

Aprendizaje profundo basado en la física

Semana 1: Introducción al aprendizaje automático

Docente: José I. Robledo - 07/04/2026

Introducción al Aprendizaje Automático

- Qué es aprendizaje automático?
 - “A computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .” *Tom Mitchel, Carnegie Mellon University 1998*
- ¿Dónde encontramos Aprendizaje automático cotidianamente?

Introducción al Aprendizaje Automático

Algoritmos de búsqueda

Google search results for "machine learning". The search bar shows "machine learning" with a microphone icon. Below the search bar are navigation tabs: "Modo IA", "Todo", "Imágenes", "Shopping", "Videos", "Videos cortos", "Noticias", "Más", and "Herramientas".

Visión general creada por IA

El Machine Learning (aprendizaje automático) es una rama de la inteligencia artificial que permite a los ordenadores aprender de datos y mejorar su experiencia sin ser programados explícitamente. Utiliza algoritmos estadísticos para identificar patrones, realizar predicciones y tomar decisiones autónomas, siendo esencial en aplicaciones como motores de recomendación, vehículos autónomos y detección de fraudes.

Este video explica qué es el Machine Learning de manera sencilla.

Tipos de machine learning | IBM
¿Qué es el machine learning (ML) o aprendizaje...

Aprendizaje automático - Wikipedia, la enciclopedia libre
Es el que se obtiene al generalizar varios conceptos...

Mostrar más

Más preguntas

- ¿Qué es machine learning y para qué sirve?
- ¿Cuáles son los 3 tipos de machine learning?
- ¿cuál es la diferencia entre IA y machine learning?

Search results for "¿Qué es el machine learning (ML) o aprendizaje automático?".

IBM
https://www.ibm.com › es-es › think › topics › machine...
¿Qué es el machine learning (ML) o aprendizaje automático?
El machine learning es el subconjunto de la inteligencia artificial (IA) centrado en algoritmos que pueden "aprender" los patrones de los datos de ...
¿Qué es el machine learning? | Cómo funciona el machine...

Google for Developers
https://developers.google.com › intro-to-ml › what-is-ml
¿Qué es el aprendizaje automático?
En términos básicos, ML es el proceso de entrenar una pieza de software, llamada modelo, para hacer predicciones útiles o generar contenido (como texto, ...
Traducido por Google · Ver original (English)


Amazon Web Services (AWS)
https://aws.amazon.com › ... › Machine learning
¿Qué es el machine learning? - Explicación de la ...
El machine learning es un tipo de inteligencia artificial que lleva a cabo tareas de análisis de datos sin instrucciones explícitas.

MIT Sloan
https://mitsloan.mit.edu › machine-learning-explained
Aprendizaje automático, explicado
21 abr 2021 — El aprendizaje automático es un subcampo de la inteligencia artificial, que se define ampliamente como la capacidad de una máquina para imitar ...
Traducido por Google · Ver original (English)

Instituto Data Science Argentina
https://institutodatascience.org › guia › machine-learning
¿Qué es el Machine Learning y cómo aplicarlo en ...
El Machine Learning (aprendizaje automático) es una rama de la Inteligencia Artificial (IA) que permite que las computadoras aprendan a partir de los datos sin ...

Introducción al Aprendizaje Automático

Sistemas de recomendación

Made For **José Robledo**  [Show all](#)

Daily Mix 03
Joseph Banowetz, Janos Sebestyen,...

Daily Mix 04
The Rolling Stones, Led Zeppelin,...

Daily Mix 05
Willy González, Raúl Carnota, Ernesto...

Daily Mix 06
Red Hot Chili Peppers, Muse, Foo Fighters...

Your top mixes [Show all](#)

Ramin Djawadi Mix
Alexandre Desplat, Steven Price and...

AnnenMayKantereit Mix
Provinz, Von Wegen Lisbeth and Mayberg

2000s Mix
Los Ángeles Azules, Nightwish, Gorillaz...

Happy Mix
Edward Sharpe & The Magnetic Zeros, La...

NETFLIX Home Shows Movies Games New & Popular My List Browse by Languages

Frankenstein Oscar Nominee

THE SUPER MARIO BROS. MOVIE

Agatha Christie's SEVEN DIALS TOP 10 Recently Added

WEDNESDAY

Because you liked Leviathan

LOVE THROUGH A PRISM Recently Added

COSMIC PRINCESS HAGUYA! Recently Added

Fragrant Flower Book of the Month

The Apothecary Diaries

Gems for You

the ring two

DHURANDHAR Recently Added

BONES New Seasons

SULLIVAN'S CROSSING Recently Added

Only on Netflix

TRAIN DREAMS Oscar Nominee

ADOLESCENCE

THE CAPTIVE Recently Added

BOLIVAR

We Think You'll Love These

QUEER Recently Added

ANNE WITH AN E

THE SEA BEAST

LIFE ON OUR PLANET

Introducción al Aprendizaje Automático

Inversiones automatizadas


Revolut

Robo advisor

Déjasele a Robo advisor

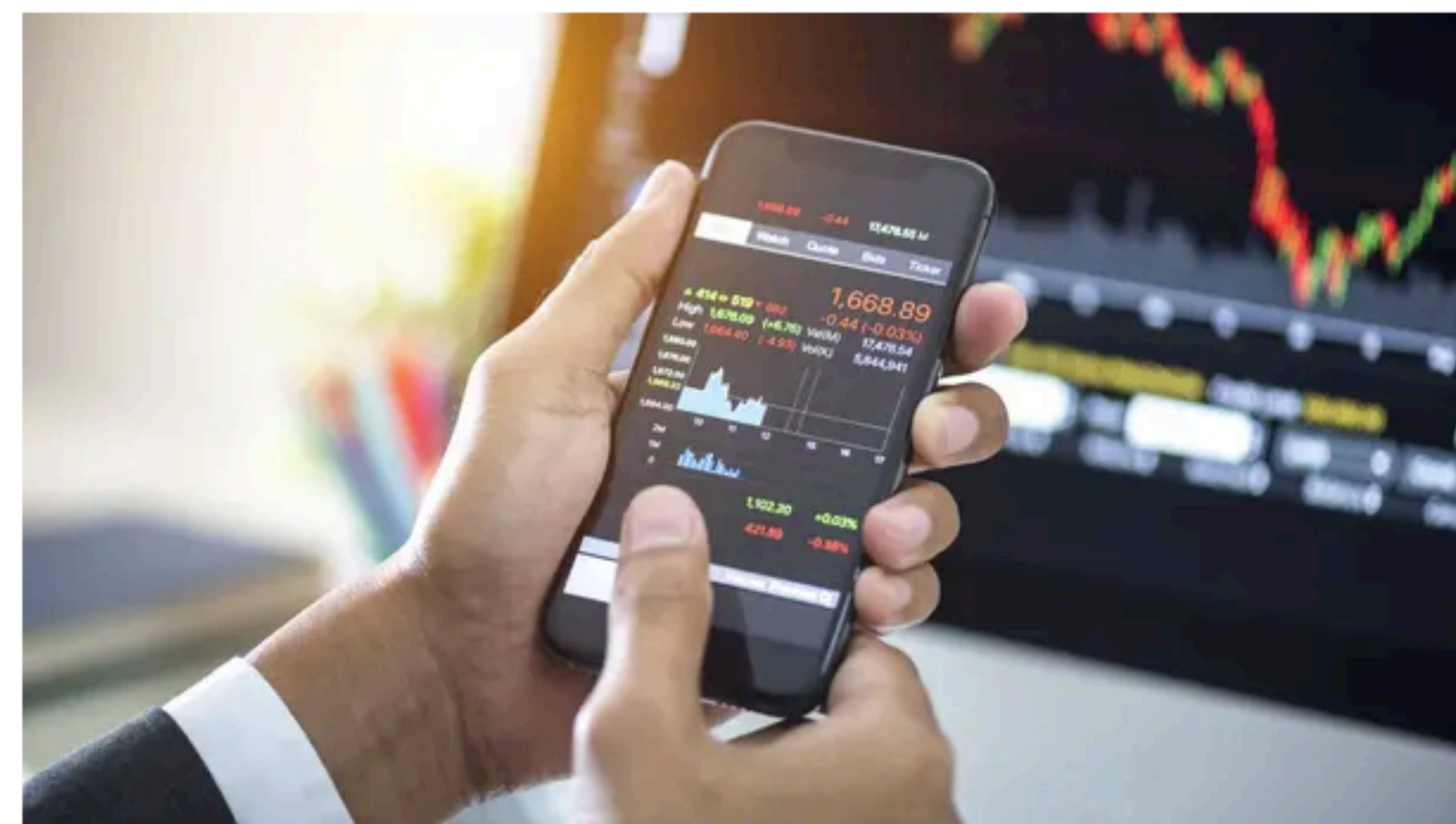
Permite que la automatización haga el trabajo duro por ti. En solo unos minutos, recibirás una cartera personalizada diseñada para ayudarte a alcanzar tus objetivos de inversión.

Invierte con robo advisor



IPROFESIONAL | FINANZAS | INVERSIÓN E INNOVACIÓN

Sacale el "jugo" a tus ahorros, sin salir de casa: por qué los roboadvisors son tendencia en EE. UU. y cómo operar desde Argentina



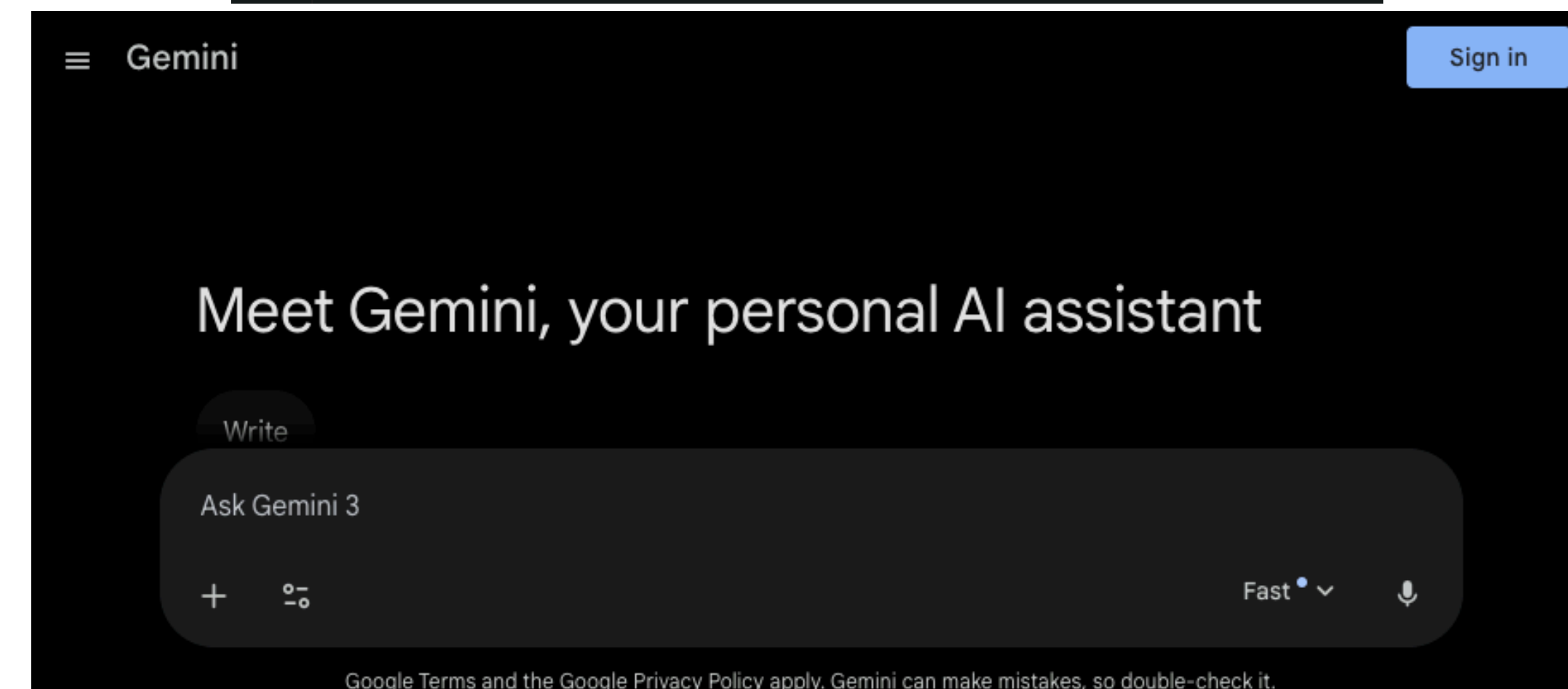
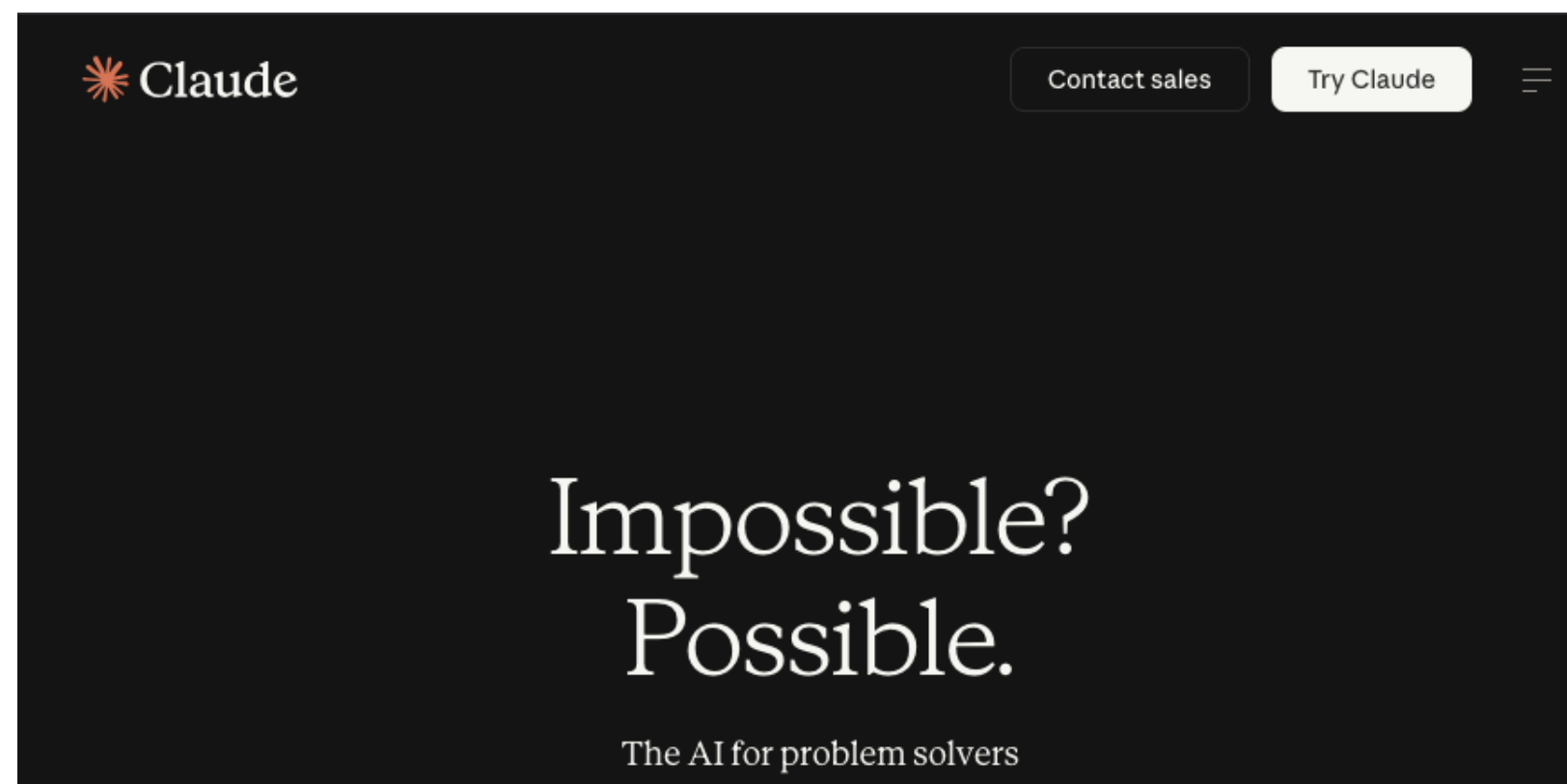
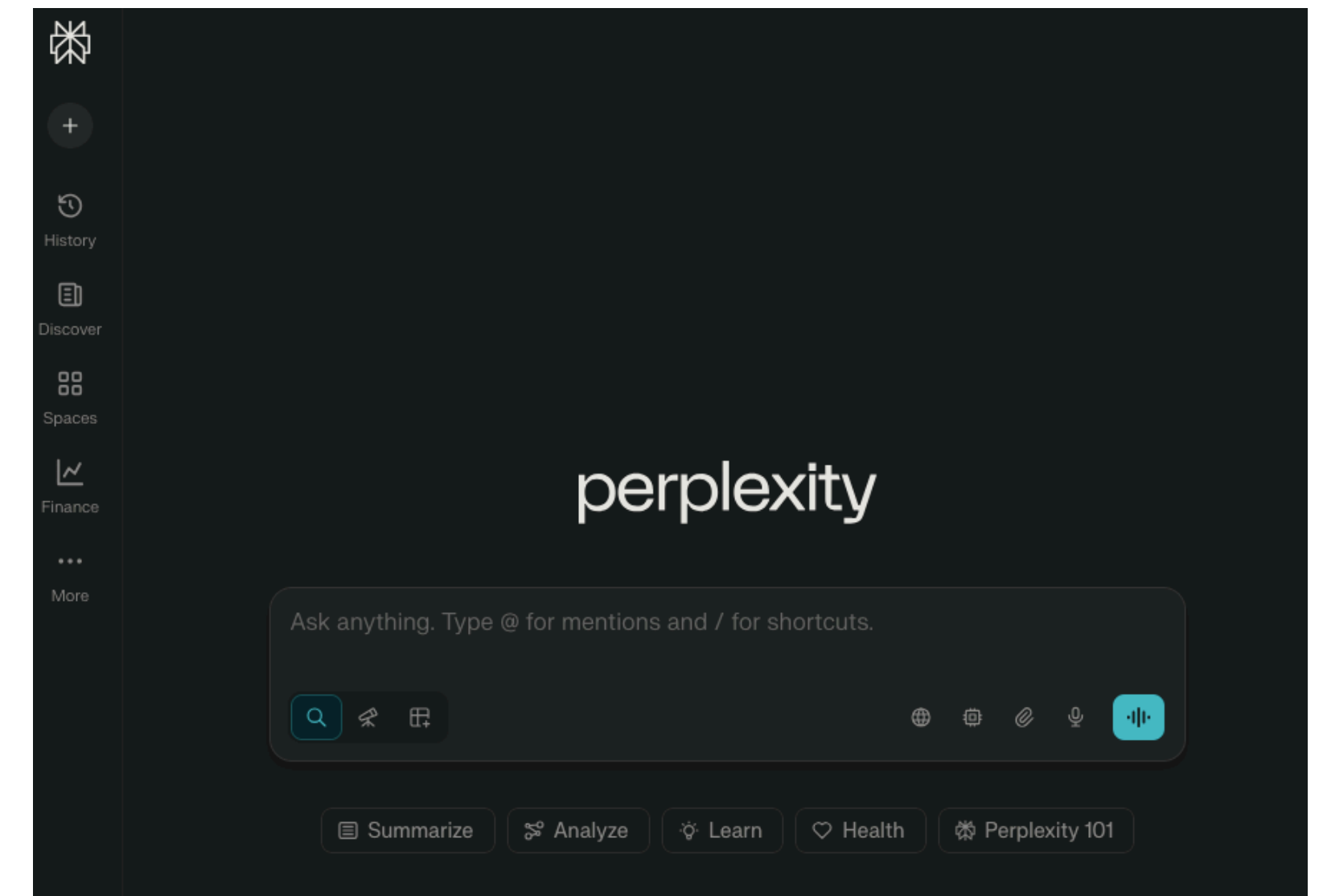
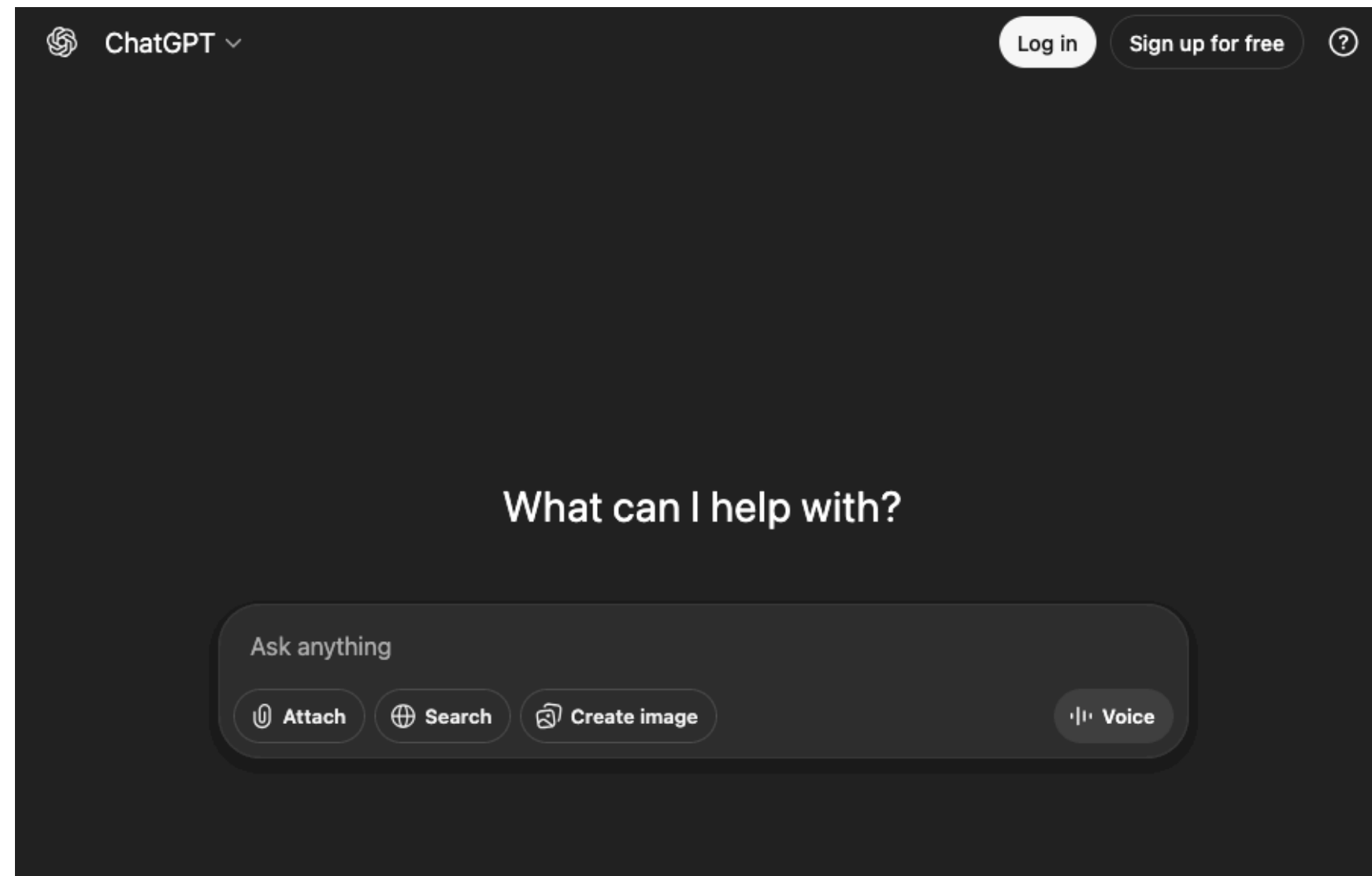
Introducción al Aprendizaje Automático

Aprender a jugar, incluso mejor que el humano



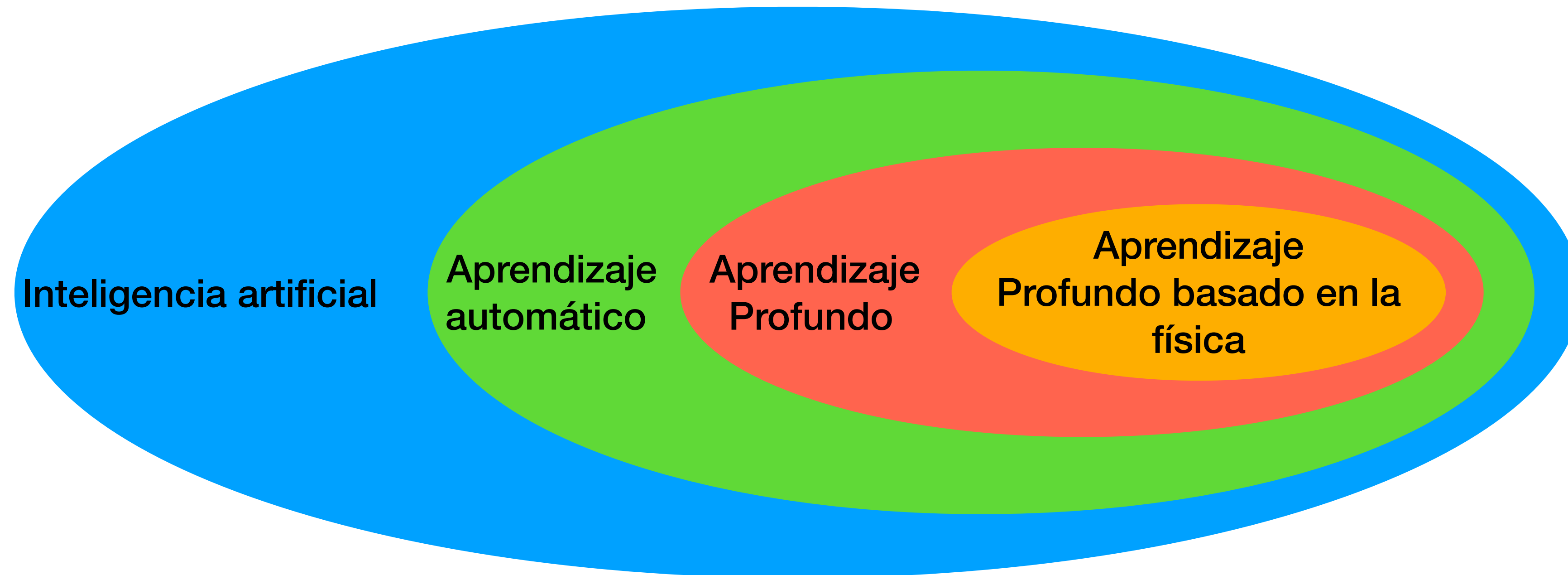
Introducción al Aprendizaje Automático

LLMS revolucionando diversos campos de la ciencia y de la vida



Introducción al Aprendizaje Automático

AI, ML, DL, PBDL



Comienza en 1940, con la aparición de las computadoras. Turing y von Neumann comienzan a preguntarse: Y si podemos formalizar computaciones, podríamos formalizar “el pensamiento”?

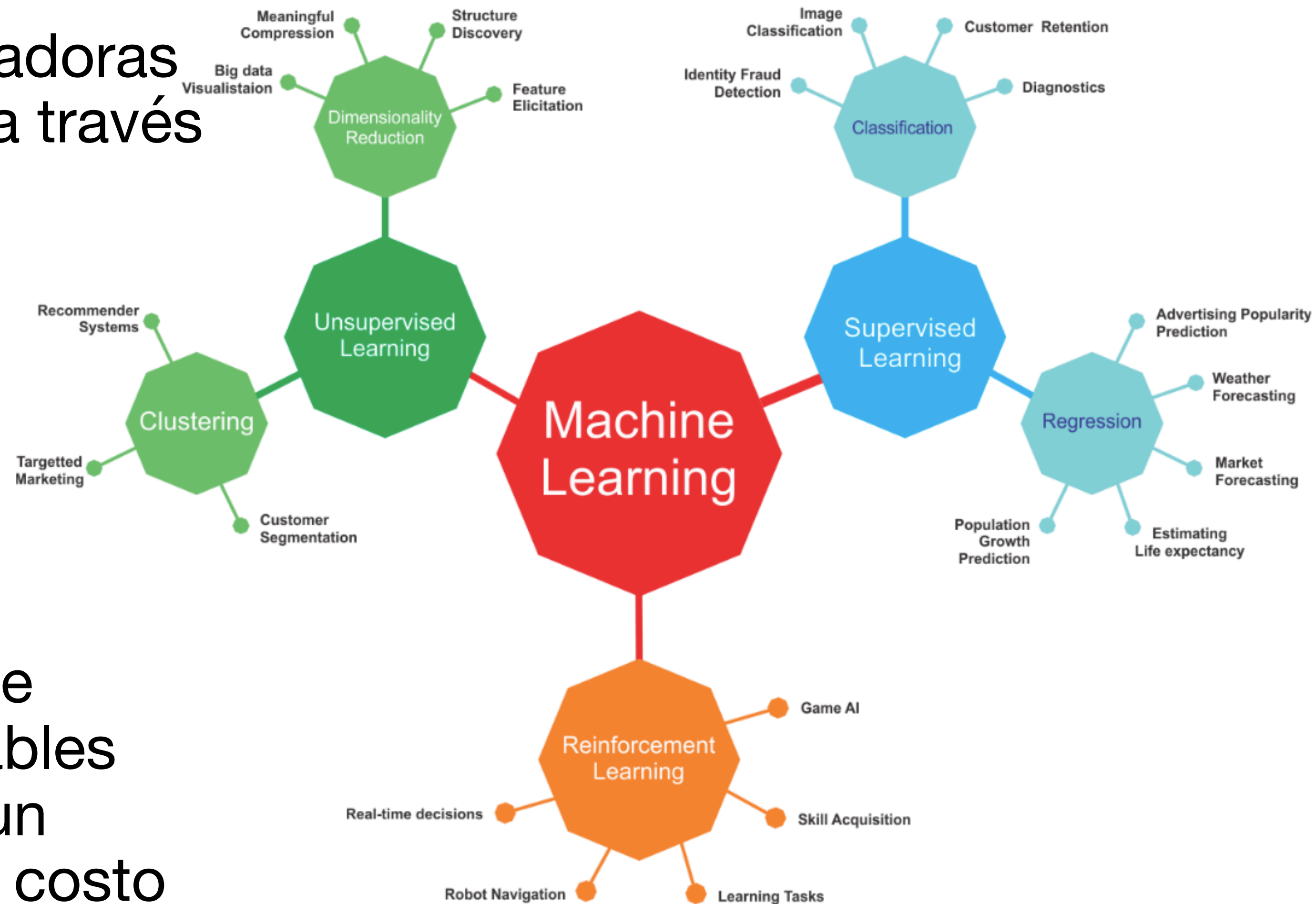
Introducción al Aprendizaje Automático

Aprendizaje automático: Machine learning

Subcampo de la IA que permite computadoras aprender a realizar una tarea específica a través de la experiencia

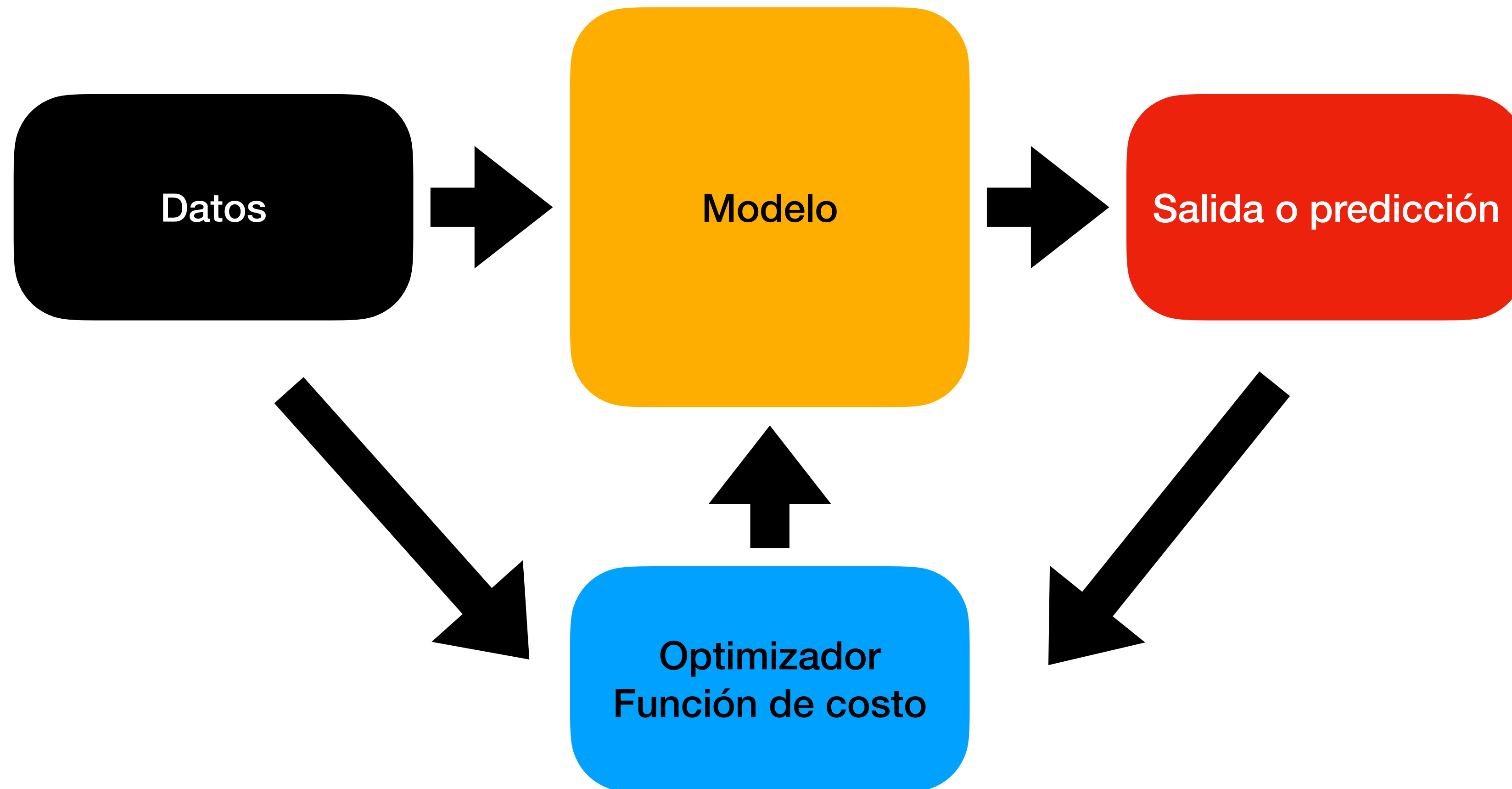
- Aprende a partir de los **datos**
- Generaliza a nuevos casos
- Automatiza decisiones o predicciones

Un modelo de ML busca una función que relacione variables de **entrada** con variables de **salida**, mediante la optimización de un criterio de error, denominado función de costo o pérdida



Introducción al Aprendizaje Automático

Aprendizaje automático: Machine learning



Tipos principales:

- Supervisado
- No supervisado
- Por refuerzo

Otros tipos:

- Semi-supervisado
- Auto-supervisado

Introducción al aprendizaje automático

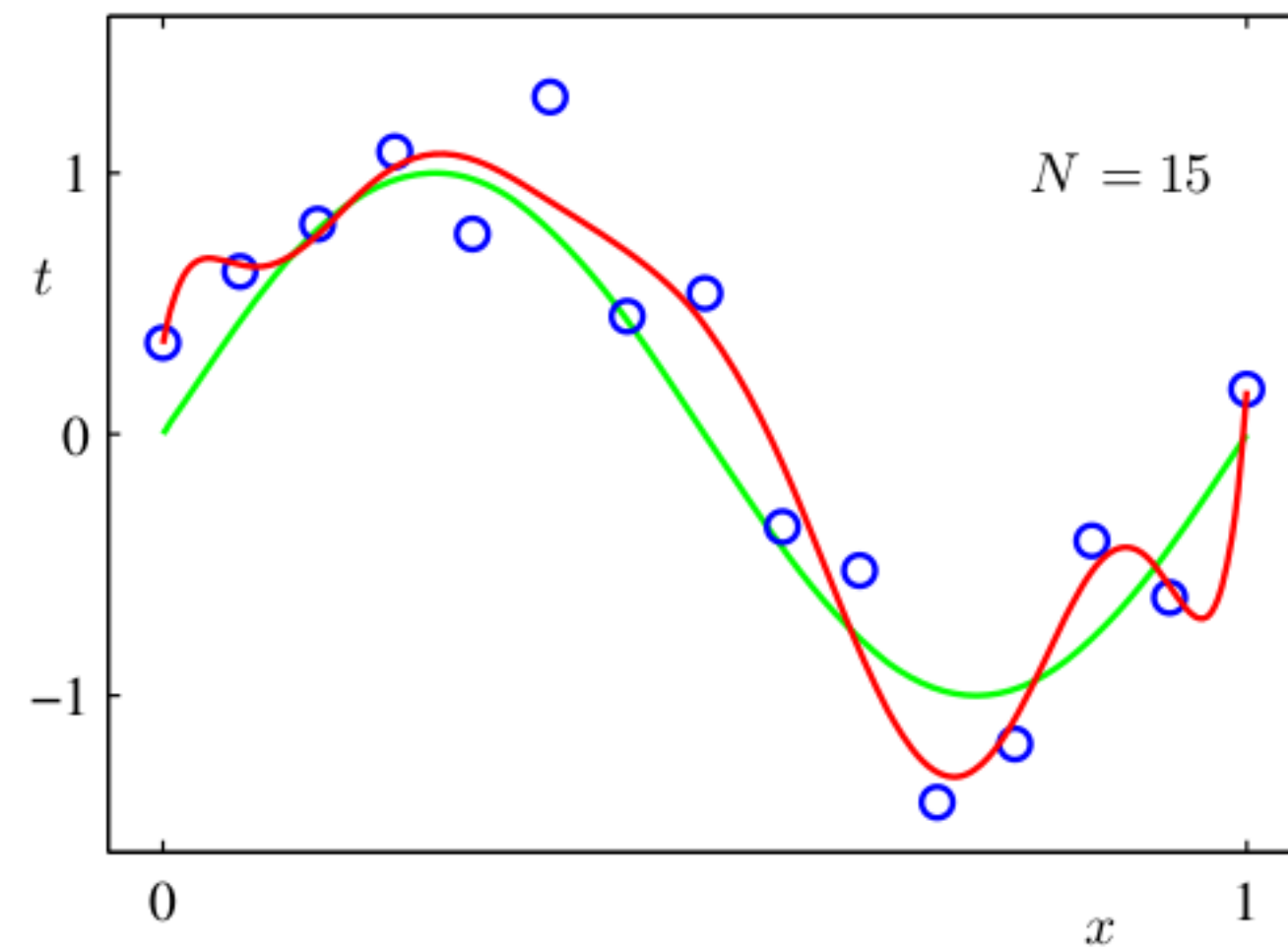
Aprendizaje supervisado

- La tarea a aprender es conocida
- La tarea está descrita por un conjunto de **ejemplos de entrada** y un **valor asociado** a cada entrada (el valor esperado).
- El objetivo es predecir la **salida** que corresponde a una dada **entrada**.
- El modelo **generaliza** los datos. Aprende correlaciones en ellos, por lo que después puede **predecir** en ejemplos que no ha visto.
- Dos tareas principales: **Clasificación**, en donde la salida a predecir es un conjunto de *valores discretos*; **Regresión**, en donde la salida a predecir es un *valor continuo*.
- Otros objetivos como ranking, multitiqueta, estimación de probabilidades, series temporales, etc.

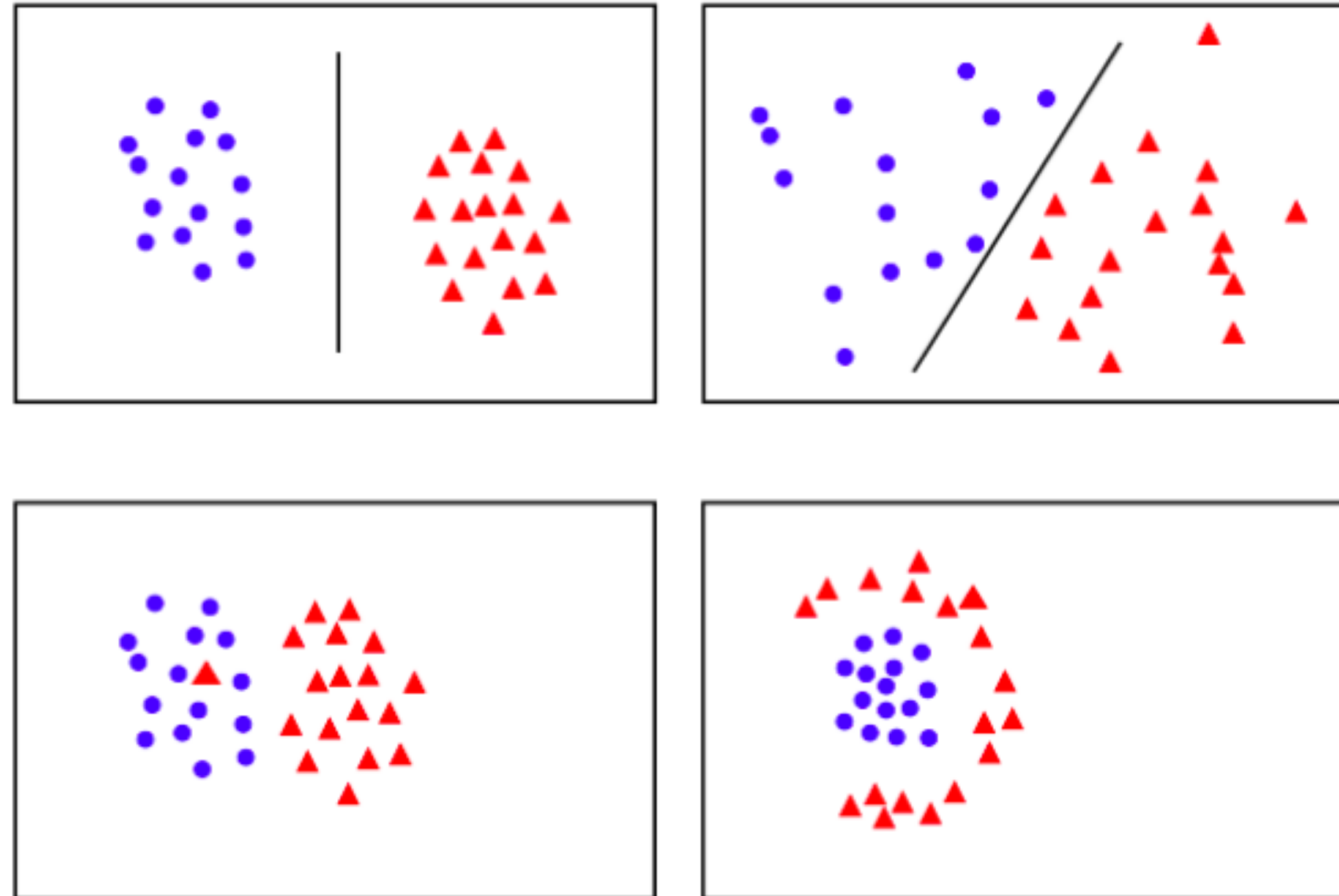
Introducción al aprendizaje automático

Aprendizaje supervisado

Regresión



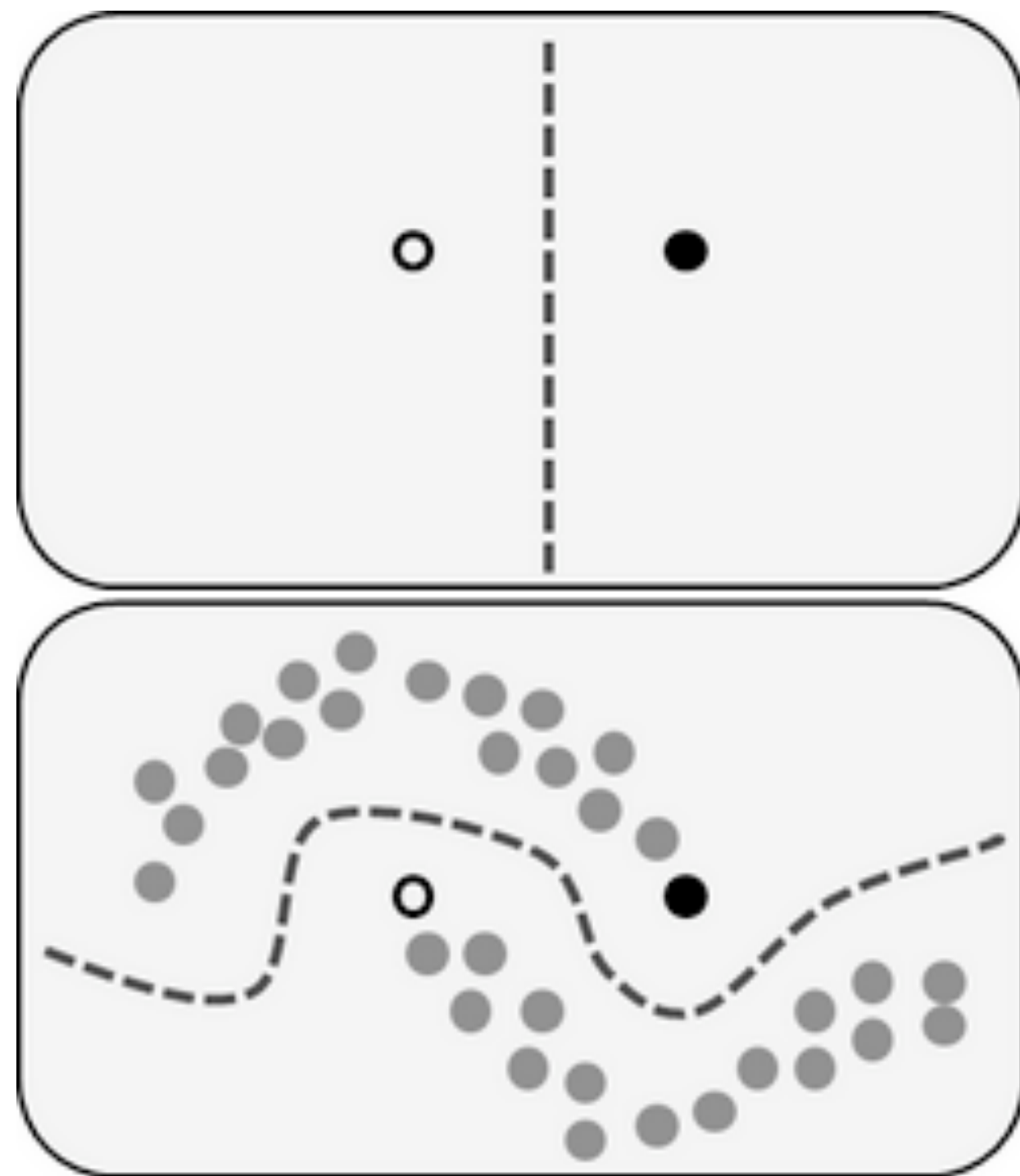
Clasificación



Introducción al aprendizaje automático

Aprendizaje supervisado - variantes

Semi supervisado



Algunas etiquetas presentes

Auto-supervisado

El perro tiene una **cola**

El perro tiene una **boca**

El perro tiene cuatro **patas**

El perro tiene dos **orejas**

El perro tiene dos **ojos**

Introducción al aprendizaje automático

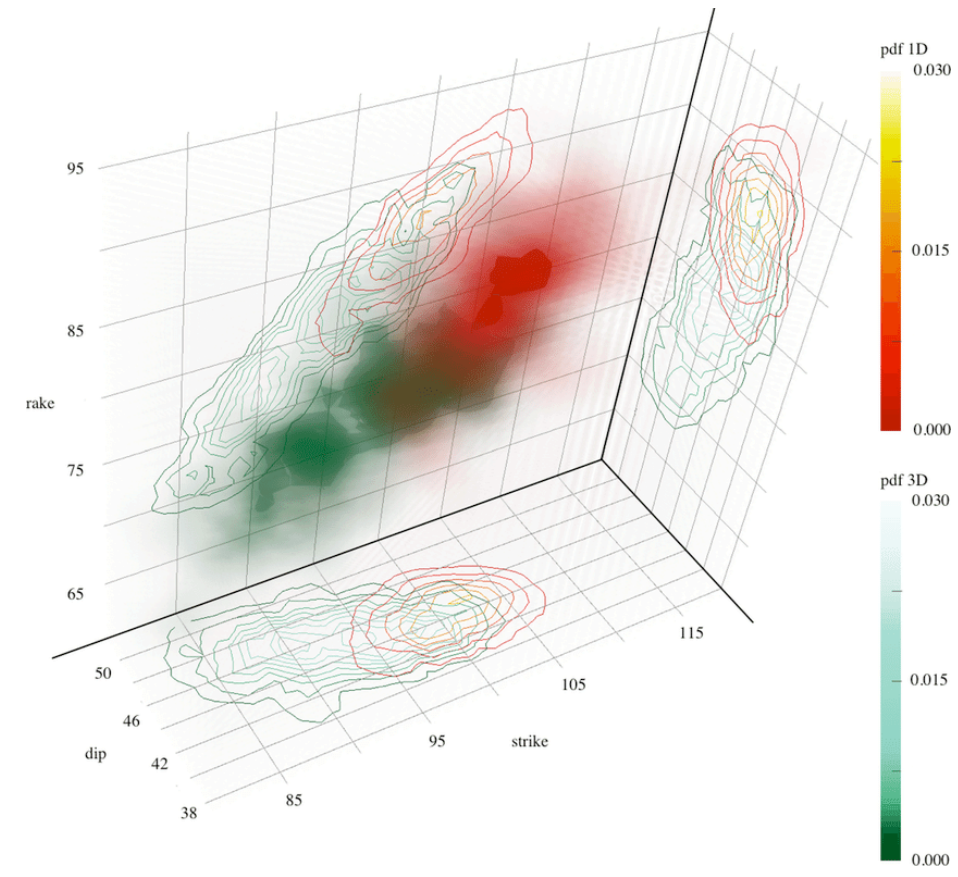
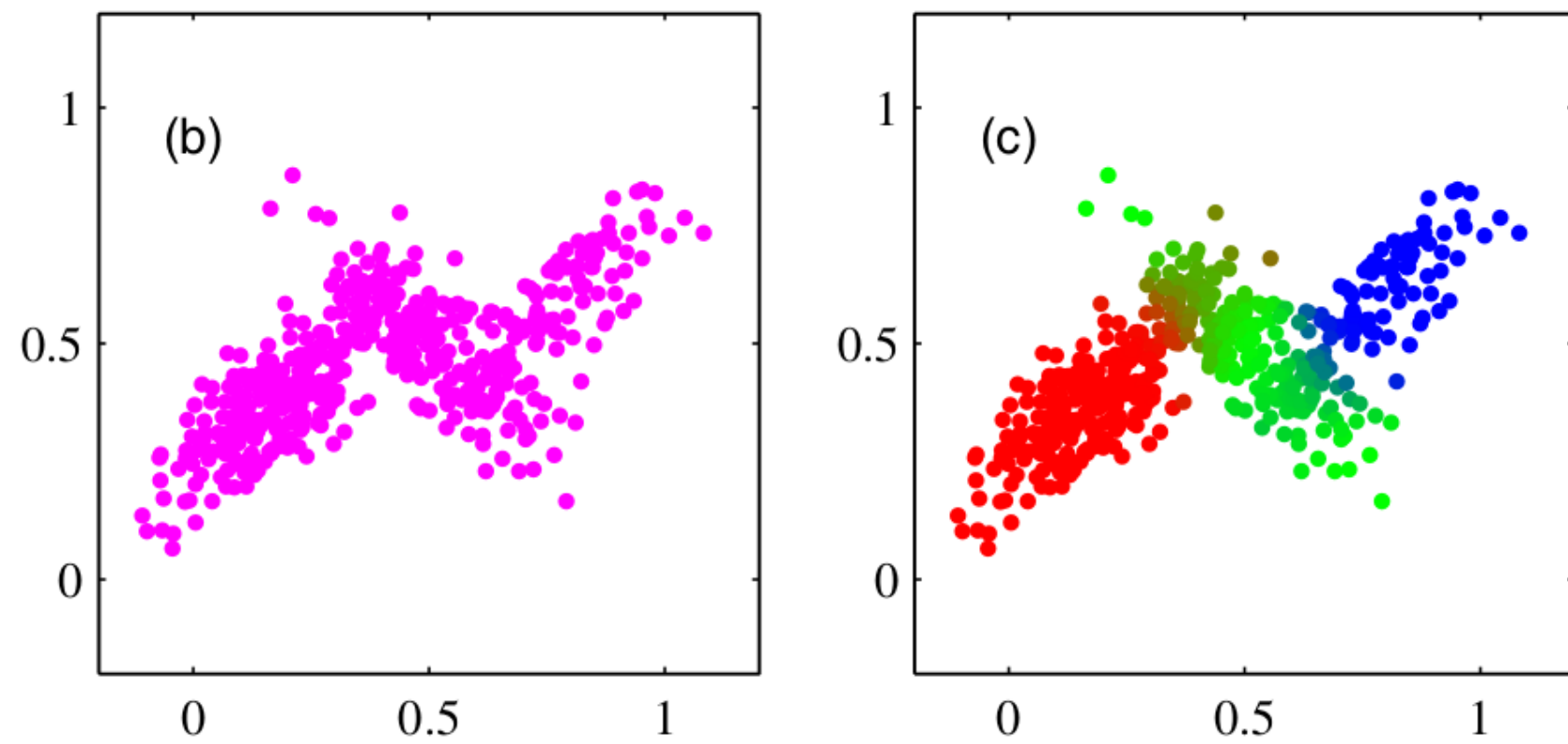
Aprendizaje no supervisado

- La tarea a aprender es desconocida (exploración y descubrimiento)
- La tarea está descrita únicamente por un conjunto de **ejemplos de entrada** (no hay etiquetas).
- Varios objetivos posibles:
 - *agrupar* diferentes conjuntos de datos según patrones (clustering). Ejemplos: kNN, HCA, DBSCAN, etc.
 - *transformar* los datos para simplificar representación (reducción de dimensionalidad). Ejemplos: PCA, t-SNE, UMAP, etc.
 - *detección de anomalías*: SVM, Autoencoders, etc.
 - *Estimar* densidades en los datos para comprender o generar datos. Ejemplo: KDE, GMM, etc.

Introducción al aprendizaje automático

Aprendizaje no supervisado

clustering

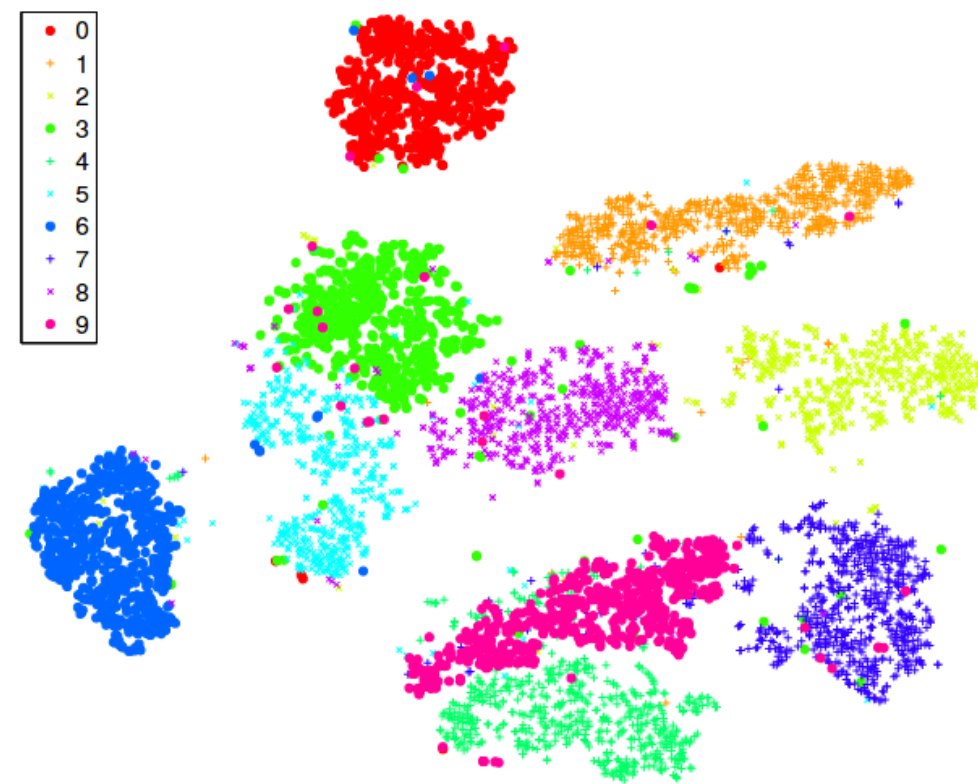
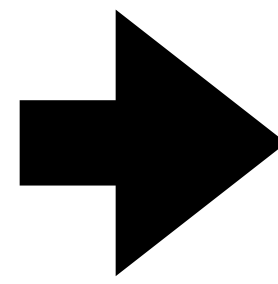
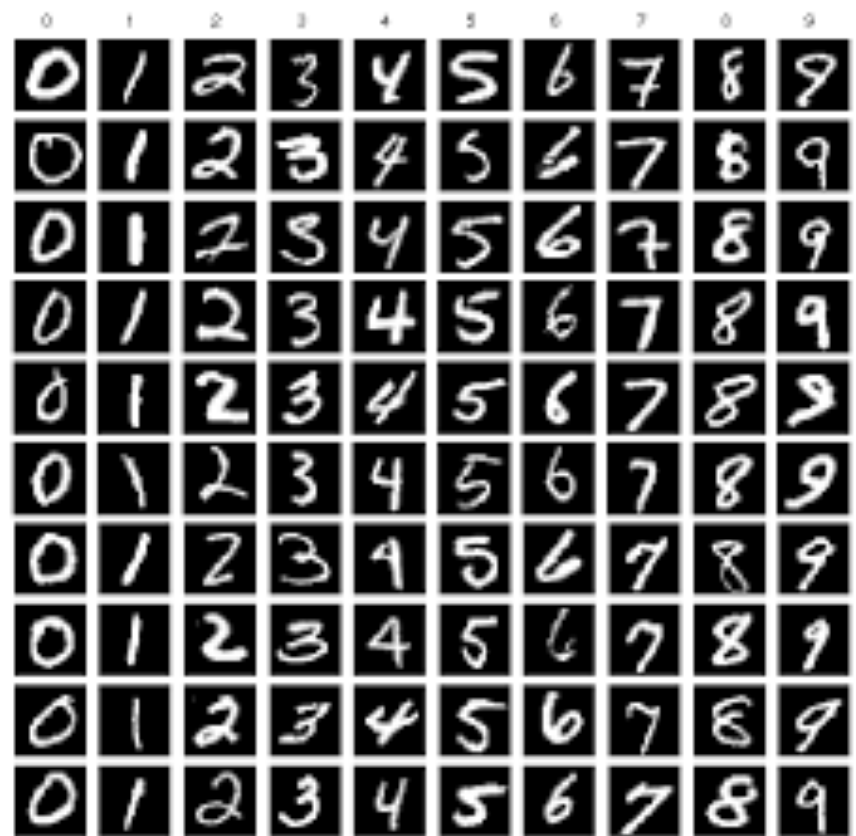


Estimación de densidad

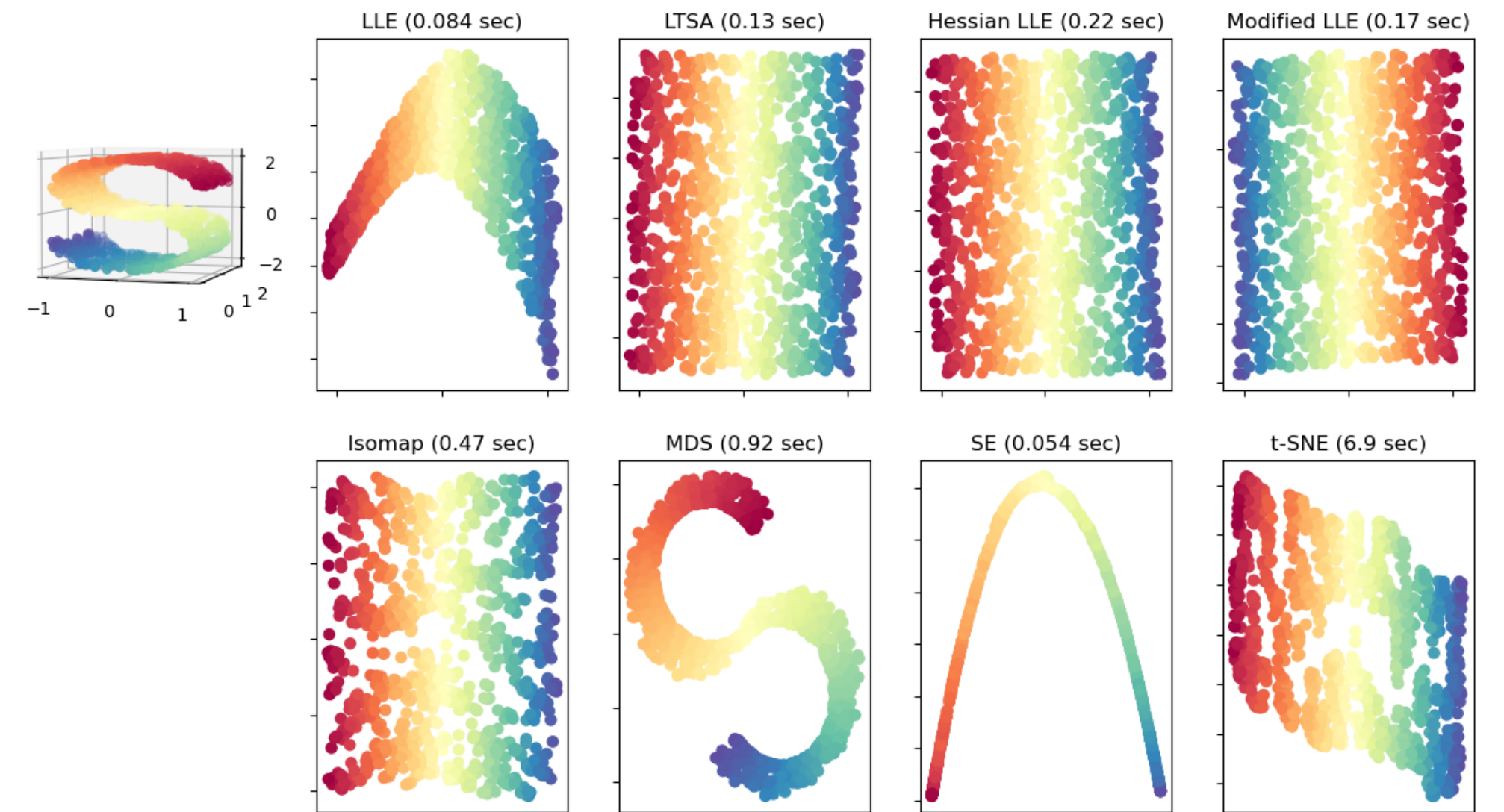
Transformaciones

Manifold Learning with 1000 points, 10 neighbors

Espacios latentes



(a) Visualization by t-SNE.



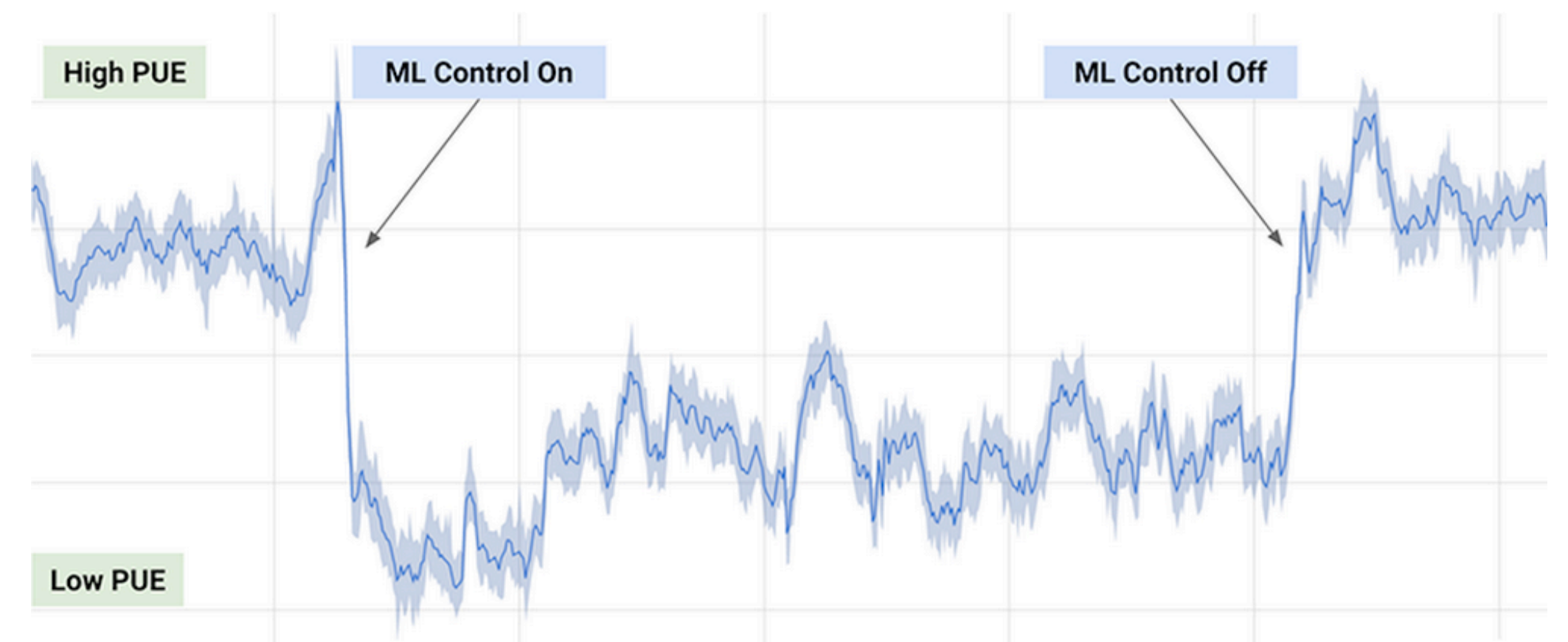
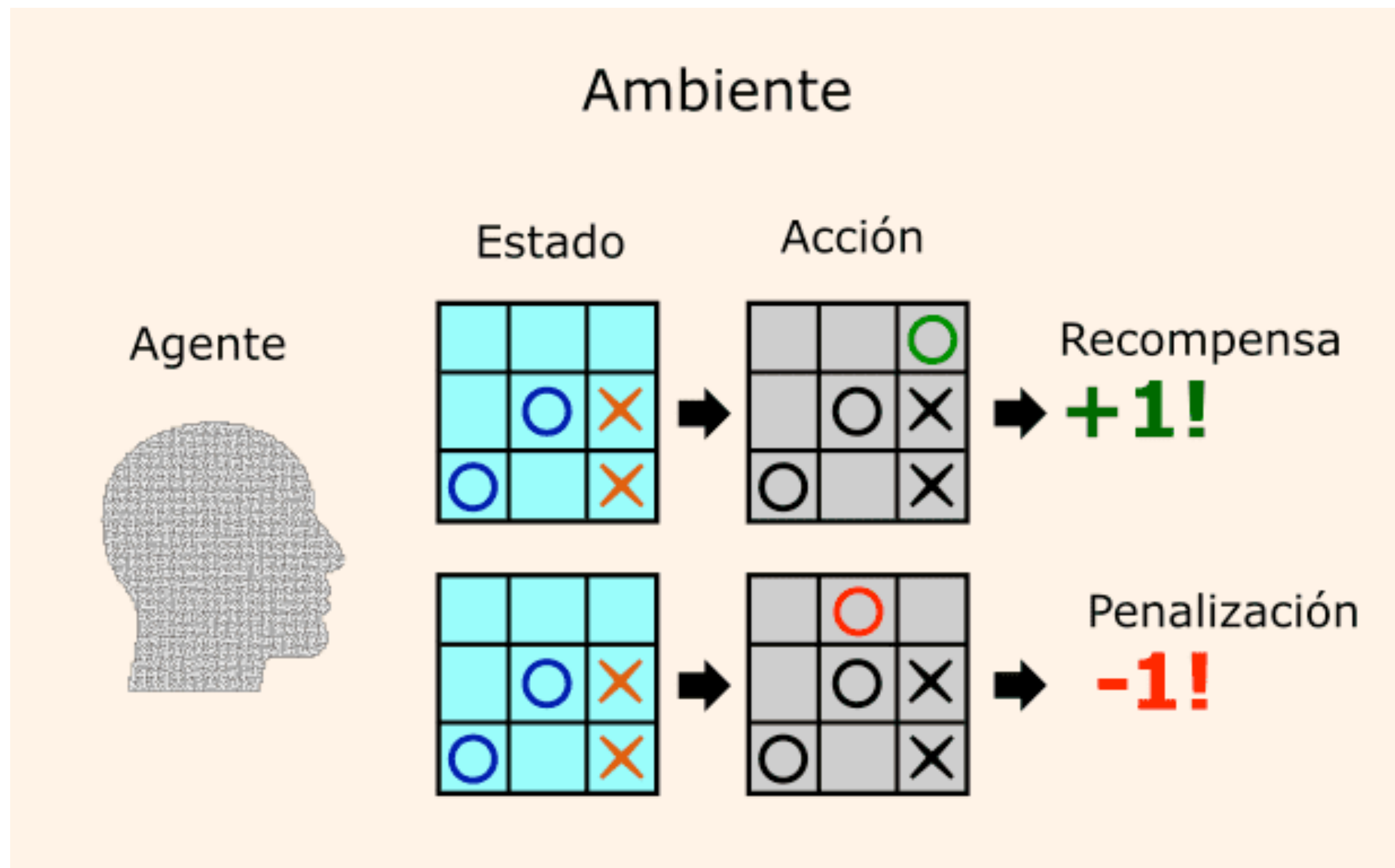
Introducción al aprendizaje automático

Aprendizaje por refuerzo

- Especializado en tareas de resolución de problemas.
- Un **agente** tiene un conjunto de **acciones** disponibles para resolver una **tarea** compleja en un **entorno**.
- El agente **explora** el entorno utilizando las acciones
- El entorno retroalimentación al agente en función de su desempeño (este es el refuerzo)
- Aprende a asociar un valor a las acciones para un estado del entorno según su ventaja para alcanzar el objetivo de la tarea.
- Después de aprender, el modelo se utiliza para tomar decisiones en el entorno

Introducción al aprendizaje automático

Aprendizaje por refuerzo



Resultados del algoritmo de DeepMind sobre el consumo energético de los centros de datos de Google (Power Usage Effectiveness)

Aprender automáticamente

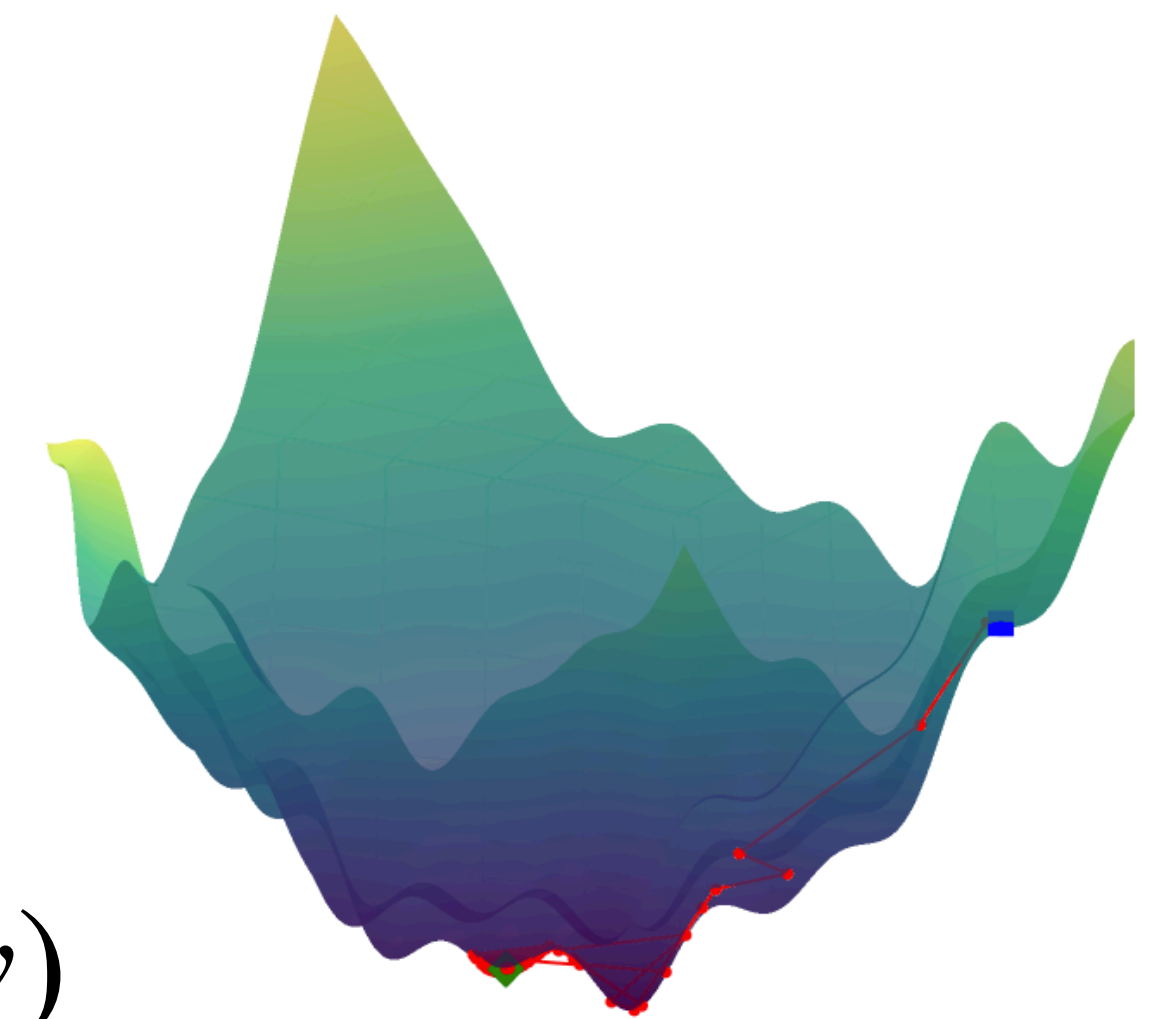
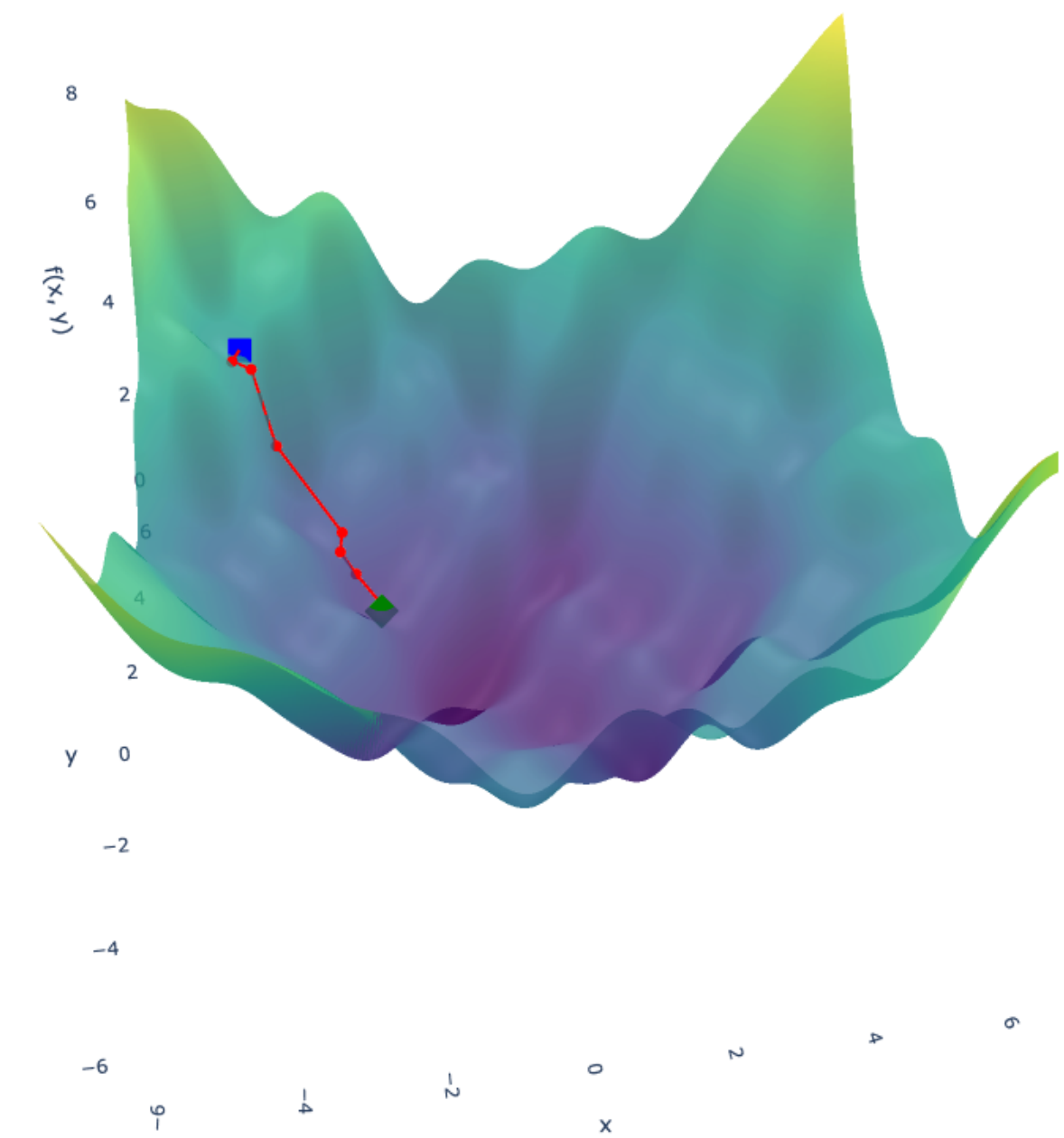
Optimización y aprendizaje

- Un problema típico de ML se puede escribir como

$$\mathcal{L}(w) = \sum_{i=1}^N l(y_i, f_w(x_i)) + \lambda R(w)$$

- w parámetros del modelo
- l **función de costo** del par de datos (x_i, y_i)
- $f_w(x_i)$ predicción del modelo
- $\lambda R(w)$ término de regularización para controlar sobreajuste. No depende de los datos, sólo de los parámetros.

- Aprender significa encontrar w^* tal que $w^* = \arg \min_w \mathcal{L}(w)$



Aprender automáticamente

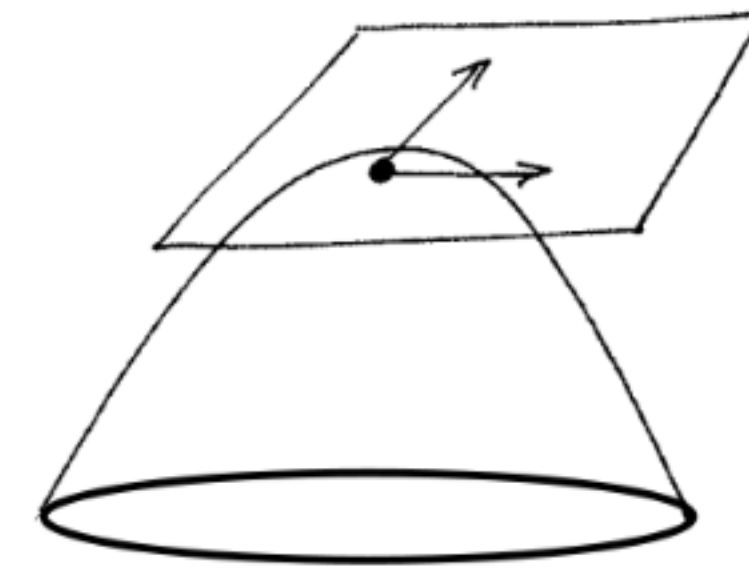
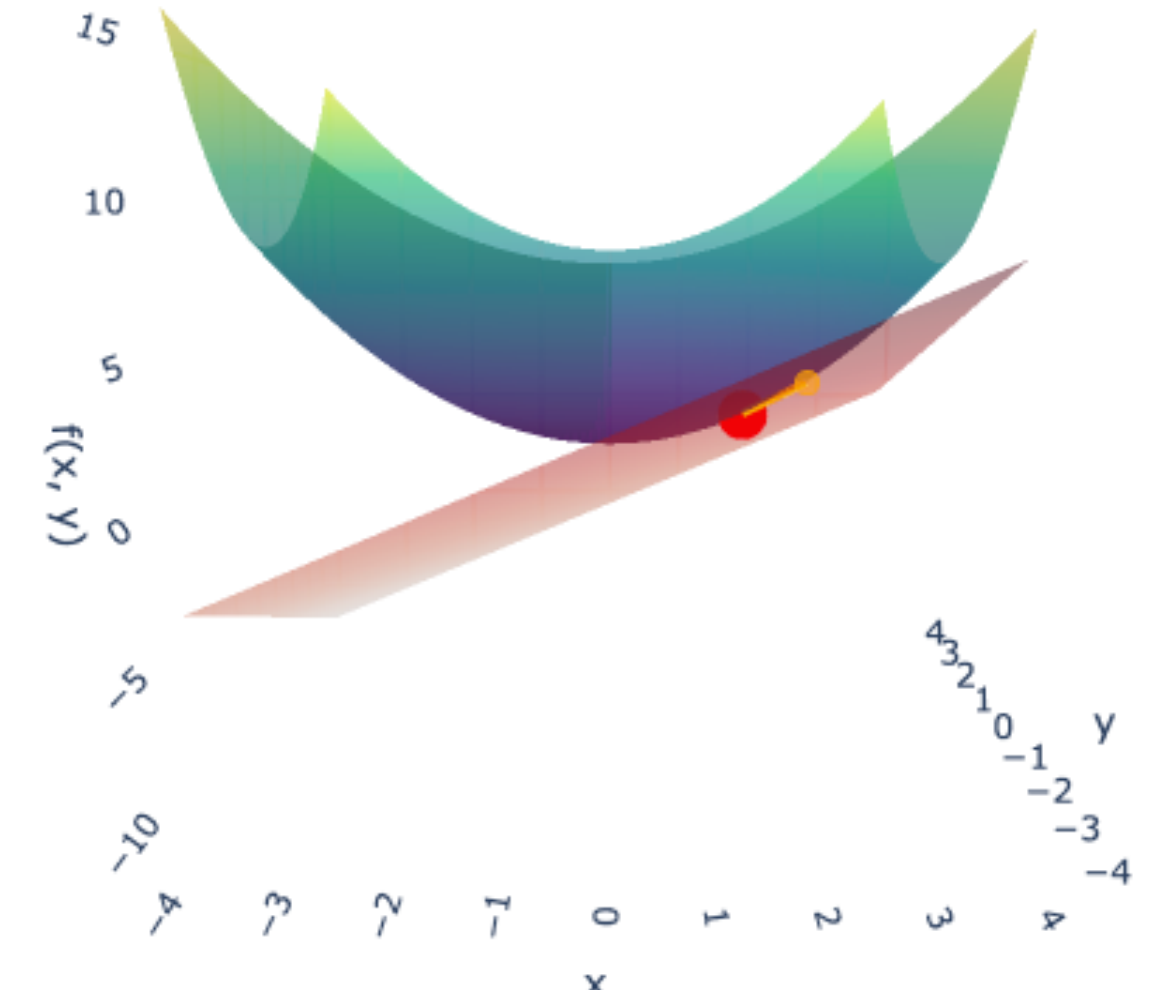
Optimización

- En funciones nD , $f: \mathbb{R}^n \rightarrow \mathbb{R}$,
- f tiene un punto estacionario en \vec{x}_0 cuando

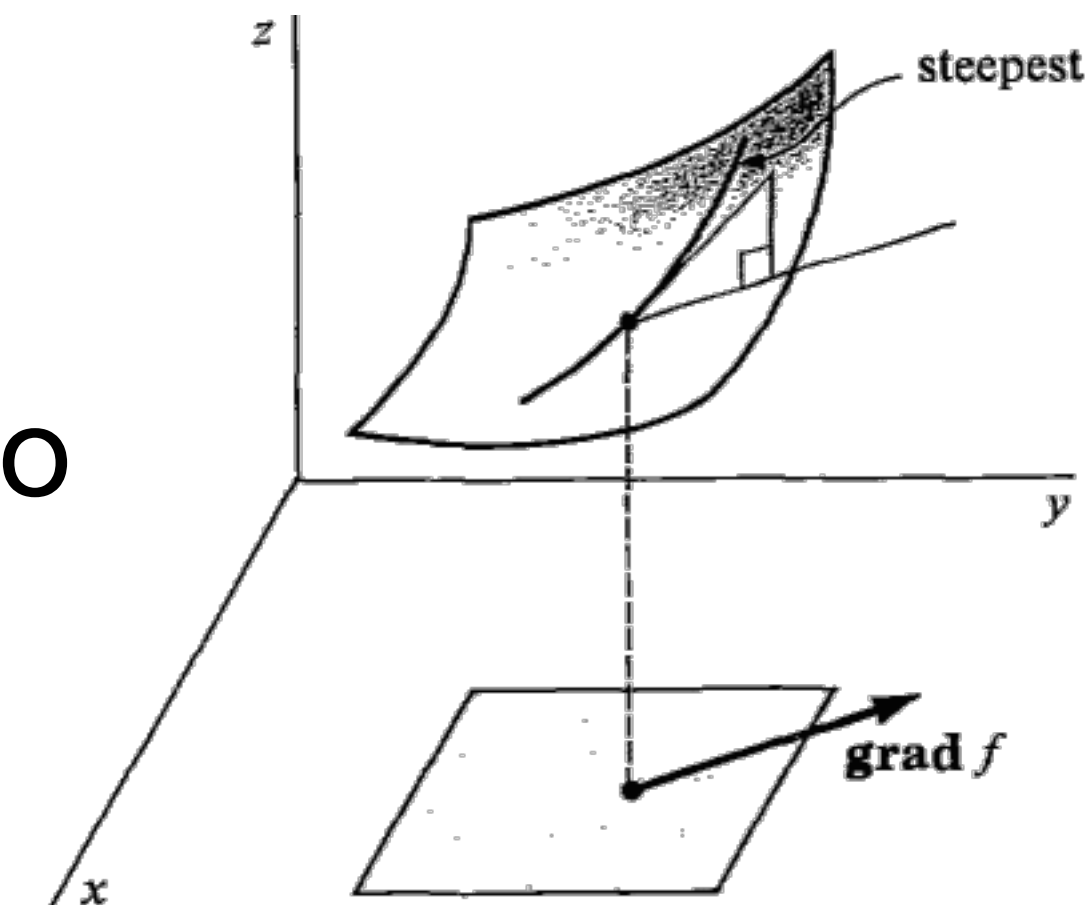
$$\nabla f(\vec{x}_0) = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right) = 0$$

El tipo de punto estacionario está definido por la matriz hessian.

- El gradiente en un punto da la dirección de máximo crecimiento



$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)^T = 0$$



Aprender automáticamente

Optimización

- Idea general en los métodos de descenso por el gradiente iterativos:

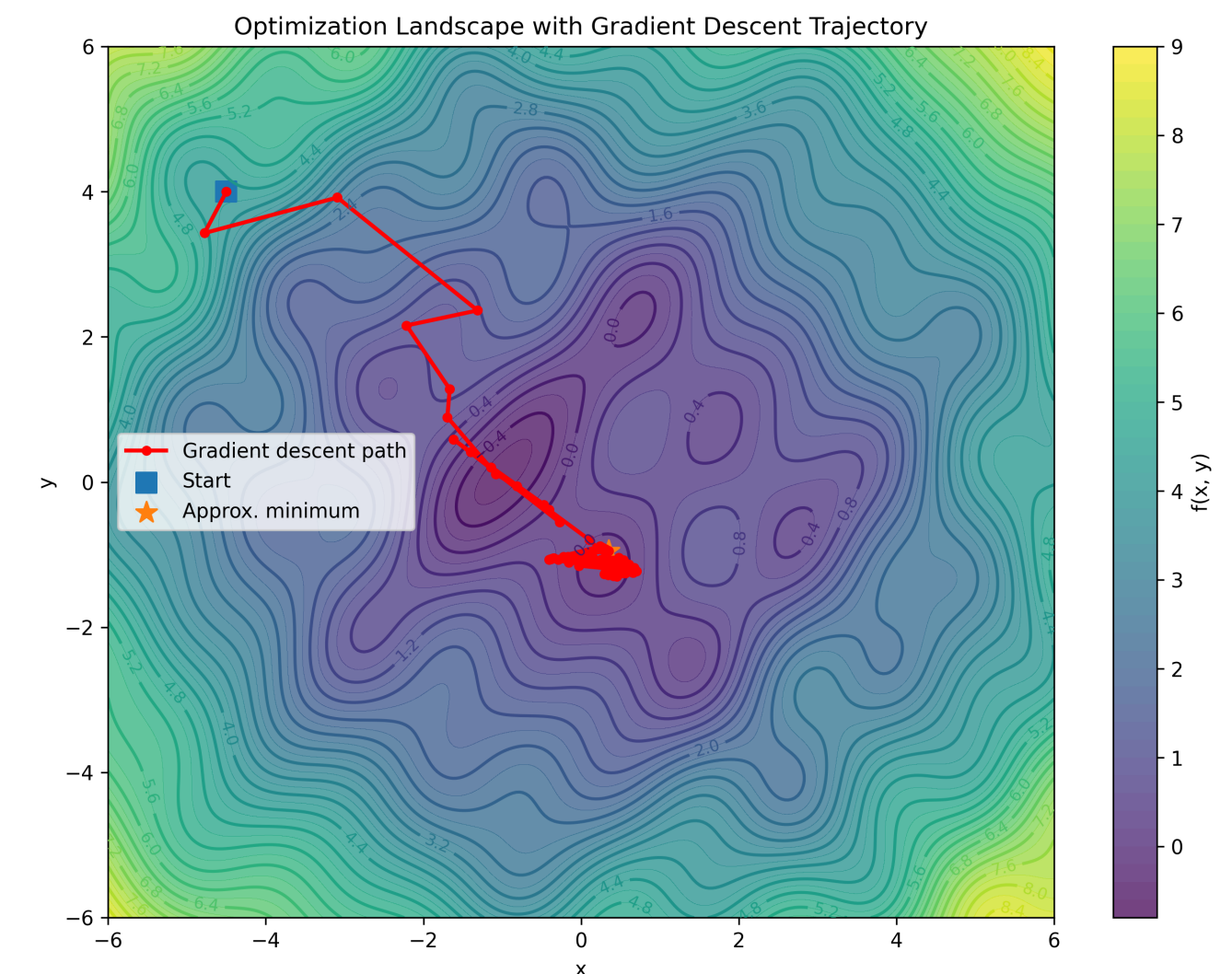
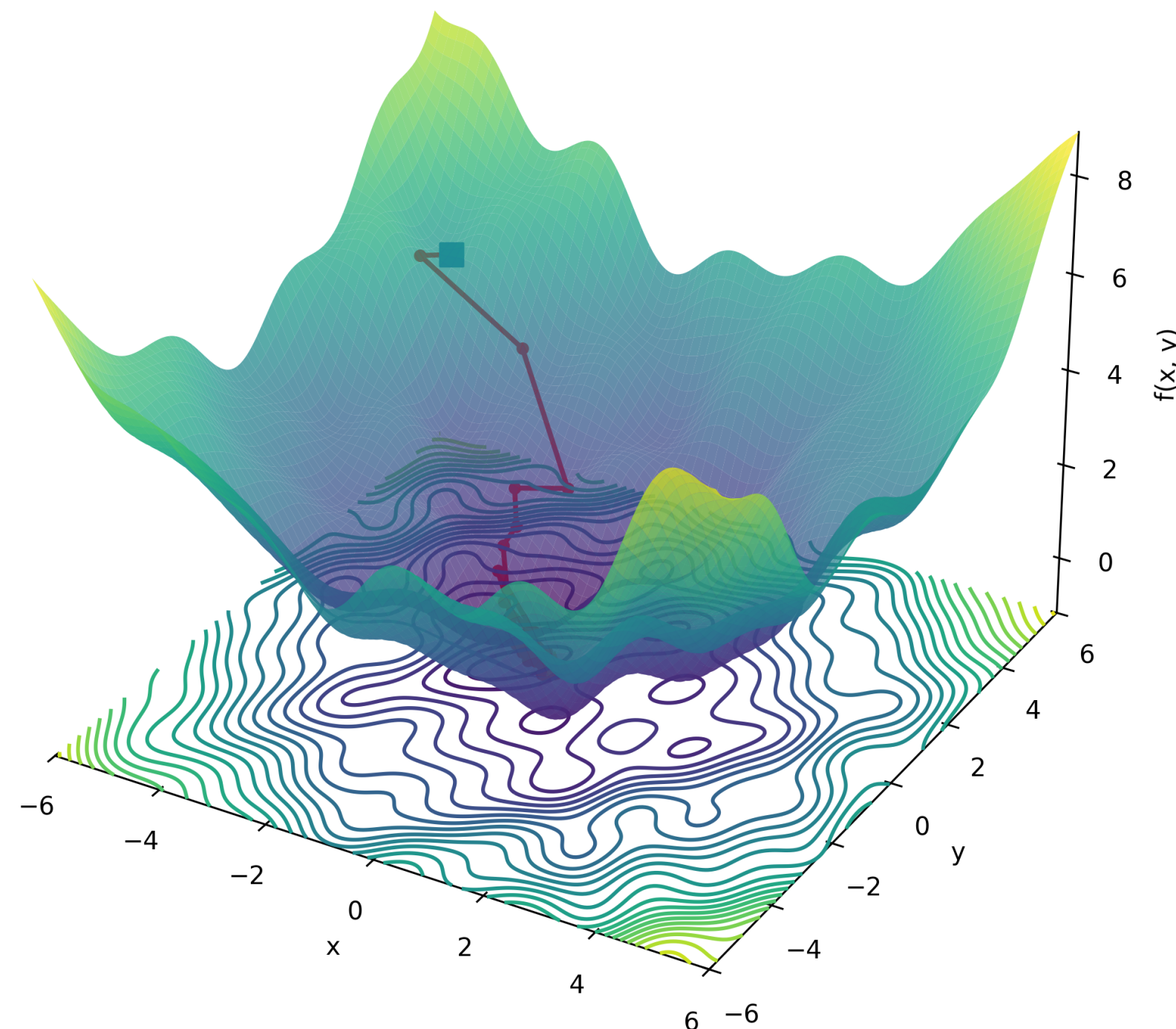
A partir de algún $w^{(0)}$ a $i = 0$, avanzar en la dirección opuesta al gradiente con un paso $\Delta w^{(i)}$

$$w^{(i+1)} \leftarrow w^{(i)} + \Delta w^{(i)},$$

$$\text{Donde } \Delta w^{(i)} = -\eta_i \nabla \mathcal{L}(w^{(i)})$$

η_i : tasa de aprendizaje

Ejemplo interactivo: landscape.py



PyTorch

Introducción práctica en Colab

Intro a Clases en Python

intro_pytorch.ipynb - Parte 1

PyTorch

Introducción práctica en Colab

Práctica 1 - Hasta Ejercicio 14

Funciones de costo

Introducción

Una función de costo $\mathcal{L}(\theta)$ es una función escalar que mide el error o la discrepancia entre la predicción de un modelo parametrizado por los parámetros θ y la información disponible.

Entrenar un modelo de ML es resolver un problema de optimización típicamente no lineal y de alta dimensión. Generalmente se busca minimizar (o maximizar) $\mathcal{L}(\theta)$:

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta)$$

Definir la función de costo resulta **central** en el diseño de un algoritmo de aprendizaje, ya que define qué significa **aprender**.

Funciones de costo: Regresión

Error cuadrático medio (MSE)

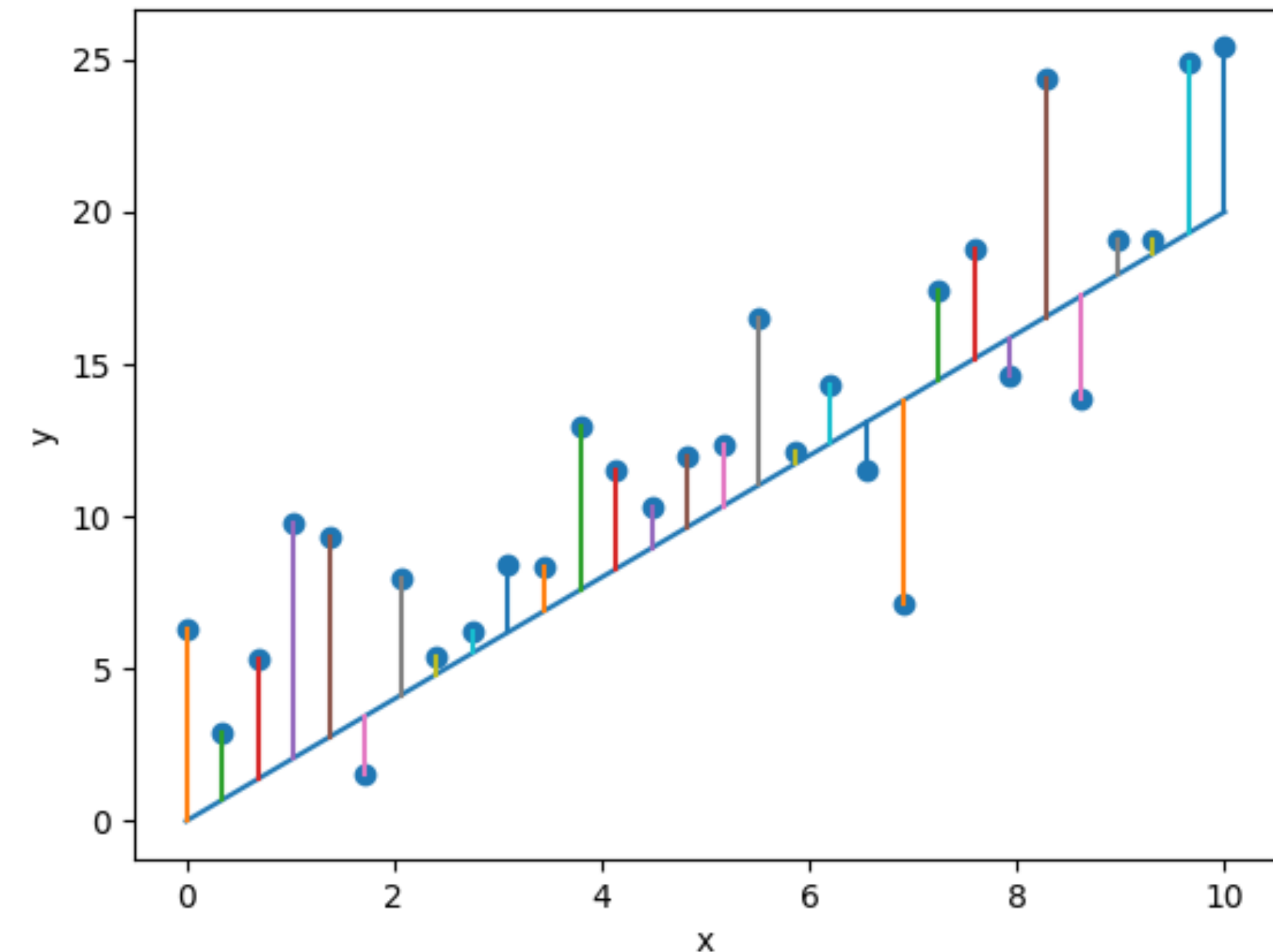
$$\mathcal{L}_{\text{MSE}}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Error absoluto medio (MAE)

$$\mathcal{L}_{\text{MAE}}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

Mean Squared Error (MSE) = 15.96

Mean Absolute Error (MAE) = 3.31



Funciones de costo: Regresión

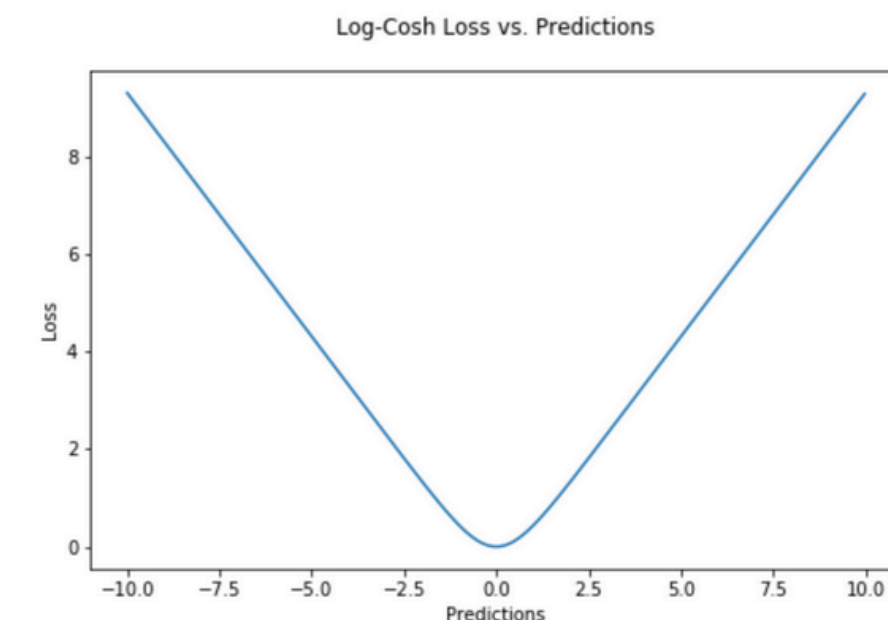
Costo de Hubber

$$l_{\text{Huber}}(y_i, \hat{y}_i) = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2, & \text{si } |y_i - \hat{y}_i| \leq \delta, \\ \delta |y_i - \hat{y}_i| - \frac{1}{2}\delta^2, & \text{si } |y_i - \hat{y}_i| > \delta, \end{cases}$$

<https://docs.pytorch.org/docs/stable/generated/torch.nn.HuberLoss.html>

Costo log-cosh

$$\mathcal{L}_{\text{log-cosh}}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N \log(\cosh(y_i - \hat{y}_i))$$



Funciones de costo: Clasificación

Costo 0-1

$$l_{0-1}(y_i, \hat{y}_i) = \begin{cases} 0, & \text{si } y_i = \hat{y}_i, \\ 1, & \text{si } y_i \neq \hat{y}_i \end{cases} = \mathbb{1}(y_i \neq \hat{y}_i)$$

$$\mathcal{L}_{0-1}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(y_i \neq \hat{y}_i)$$

Entropía cruzada binaria (log-loss o pérdida logística)

$$\mathcal{L}_{\log}(y, \hat{p}) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)]$$

Minimizar la entropía cruzada equivale a maximizar la log-verosimilitud bajo hipótesis $Y_i \sim \text{Bernoulli}(p_i)$

$$P(Y = y_i) = p_i^{y_i} (1 - p_i)^{1 - y_i}$$

Funciones de costo: Clasificación

Entropía cruzada categórica (K clases)

Softmax(z_k)

$$\mathcal{L}_{\text{CE}}(y, p) = - \sum_{k=1}^K y_k \log(p_k), \quad p_k = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}}$$

Entropía cruzada binaria (multiclase, con K clases)

$$\mathcal{L}_{\text{BCE}}(y, \hat{p}) = - \sum_{k=1}^K [y_k \log(\hat{p}_k) + (1 - y_k) \log(1 - \hat{p}_k)]$$

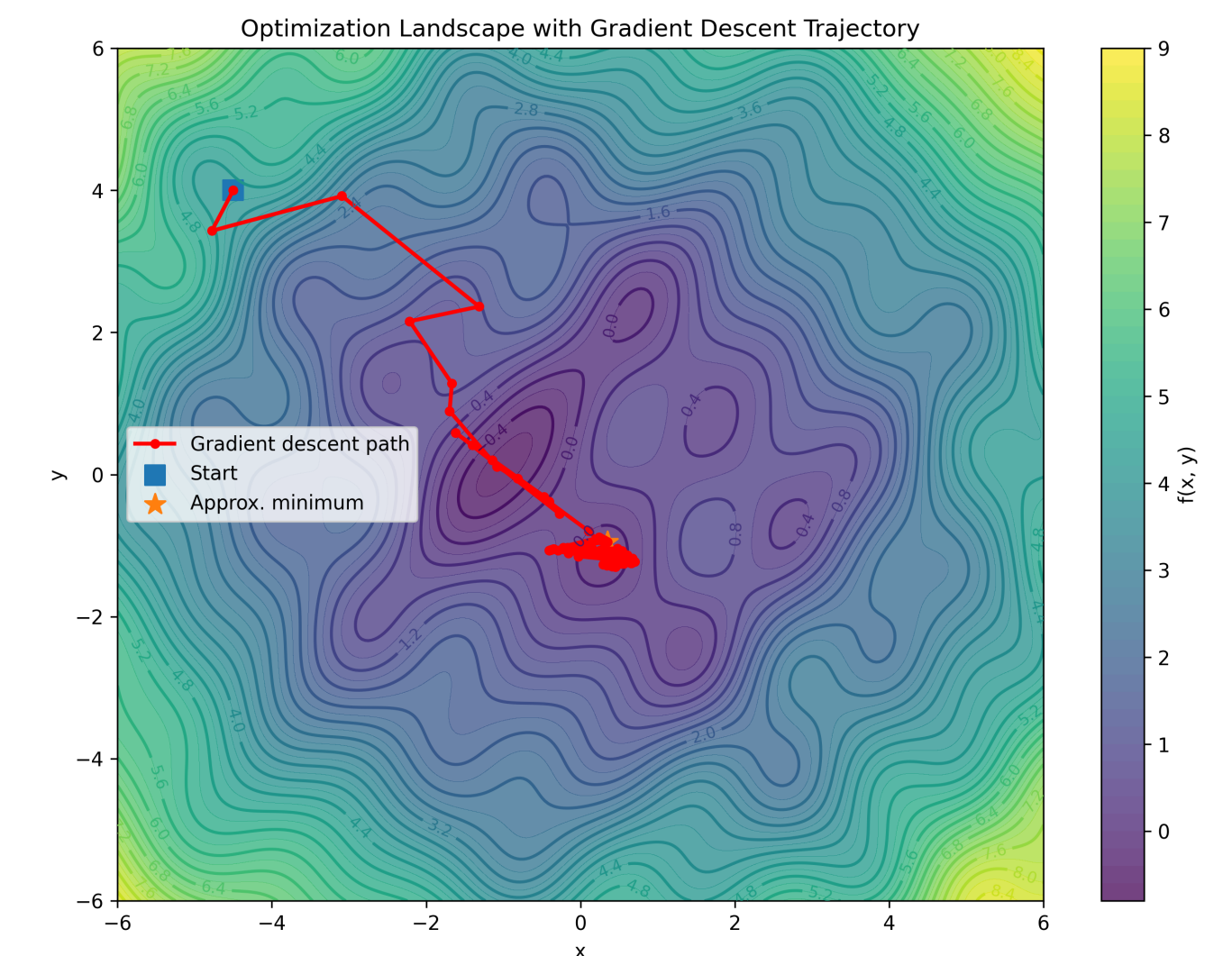
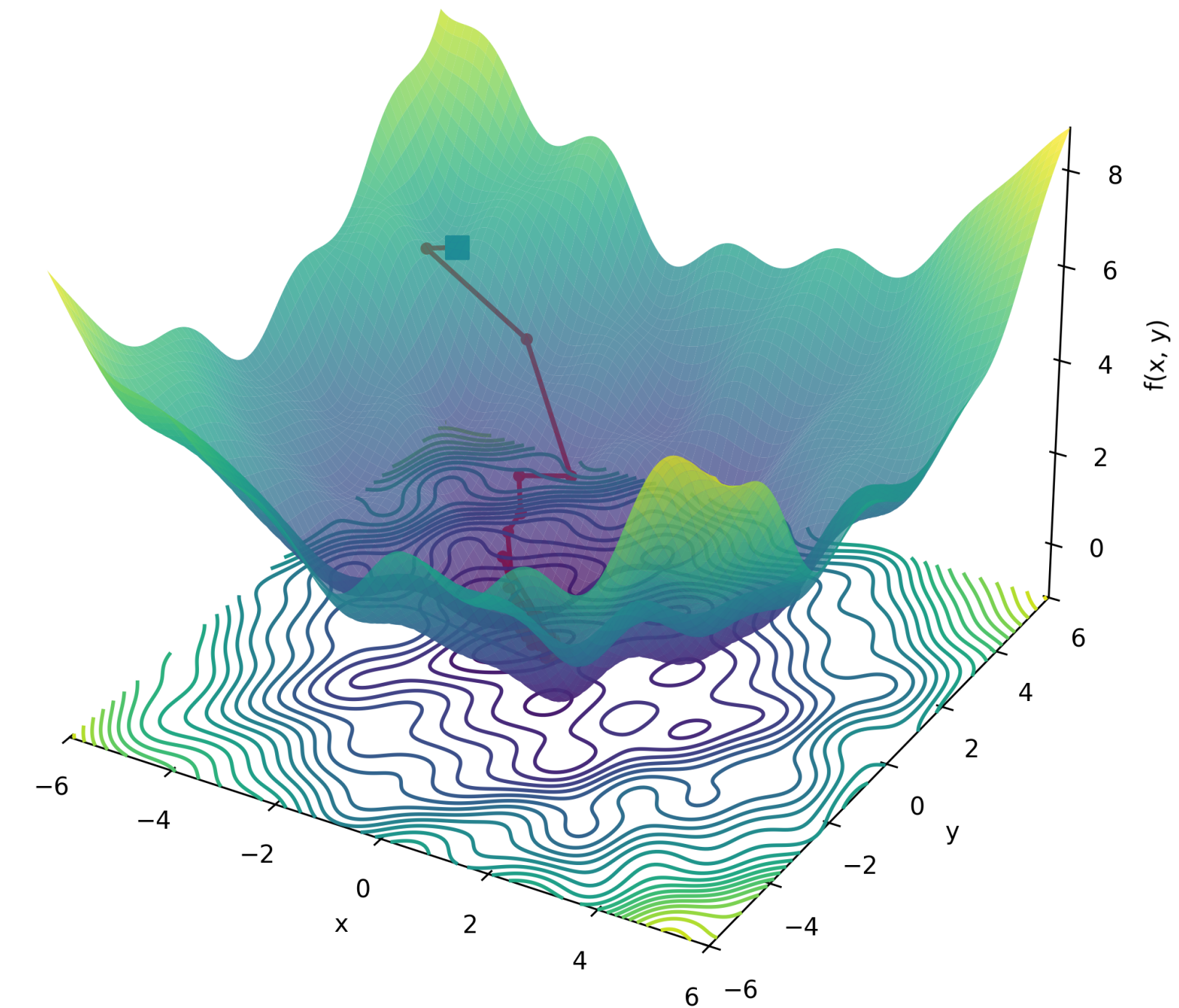
Algoritmos de optimización

Optimización

$$w^{(i+1)} \leftarrow w^{(i)} - \eta_i \nabla \mathcal{L}(w^{(i)})$$

η_i : tasa de aprendizaje

- Descenso por gradiente estocástico (SGD)
- SGD con momento
- RMSProp
- Adam



Algoritmos de optimización

Descenso por gradiente

$$w^{(i+1)} \leftarrow w^{(i)} - \eta g(w^{(i)})$$

$$g(w^{(i)}) = \nabla_w \mathcal{L}(w^{(i)}) \rightarrow \text{Común}$$

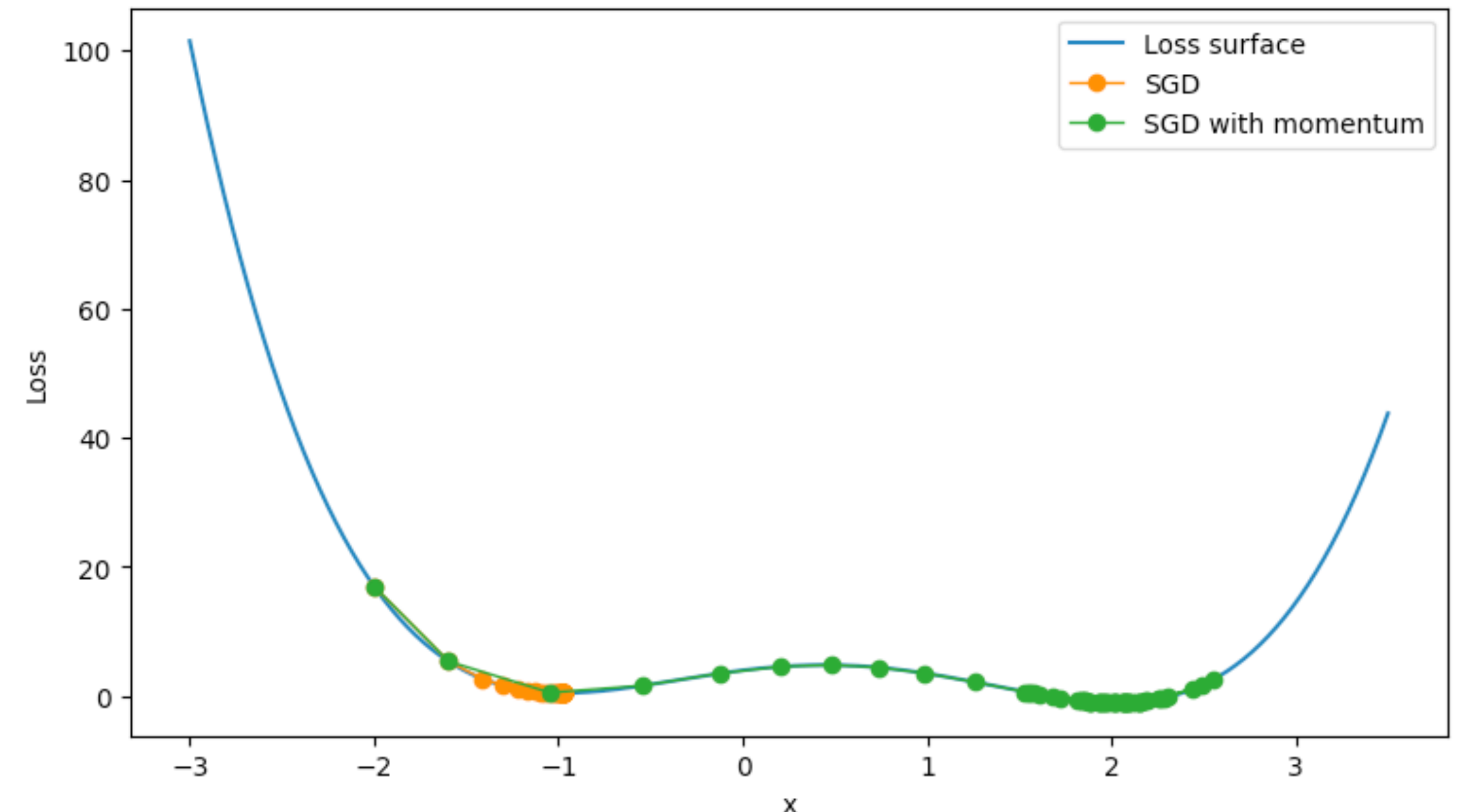
$$g(w^{(i)}) = \nabla_w l_j(w^{(i)}) \rightarrow \text{Estocástico}$$

$$g(w^{(i)}) = \frac{1}{B} \sum_{j \in B} \nabla_w l_j(w^{(i)}) \rightarrow \text{mini-batch}$$

Descenso por gradiente con Momento

$$w^{(i+1)} \leftarrow w^{(i)} - v^{(i+1)}$$

$$v^{(i+1)} = \eta \nabla_w \mathcal{L}(w^{(i)}) + \mu v^{(i)}$$



Algoritmos de optimización

RMSProp

$$w^{(i+1)} \leftarrow w^{(i)} - \eta \frac{g(w^{(i)})}{\sqrt{s^{(i+1)} + \epsilon}}$$

Gradiente

$$g(w^{(i)}) = \nabla_w \mathcal{L}(w^{(i)})$$

Acumulador de gradientes al cuadrado

$$s^{(i+1)} = \rho s^{(i)} + (1 - \rho)(g^{(i)})^2$$

$\rho \in [0,1)$ Factor de decaimiento (~0.8)

Adam (adaptive Momentum)

$$w^{(i+1)} \leftarrow w^{(i)} - \eta \frac{\hat{m}^{(i+1)}}{\sqrt{\hat{v}^{(i+1)} + \epsilon}}$$

$$\hat{m}^{(i+1)} = \frac{m^{(i+1)}}{1 - \beta_1^{i+1}} \leftarrow m^{(i+1)} = \beta_1 m^{(i)} + (1 - \beta_1)g(w^{(i)})$$

$$\hat{v}^{(i+1)} = \frac{v^{(i+1)}}{1 - \beta_2^{i+1}} \leftarrow v^{(i+1)} = \beta_2 v^{(i)} + (1 - \beta_2)(g(w^{(i)}))^2$$

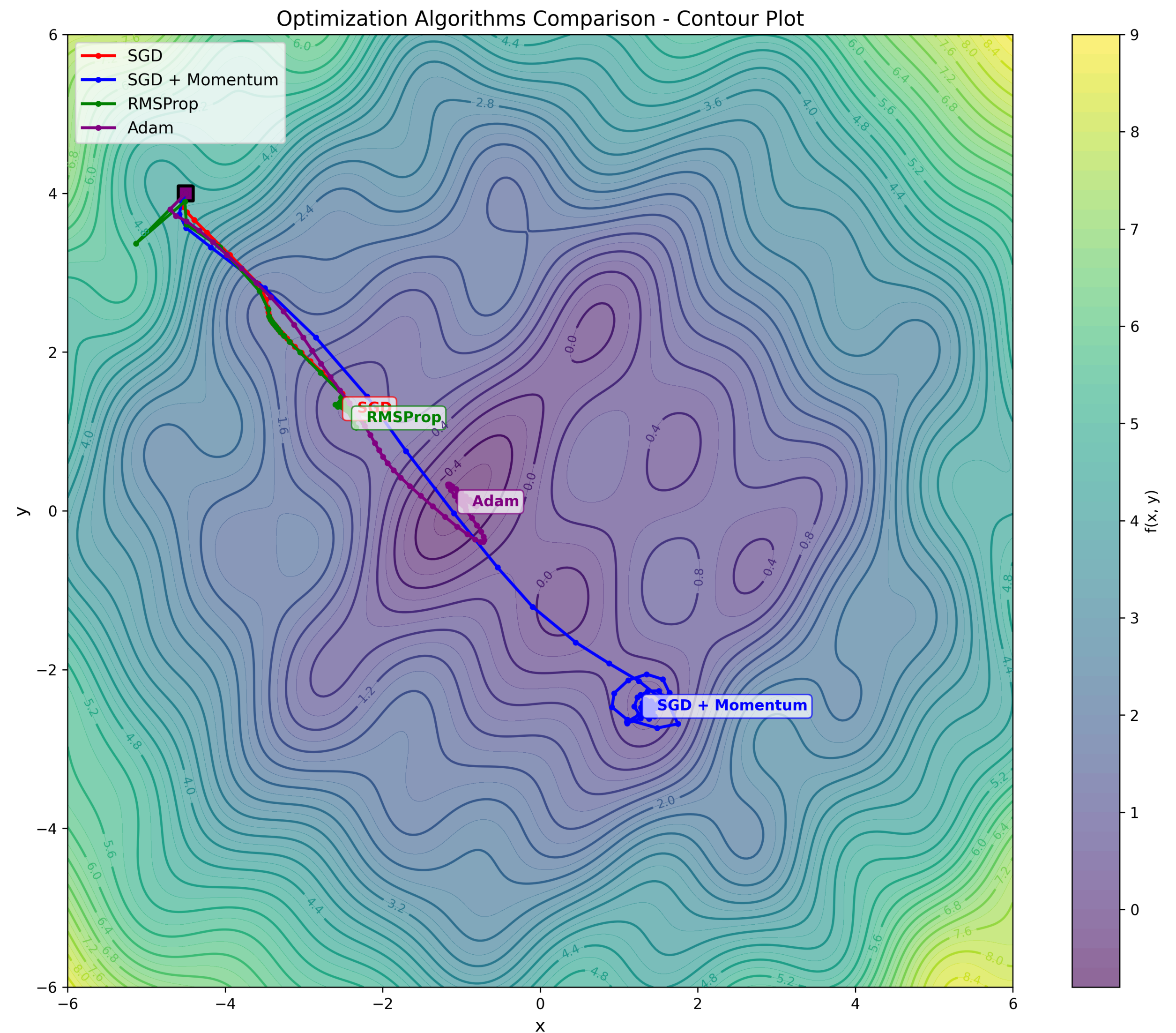
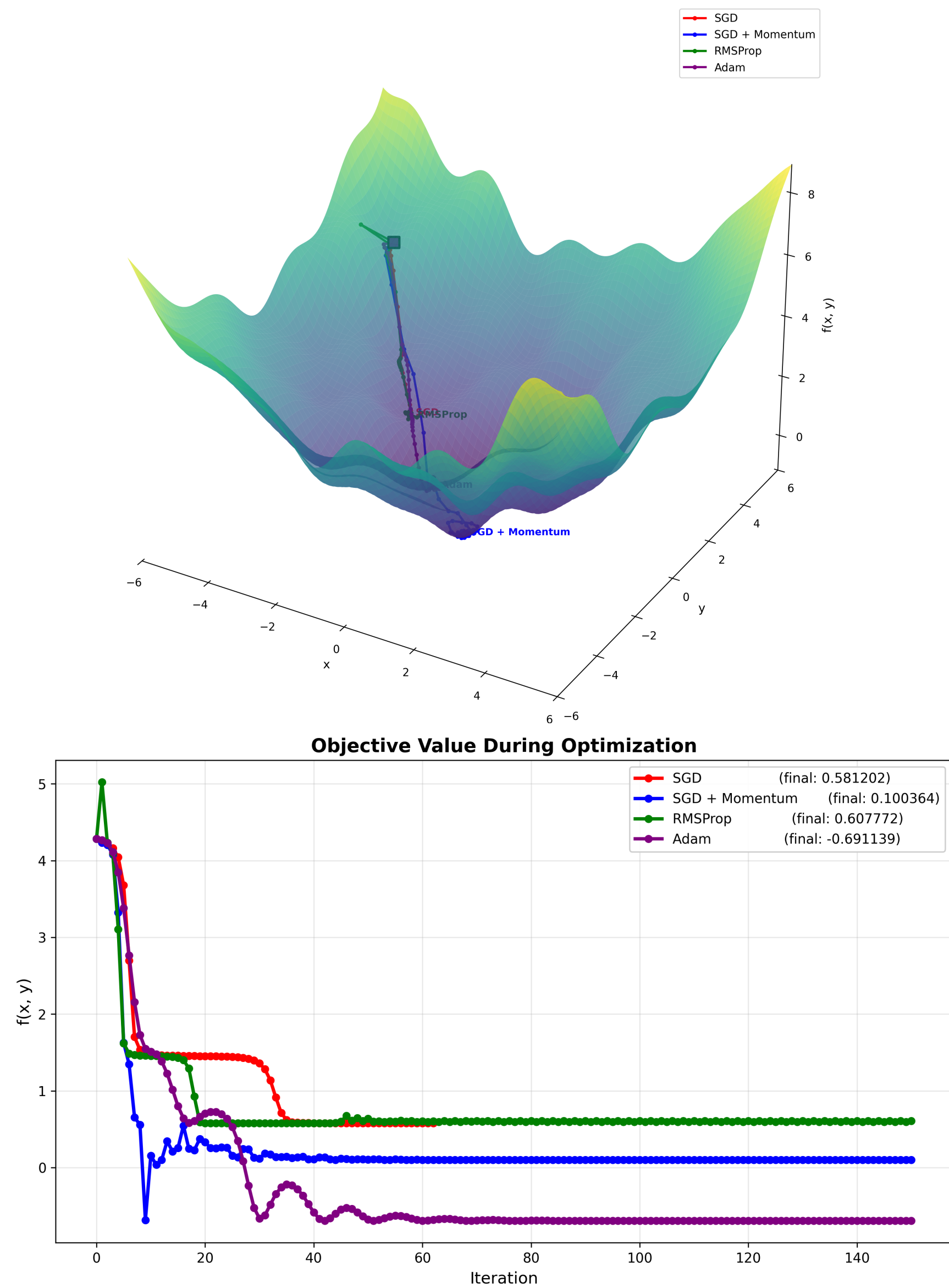
$$\beta_1 \in (0.8, 0.99)$$

Suaviza gradiente (momento)
más alto más inercia

$$\beta_2 \in (0.9, 0.9999)$$

Suaviza magnitud (RMSProp)
más alto promedio suave

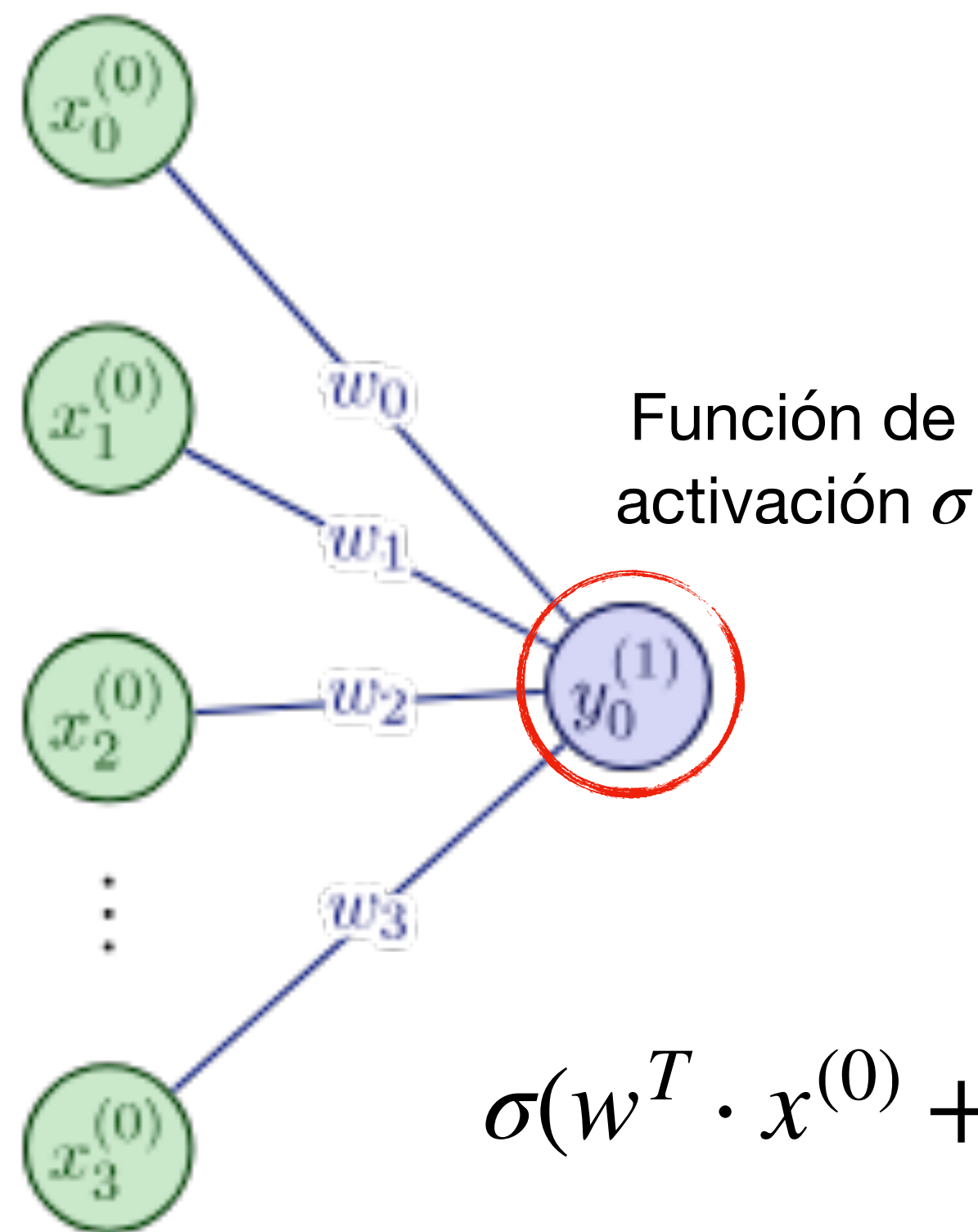
Algoritmos de optimización



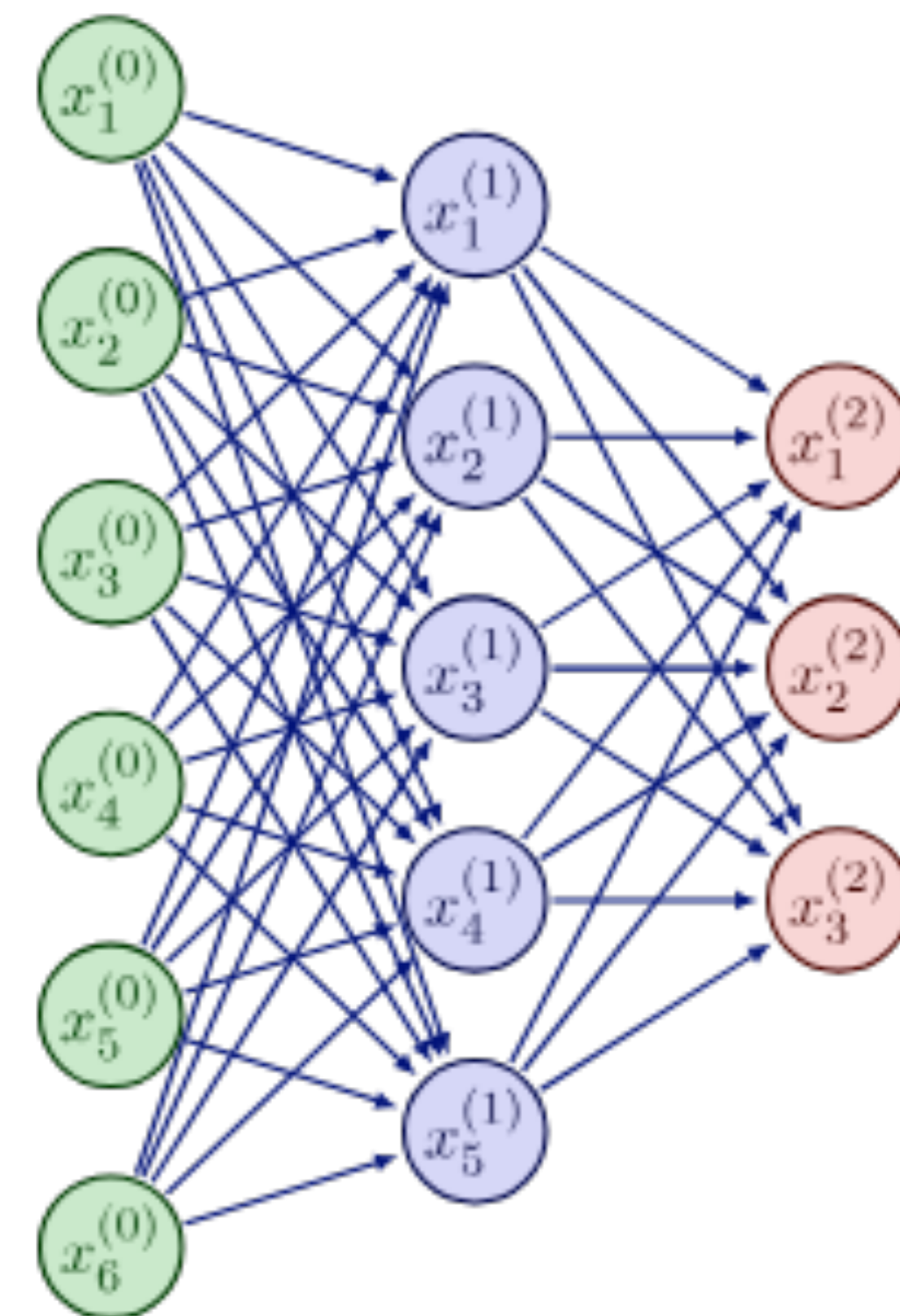
Redes neuronales

Introducción

Las primeras redes neuronales artificiales derivaron de las ideas del perceptrón simple (Frank Rosenblatt 1958) y el perceptrón multicapa.

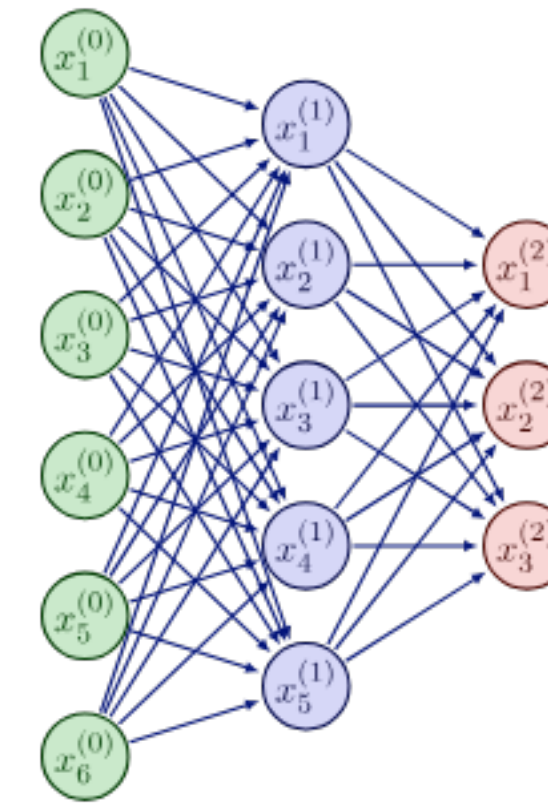
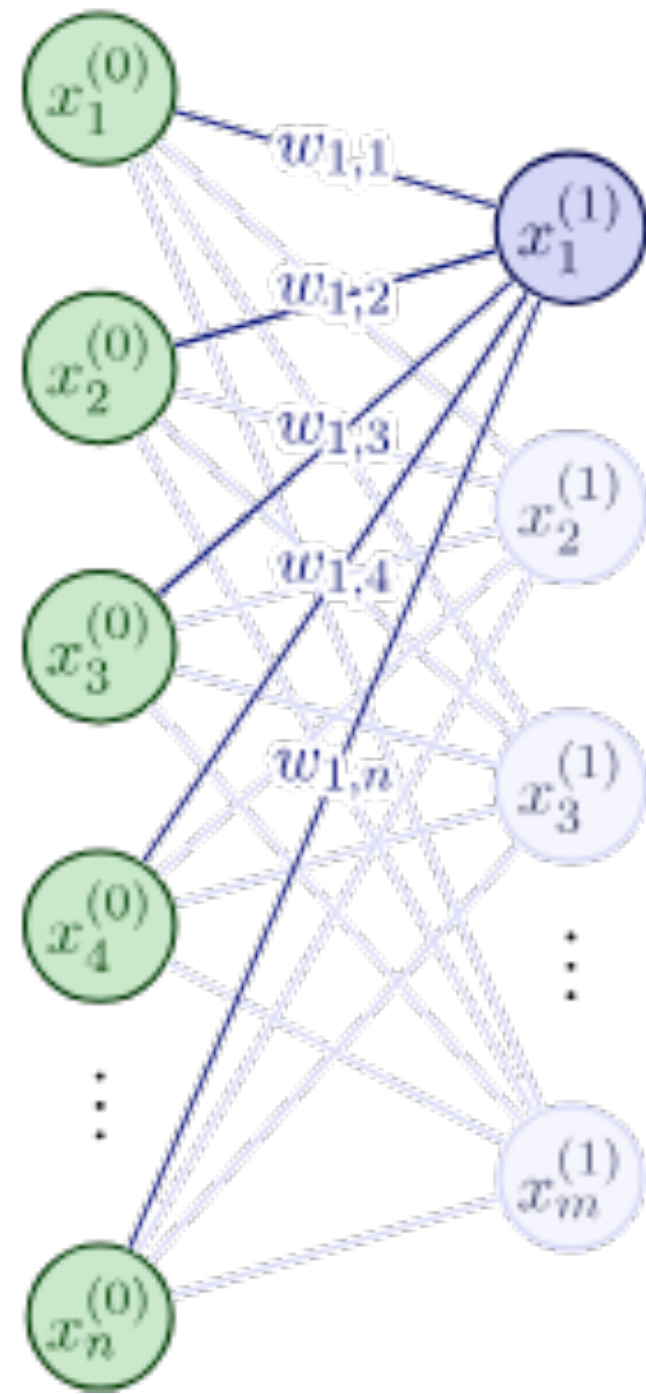


$$\sigma(w^T \cdot x^{(0)} + b) = \sigma \left(\sum_{i=1}^n x_i^{(0)} w_i + b \right)$$



Redes neuronales

Introducción



Neuronas:

- Valores de entrada (input values) o características (features)

$$x = (x_1, x_2, \dots, x_{n_0}) \in \mathbb{R}^{n_0}$$

- Pesos y sesgos (entrenables) para la capa l y neurona j

$$w_j^{(l)} = (w_{j,1}^{(l)}, w_{j,2}^{(l)}, \dots, w_{j,n_{l-1}}^{(l)}) \in \mathbb{R}^{n_{l-1}} \text{ y } b^{(l)} \in \mathbb{R}^{n_l}$$

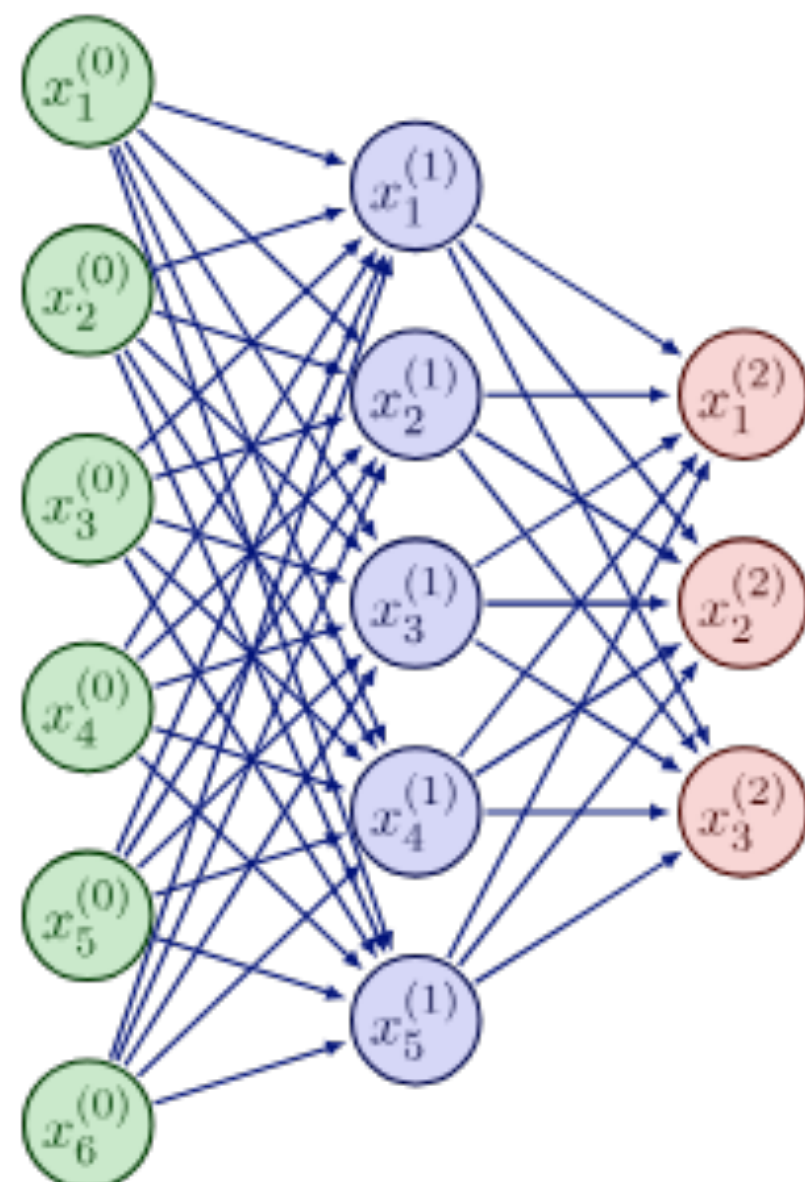
$$x_j^{(l)} = h_{l,j}(x^{(l-1)}) := \sigma^{(l)} \left(\sum_{i=1}^{n_{l-1}} x_i^{(l-1)} w_{j,i}^{(l)} + b_j^{(l)} \right) = \sigma^{(l)} \left((w_j^{(l)})^T \cdot x^{(l-1)} + b_j^{(l)} \right)$$

Redes neuronales

Introducción

$$x^{(l)} = h_l(x^{(l-1)}) = \sigma^{(l)}(W^{(l)}x^{(l-1)} + b^{(l)}) = \sigma^{(l)}(a^{(l)})$$
$$x^{(l-1)} \in \mathbb{R}^{n_{l-1}}$$
$$W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$$
$$b^{(l)}, a^{(l)} \in \mathbb{R}^{n_l}$$

Las redes neuronales se pueden representar mediante la composición de funciones afines y funciones de activación no lineales.



Si la red tiene L capas, entonces la salida final será

$$f_w(x^{(0)}) = (h_L \circ h_{L-1} \circ \dots \circ h_1)(x^{(0)}) = x^{(L)}$$

De lo cual, si volvemos a la expresión de la función de costo

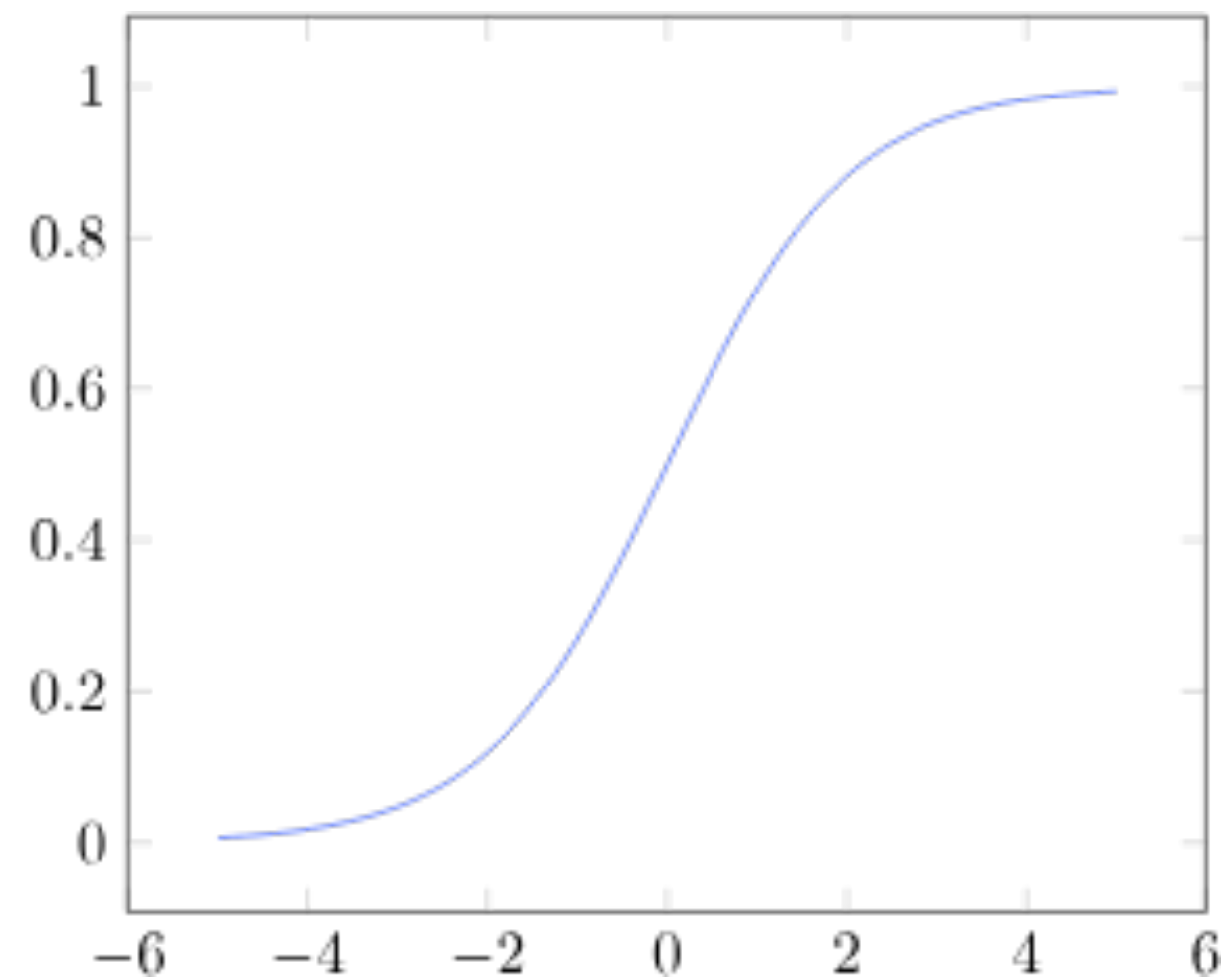
$$\mathcal{L}(w) = \sum_{i=1}^N l(y_i, f_w(x_i))$$

Vemos que dependerá de todos los parámetros $w = \{W^{(l)}\}_{l=1, \dots, L}$

Redes neuronales

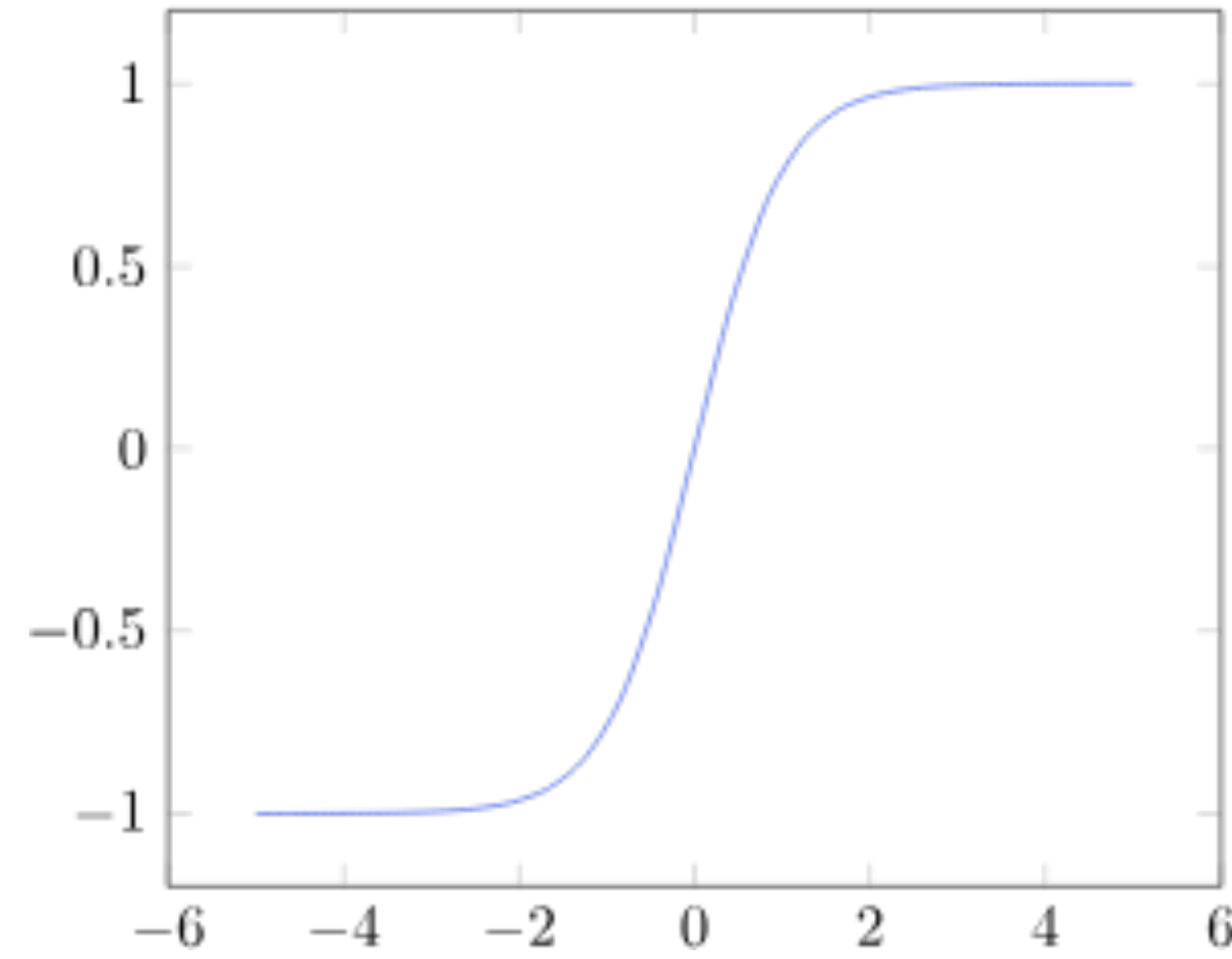
Algunas funciones de activación

Sigmoide



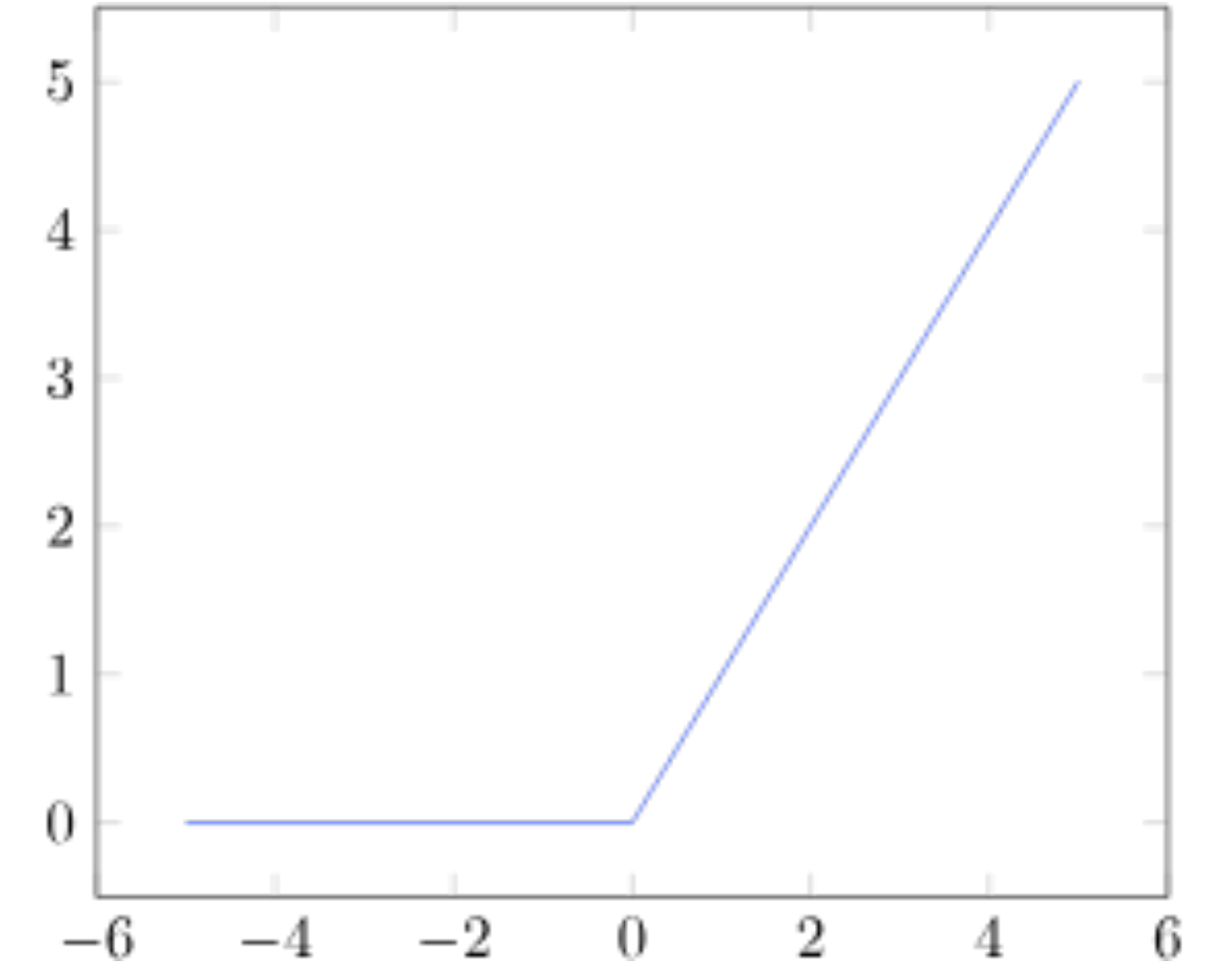
$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

Tanh



$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = 2\sigma(2x) - 1$$

ReLU



$$\text{ReLU}(x) = \max(0, x)$$

Redes neuronales

Teorema de Aproximación Universal

Una red neuronal de propagación hacia adelante (feedforward) con **una sola capa oculta** (shallow) y suficientes neuronas puede aproximar **cualquier función continua** en un dominio compacto, con la precisión que uno desee.

Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ una función continua en un conjunto compacto. Entonces, para todo ϵ , existe una red neuronal de la forma

$$F(x) = \sum_{j=1}^N w_j^{(2)} \sigma((w_j^{(1)})^T x + b_j^{(1)}) + b^{(2)}$$

Tal que

$$|F(x) - f(x)| < \epsilon \text{ para todo } x$$

PyTorch

Introducción práctica en Colab

[intro_pytorch.ipynb - Parte 2](#)

PyTorch

Introducción práctica en Colab

Práctica 1 - Continuación