



UNIVERSIDAD
NACIONAL
DE COLOMBIA

VISIÓN ARTIFICIAL

2020 – 02



Github: https://github.com/jwbranch/Vision_Artificial
MinasLAP: <https://minaslap.net/course/view.php?id=510>

JOHN W. BRANCH

Profesor Titular

Departamento de Ciencias de la Computación y de la Decisión

Director del Grupo de I+D en Inteligencia Artificial – GIDIA

jwbranch@unal.edu.co

ESTEBAN BRITO

Monitor

dbrito@unal.edu.co

LOS MATERIALES DE ESTA ASIGNATURA, SE BASAN EN LA EVOLUCIÓN Y ELABORACIÓN DE ANTERIORES

SEMESTRES, EN LOS CUALES HAN CONTRIBUIDO Y COLABORADO, LOS PROFESORES DIEGO PATIÑO, CARLOS

MERA, PEDRO ATENCIO, ALBERTO CEBALLOS Y JAIRO RODRÍGUEZ, A LOS CUALES DAMOS CRÉDITO.

METODOLOGÍA ENSEÑANZA – APRENDIZAJE

Sesiones Remotas vía Google.Meet Sincrónicas y Asincrónicas

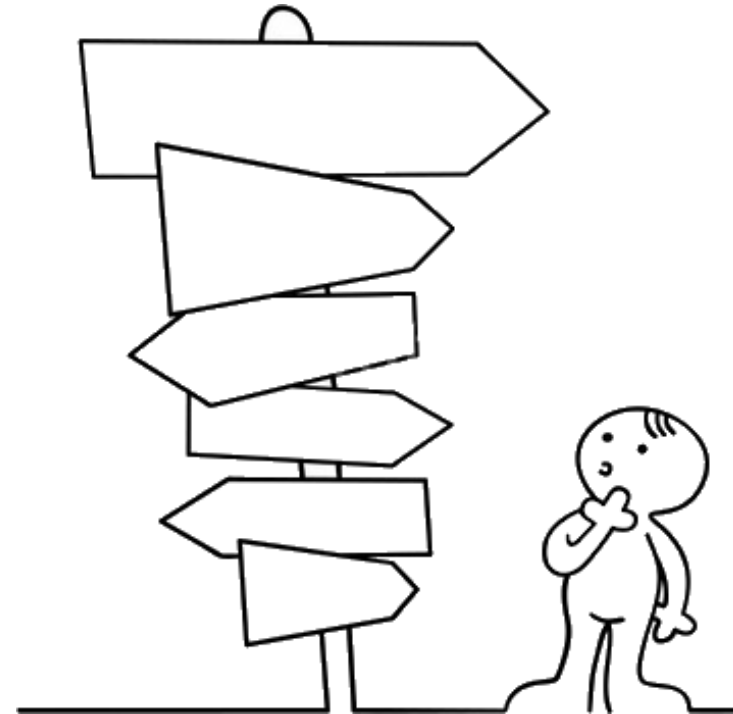
El aprendizaje sincrónico involucra estudios online a través de una plataforma. Este tipo de aprendizaje sólo ocurre en línea. Al estar en línea, el estudiante se mantiene en contacto con el docente y con sus compañeros. Se llama aprendizaje sincrónico porque la plataforma permite que los estudiantes pregunten al docente o compañeros de manera instantánea a través de herramientas como el chat o el video chat.

El aprendizaje asincrónico puede ser llevado a cabo online u offline. El aprendizaje asincrónico implica un trabajo de curso proporcionado a través de la plataforma o el correo electrónico para que el estudiante desarrolle, de acuerdo a las orientaciones del docente, de forma independiente. Un beneficio que tiene el aprendizaje asincrónico es que el estudiante puede ir a su propio ritmo.

EN LA CLASE DE HOY ...

Reconocimiento de Patrones

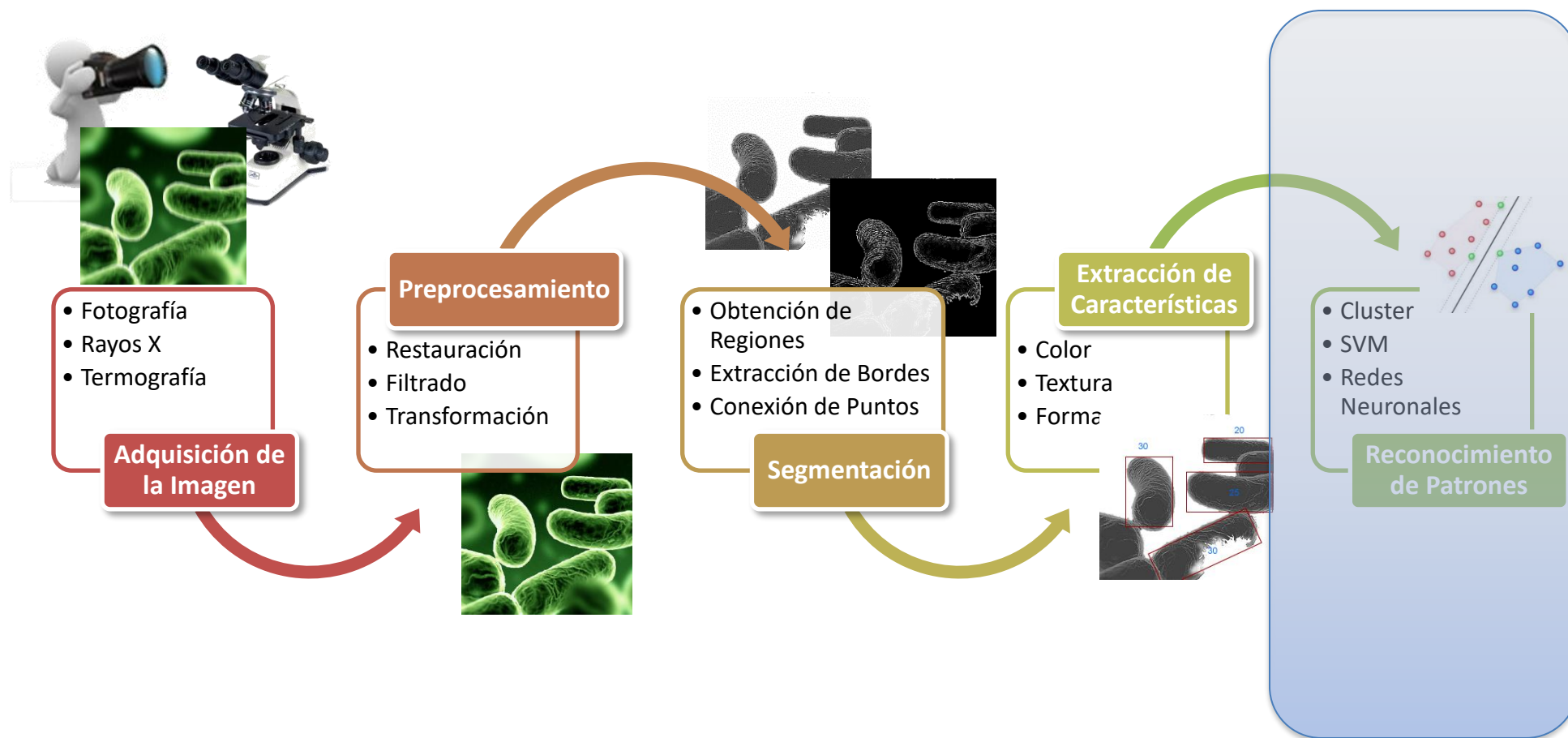
- 🌀 Introducción
- 🌀 Métodos Supervisados
- 🌀 Métodos No Supervisados





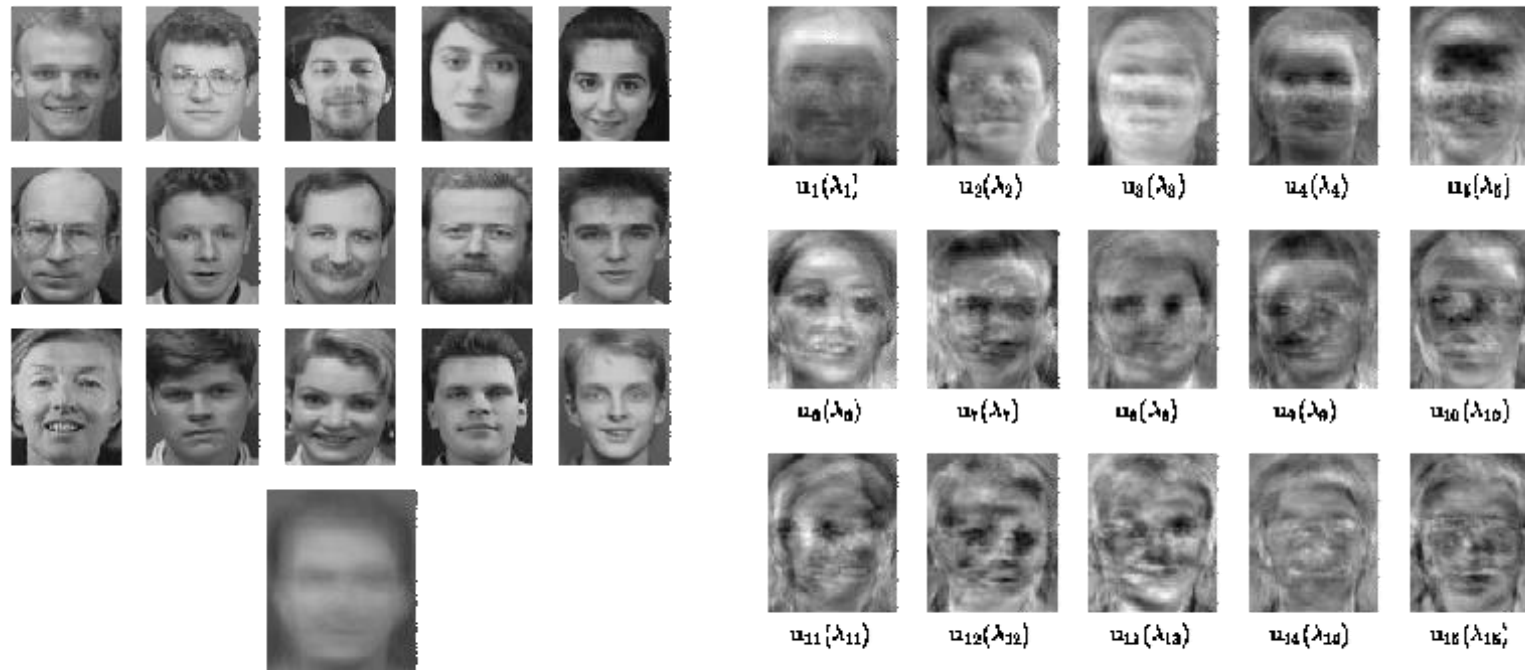
A MODO DE RESUMEN ...

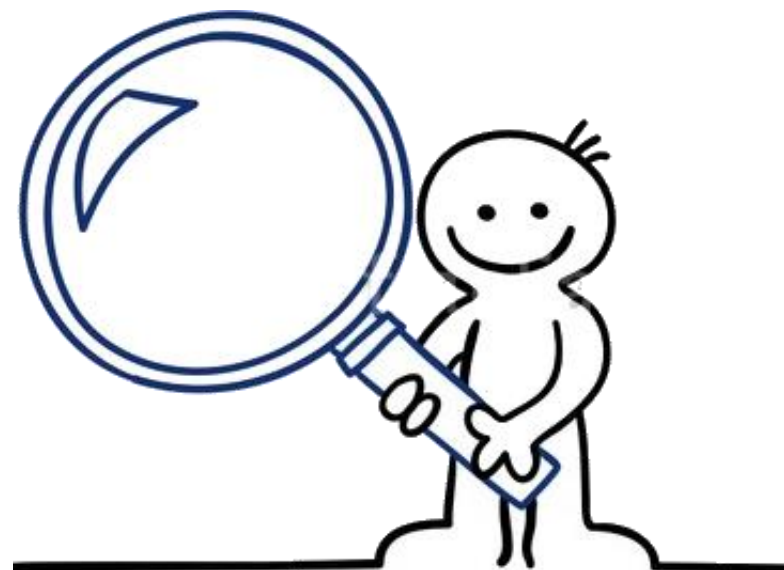
ETAPAS DE UN SISTEMA DE VISIÓN ARTIFICIAL



Ejemplo con Imágenes para Reconocimiento de un Patrón en las imágenes

Eigenfaces for Recognition, Turk, M. & Pentland, A. ,
Journal of Cognitive Neuroscience, 3, 71-86, 1991.



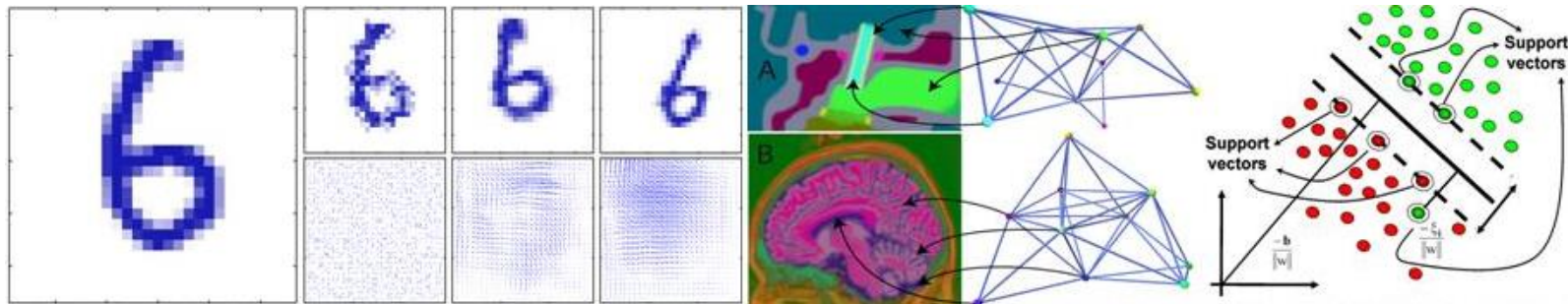


RECONOCIMIENTO DE PATRONES

Reconocimiento de Patrones

🦋 INTRODUCCIÓN

- 🌀 El *Reconocimiento de Patrones* es la última etapa dentro de un sistema de visión artificial, en la que a partir de las características encontradas, los posibles objetos se **CLASIFICAN** en dos o más clases.



- 🌀 **Clasificar** (o reconocer) significa, en este contexto, asociar a clases (o prototipos) una serie de elementos (u objetos). Esta asociación se realiza en base a las características o propiedades de los objetos.

Reconocimiento de Patrones

CONSIDERACIONES

Las características de las regiones u objetos segmentados se representan usando **vectores de características** normalizados.

Las características usadas para el reconocimiento deben ser cuidadosamente seleccionadas (p. ej. elección de características invariantes a transformaciones geométricas)

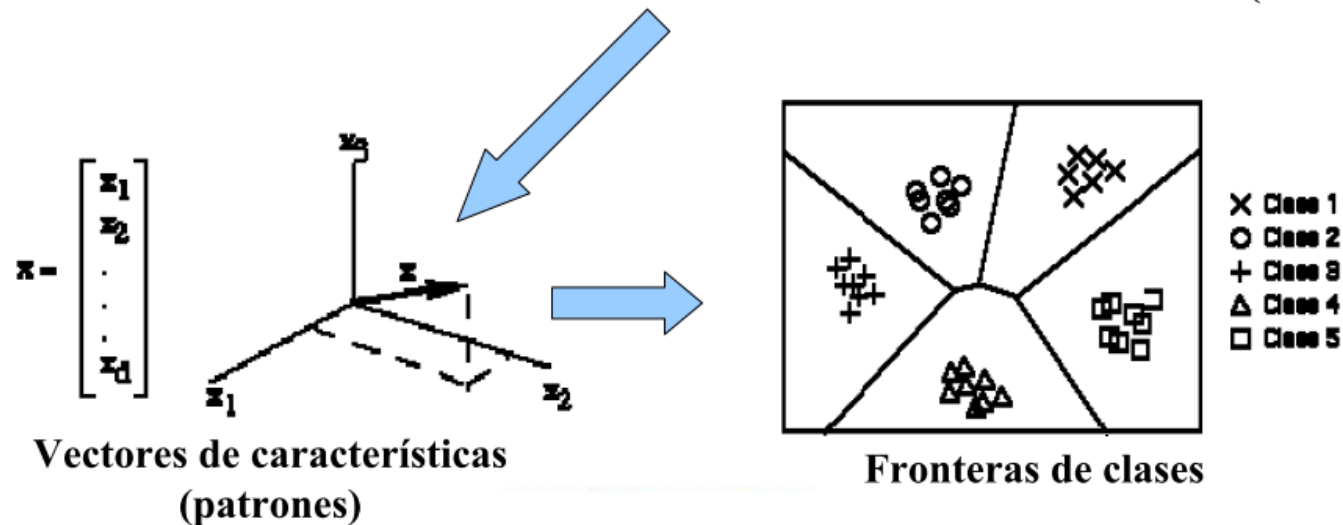
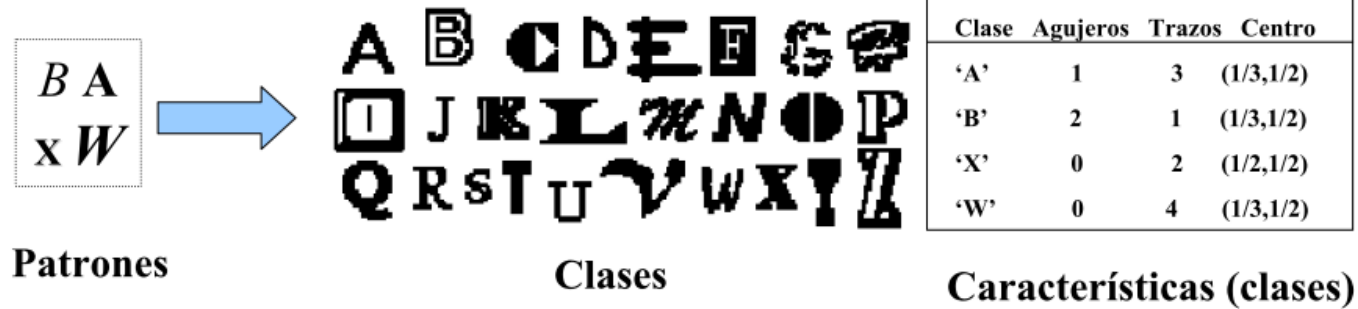
Reconocer o clasificar no son tareas fáciles: las clases pueden no estar correctamente definidas, la información sobre los objetos a clasificar puede ser incompleta.

La interpretación de imágenes (o escenas) requiere el uso de modelos y técnicas de **Inteligencia Artificial**

Métodos de clasificación diferentes → clasificaciones diferentes.

Reconocimiento de Patrones

🦋 EJEMPLO DE RECONOCIMIENTO DE CARACTERES



Reconocimiento de Patrones

🦋 **IMPORTANTE:** Si los descriptores elegidos son adecuados, objetos similares tendrán patrones próximos en el espacio de características.



Patrones que describen objetos de una misma clase, presentan características similares.

Patrones que describen objetos de diferentes clases presentan características diferenciadas.

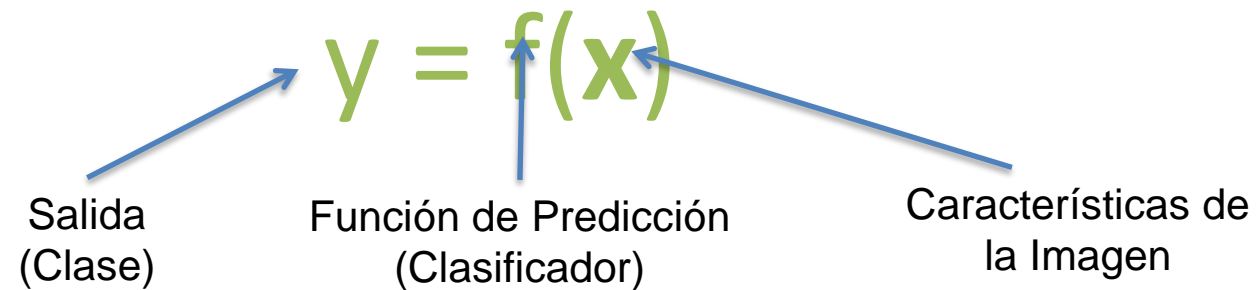
Reconocimiento de Patrones

🚀 MODELO GENERAL DE UN CLASIFICADOR

- 🌀 Aplicar una función de predicción en una representación de las características de la imagen para obtener el resultado deseado

$$f(\text{[apple image]}) = \text{"manzana"}$$

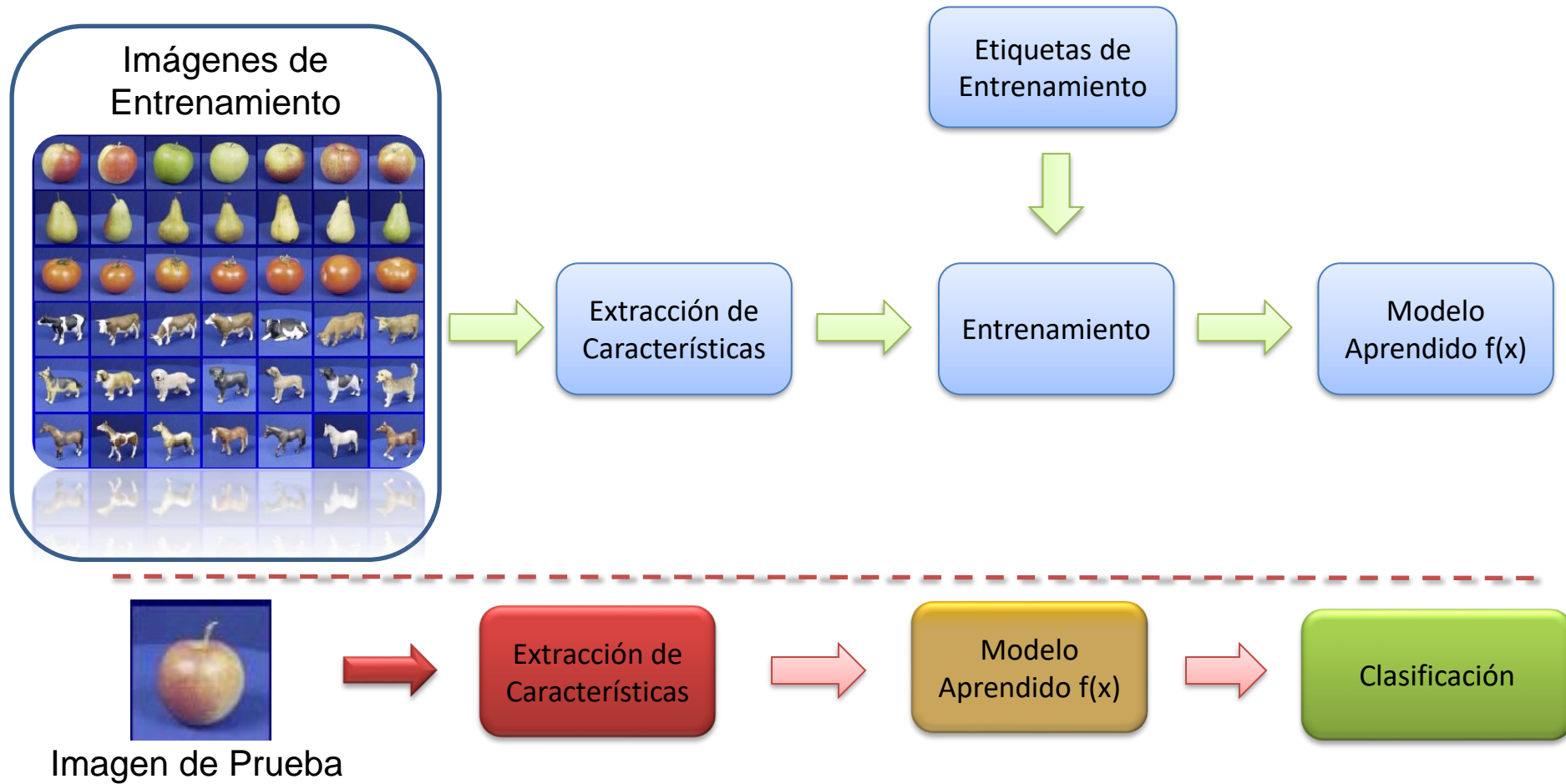
$$f(\text{[cow image]}) = \text{"vaca"}$$



- 🌀 **Entrenamiento:** dado un conjunto de ejemplos $\{(x_1, y_1), \dots, (x_n, y_n)\}$, calcular la predicción de la función f , **REDUCIENDO AL MÍNIMO EL ERROR DE PREDICCIÓN** en el conjunto de entrenamiento

Reconocimiento de Patrones

MODELO GENERAL DE UN CLASIFICADOR

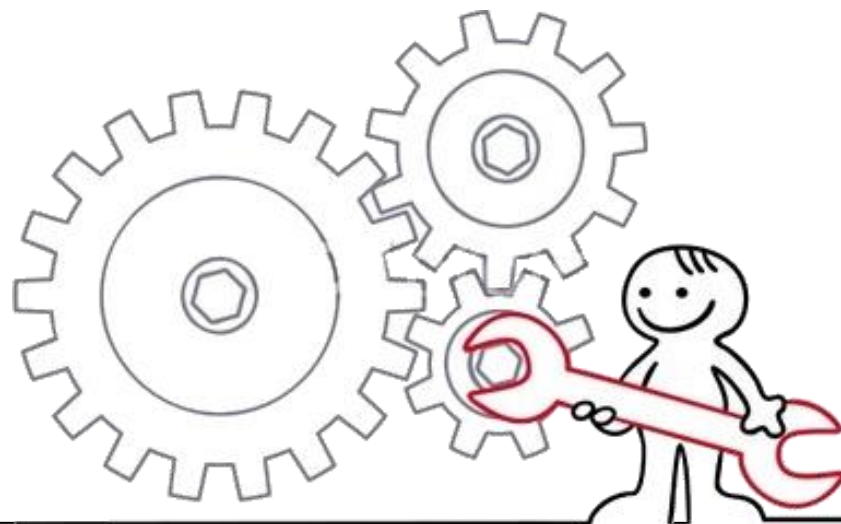


Reconocimiento de Patrones

TIPOS DE CLASIFICADORES

Atendiendo a la información que se proporciona en el proceso de construcción del clasificador se puede hablar de dos tipos de clasificadores: supervisados y no supervisados:

- 🌀 **Clasificadores Supervisados:** el conjunto de entrenamiento es dividido por un supervisor externo, en las diferentes clases ya conocidas en las que se desea clasificar, así el clasificador aprende las características que definen cada clase.
- 🌀 **Clasificadores NO Supervisados:** sin la necesidad de ningún supervisor externo, el clasificador determina las clases que representan los datos de entrenamiento.



TÉCNICAS DE RECONOCIMIENTO DE PATRONES

Técnicas Supervisadas

- a. Regresión Logística
- b. Árbol de Decisión
- c. K-NN
- d. Máquina de Vectores de Soporte
- e. Algoritmo XGBoost
- f. Redes Neuronales

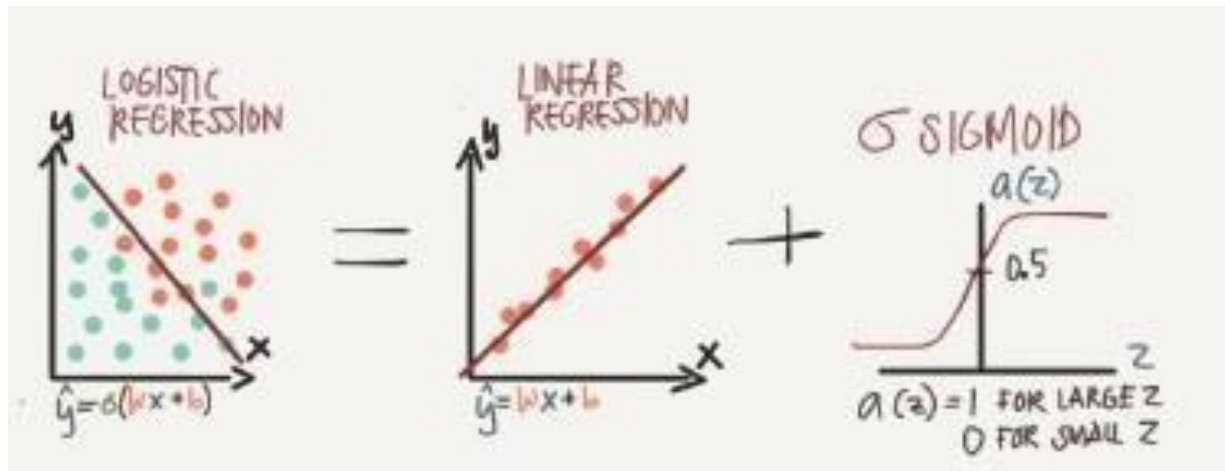
Técnicas NO Supervisadas

- g. Clustering (agrupamiento)

Regresión Logística

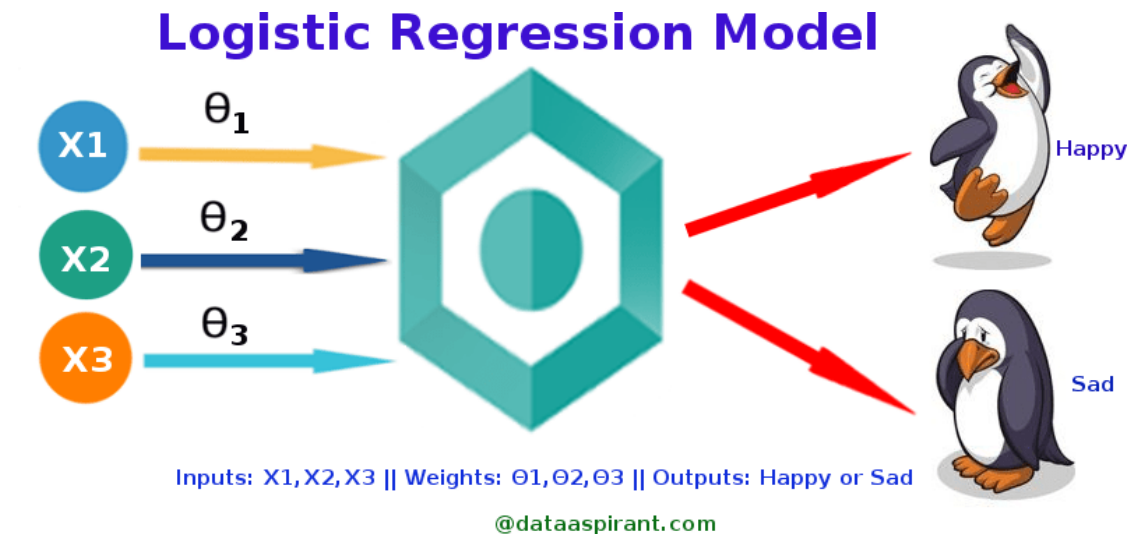
DEFINICIÓN:

La regresión logística es un modelo de clasificación que se utiliza para **predecir la probabilidad** $P(y = 1)$ **de una variable dependiente categórica** en función de x . Así, la variable y es una variable binaria codificada como 1 (positivo, éxito, etc.) o 0 (negativo, falla, etc.).



Algunos ejemplos de aplicación:

- E-mail: spam/no spam
- Transacciones en línea: fraude/no fraude
- Tumores: maligno/no maligno



Regresión Logística

VENTAJAS:

- Es un modelo de clasificación eficiente y simple.
- No es necesario disponer de grandes recursos computacionales.
- Los resultados son altamente interpretables.



DESVENTAJAS:

- Imposibilidad de resolver directamente problemas no lineales.
- La variable objetivo debe ser linealmente separable.
- La regresión logística no es uno de los algoritmos más potentes que existen.

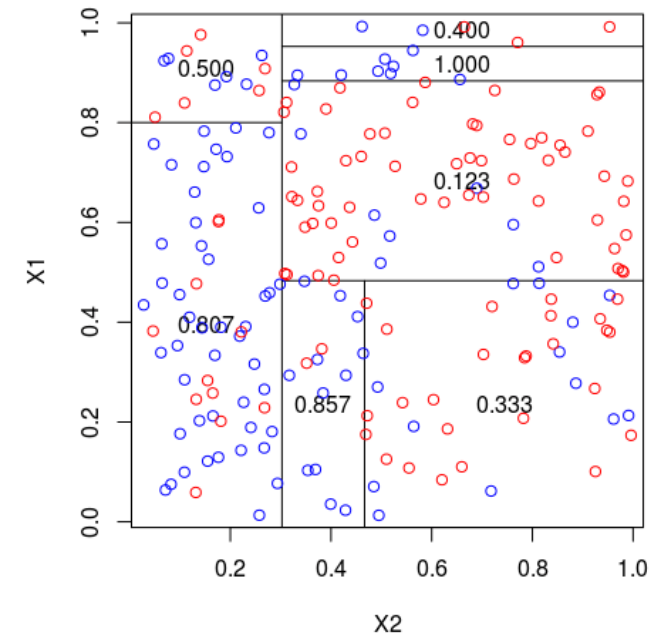
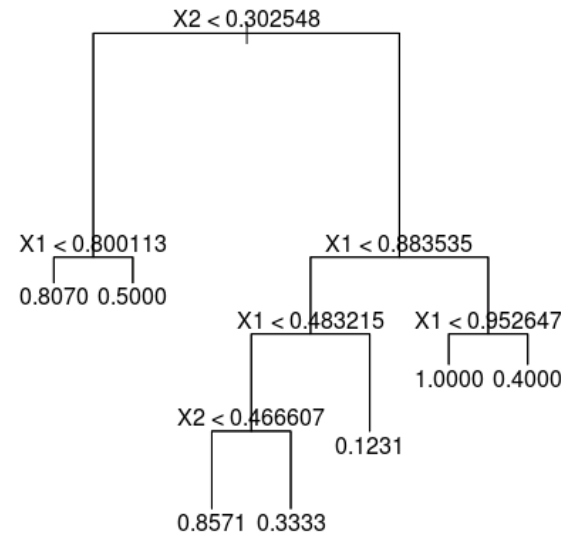


Árbol de Decisión

Los Árboles de Decisión pueden resumirse con los siguientes puntos:

- Los árboles de decisión son modelos predictivos que utilizan un conjunto de reglas binarias para calcular un valor objetivo.
- Cada árbol individual es un modelo bastante simple que tiene ramas, nodos y hojas.

Un árbol de decisión **consisten en estimar un resultado haciendo una serie de preguntas a los datos**, cada pregunta estrechando nuestros posibles valores hasta que el modelo tenga la suficiente confianza como para hacer una sola predicción. El orden de la pregunta así como su contenido son determinados por el modelo. Además, las preguntas formuladas están todas en forma de Verdadero/Falso.



Árbol de Decisión

Durante el entrenamiento, el modelo se ajusta con cualquier dato histórico que sea relevante para el dominio del problema y el verdadero valor que queremos que el modelo aprenda a predecir. El modelo aprende cualquier relación entre los datos y la variable objetivo.

Después de la fase de entrenamiento, el árbol de decisión produce un árbol calculando las mejores preguntas, así como su orden para hacer las estimaciones más precisas posibles. Cuando se quiere hacer una predicción se debe proporcionar al modelo el mismo formato de datos para hacer una predicción. La predicción será una estimación basada en los datos del entrenamiento en el que se ha entrenado.

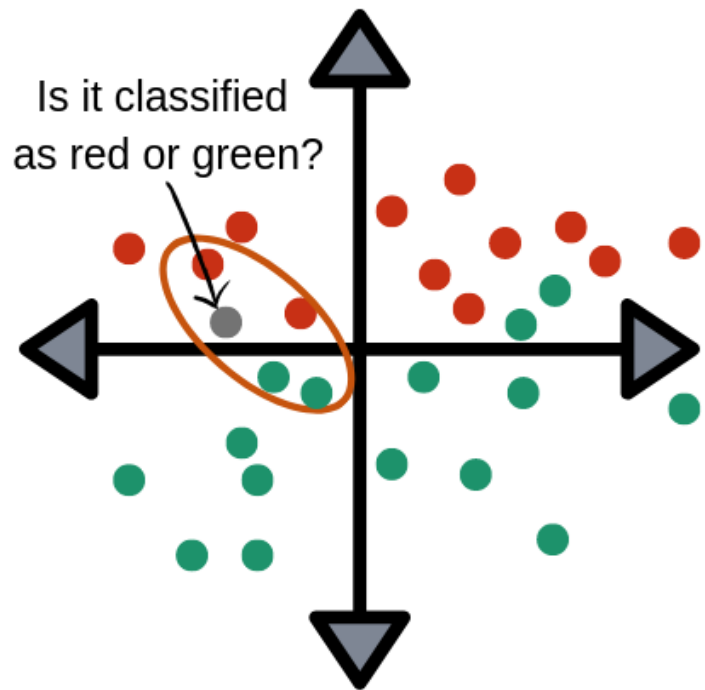
DECISION TREE REGRESSION

Similar to decision tree classification, however uses Mean Squared Error or similar metrics instead of cross-entropy or Gini impurity to determine splits.



Clasificador KNN

La idea básica sobre la que se fundamenta este paradigma es que un nuevo caso se va a clasificar en la clase más frecuente a la que pertenecen sus K vecinos más cercanos



COMIENZO

Entrada: $D = \{(\mathbf{x}_1, c_1), \dots, (\mathbf{x}_N, c_N)\}$

$\mathbf{x} = (x_1, \dots, x_n)$ nuevo caso a clasificar

PARA todo objeto ya clasificado (x_i, c_i)

calcular $d_i = d(\mathbf{x}_i, \mathbf{x})$

Ordenar $d_i (i = 1, \dots, N)$ en orden ascendente

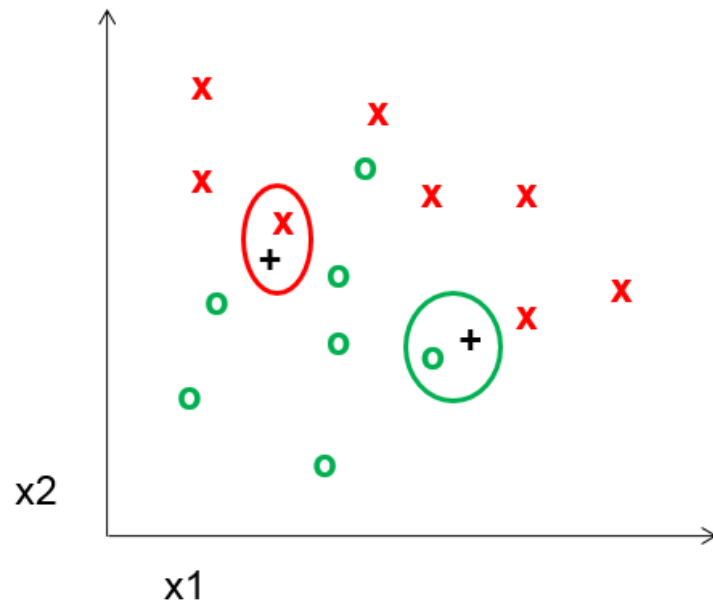
Quedarnos con los K casos $D_{\mathbf{x}}^K$ ya clasificados más cercanos a \mathbf{x}

Asignar a \mathbf{x} la clase más frecuente en $D_{\mathbf{x}}^K$

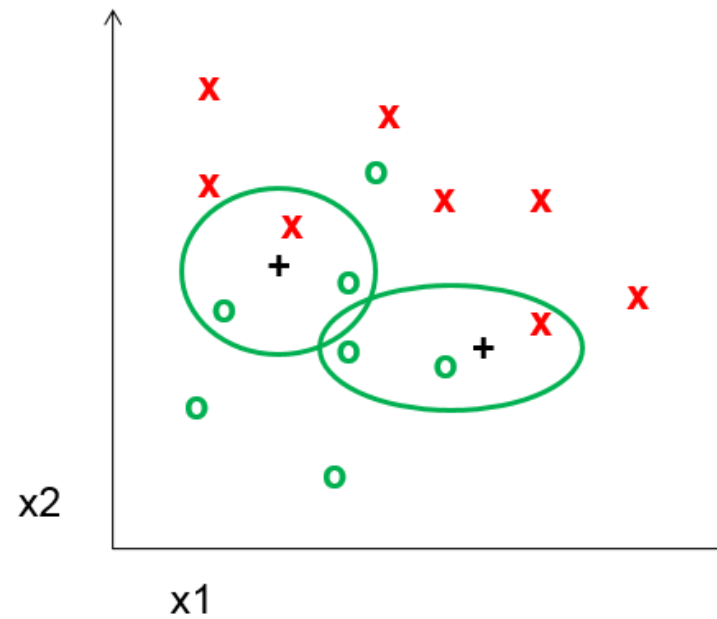
FIN

Clasificador KNN

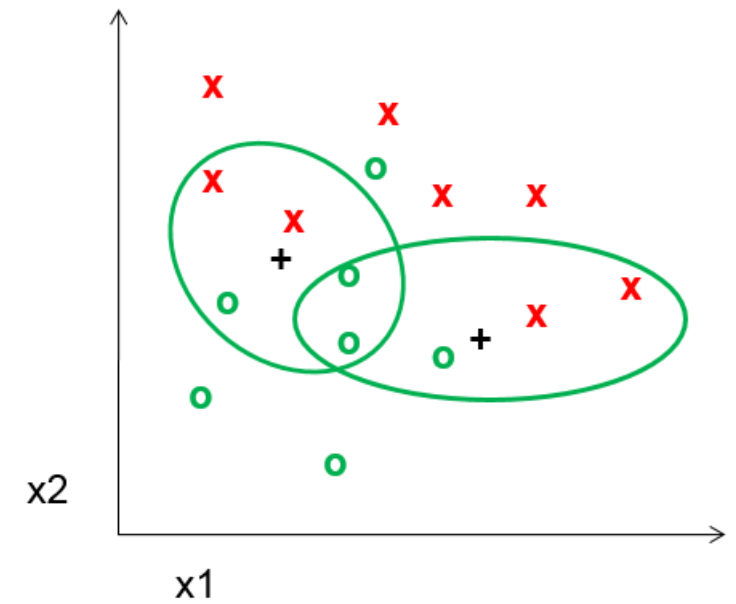
KNN-1 (1 Vecino más cercano)



KNN-3 (3 Vecinos más cercanos)



KNN-5 (5 Vecinos más cercanos)

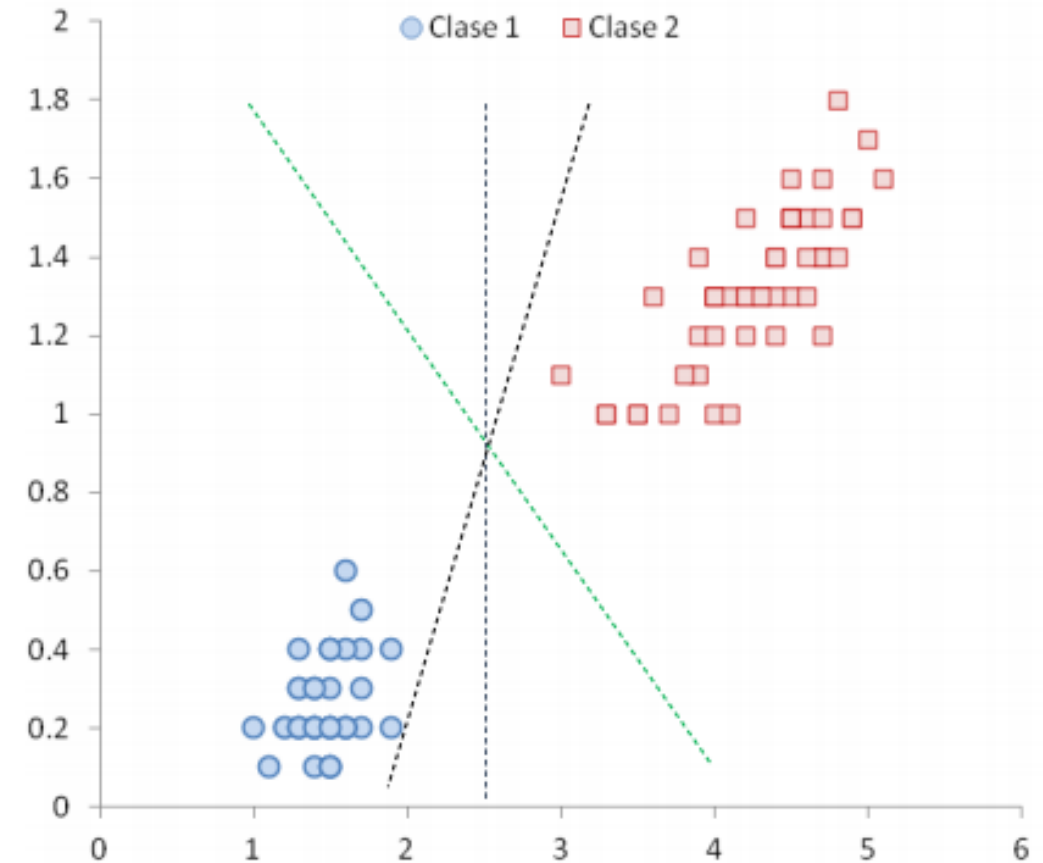


Máquinas de Vectores de Soporte

Las Máquinas de Vectores de Soporte (SVM) buscan el **hiperplano que separe de forma óptima** los puntos de una clase de otra.

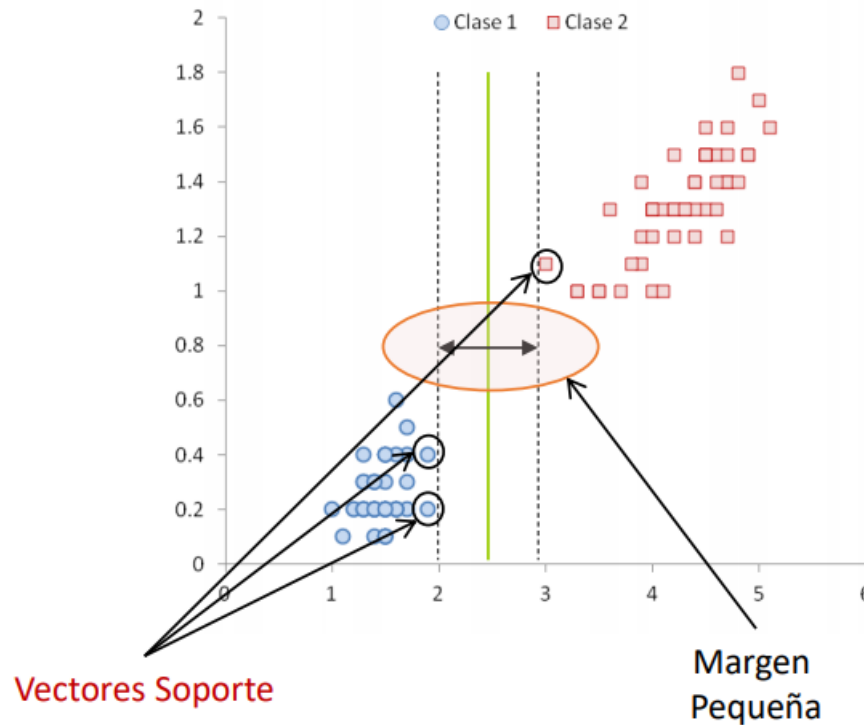
En un problema de dos clases, linealmente separables, pueden existir muchas fronteras de decisión (o hiperplanos) que pueden separar las clases.

Sin embargo, ¿**Son todas esas fronteras igual de buenas?**

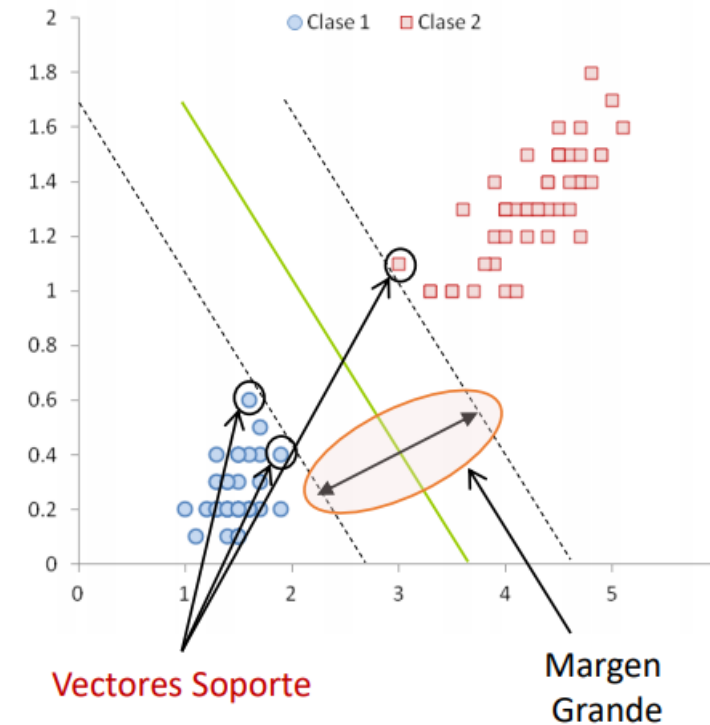


Máquinas de Vectores de Soporte

La SVM busca el hiperplano que maximiza la distancia (o margen) con los puntos que estén más cerca de él, razón por la cual también se les conoce a las SVM como clasificadores de margen máximo.



Como el hiperplano separa las muestras positivas (+1) de las negativas (-1), los puntos que están en el hiperplano deben satisfacer la ecuación: $\mathbf{w}^T \mathbf{x} + b = 0$.



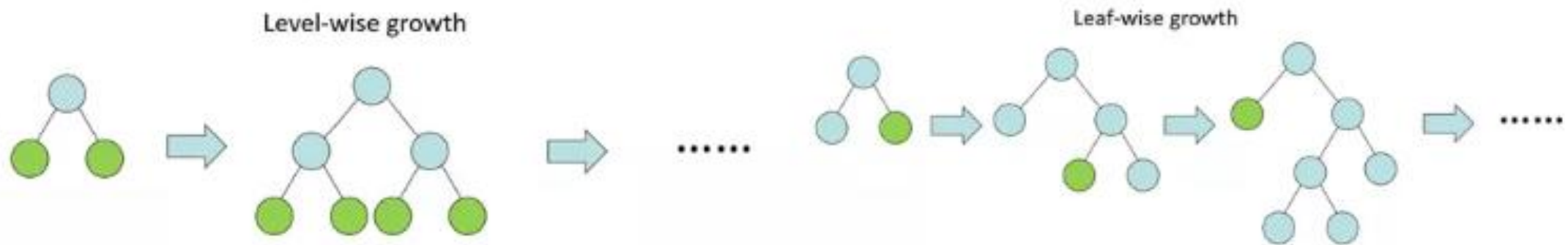
XGBoost: eXtreme Gradient Boosting

Links de interés:

- <https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/>

Xgboost es un algoritmo de ensamble, tipo boosting, de árboles de decisión. **En un ensamble boosting los árboles son construidos de manera secuencial**, por lo que cada árbol siguiente reduce los errores de los árboles previos. Cada árbol aprende de sus predecesores y actualiza los errores.

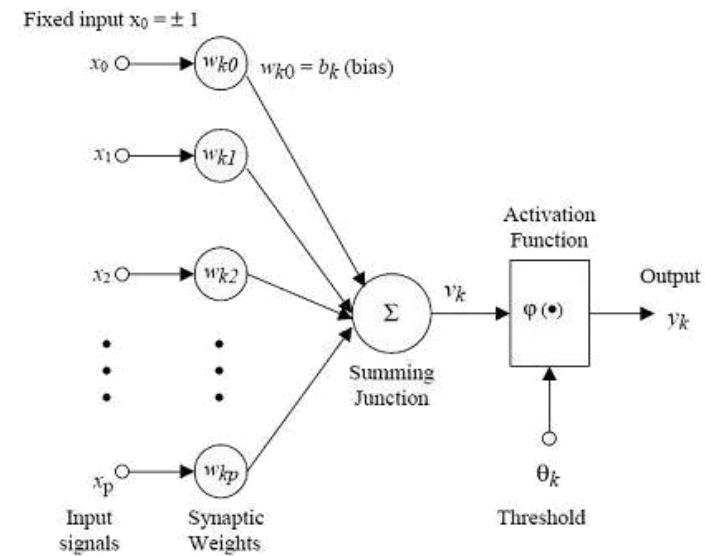
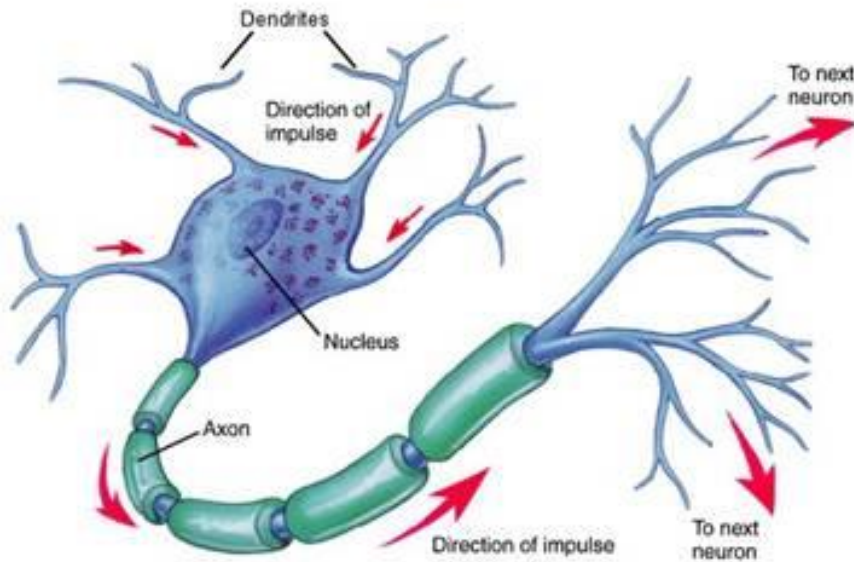
Xgboost es uno de los métodos más populares de modelado de bases de datos tabulares de cualquier tamaño, es muy rápido y escalable.



An illustration demonstrating the difference between level-wise and leaf-wise growth

Redes Neuronales Artificiales

- Por su capacidad de aprendizaje **las neuronas** de los organismos biológicos se han estudiado para su aplicación en sistemas de aprendizaje automático.
- Al igual que las neuronas biológicas están conectadas, las redes de neuronas artificiales están formadas por elementos sencillos de cómputo interconectados según diferentes modelos.



Redes Neuronales Artificiales

“El objetivo del entrenamiento de una RNA es conseguir que una aplicación determinada, para un conjunto de entradas produzca el conjunto de salidas deseadas o mínimamente consistentes”

El conjunto de Aprendizaje debe ser:

Significativo: Número suficiente de ejemplos.

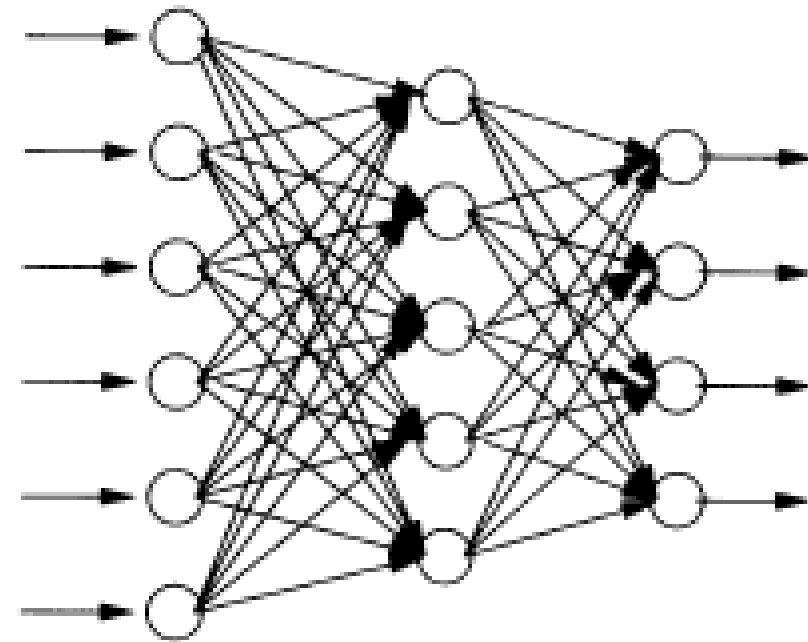
Representativo: Conjunto de aprendizaje diverso.

La convergencia en el entrenamiento se da por:

Número fijo de ciclos.

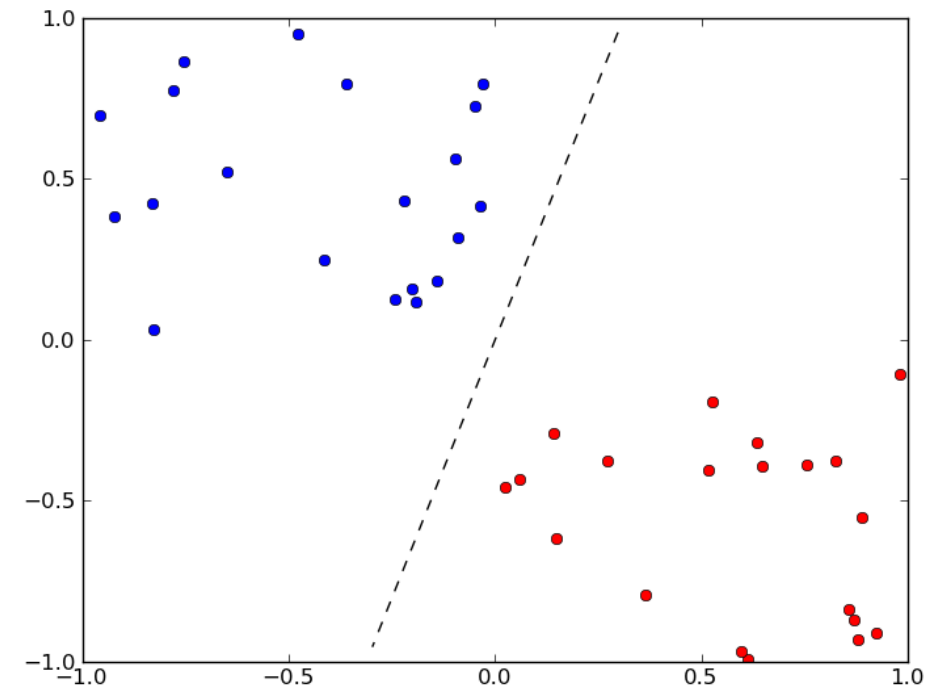
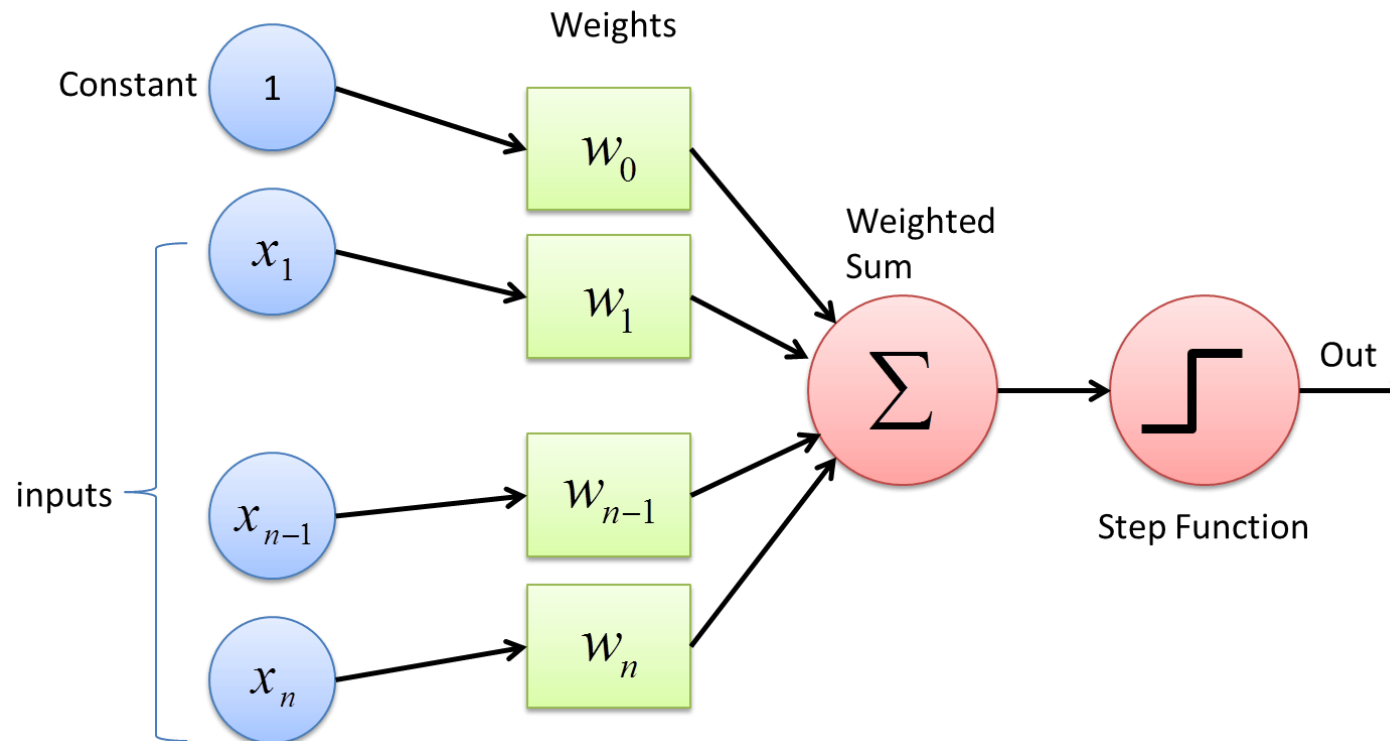
Error descienda de cierta cantidad.

Modificación de los pesos sea irrelevante.



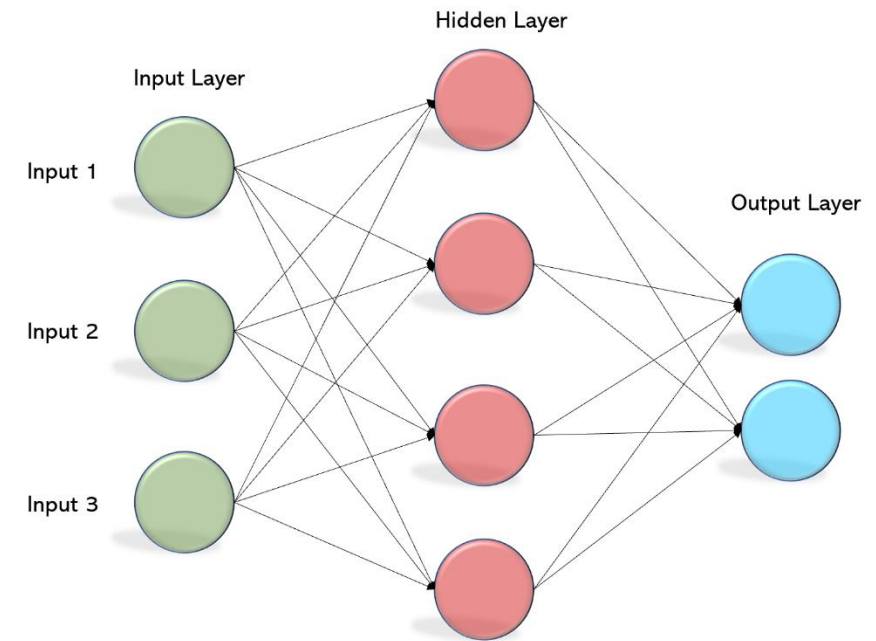
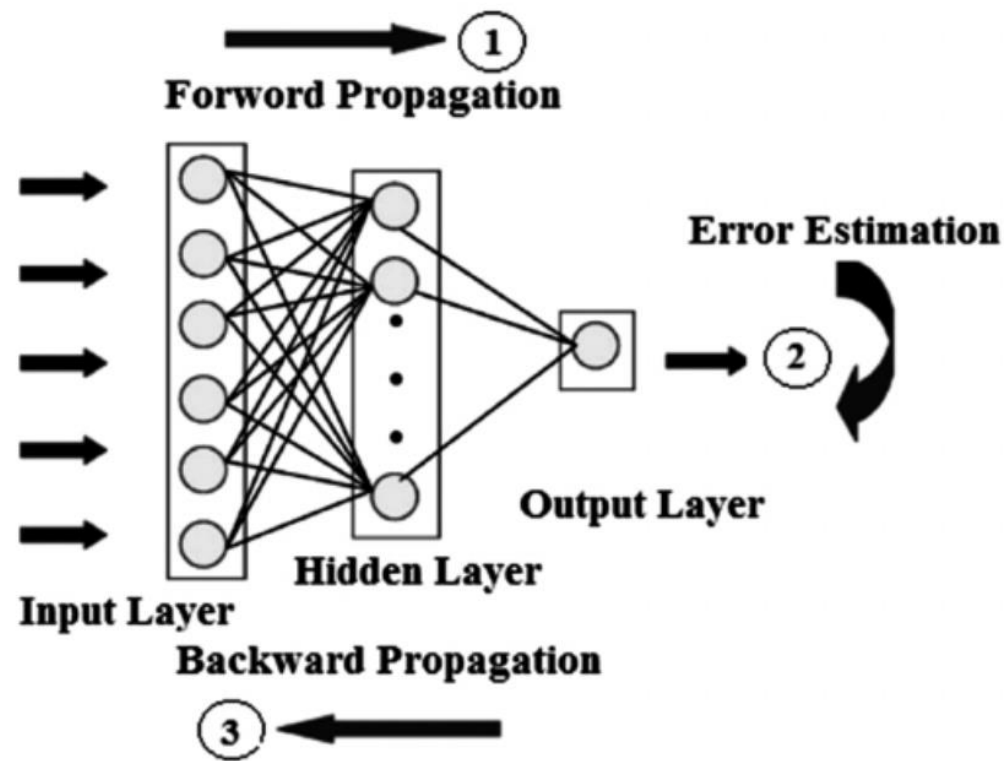
Redes Neuronales Artificiales

Un perceptron (la red neuronal artificial más simple) permite resolver problemas linealmente separables. Este funciona multiplicando cada valor de entrada por un peso (que aprende la red) y sumado para una sola neurona. Por último, el valor obtenido es pasado por una función de activación para poder dividir el valor resultante en dos conjuntos.



Redes Neuronales Artificiales

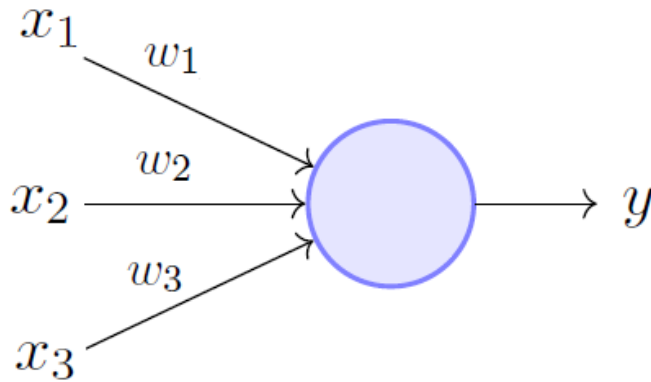
Un multi-perceptron (una red densa o totalmente conectada) permite resolver problemas linealmente NO separables. Esta es la ventaja que tiene contra el perceptrón simple. Como contiene más neuronas y capas ocultas, es capaz de mapear funciones más complejas a través de los pesos aprendidos con las épocas de entrenamiento. Lo que vuelve a una red neuronal compleja poder resolver problemas linealmente NO separables, son las funciones de activación no lineales.



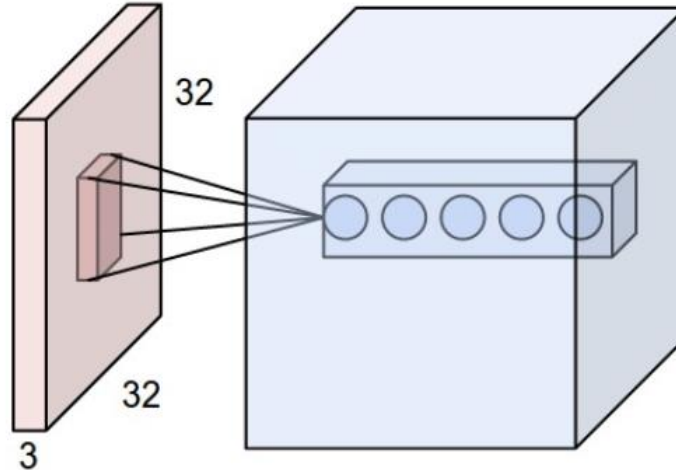
Tipos de neuronas más utilizadas

Existen un alto número de tipos de neuronas diseñados para las redes neuronales. Entre las más comunes tenemos:

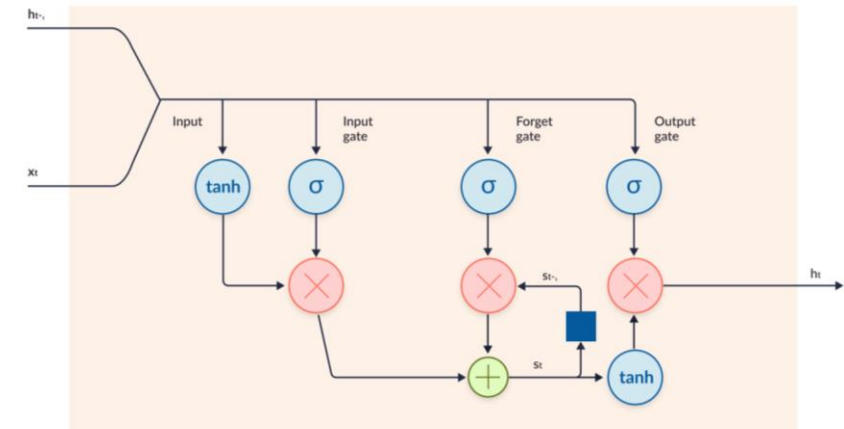
Perceptron



Convolutucional

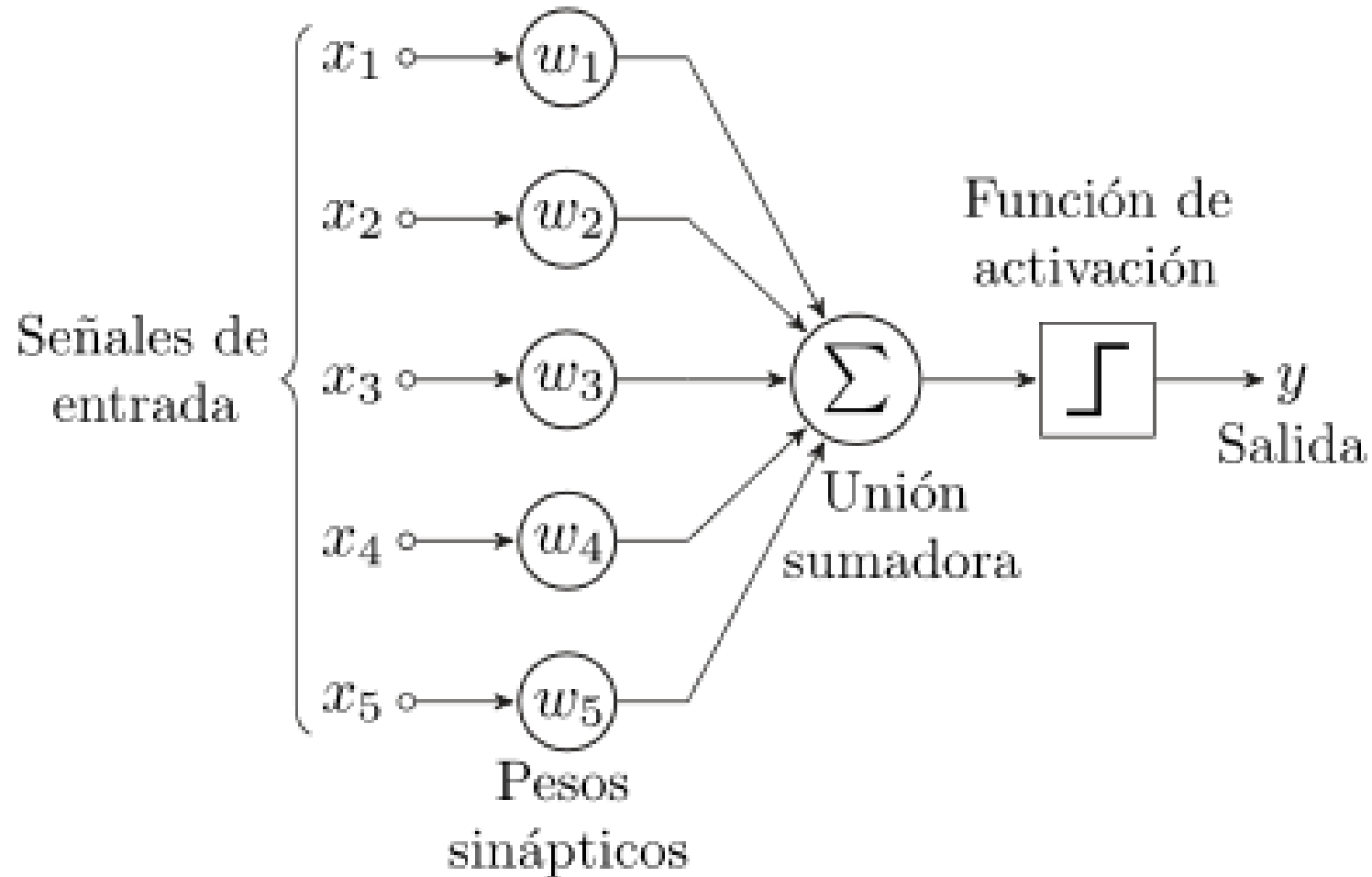


Recurrente



Perceptron

Es la neurona tradicional y se calcula su valor multiplicando todas las entradas por sus pesos de conexión y sumados todos en un solo valor. Sus entradas son de 1 dimensión.



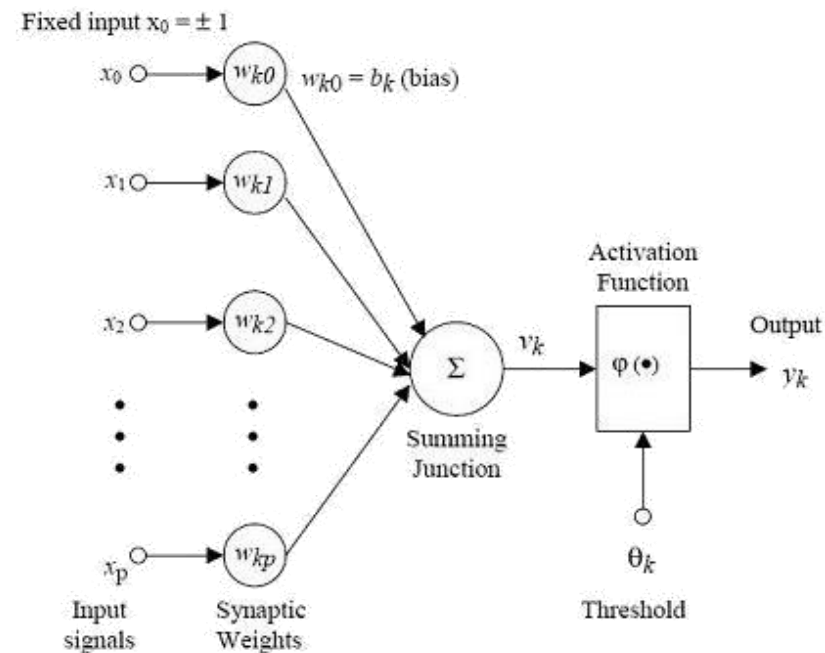
Perceptron

- El **Perceptrón**, en su forma básica, consiste en una neurona que es capaz de aprender una función discriminante lineal v_k , que permite dividir a dos conjuntos de entrenamiento linealmente separables. Su respuesta consiste en una suma ponderada de sus entradas que representa la ecuación de un hiperplano en el espacio p-dimensional :

$$v_k = \sum_{j=1}^p w_{kj} x_j$$

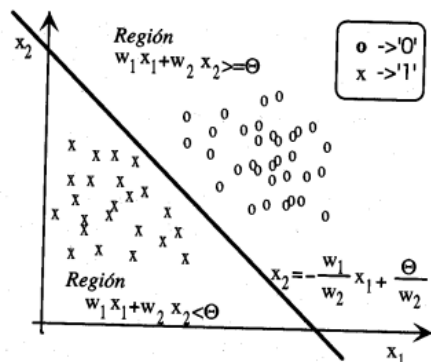
- A la salida se aplica una función de activación $\phi(v)$ (escalón, sigmoide, etc) que indica si se activa o no la neurona.

$$\phi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \quad \phi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1 - \exp(-v)}{1 + \exp(-v)}$$

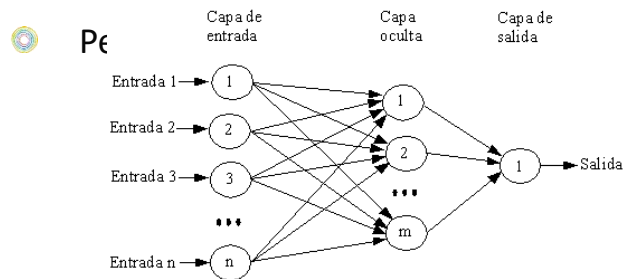


Perceptron

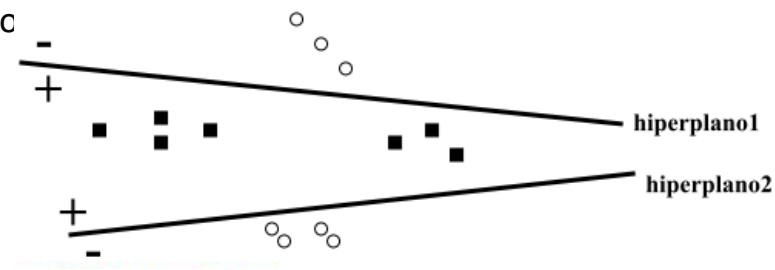
Perceptrón: Separación de d



rceptrón:



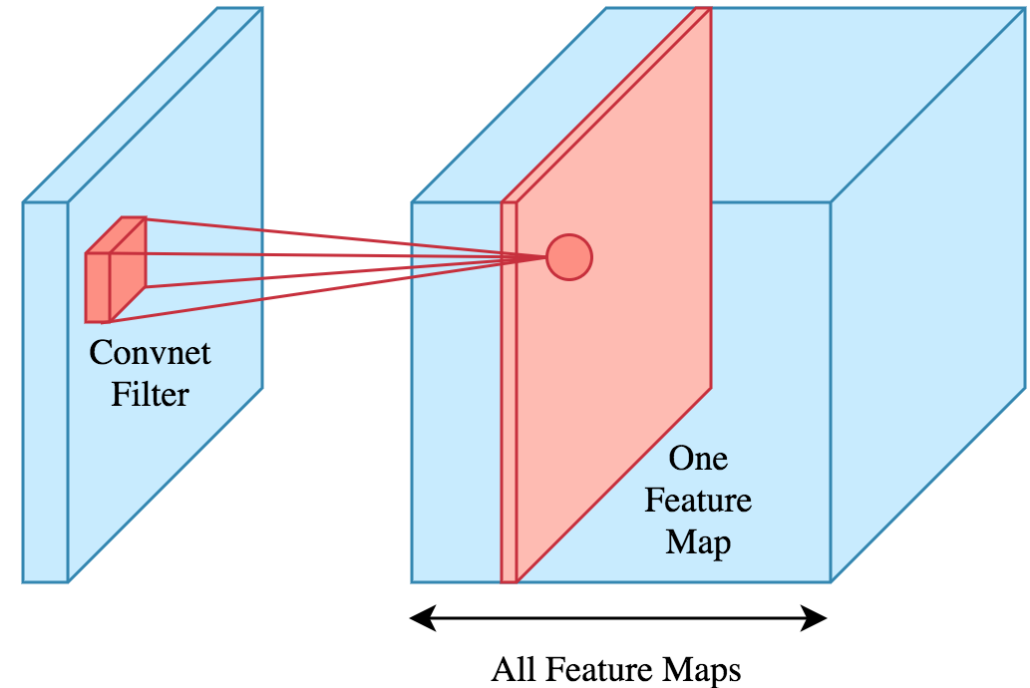
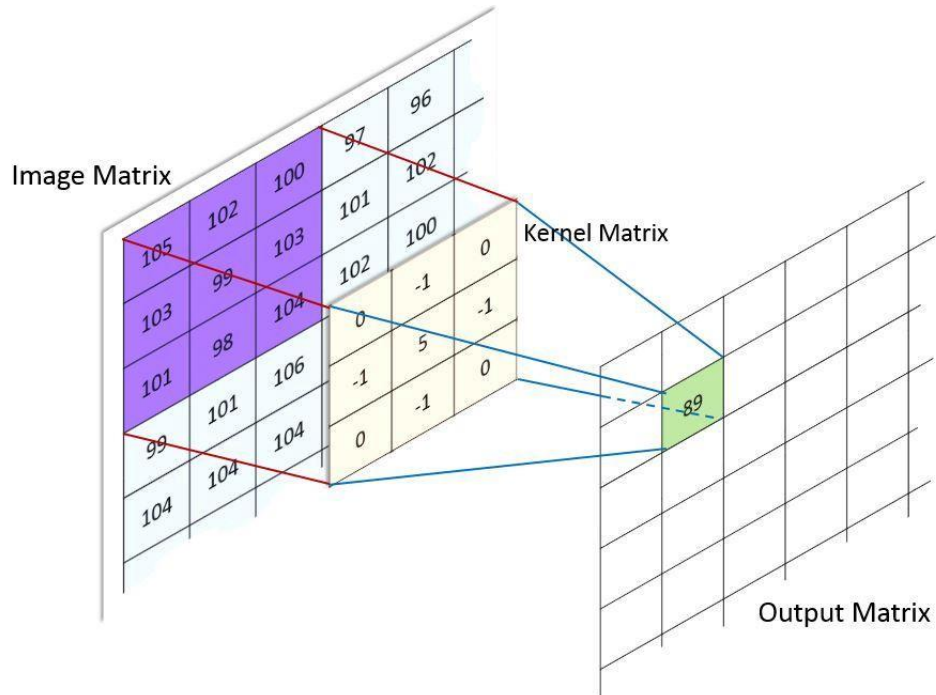
y ejemplo



Una vez una vez entrenada la red con un conjunto de patrones de entrenamiento, ésta es capaz de resolver el problema para patrones desconocidos.

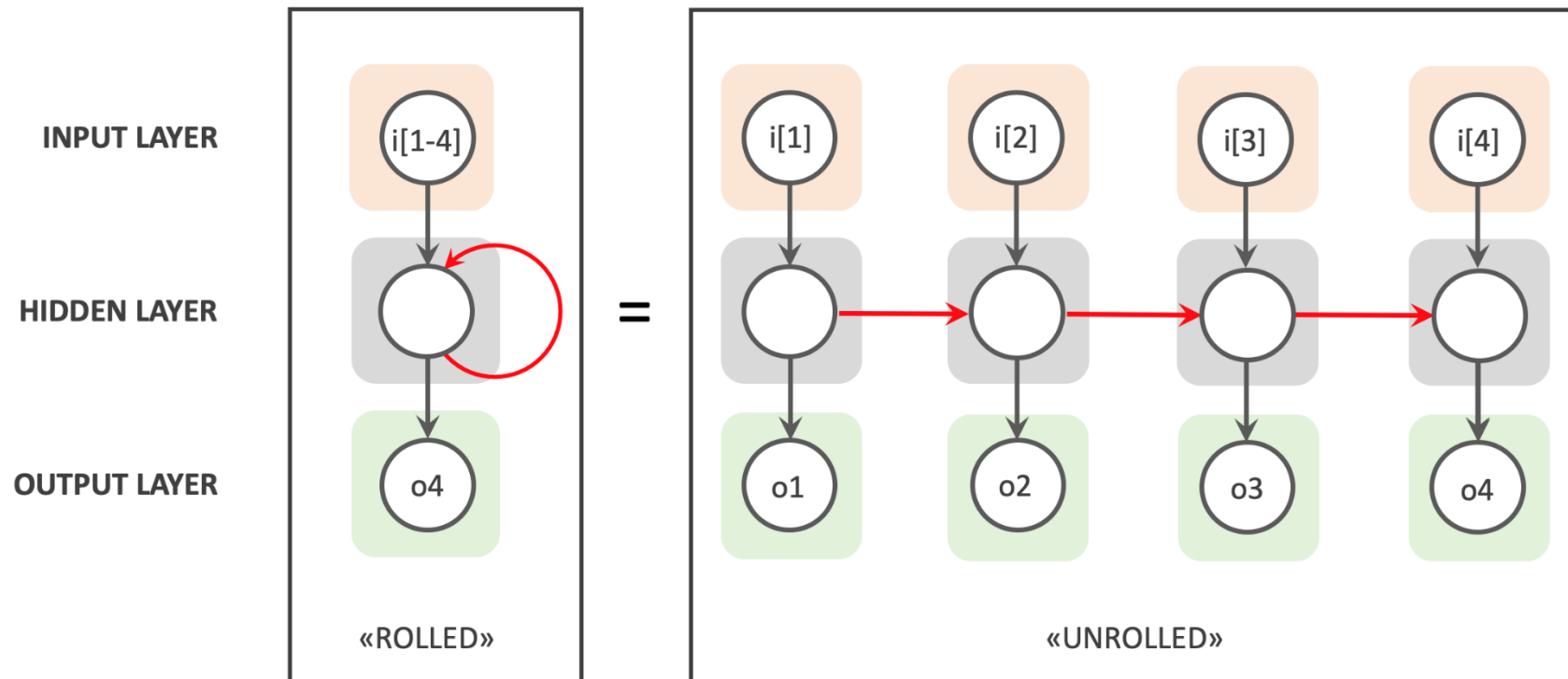
Convolutacional

Es la neurona especial que recibe como entrada un valor de dos o tres dimensiones (una imagen). Esta es capaz de explotar la relación espacial que los datos de entrada contienen. Dependiendo de sus parámetros, el resultado de la capa es otra imagen (datos de dos o tres dimensiones) en otra distribución de datos. De una manera más directa, esta neurona es capaz de aprender “filtros” para las entradas.














Recurrentes

Es la neurona especial que recibe como entrada un valor de dos dimensiones (una señal en el tiempo). Esta es capaz de explotar la relación temporal que los datos de entrada contienen. La salida de esta neurona, dependiendo de los parámetros, puede ser una única salida al final o puede ser una salida en cada tiempo (dato) de entrada.



A mostly complete chart of Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

-  Backfed Input Cell
-  Input Cell
-  Noisy Input Cell
-  Hidden Cell
-  Probabilistic Hidden Cell
-  Spiking Hidden Cell
-  Output Cell
-  Match Input Output Cell
-  Recurrent Cell
-  Memory Cell
-  Different Memory Cell
-  Kernel
-  Convolution or Pool

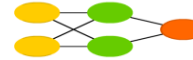
Perceptron (P)



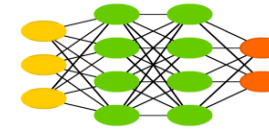
Feed Forward (FF)



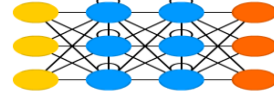
Radial Basis Network (RBF)



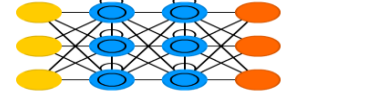
Deep Feed Forward (DFF)



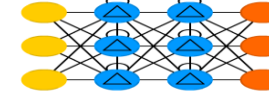
Recurrent Neural Network (RNN)



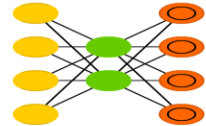
Long / Short Term Memory (LSTM)



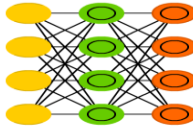
Gated Recurrent Unit (GRU)



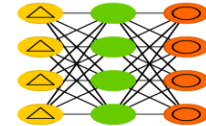
Auto Encoder (AE)



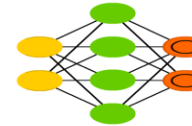
Variational AE (VAE)



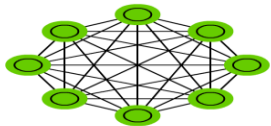
Denoising AE (DAE)



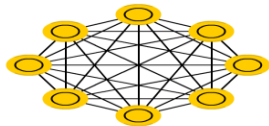
Sparse AE (SAE)



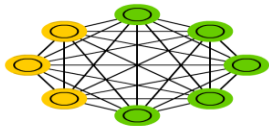
Markov Chain (MC)



Hopfield Network (HN)



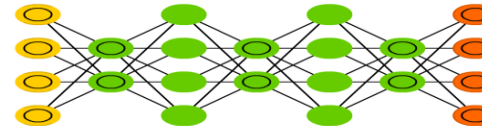
Boltzmann Machine (BM)



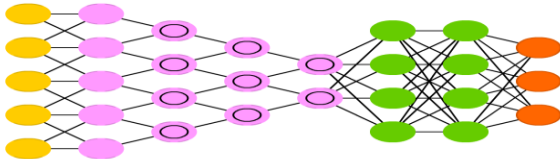
Restricted BM (RBM)



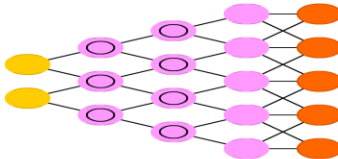
Deep Belief Network (DBN)



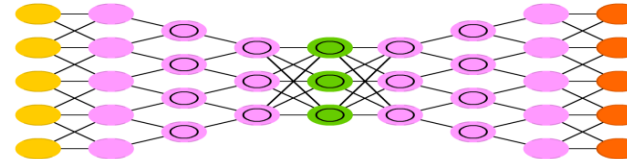
Deep Convolutional Network (DCN)



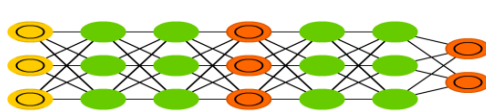
Deconvolutional Network (DN)



Deep Convolutional Inverse Graphics Network (DCIGN)



Generative Adversarial Network (GAN)



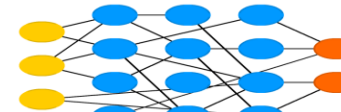
Liquid State Machine (LSM)



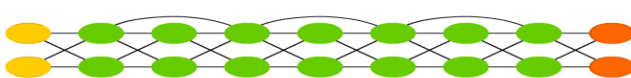
Extreme Learning Machine (ELM)



Echo State Network (ESN)



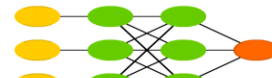
Deep Residual Network (DRN)



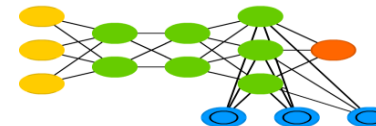
Kohonen Network (KN)



Support Vector Machine (SVM)



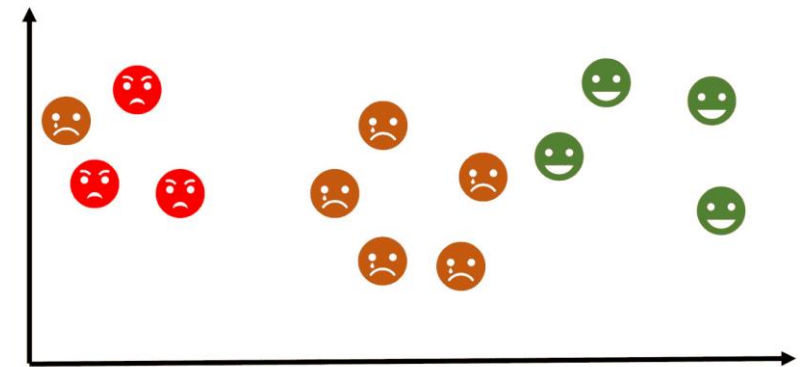
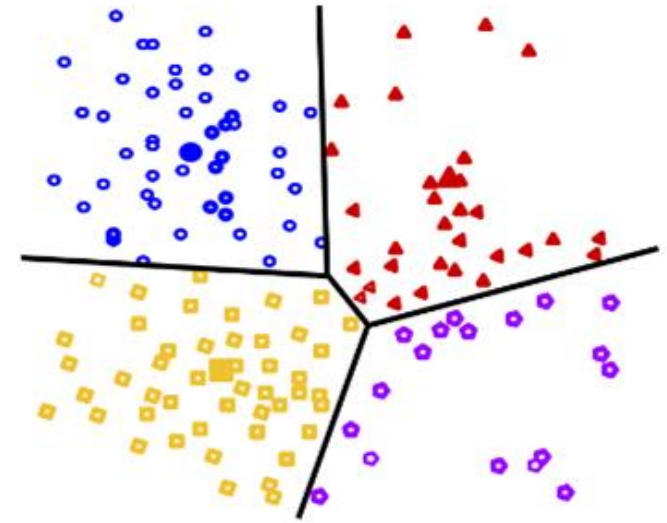
Neural Turing Machine (NTM)



Clasificación No Supervisada

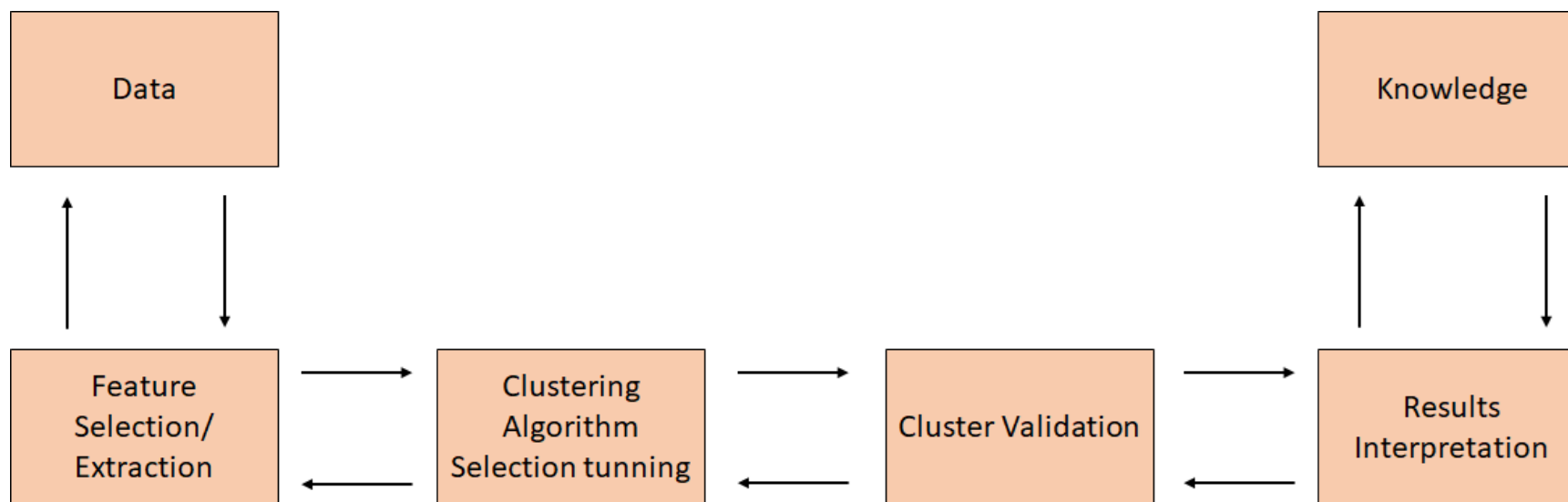
Se cuenta con un conjunto de datos de entrenamiento, pero no hay una variable específica de salida (se desconocen las clases). En este sentido, el objetivo de los problemas del aprendizaje no supervisado es, agrupar los datos de entrada con base en algún criterio de similitud o disimilitud o determinar la distribución estadística de los datos, conocida como estimación de la densidad.

Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(m)}\}$



Proceso de Clasificación No Supervisada

El proceso general que seguiremos al desarrollar un modelo de aprendizaje no supervisado se puede resumir en el siguiente cuadro:

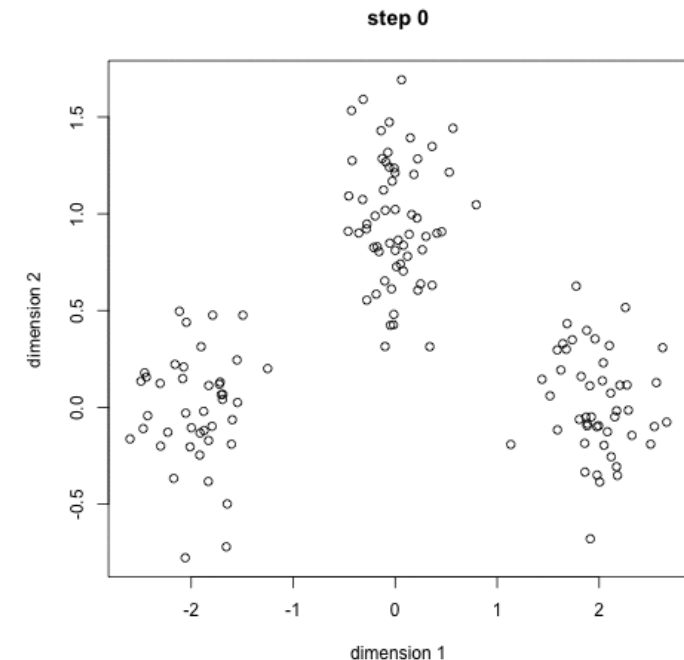


<https://towardsdatascience.com/unsupervised-machine-learning-clustering-analysis-d40f2b34ae7e>

Agrupamiento (Clustering)

El agrupamiento es una técnica que implica la agrupación de puntos de datos. Dado un conjunto de puntos de datos, podemos usar un algoritmo de agrupamiento para clasificar cada punto en un grupo específico.

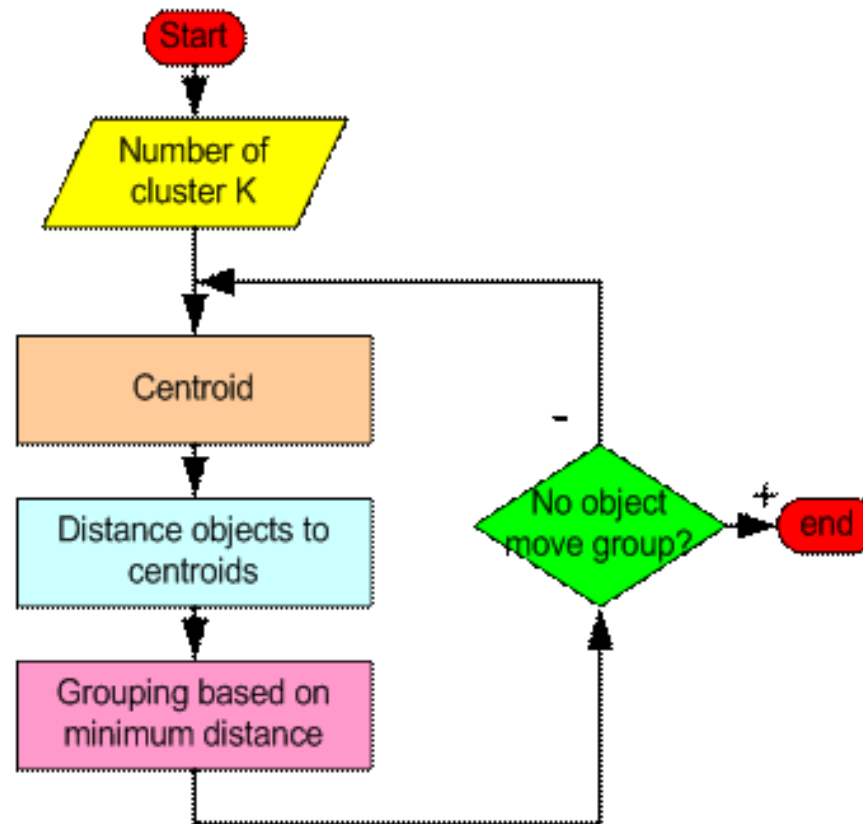
En teoría, los puntos de datos que están en el mismo grupo deberían tener propiedades y / o características similares, mientras que aquellos en diferentes grupos deberían tener propiedades y / o características muy diferentes. La similitud entre puntos generalmente se cuantifica mediante una métrica de distancia basada en algún tipo de conjunto de variables de características.



<https://towardsdatascience.com/an-easy-introduction-to-unsupervised-learning-with-4-basic-techniques-da7fbf0c3adf>

Agrupamiento (Clustering)

- Los algoritmos de **clustering** o agrupamiento intentan **dividir** el conjunto de datos de entrenamiento en **k grupos**, de acuerdo con un criterio de cercanía que se define en términos de una función de distancia, como la Euclidiana, la Manhattan o la de Mahalanobis.



Agrupamiento (Clustering)

- Cada una de las k clases se representa con un prototipo Z_k o centroide que es un vector d -dimensional:

$$Z_k = \frac{1}{n_k} \sum_{j=1}^{n_k} x_{kj}$$

Siendo: x_{kj} el j -ésimo vector de características (patrón) de la clase k .

- La distancia euclídeana d_E de un nuevo patrón X a la clase C_k es:

$$d_E(X, Z_k) = \|X - Z_k\| = \sqrt{\sum_{i=1}^d (X_i - Z_{ki})^2}$$

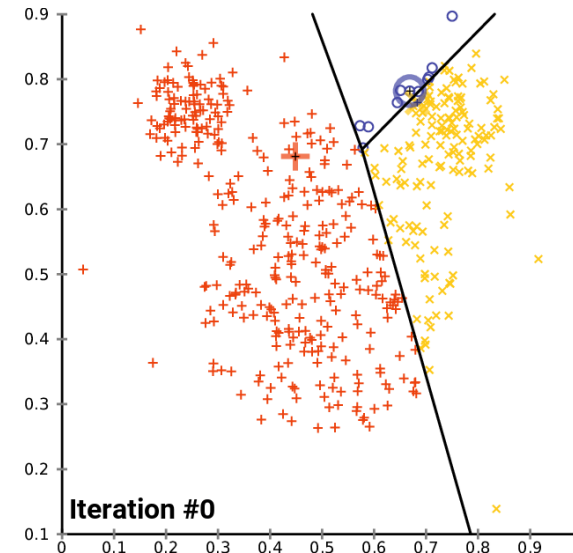
La fórmula anterior es equivalente a evaluar la expresión de la función discriminante de cada clase $fd_k(X)$, siendo: $k \in 1..N$, para el patrón X y asignarlo a la clase C_k para la que $fd_k(X)$ sea máximo.

K-means

El algoritmo de agrupamiento más común y simple que existe es el agrupamiento de K-Means. Este algoritmo implica que le diga cuántos clústeres (o K) posibles hay en el conjunto de datos. Luego, el algoritmo mueve iterativamente los k-centros y selecciona los puntos de datos más cercanos al centroide en el grupo.

Input:

- K (number of clusters)
- Training set $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$



<https://towardsdatascience.com/clustering-based-unsupervised-learning-8d705298ae51>

Algoritmo K-means

1. Initialize **cluster centroids** $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$ randomly.

2. Repeat until convergence: {

For every i , set

$$c^{(i)} := \arg \min_j \|x^{(i)} - \mu_j\|^2.$$

For each j , set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

}

Donde:

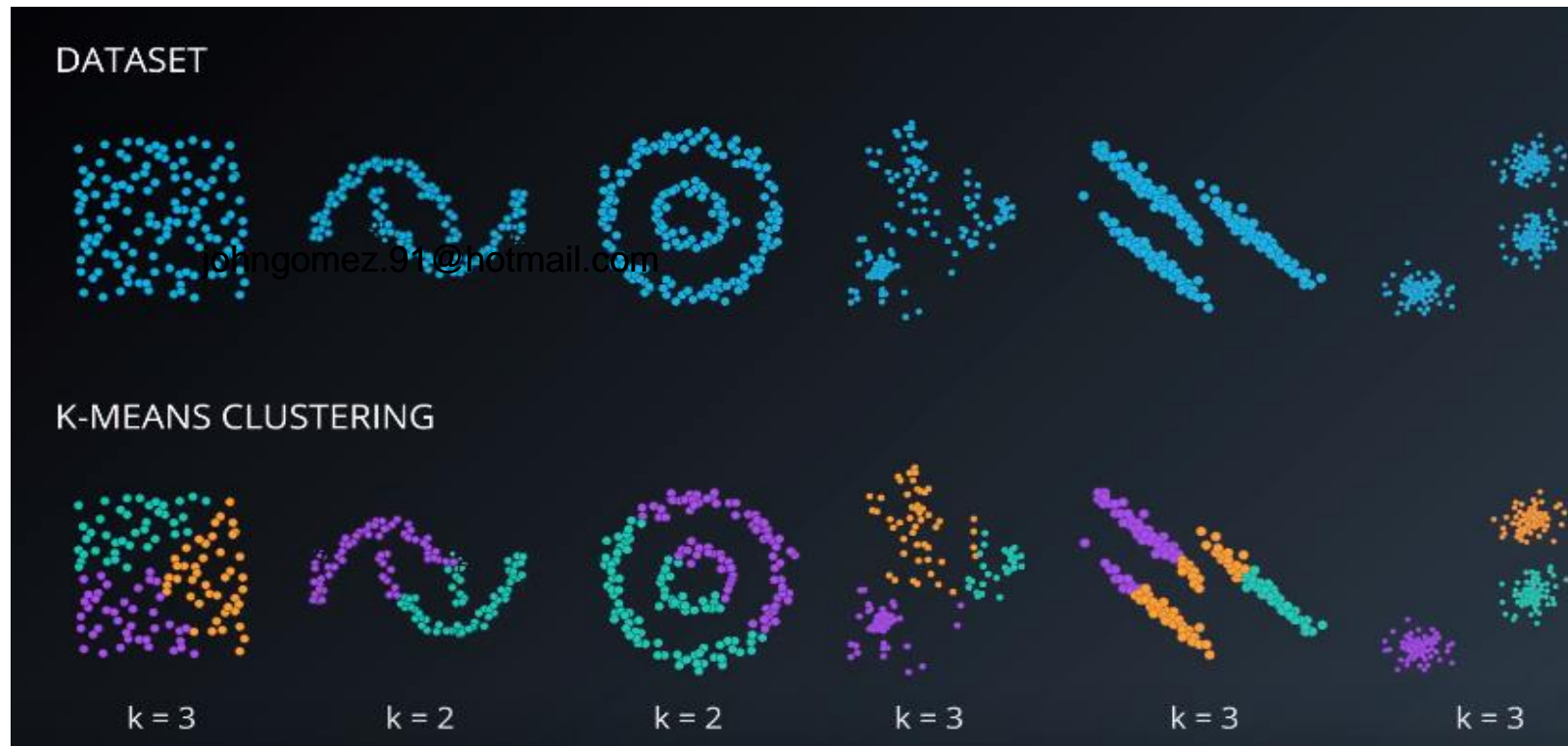
U_k = Cantidad de centroides para inicializar

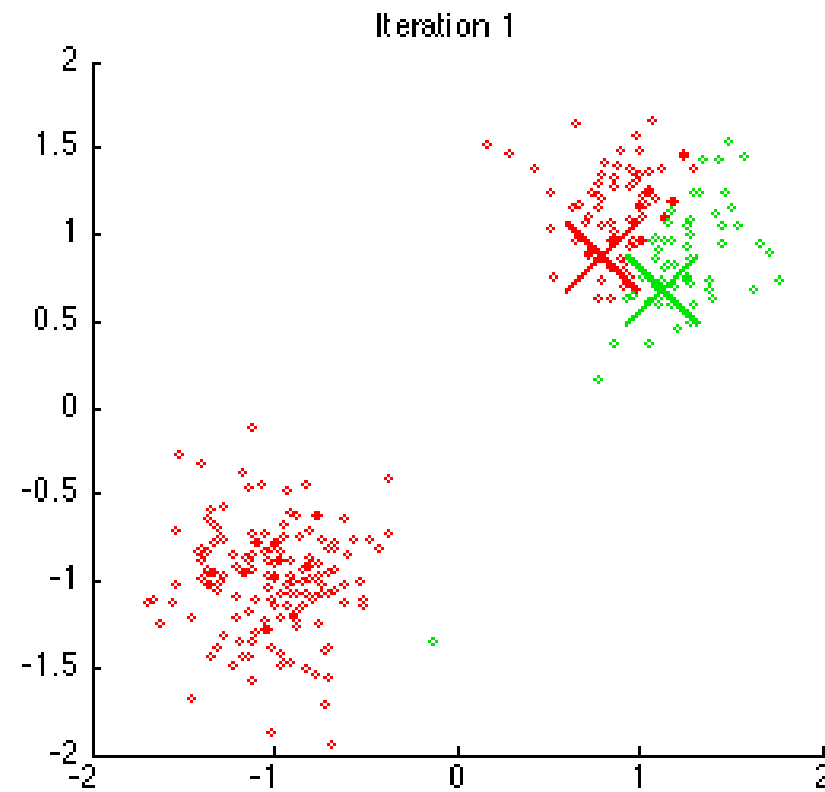
$C(i)$ distancia para cada punto y los centroides. y se queda con el menor (mas cerca)

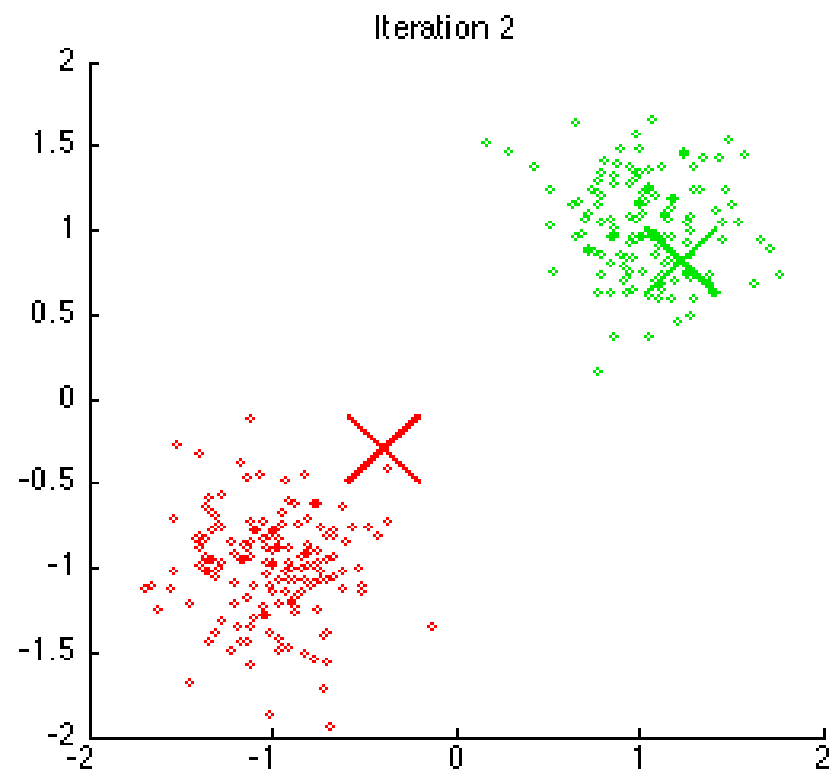
$\mu(j)$ es el centroide para el grupo j

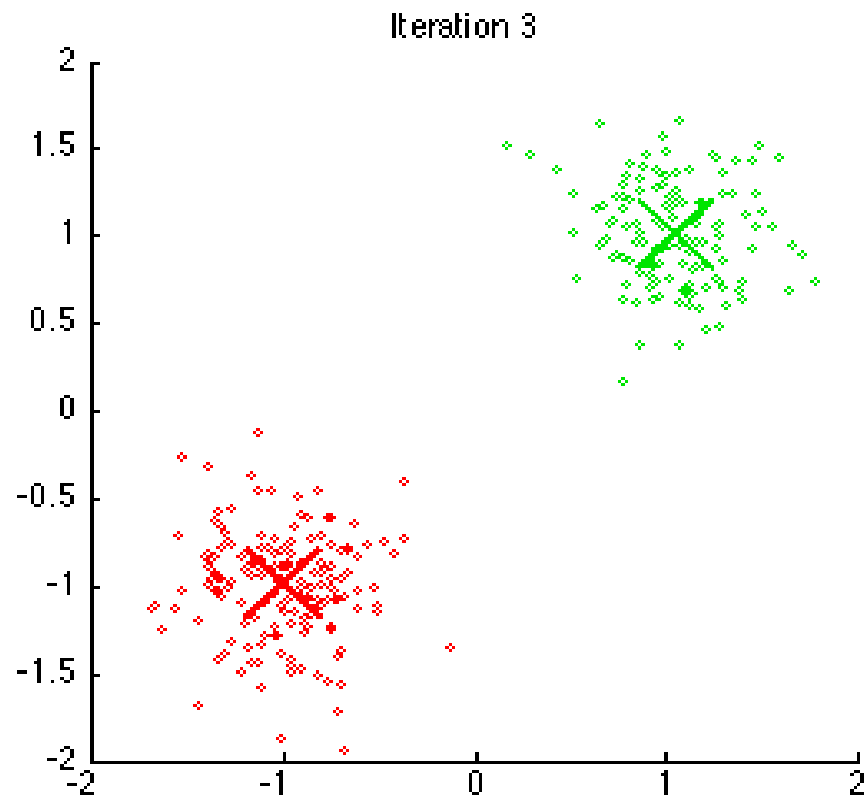
Ejemplos K-means

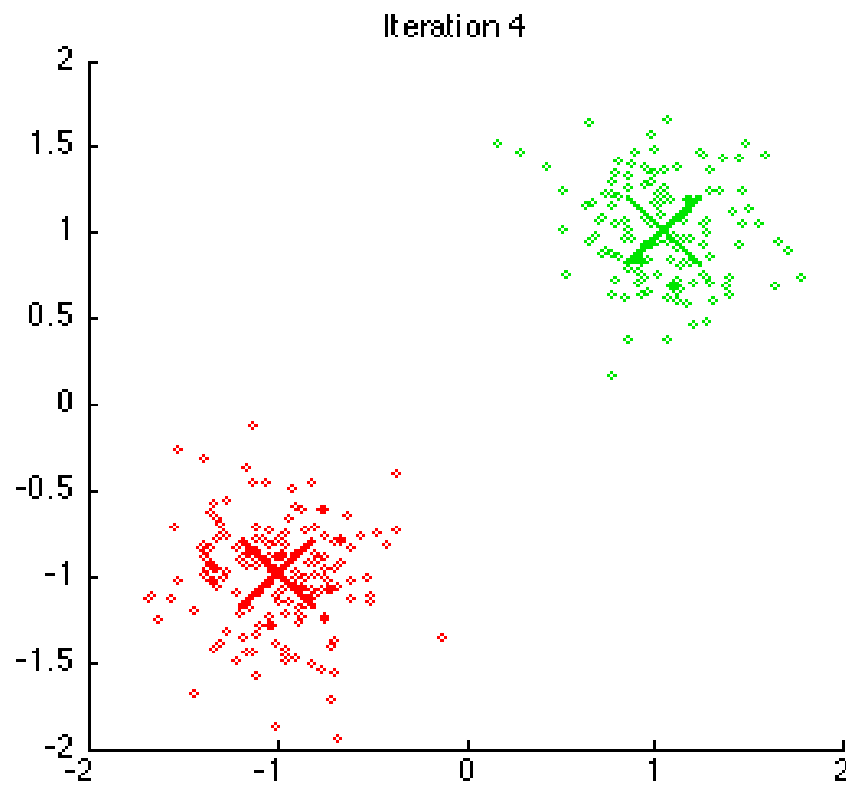
La siguiente imagen muestra lo que obtendríamos si utilizamos el agrupamiento de K-means en cada conjunto de datos, incluso si supiéramos de antemano el número exacto de grupos:











Preguntas



MOTIVACIÓN

🦋 OBSERVE EL VIDEO Y RESPONDA A LAS SIGUIENTES PREGUNTAS:

🦋 ¿CUÁNTOS DATOS SE REQUIEREN PARA ENTRENAR UN SISTEMA DE VISIÓN ARTIFICIAL?

🦋 ¿ES POSIBLE DECIR QUE LOS COMPUTADORES YA SOBREPASARON LA CAPACIDAD HUMANA?

🦋 ¿QUÉ PROBLEMAS EVIDENCIAN LOS SISTEMAS DE VISIÓN ARTIFICIAL, Y EN GENERAL DE LOS SISTEMAS DE RECONOCIMIENTO DE PATRONES?



<https://www.ted.com/talks/fei-fei-li-how-we-re-teaching-computers-to-understand-pictures?language=es>



UNIVERSIDAD
NACIONAL
DE COLOMBIA