Contenido

DEFINICIONES

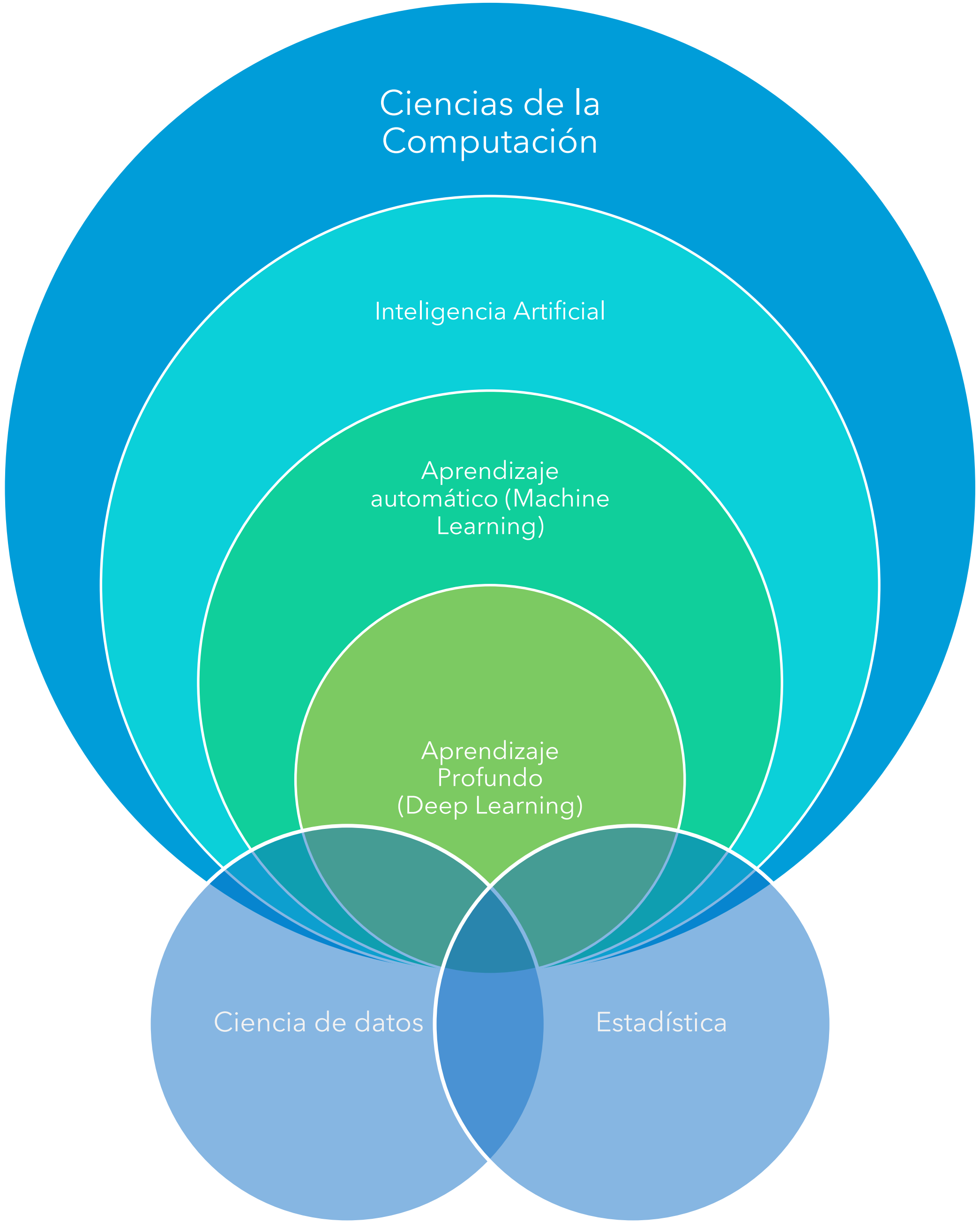
CLASIFICACIÓN Y TAXONOMÍA DE LA IA.

HISTORIA

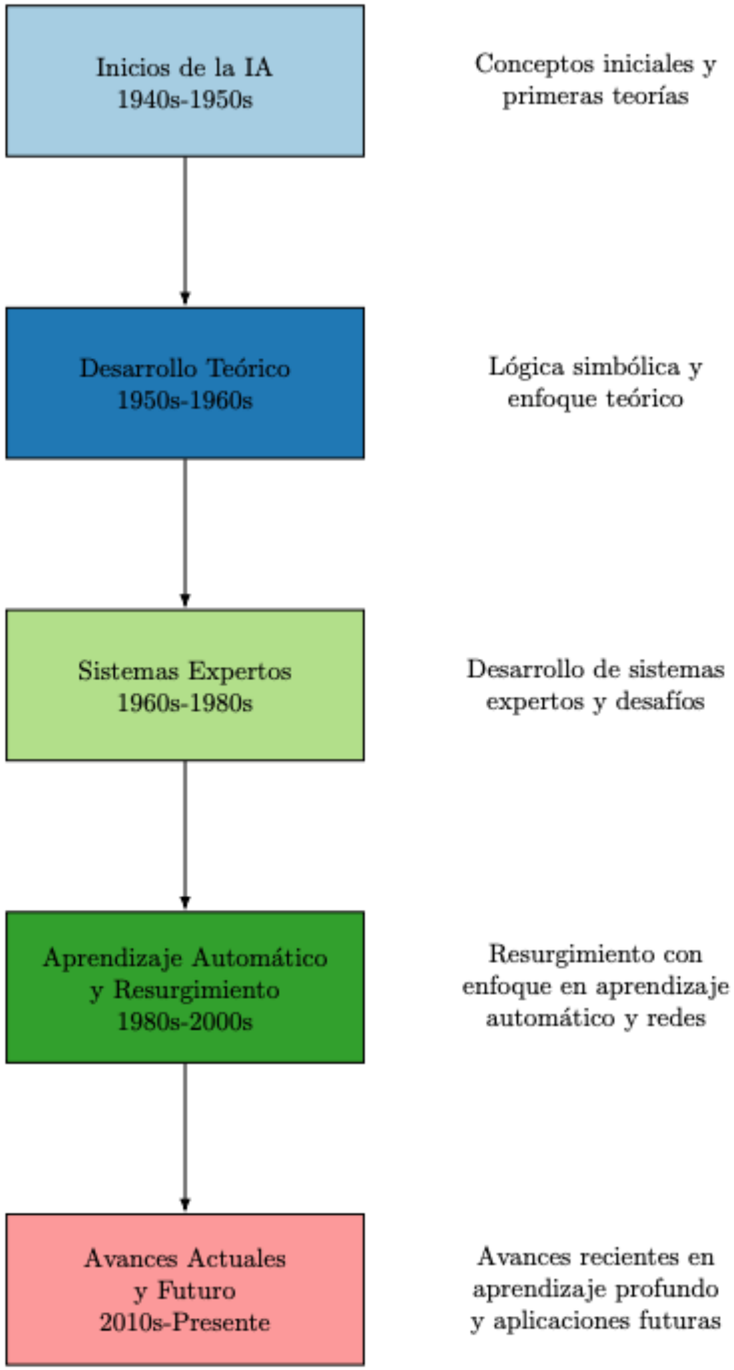
TIPOS DE APRENDIZAJE

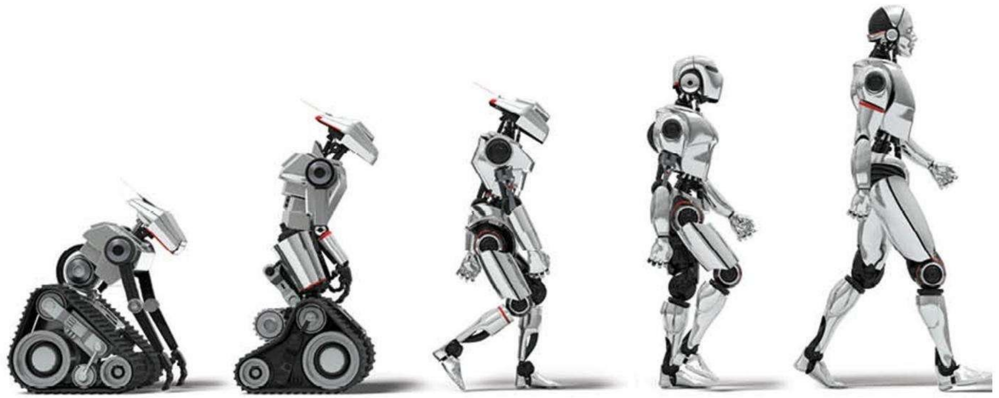
APRENDIZAJE PROFUNDO

Taxonomía de la IA



Historía de la inteligencia Artifical ¿ ?





Aprendizaje automático (Machine Learning)

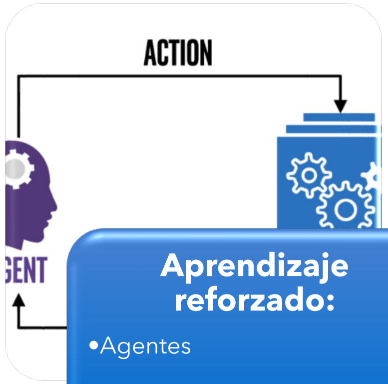
- "Extraer conocimiento de los datos". Se basa en estadística, especialmente en métodos como regresiones lineales y estadística bayesiana.

Aprendizaje supervisado:

- Predecir salida correcta o etiqueta.
- Clasificación
- Respuesta: si/no

Aprendizaje no supervisado:

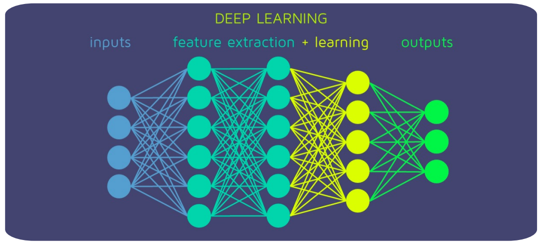
- Descubrir la estructura de los datos.
- Clusterizado
- Reducción

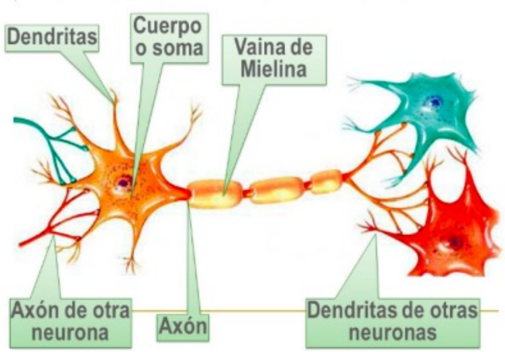


Aprendizaje reforzado:

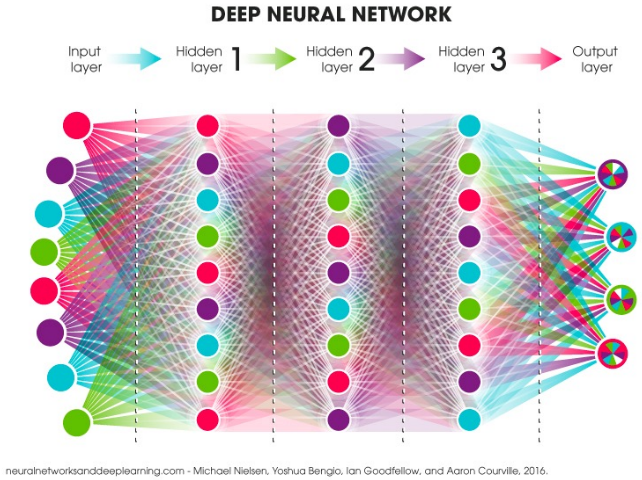
- Agentes
- Se provee una retroalimentación
- Buenas o malas decisiones.

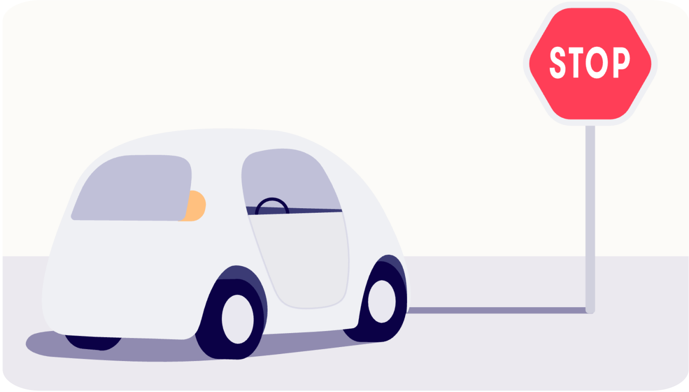
Deep Learning (Redes Neuronales)





- Deep Learning: se le conoce a ciertos tipos de técnicas de aprendizaje automático en las que varias "capas" de unidades de procesamiento simples están conectadas en una red para que la entrada al sistema pase a través de cada una de ellas.



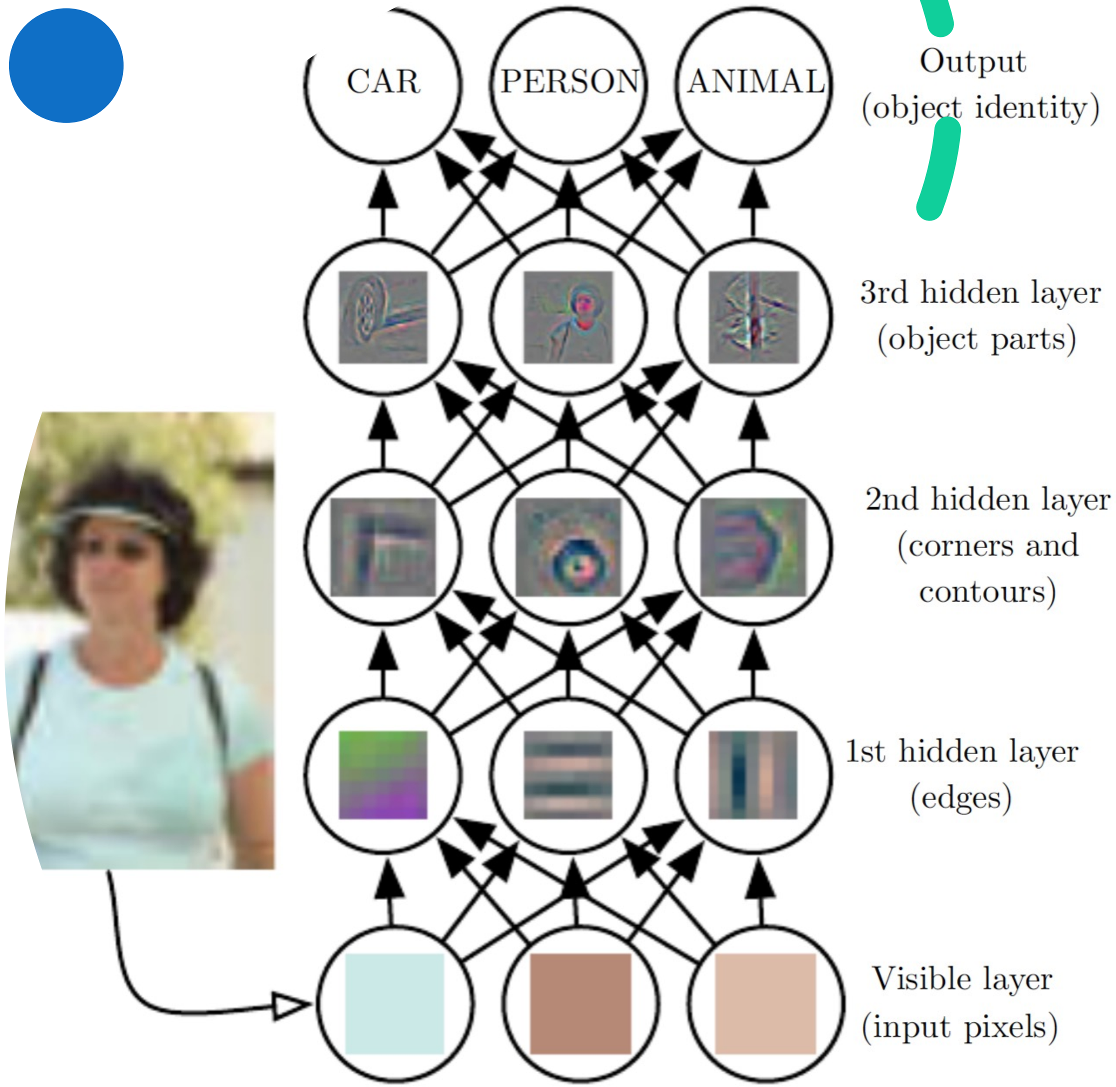


Entrenamiento

- En el proceso de entrenamiento, a cada nodo de las capas internas de la red se le asigna un peso (función de activación). Estos pesos deben ser ajustados (datos) para producir salidas corrrectas (Optimización).

Clasificación

Toda vez que un modelo se ha entrenado, es posible (clasificar) datos de entrada.



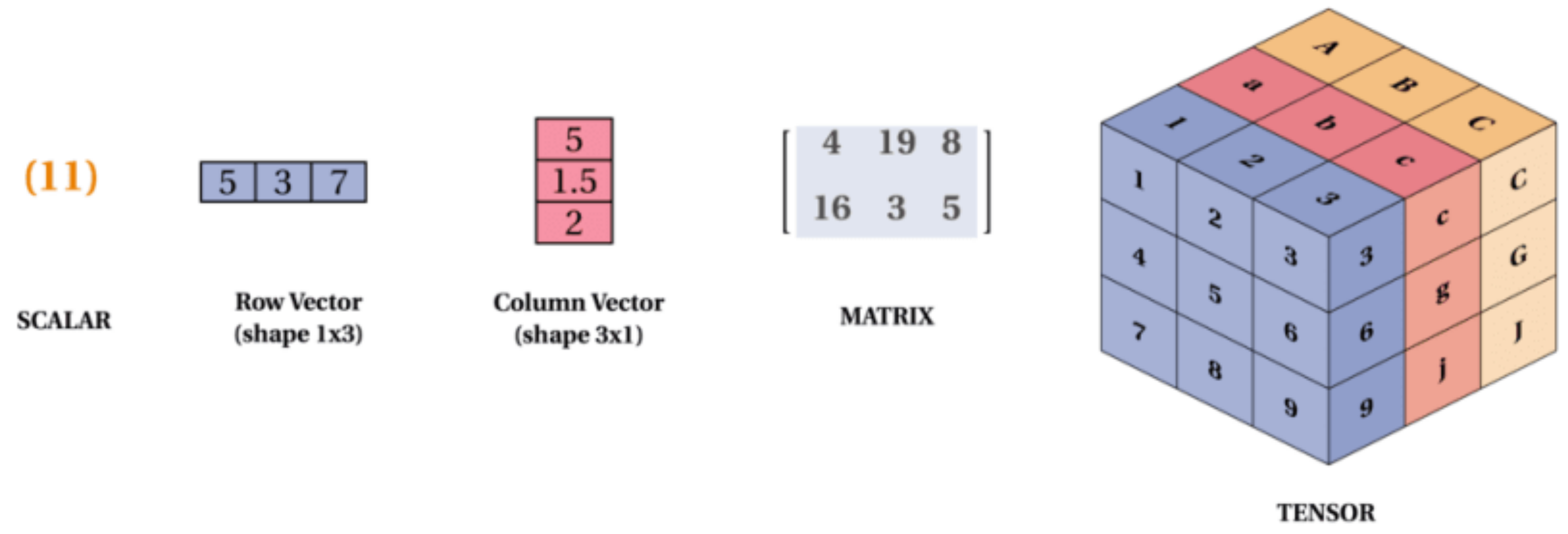
Ejercicio (15 min)

Búsqueda de **aplicaciones disruptivas** de la IA.

1. Conducción autónoma
2. Diagnósticos médicos
3. Asistentes virtuales
4. Procesamiento de Lenguaje Natural
5. Industria 4.0
6. Robótica y Drones
7. Educación

Conceptos fundamentales

Inteligencia Artificial



Tipos de objetos en álgebra lineal

- **Escalares**
- **Vectores**
- **Matrices**
- **Tensores**

Operaciones matriciales

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

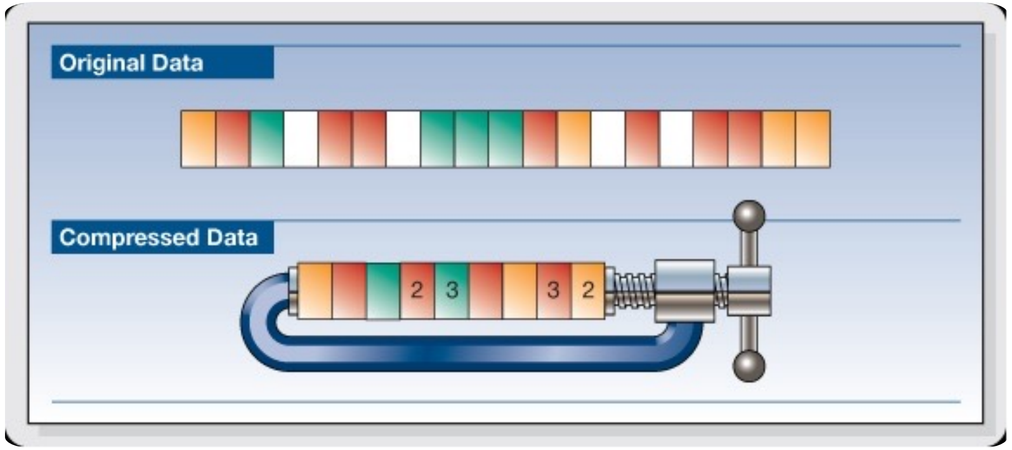
$$I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$I_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- Matriz Transpuesta
- Multiplicación de matrices y vectores
- Matriz identidad
- Matriz Inversa
- Normas
- Eigenvalores y eigenvectores
- Trazas
- Determinantes

Análisis de Componentes Principales

- El objetivo principal del análisis de componentes principales (PCA) es reducir la dimensión de los datos. Tiene aplicaciones en:



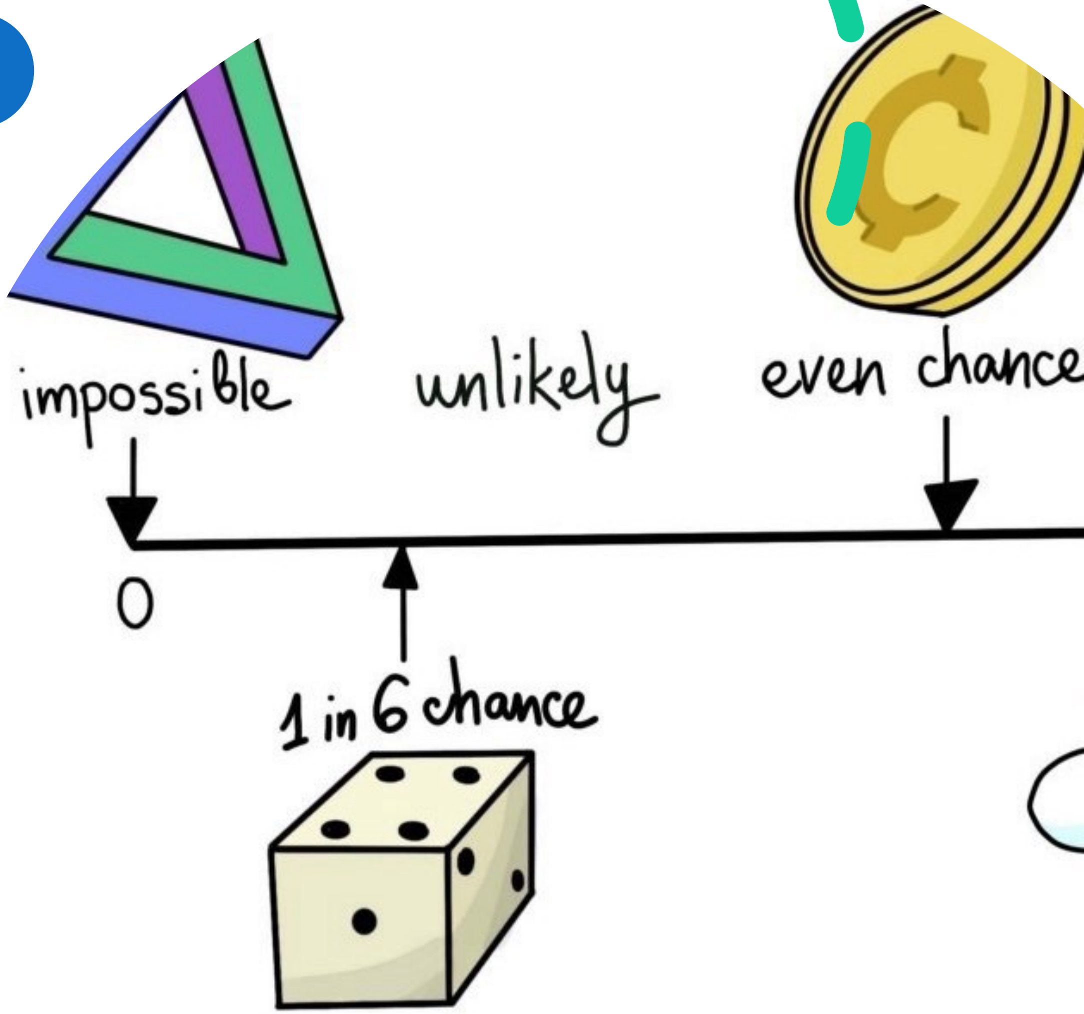
Visualización → Extracción de características → Compresión de datos → Entre otros

$$\begin{pmatrix} \lambda_0 & 0 & ...0 \\ 0 & \lambda_1 & ...0 \\ ... & ... & ... \\ 0 & 0 & ...\lambda_{n-1} \end{pmatrix}$$

Probabilidad e incertidumbre

¿Por qué probabilidad?

- Casi todas las actividades en la vida real requieren alguna habilidad para razonar frente a la **incertidumbre**.
- Una fuente de incertidumbre es una **observación incompleta**.
- En ML ocurre lo mismo, se puede observar un conjunto de datos muy grandes. Pero no todas las posibles **variantes** de esos datos.
- En algunos casos, **no es posible tener todos los datos**, tampoco observar los procesos que los originan.



Probabilidad

- Es posible representar la incertidumbre a través de probabilidad.

- La teoría de la probabilidad fue diseñada originalmente para analizar la frecuencia de eventos.
  - Poker
  - Dados (1/6)
  - Lanzamiento de una moneda (1/2)
- **Probabilidad frecuentista**

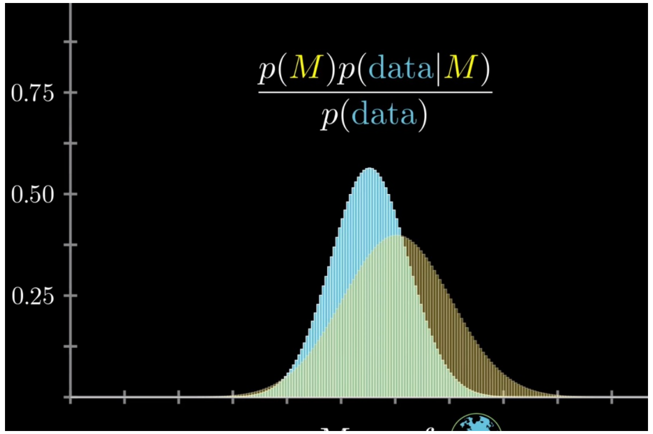
Probabilidad Bayesiana

- La teoría de la probabilidad se modifica y se le da un nuevo sentido.

- En el caso de un paciente, un médico podría diagnosticar a un paciente COVID de acuerdo a sus síntomas.

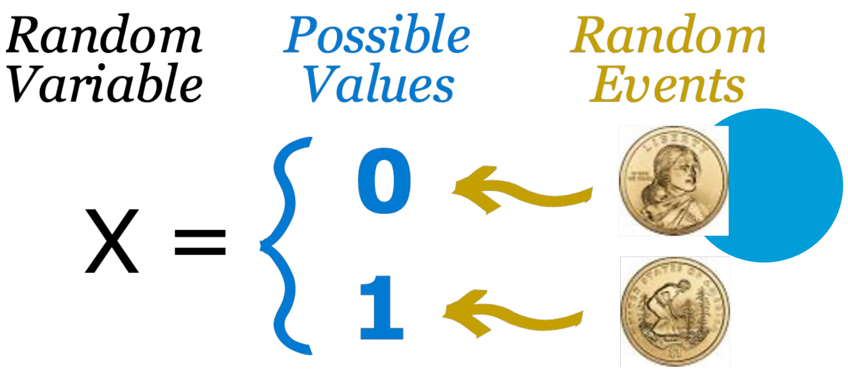
  - 0 – Incertidumbre total
  - 1 – Certidumbre total

- Niveles de certidumbre
- **Probabilidad Bayesiana**

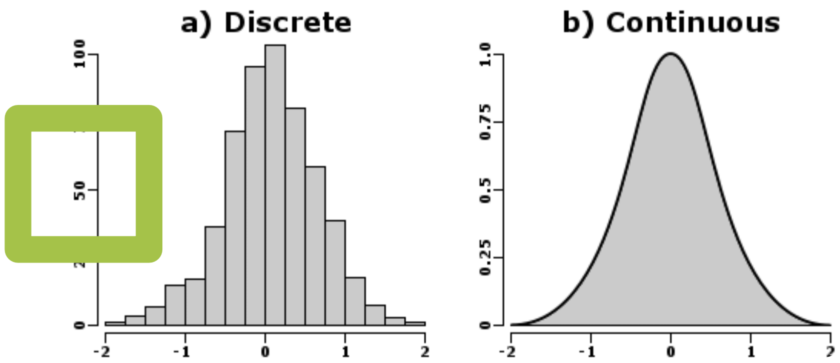


$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)},$$

Variables aleatorias



Una variable aleatoria es aquella que puede tomar distintos valores aleatorios.



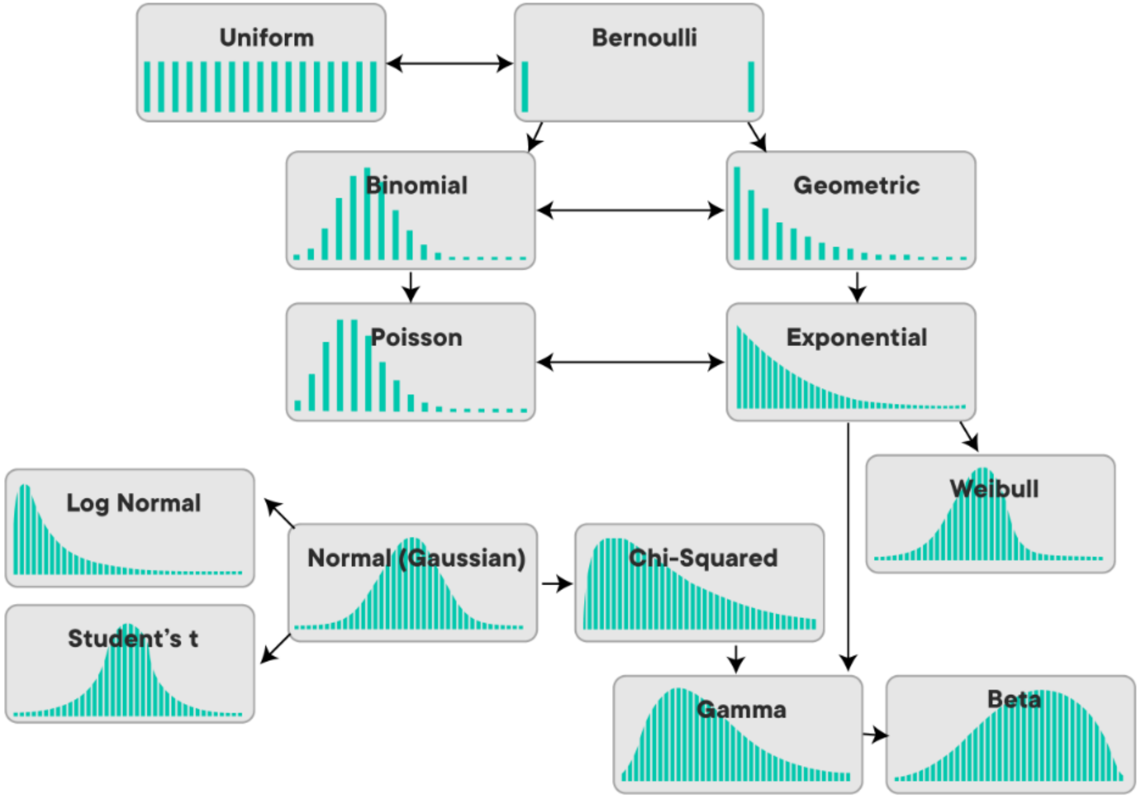
Generalmente se denotan con letras mayúsculas.

Estas variables pueden ser discretas o continuas.

Distribuciones de Probabilidad

Una distribución de probabilidad es una descripción de que tan probable es que una variable o conjunto de variables tome un posible estado.

- La forma en la que se describe una variable depende del tipo de variable:
  - **Continuas**
  - **Discretas**



Variables Discretas
Función de masa de probabilidad (PMF)

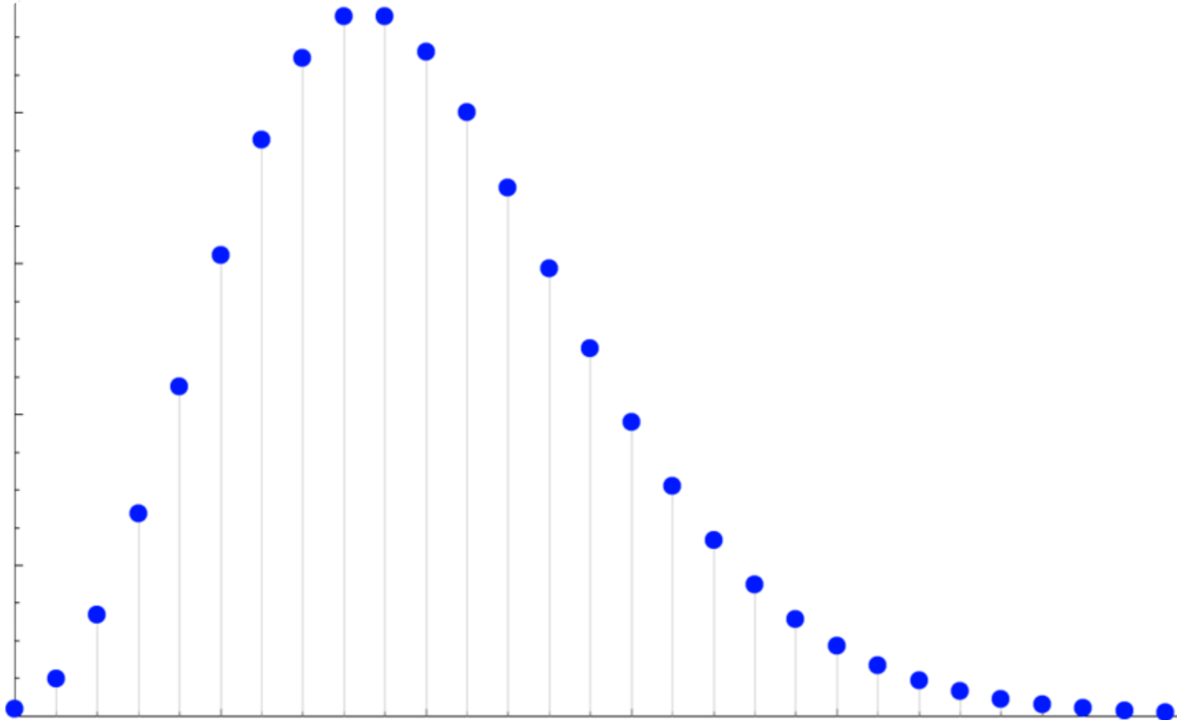
- Usualmente se denota a la función de masa de probabilidad con la letra **P** mayúscula.

- $P(x)$ denota la probabilidad que la variable X tome el valor x

  - $P(X = x) = 0$ Indica que es imposible que esto ocurra
  - $P(X = x) = 1$ Indica que existe certeza que esto ocurra

- La función de masa de probabilidad puede actuar sobre múltiples variables de forma simultanea.

  - $P(X = x, Y = y)$ se puede abreviar como $P(x, y)$



Variables Discretas
Función de masa de probabilidad (PMF)

Propiedades:

1. El dominio de la función se establece como todos los posibles estados (valores) de x

2. El rango de la función se encuentra dentro del intervalo real [0,1]

3. Se cumple que $\sum_{x \in \mathrm{x}} f(x) = 1$

Función de masa de probabilidad (PMF) Ejemplo

Considere una variable aleatoria discreta x con k distintos estados.

Asumiendo una distribución uniforme sobre x, la función de masa de probabilidad es:

$$P(\mathrm{x} = x_i) = \frac{1}{k}$$

$$\sum_i P(\mathrm{x} = x_i) = \sum_i \frac{1}{k} = \frac{k}{k} = 1$$

Distribuciones de Probabilidad Más Comunes

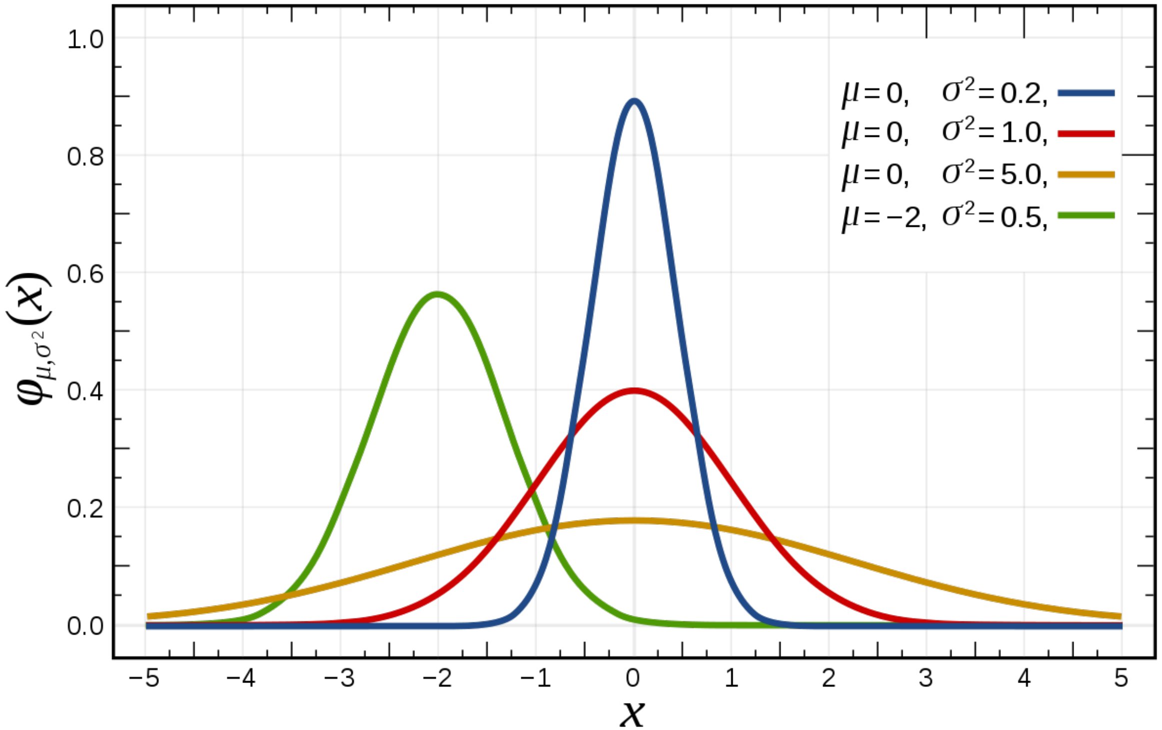
Distribución Gaussiana

Esta es una de las distribuciones de probabilidad más utilizadas, también se conoce normalmente como distribución normal.

$$\mathcal{N}(x \mid \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right).$$

Los dos parámetros principales son:

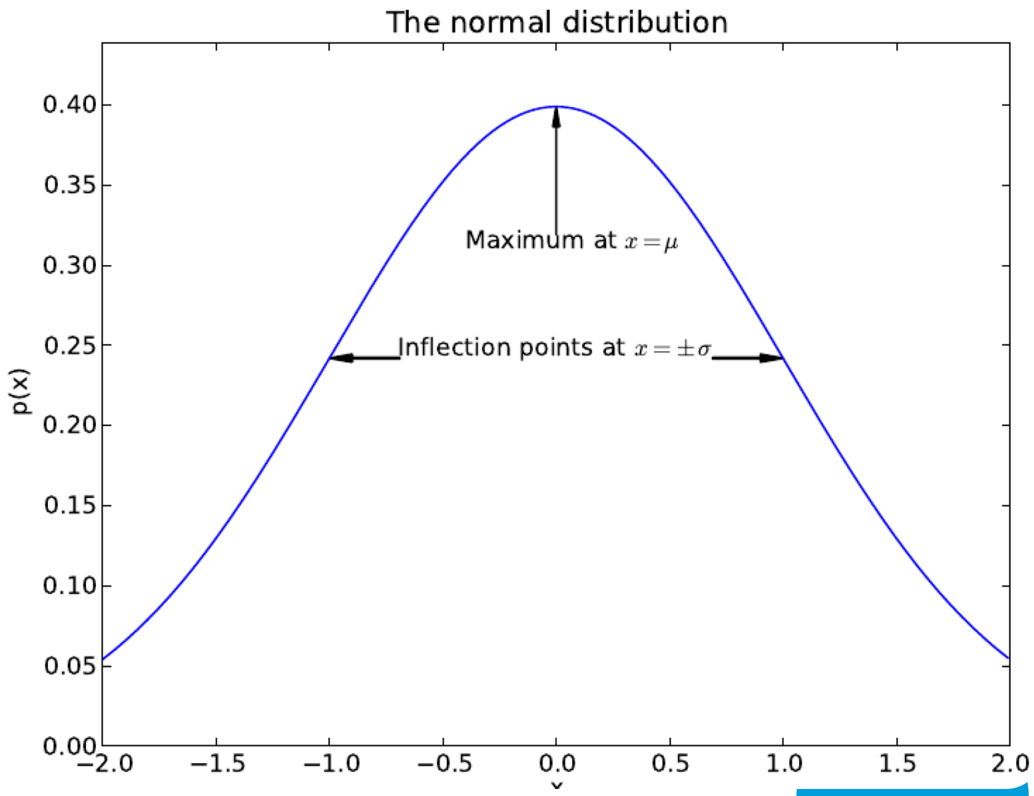
- $\mu \in \mathbb{R}$: Media (tendencia central)
- $\sigma \in \mathbb{R}^+$: Desviación estándar (Dispersión de datos)
- σ^2: Varianza



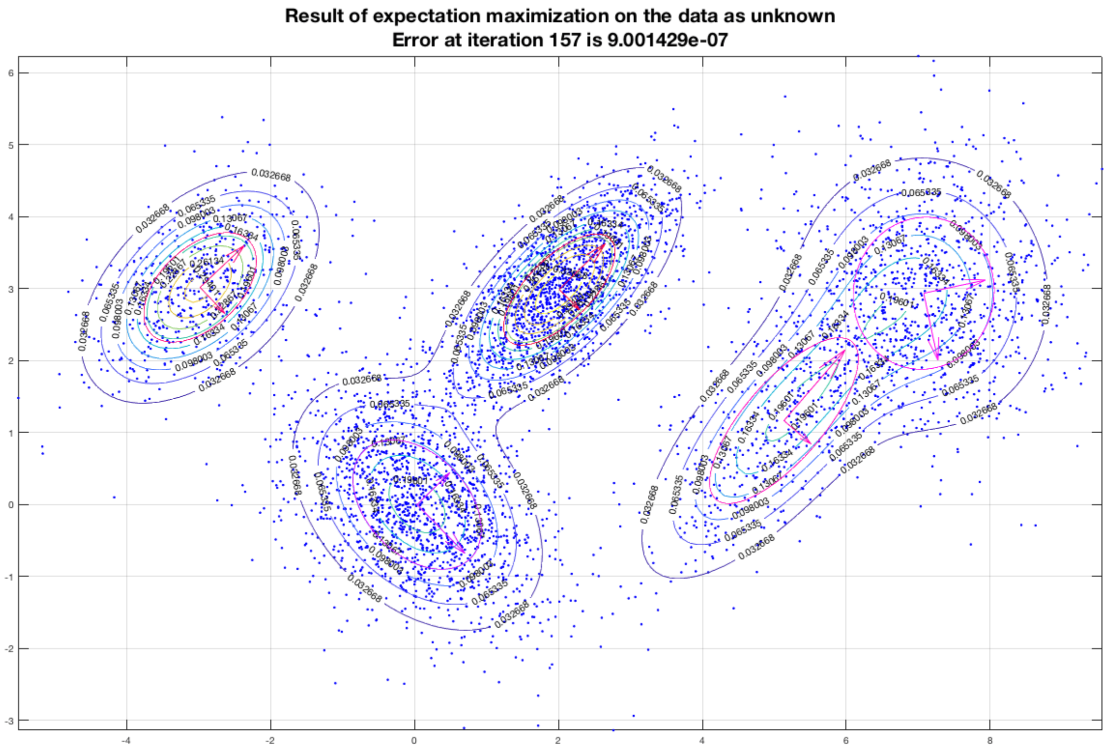
Distribución Gaussiana (Normal)

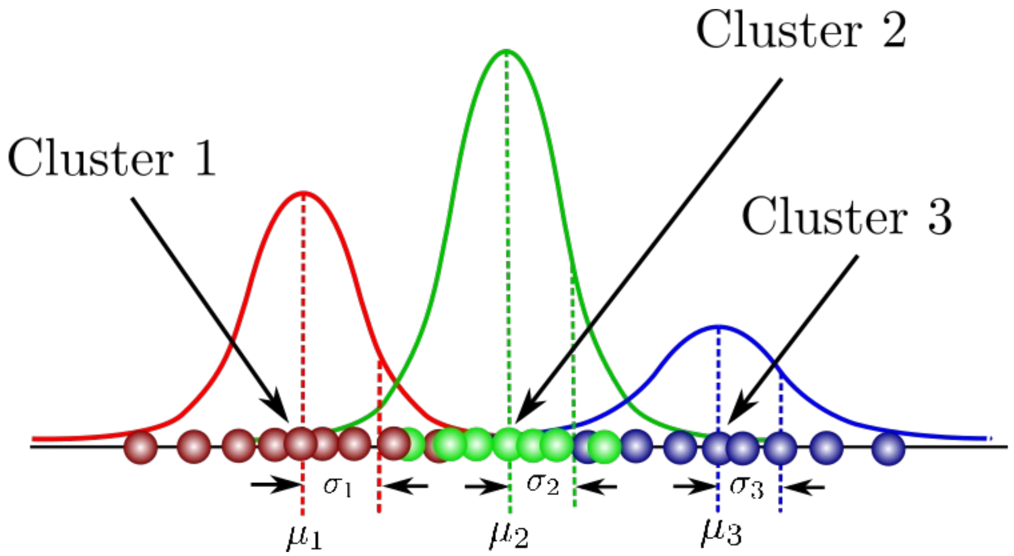
La distribución normal exhibe una forma clásica de campana, con un máximo en la media μ, la amplitud de esta curva esta controlada por la desviación estándar σ.

En el ejemplo se muestra una distribución con media = 0 y desviación estándar = 1



Mezcla de Distribuciones Gaussianas





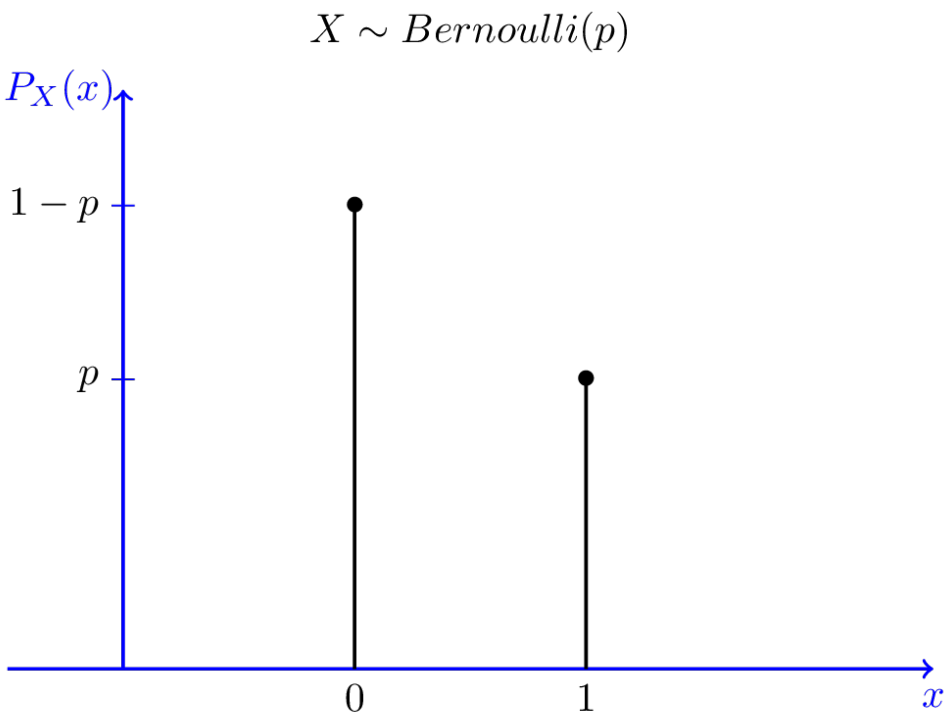
$$P(\mathrm{x}) = \sum_i P(\mathrm{c} = i) P(\mathrm{x} \mid \mathrm{c} = i)$$

Distribución Bernoulli

- Esta distribución fue llamada en honor al matemático Suizo Jacob Bernoulli.
- Esta distribución fue ideada como un modelo para representar un experimento donde se cuenta con repuestas del tipo Si y No.
- Toma un valor de 1 con una probabilidad p y el valor 0 con una probabilidad $q = 1 - p$.

$$f(x) = \begin{cases} 1-p & \text{si } x = 0 \\ p & \text{si } x = 1 \\ 0 & \text{en cualquier otro caso} \end{cases}$$

Distribución Bernoulli

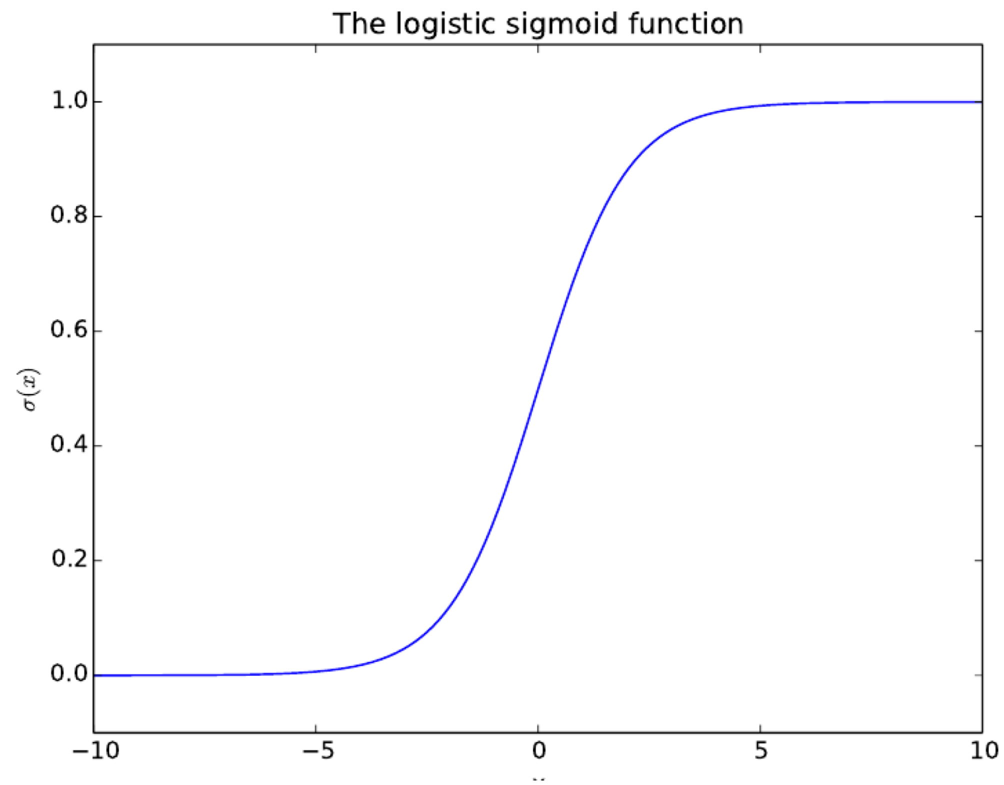


Bernoulli	
Parámetros	$0 < p < 1$
Dominio	$x = \{0, 1\}$
Función de probabilidad (fp)	$p^x (1-p)^{1-x}$
Función de distribución (cdf)	$\begin{cases} 0 & x < 0 \\ 1-p & 0 \leq x < 1 \\ 1 & x \geq 1 \end{cases}$
Media	p
Mediana	N/A
Moda	$\begin{cases} 0 & q > p \\ 0, 1 & q = p \\ 1 & q < p \end{cases}$
Varianza	$p(1-p)$
Coeficiente de simetría	$\frac{q-p}{\sqrt{pq}}$

Propiedades de funciones comunes

- Algunas funciones son de interés particularmente para modelos de Deep learning. Una de ellas es la función logística sigmoide.
- Esta función se utiliza para generar el parámetro p de una distribución Bernoulli.

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

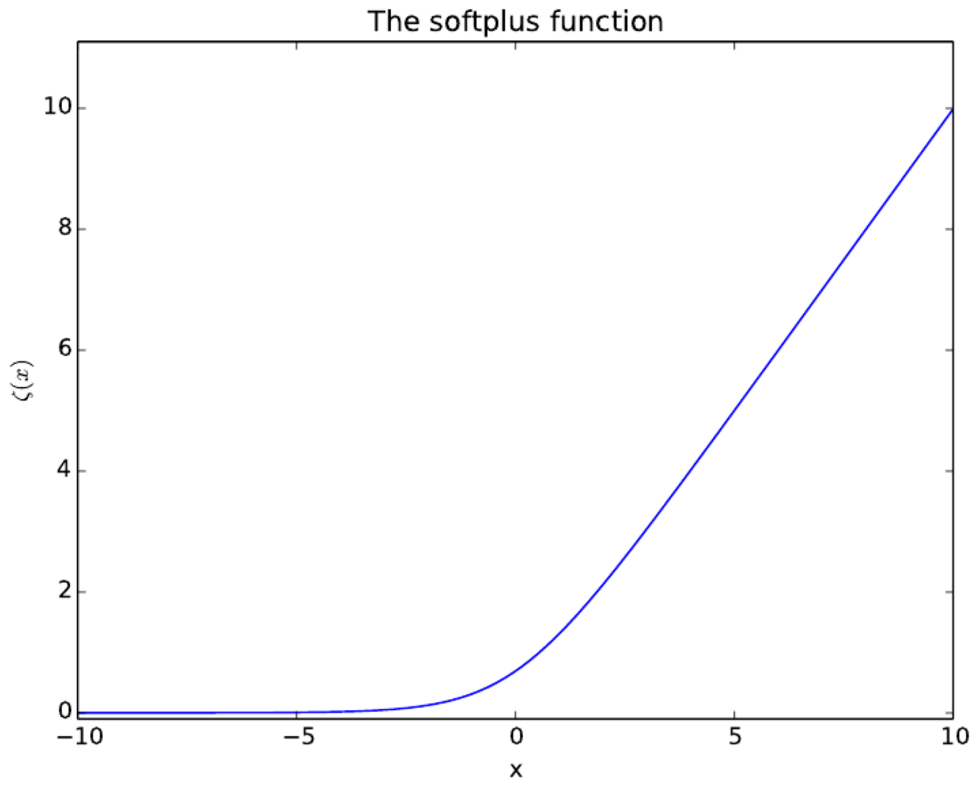


Propiedades de funciones comunes

- Otra función ce interés es la softplus. Es útil para para producir el parámetro β o σ de una distribución Normal

$$\zeta(x) = \log\left(1 + \exp(x)\right)$$

$$\sigma(x) = \frac{\exp(x)}{\exp(x) + \exp(0)}$$



Ejercicio (10 - 15 min)

- Escribir en Python un método para producir cada una de las funciones:
  - Logística sigmoide.

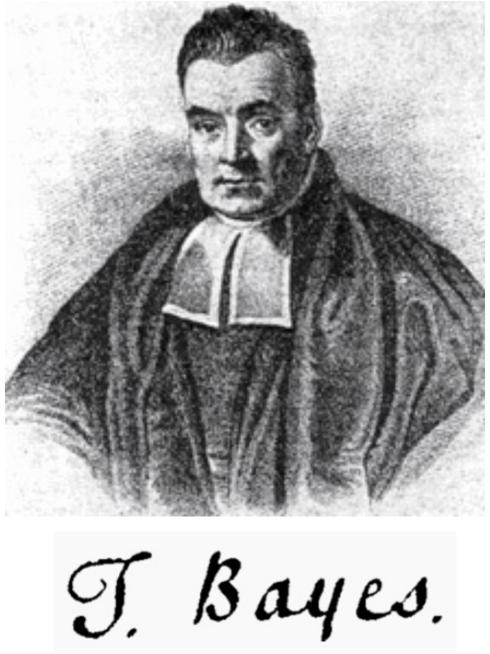
$$\sigma(x) = \frac{1}{1+\exp(-x)}.$$

  - Softplus

$$\zeta(x) = \log\left(1+\exp(x)\right)$$

- Se deberán desplegar las gráficas resultantes sobre un intervalo definido [a, b]

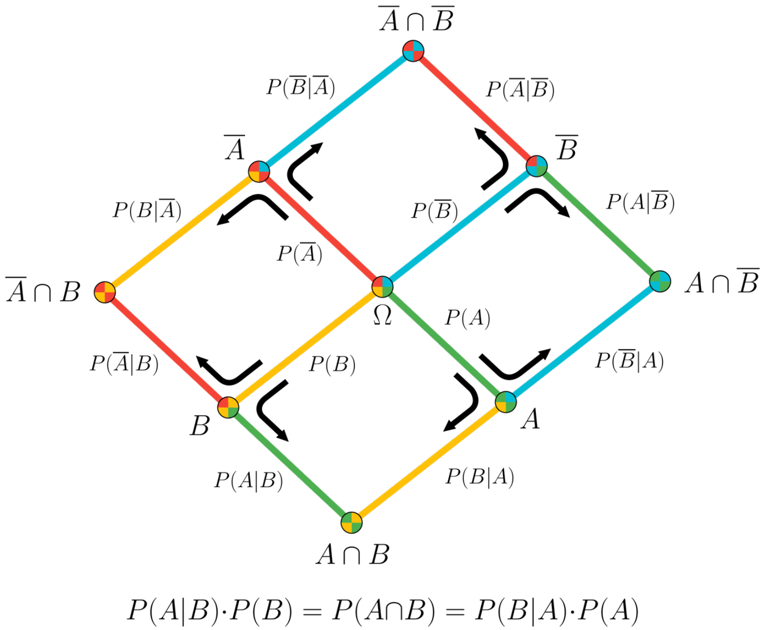
Teorema de Bayes



- El teorema de. Bayes es una proposición planteada por Thomas Bayes (1702 – 1763).

- Vincula la probabilidad de un evento A dado otro evento B, con la probabilidad de B dado A.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$



Teorema de Bayes

Sea $\{A_1, A_2, \ldots, A_i, \ldots, A_n\}$ un conjunto de sucesos mutuamente excluyentes y exhaustivos, y tales que la probabilidad de cada uno de ellos es distinta de cero (0). Sea B un suceso cualquiera del que se conocen las probabilidades condicionales $P(B|A_i)$. Entonces, la probabilidad $P(A_i|B)$ viene dada por la expresión:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}$$

donde:

- $P(A_i)$ son las probabilidades a priori,
- $P(B|A_i)$ es la probabilidad de B en la hipótesis A_i,
- $P(A_i|B)$ son las probabilidades a posteriori.

Thomas Bayes (1763)

Ejemplo: Naive Bayes (clasificación)

El modelo de Naive Bayes es un modelo probabilístico simple que se usa usualmente para reconocer patrones.

$$p(C|F_1,\dots,F_n)$$

- El modelo consiste en la generación de una variable aleatoria C, la cual representa una categoría y un conjunto de variables aleatorias $F = [f^1, f^2, \dots, f^n]$, las cuales representan las características de los objetos en cada categoría.

$$p(C|F_1,\dots,F_n) = \frac{p(C)\ p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)}.$$

- Este modelo asume que todas las características son independientes.

Cómputo Numérico

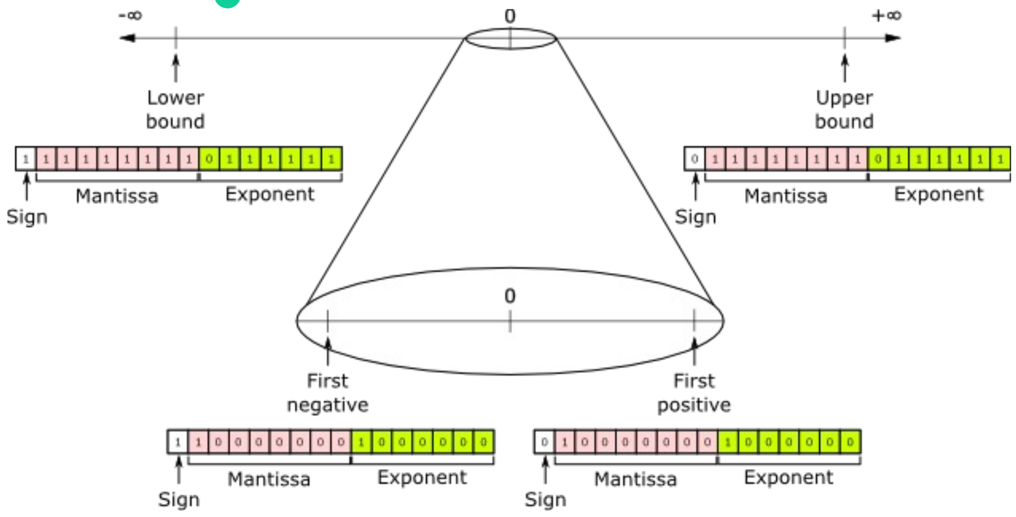
RETOS

Cómputo Numérico

La dificultad fundamental es la representación de números infinitos con número finitos.

- **Underflow:** Ocurre cuando los números cercanos a cero se redondean a cero. (Ejemplo: Eliminación Gaussiana).

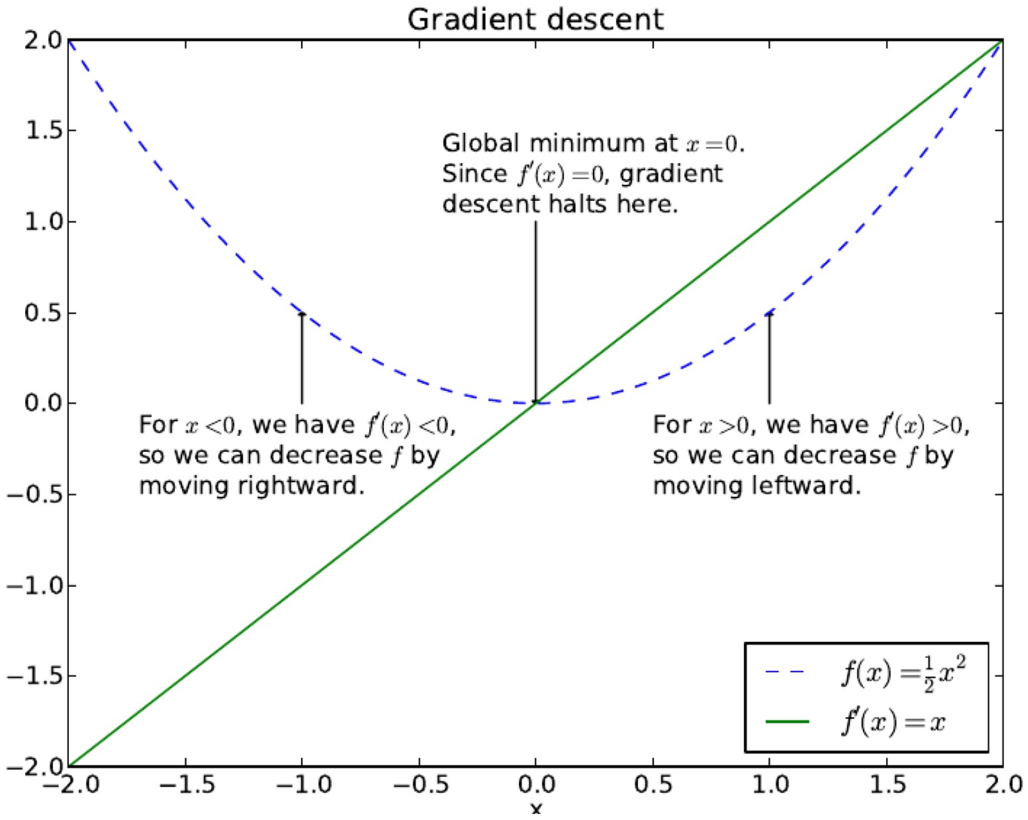
- **Overflow:** Ocurre cuando se tienen número cercanos a infinito y no se pueden realizar operaciones aritméticas.

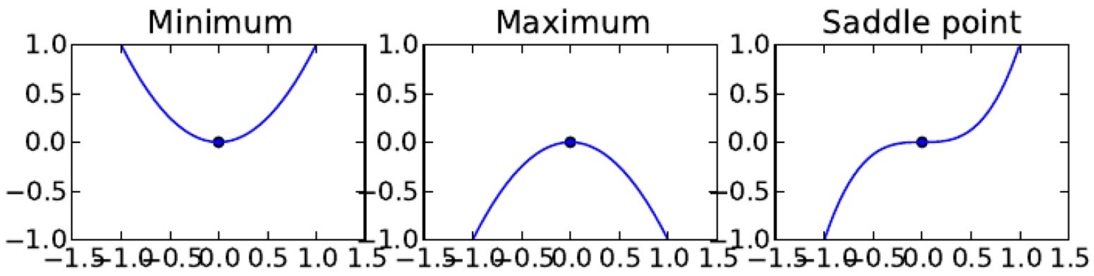


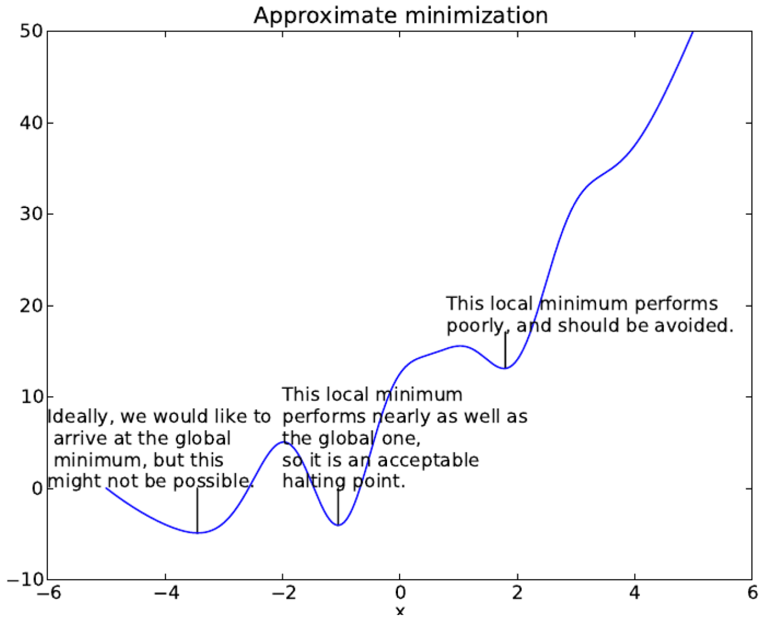
Optimización

El método del gradiente

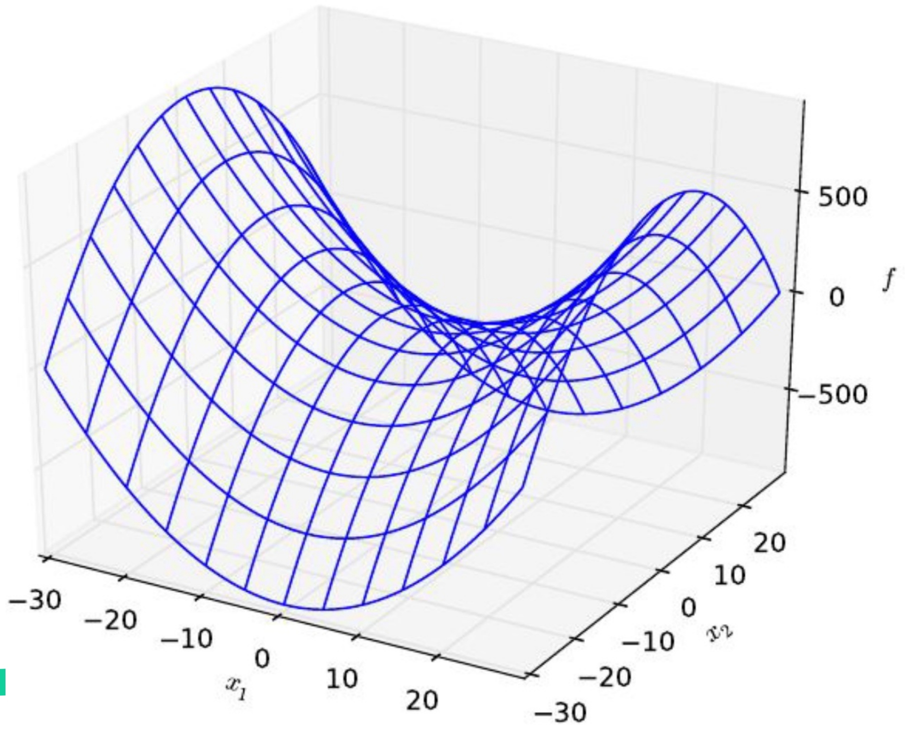
Optimización basada en el método del gradiente

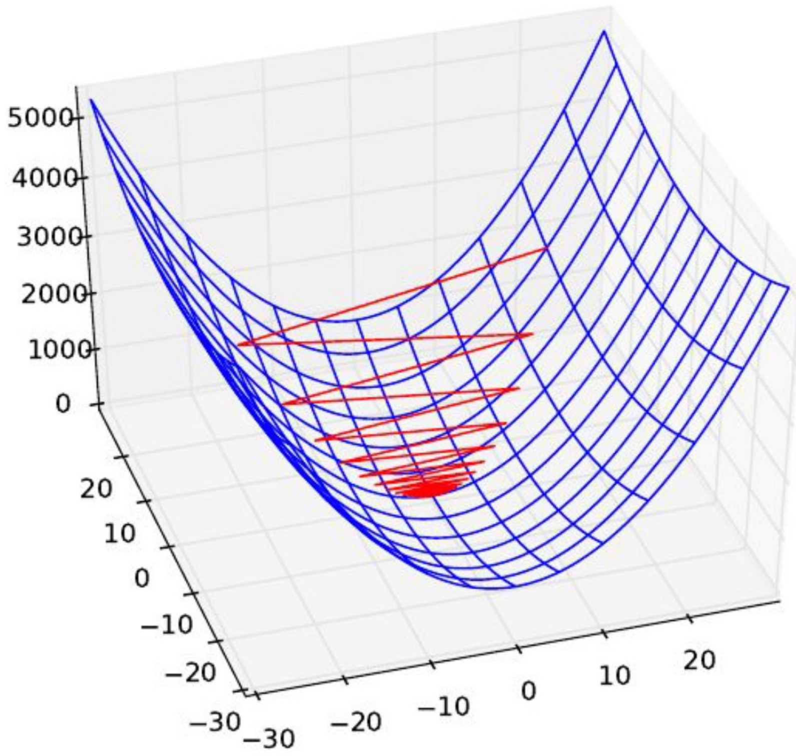






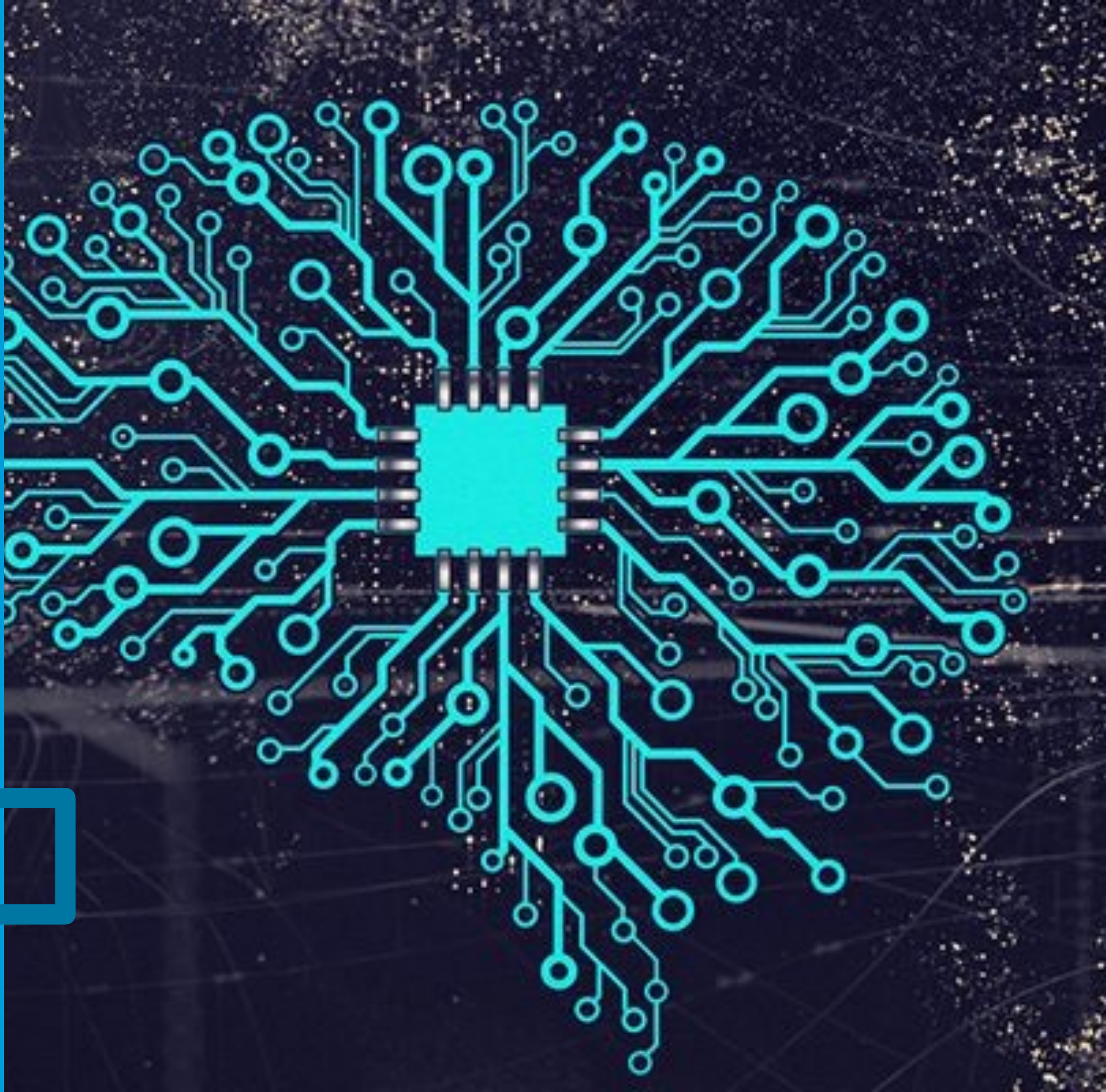
Optimización con restricciones





Conceptos Básicos de Algoritmos de Aprendizaje

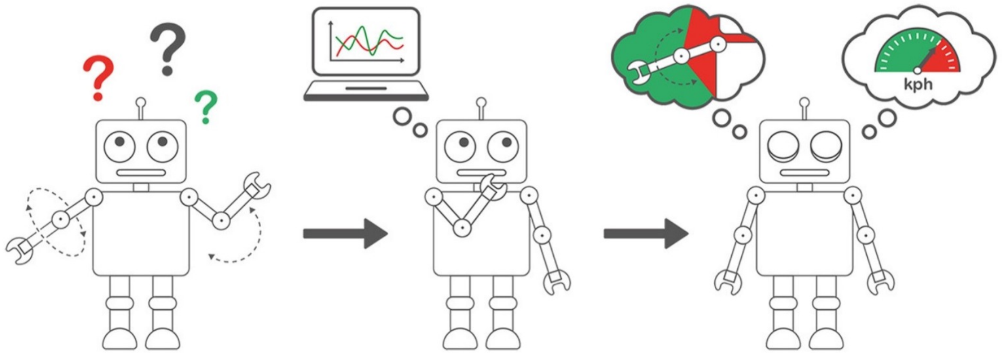
Inteligencia Artificial



Algoritmos de Aprendizaje

Un algoritmo de aprendizaje es capaz de aprender a partir de los datos.

"Se dice que un programa de computadora aprende de la experiencia **E** con respecto a alguna clase de tareas **T** y medida de desempeño **P**, si su desempeño en tareas en **T**, medido por **P**, mejora con la experiencia **E**" (Mitchell, 1997).



Una Tarea T

- Aprendizaje de máquinas: resolver problemas que son muy difíciles de resolver con programas fijos escritos por seres humanos.

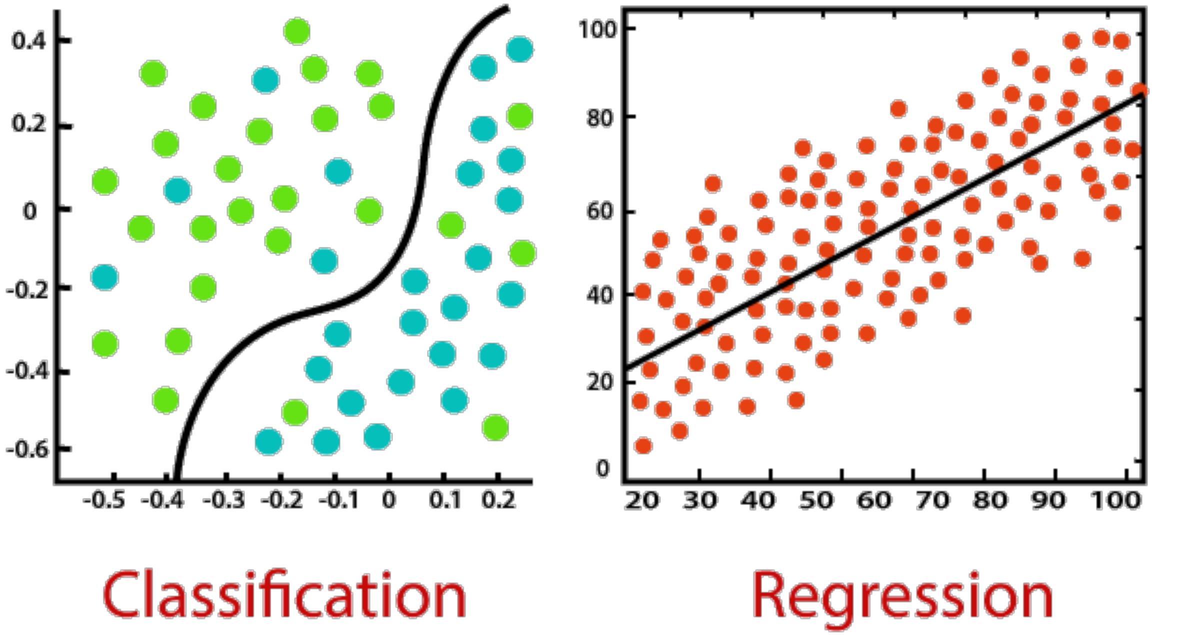
- Replicar comportamientos inteligentes

- Un comportamiento inteligente es capaz de completar ciertas tareas **T**.

Tipos de tareas

Muchas tareas pueden resolverse utilizando aprendizaje automático.

- Clasificación
- Regresión
- Transcripción
- Estimación de densidad
- Detección de anomalías
- Síntesis y muestreo
- Imputación de valores perdidos



Clasificación

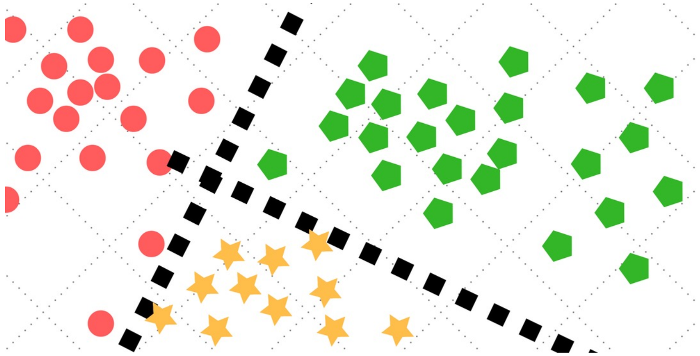
En este tipo de tareas, el algoritmo deberá generar una función del tipo:

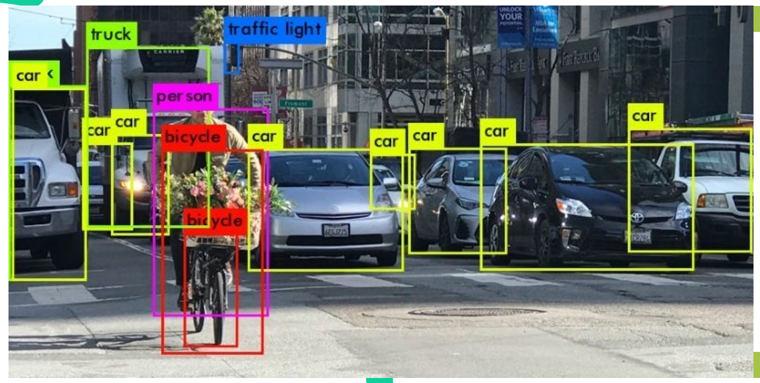
$$f : \mathbb{R}^n \to \{1, \ldots, k\}.$$

$f(x)$ puede ser interpretado como una estimación de la categoría a la que pertenece x.

Hay otras variantes de estos algoritmos donde $f(x)$ genera una distribución de probabilidad entre clases.

Ejemplo: Identificación (reconocimiento) de objetos en una imagen.



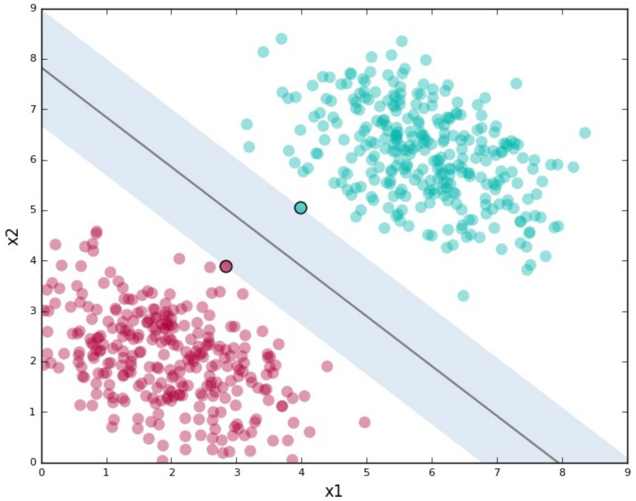


Regresión

En este tipo de tarea el algoritmo produce una función del tipo:

$$f : \mathbb{R}^n \rightarrow \mathbb{R}.$$

Es similar a la clasificación, excepto que el sistema de aprendizaje automático predice una salida con valor real en lugar de un código de categoría.



Ejemplo: Predicción del monto de reclamación que una persona ejercerá en un seguro, precio de un producto particular, etc…

Transcripción

- Este tipo de tarea es similar a la clasificación, excepto que la salida es una secuencia de símbolos.

Ejemplos: En el reconocimiento de voz y en la traducción automática, la salida de destino es una secuencia de palabras.

Detección de Anomalías

- En este tipo de tarea, el algoritmo de aprendizaje automático es identificar qué eventos son normales y qué eventos son inusuales.
- Un ejemplo de una tarea de detección de anomalías es la detección de fraudes con tarjetas de crédito.

Medición de Rendimiento, P

Para evaluar un algoritmo de aprendizaje, es necesario contar con una métrica cuantitativa. Esta métrica es especifica para una tarea **T**.

- Clasificación y transcripción
  - Precisión del modelo (Cantidad de respuestas correctas).
- Estimación de densidad
  - Probabilidad de que el modelo asigne algunos ejemplos.

Es muy importante evaluar el rendimiento del algoritmo con aplicaciones de la vida real, usualmente se reserva un conjunto de datos para pruebas (test data). Estos datos se separan de los datos de entrenamiento.

Rendimiento (pérdida, loss)

Usualmente en el contexto de rendimiento se hace referencia a la cuantificación de la pérdida (loss).

- Costo asociado a un evento particular

- Minimizar la pérdida
  - Error: proporción de ejemplos en los cuales se producen salidas incorrectas.
  - Al minimizar el error se incrementa la precisión.

Experiencia, E

Usualmente la experiencia se relaciona con permitir al algoritmo de aprendizaje observar múltiples ejemplos, caso más simple.

- Experiencias interactivas
  - Aprendizaje activo: Adiciones a los datos originales
  - Aprendizaje reforzado: El algoritmo interactúa con el ambiente y obtiene retroalimentación.

Datasets

Colección de ejemplos, cada ejemplo es una colección de observaciones (características).

- Se recopilan en un momento o lugar distinto.

Ejemplo:

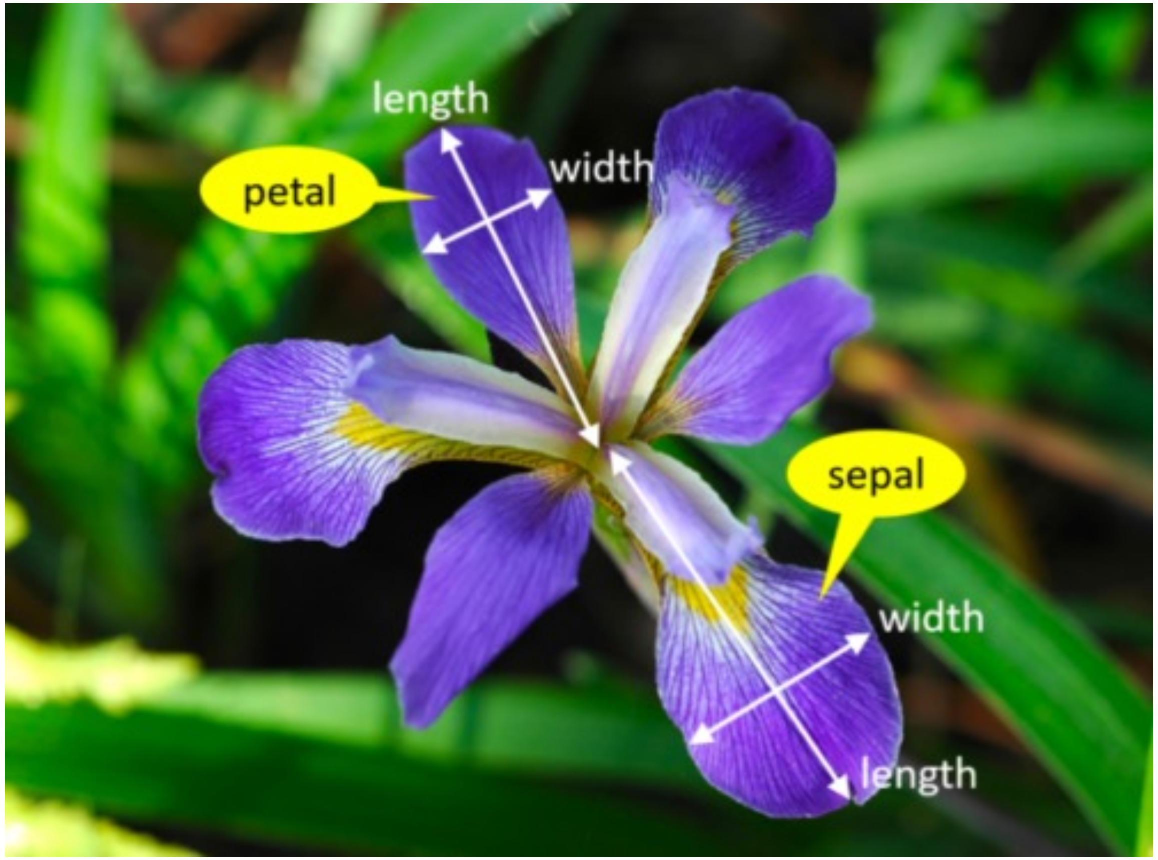
- Reconocimiento de objetos en imágenes, cada ejemplo es una imagen.
- Los características dentro del ejemplos son los valores de brillo de cada pixel en la imagen.

Matriz de diseño: contiene un ejemplo distinto en cada fila y una característica distinta en cada columna.

Dataset IRIS
150 plantas con 4 características.

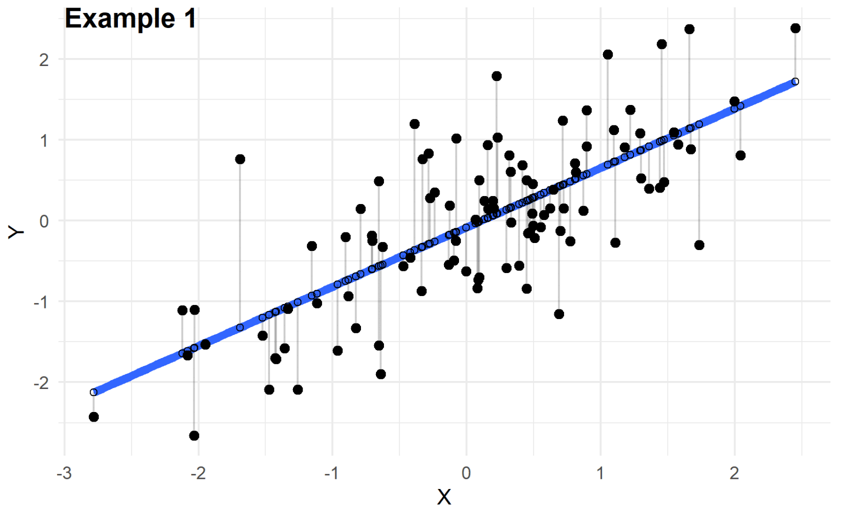
Matriz de diseño $\mathbf{X} \in \mathbb{R}^{150 \times 4}$

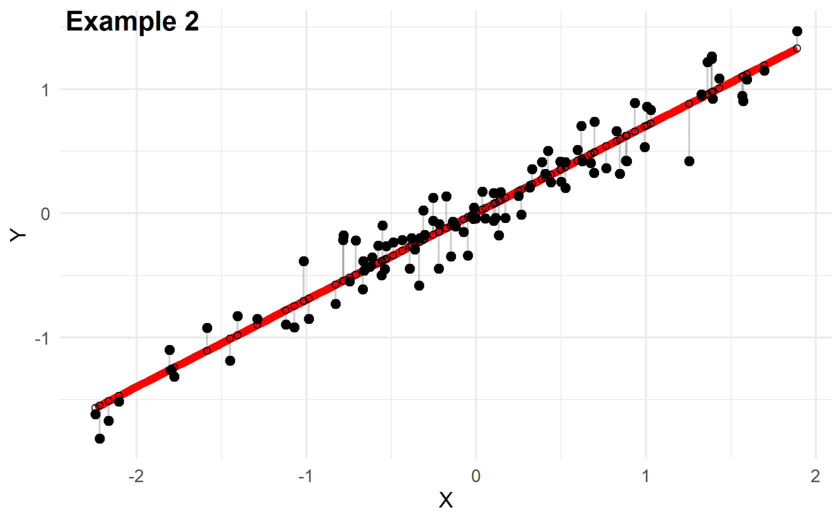
	X	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	1	5.1	3.5	1.4	0.2	setosa
2	2	4.9	3.0	1.4	0.2	setosa
3	3	4.7	3.2	1.3	0.2	setosa
4	4	4.6	3.1	1.5	0.2	setosa
5	5	5.0	3.6	1.4	0.2	setosa
6	6	5.4	3.9	1.7	0.4	setosa
7	7	4.6	3.4	1.4	0.3	setosa
8	8	5.0	3.4	1.5	0.2	setosa
9	9	4.4	2.9	1.4	0.2	setosa
10	10	4.9	3.1	1.5	0.1	setosa
11	11	5.4	3.7	1.5	0.2	setosa
12	12	4.8	3.4	1.6	0.2	setosa
13	13	4.8	3.0	1.4	0.1	setosa
14	14	4.3	3.0	1.1	0.1	setosa
15	15	5.8	4.0	1.2	0.2	setosa
16	16	5.7	4.4	1.5	0.4	setosa
17	17	5.4	3.9	1.3	0.4	setosa
18	18	5.1	3.5	1.4	0.3	setosa
19	19	5.7	3.8	1.7	0.3	setosa



Generalización

- En ML, la generalización es la habilidad que tiene un modelo para desempeñarse bien en inputs no observados previamente:
    1. Reducir el error (minimización)
    2. Hacer que el error entre el set de entrenamiento y set de prueba sea pequeño.

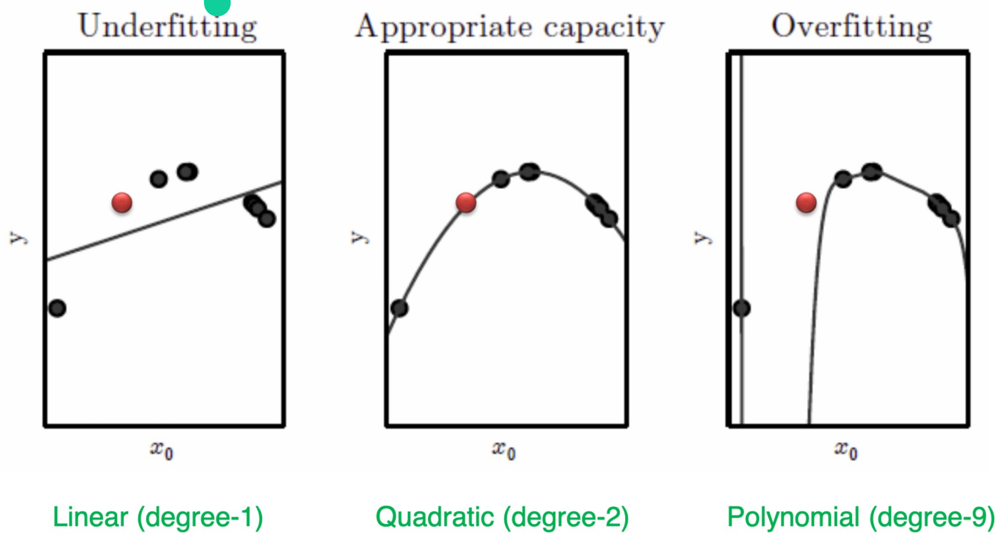




Generalización

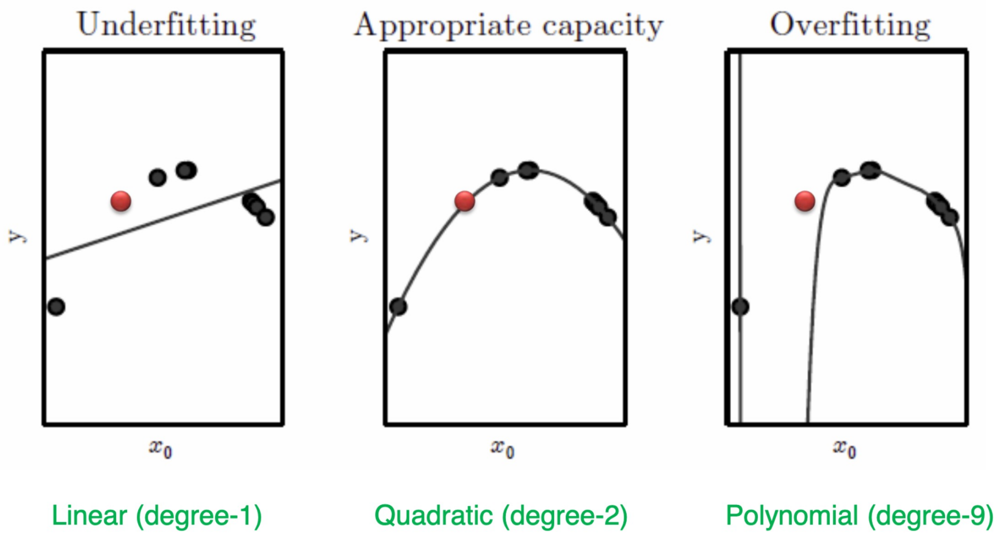
1. **Underfitting (sub-estimación):** Ocurre cuando el modelo no es capaz de obtener un error lo suficiente en el set de entrenamiento.

2. **Overfitting (sobre-estimación):** Ocurre cuando existe un error muy grande entre el set de entrenamiento y el set de prueba.

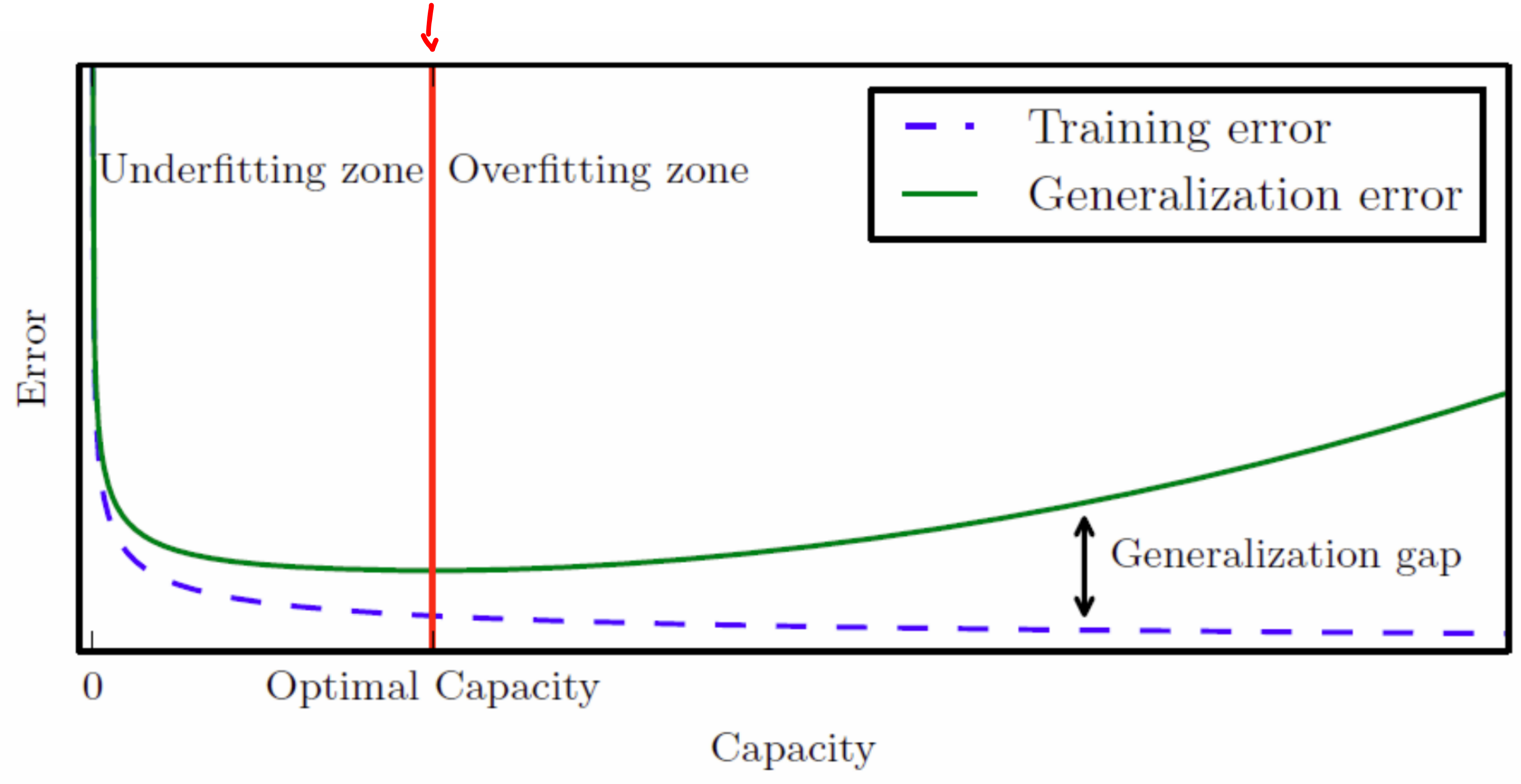


Relación entre capacidad y Error

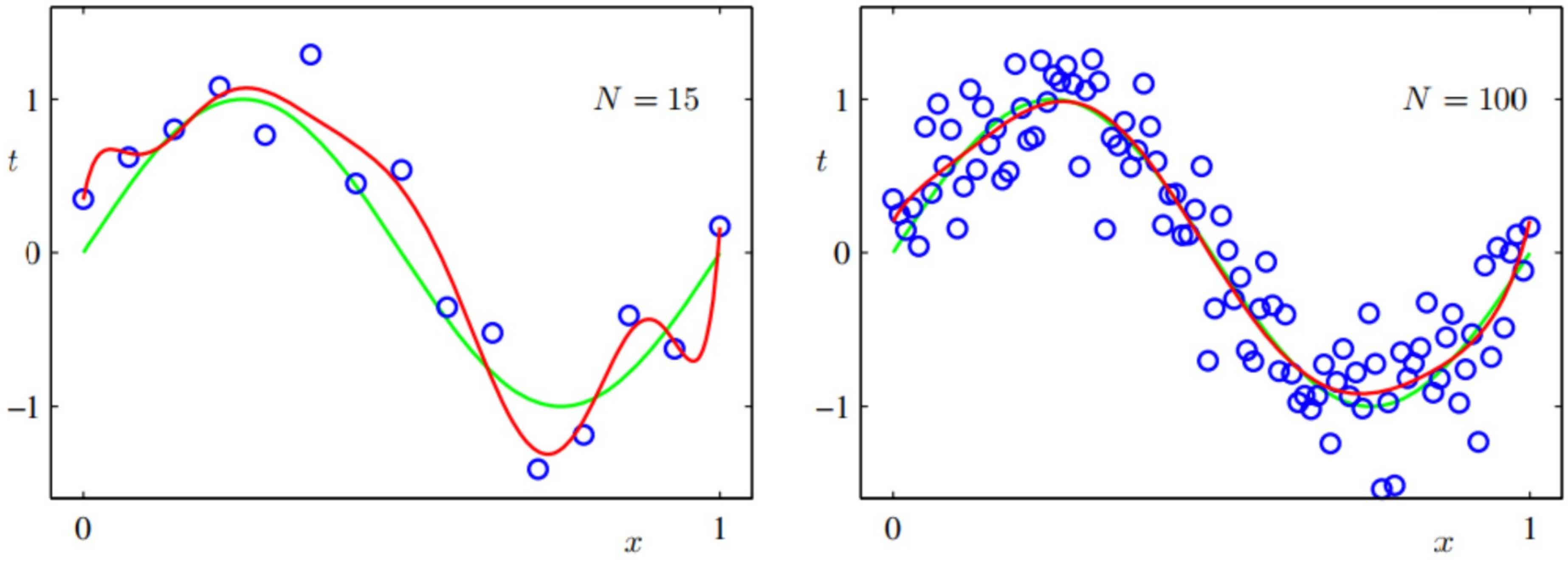
La capacidad se define como la habilidad de un modelo para descubrir una función para ajustar a una función un conjunto de datos de entrada.



Relación entre Capacidad y Error



Tamaño del set de entrenamiento y generalización



Polynomial (degree-9)

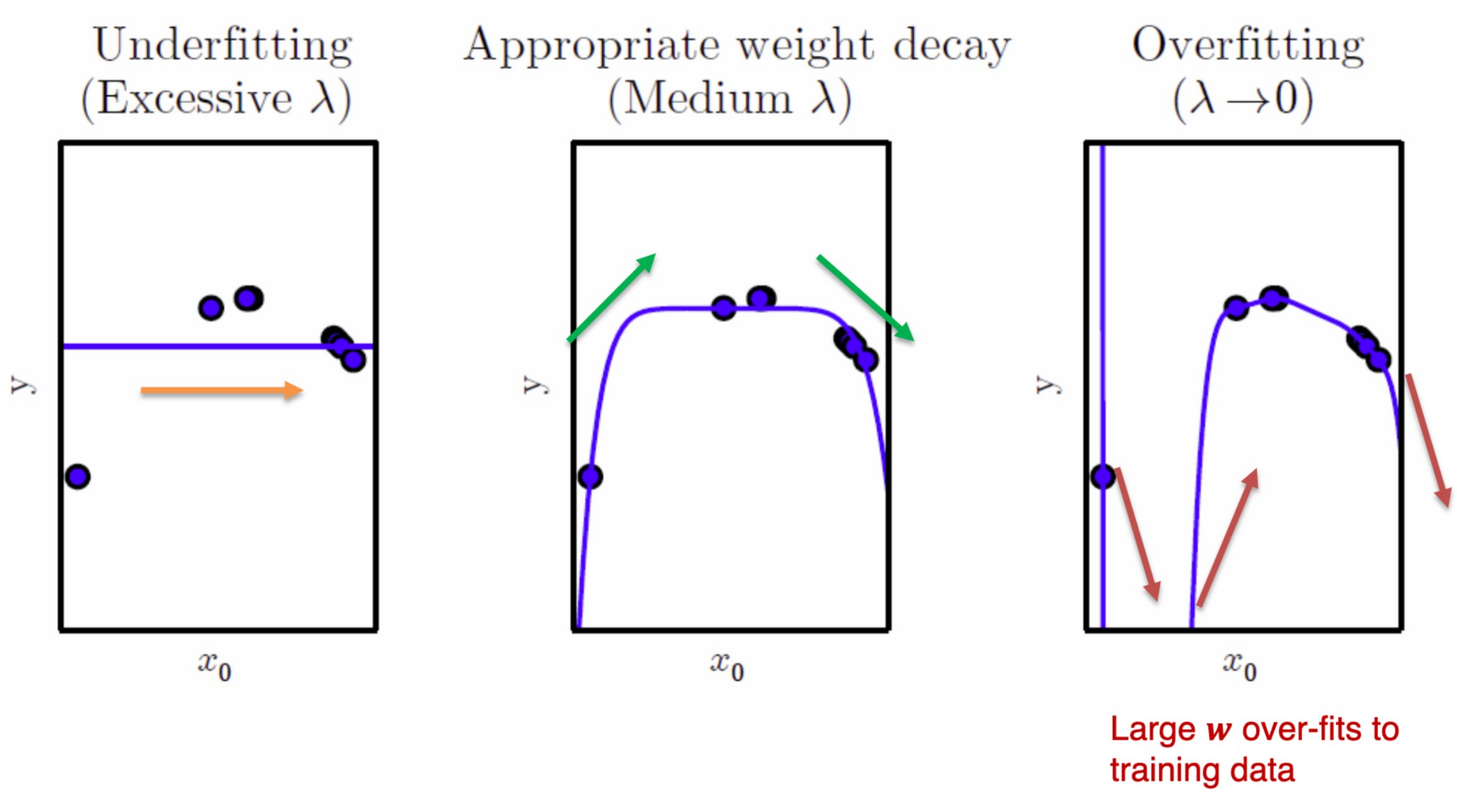
Regularización

- La regularización es cualquier modificación que previene el sobreajuste (overfitting).
- Controla el rendimiento de un algoritmo.
  - El objetivo es reducir el error del set de prueba pero no el de entrenamiento.
- Ejemplo: Decaimiento de peso para una regresión

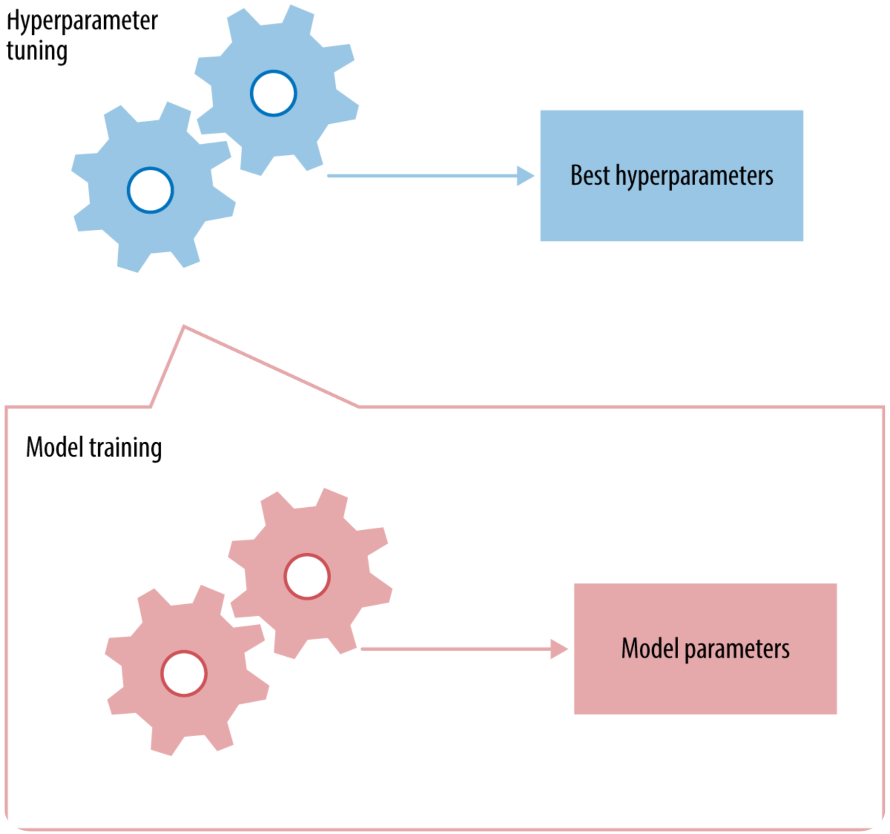
$$J(w) = MSE_{train} + \lambda w^T w$$

  - J(·): Función de costo a minimizar durante el entrenamiento
  - λ: Factor de control para un peso pequeño ($\lambda \geq 0$)
- Balance: Ajuste de los datos para entrenamiento para pesos pequeños.

Modelo de regresión de grado 9 con Decaimiento de Peso



Parámetros VS Hiperparámetros



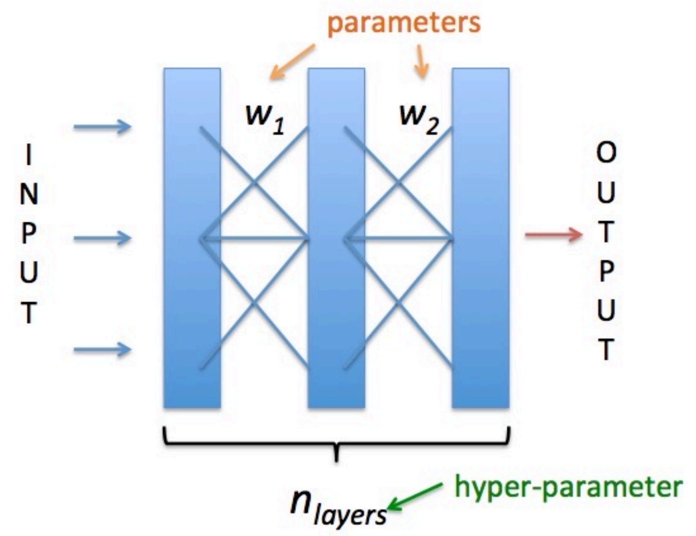
- Los hiperparámetros son propiedades de alto nivel asociados a un modelo
    - Decide la capacidad de un modelo (complejidad)
    - No se aprenden durante el proceso de entrenamiento
    - Ejemplo: Grado de un modelos de regresión, decaimiento de peso

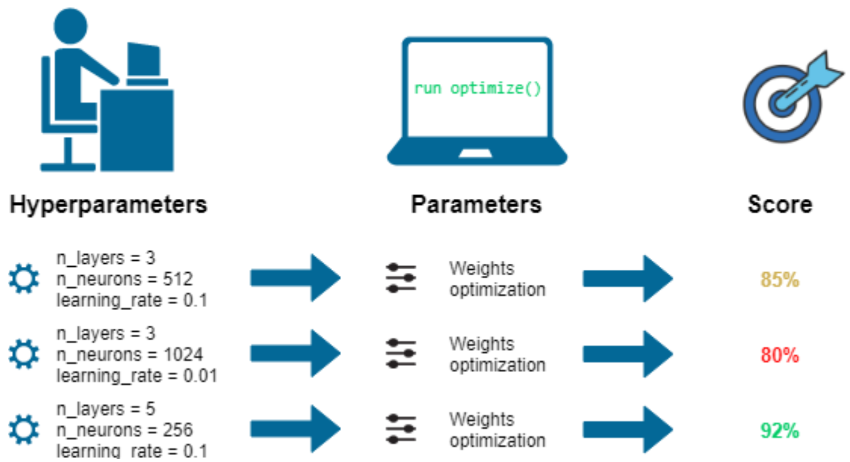
- Los parámetros son propiedades de los datos de entrenamiento.
    - Se aprenden durante el proceso de entrenamiento por el modelo de ML.
    - Ejemplo: Pesos de un modelo d regresión lineal

Set de entrenamamiento

- Establecer hiperparámetros durante el proceso de entrenamiento no es apropiado.
  - Estos parámetros se establecen para producir un sobreajuste (overfitting).
    - Ejemplo: un grado alto en el modelo de regresión

- El set de prueba no se considera para el entrenamiento ni para la selección del modelo. (establecer hiperparámetros)

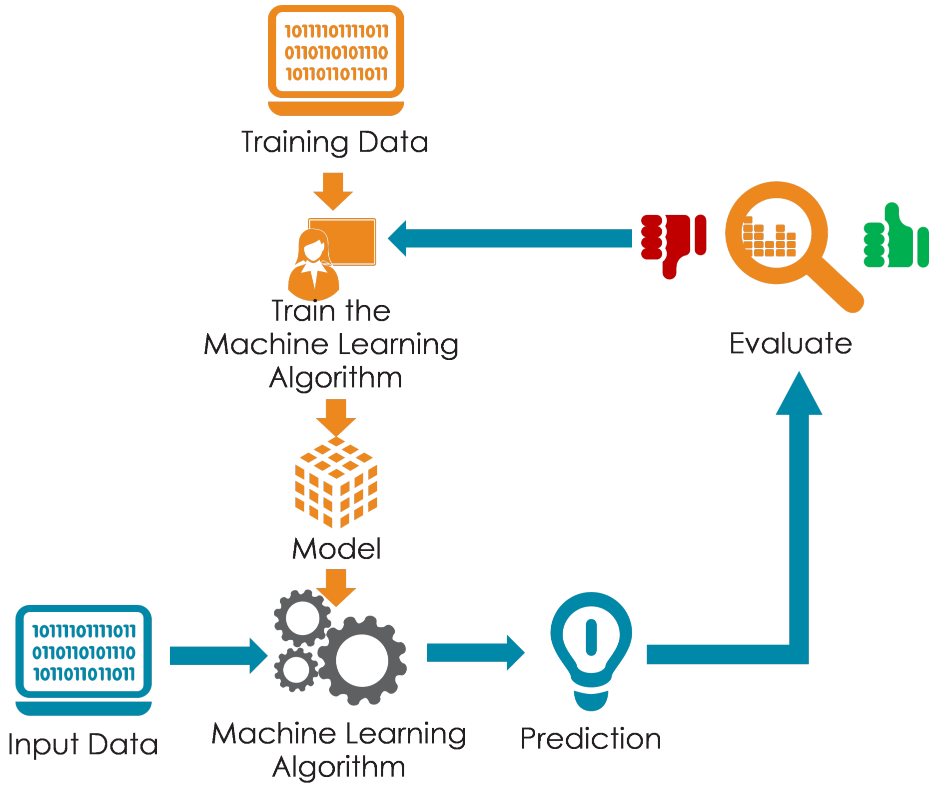




Set de validación

- Es necesario un set de validación que no sea visible para el algoritmo de entrenamiento.

  - División de la totalidad de los datos en dos conjuntos



Validación Cruzada