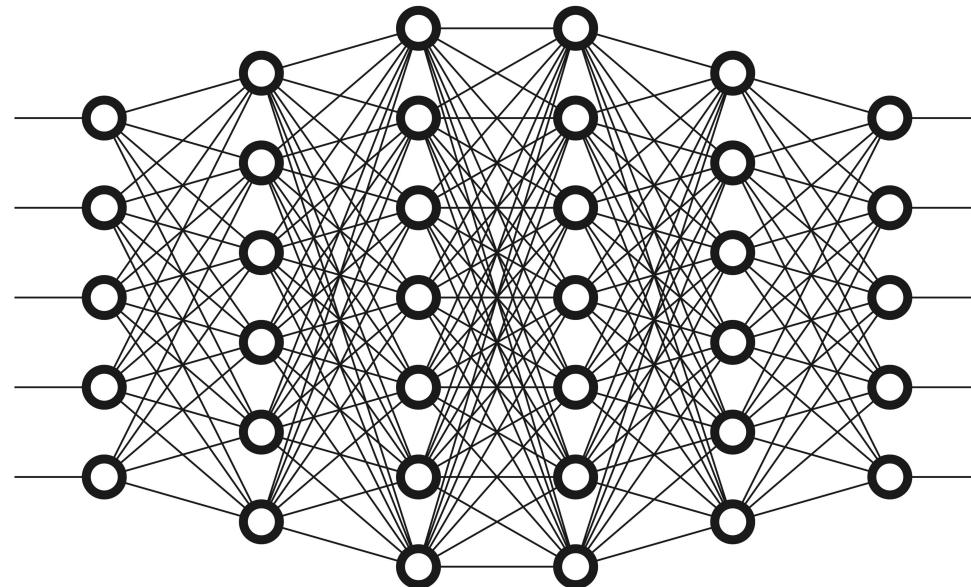

Clase 1

Introducción al aprendizaje profundo

Enzo Ferrante

 eferrante@sinc.unl.edu.ar

 @enzoferante





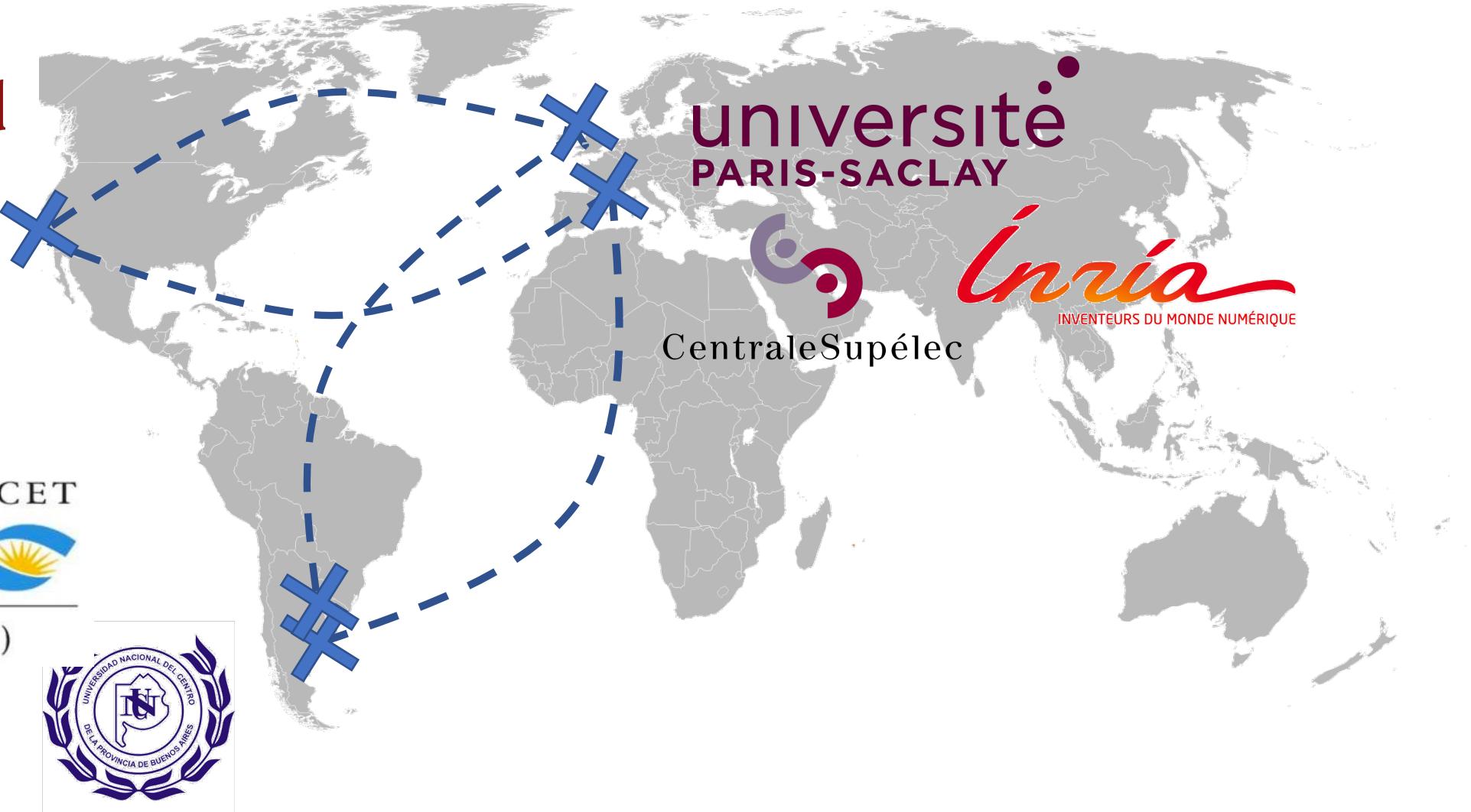
Imperial College
London



UNIVERSITY OF
CAMBRIDGE



Stanford
University

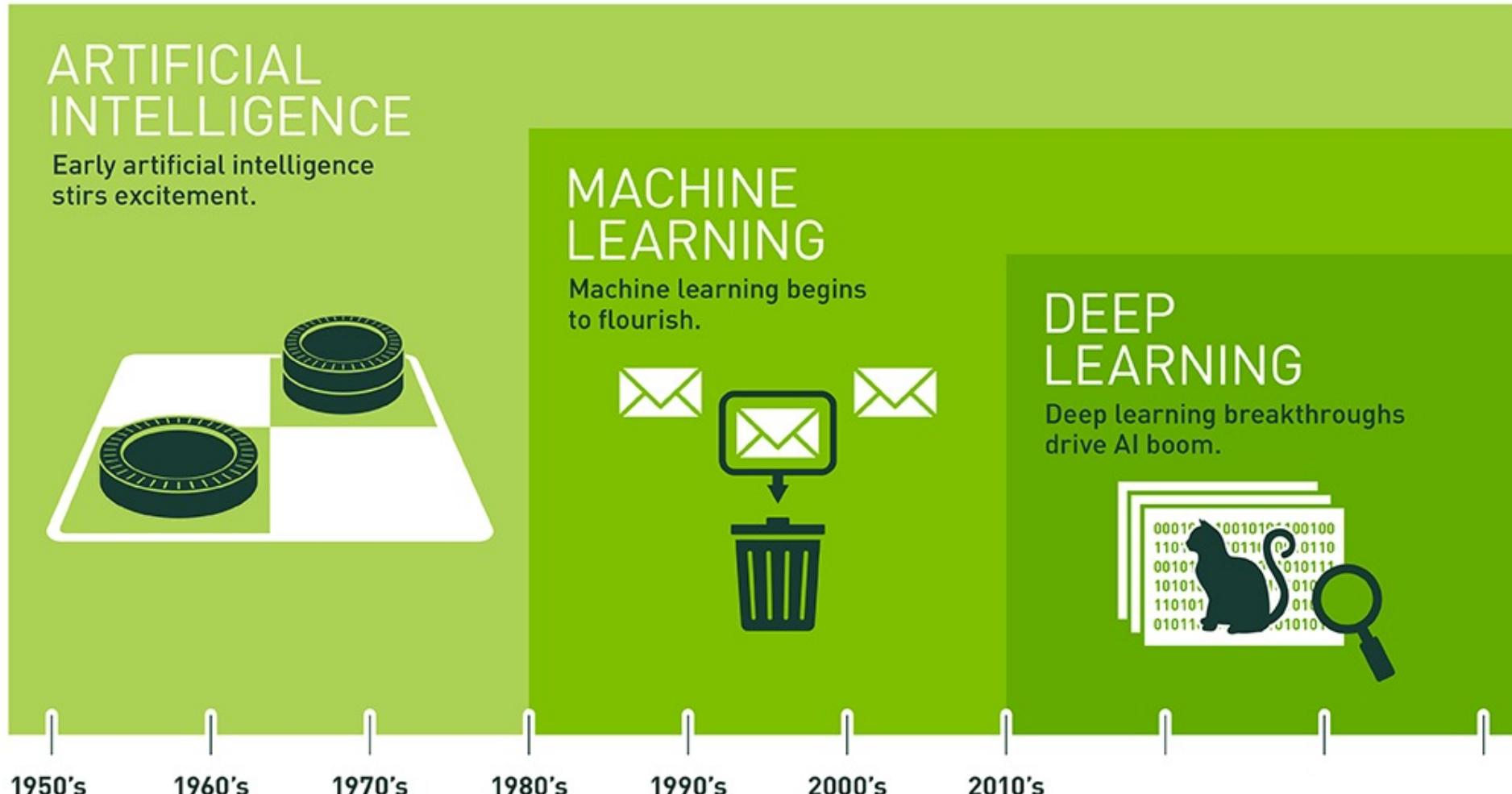




Machine Learning for Biomedical Image Analysis

Aclarando conceptos: IA, Aprendizaje Automático y Aprendizaje Profundo

Deep Learning



Inteligencia Artificial

Sus orígenes se consideran en la Conferencia de
Inteligencia Artificial de Dartmouth en 1955

A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

J. McCarthy, Dartmouth College

M. L. Minsky, Harvard University

N. Rochester, I.B.M. Corporation

C.E. Shannon, Bell Telephone Laboratories

August 31, 1955

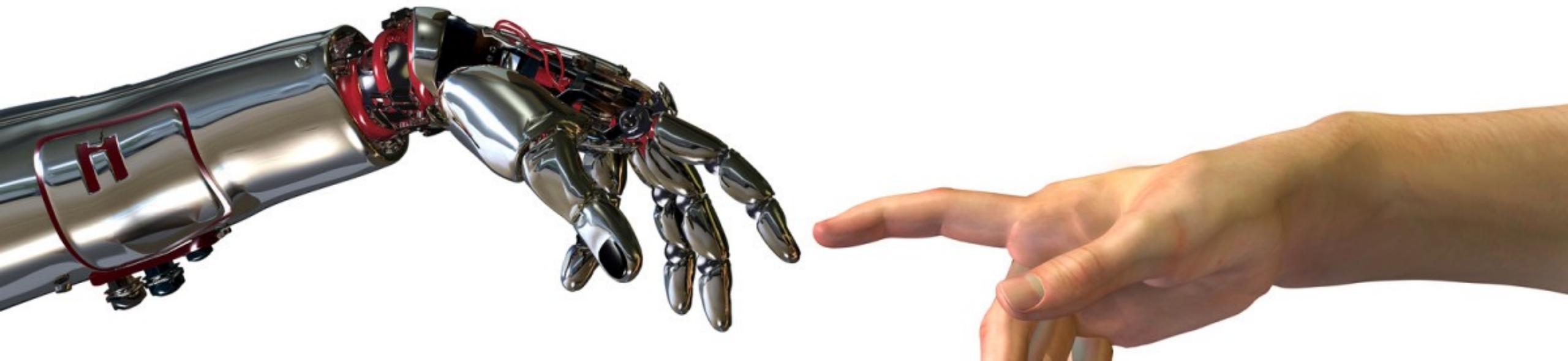
We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.

Inteligencia Artificial

Comportamiento humano realizado por máquinas

inteligencia artificial.

1. f. Inform. Disciplina científica que se ocupa de crear programas informáticos que ejecutan operaciones comparables a las que realiza la mente humana, como el aprendizaje o el razonamiento lógico.



Aprendizaje Automático

Una estrategia para alcanzar la inteligencia artificial

Aprendizaje automático.

1. El estudio y construcción de algoritmos que pueden aprender de y hacer predicciones sobre datos.

Término acuñado en 1959 por Arthur Samuel (IBM)

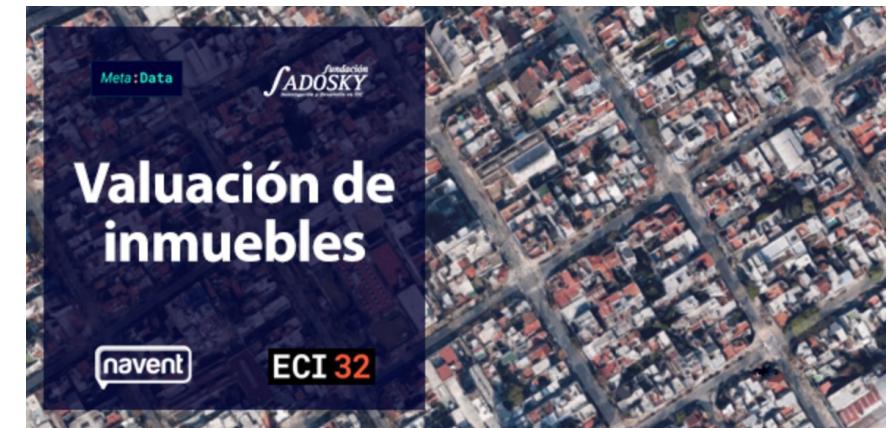
Aprendizaje Automático

Ejemplos de problemas que pueden ser resueltos con aprendizaje automático

- *direccion*: La dirección de la propiedad
- *ciudad*: La ciudad de la propiedad
- *provincia*: La provincia donde está localizada la propiedad
- *lat*: Latitud
- *Ing*: Longitud
- *tipodepropiedad*: El tipo de propiedad (Casa, departamento, etc)
- *metros totales*: Metros totales de la propiedad
- *metros cubiertos*: Metros cubiertos de la propiedad
- *antiguedad*: Antigüedad de la propiedad
- *habitaciones*: Cantidad de habitaciones
- *garages*: Cantidad de garajes
- *banos*: Cantidad de baños
- *fecha*: Fecha de publicación
- *gimnasio*: Si el edificio o la propiedad tiene un gimnasio
- *usos multiples*: Si el edificio o la propiedad tiene un SUM
- *piscina*: Si el edificio o la propiedad tiene una piscina
- *escuelas cercanas*: Si la propiedad tiene escuelas cerca
- *centros comerciales cercanos*: Si la propiedad tiene centros comerciales cerca



Valor del inmueble



Aprendizaje Automático

Ejemplos de problemas que pueden ser resueltos con aprendizaje automático

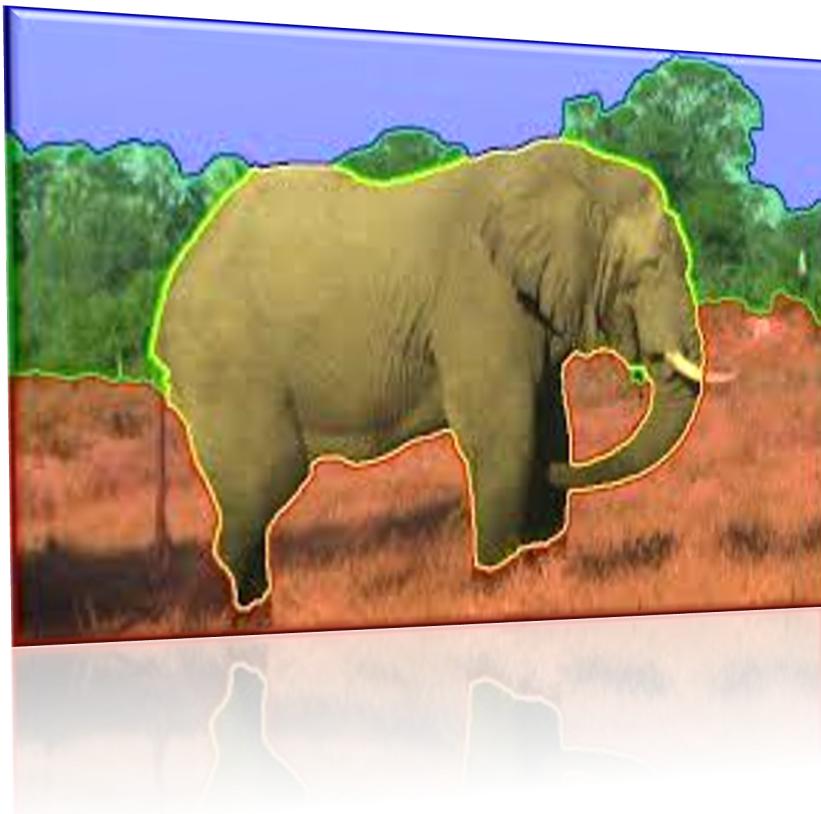


Chihuaua vs Muffin

Clasificación de imágenes

Aprendizaje Automático

Ejemplos de problemas que pueden ser resueltos con aprendizaje automático



Segmentación de imágenes

Aprendizaje Automático

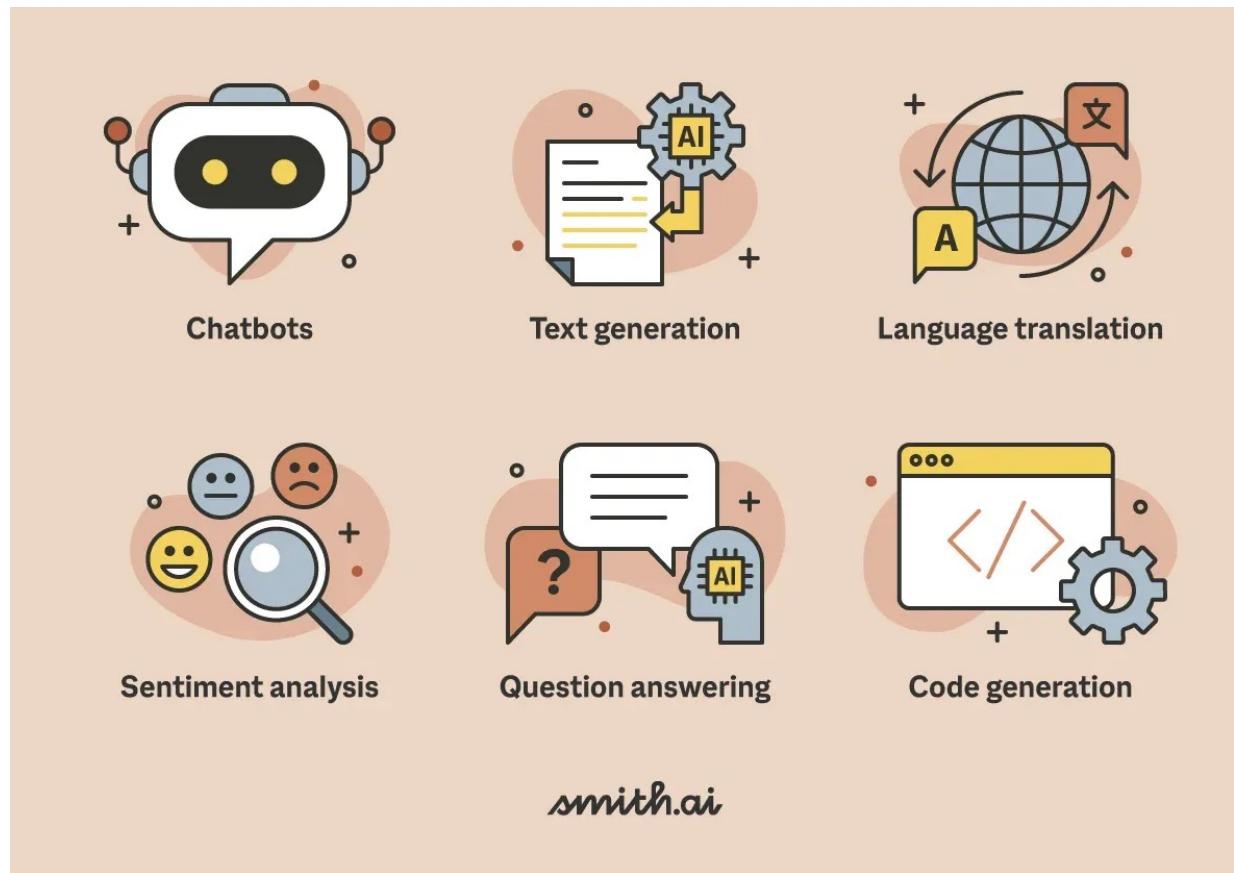
Ejemplos de problemas que pueden ser resueltos con aprendizaje automático



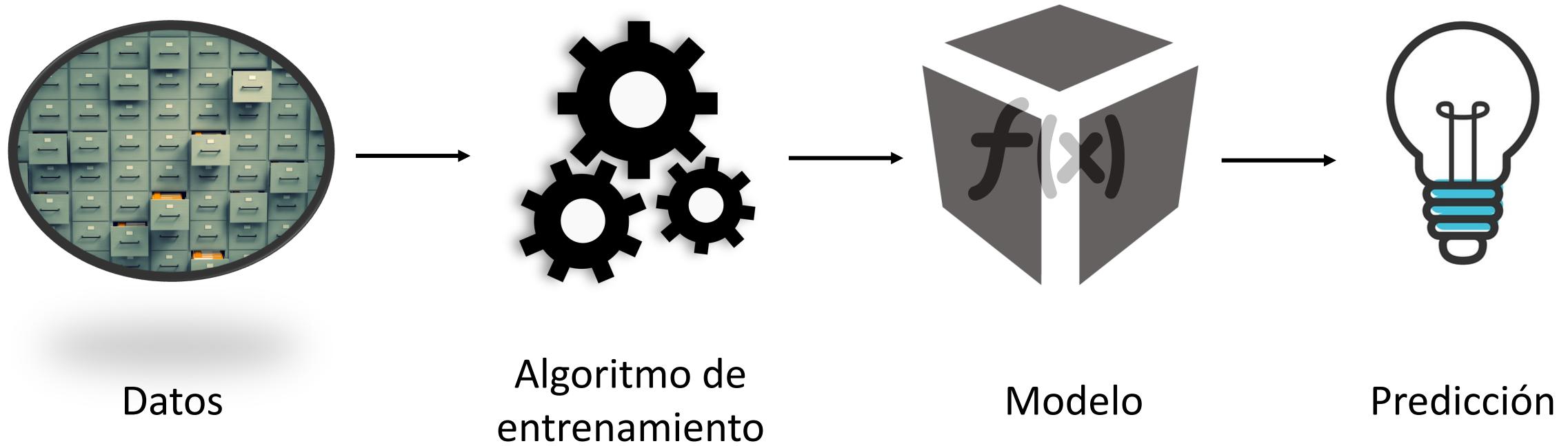
Reconocimiento de voz

Aprendizaje Automático

Ejemplos de problemas que pueden ser resueltos con aprendizaje automático



Aprendizaje Automático



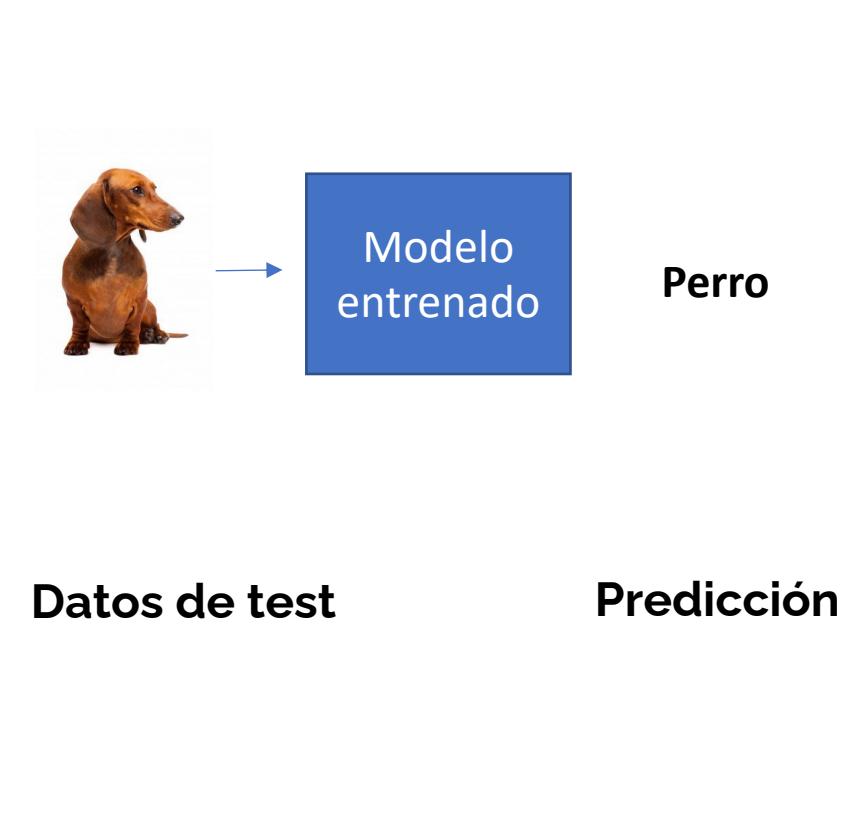
Aprendizaje supervisado

Datos de entrenamiento

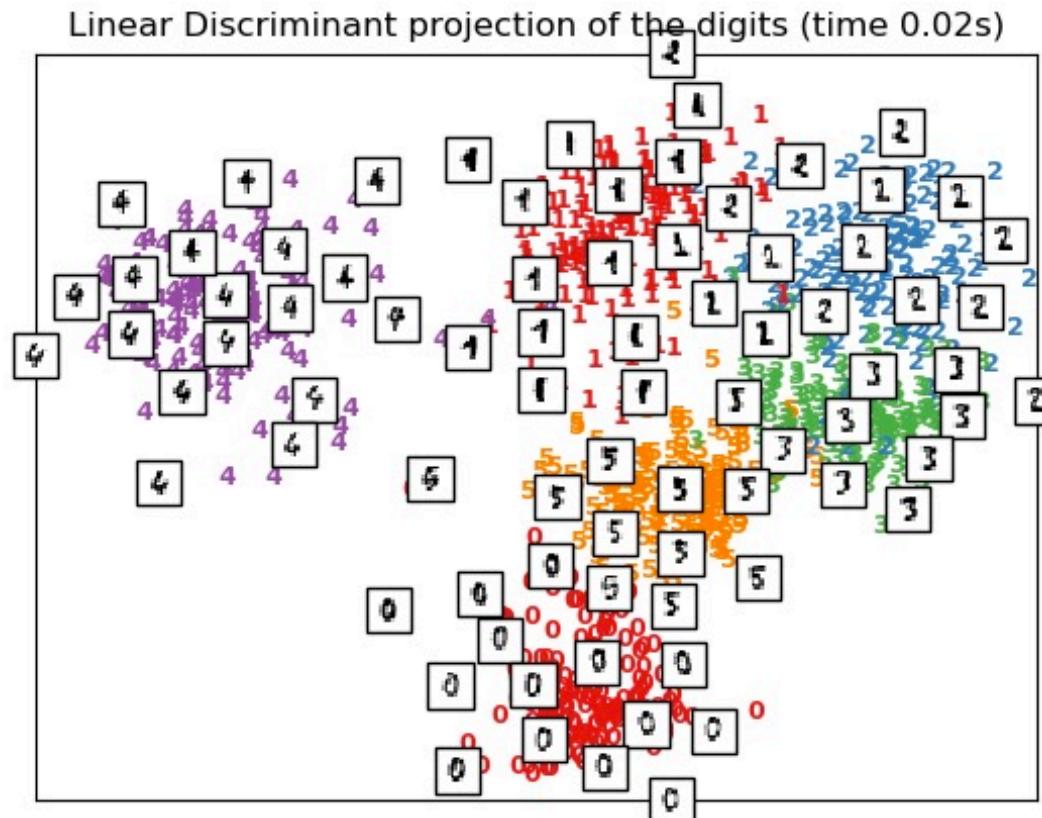
Dato	Etiqueta	Dato	Etiqueta
	Perro		Gato
	Perro		Gato
	Perro		Gato
	Perro		Gato



Modelo
entrenado



Aprendizaje no-supervisado

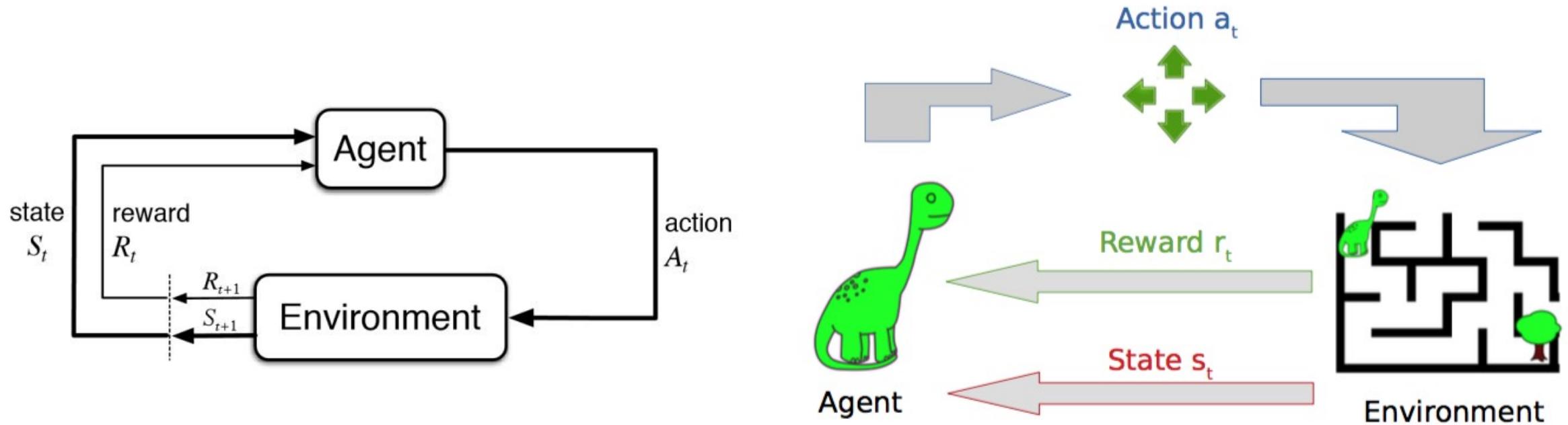


Aprendizaje semi-supervisado

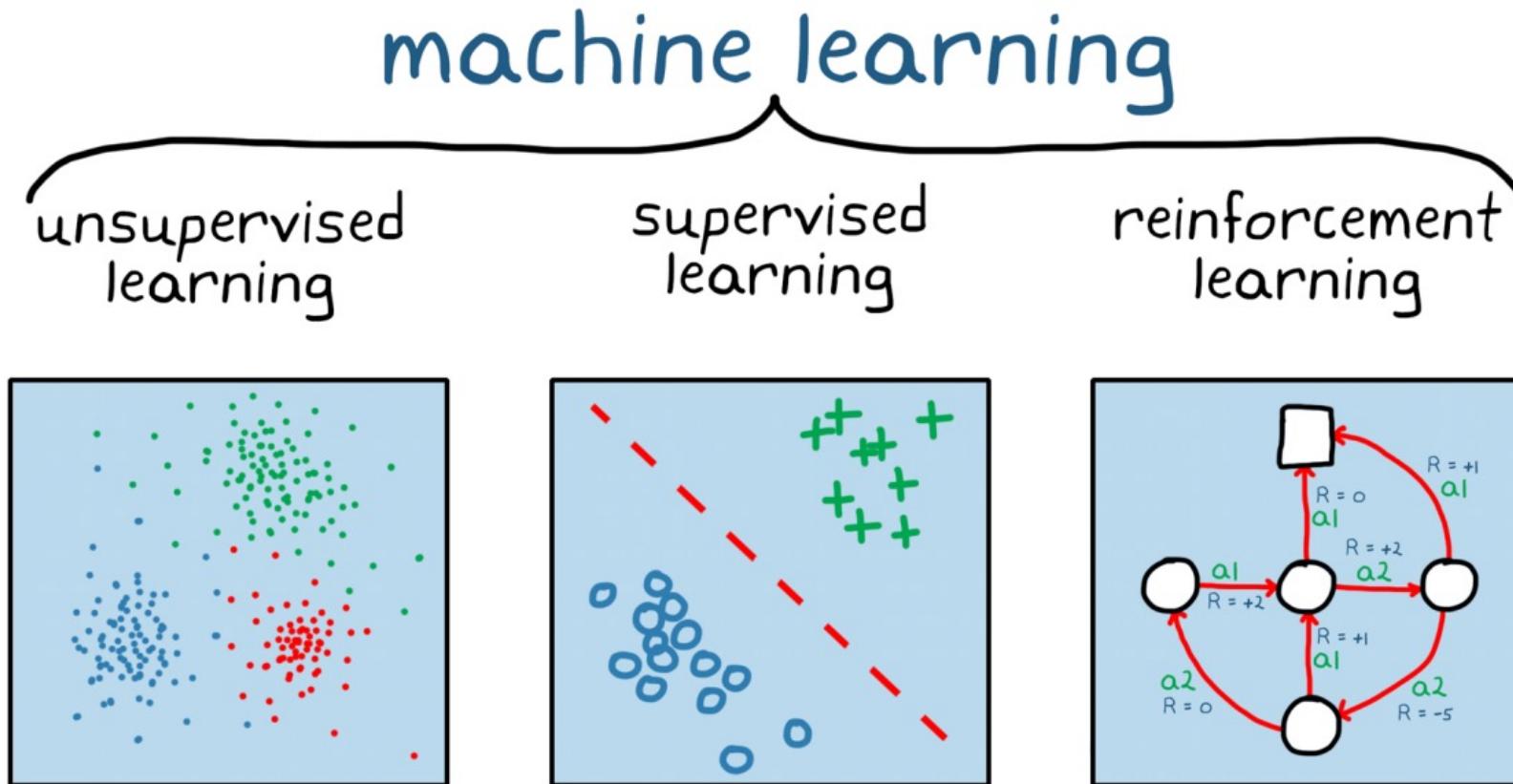
Los algoritmos de aprendizaje semi supervisado utilizan tanto datos etiquetados como no etiquetados durante el proceso de aprendizaje

Aprendizaje por refuerzo

Aprendizaje a partir de la interacción con el ambiente

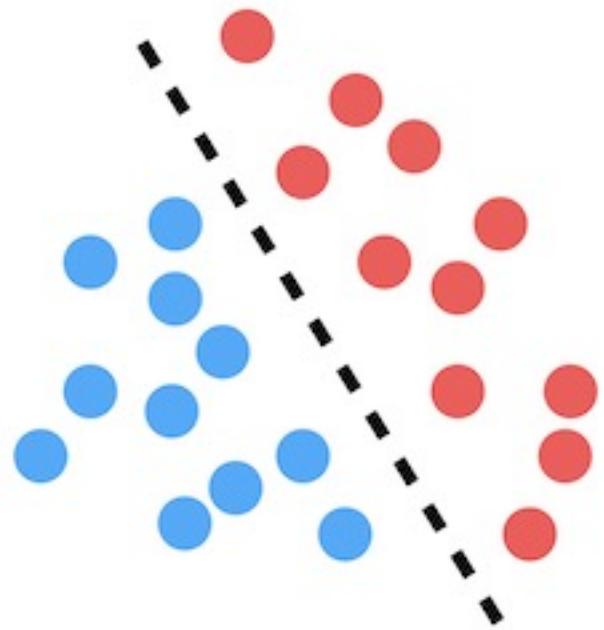


Aprendizaje automático

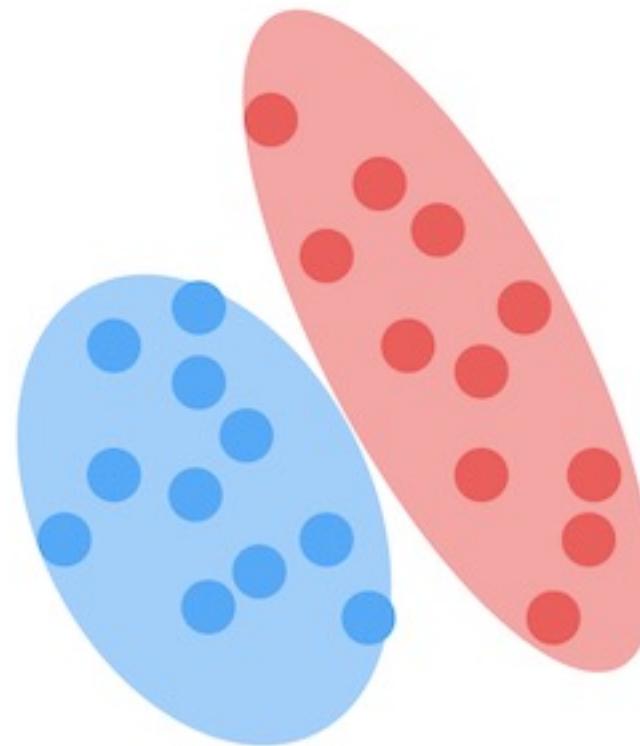


Tipos de modelo

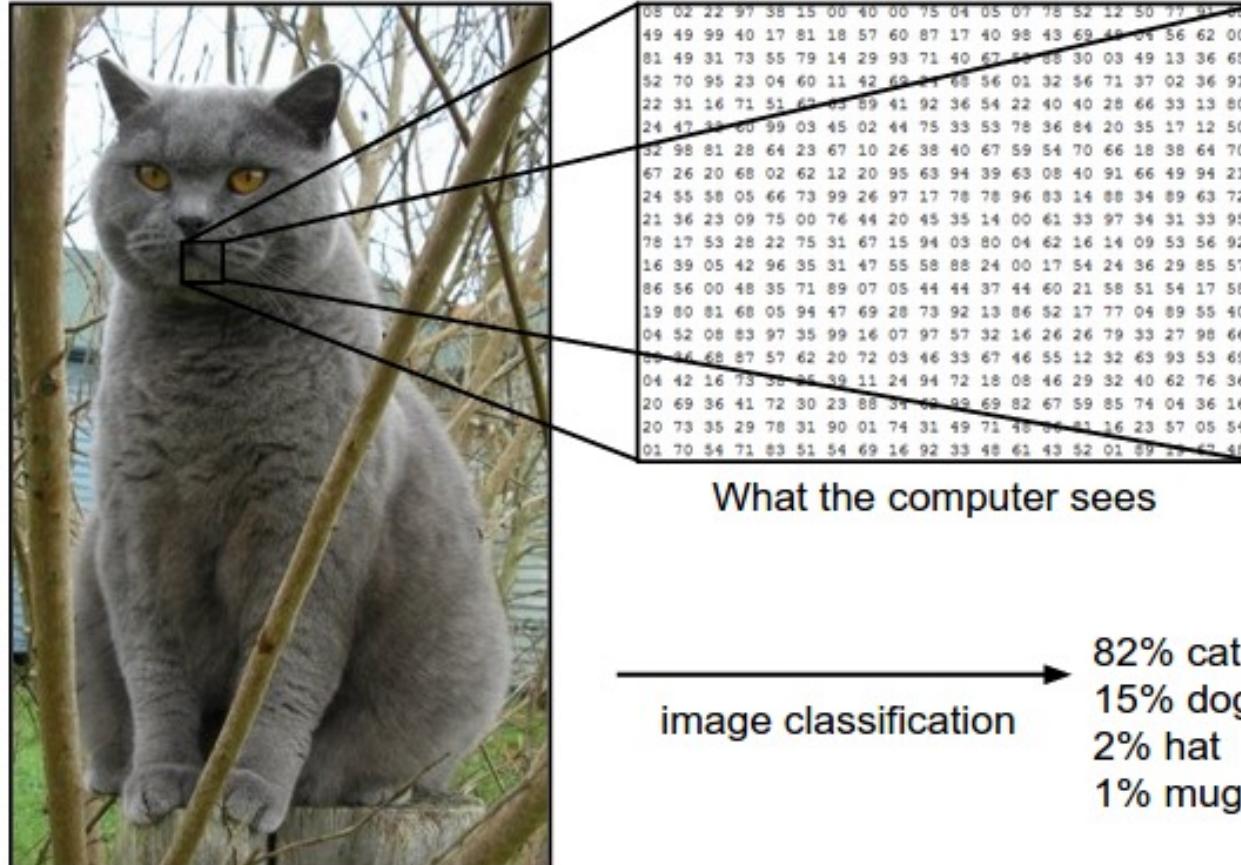
Discriminative



Generative



Ejemplo: clasificación de imágenes



Clasificación de imágenes



23	11	12	200	34	35	45	46	50	25
1	89	1	33	78	60	55	1	76	99
89	0	56	45	90	91	88	3	80	87
67	56	77	90	23	1	99	34	90	68
44	99	80	65	40	100	8	7	8	49
55	32	23	55	71	0	19	200	33	45
45	34	21	44	207	65	18	33	12	77
88	45	22	33	45	66	78	89	0	77
9	10	13	57	89	88	90	200	208	100
11	23	12	7	209	56	78	45	88	78

Tiene 4 patas

Tiene cola

Tiene dos orejas largas

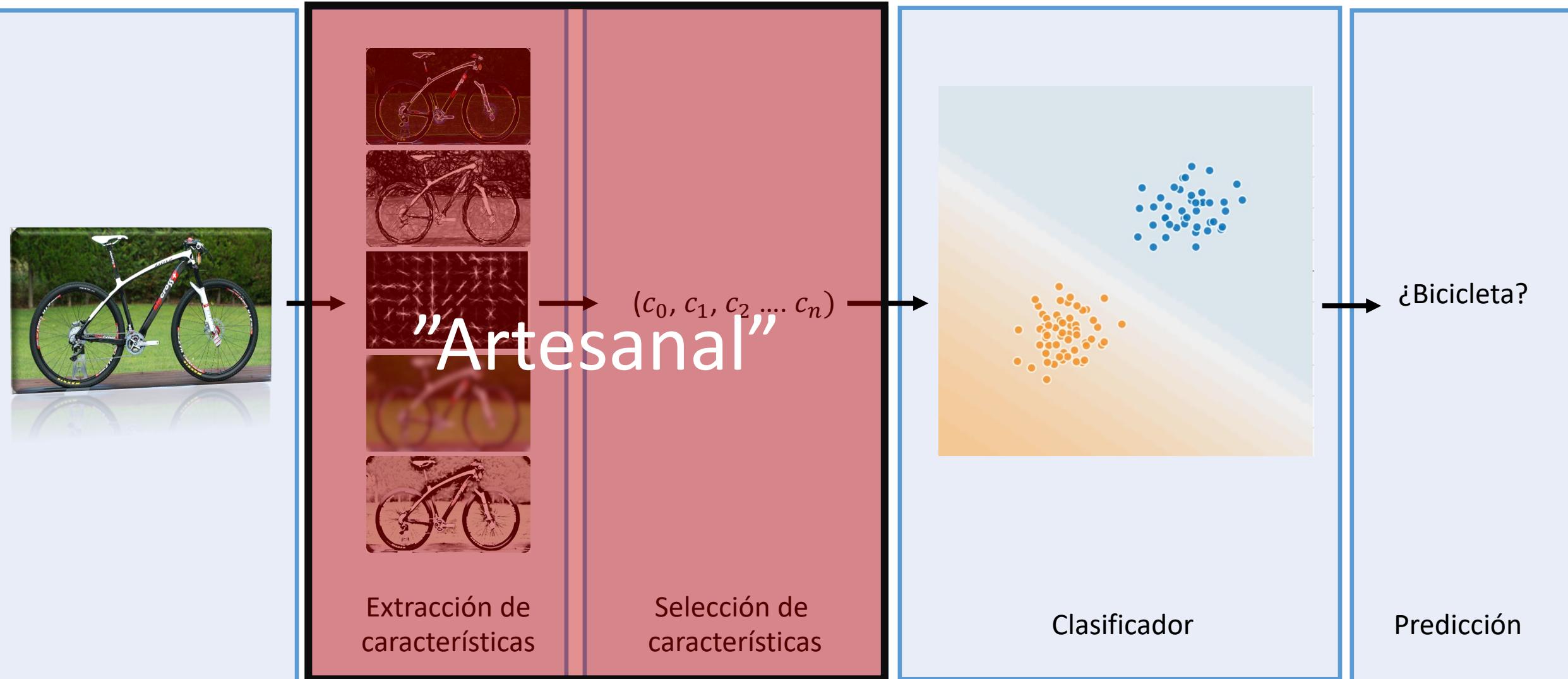
Tiene ocico

Es bajito

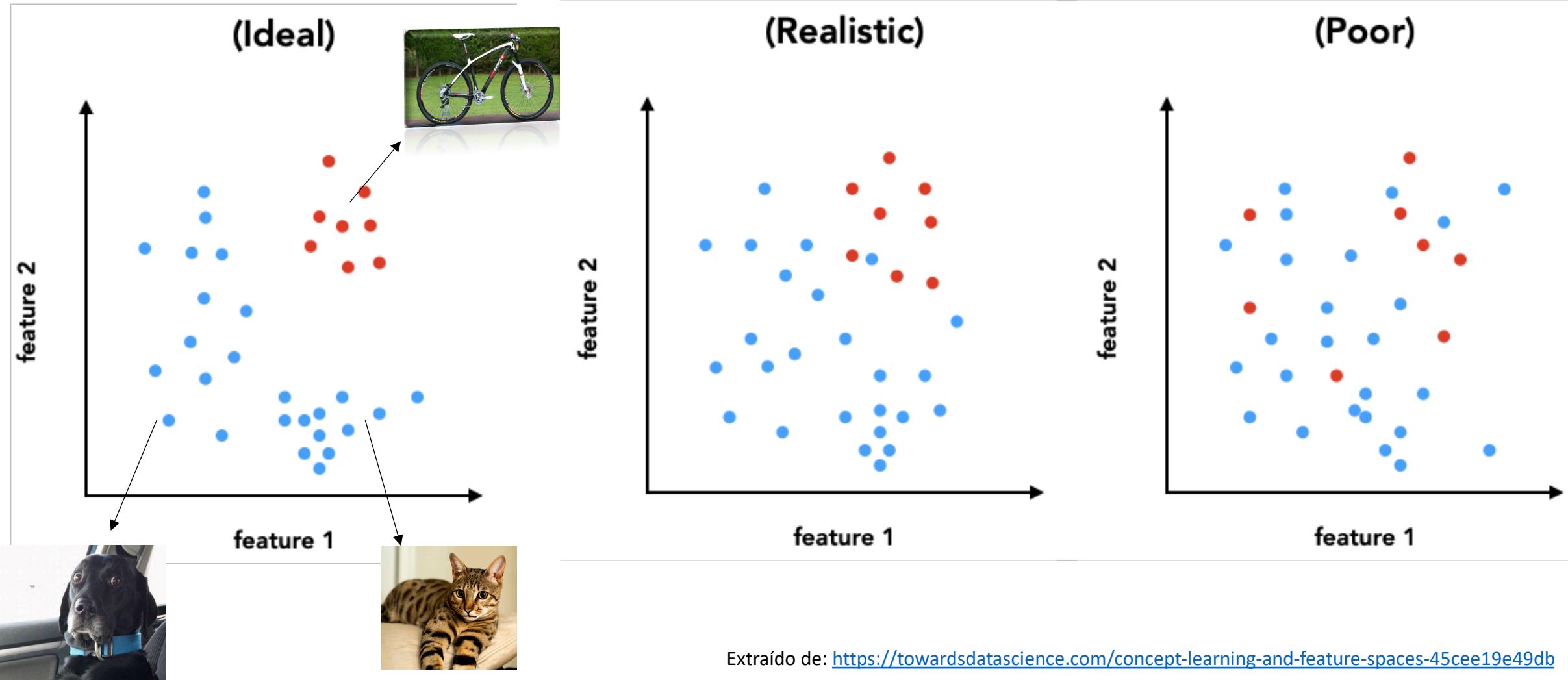
Perro salchicha!

Features

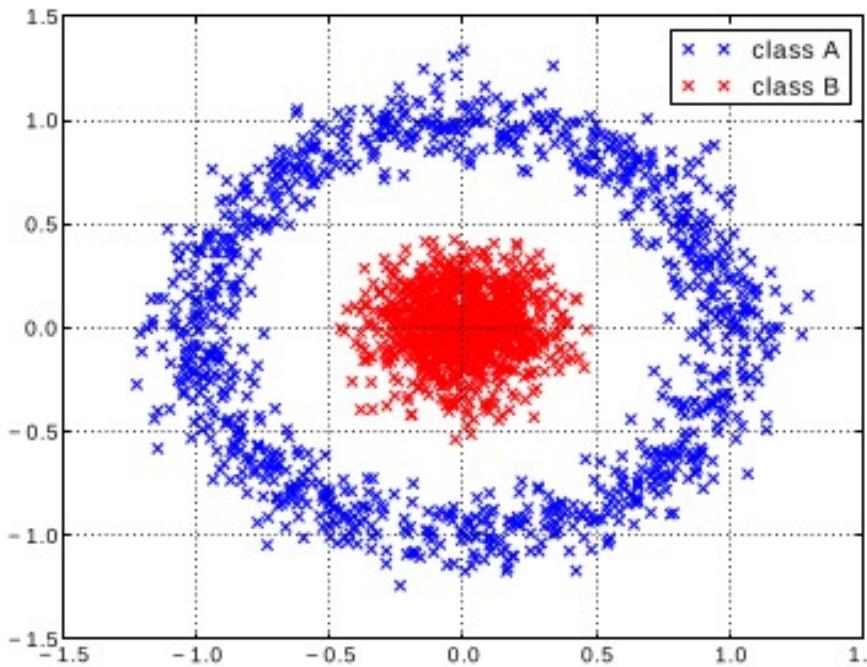
Enfoque tradicional para la clasificación de imágenes en aprendizaje maquinal supervisado



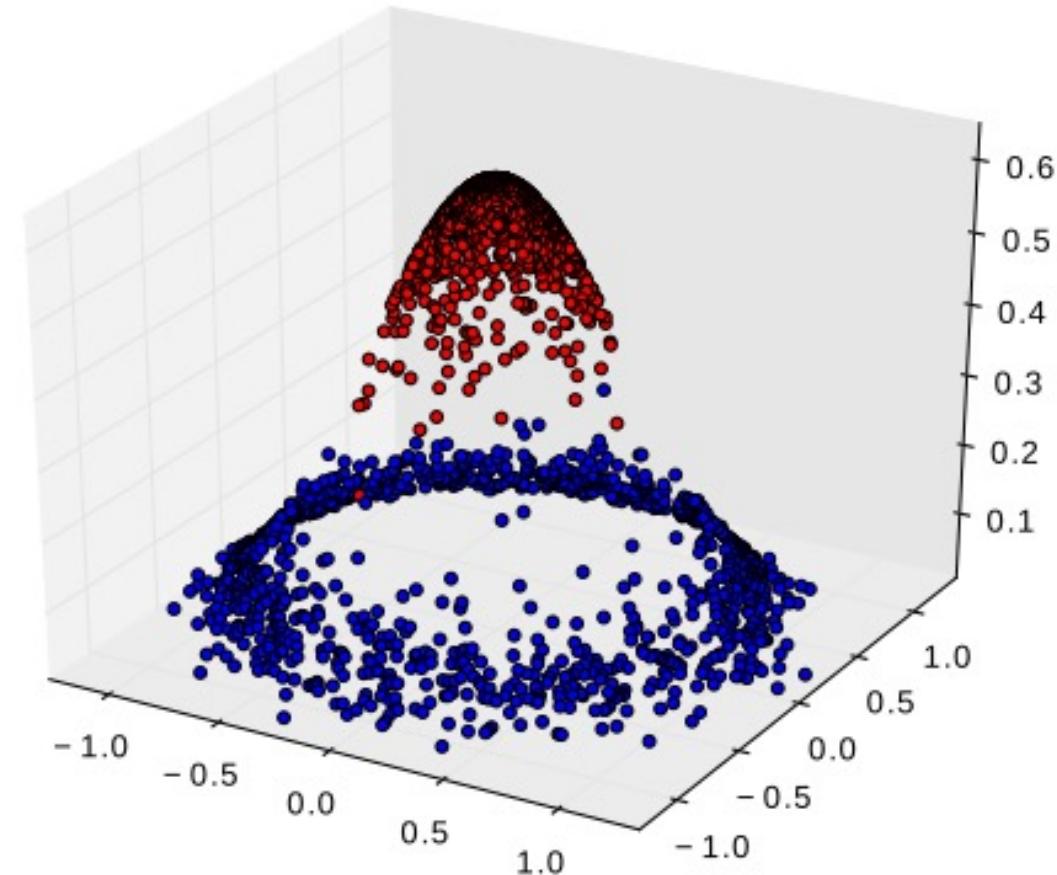
Espacio de características



Espacio de características



(a) A non linearly separable dataset.



(b) Possible feature space representation.

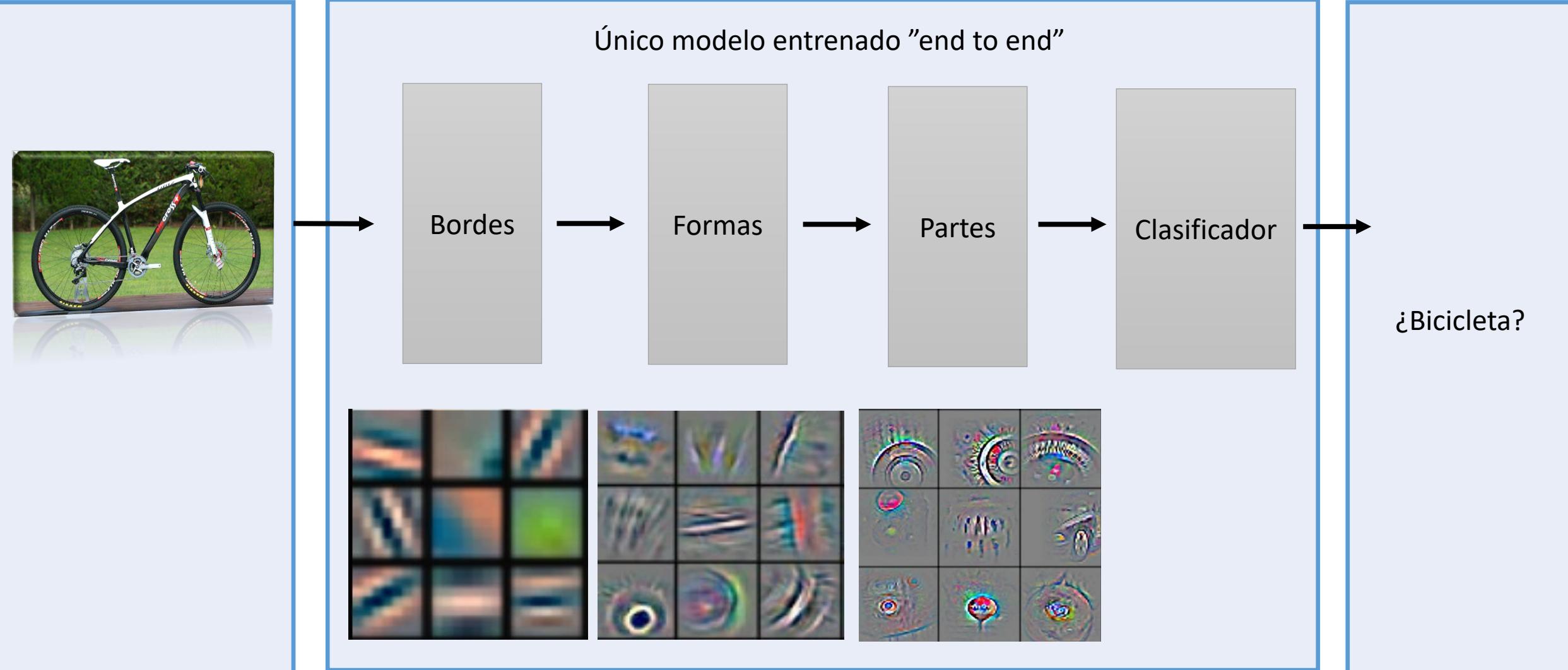
Deep Learning

Una técnica para implementar aprendizaje maquinal

Los modelos basados en deep learning son capaces de aprender **representaciones** de los datos de entrenamiento en **múltiples niveles de abstracción** (capas), componiendo módulos simples que sucesivamente transforman dichas representaciones en otras con mayor nivel de abstracción.

Deep Learning

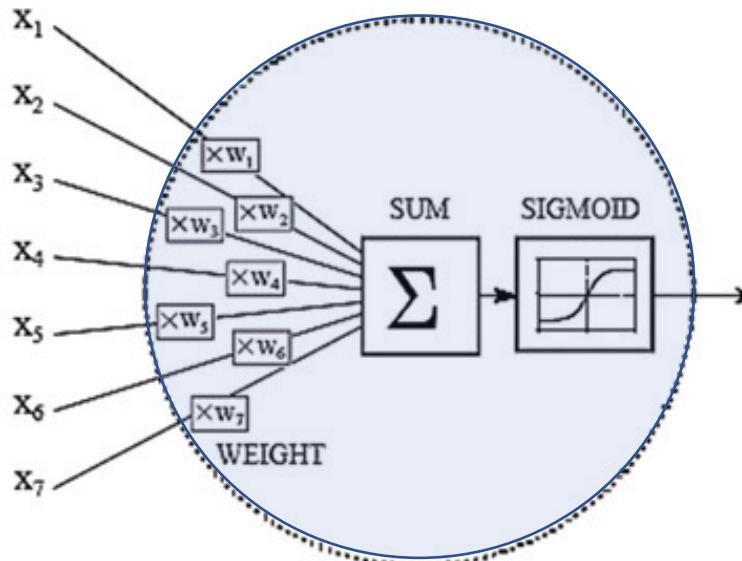
Multiples niveles de abstracción



?



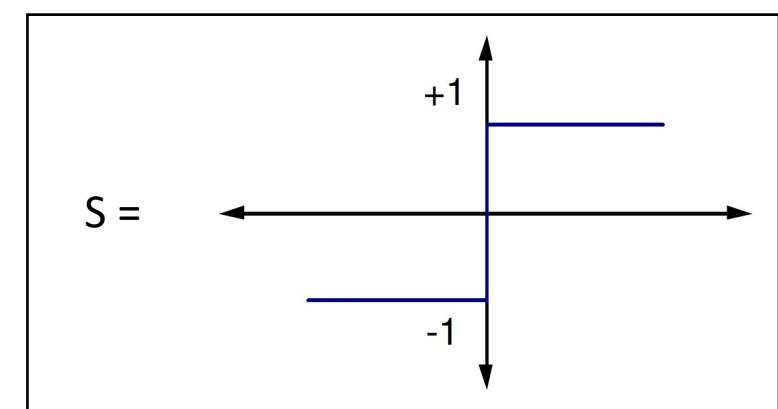
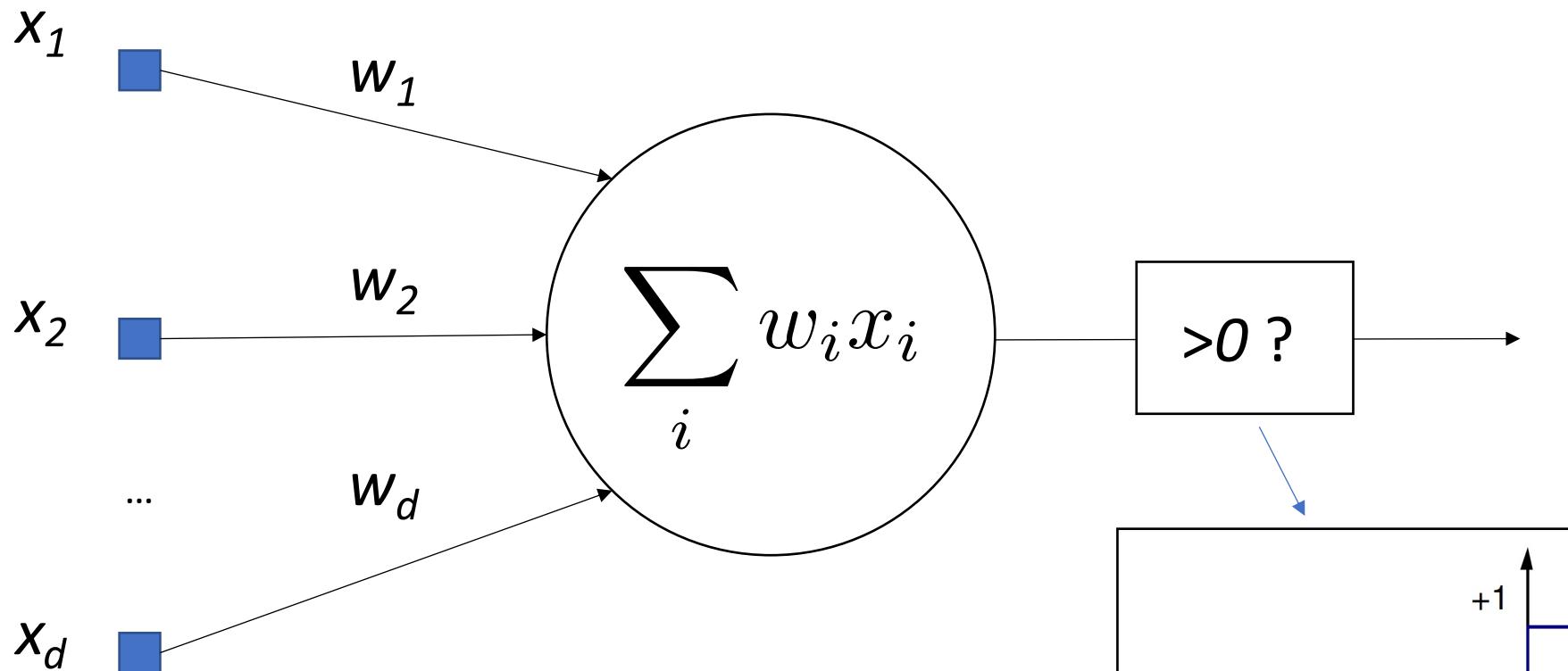
Redes Neuronales Artificiales



$$s\left(\sum x_i w_i \right)$$

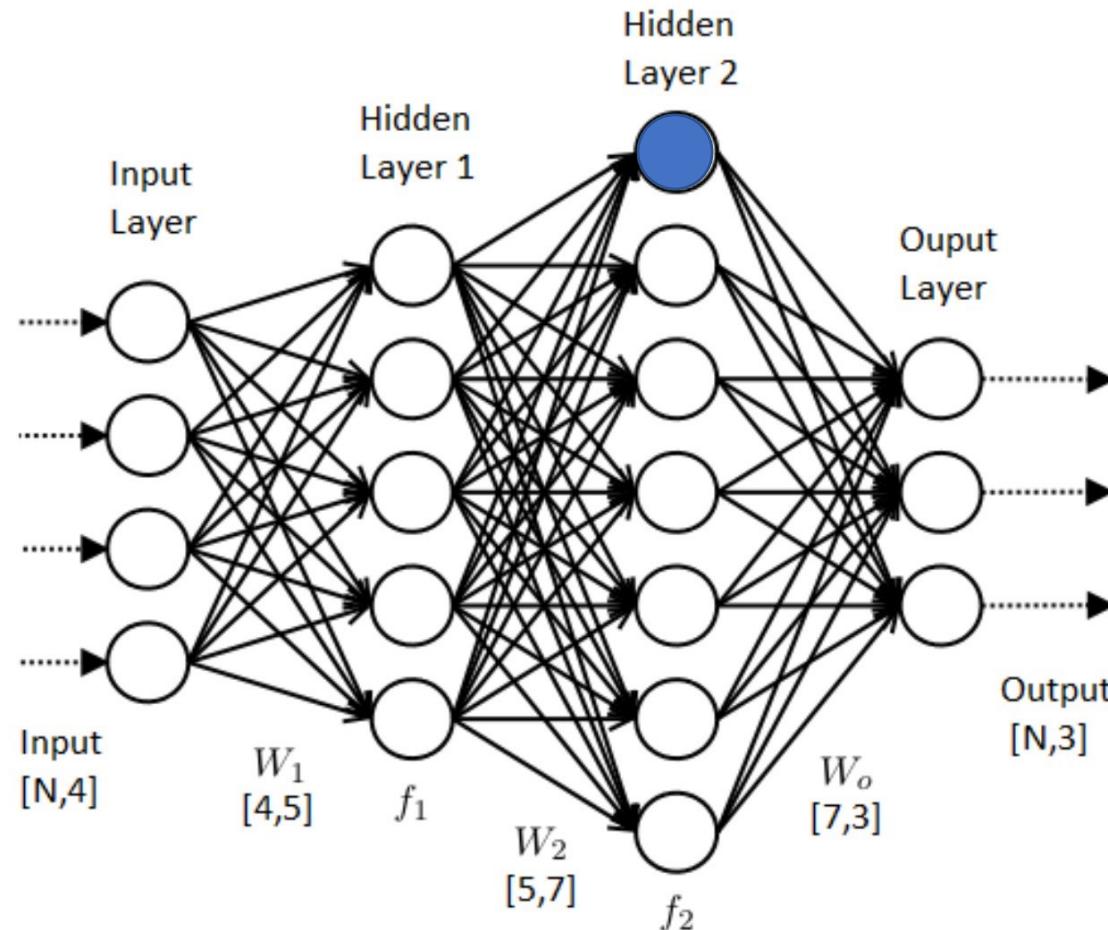
Redes Neuronales Artificiales

Perceptrón simple



Redes Neuronales Artificiales

Perceptrón Multi Capa



Turing Award Won by 3 Pioneers in Artificial Intelligence



From left, Yann LeCun, Geoffrey Hinton and Yoshua Bengio. The researchers worked on key developments for neural networks, which are reshaping how computer systems are built.

From left, Facebook, via Associated Press; Aaron Vincent Elkaim for The New York Times; Chad Buchanan/Getty Images

By Cade Metz

March 27, 2019



Redes neuronales artificiales

Parámetros a aprender

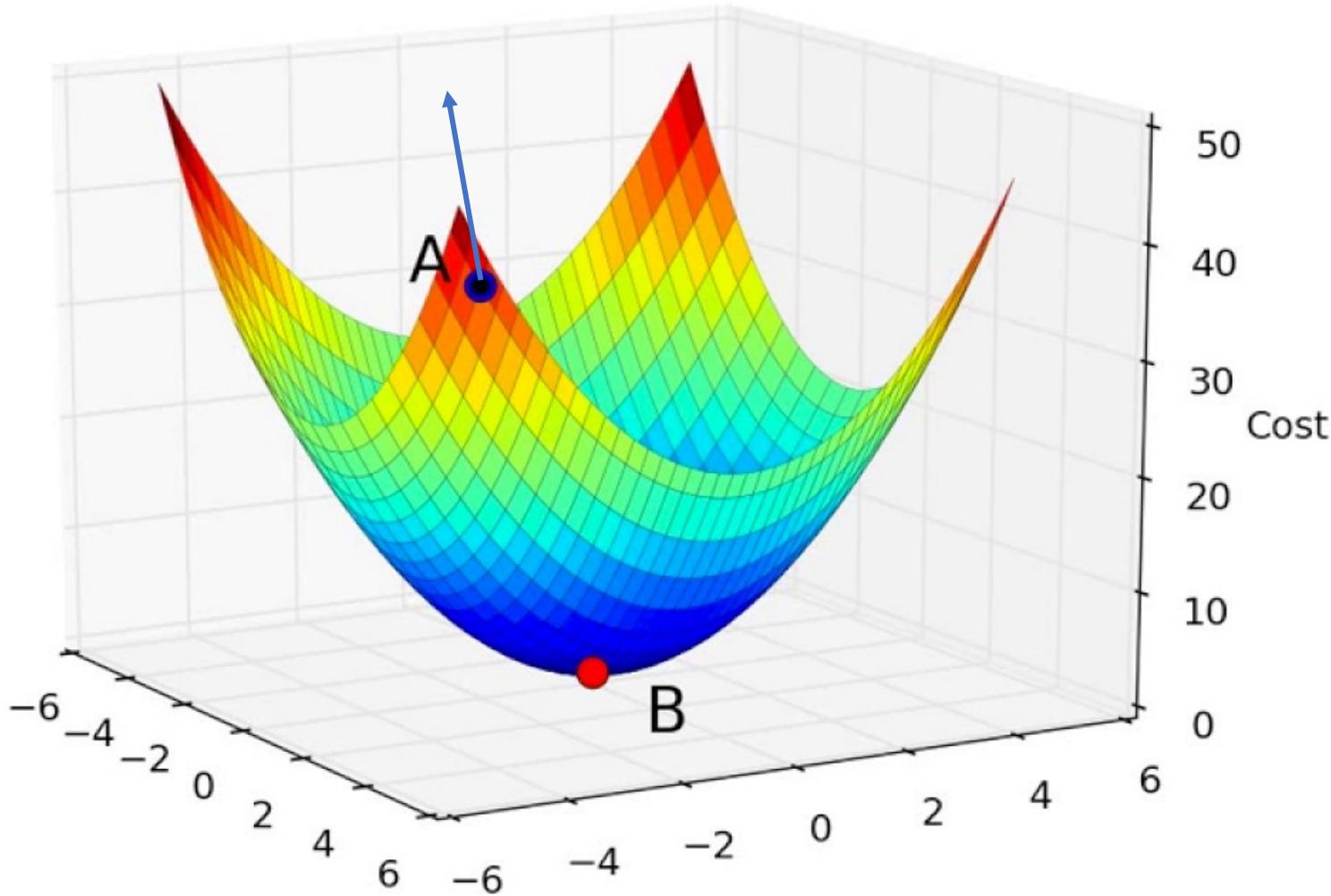
Red neuronal artificial $\rightarrow y = f(x, w)$ \longrightarrow Ej: $f(x, w) = s(\sum x_i w_i)$

Función de pérdida $\rightarrow L(y, \bar{y})$ \longrightarrow Ej: $L(y, \bar{y}) = |y - \bar{y}|^2$

Etiqueta o Ground Truth

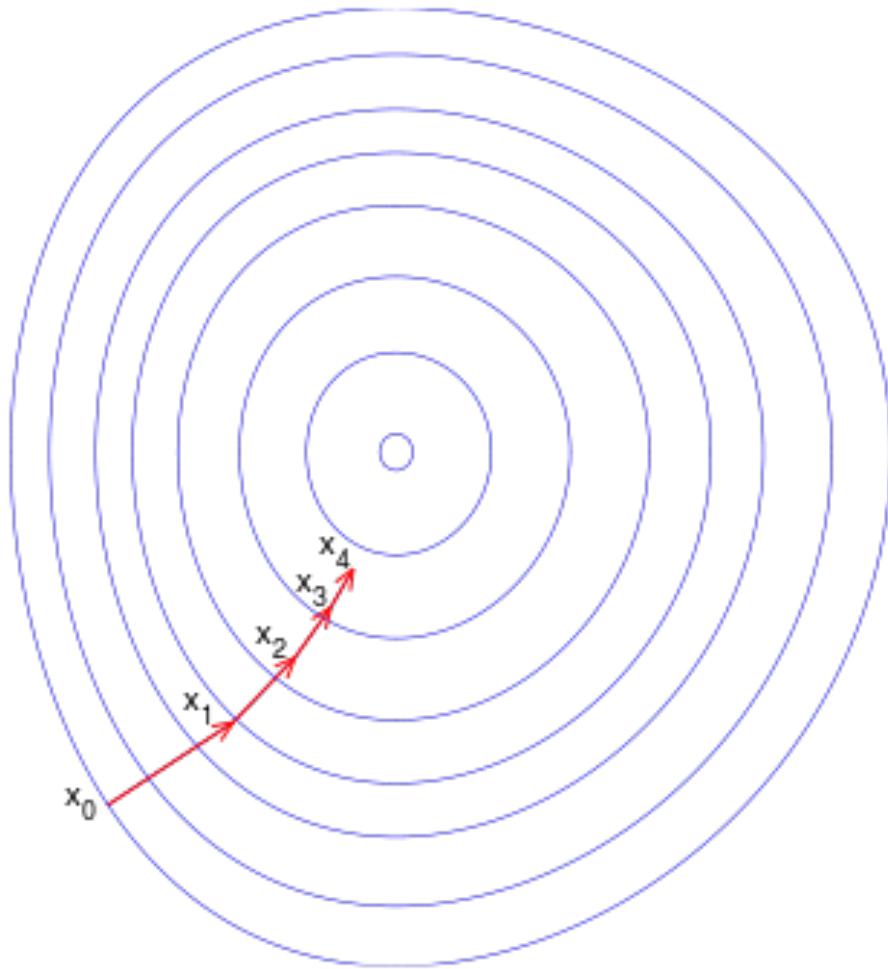
Cómo aprendemos w ? \rightarrow Gradiente descendiente

Gradiente descendente



$$\nabla_{\mathbf{w}} \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial w_1}, \dots, \frac{\partial \mathcal{L}}{\partial w_D} \right)$$

Gradiente descendente



Cómo calcular el gradiente?

- **1. Derivación analítica:** derivar a mano y escribir el código
- **2. Derivación numérica :** diferencias finitas
- **3. Derivación simbólica:** se realiza utilizando las reglas estudiadas en Análisis matemático pero automatizadas (ej: Maple, Mathematica) Backpropagation
- **4. Derivación automática:**
 - Se definen las derivadas para las operaciones 'primitivas' (matemáticas y de control)
 - Se construye un grafo de operaciones y se deriva siguiendo la regla de la cadena.

Frameworks que implementan diferenciación automática



TensorFlow

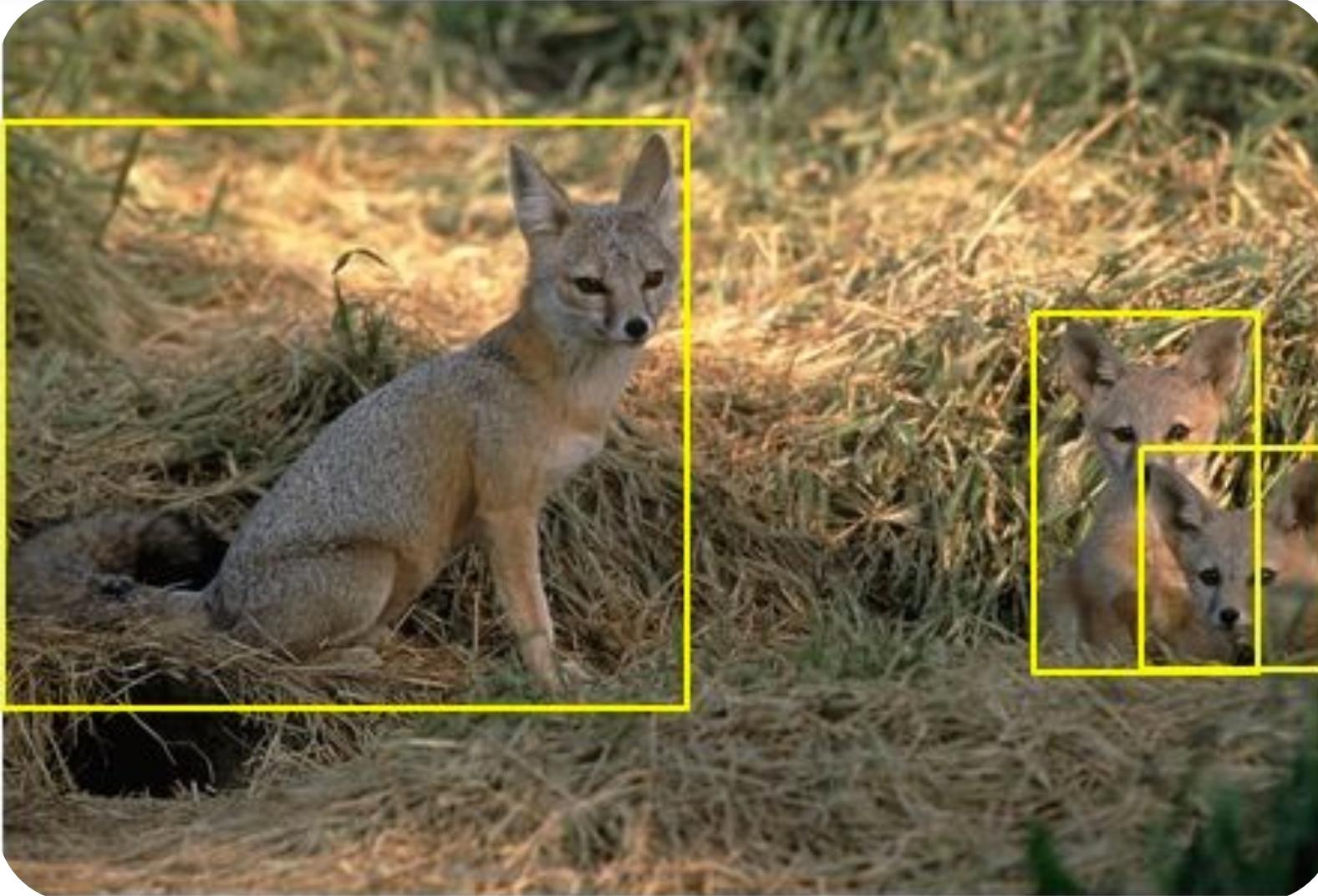
P Y Torch R C H



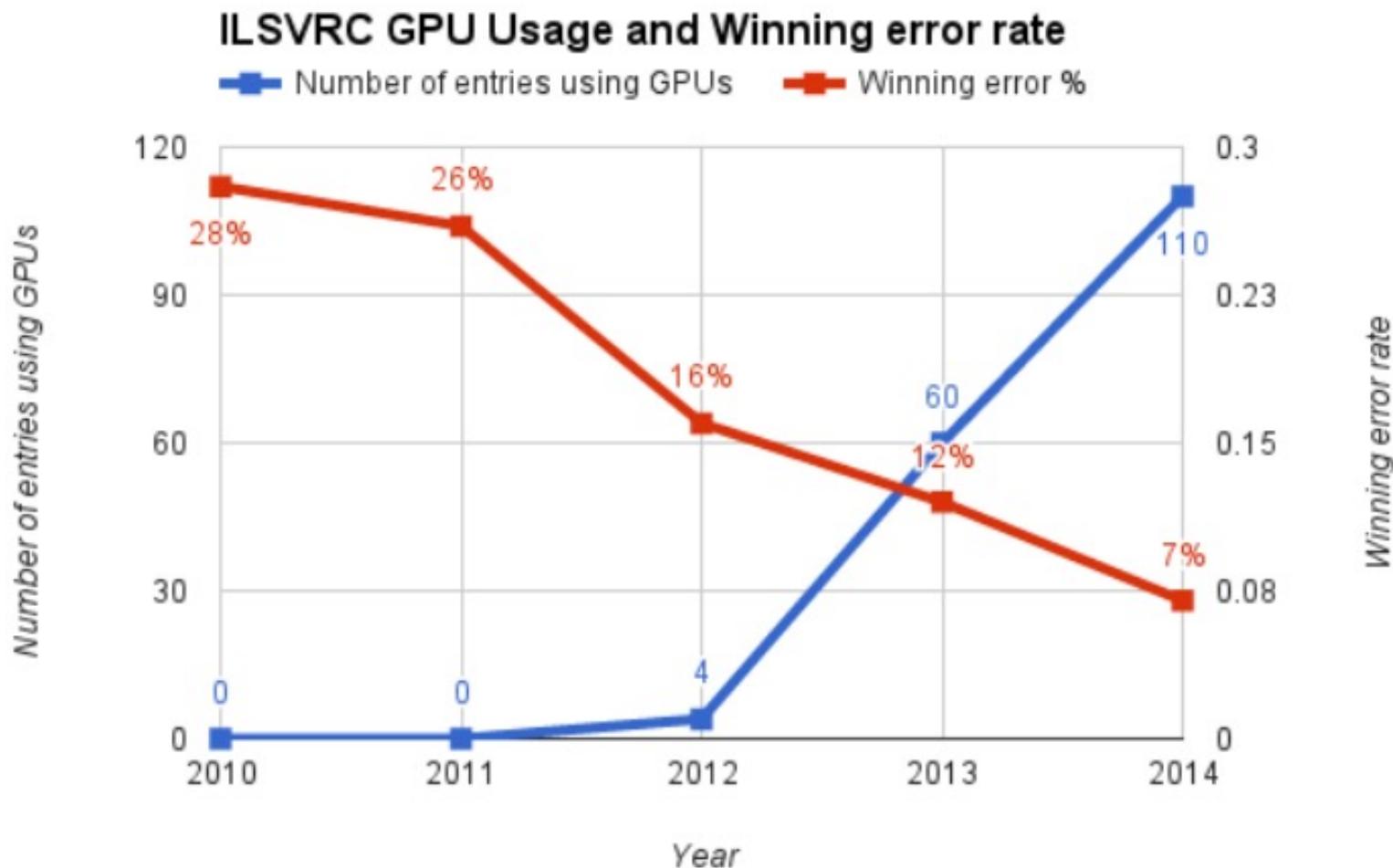
ImageNet

15.000.000 millones de imágenes, 1000 categorías

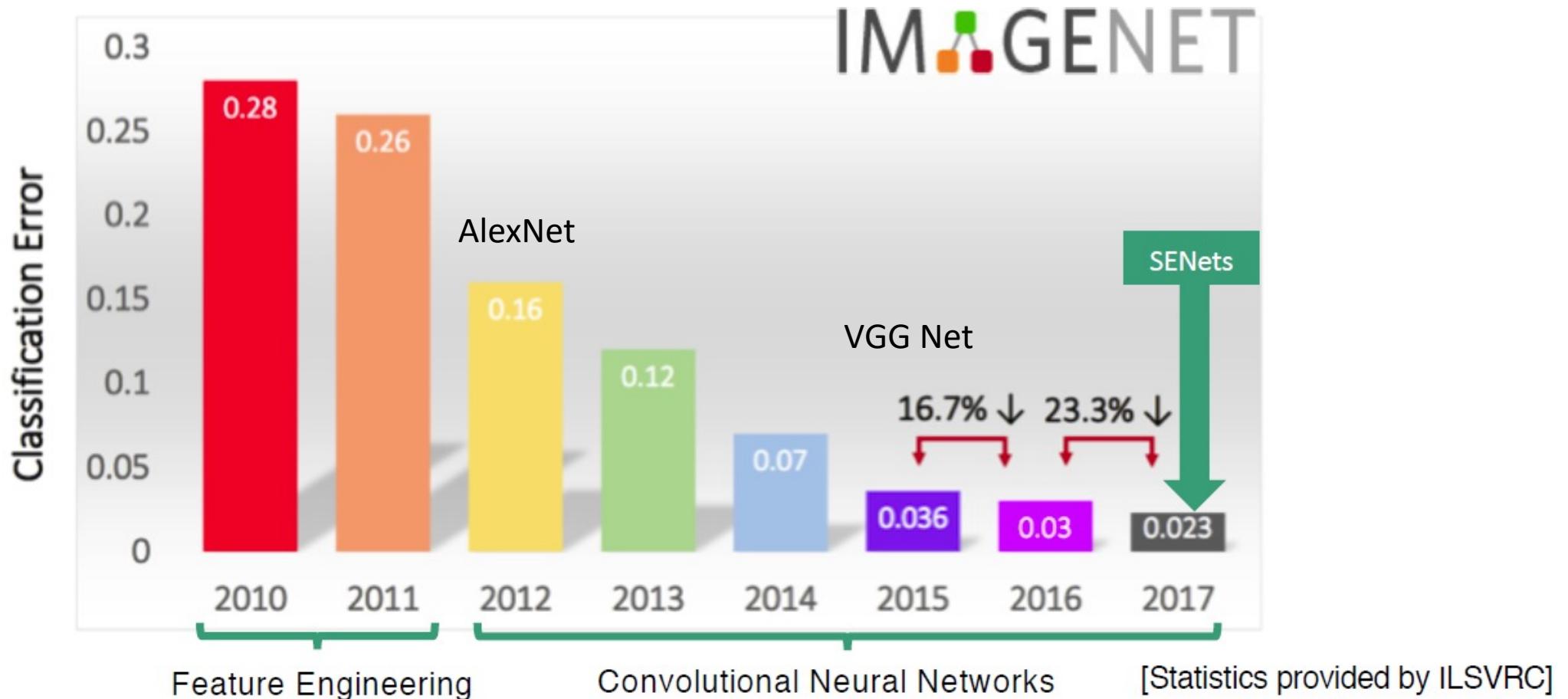
ImageNet Challenge



ImageNet Challenge



ImageNet

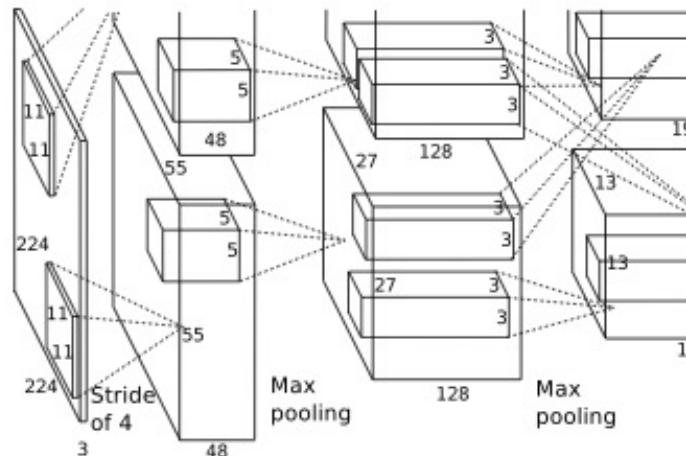


Computer Vision Revolution

Más datos



Mejores modelos



GPUs más poderosas



ImageNet Challenge



mite

container ship

motor scooter

leopard

mite	container ship	motor scooter	leopard
black widow	lifeboat	go-kart	jaguar
cockroach	amphibian	moped	cheetah
tick	fireboat	bumper car	snow leopard
starfish	drilling platform	golfcart	Egyptian cat



grille



mushroom



cherry



Madagascar cat

convertible	agaric	dalmatian	squirrel monkey
grille	mushroom	grape	spider monkey
pickup	jelly fungus	elderberry	titi
beach wagon	gill fungus	ffordshire bullterrier	indri
fire engine	dead-man's-fingers	currant	howler monkey

Algunas aplicaciones

AlphaGO

AlphaGO

ARTICLE

Mastering the game of Go with deep neural networks and tree search

David Silver^{1*}, Aja Huang^{1*}, Chris J. Maddison¹, Arthur Guez¹, Laurent Sifre¹, George van den Driessche¹, Julian Schrittwieser¹, Ioannis Antonoglou¹, Veda Panneershelvam¹, Marc Lanctot¹, Sander Dieleman¹, Dominik Grewe¹, John Nham², Nal Kalchbrenner¹, Ilya Sutskever², Timothy Lillicrap¹, Madeleine Leach¹, Koray Kavukcuoglu¹, Thore Graepel¹ & Demis Hassabis¹

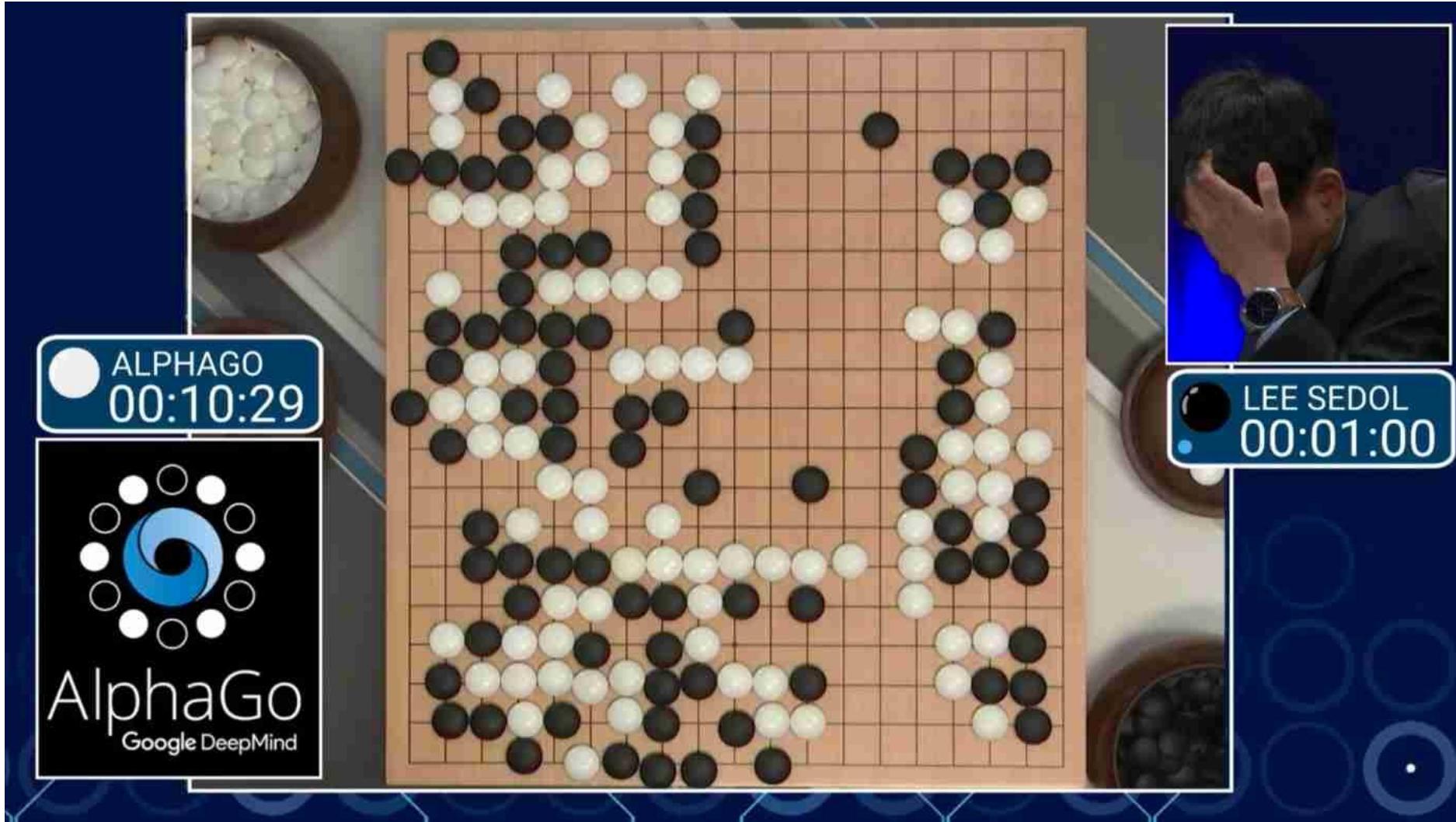
doi:10.1038/nature16961

The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. Here we introduce a new approach to computer Go that uses ‘value networks’ to evaluate board positions and moves. These deep neural networks are trained by a novel combination of supervised learning and ‘policy networks’ to select moves. These deep learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play. We also introduce a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, our program AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.

Recently, deep convolutional neural networks have achieved unprecedented performance in visual domains: for example, image classification¹⁸ and playing Atari games¹⁹. They use many convolutional layers to learn features such as edges and shapes²⁰. We

¹ Google DeepMind, London, UK. ² University College London, London, UK. *These authors contributed equally to this work. Correspondence should be addressed to D.H. (e-mail: demis.hassabis@google.com). DOI: 10.1038/nature16961

AlphaGo



AlphaGo

- Entrenamiento inicial con 30 millones de tableros jugados por humanos expertos
- Luego refinado con entrenamiento basado en Aprendizaje por Refuerzo



AlphaGO Zero

AlphaGo Zero

ARTICLE

doi:10.1038/nature24270

Mastering the game of Go without human knowledge

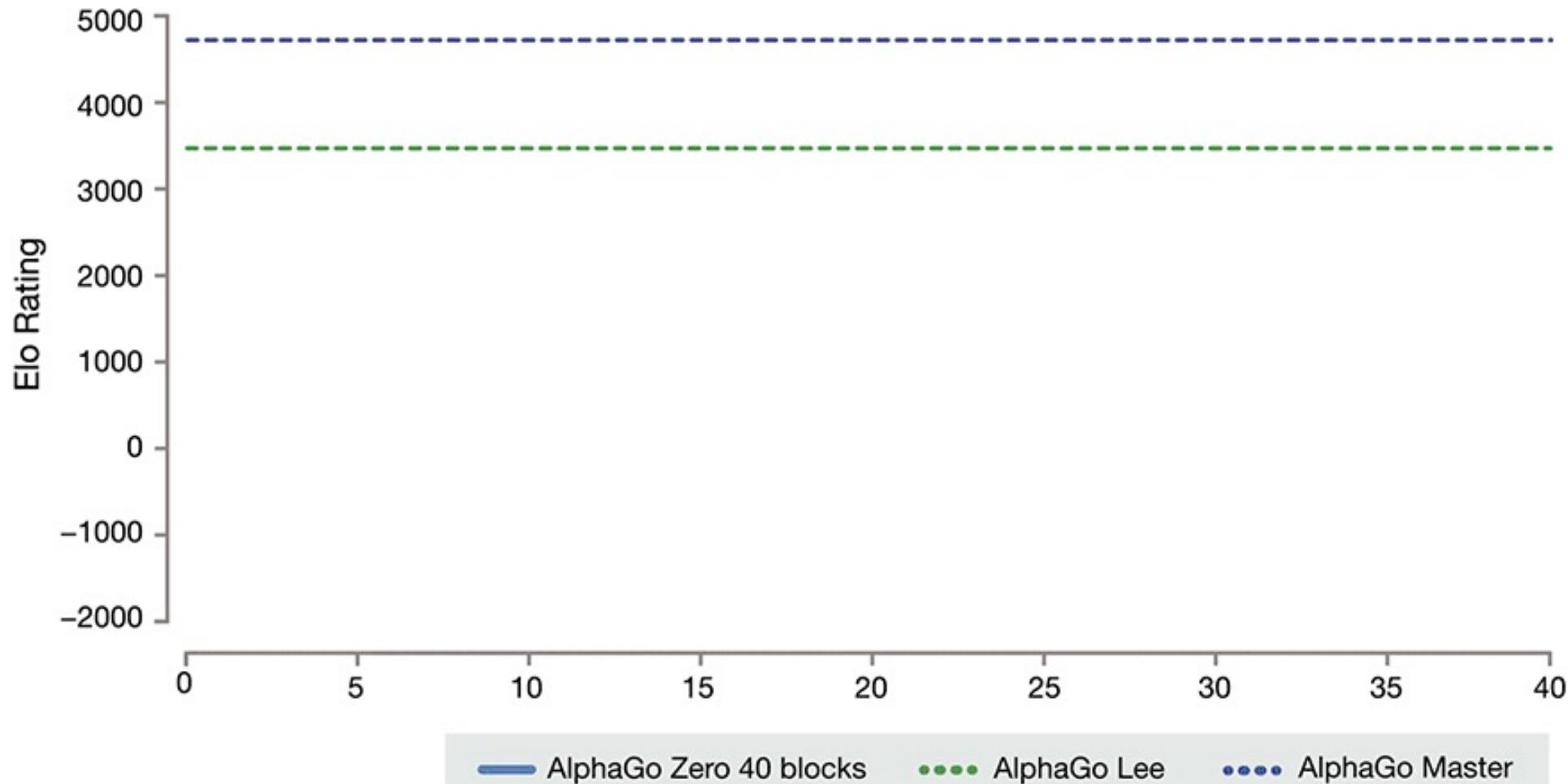
David Silver^{1*}, Julian Schrittwieser^{1*}, Karen Simonyan^{1*}, Ioannis Antonoglou¹, Aja Huang¹, Arthur Guez¹, Thomas Hubert¹, Lucas Baker¹, Matthew Lai¹, Adrian Bolton¹, Yutian Chen¹, Timothy Lillicrap¹, Fan Hui¹, Laurent Sifre¹, George van den Driessche¹, Thore Graepel¹ & Demis Hassabis¹

A long-standing goal of artificial intelligence is an algorithm that learns, *tabula rasa*, superhuman proficiency in challenging domains. Recently, AlphaGo became the first program to defeat a world champion in the game of Go. The tree search in AlphaGo evaluated positions and selected moves using deep neural networks. These neural networks were trained by supervised learning from human expert moves, and by reinforcement learning from self-play. Here we introduce an algorithm based solely on reinforcement learning, without human data, guidance or domain knowledge beyond game rules. AlphaGo becomes its own teacher: a neural network is trained to predict AlphaGo's own move selections and also the winner of AlphaGo's games. This neural network improves the strength of the tree search, resulting in higher quality move selection and stronger self-play in the next iteration. Starting *tabula rasa*, our new program AlphaGo Zero achieved superhuman performance, winning 100–0 against the previously published, champion-defeating AlphaGo.

Much progress towards artificial intelligence has been made using supervised learning systems that are trained to replicate the decisions of human experts^{1–4}. However, expert data sets are often expensive, unreliable or simply unavailable. Even when reliable, data sets trained in one domain may impose a cognitive bias onto the system.

trained solely by self-play reinforcement learning, starting from scratch, can learn to play Go at superhuman level. Our approach uses only the basic rules of the game, and no human data, domain knowledge or supervision.

AlphaGo Zero



2015: Deep Reinforcement Learning en Atari

Deep Reinforcement Learning

LETTER

Human-level control through deep reinforcement learning

doi:10.1038/nature14236

Volodymyr Mnih^{1*}, Koray Kavukcuoglu^{1*}, David Silver^{1*}, Andrei A. Rusu¹, Joel Veness¹, Marc G. Bellemare¹, Alex Graves¹, Martin Riedmiller¹, Andreas K. Fidjeland¹, Georg Ostrovski¹, Stig Petersen¹, Charles Beattie¹, Amir Sadik¹, Ioannis Antonoglou¹, Helen King¹, Dharshan Kumaran¹, Daan Wierstra¹, Shane Legg¹ & Demis Hassabis¹

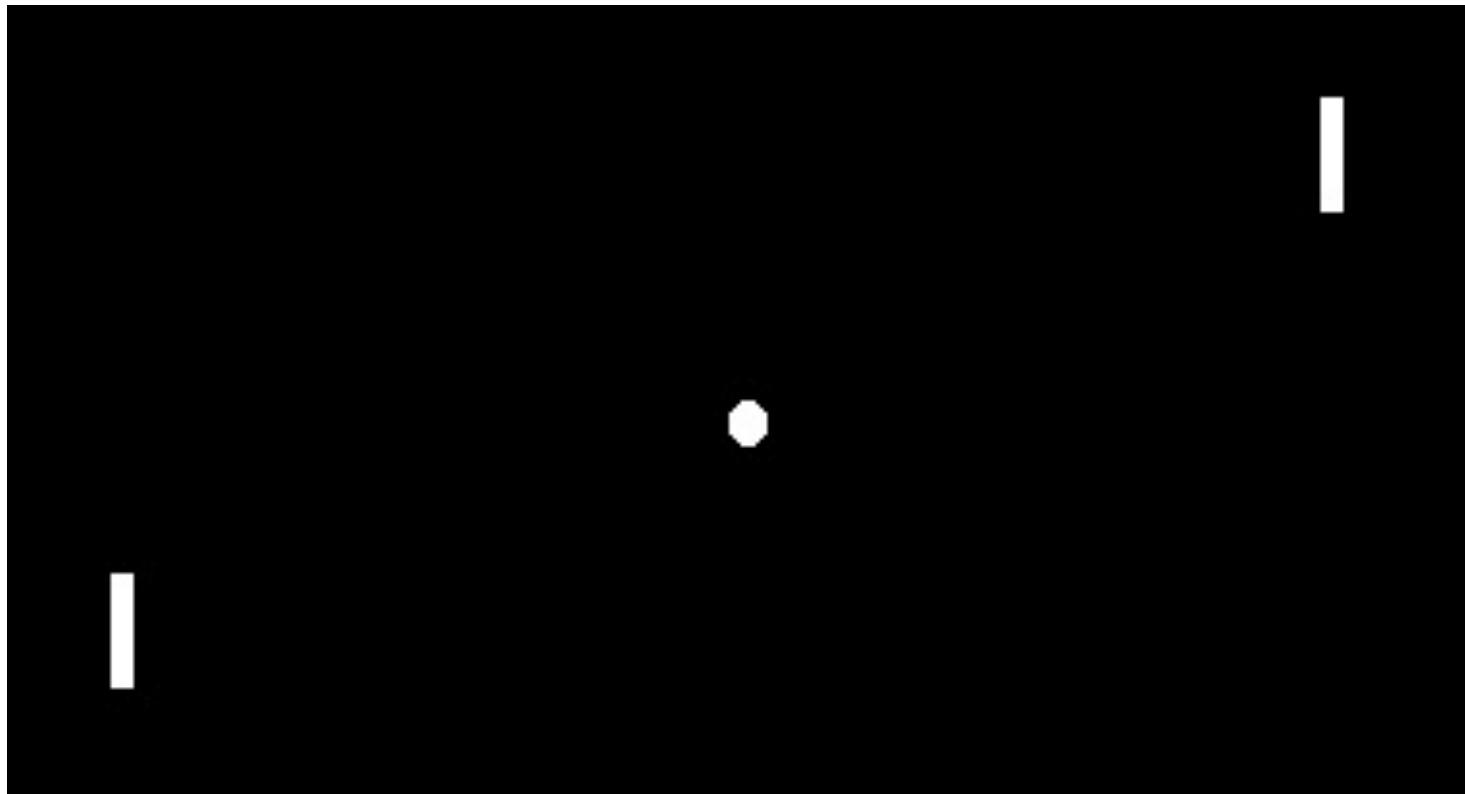
The theory of reinforcement learning provides a normative account¹, deeply rooted in psychological² and neuroscientific³ perspectives on animal behaviour, of how agents may optimize their control of an environment. To use reinforcement learning successfully in situations approaching real-world complexity, however, agents are confronted with a difficult task: they must derive efficient representations of the environment from high-dimensional sensory inputs, and use these to generalize past experience to new situations. Remarkably, humans and other animals seem to solve this problem through a combination of reinforcement learning^{4,5} and other cognitive processes, revealing

agent is to select actions in a fashion that maximizes cumulative future reward. More formally, we use a deep convolutional neural network to approximate the optimal action-value function

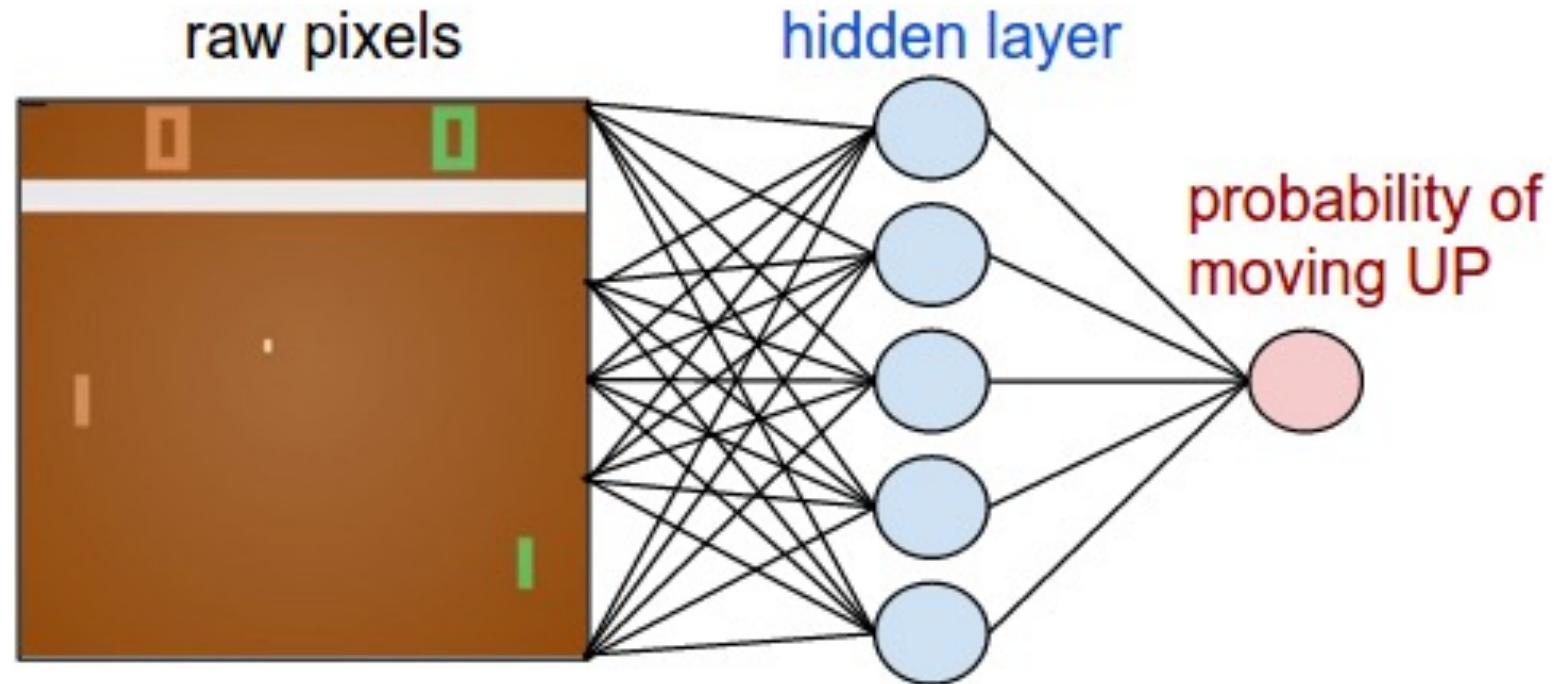
$$Q^*(s,a) = \max_{\pi} \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots | s_t = s, a_t = a]$$

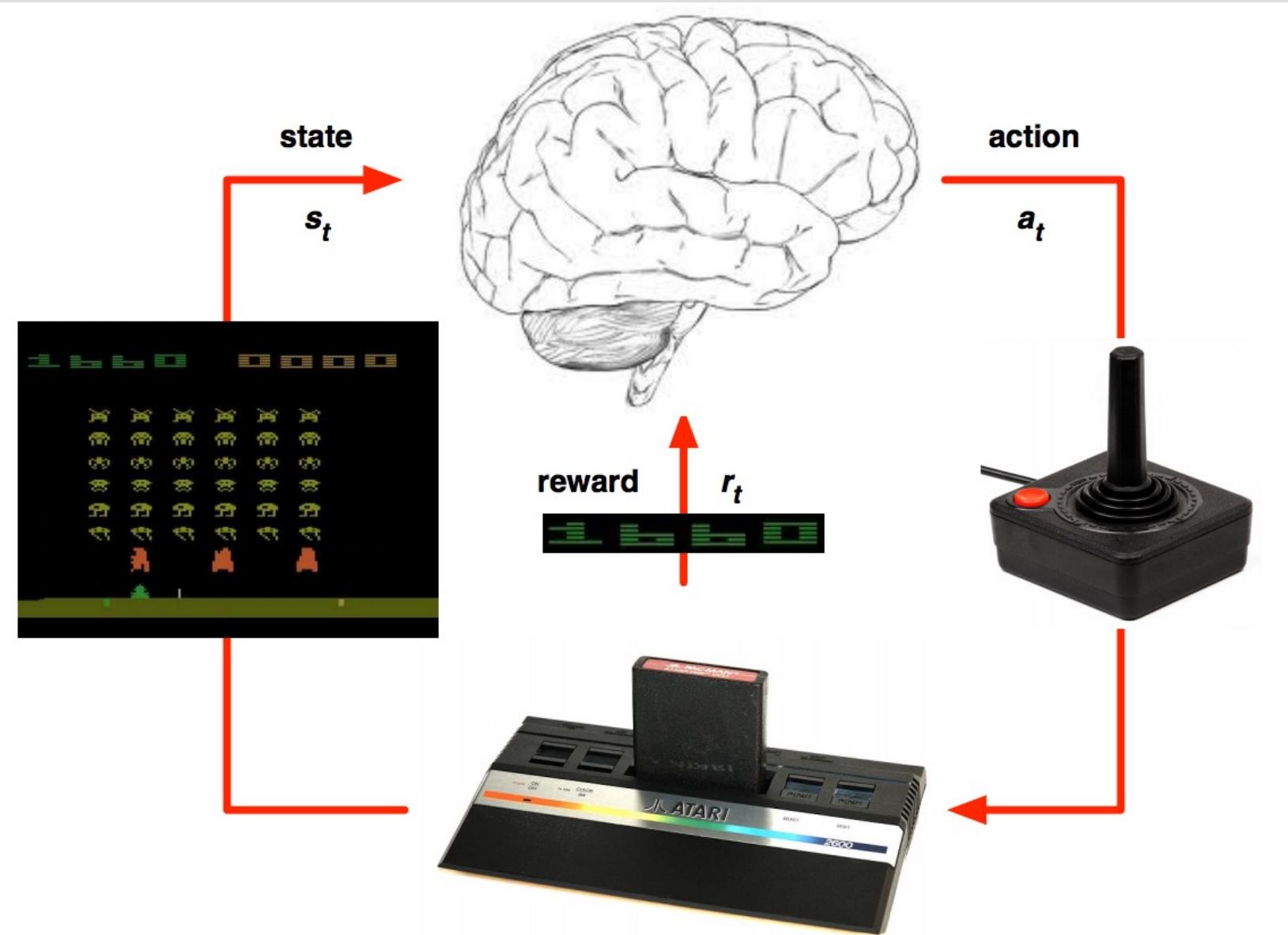
which is the maximum sum

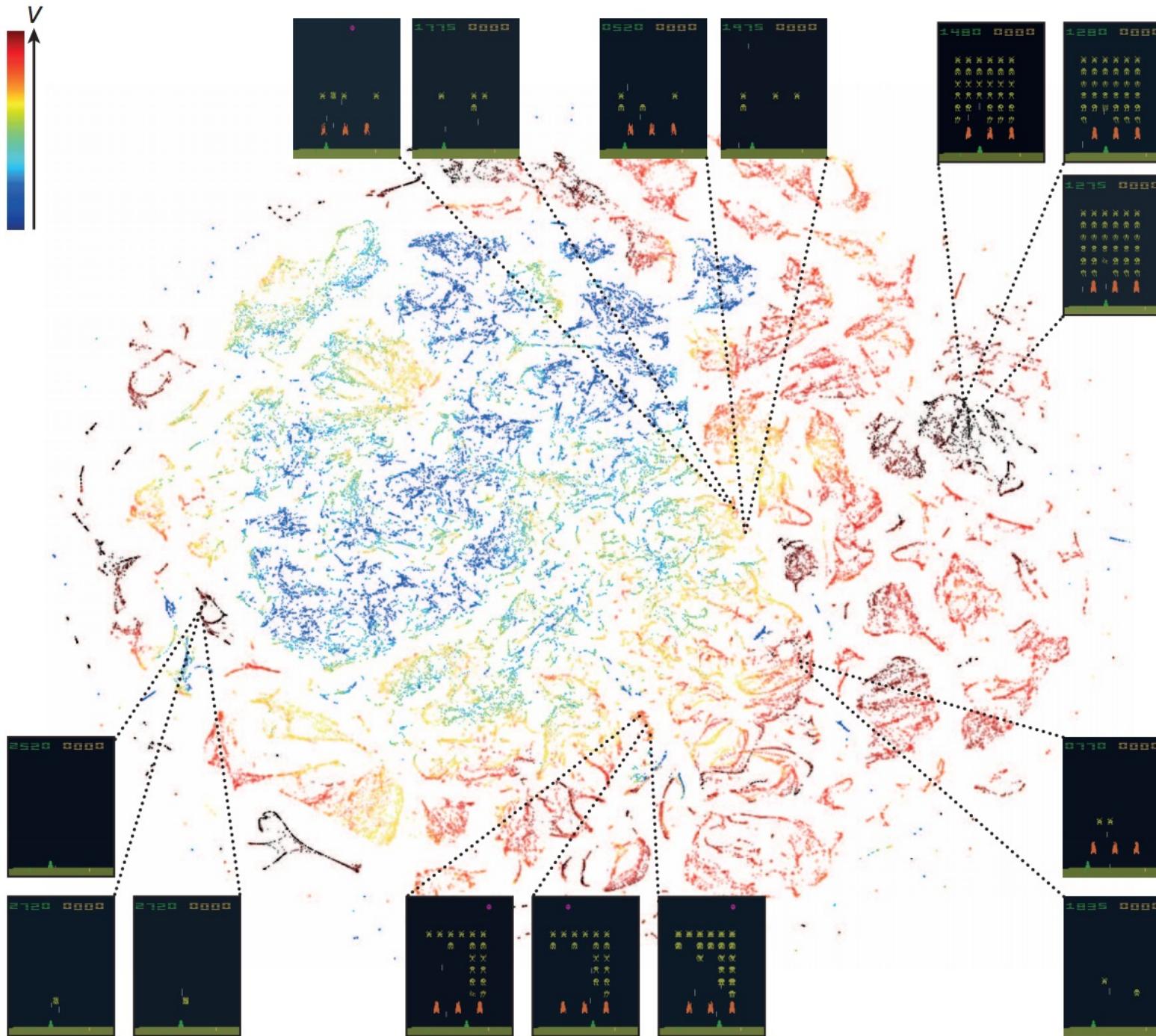
Arcades



Arcades







Arcades

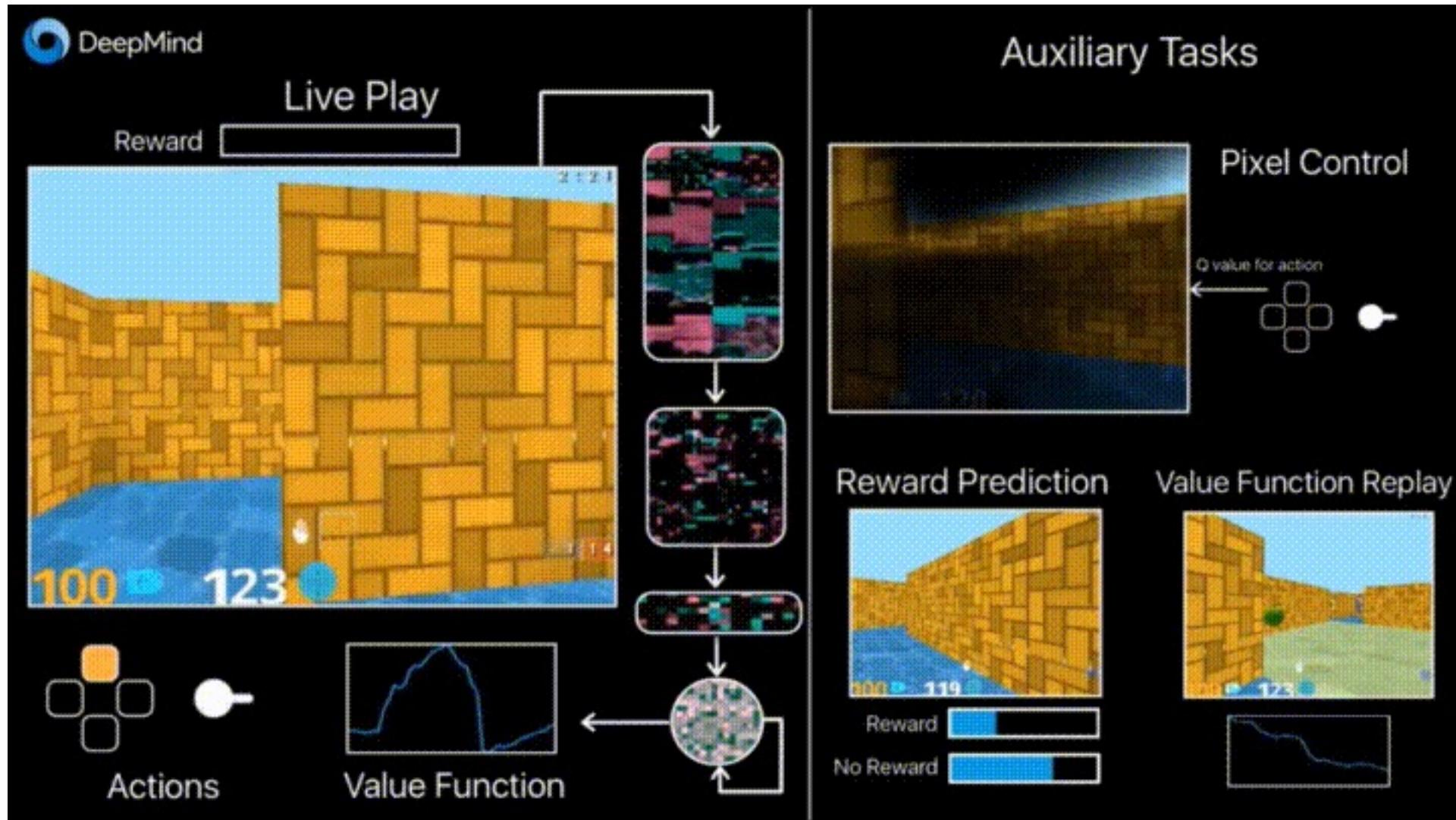
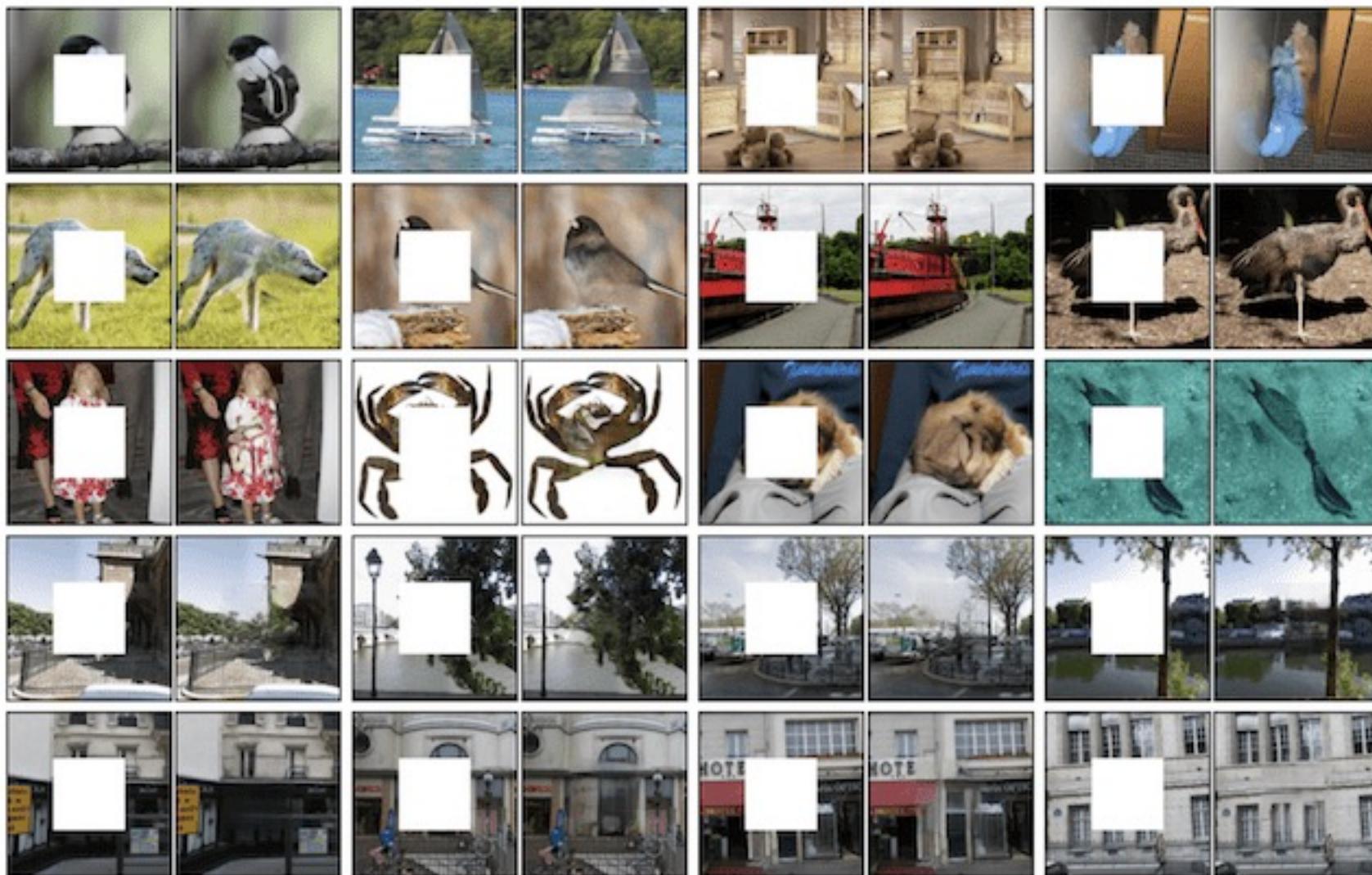


Photo Inpainting

Photo Inpainting

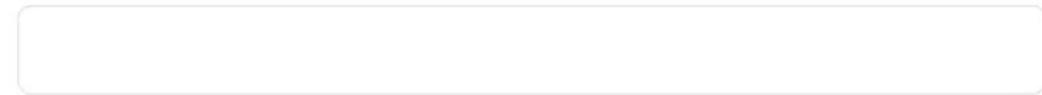


Modelos de lenguaje

Generación de código asistida

Describe a layout.

Just describe any layout you want, and it'll try to render below!

Generate

Generación de código asistida



Chatbots

ChatGPT

Examples	Capabilities	Limitations
"Explain quantum computing in simple terms"	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?"	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?"	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

Free Research Preview: ChatGPT is optimized for dialogue. Our goal is to make AI systems more natural to interact with, and your feedback will help us improve our systems and make them even more useful.

<https://chat.openai.com/>

Chatbots



Escribe un texto de bienvenida para alumnos de un curso de introducción al aprendizaje profundo



Bienvenidos alumnos al curso de introducción al aprendizaje profundo. Este es un viaje emocionante hacia el descubrimiento de los algoritmos de aprendizaje automático y su capacidad para transformar la forma en que se realiza la toma de decisiones y se resuelven problemas complejos. Juntos, exploraremos los conceptos clave, las técnicas y las herramientas necesarias para aplicar el aprendizaje profundo en una variedad de situaciones y industrias.



Este curso está diseñado para proporcionar una base sólida en el aprendizaje profundo y preparar a los alumnos para seguir adelante en su carrera en la inteligencia artificial y el aprendizaje automático. Estamos emocionados de trabajar juntos y ver cómo crecen y desarrollan sus habilidades a lo largo del curso.

¡Así que preparamos nuestros ordenadores y empecemos a aprender! ¡Bienvenidos al curso de introducción al aprendizaje profundo!

Predicción de apariencia futura

0-18



19-29



30-39



40-49



50-59



60+



Traducción de imágenes

Monet Photos



Monet → photo

Zebras Horses



zebra → horse

Summer Winter



summer → winter

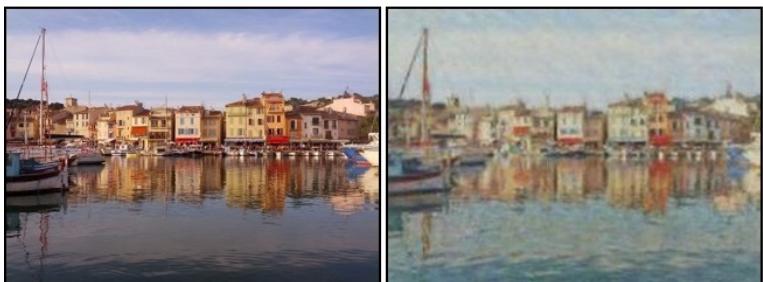


photo → Monet



horse → zebra



winter → summer



Monet



Van Gogh



Cezanne



Ukiyo-e

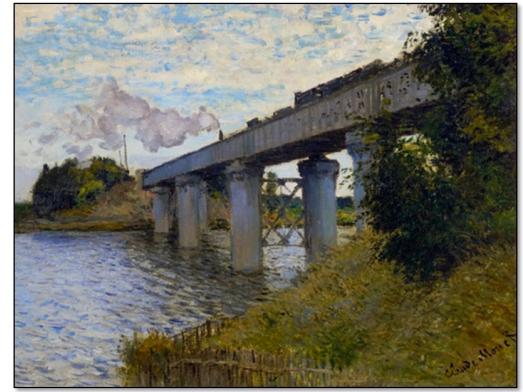
Input



Output



Input

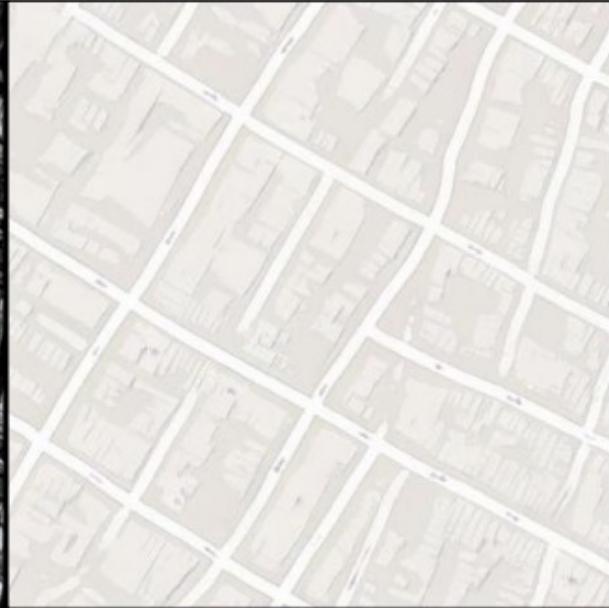
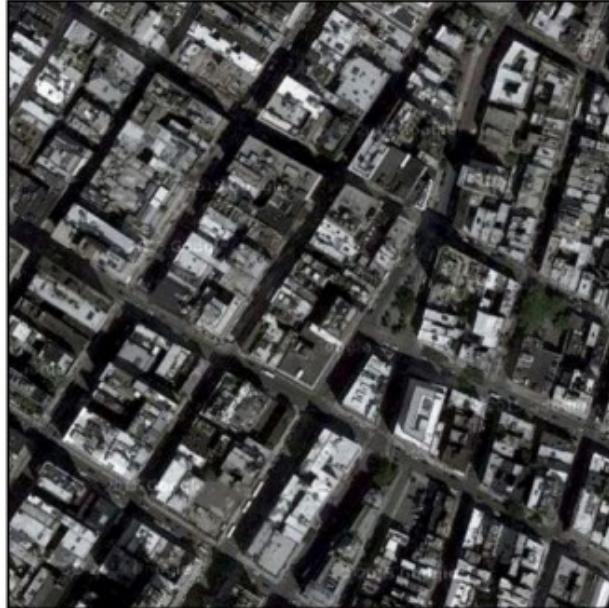
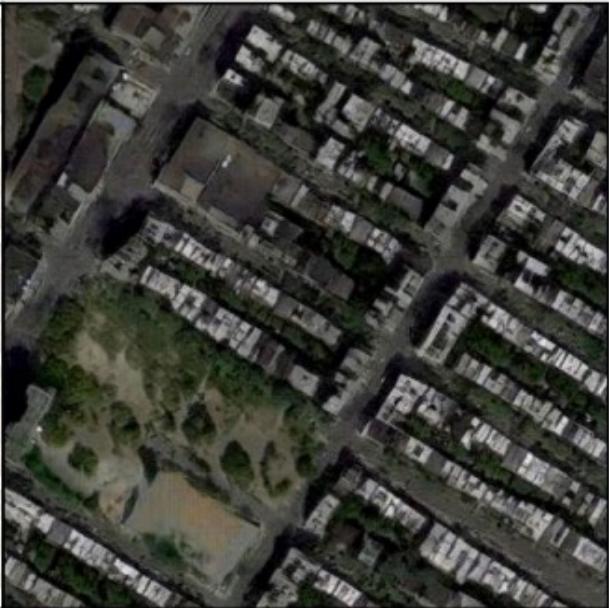
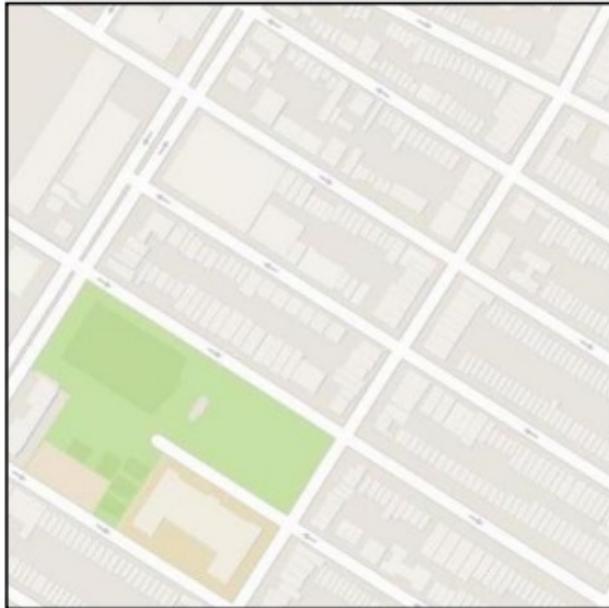


Output

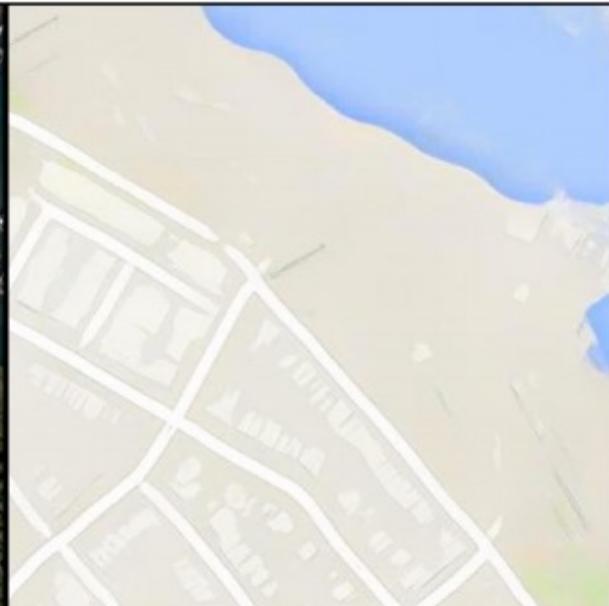
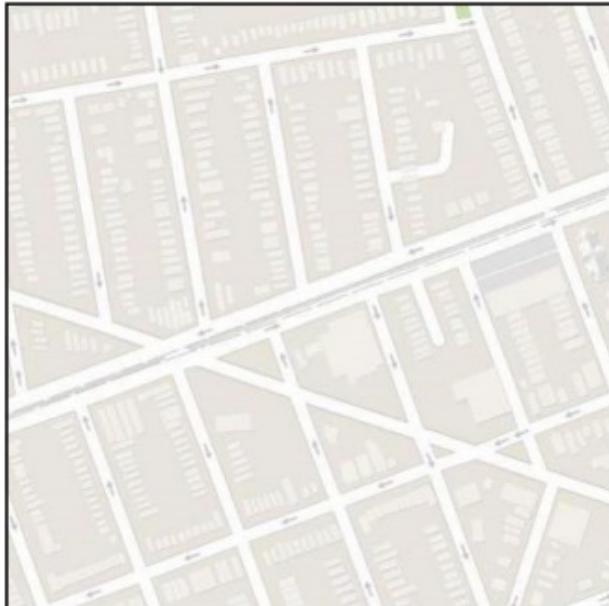




Aerial photo to map



Map to aerial photo



input

output

input

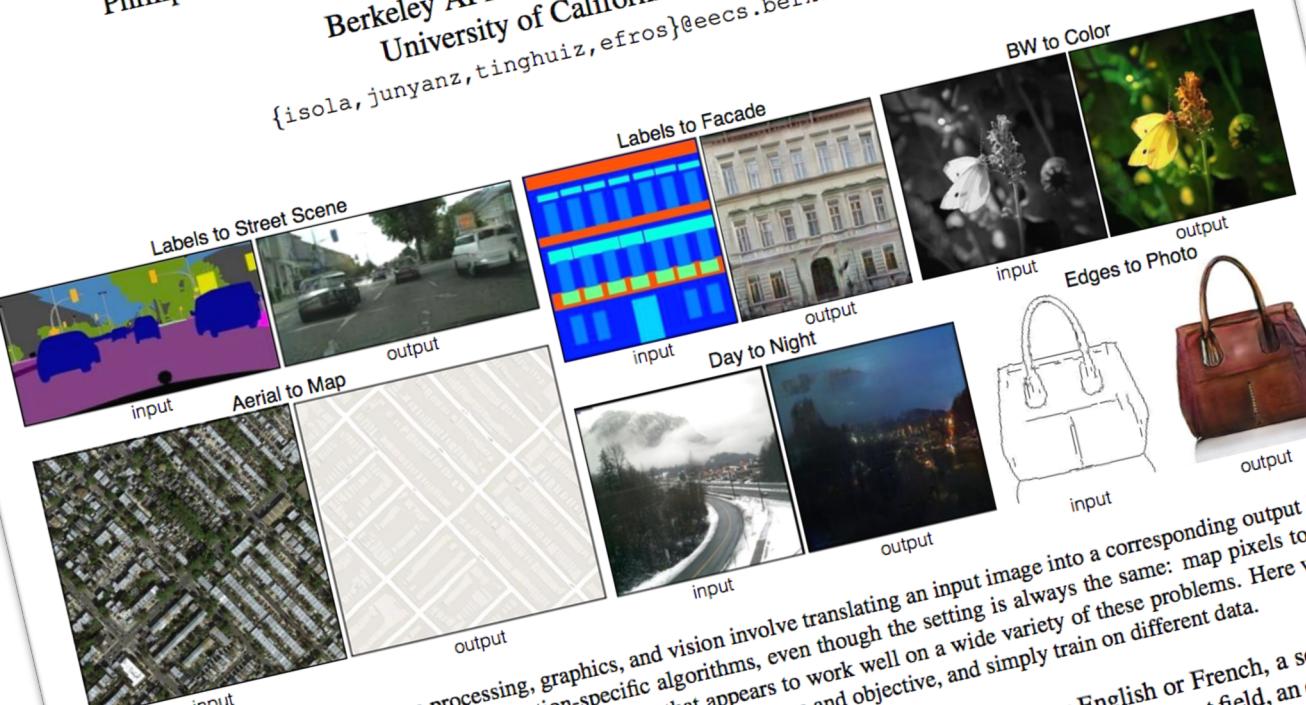
output

Traducción de imágenes con redes adversarias

Image-to-Image Translation with Conditional Adversarial Networks

Phillip Isola Jun-Yan Zhu Tinghui Zhou Alexei A. Efros

Berkeley AI Research (BAIR) Laboratory
University of California, Berkeley
`{isola, junyanz, tinghui, efros}@eecs.berkeley.edu`



Many problems in image processing, graphics, and vision involve translating an input image into a corresponding output image. These problems are often treated with application-specific algorithms, even though the setting is always the same: map pixels to pixels. We propose a general purpose solution that appears to work well on a wide variety of these problems. Here we show the same architecture and objective, and simply train on different data.

expressed in either English or French, a scene may be a grayscale image, a gradient field, an edge map, or a semantic segmentation map. This allows us to automatically learn image-to-image translation from pairs of images and their corresponding labels, without any supervision.

Generación de rostros realistas



thispersondoesnotexist.com

TEXT DESCRIPTION

An astronaut Teddy bears A bowl of soup

riding a horse lounging in a tropical resort in space playing basketball with cats in space

in a photorealistic style in the style of Andy Warhol as a pencil drawing



DALL-E 2



Dall-e

Generación de música

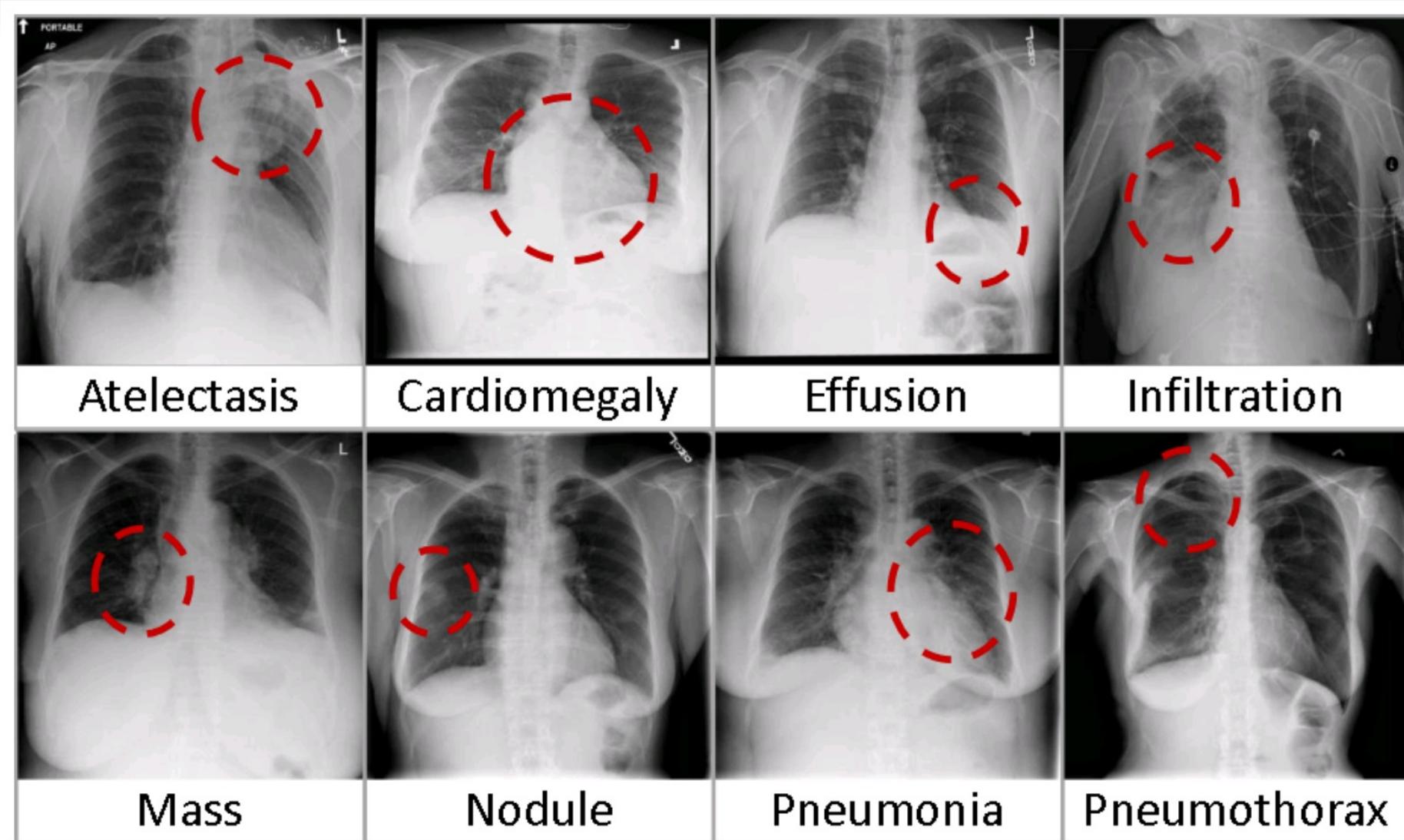
MuseNet

We've created MuseNet, a deep neural network that can generate 4-minute musical compositions with 10 different instruments, and can combine styles from country to Mozart to the Beatles. MuseNet was not explicitly programmed with our understanding of music, but instead discovered patterns of harmony, rhythm, and style by learning to predict the next token in hundreds of thousands of MIDI files. MuseNet uses the same general-purpose unsupervised technology as [GPT-2](#), a large-scale [transformer](#) model trained to predict the next token in a sequence, whether audio or text.

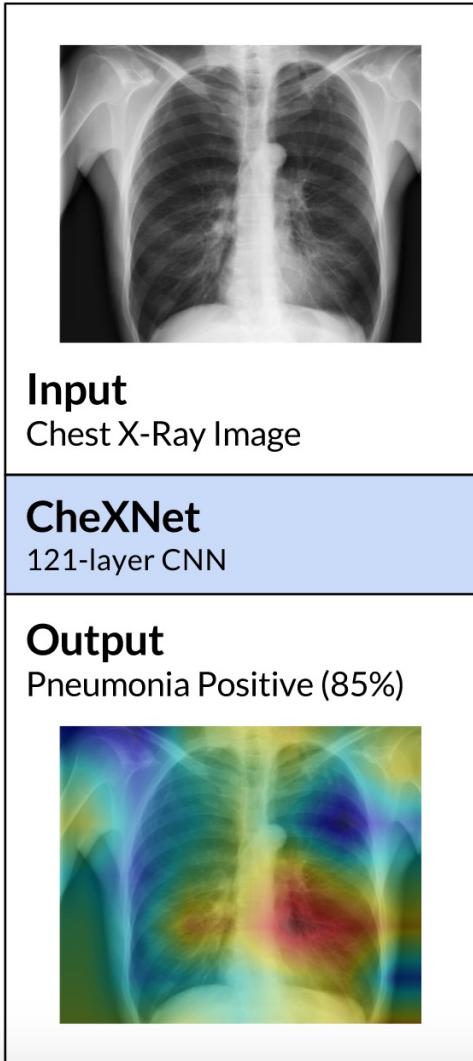


Clasificación de patologías en imágenes médicas

Clasificación de patologías en imágenes de rayos X



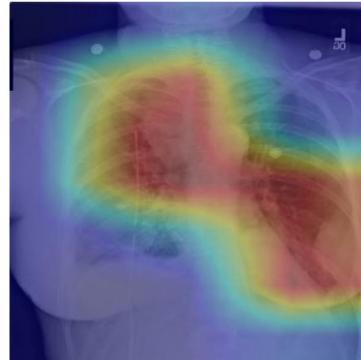
Clasificación de patologías en imágenes de rayos X con localización automática



(a) Patient with multifocal community acquired pneumonia. The model correctly detects the airspace disease in the left lower and right upper lobes to arrive at the pneumonia diagnosis.



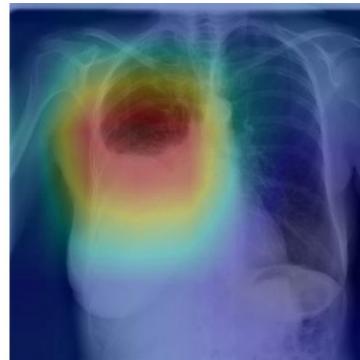
(b) Patient with a left lung nodule. The model identifies the left lower lobe lung nodule and correctly classifies the pathology.



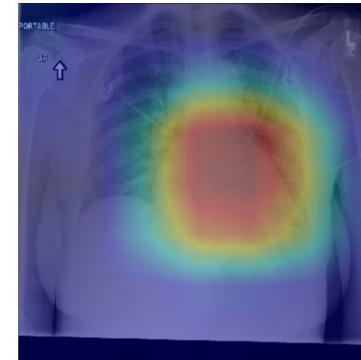
(c) Patient with primary lung malignancy and two large masses, one in the left lower lobe and one in the right upper lobe adjacent to the mediastinum. The model correctly identifies both masses in the X-ray.



(d) Patient with a right-sided pneumothorax and chest tube. The model detects the abnormal lung to correctly predict the presence of pneumothorax (collapsed lung).



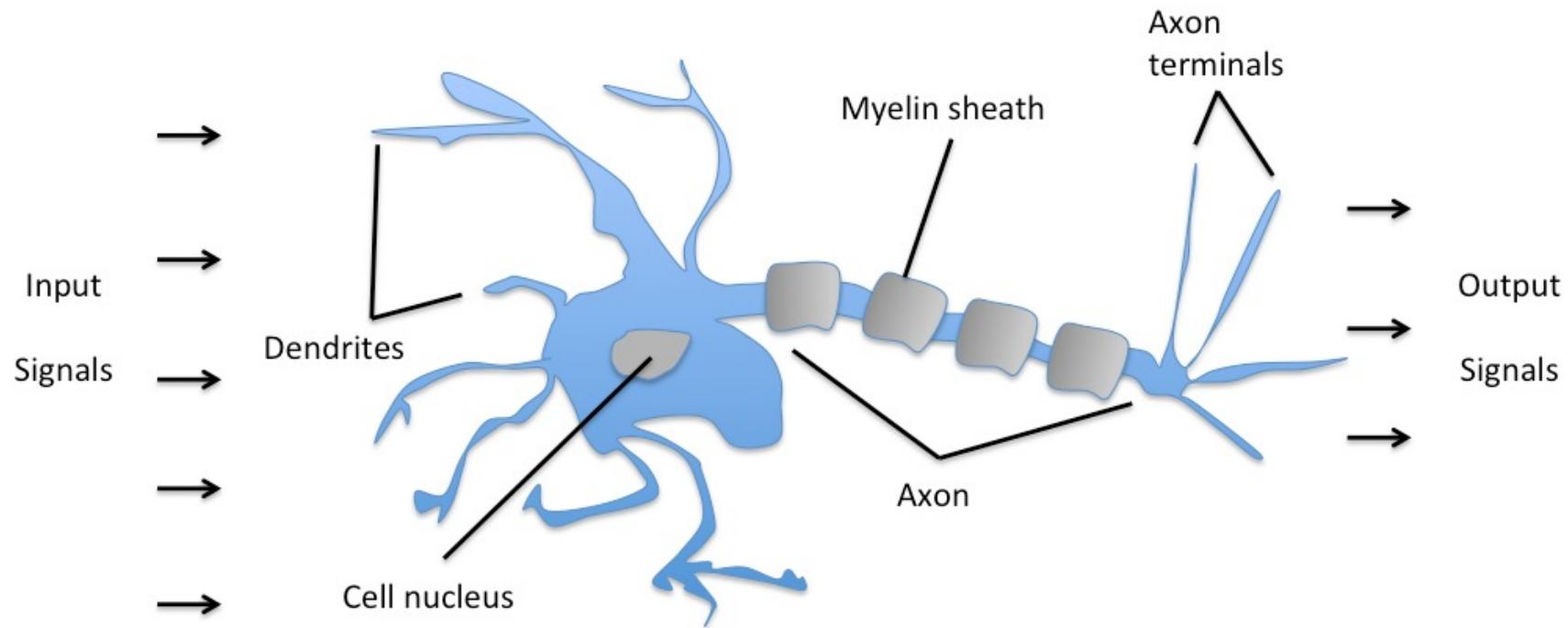
(e) Patient with a large right pleural effusion (fluid in the pleural space). The model correctly labels the effusion and focuses on the right lower chest.



(f) Patient with congestive heart failure and cardiomegaly (enlarged heart). The model correctly identifies the enlarged cardiac silhouette.

Perceptrón Simple

Neurona biológica



Schematic of a biological neuron.

Perceptrón

(Rosenblatt, 1958)

Psychological Review
Vol. 65, No. 6, 1958

THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN¹

F. ROSENBLATT

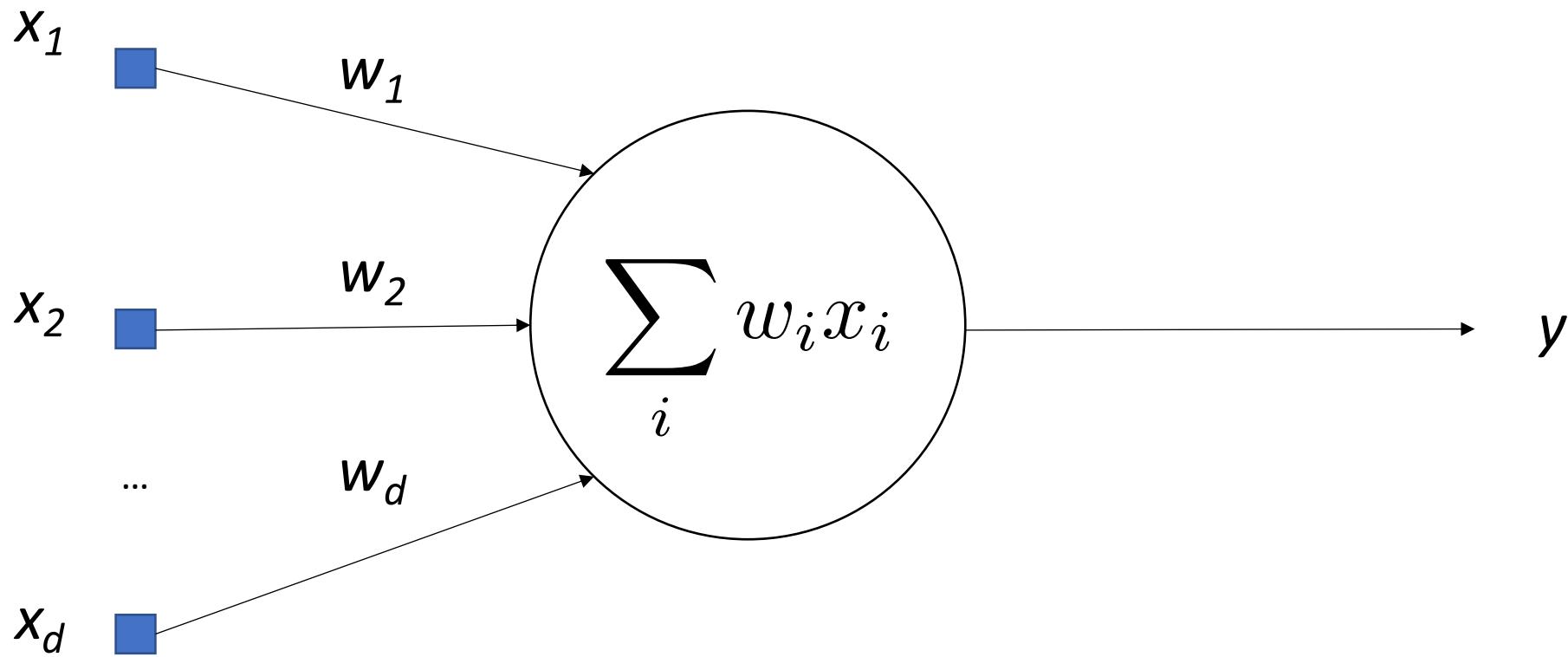
Cornell Aeronautical Laboratory

If we are eventually to understand the capability of higher organisms for perceptual recognition, generalization, recall, and thinking, we must first have answers to three fundamental questions:

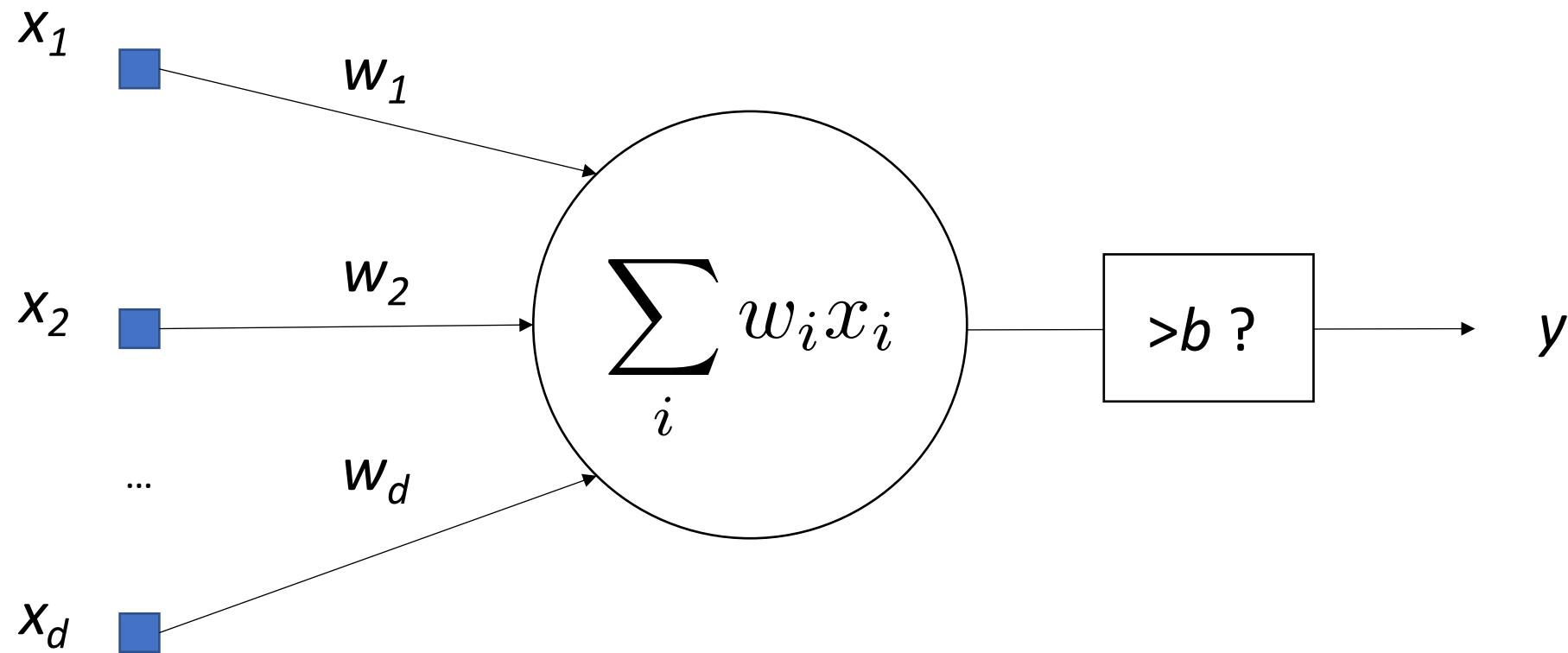
1. How is information about the physical world sensed, or detected, by the biological system?
2. In what form is information stored, or remembered?
3. How does information come to be stored?

and the stored pattern. According to this hypothesis, if one understood the code or "wiring diagram" of the nervous system, one should, in principle, be able to discover exactly what an organism remembers by reconstructing the original sensory patterns from the "memory traces" which they have left, much as we might develop a photographic negative, or translate the pattern of electrical charges "mem-

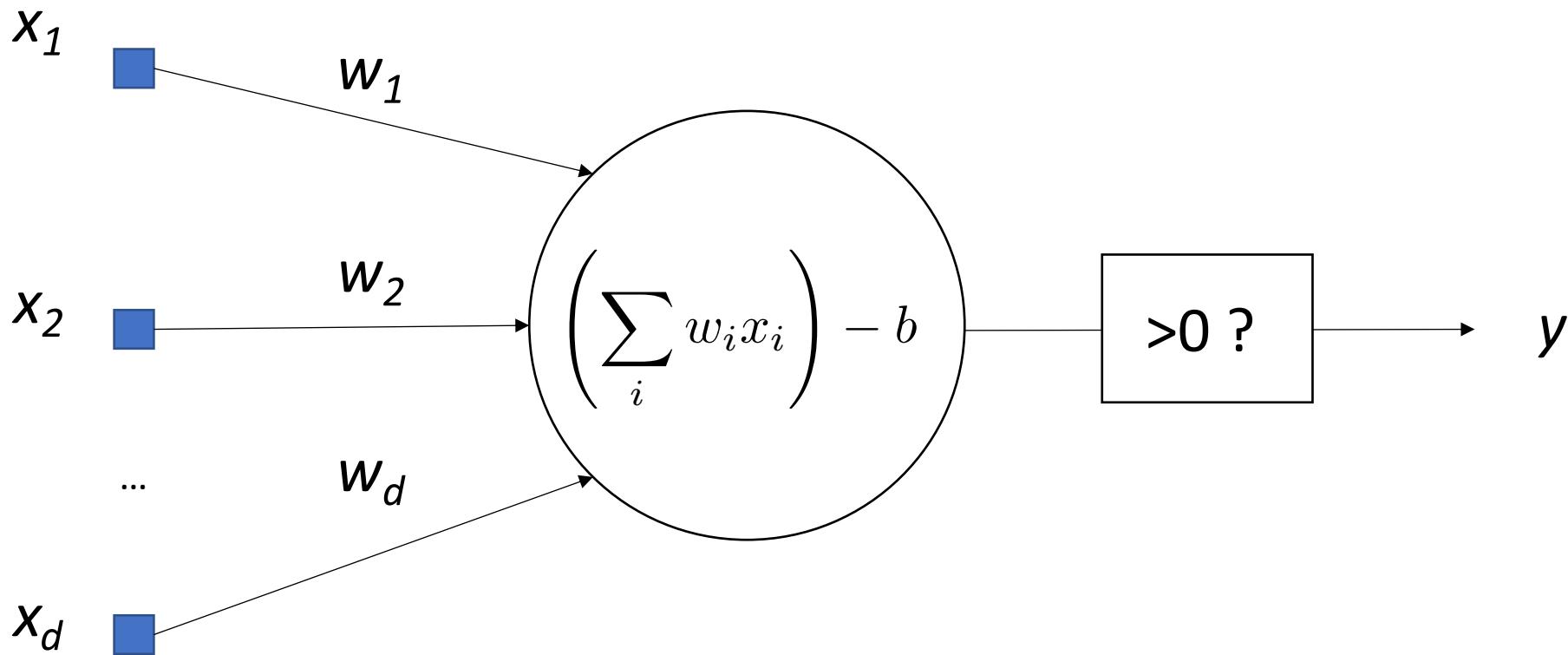
Neurona artificial



Neurona artificial

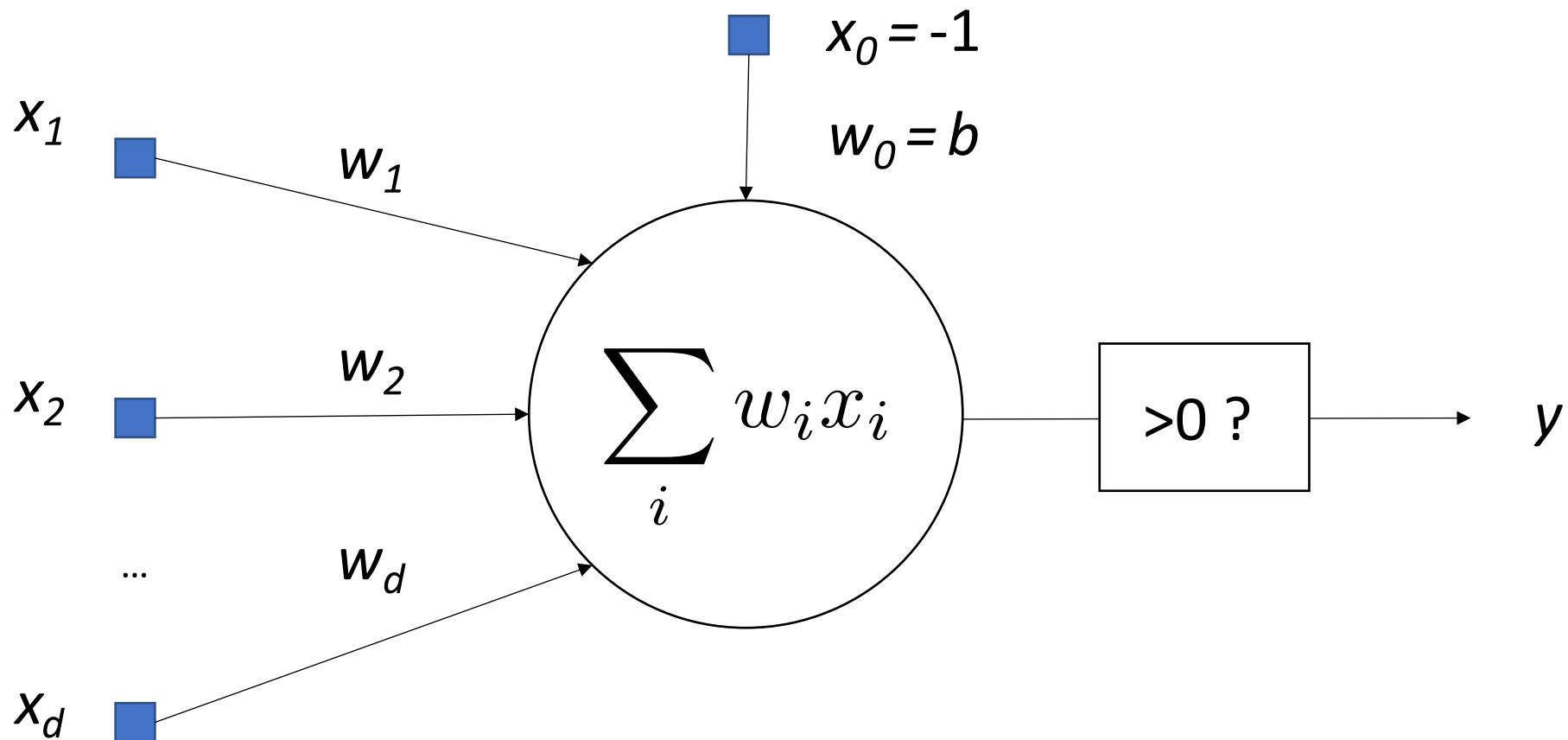


Neurona artificial



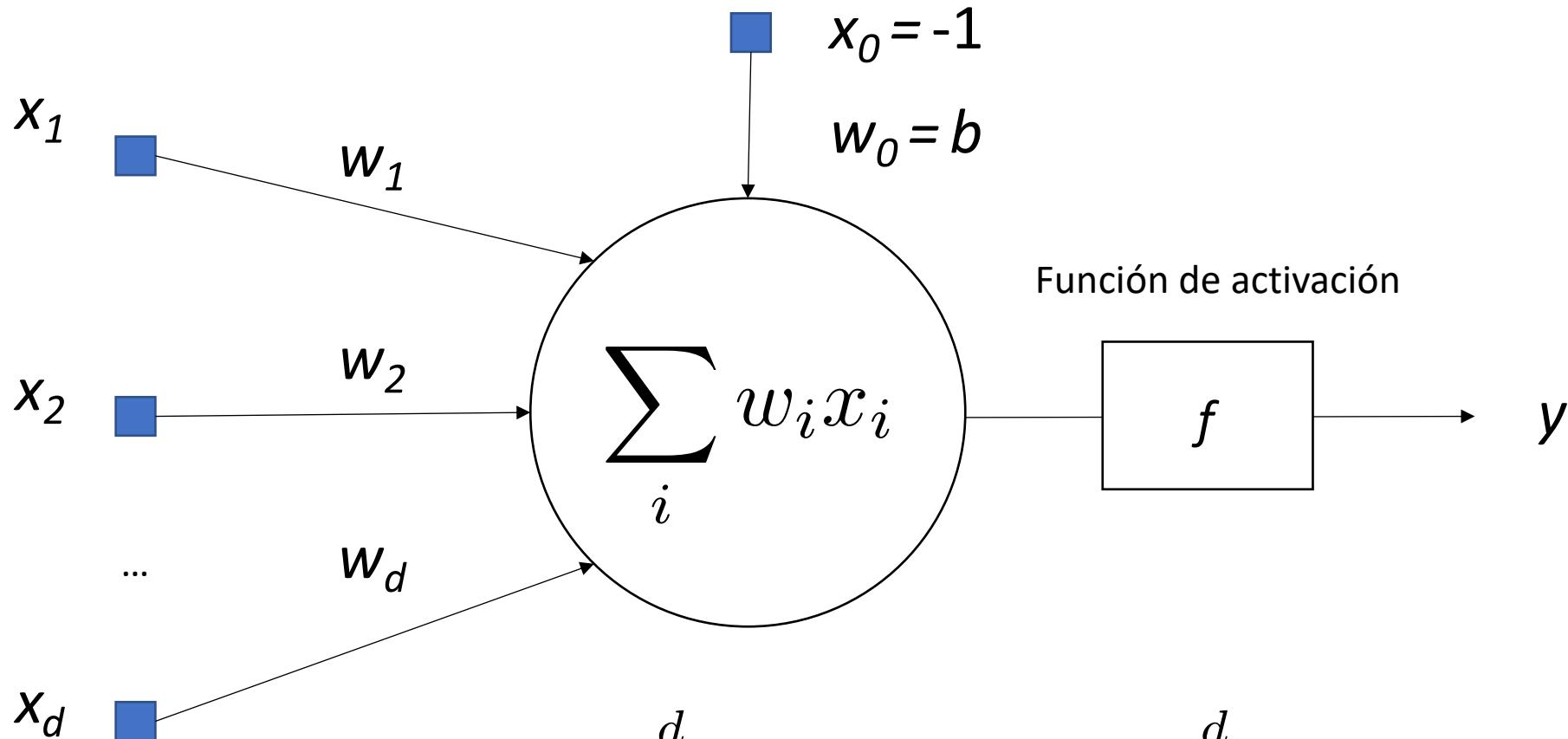
Neurona artificial

(Bias trick)



Perceptrón simple

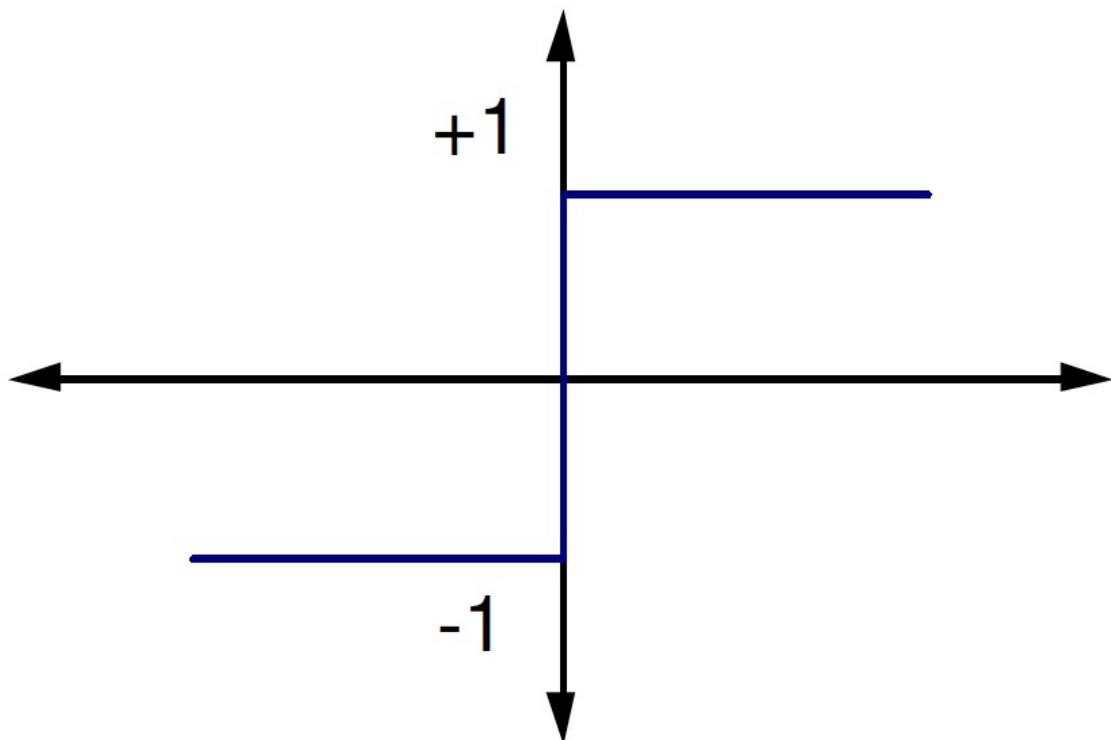
Perceptrón simple



$$y = f\left(\sum_{i=1}^d w_i x_i - b\right) = f\left(\sum_{i=0}^d w_i x_i\right) = f(\langle \mathbf{w}, \mathbf{x} \rangle)$$

Funciones de activación

Función signo



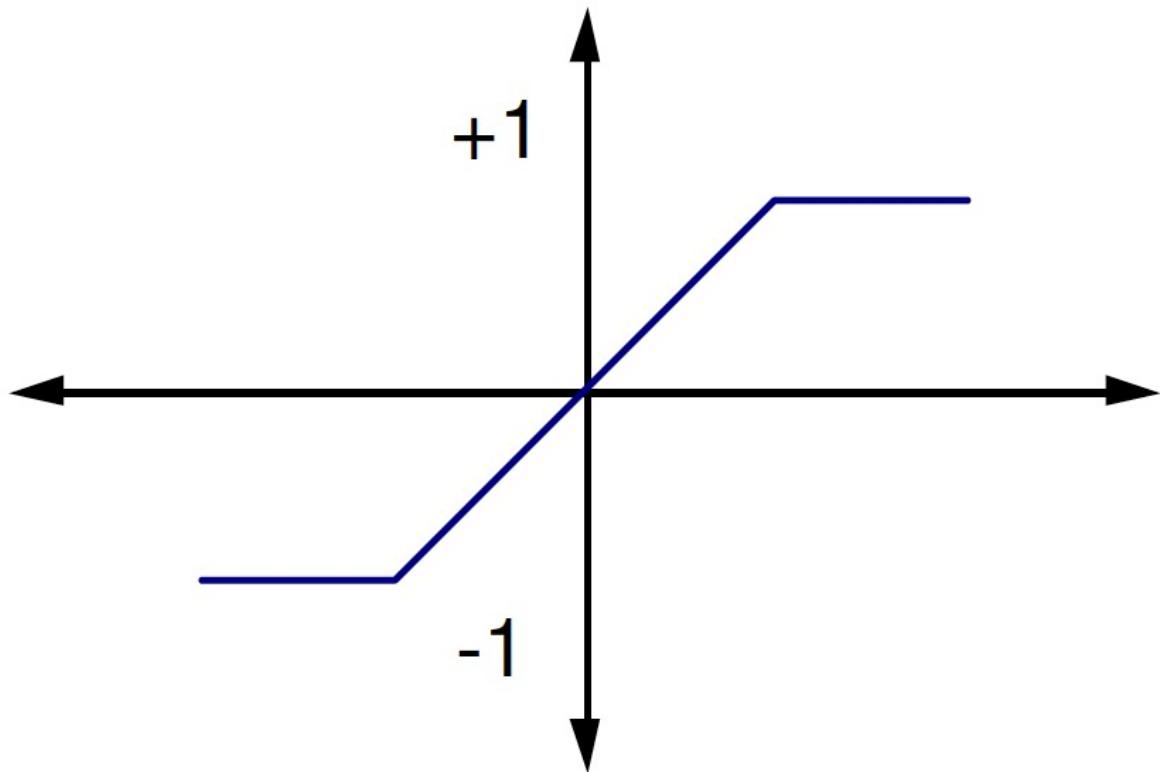
$$y = f(\langle \mathbf{w}, \mathbf{x} \rangle)$$

$$y = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle)$$

$$\text{sgn}(z) = \begin{cases} -1 & \text{si } z < 0 \\ +1 & \text{si } z \geq 0 \end{cases}$$

Funciones de activación

Función signo lineal



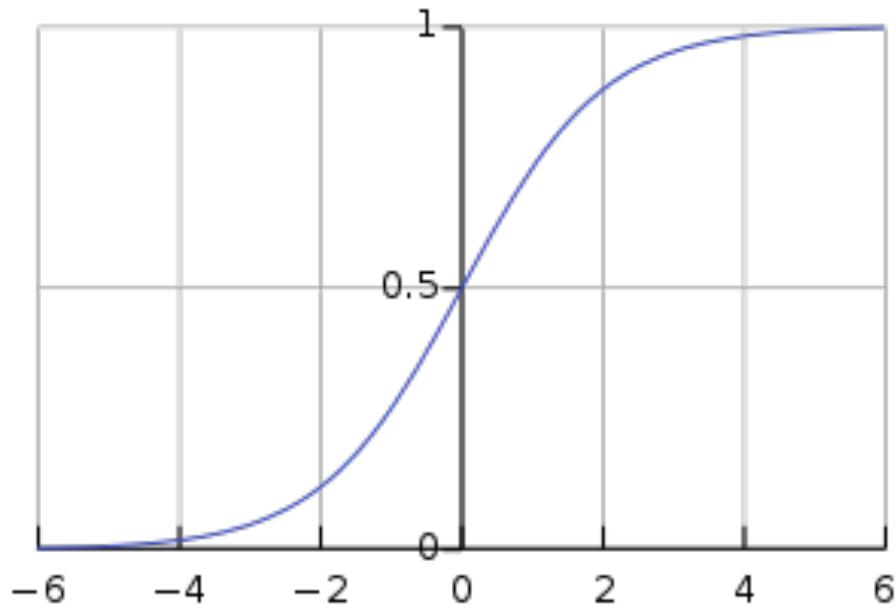
$$y = f(\langle \mathbf{w}, \mathbf{x} \rangle)$$

$$y = \text{sln}(\langle \mathbf{w}, \mathbf{x} \rangle)$$

$$\text{sln}(z) = \begin{cases} -1 & \text{si } z < -a \\ \alpha z & \text{si } -a < z < a \\ +1 & \text{si } z \geq a \end{cases}$$

Funciones de activación

Función Sigmoidea



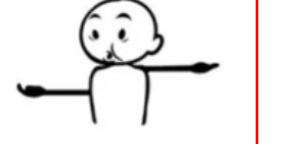
$$y = f(\langle \mathbf{w}, \mathbf{x} \rangle)$$

$$y = \sigma(\langle \mathbf{w}, \mathbf{x} \rangle)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

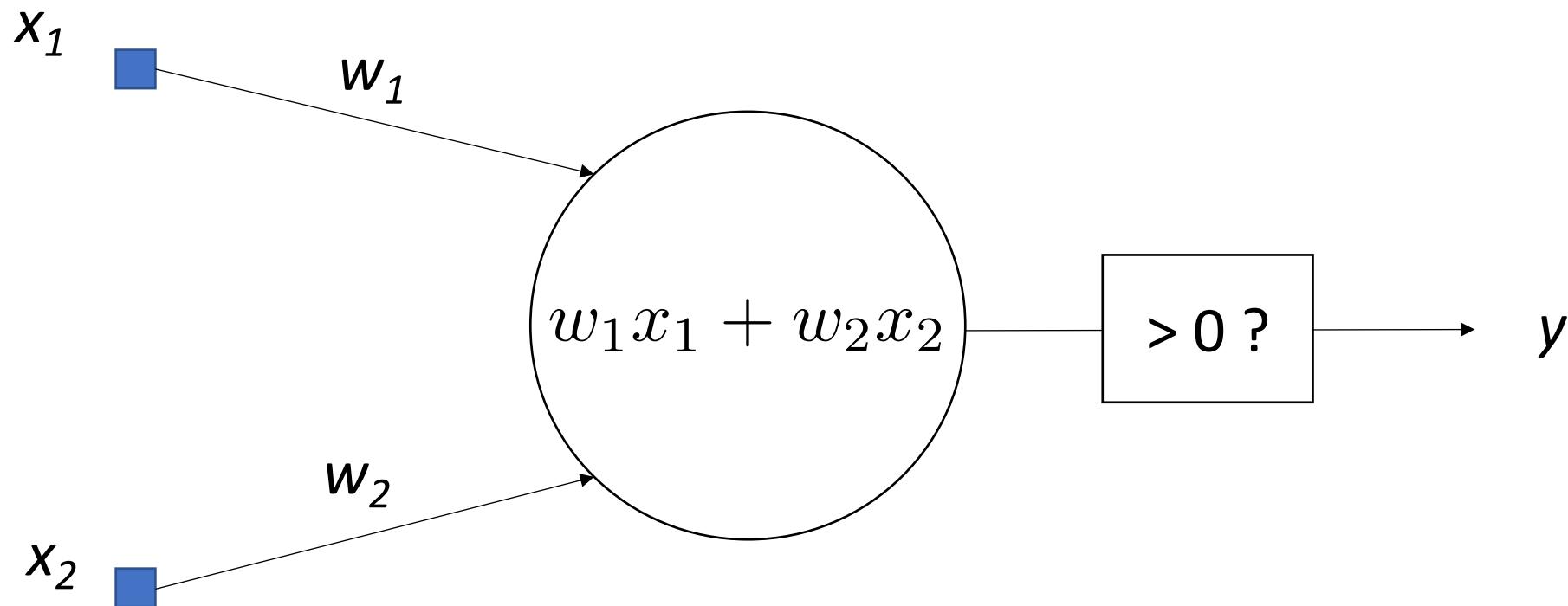
Funciones de activación

source: sefiks

Sigmoid	Tanh	Step Function	Softplus
 $y = \frac{1}{1+e^{-x}}$	 $y = \tanh(x)$	 $y = \begin{cases} 0, & x < n \\ 1, & x \geq n \end{cases}$	 $y = \ln(1+e^x)$
ReLU	Softsign	ELU	Log of Sigmoid
 $y = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$	 $y = \frac{x}{(1+ x)}$	 $y = \begin{cases} \alpha(e^x - 1), & x < 0 \\ x, & x \geq 0 \end{cases}$	 $y = \ln\left(\frac{1}{1+e^{-x}}\right)$
Swish	Sinc	Leaky ReLU	Mish
 $y = \frac{x}{1+e^{-x}}$	 $y = \frac{\sin(x)}{x}$	 $y = \max(0.1x, x)$	 $y = x(\tanh(\text{softplus}(x)))$

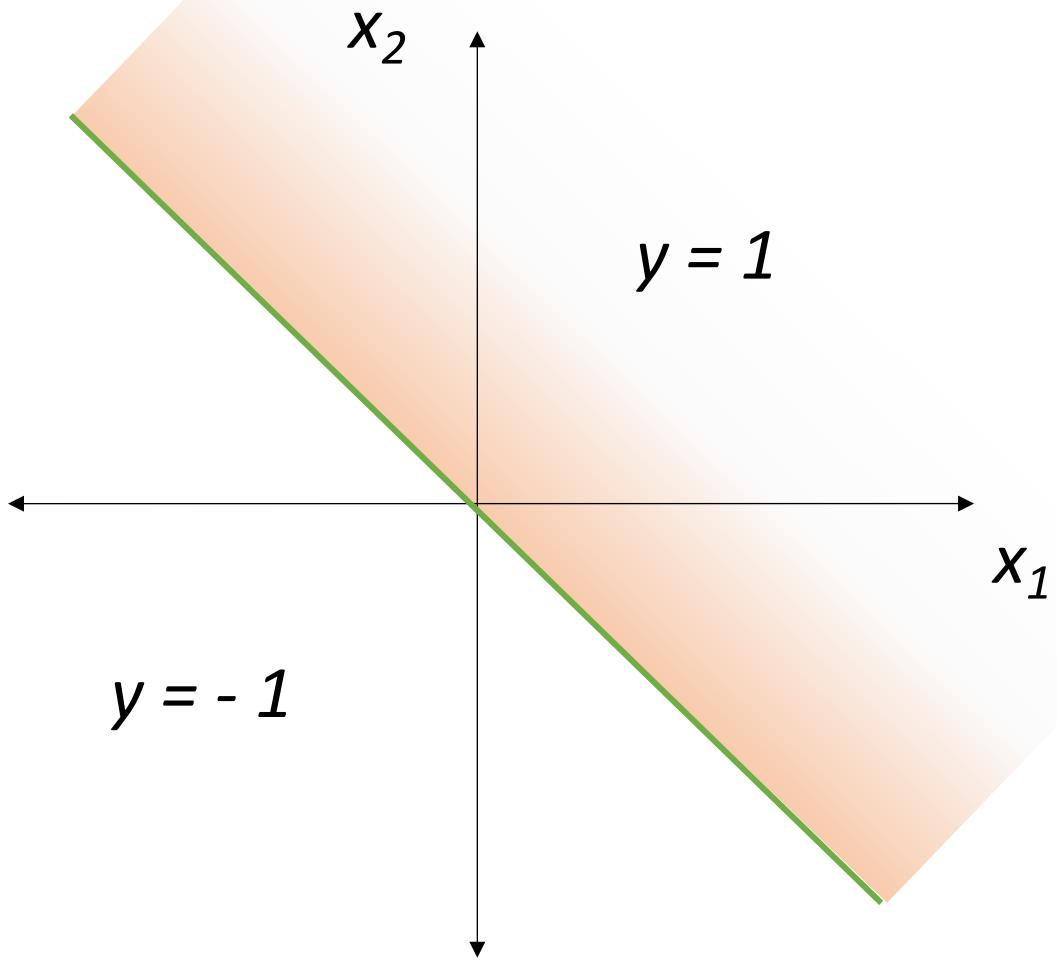
Ejemplo

Perceptrón simple con 2 entradas



Ejemplo

Representación gráfica de la frontera de decisión



$$w_1x_1 + w_2x_2 = 0$$

$$x_2 = -\frac{w_1}{w_2}x_1$$

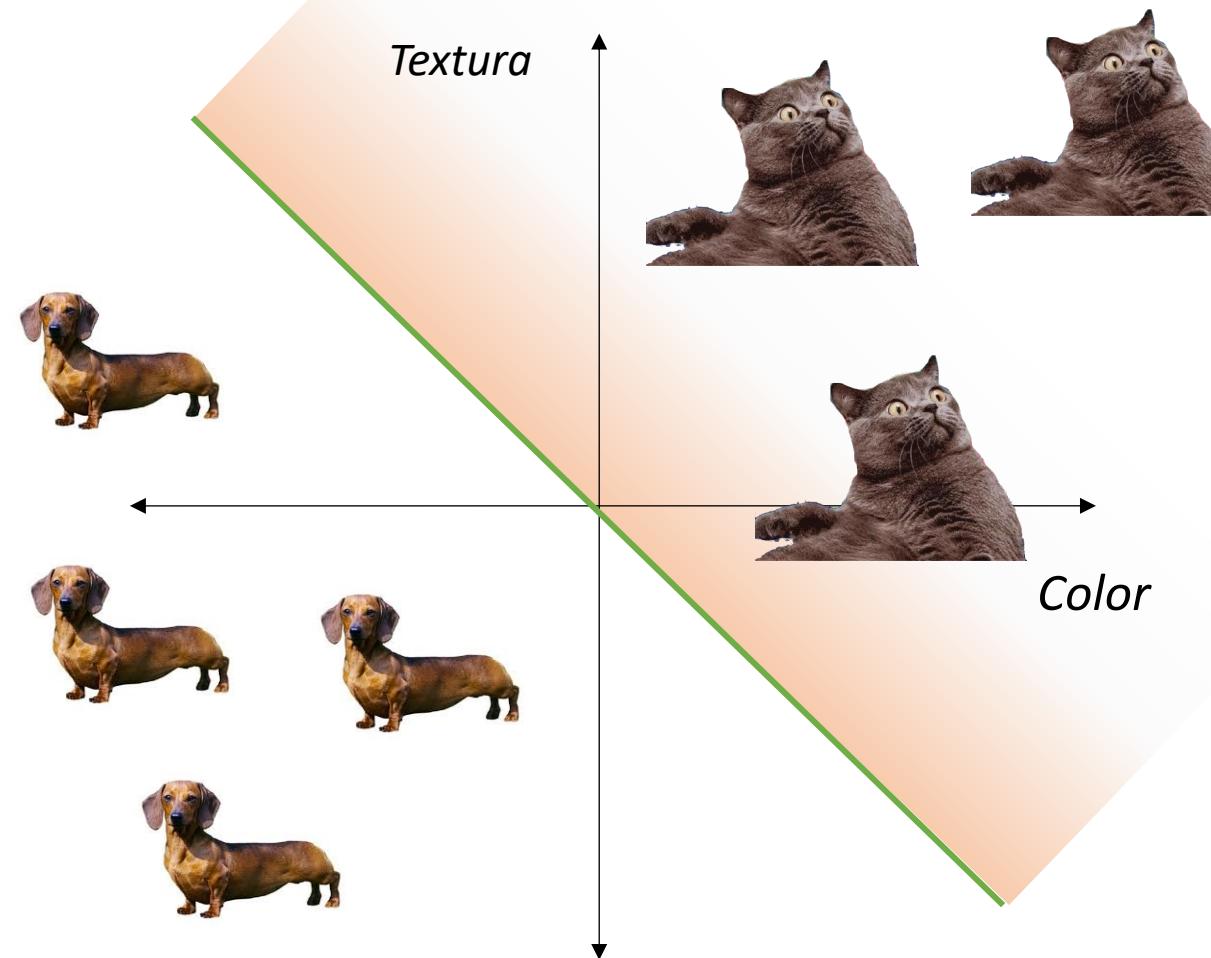
Ej: $w_1 = 1, w_2 = 1$

$$x_2 = -x_1$$

$$y = \text{sgn}(x_1w_1 + x_2w_2)$$

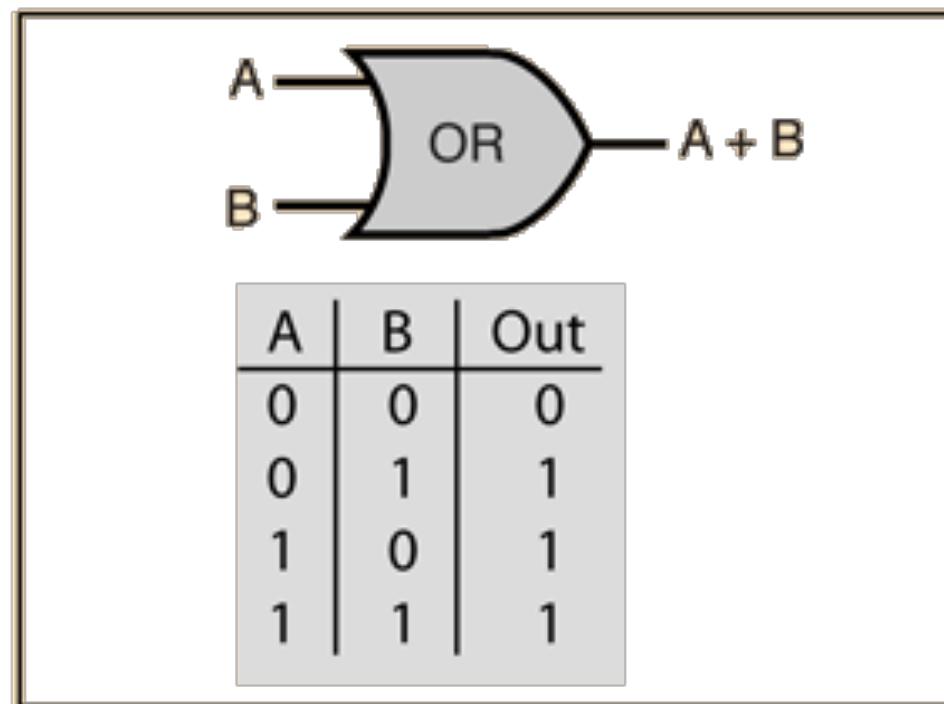
Ejemplo

Representación gráfica de la frontera de decisión al clasificar imágenes



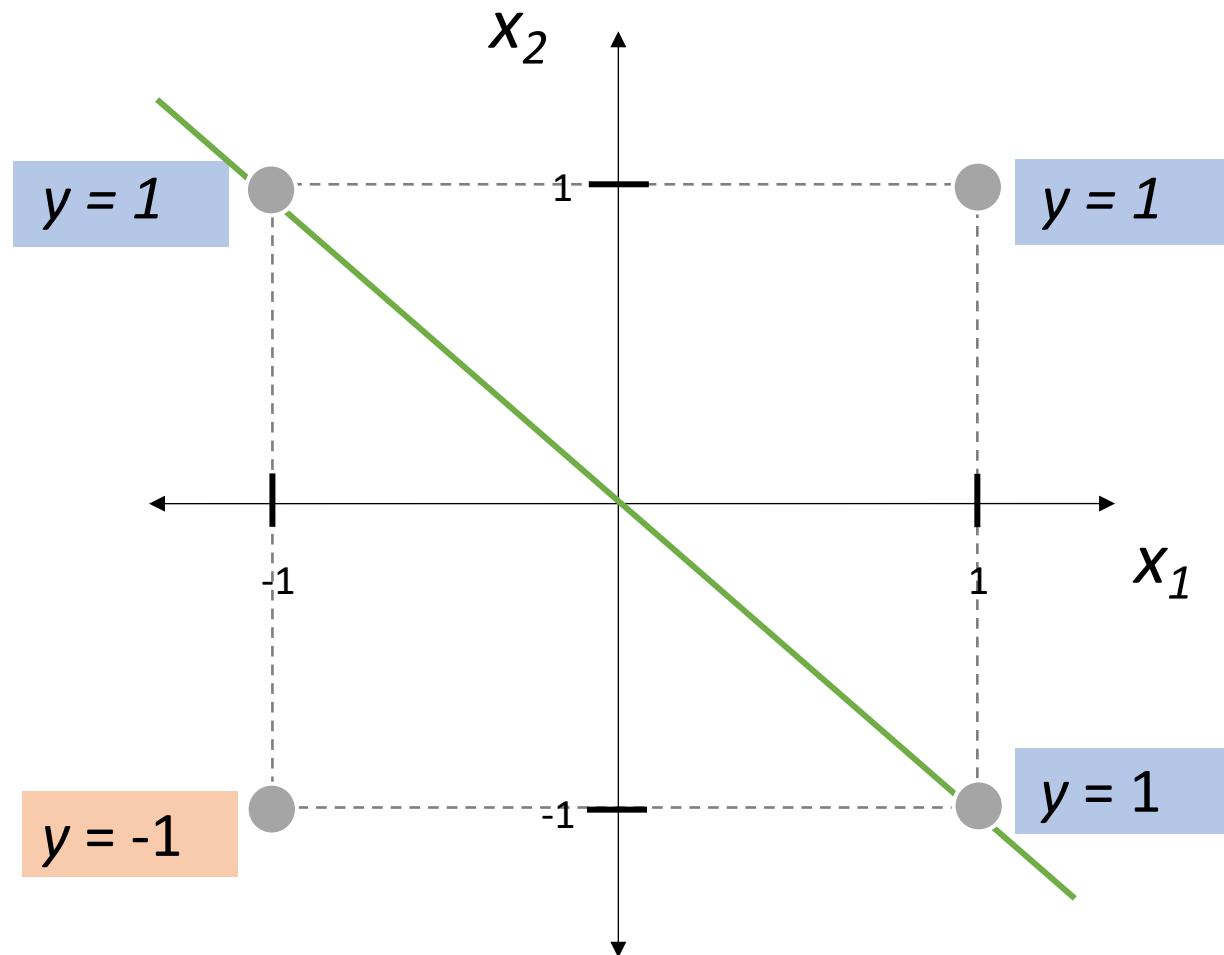
Ejemplo

Modelando la operación **x1 OR x2**



Ejemplo

Modelando la operación **x1 OR x2** (cambiando 0 por -1)



$$w_1x_1 + w_2x_2 = 0$$

Se necesita un término de bias

$$w_1x_1 + w_2x_2 - w_0 = 0$$

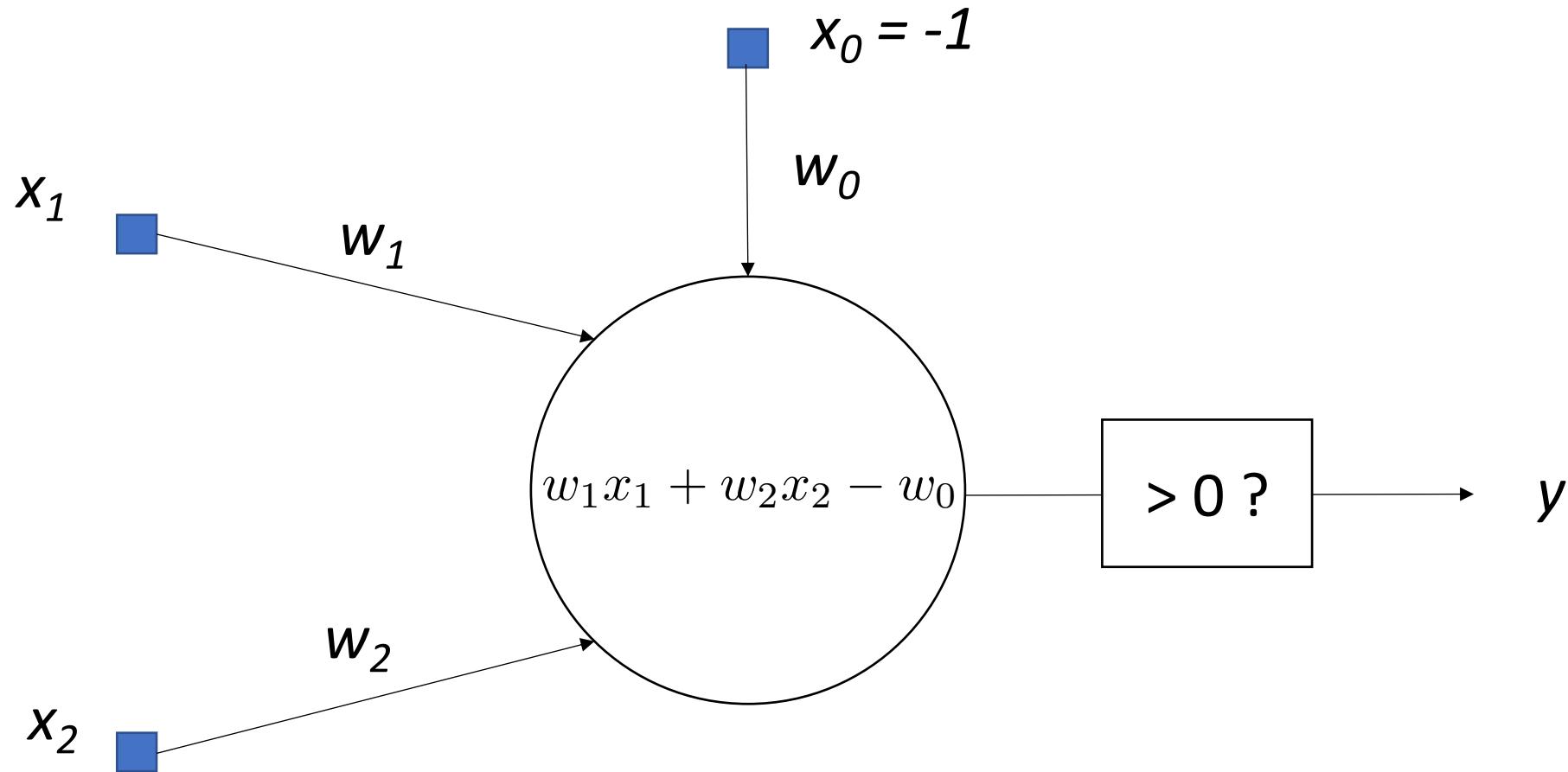
$$-\frac{w_1}{w_2}x_1 + \frac{w_0}{w_2} = x_2$$

Ej: $w_1 = w_2 = 1, w_0 = -1$

$$-x_1 - 1 = x_2$$

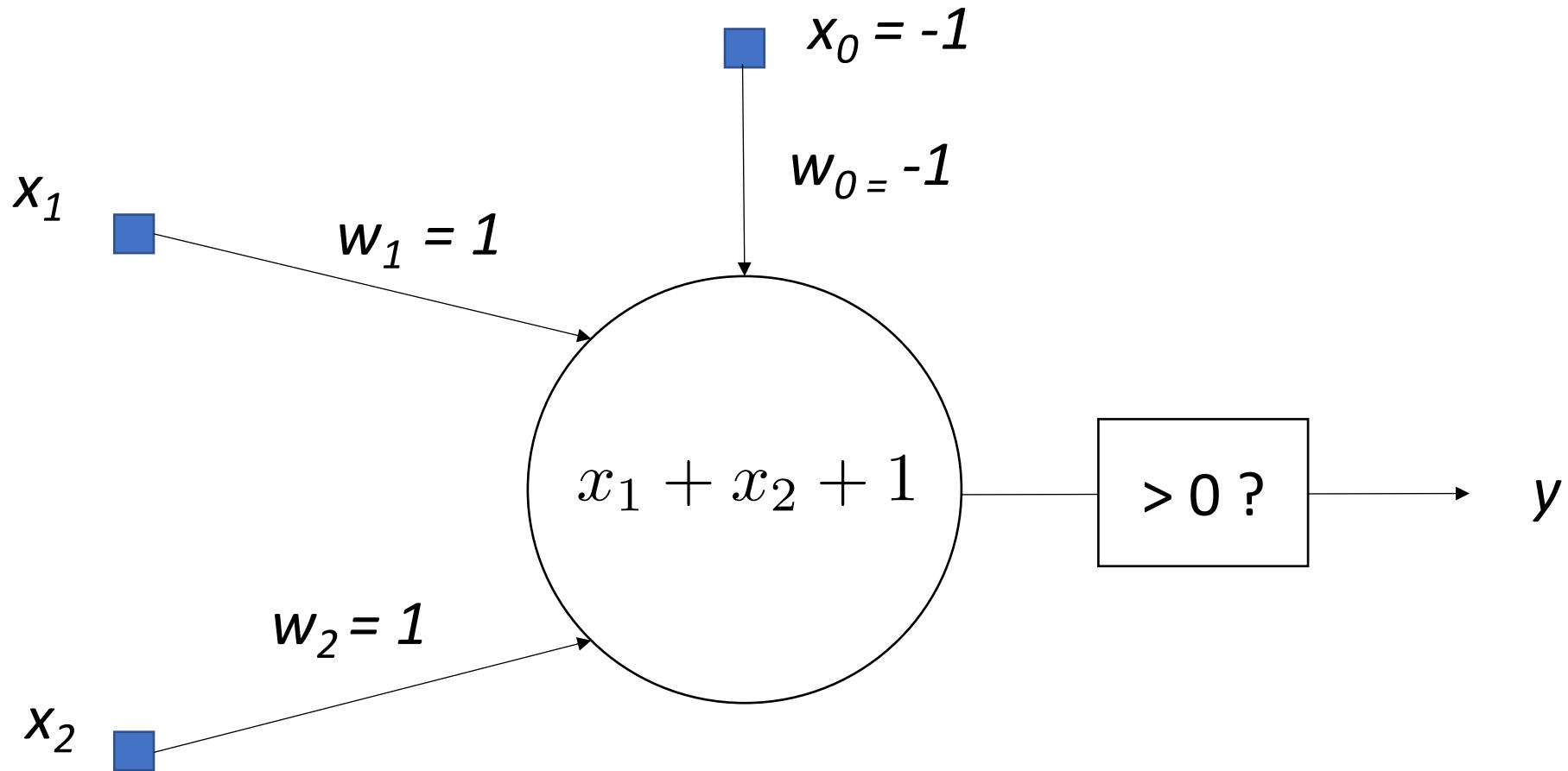
Ejemplo

Perceptrón simple con 2 entradas y bias



Ejemplo

Perceptrón simple con 2 entradas y bias modelando la operación **x1 OR x2**



Cómo entrenar mi perceptrón?

Cómo entrenar mi perceptrón?

Dataset de entrenamiento

$$\mathcal{D} = \{(\mathbf{x}, d)_n\} = \{(\mathbf{x}, d)_1, \dots (\mathbf{x}, d)_N\}$$

En nuestro problema de ejemplo:

$$\mathcal{D} =$$

$\mathbf{x} = (-1, -1)$	$d = -1$
$\mathbf{x} = (1, -1)$	$d = 1$
$\mathbf{x} = (-1, 1)$	$d = 1$
$\mathbf{x} = (1, 1)$	$d = 1$

Cómo entrenar mi perceptrón?

Función de pérdida (o función de costo)

- En el contexto del aprendizaje automático, es una función que **mide la calidad de una predicción basándose en el acuerdo entre dicha predicción y la etiqueta provista** en el dataset de entrenamiento
- La función de costo **devuelve un valor alto si nuestra predicción es incorrecta, y bajo si la predicción es correcta**
- Sirve como **guía en el proceso de búsqueda de los parámetros** de nuestro modelo a entrenar

Cómo entrenar mi perceptrón?

Función de pérdida

$$\mathcal{L}((\mathbf{x}, d); \mathbf{w})$$

Función de pérdida

Error cuadrático instantáneo

$$\mathcal{L}((\mathbf{x}, d); \mathbf{w}) = (d - \langle \mathbf{w}, \mathbf{x} \rangle)^2$$

Caso Lineal
(sin función de activación)

Aprendizaje como un problema de optimización

Algoritmo de optimización Función de pérdida

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N \mathcal{L}((\mathbf{x}, d)_n; \mathbf{w})$$

Parámetros del modelo
entrenado

Estrategias para buscar los parámetros óptimos

1. Búsqueda aleatoria

$$\mathcal{L}_{\text{Best}} = \inf$$

for $i = 1$ to #iteraciones:

\mathbf{w}_i = generar pesos aleatorios

$$\hat{\mathcal{L}} = \frac{1}{N} \sum_{n=1}^N \mathcal{L}((\mathbf{x}, d)_n; \mathbf{w}_i)$$

If $\hat{\mathcal{L}} < \mathcal{L}_{\text{Best}}$

$$\mathcal{L}_{\text{Best}} = \hat{\mathcal{L}}$$

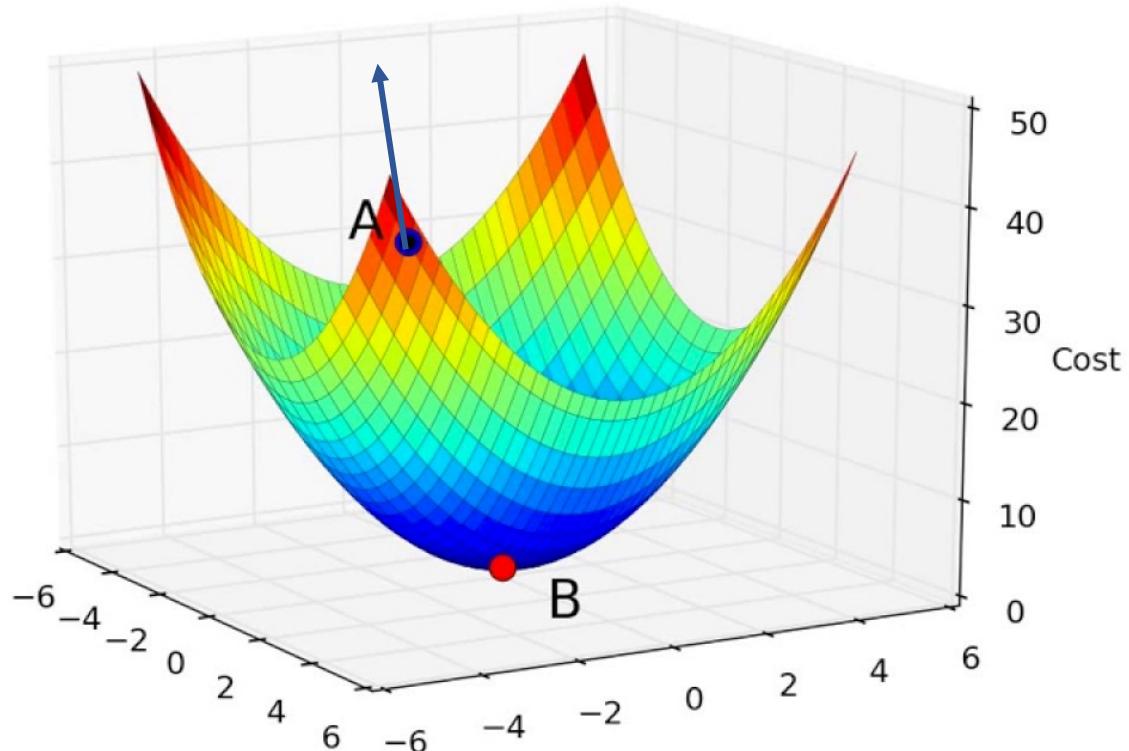
$$\mathbf{w}_{\text{best}} = \mathbf{w}_i$$

Y si en lugar de buscar aleatoriamente vamos refinando en cada paso?

Estrategias para buscar los parámetros óptimos

2. Siguiendo el gradiente

Actualizar los pesos en la dirección en **que garantice una reducción de la función de pérdida**



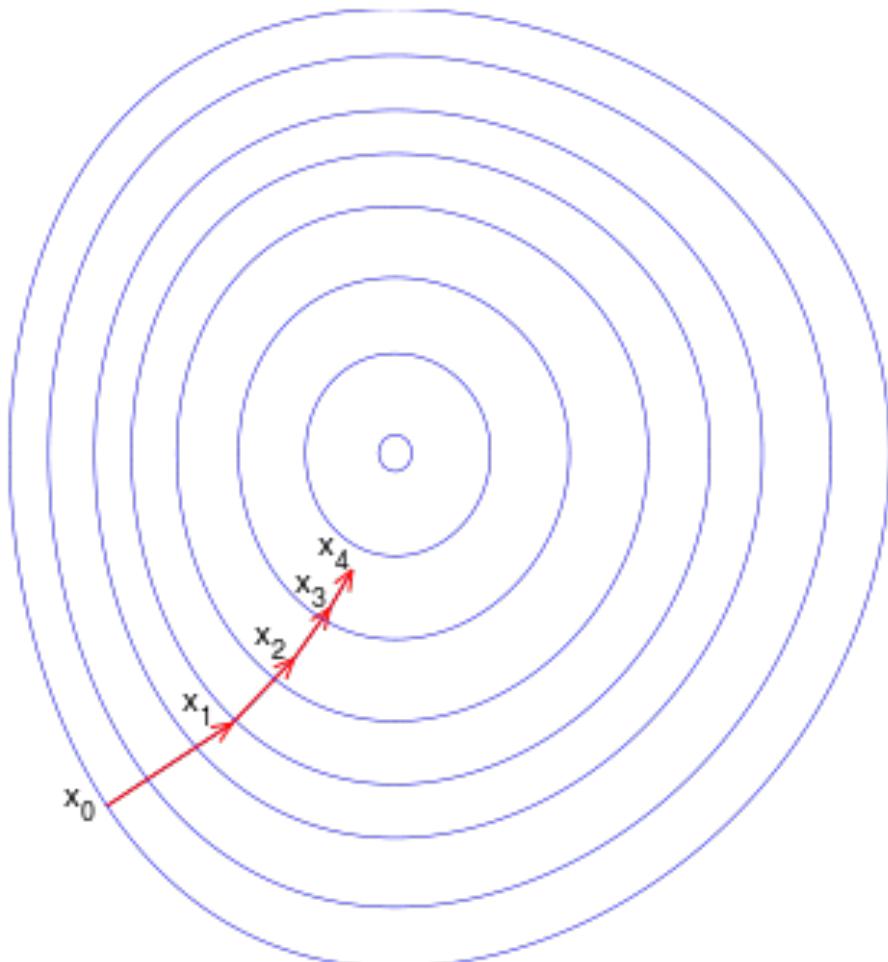
El vector **gradiente** apunta en la dirección de mayor crecimiento de la función

$$\nabla_{\mathbf{w}} \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial w_1}, \dots, \frac{\partial \mathcal{L}}{\partial w_D} \right)$$

Indica cómo cambiará el valor de la función en un entorno infinitesimalmente pequeño alrededor de un punto al variar el peso w_1

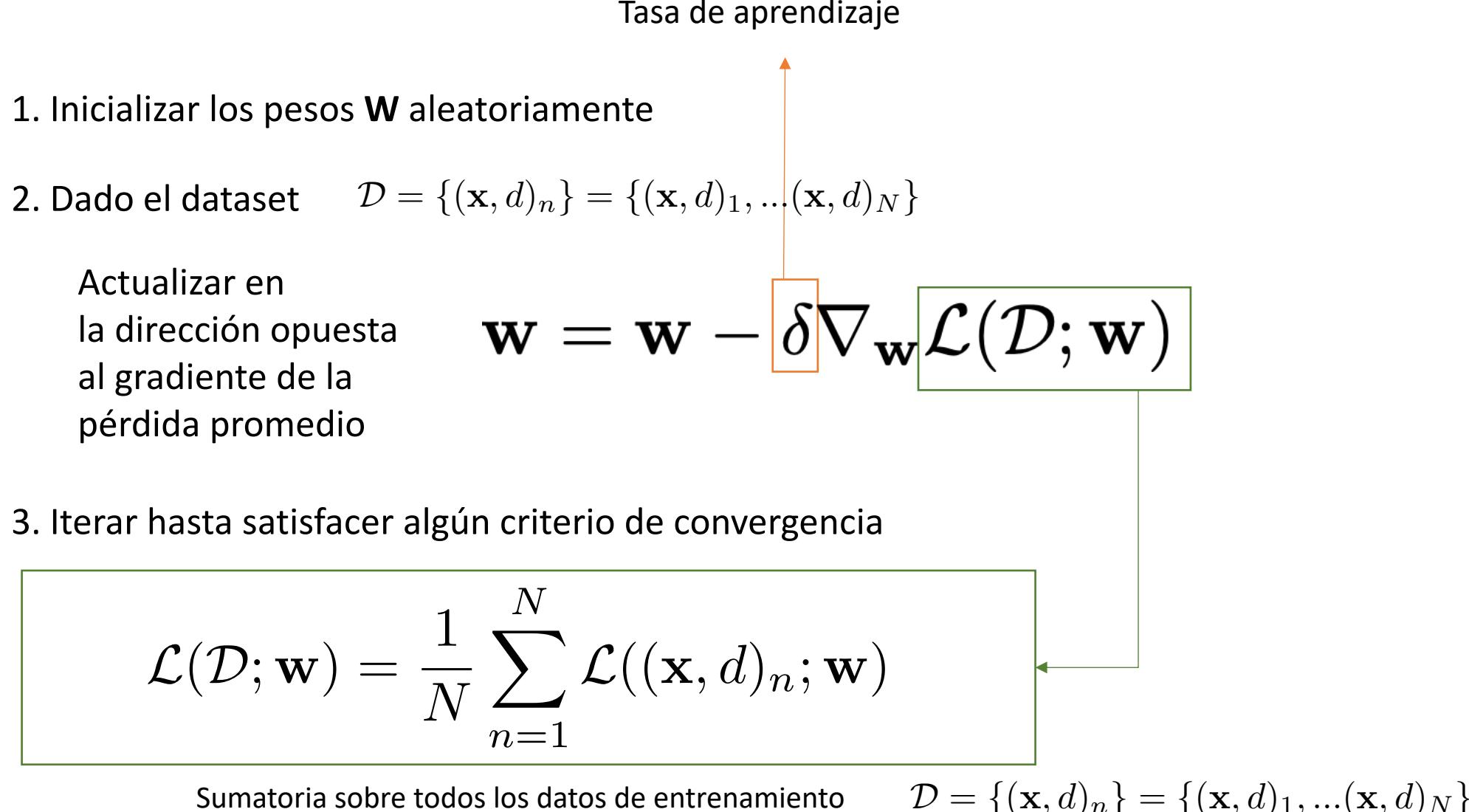
Estrategias para buscar los parámetros óptimos

3. Siguiendo el gradiente



- El método del gradiente descendente **garantiza la convergencia a un mínimo local** de la función a optimizar.
- Si la función es **convexa, el mínimo local es a su vez el mínimo global.**
- El **error cuadrático** utilizado como función de pérdida del **perceptrón simple en el caso lineal** resulta en una función **convexa**.

Método del gradiente descendente



Ejemplo de cálculo del gradiente

Caso lineal del perceptrón

Función de pérdida

$$\mathcal{L}((\mathbf{x}, d); \mathbf{w}) = (d - \langle \mathbf{w}, \mathbf{x} \rangle)^2$$

Gradiente de la función de pérdida

$$\nabla \mathcal{L}_{\mathbf{w}} = 2(d - \langle \mathbf{w}, \mathbf{x} \rangle)(-\mathbf{x})$$

Método del gradiente descendente

1. Inicializar los pesos \mathbf{W} aleatoriamente
2. Dado el dataset $\mathcal{D} = \{(\mathbf{x}, d)_n\} = \{(\mathbf{x}, d)_1, \dots, (\mathbf{x}, d)_N\}$

Actualizar en
la dirección opuesta
al gradiente de la
pérdida promedio

$$\mathbf{w} = \mathbf{w} - \delta \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}; \mathbf{w})$$

3. Iterar hasta satisfacer algún criterio de convergencia

Método del gradiente descendente

1. Inicializar los pesos \mathbf{W} aleatoriamente
2. Dado el dataset $\mathcal{D} = \{(\mathbf{x}, d)_n\} = \{(\mathbf{x}, d)_1, \dots, (\mathbf{x}, d)_N\}$

Actualizar en
la dirección opuesta
al gradiente de la
pérdida promedio

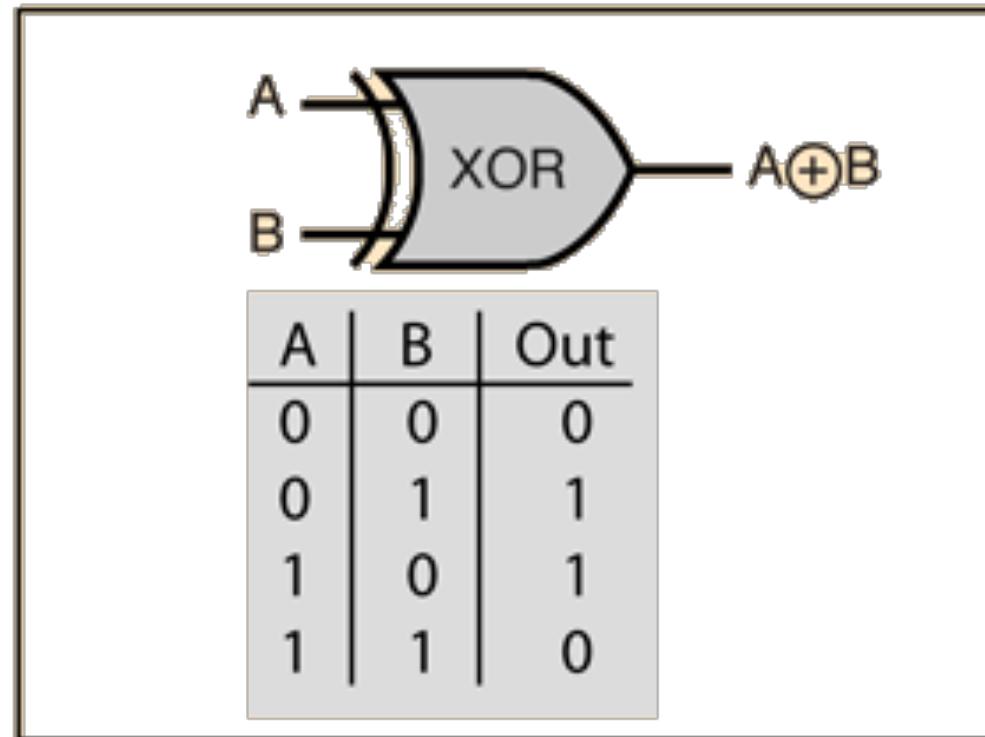
$$\mathbf{w} = \mathbf{w} - \delta \frac{1}{N} \sum_{n=1}^N 2(d_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)(-\mathbf{x}_i)$$

3. Iterar hasta satisfacer algún criterio de convergencia

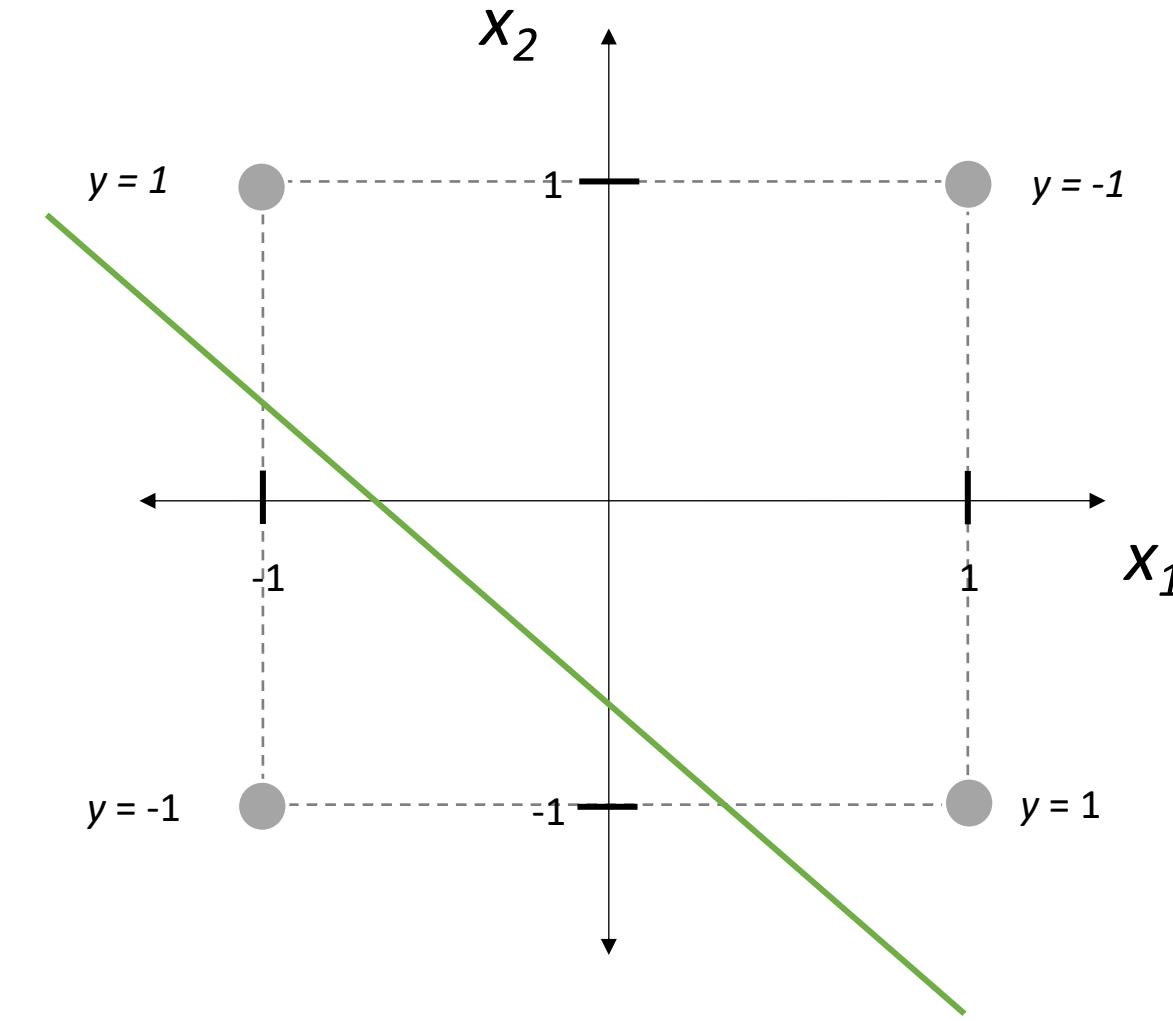
$$\mathcal{D} =$$

$\mathbf{x} = (-1, -1)$	$d = -1$
$\mathbf{x} = (1, -1)$	$d = 1$
$\mathbf{x} = (-1, 1)$	$d = 1$
$\mathbf{x} = (1, 1)$	$d = 1$

El perceptrón simple puede modelar el XOR?

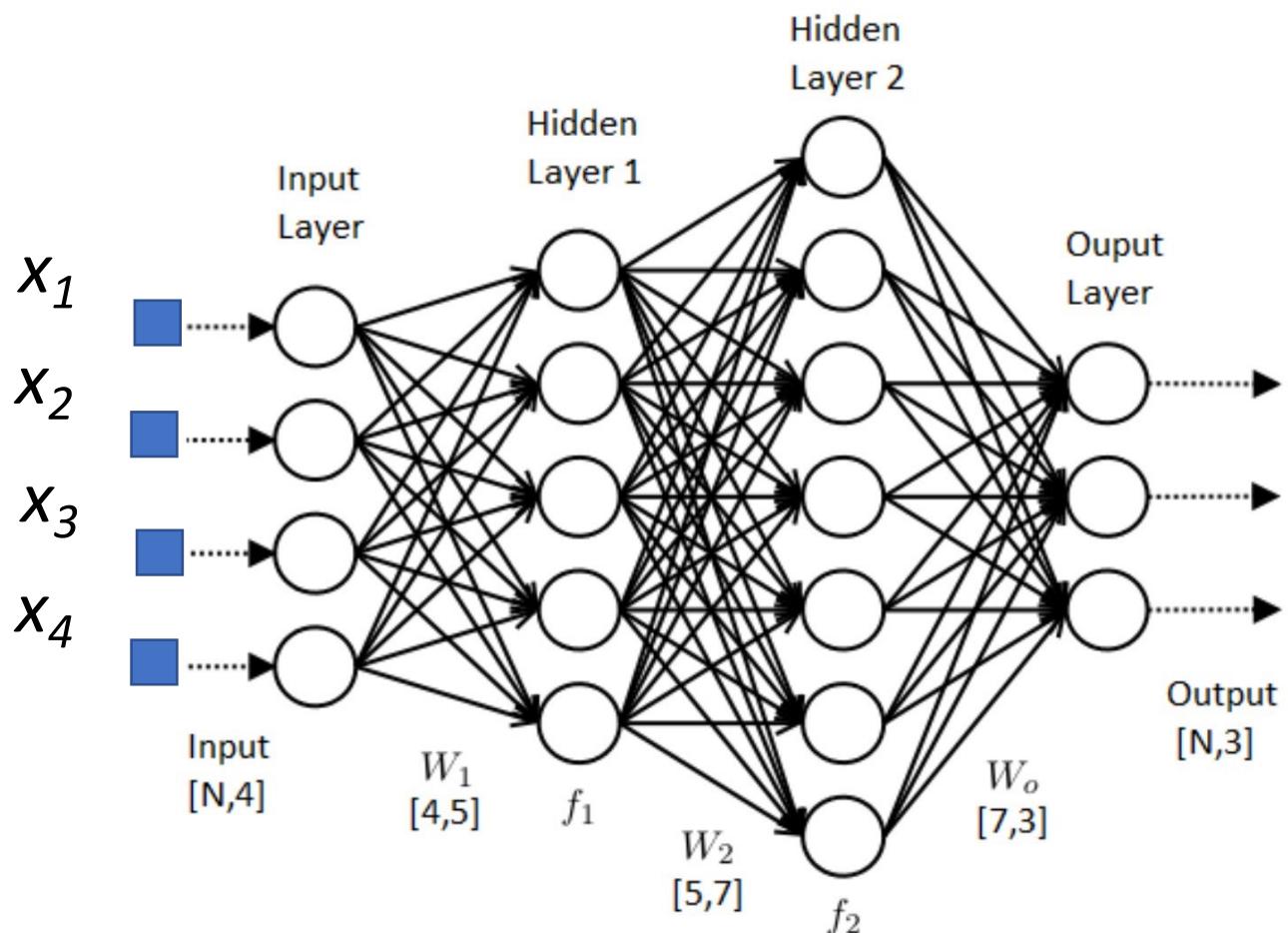


El perceptrón simple puede modelar el XOR?



Perceptrón Multicapa

Multi layer perceptron (MLP)



Continuará ...