tanh
C5	Convolution	120	1x1	5x5	1	tanh
S4	Avg pooling	16	5x5	2x2	2	tanh
C3	Convolution	16	10x10	5x5	1	tanh
S2	Avg pooling	6	14x14	2x2	2	tanh
C1	Convolution	6	28x28	5x5	1	tanh
In	Input	1	32x32	–	–	–

Convolutional Neural Networks (CNNs)

3. Arquitecturas de CNN

AlexNet



2012
Alex Krizhevsky
et. al

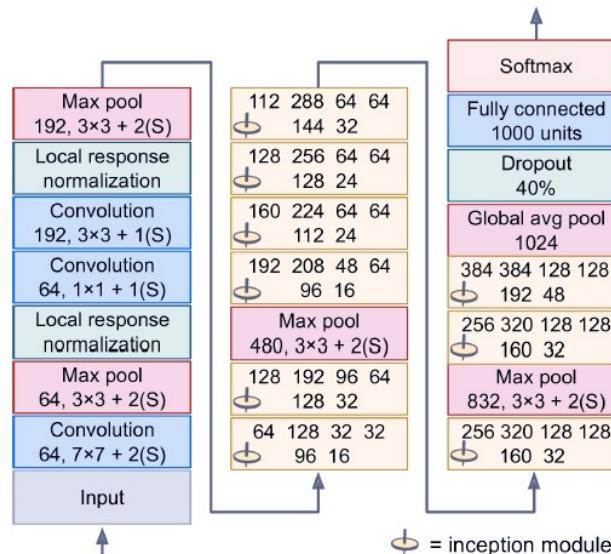


Layer	Type	Maps	Size	Kernel size	Stride	Padding	Activation
Out	Fully connected	–	1,000	–	–	–	Softmax
F10	Fully connected	–	4,096	–	–	–	ReLU
F9	Fully connected	–	4,096	–	–	–	ReLU
S8	Max pooling	256	6 x 6	3 x 3	2	valid	–
C7	Convolution	256	13 x 13	3 x 3	1	same	ReLU
C6	Convolution	384	13 x 13	3 x 3	1	same	ReLU
C5	Convolution	384	13 x 13	3 x 3	1	same	ReLU
S4	Max pooling	256	13 x 13	3 x 3	2	valid	–
C3	Convolution	256	27 x 27	5 x 5	1	same	ReLU
S2	Max pooling	96	27 x 27	3 x 3	2	valid	–
C1	Convolution	96	55 x 55	11 x 11	4	valid	ReLU
In	Input	3 (RGB)	227 x 227	–	–	–	–

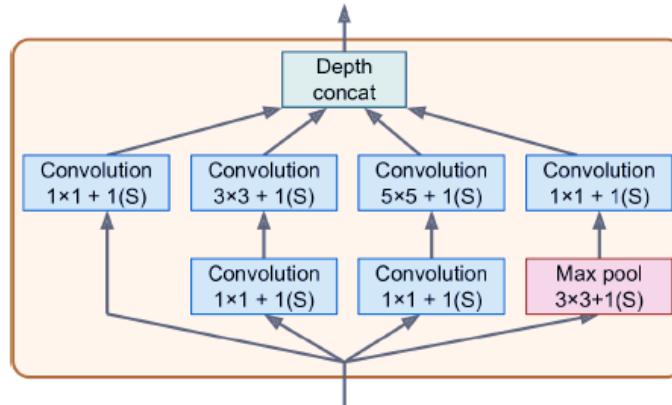
Convolutional Neural Networks (CNNs)

3. Arquitecturas de CNN

GoogLeNet



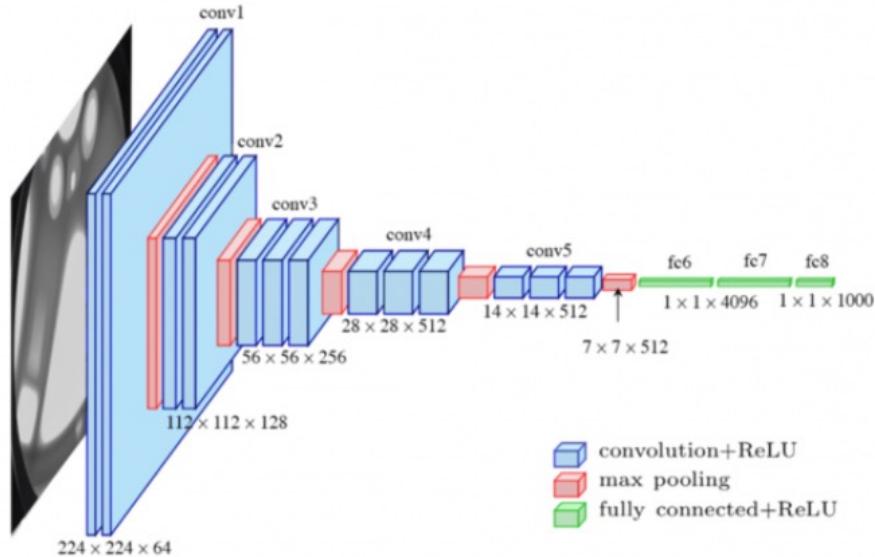
2014
Christian Szegedy
et. al



Convolutional Neural Networks (CNNs)

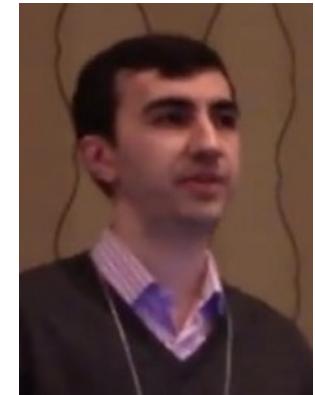
3. Arquitecturas de CNN

VGGNet (VGG16, VGG19)



2014

Karen
Simonyan



Andrew
Zisserman



Convolutional Neural Networks (CNNs)

3. Arquitecturas de CNN

ResNet

2015

Kaimi He
et. al

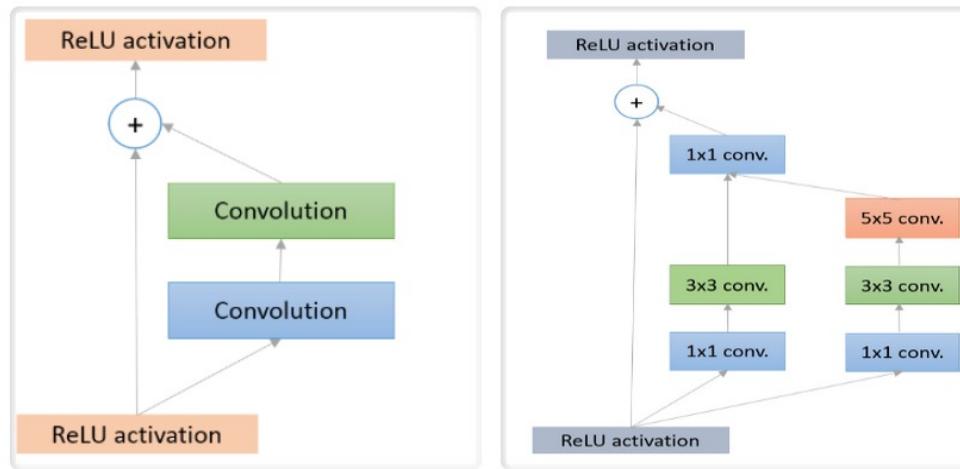


Fig. 4. Basic diagram of Residual block (Left), Basic block diagram for Inception Residual unit (Right)

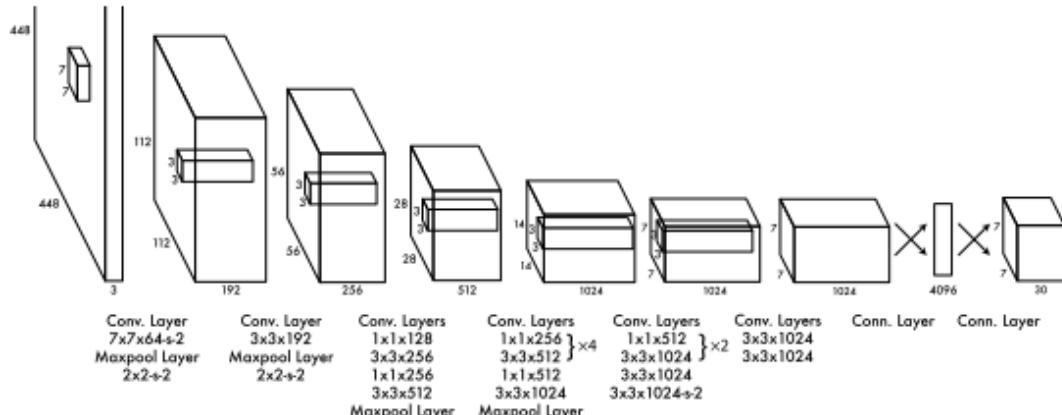
Convolutional Neural Networks (CNNs)

3. Arquitecturas de CNN

YOLO (You Only Look Once)

2015

Joseph Redmon
et. al

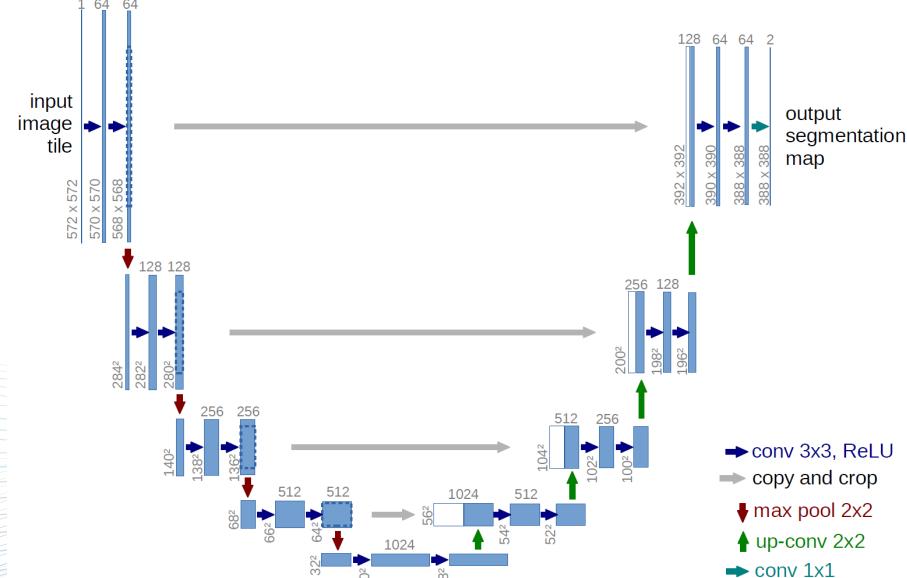


Convolutional Neural Networks (CNNs)

3. Arquitecturas de CNN

2015

U-Net



Olaf Ronneberger
et. al

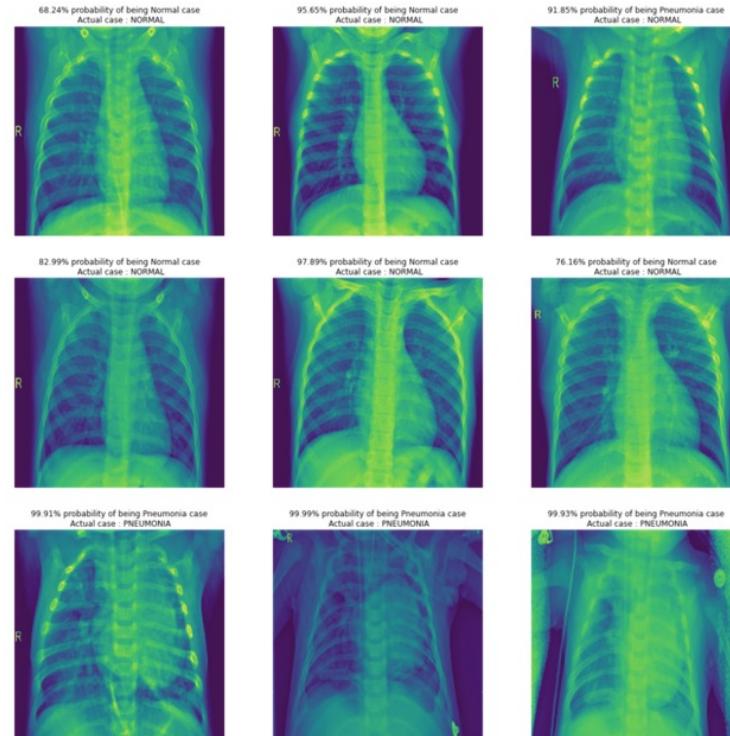


Convolutional Neural Networks (CNNs)

4. Problemas que resuelven

4.1. Clasificación

- Las CNN se entrena con imágenes 2D o 3D etiquetadas
- Ejemplos de dos clases:
 - tumor benigno o maligno
 - pulmón con neumonía o sano
- Ejemplos de múltiples clases
 - Grado de malignidad de un tumor / quiste
 - Clasificación de masas hepáticas

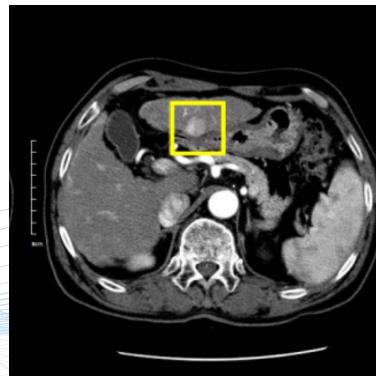
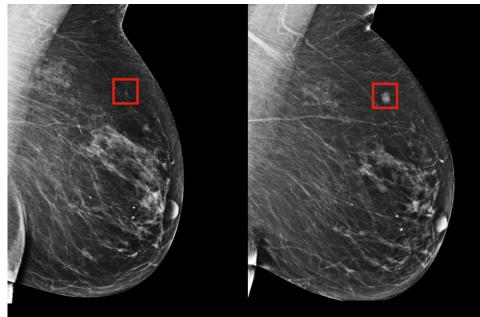


Convolutional Neural Networks (CNNs)

4. Problemas que resuelven

4.2. Detección de objetos

- Detección de anomalías
- Detección de tumores (muy avanzado en mamografías)
- Detección de objetos en general
(útil para localizar al personal trabajando con radiación)
- Ejemplo de arquitectura: **YOLO**

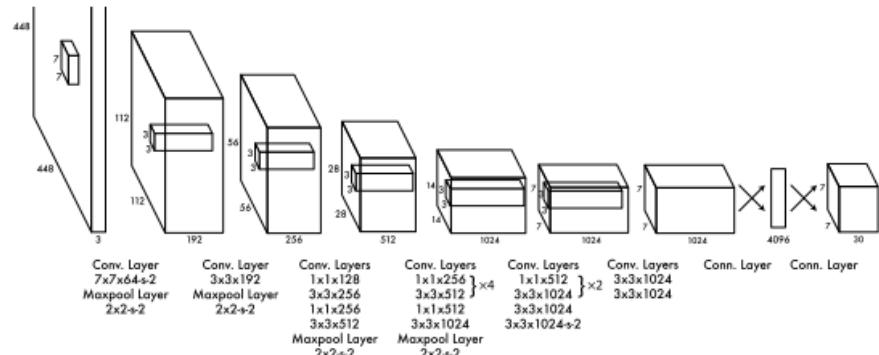
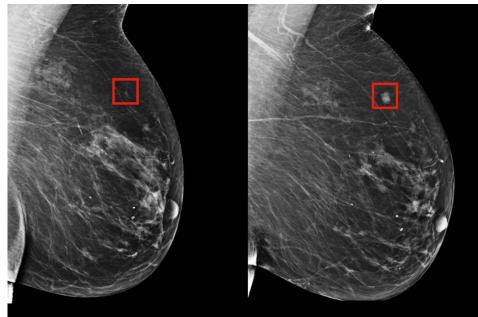


Convolutional Neural Networks (CNNs)

4. Problemas que resuelven

4.2. Detección de objetos

- Detección de anomalías
- Detección de tumores (muy avanzado en mamografías)
- Detección de objetos en general
(útil para localizar al personal trabajando con radiación)
- Ejemplo de arquitectura: **YOLO**



Convolutional Neural Networks (CNNs)

4. Problemas que resuelven

4.3. Segmentación

- Segmentación automática de órganos
- Interesante en radiología, radioterapia, medicina nuclear...
- Importante de cara a la dosimetría del paciente

NiftyNet: a deep-learning platform for medical imaging

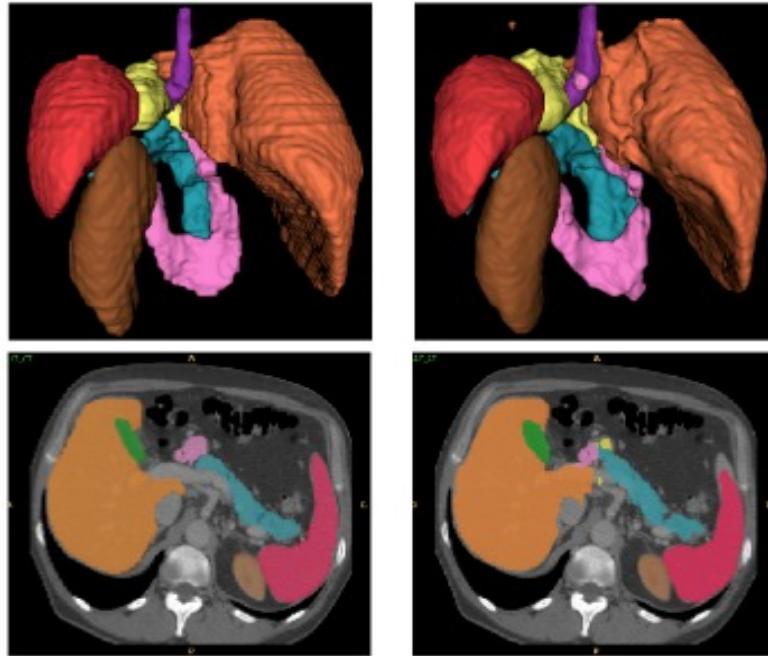
Eli Gibson ^{a,b,1}, Wenqi Li ^{a,1,*}, Carole Sudre ^b, Lucas Fidon ^a, Dzhoshikun I. Shakir ^a,
Guotai Wang ^a, Zach Eaton-Rosen ^b, Robert Gray ^{c,d}, Tom Doel ^a, Yipeng Hu ^b, Tom Whyntie ^b,
Parashkev Nachev ^{c,d}, Marc Modat ^b, Dean C. Barratt ^{a,b}, Sébastien Ourselin ^a,
M. Jorge Cardoso ^{b,2}, Tom Vercauteren ^{a,2}

^a Wellcome / EPSRC Centre for Interventional and Surgical Sciences (WEISS), University College London, UK

^b Centre for Medical Image Computing (CMIC), Departments of Medical Physics & Biomedical Engineering and Computer Science, University College London, UK

^c Institute of Neurology, University College London, UK

^d National Hospital for Neurology and Neurosurgery, London, UK

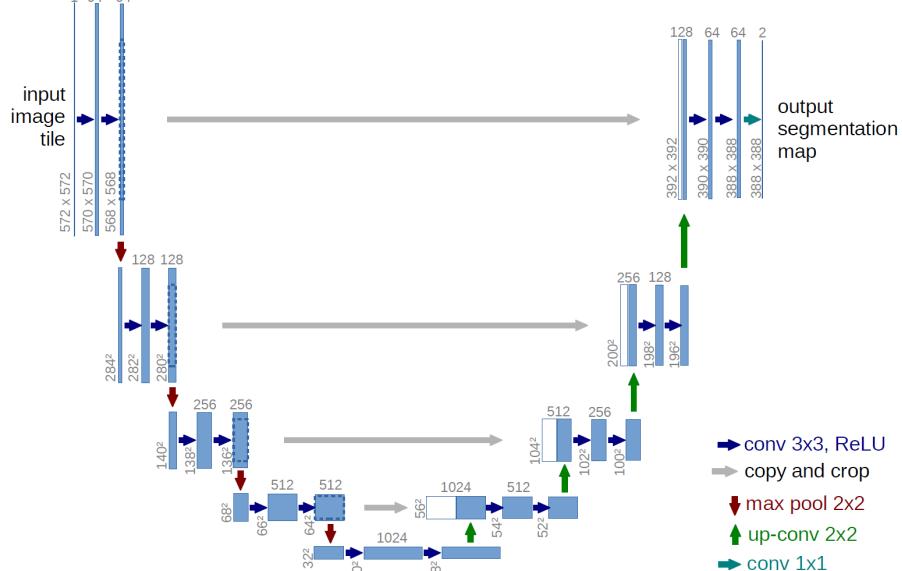


Convolutional Neural Networks (CNNs)

4. Problemas que resuelven

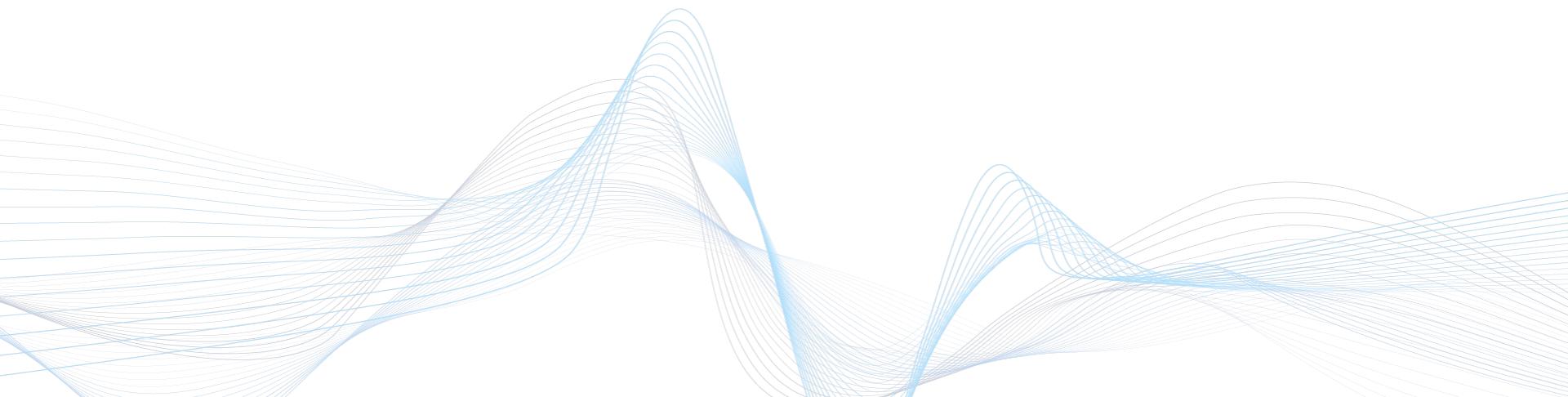
4.3. Segmentación

- Segmentación automática de órganos
 - Interesante en radiología, radioterapia, medicina nuclear...
 - Importante de cara a la dosimetría del paciente
 - Ejemplo de arquitectura: **U-Net**



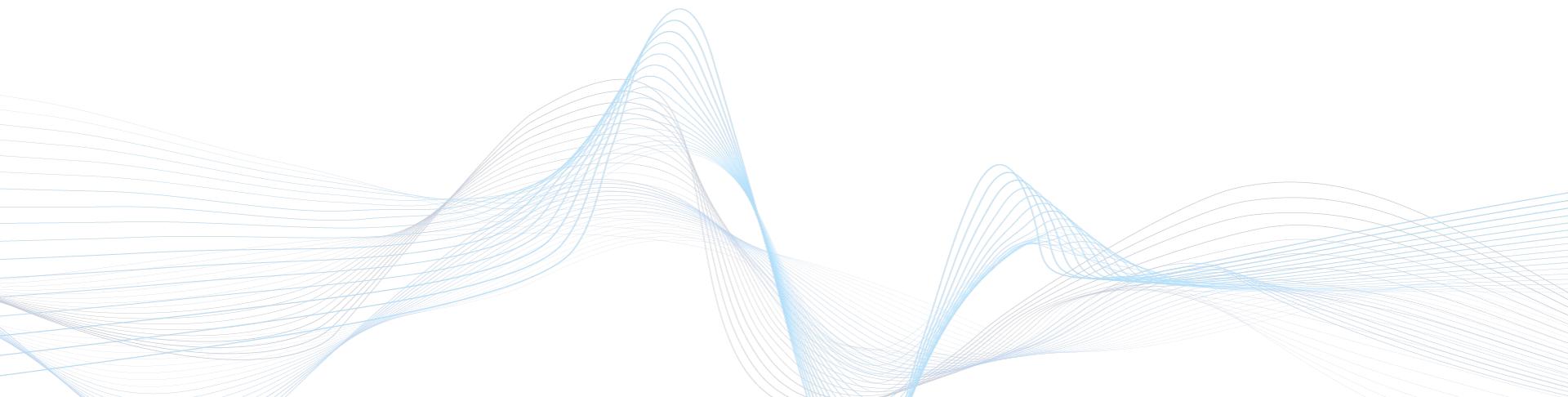
Ejemplos de aplicación

1. Sanidad
2. Radioterapia

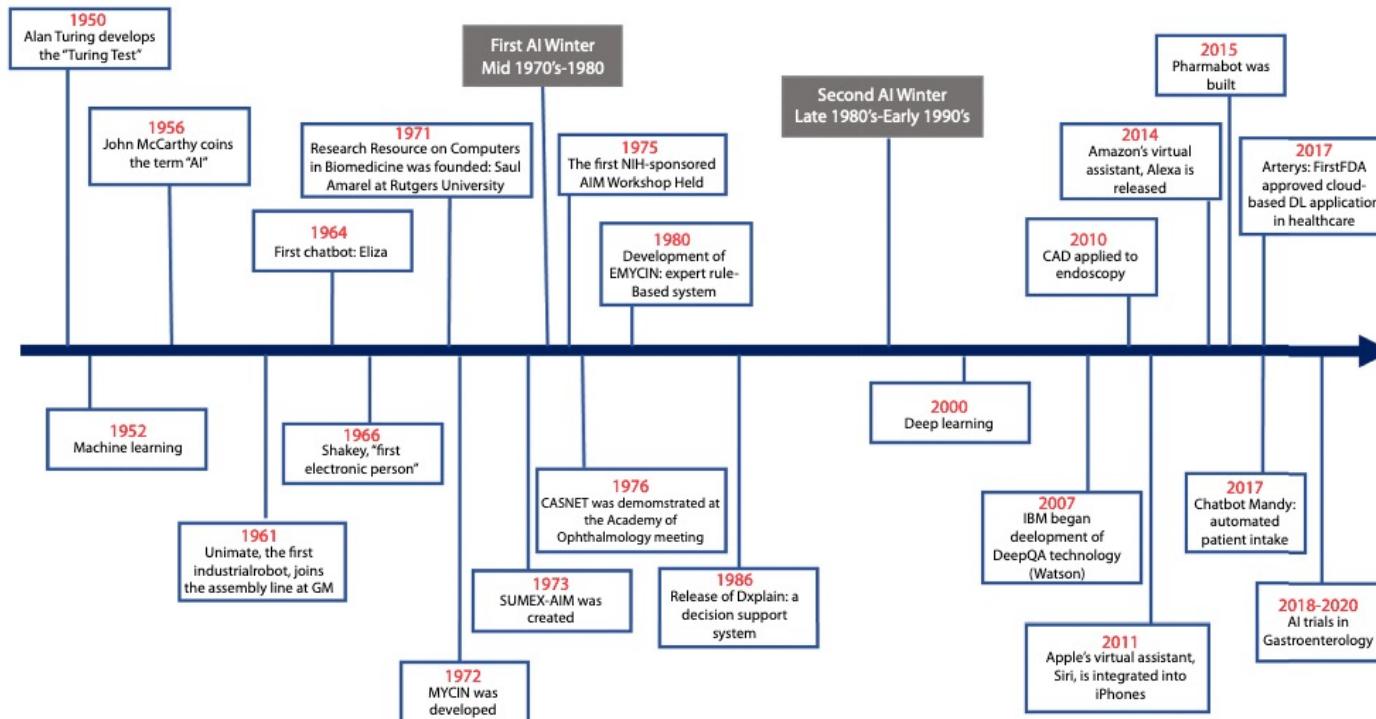


Ejemplos de aplicación

1. Sanidad
2. Radioterapia

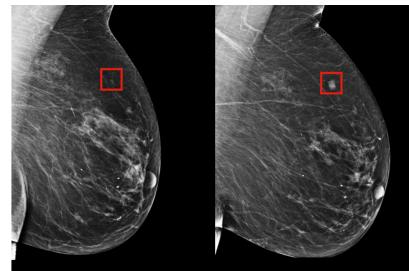
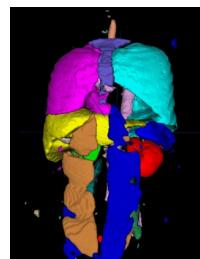


Ejemplos de aplicación en sanidad



Ejemplos de aplicación en sanidad

- Radiología
- Radioterapia
- Análisis de historias clínicas
- Análisis de señales
- Telemedicina



Ejemplos de aplicación en sanidad

- IBM Watson Health
- Microsoft Project Hanover
- Google DeepMind
- Neuralink
- Savana
- Masimo SafetyNet
- Remedy
- LimbusAI
- Atomwise
- ...



Ejemplos de aplicación en Radioterapia

- Segmentación de órganos y OARs
- Registro de imágenes
- Sistemas CAD de soporte al diagnóstico
- TPS (*Treatment Planning System*)
- Reducción de dosis



AI technology to simplify
Unity prostate workflow



To take home...

- Las redes neuronales son un tipo de algoritmo de ML inspirados en el funcionamiento del cerebro humano.
- En la actualidad tenemos la capacidad de computación necesaria para ejecutar los algoritmos.
- El algoritmo del backpropagation ha supuesto un punto de inflexión en el DL, ya que nos permite encontrar el gradiente de la función de coste de manera eficiente.
- Las redes neuronales convolucionales (CNN) son muy utilizadas para trabajar con imágenes. Se basan en el funcionamiento del córtex visual.
- Cada vez son más las empresas del sector sanitario que se dedican a la Inteligencia Artificial.

The future depends on some graduate student who is deeply suspicious of everything I have said

Geoffrey Hinton

