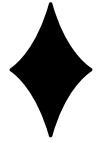


R LADIES MADRID

Implementando en R Redes Neuronales

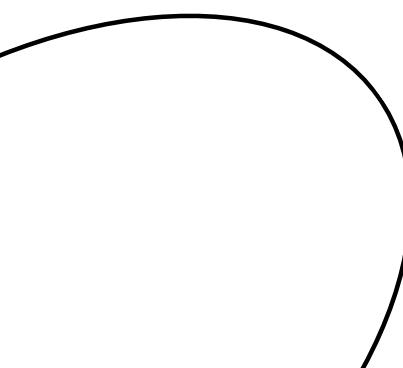
CLARA PINIELLA

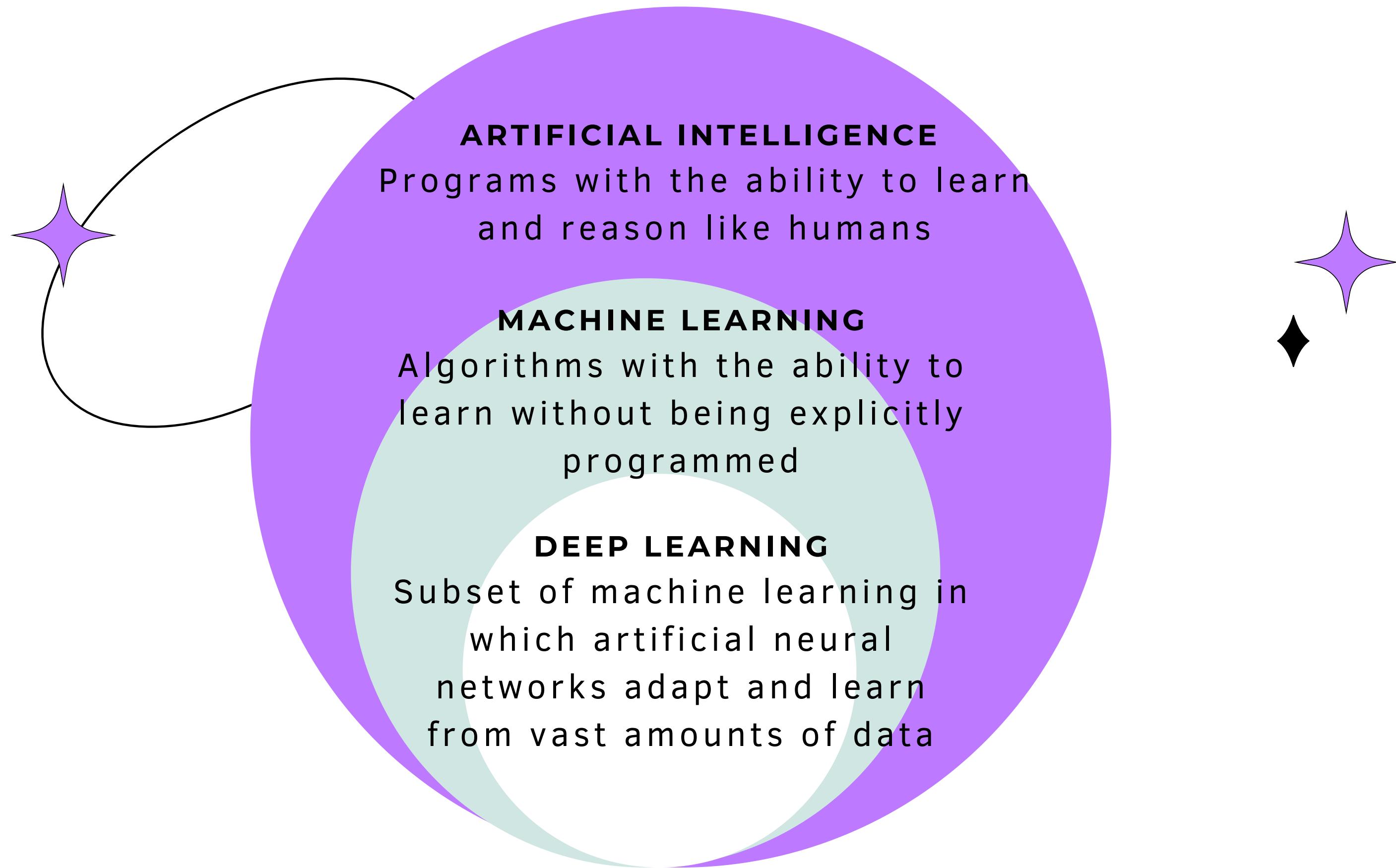


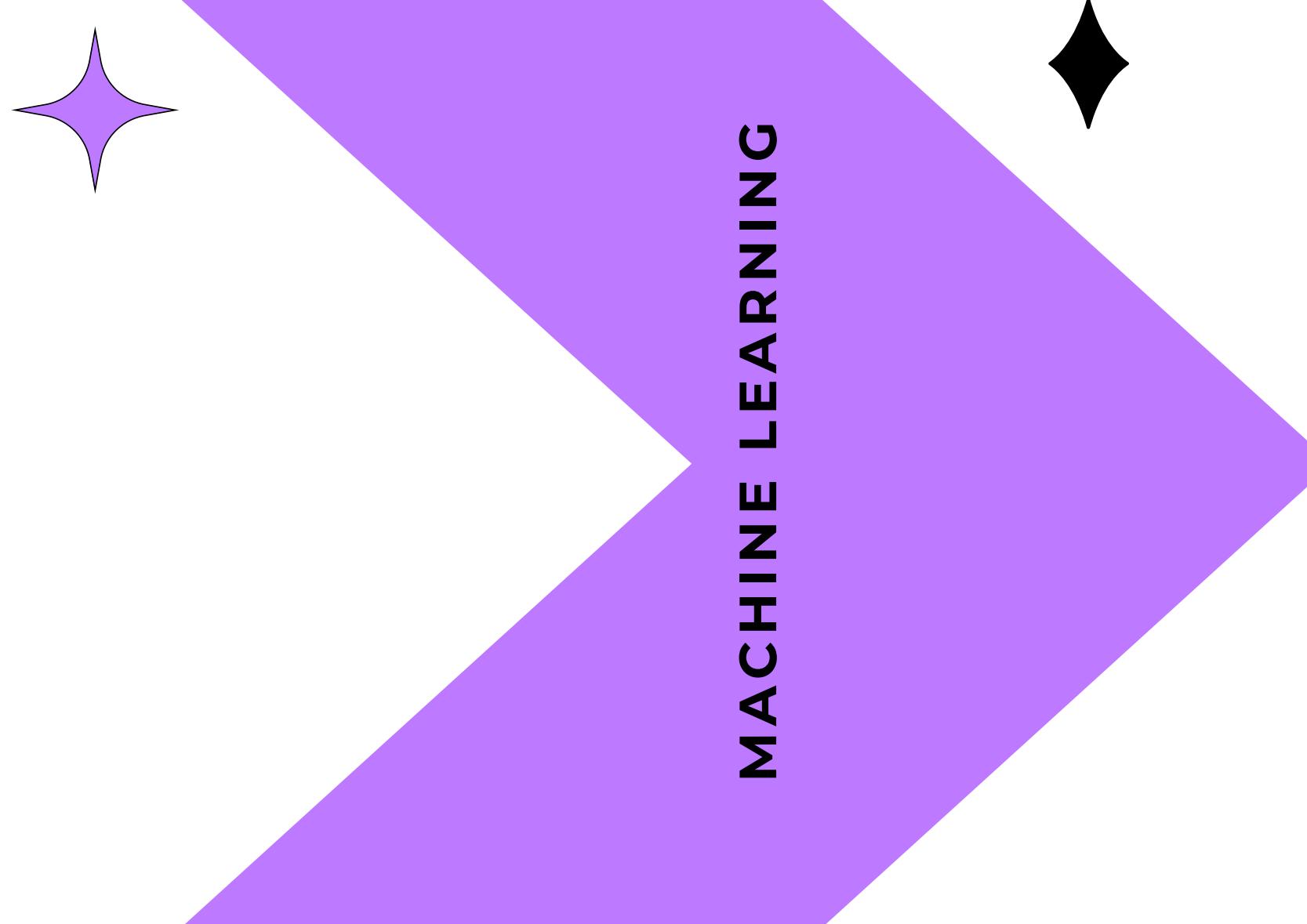
Who am I?

Clara Piniella Martínez

TA DATA SCIENCE THE BRIDGE
clapiniella@gmail.com





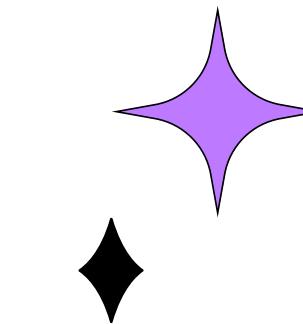


Redes Neuronales

¿Por qué ahora?

Los algoritmos tradicionales no escalan bien con gran cantidad de datos o datos no estructurados.

Ha crecido exponencialmente la capacidad de cómputo.



Pero antes...

MACHINE LEARNING

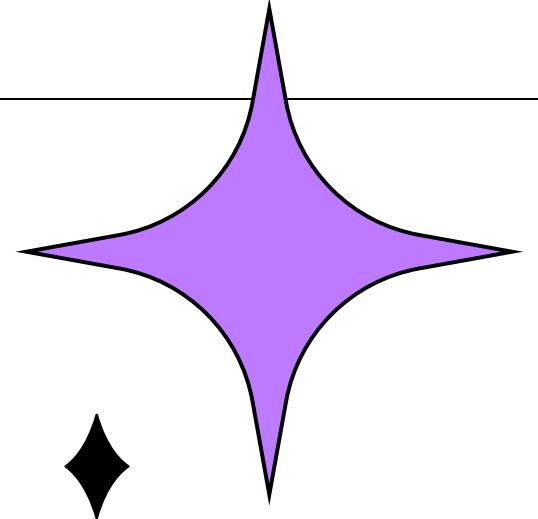
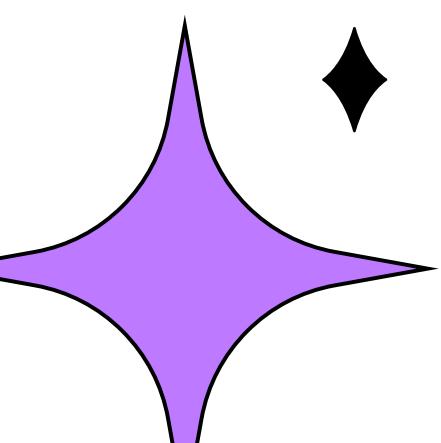
SUPERVISED LEARNING

Usa datos etiquetados como input y predice las etiquetas como output.



UNSUPERVISE LEARN.

Analiza los datos buscando variables en común para así crear grupos o clústeres.



Pero antes...

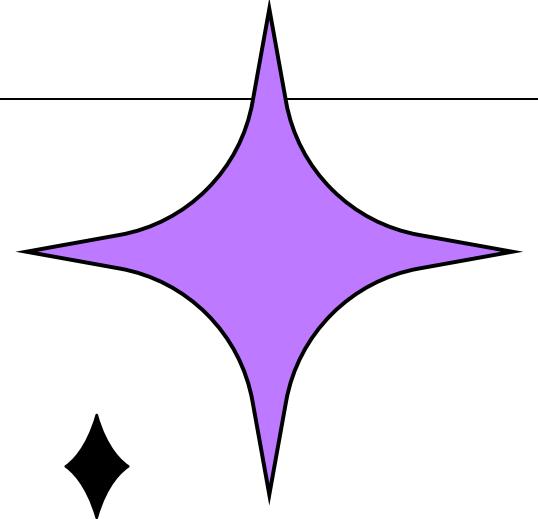
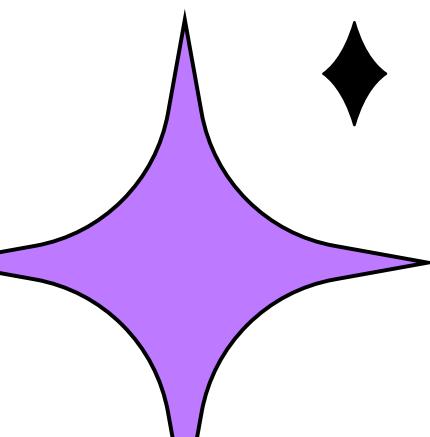
SUPERVISED LEARNING

CLASIFICACIÓN

Dada una imagen, ¿eso es un perro o un gato?

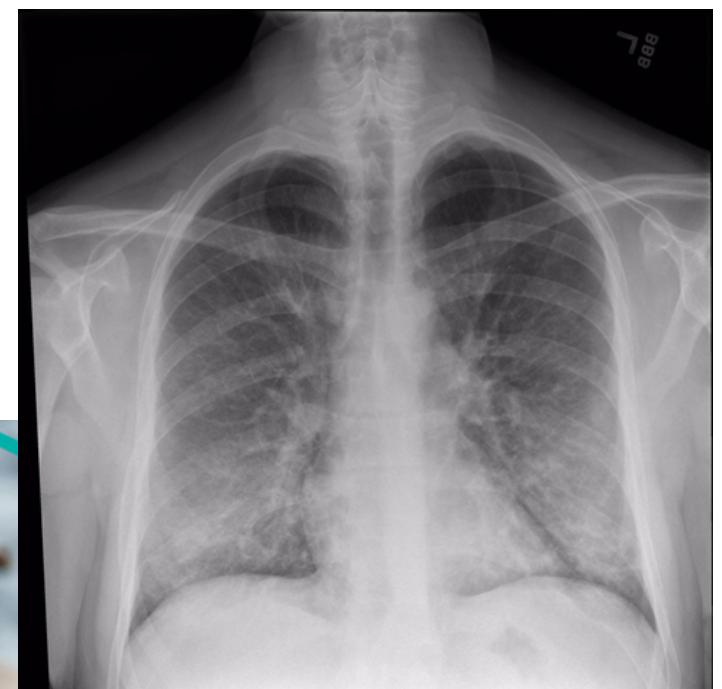
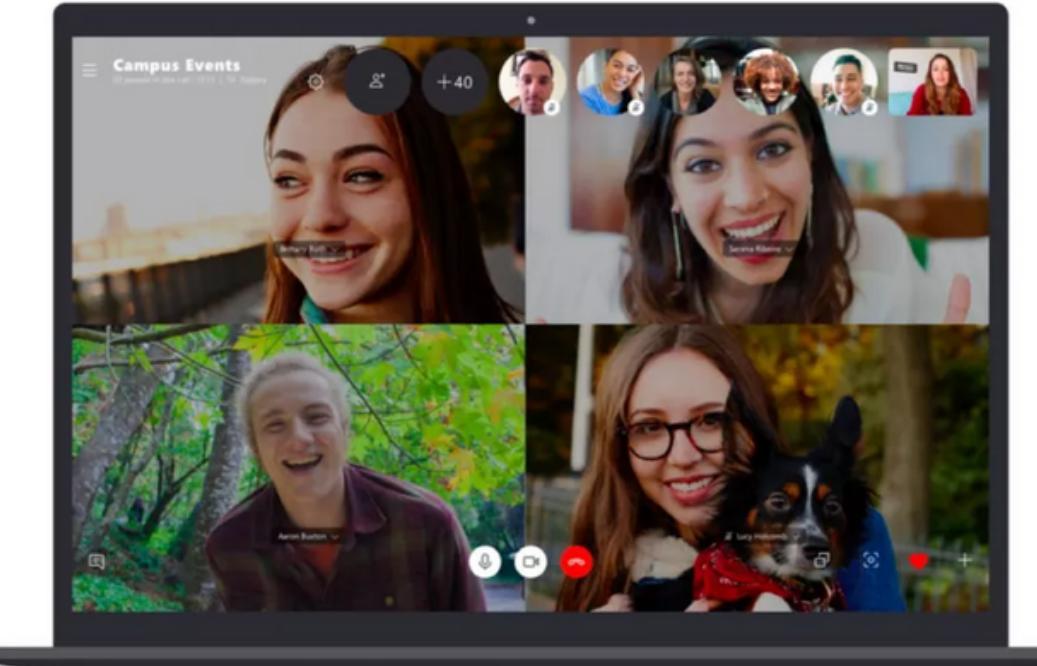
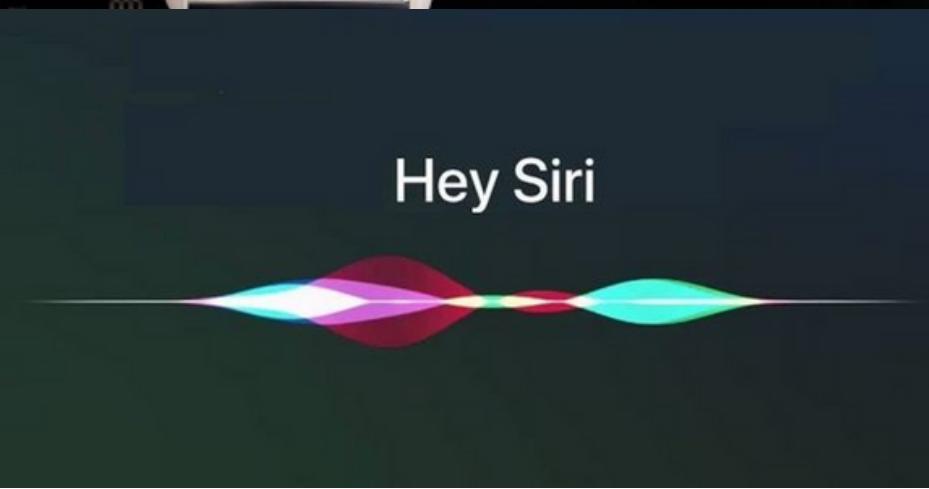
REGRESIÓN

Predecir el precio de las casas en Madrid.



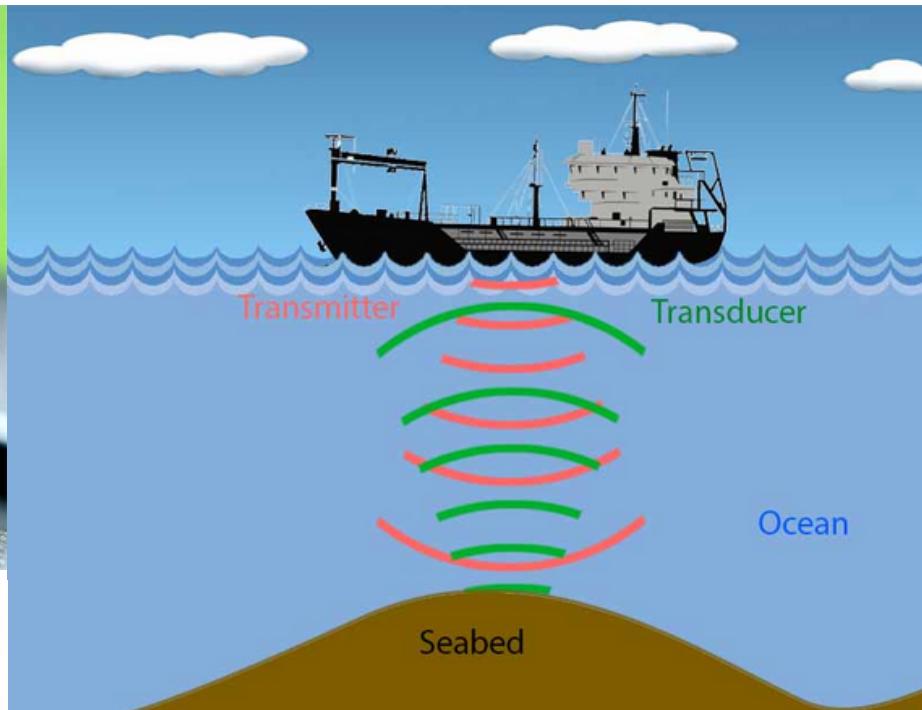


Algunas de las aplicaciones



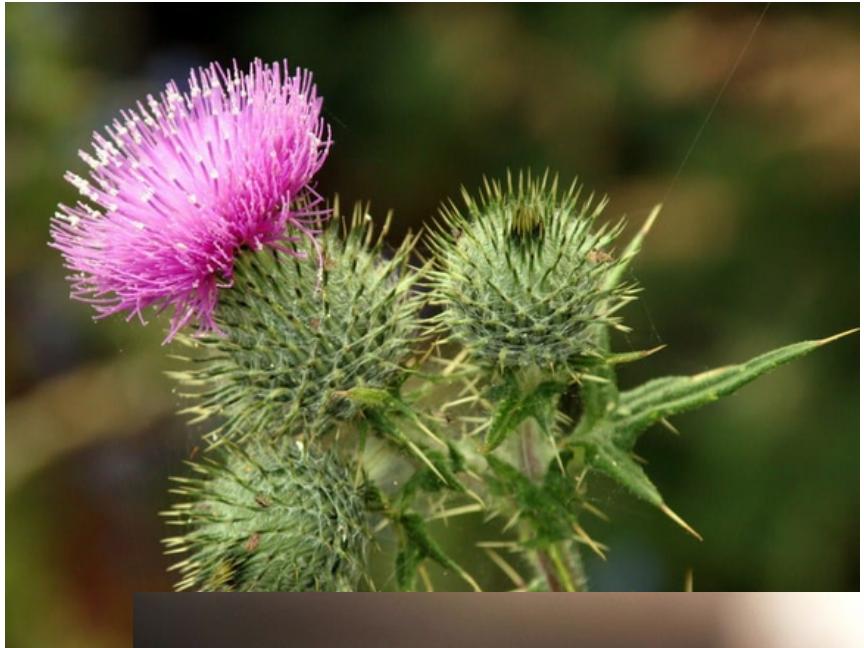
♦ ♦ ♦

¿Qué tienen estas imágenes en común?



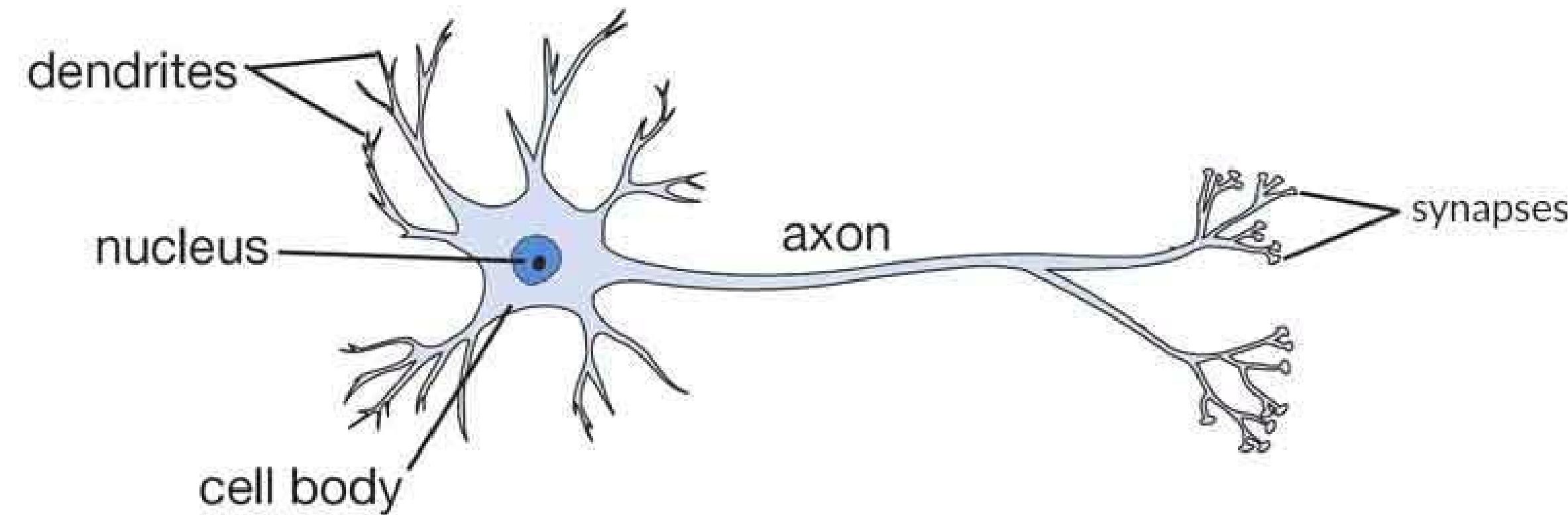


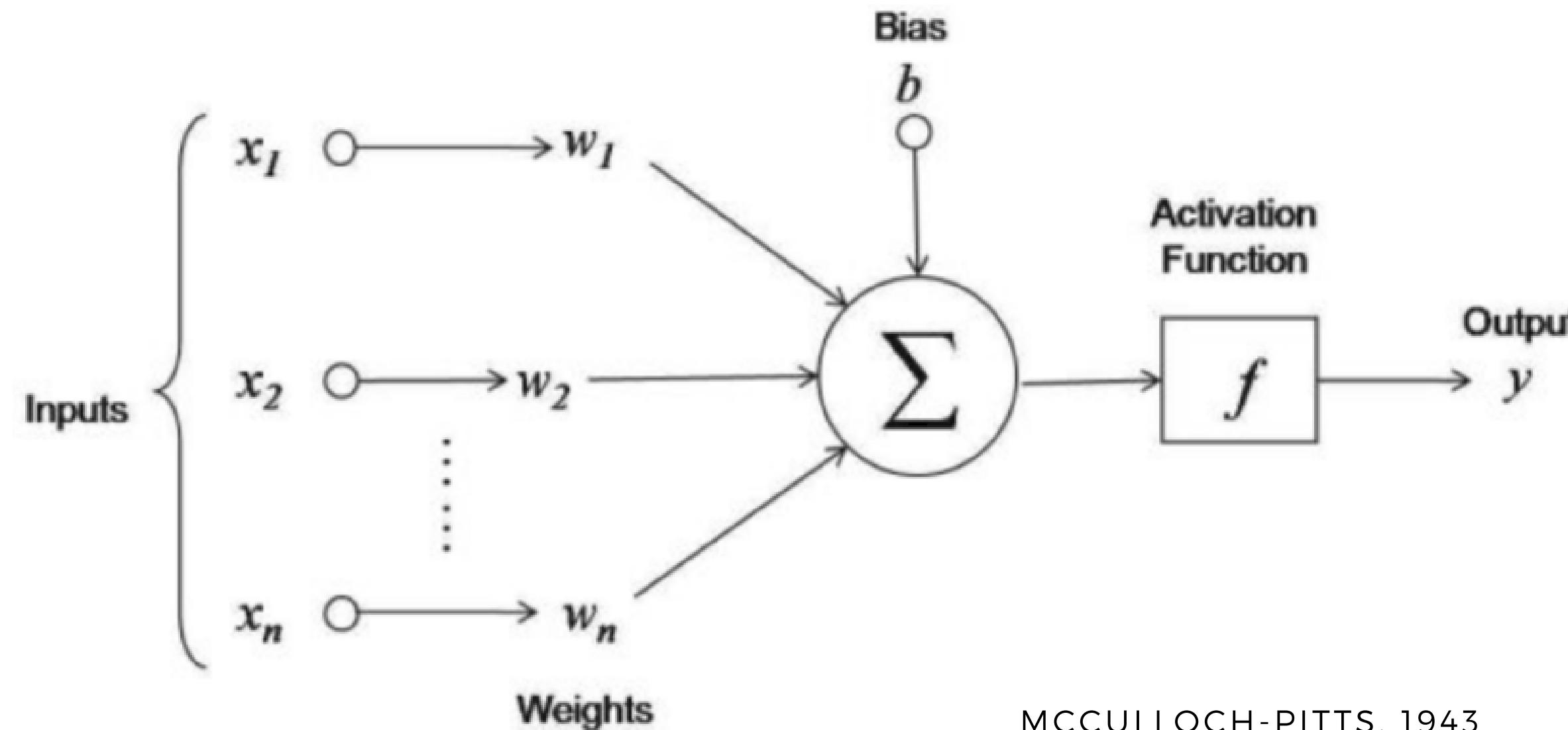
♦ ¿Qué tienen estas imágenes en común?





Biological Neuron





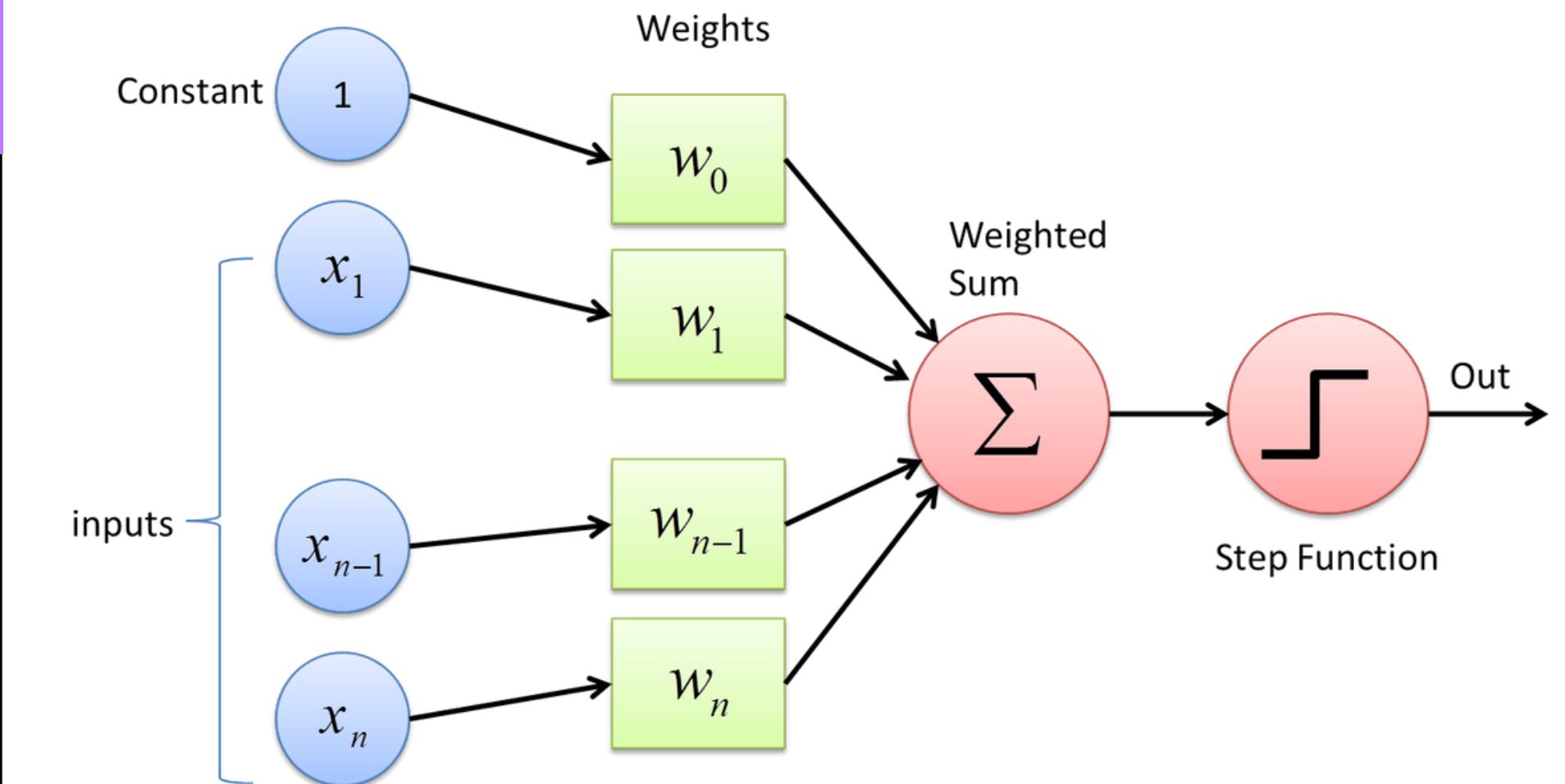
MCCULLOCH-PITTS, 1943





COMPONENTES DE LA NEURONA:

- **INPUTS**
- **PESOS** QUE RECIBE CADA INPUT, SE COMBINAN COMO SI DE UNA REGRESIÓN LINEAL SE TRATASE
- PROBLEMAS BINARIOS
- **BIAS**
- **ACTIVATION FUNCTION**: SI LA SEÑAL CUMPLE CIERTO THRESHOLD, SE ACTIVA LA SALIDA



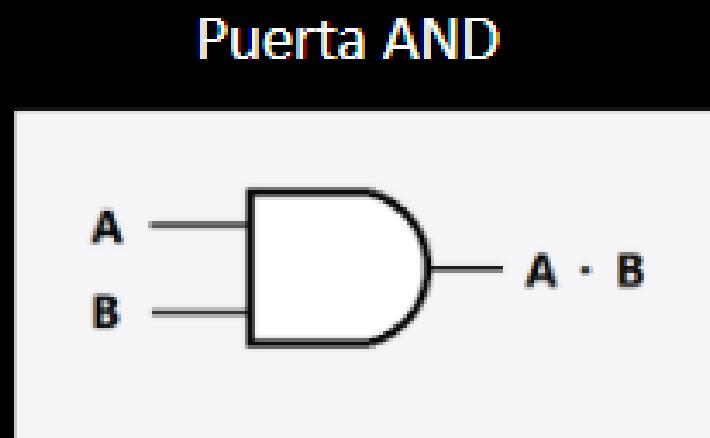
Como si fuera una regresión lineal



$$Y = \sum (\text{weight} * \text{input}) + \text{bias}$$

¿Qué resuelve una neurona?

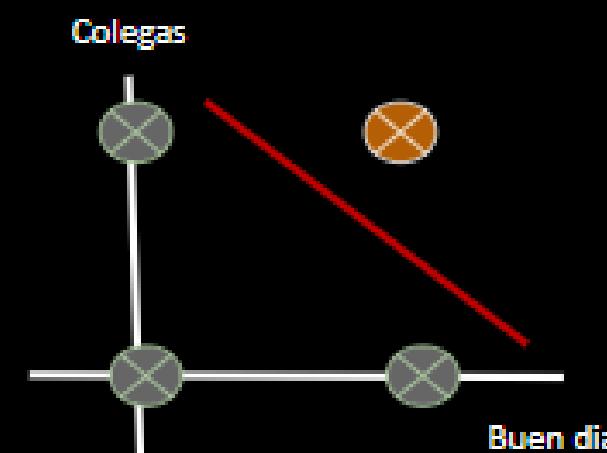
Las neuronas en solitario no resuelven problemas complejos. Veamos algunos ejemplos



Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Vamos a determinar si salimos de excursión el fin de semana mediante una puerta AND

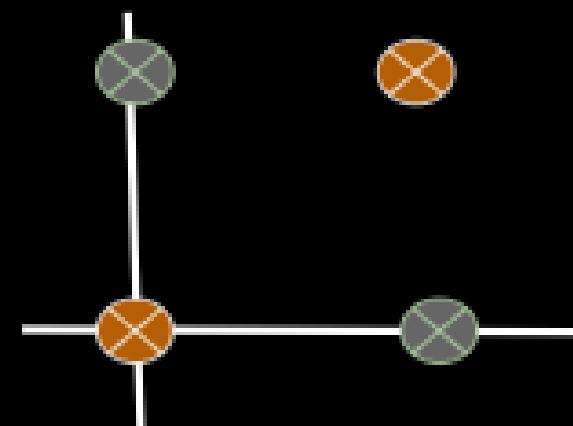
Buen dia	Colegas	Excursión
0	0	0
0	1	0
1	0	0
1	1	1



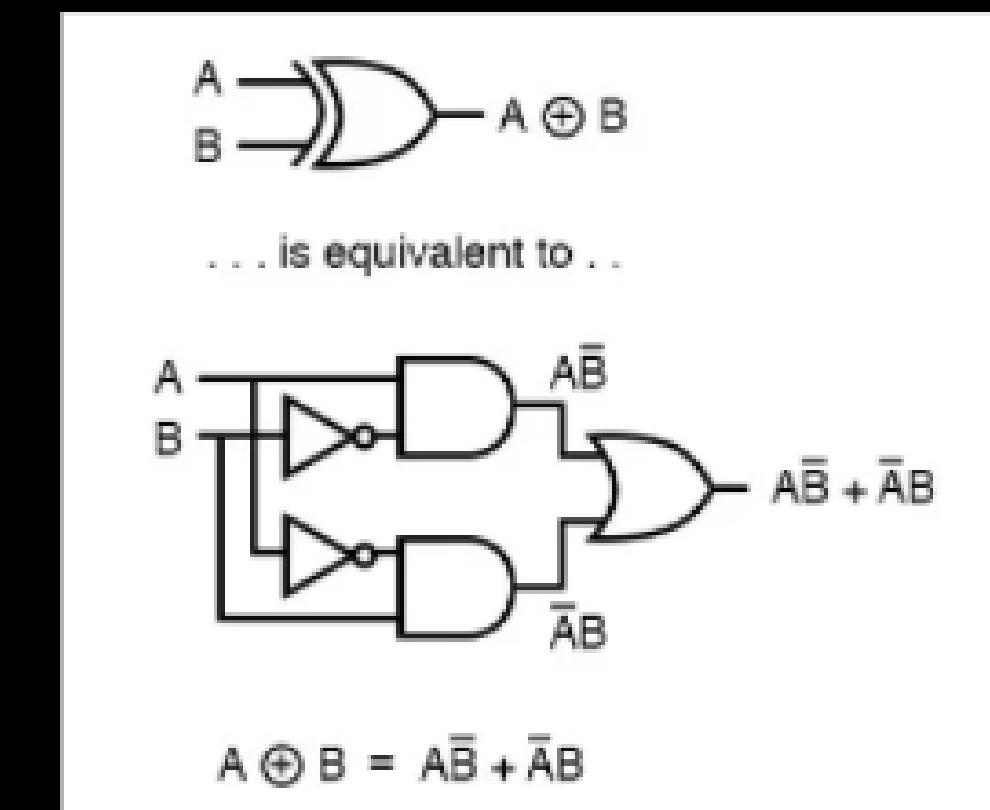
Problemas más complejos

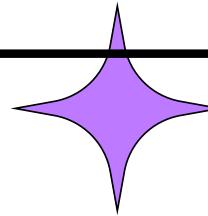
Vamos a intentar resolver una puerta XOR

Inputs	Outputs	
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0



Es **imposible** separar los puntos con un único hiperplano.
¿Solución? Combinar varias neuronas

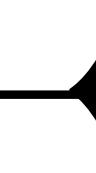




¿CÓMO SE ENTRENA?

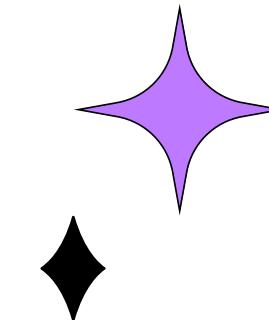
1. EL PERCEPTRÓN SE INICIALIZA CON UNOS PESOS A 0 O ALEATORIOS
2. CALCULA LAS SALIDAS, A TRAVÉS DE LA COMBINACIÓN LINEAL DE ENTRADAS Y PESOS + BIAS
3. ATRAVIESA LA FUNCIÓN DE ACTIVACIÓN
4. EVALÚA ERRORES EN LA SALIDA
5. VUELVE A AJUSTAR LOS PESOS HASTA QUE MINIMIZA EL ERROR.

No muy efectivo para problemas con separación no lineal



Perceptrón

Frank Rosenblatt, 1957

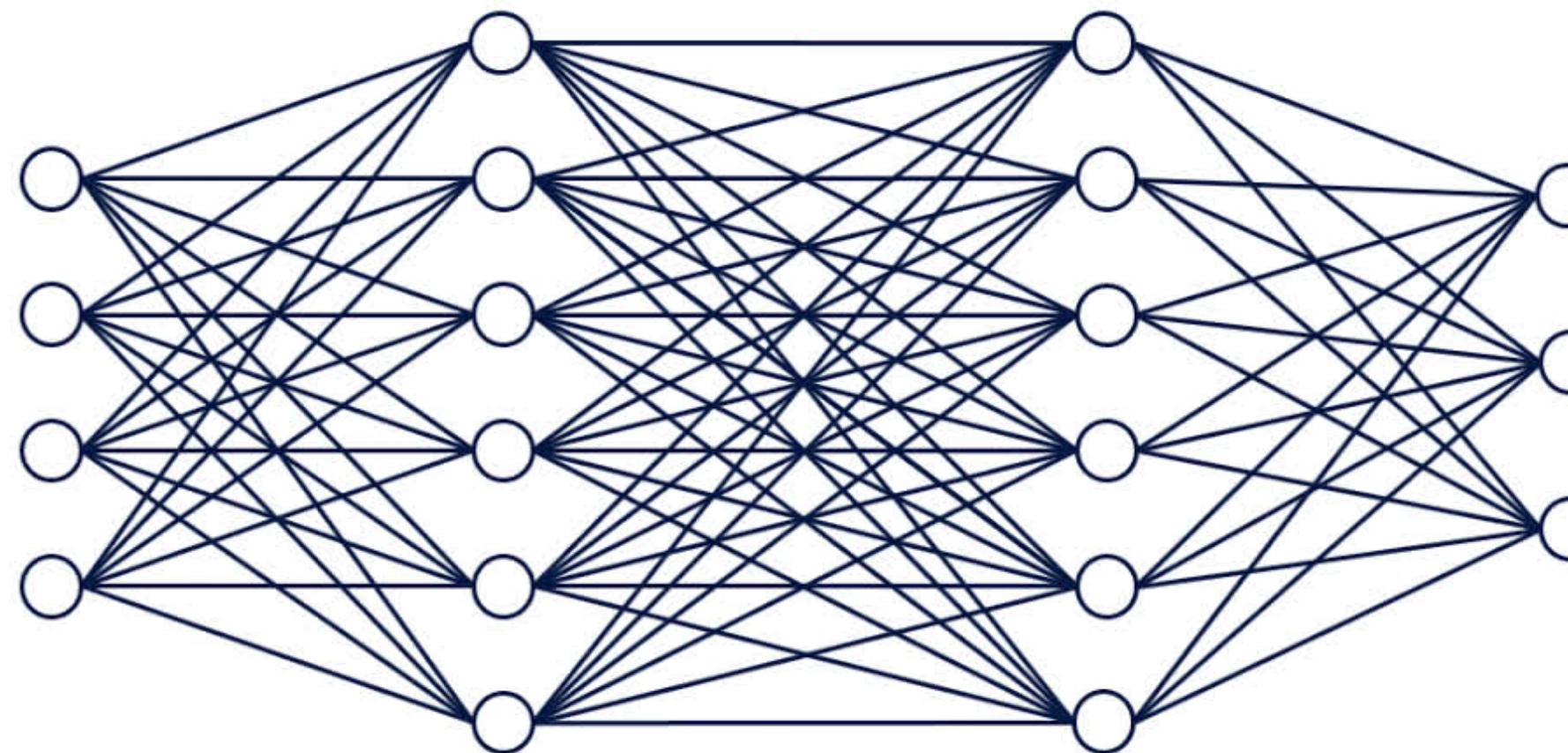


A diferencia de una simple neurona o una puerta lógica, aprende automáticamente los pesos de la neurona de McCulloch-Pitts para minimizar los errores.

Se utilizan para problemas separables linealmente. Ya no son datos binarios, sino que pueden ser números.

Red Neuronal Artificial

Multilayer Perceptron

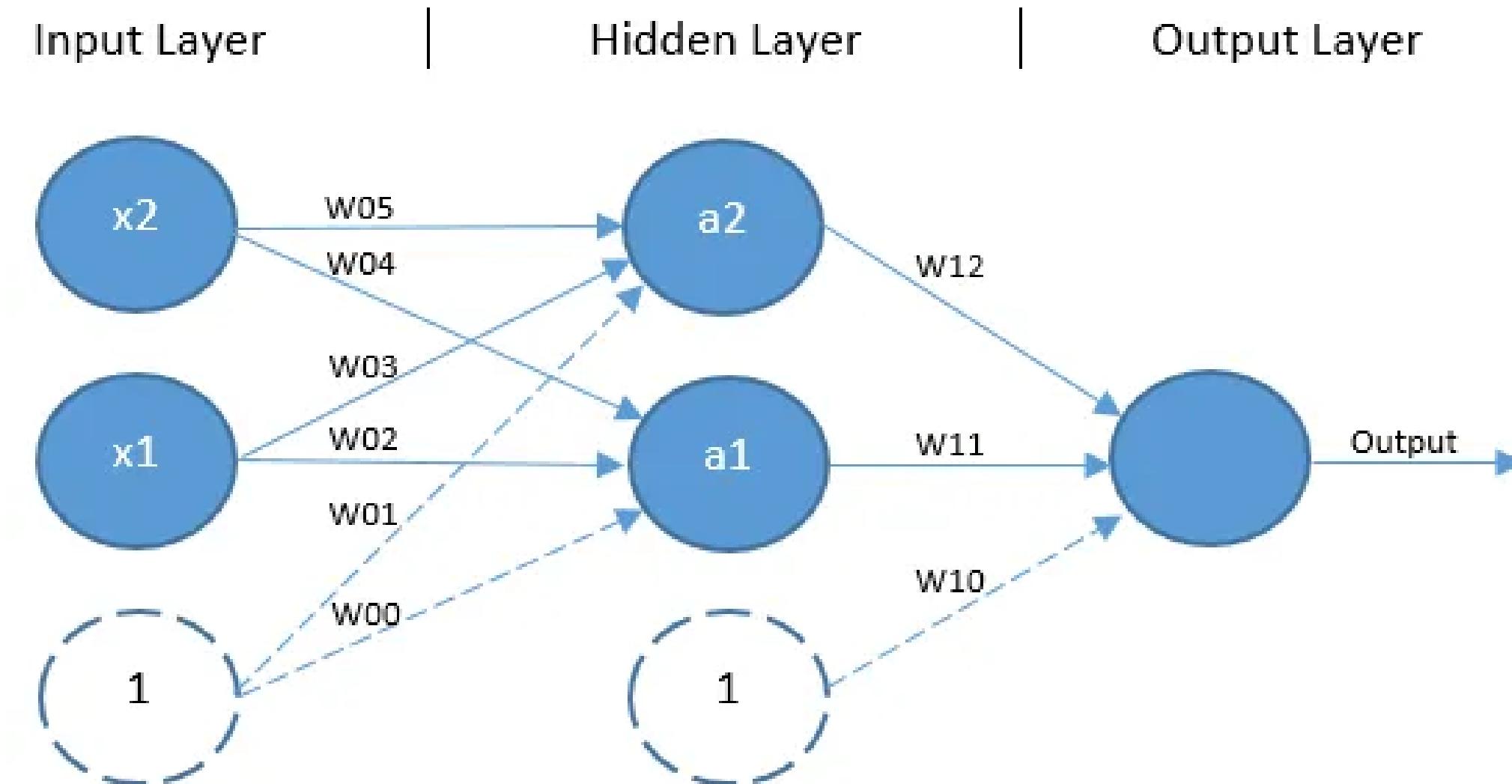


AGREGACIÓN DE NEURONAS EN CAPAS, DE TAL MANERA QUE LOS OUTPUTS DE UNA CAPA SON LOS INPUTS DE LA SIGUIENTE. CADA CONJUNTO DE NEURONAS SE VA ENTRENANDO Y ESPECIALIZANDO EN RESOLVER PARTES DEL PROBLEMA, COMO A SOLVENTAR UNA PUERTA **XOR**.

ALGORITMOS DE ENTRADA/SALIDA, QUE SIMULAN EL COMPORTAMIENTO DEL CEREBRO HUMANO. SE UTILIZAN UNA SERIE DE NEURONAS ARTIFICIALES O NODOS PARA SOLUCIONAR PROBLEMAS.

COMPONENTES MULTILAYER

1. **INPUT LAYER**: NO TIENEN CAPA DE ACTIVACIÓN. CADA NODO DE ENTRADA ES UNA FEATURE.
2. **HIDDEN LAYERS**: CAPAS INTERMEDIAS. PUEDE SER UNA O VARIAS. TIENEN FUNCIÓN DE ACTIVACIÓN Y TÉRMINO BIAS. CON UNA CAPA SE SUELTIOS LLAMAR MLP Y SI TIENE MÚLTIPLES DEEP NEURAL NETWORK.
3. **OUTPUT LAYER**: POSIBLES SALIDAS DE UNA RED NEURONAL.
 - A. **UNA NEURONA**: CLASIFICACIÓN BINARIA O REGRESIÓN
 - B. **VARIAS NEURONAS**: CLASIFICACIÓN MULTICLASE. TANTAS NEURONAS COMO CLASES.



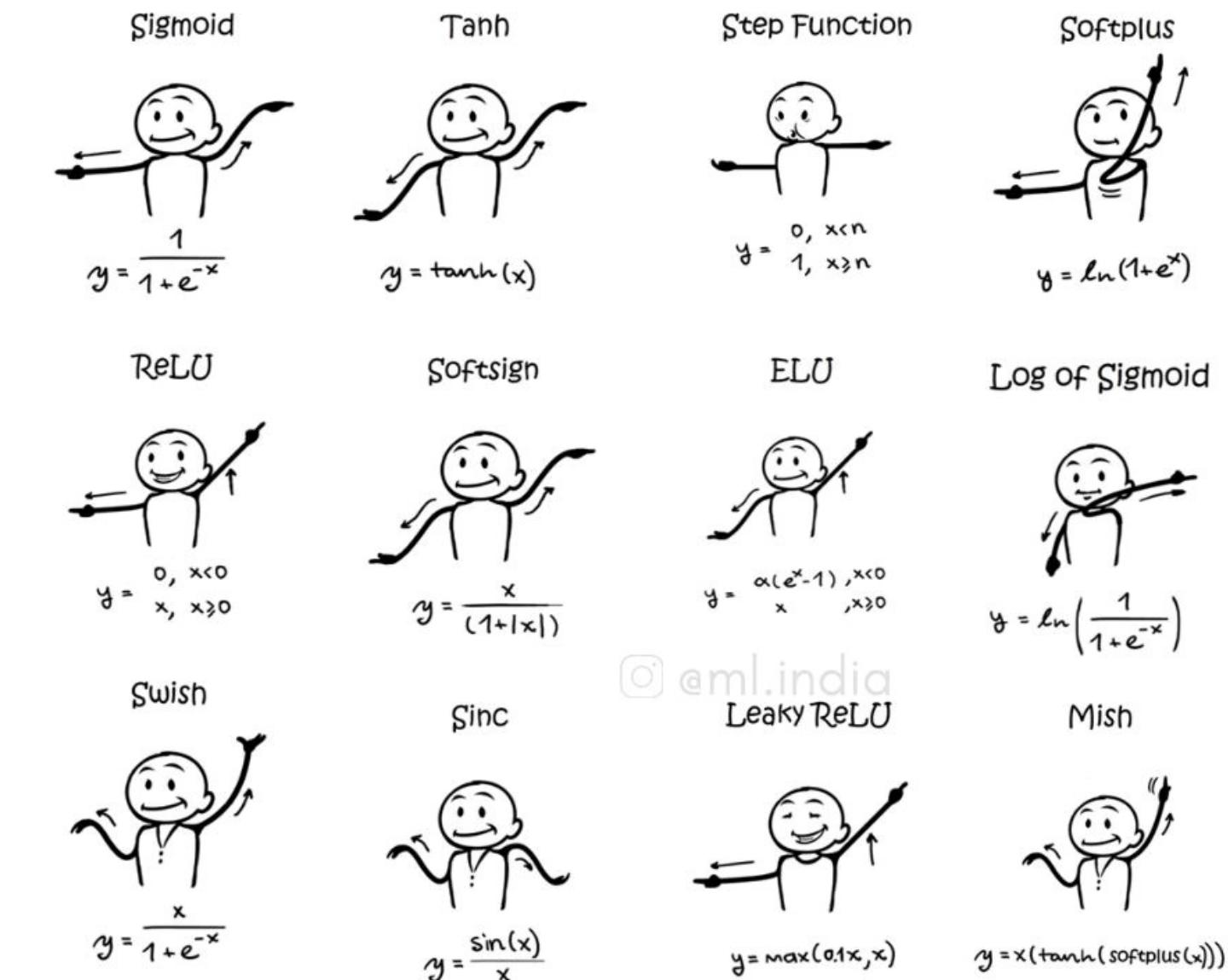
funciones activación

Permite a la red neuronal aprender patrones no lineares y adaptarse a problemas más complejos.

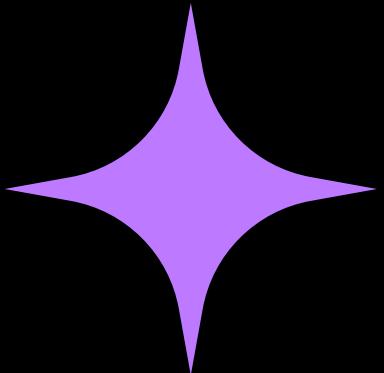
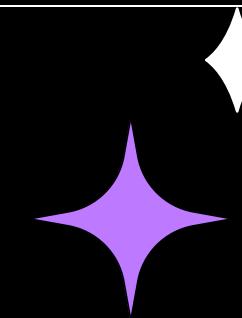
Dado que la **step function** no es derivable, hay que acudir a otras funciones de activación para poder recurrir en el entrenamiento a métodos como Gradient Descent.

La que más se suele utilizar es la **ReLU** (Rectified linear), a pesar de no ser derivable en 0. Computacionalmente es eficiente.

Deep Learning Activation Functions: Dance Moves



Conceptos

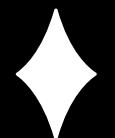


NEURONA ARTIFICIAL

UNIDAD LÓGICA INSPIRADA EN EL COMPORTAMIENTO DE LAS NEURONAS CEREBRALES. PERMITE RESOLVER PROBLEMAS SENCILLOS COMO UNA PUERTA LÓGICA AND U OR.

ANN - RED NEURONAL ARTIFICIAL

MODELO MATEMÁTICO QUE SIMULA LA FORMA EN QUE EL CEREBRO RESPONDE A ESTÍMULOS. ESTÁ COMPUESTA DE UNA O MÁS NEURONAS ARTIFICIALES. SE UTILIZA EN PROBLEMAS DE CLASIFICACIÓN O REGRESIÓN. UN DIAGRAMA DE RED DEFINE LAS RELACIONES ENTRE LAS SEÑALES DE ENTRADA (X) Y SALIDA (Y).



PERCEPTRÓN

LA RED NEURONAL ARTIFICIAL MÁS SIMPLE. EMPLEA UNA NEURONA ARTIFICIAL PARA RESOLVER PROBLEMAS DE CLASIFICACIÓN Y REGRESIÓN. EL PERCEPTRÓN ENTRENA CON LOS DATOS DE ENTRADA COMO CUALQUIER ALGORITMO DE MACHINE LEARNING, INTENTANDO MINIMIZAR SUS ERRORES.

MLP - MULTILAYER PERCEPTRON

SE TRATA DE UNA ANN COMPUESTA POR VARIAS NEURONAS, PERO EN ESTE CASO SE DISTRIBUYEN EN CAPAS. TIENE UNA INPUT LAYER, VARIAS HIDDEN LAYERS Y UNA OUTPUT LAYER. SE SUELE DENOMINAR MLP A LAS REDES NEURONALES CON UNA SOLA HIDDEN LAYER. TAMBIÉN SE LES LLAMA SHALLOW NEURAL NETWORK.

DEEP NEURAL NETWORK

LAS REDES CON UNA ÚNICA HIDDEN LAYER SE SUELEN DENOMINAR MLP, MIENTRAS QUE LAS QUE TIENEN VARIAS HIDDEN LAYERS SON DEEP NEURAL NETWORKS.

DEEP LEARNING

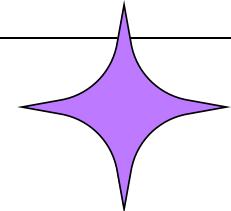
ALGORITMOS DE MACHINE LEARNING QUE UTILIZAN REDES NEURONALES PARA APRENDER PATRONES EN LOS DATOS Y RESOLVER PROBLEMAS COMPLEJOS COMO TRATAMIENTO DE IMÁGENES O DE SONIDO.

¿CÓMO SE ENTRENA UNA RED NEURONAL?

1. SE INICIALIZAN LOS PESOS DE LAS CONEXIONES DE MANERA ALEATORIA
2. PASAMOS LOS DATOS DE ENTRENAMIENTO A TRAVÉS DE LA RED
3. COMPARAMOS A LA SALIDA CON LOS TRUE LABELS
4. ESTO NOS DA UN ERROR (COST FUNCTION), QUE DEPENDE DE TODOS LOS PESOS Y LOS BIAS DE LA RED. EL ERROR NO ES MÁS QUE EL MSE (MEAN SQUARED ERROR)
5. ¿OBJETIVO? MINIMIZAR ESE ERROR. → OPTIMIZADORES (GRADIENT DESCENT, SGD, ADAM...)
6. SE PROPAGA EL ERROR HACIA ATRÁS PARA VER QUÉ NEURONA HA TENIDO MAYOR RESPONSABILIDAD.
7. SE COMPUTA EL GRADIENTE PARA CADA UNO DE LOS PESOS, Y SE ACTUALIZAN EN LA DIRECCIÓN DE MINIMIZACIÓN DEL ERROR
8. CUANDO EL GRADIENTE SEA 0, YA HEMOS ALCANZADO EL MÍNIMO DE ERROR, EL ALGORITMO TERMINA, Y SI NO, VOLVEMOS AL PUNTO 6.

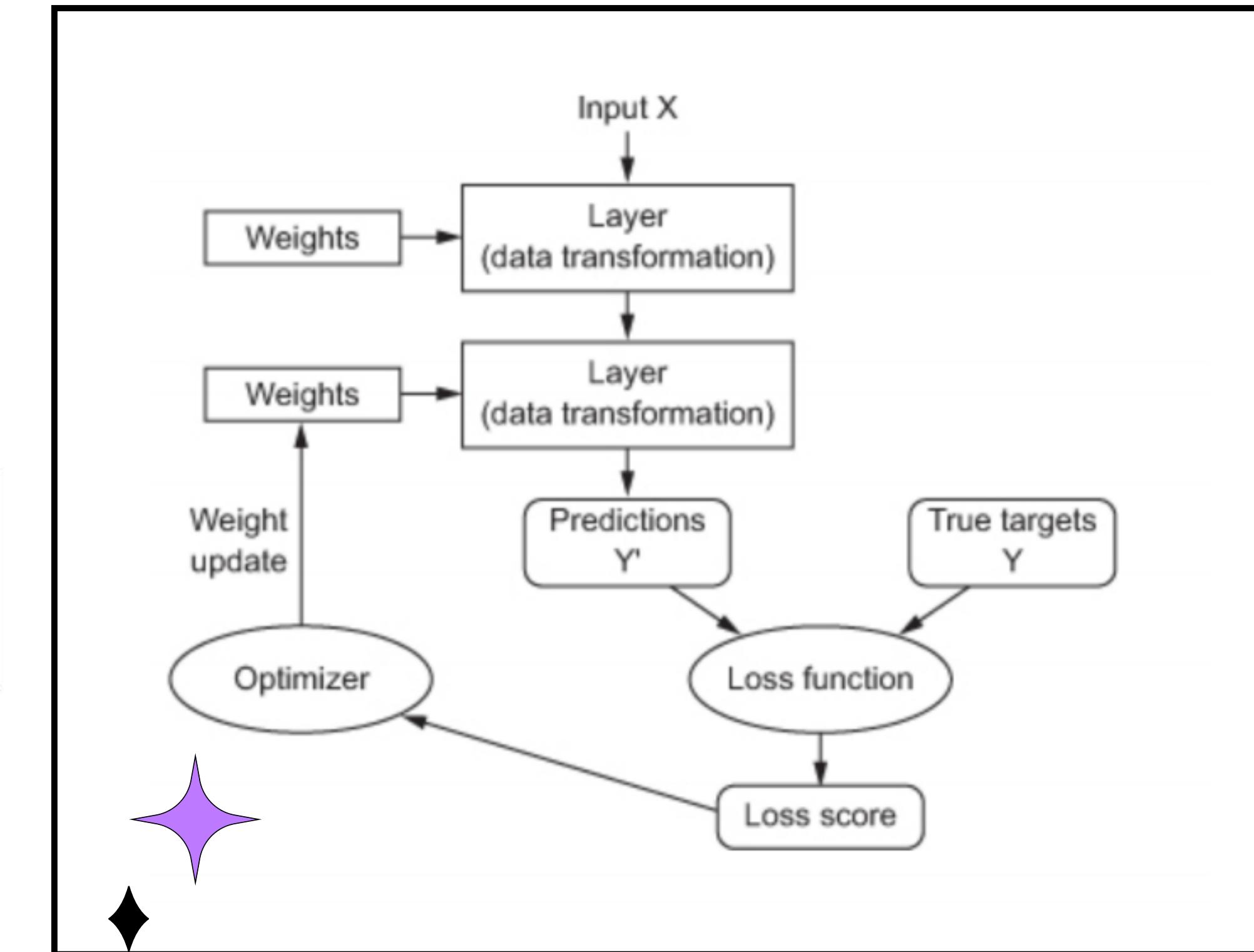
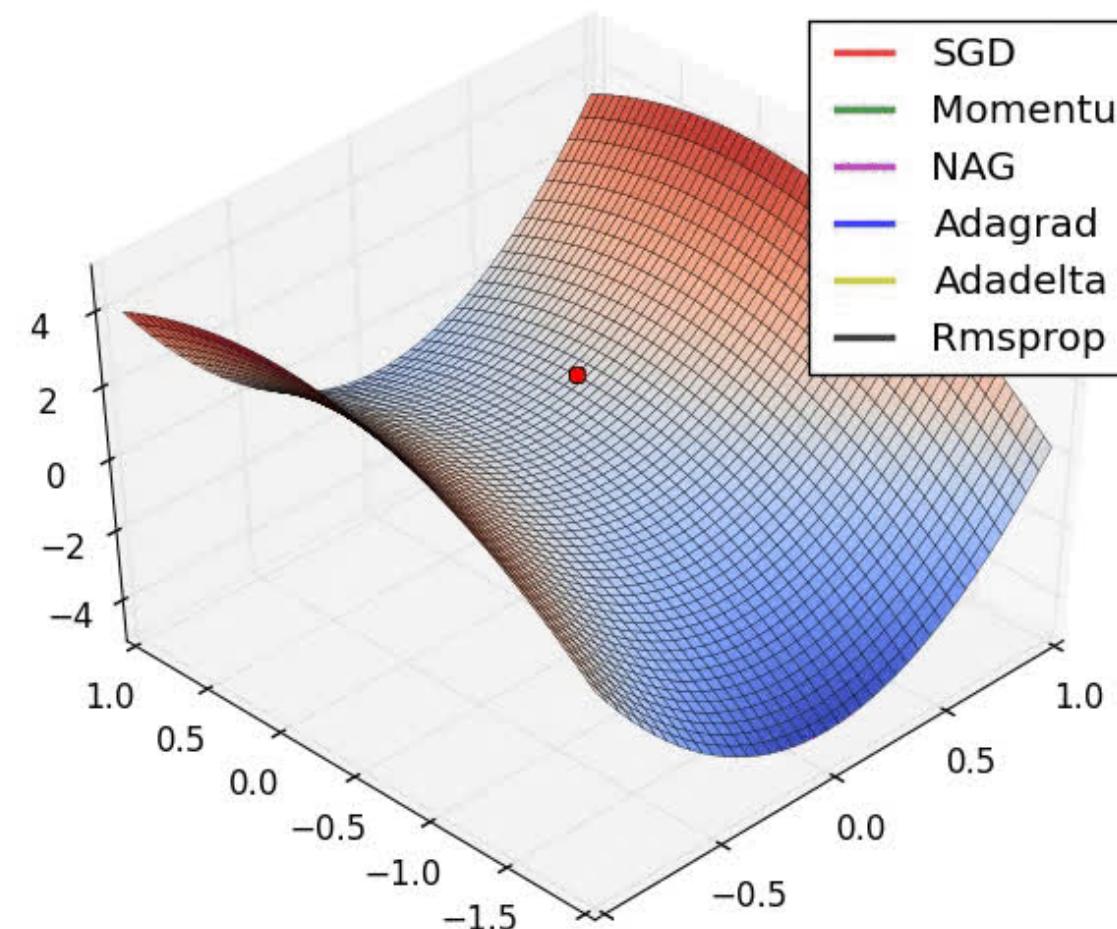
Backpropagation

Algoritmo de entrenamiento de una red neuronal. Tras la forward propagation, la predicción se compara con el true label y obtenemos un error. Este error se propaga hacia atrás para corregir los valores de los pesos de las funciones de transformación de las neuronas



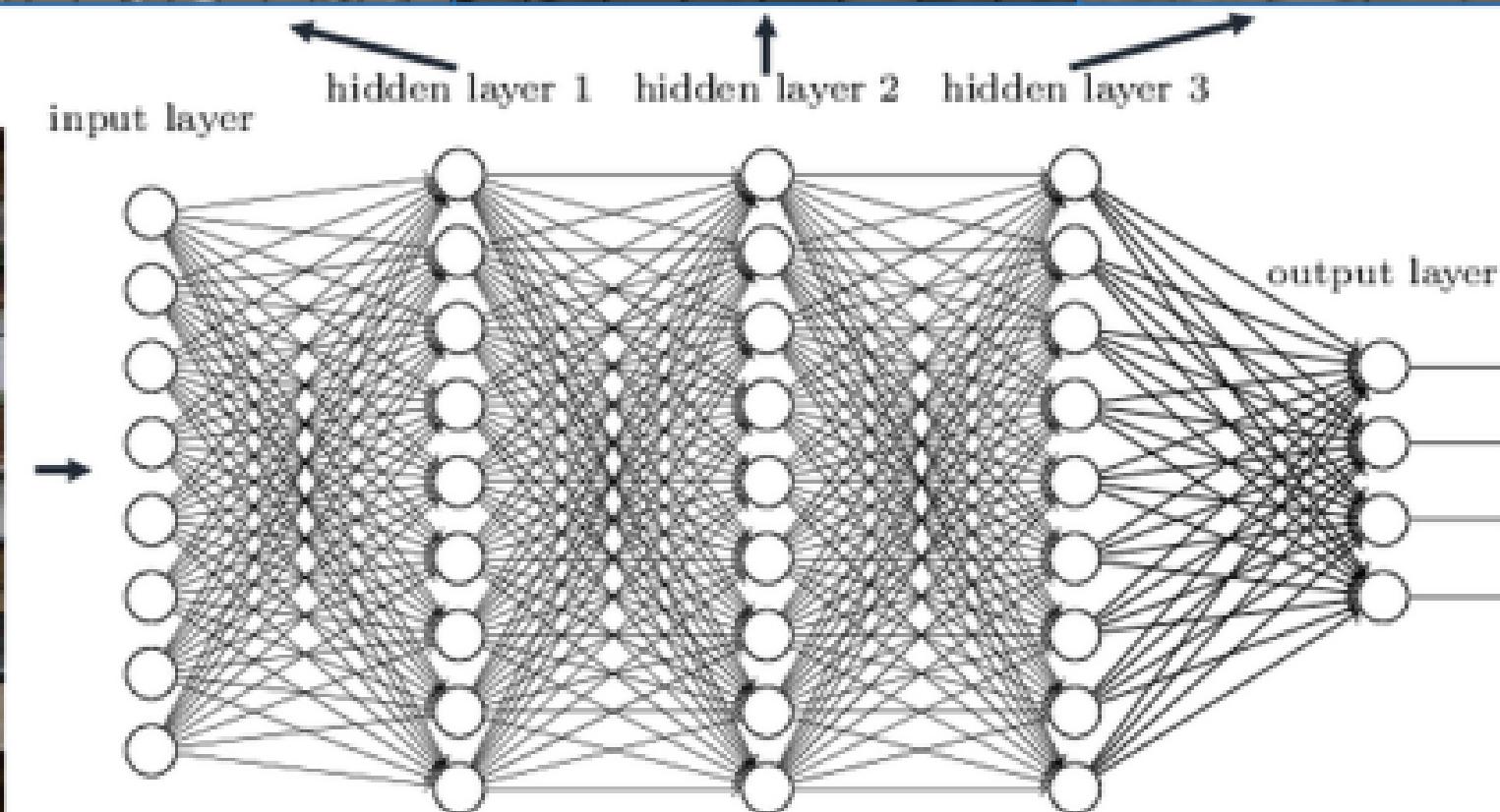
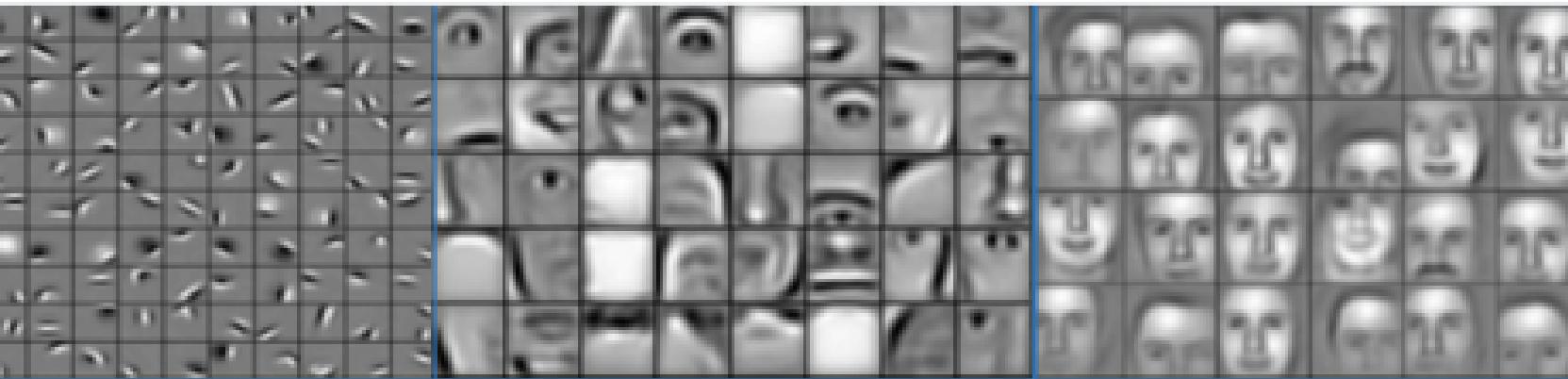
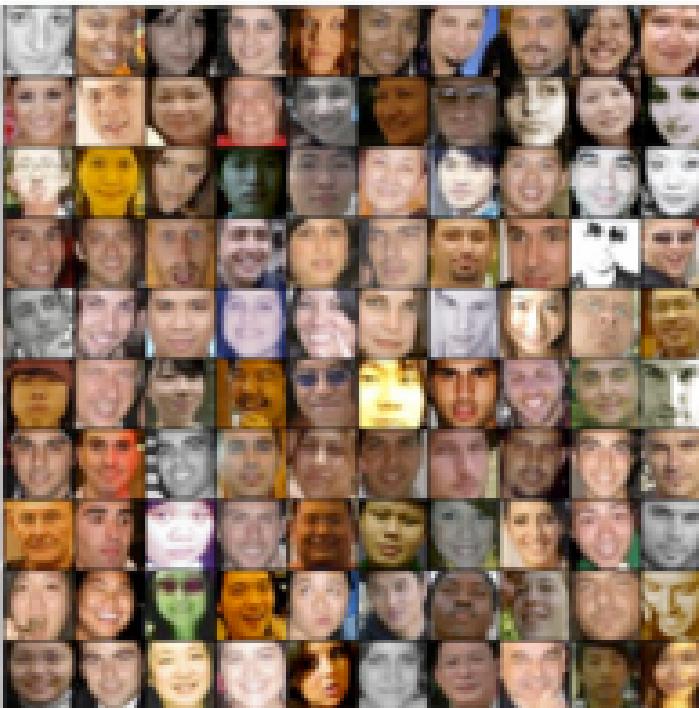
Optimización

Algoritmos iterativos que se encargan de llegar al mínimo de esa función, como el Gradient Descent.



Convolutional Neural Net

Deep neural networks learn hierarchical feature representations



RNN,
NLP,
GAN
and more...

Paquetes de R

CARET

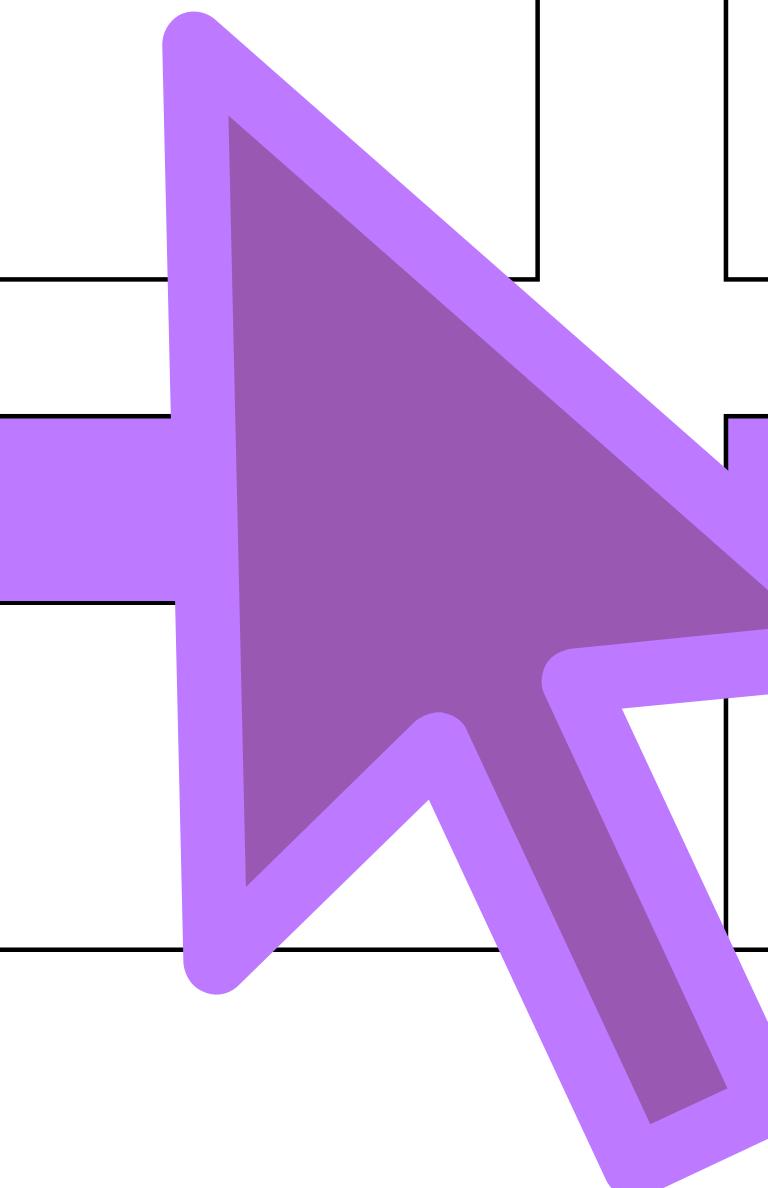
NEURALNET

H2O

MXNET

KERAS

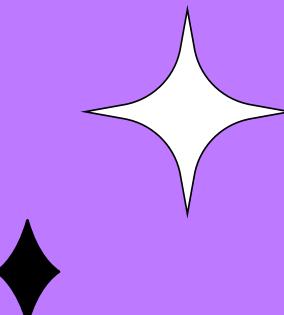
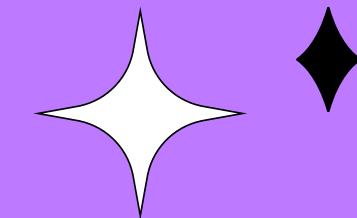
TENSORFLOW





Con código...

NUESTRO OBJETIVO ES PREDECIR SI UNA
PERSONA ENTRARÁ EN UNA ESCUELA
UTILIZANDO COMO INPUT EL
CONOCIMIENTO TÉCNICO Y SU CAPACIDAD
DE COMUNICACIÓN



**Do you
have any
questions?**

CLARA PINIELLA MARTINEZ

E M A I L A D D R E S S

clapiniella@gmail.com

#CODELIKEAGIRL