



UNIVERSIDAD
NACIONAL
DE COLOMBIA

ANALÍTICA PREDICTIVA

CARLOS A. MADRIGAL

PROFESOR OCASIONAL

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y DE LA DECISIÓN

MAESTRÍA EN INGENIERÍA - INGENIERÍA DE SISTEMAS

MAESTRÍA EN INGENIERÍA - ANALÍTICA

ESPECIALIZACIÓN EN SISTEMAS

CONTENIDO

Machine Learning

- Tipos de Aprendizaje: supervisado, no supervisado
- Extracción de características
- Aplicaciones
- Evaluación

Introducción a las Redes Neuronales Artificiales

- Conceptos, aplicaciones, historia
- Entrenamiento
- Perceptrón simple
- Red Adaline

MACHINE LEARNING

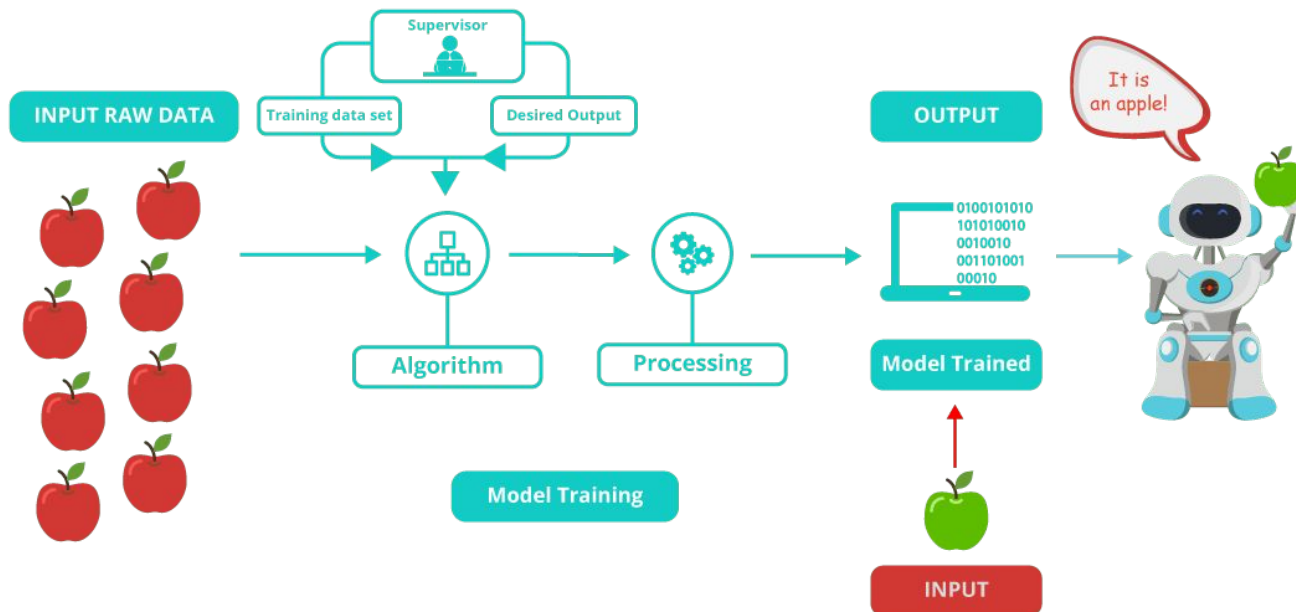
“Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed”. Arthur Samuel (1959).

“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 Mitchell (1997), Carnegie Mellon University.

TIPOS DE APRENDIZAJE

SUPERVISADO

El algoritmo es entrenado con un conjunto previo de “ejemplos de entrenamiento” o **ground truth**, de los cuales se conoce el valor de salida para cada entrada.

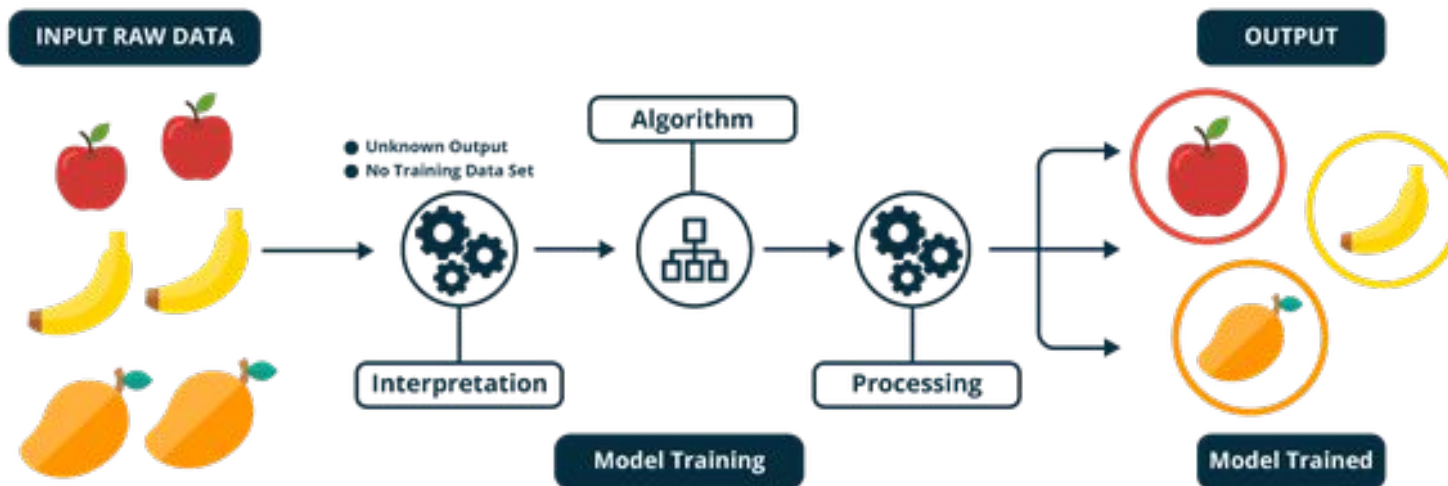


- Regresión Logística
- Naive Bayes
- Máquinas de Soporte Vectorial
- Redes Neuronales
- Random Forest.

TIPOS DE APRENDIZAJE

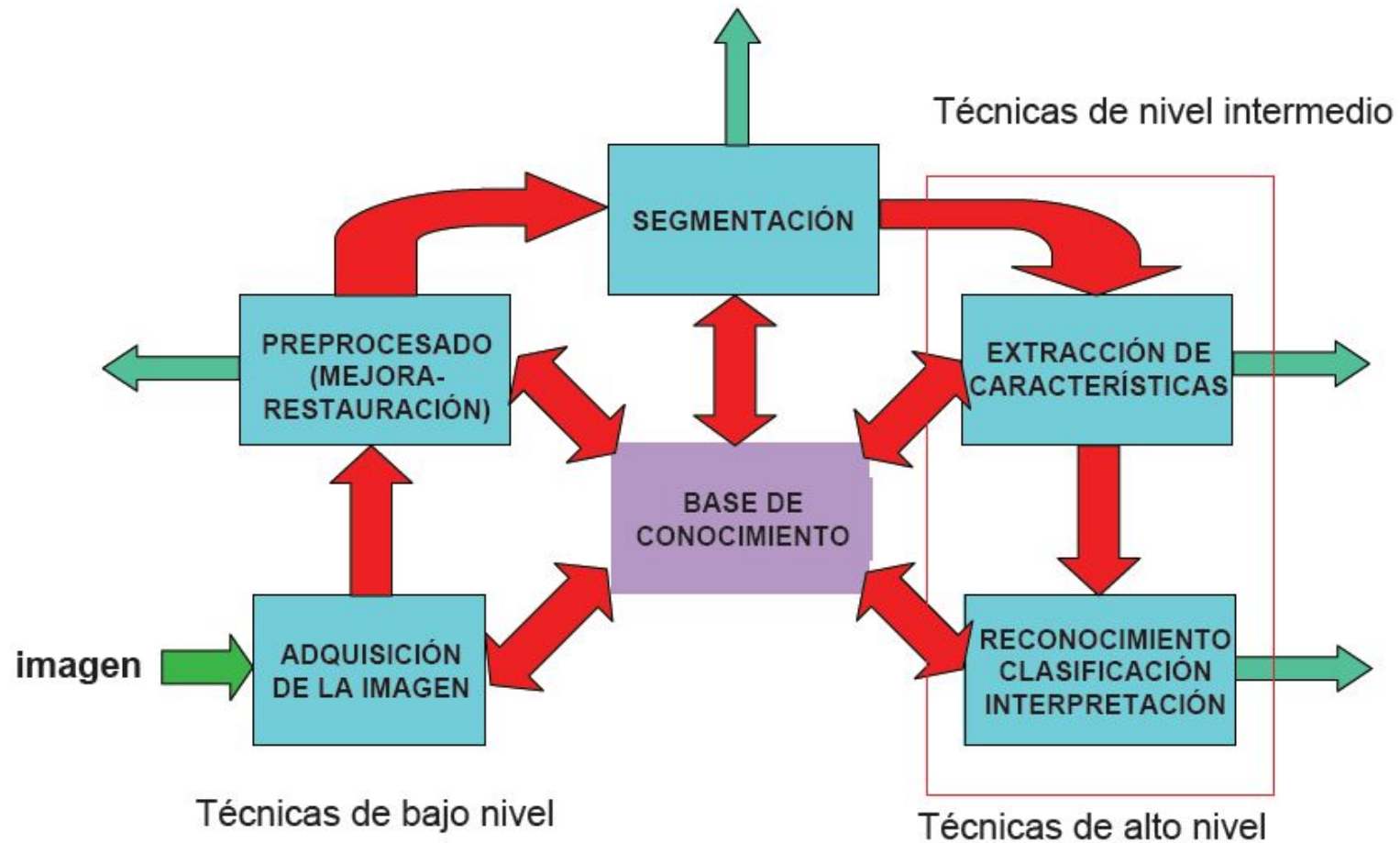
No SUPERVISADO

El algoritmo debe encontrar patrones y similitudes entre el conjunto de datos de los cuales se desconoce su salida. Comúnmente son llamados algoritmos de *Clustering*



- Kmeans
- Análisis de Componentes Principales
- AutoEncoders

RECONOCIMIENTO DE PATRONES CON APLICACIÓN A LAS IMÁGENES



APLICACIÓN DE CLASIFICACIÓN



¿Es posible de manera automática clasificar entre los granos maduros, verdes y con broca?

APLICACIÓN DE CLASIFICACIÓN

Característica: % Rojo

67.9%



21.2%



58.5%



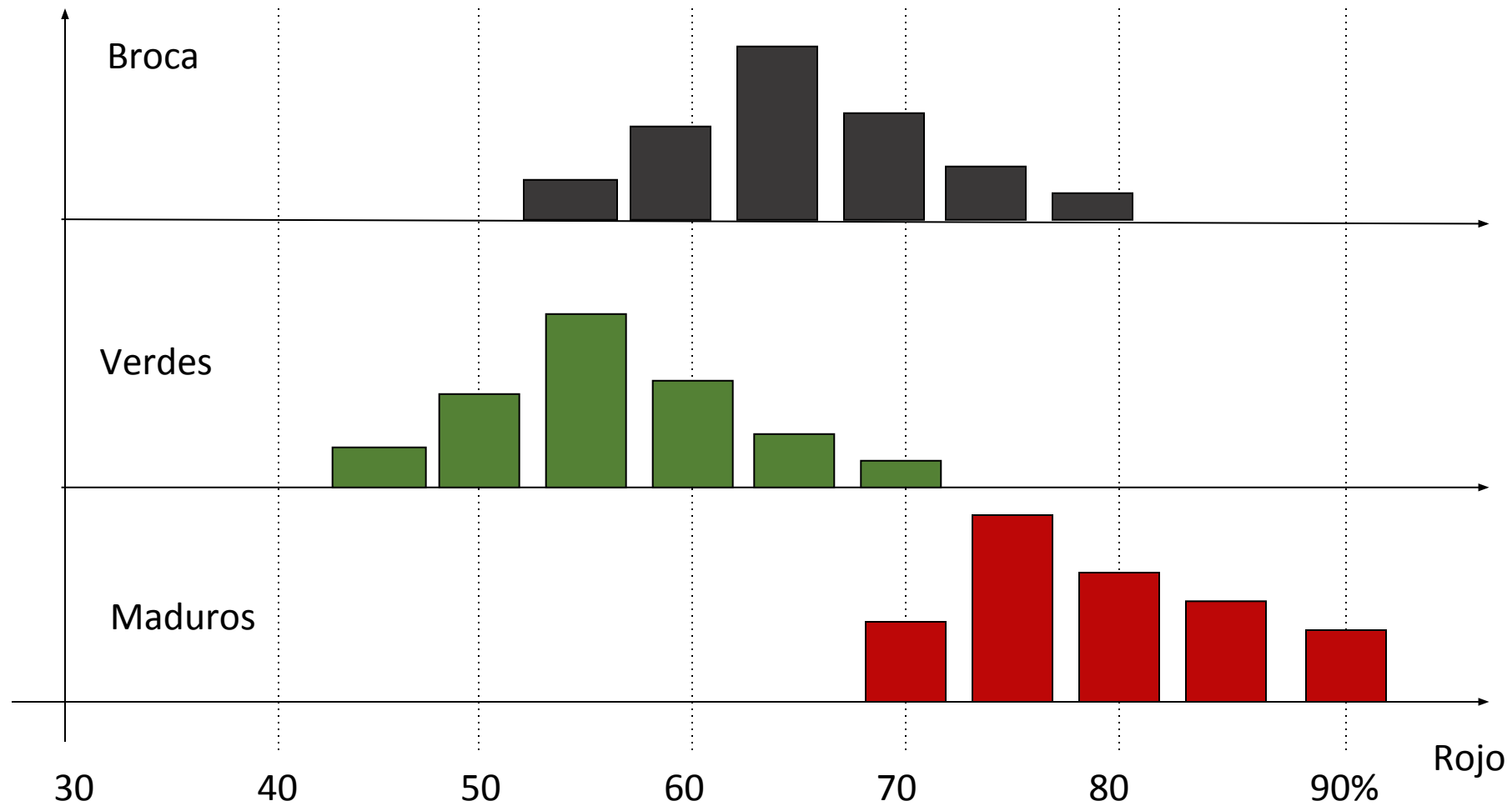
APLICACIÓN DE CLASIFICACIÓN

Característica: % Rojo

Maduro-01	75.2	Verde-01	32.6	Broca-01	52.6
Maduro-02	68.8	Verde-02	30.1	Broca-02	62.5
Maduro-03	63.5	Verde-03	61.1	Broca-03	48.2
Maduro-04	80.6	Verde-04	50.9	Broca-04	25.4
Maduro-05	55,8	Verde-05	29.7	Broca-05	23.8
Maduro-06	74.5	Verde-06	35,7	Broca-06	75.5
Maduro-07	81,2	Verde-07	48,9	Broca-07	41.6
:		:		:	
Maduro-50	61.7	Verde-50	32,4	Broca-50	24.9

Histogramas:

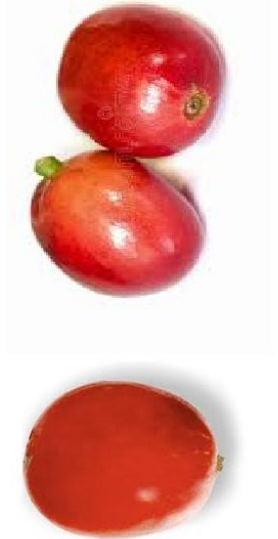
Una característica



APLICACIÓN DE CLASIFICACIÓN

Diámetro Ombligo (Píxeles)

18 pix.



15 pix.



26 pix.



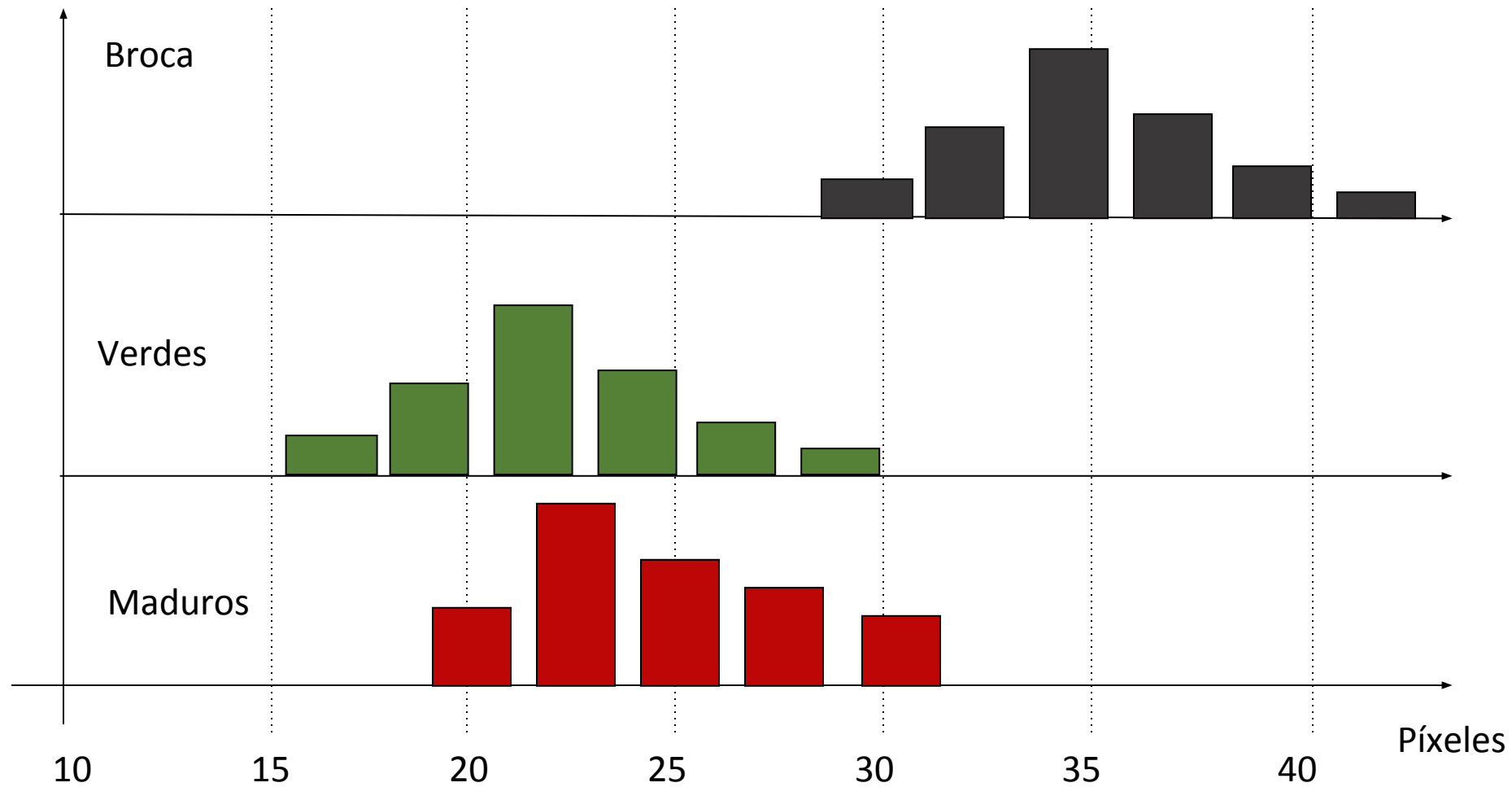
APLICACIÓN DE CLASIFICACIÓN

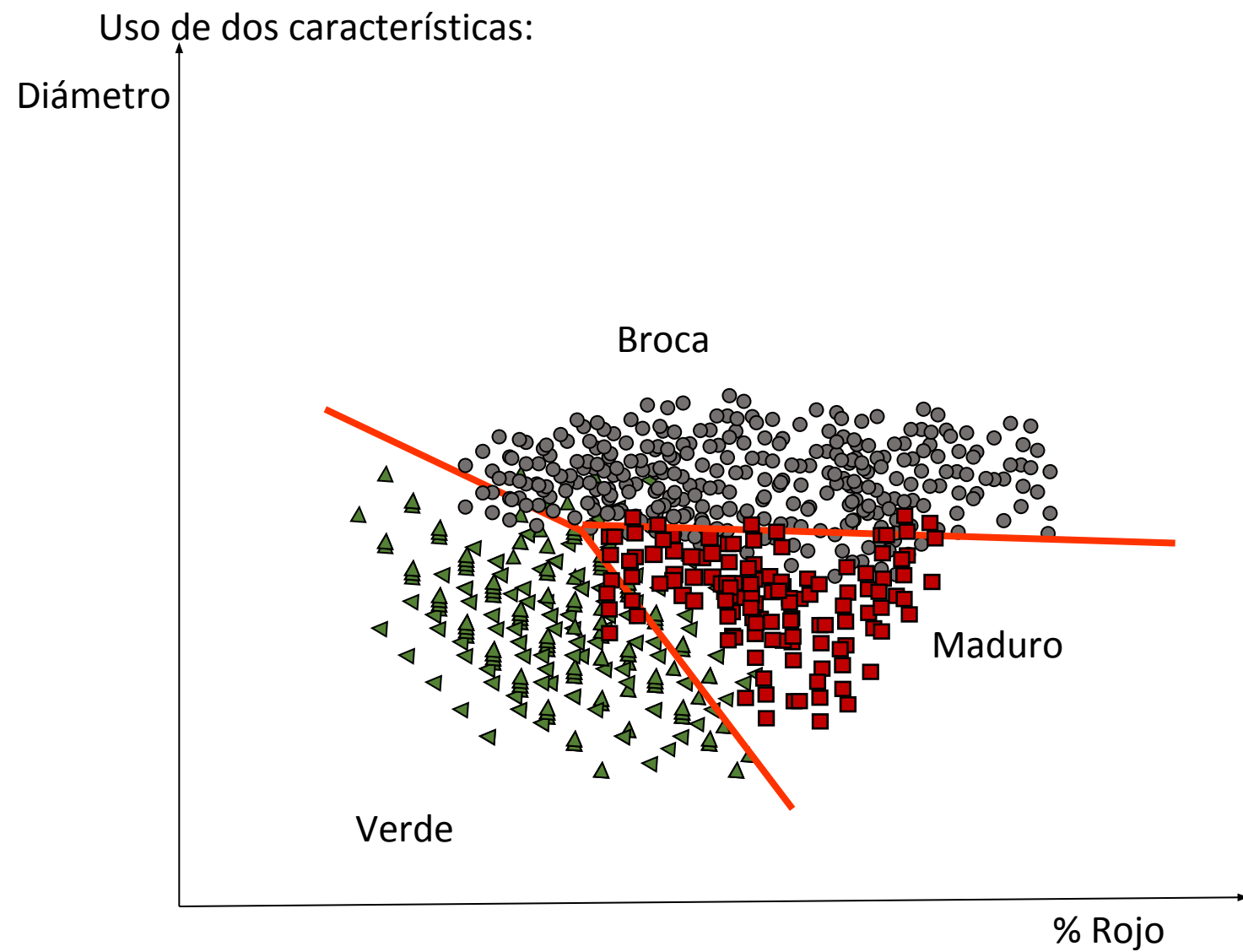
Diámetro Ombligo (Píxeles)

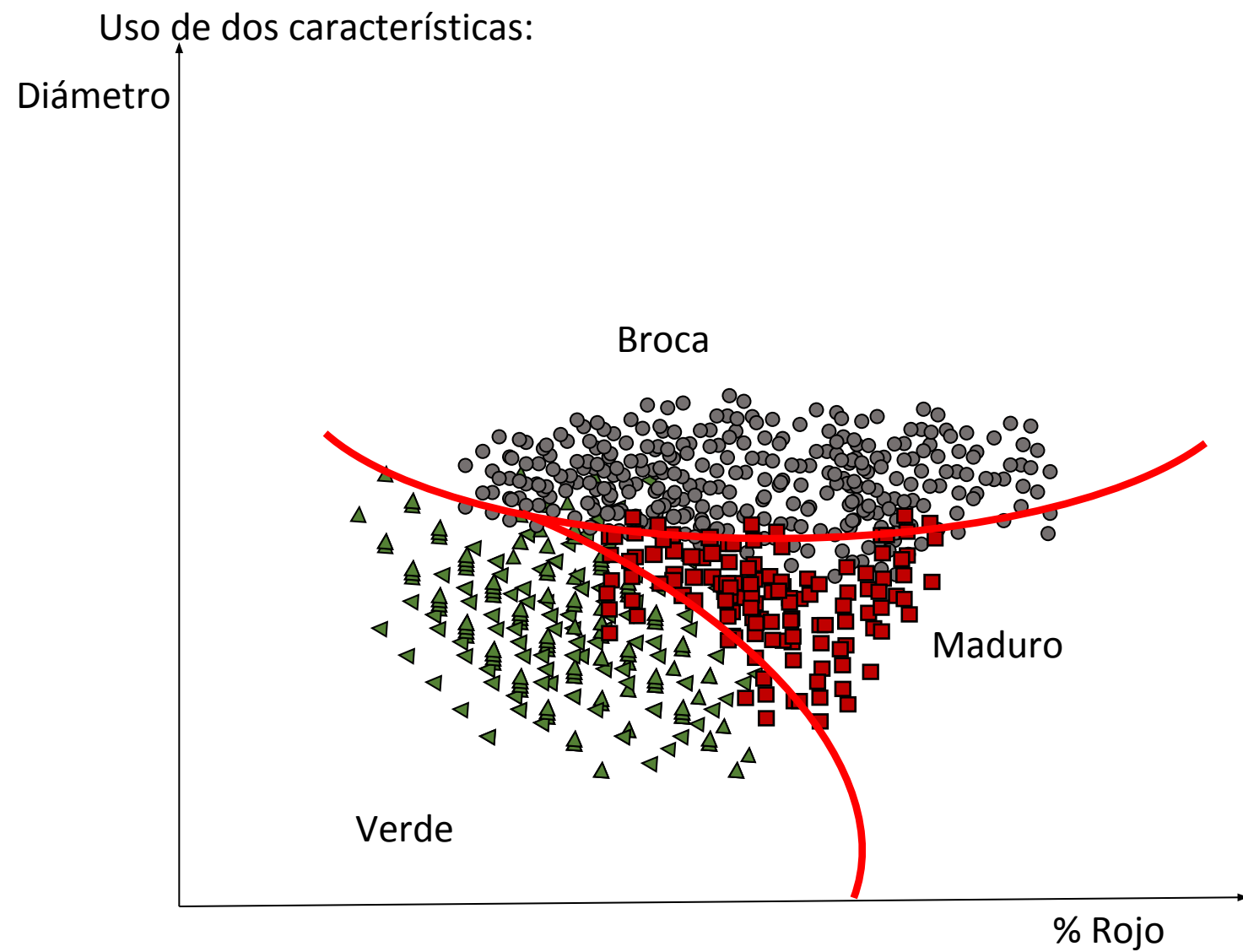
Maduro-01	17	Verde-01	13	Broca-01	25
Maduro-02	20	Verde-02	15	Broca-02	24
Maduro-03	13	Verde-03	16	Broca-03	21
Maduro-04	21	Verde-04	15	Broca-04	20
Maduro-05	18	Verde-05	15	Broca-05	24
Maduro-06	19	Verde-06	18	Broca-06	22
Maduro-07	16	Verde-07	20	Broca-07	23
:		:	21	:	
Maduro-50	19	Verde-50	19	Broca-50	19

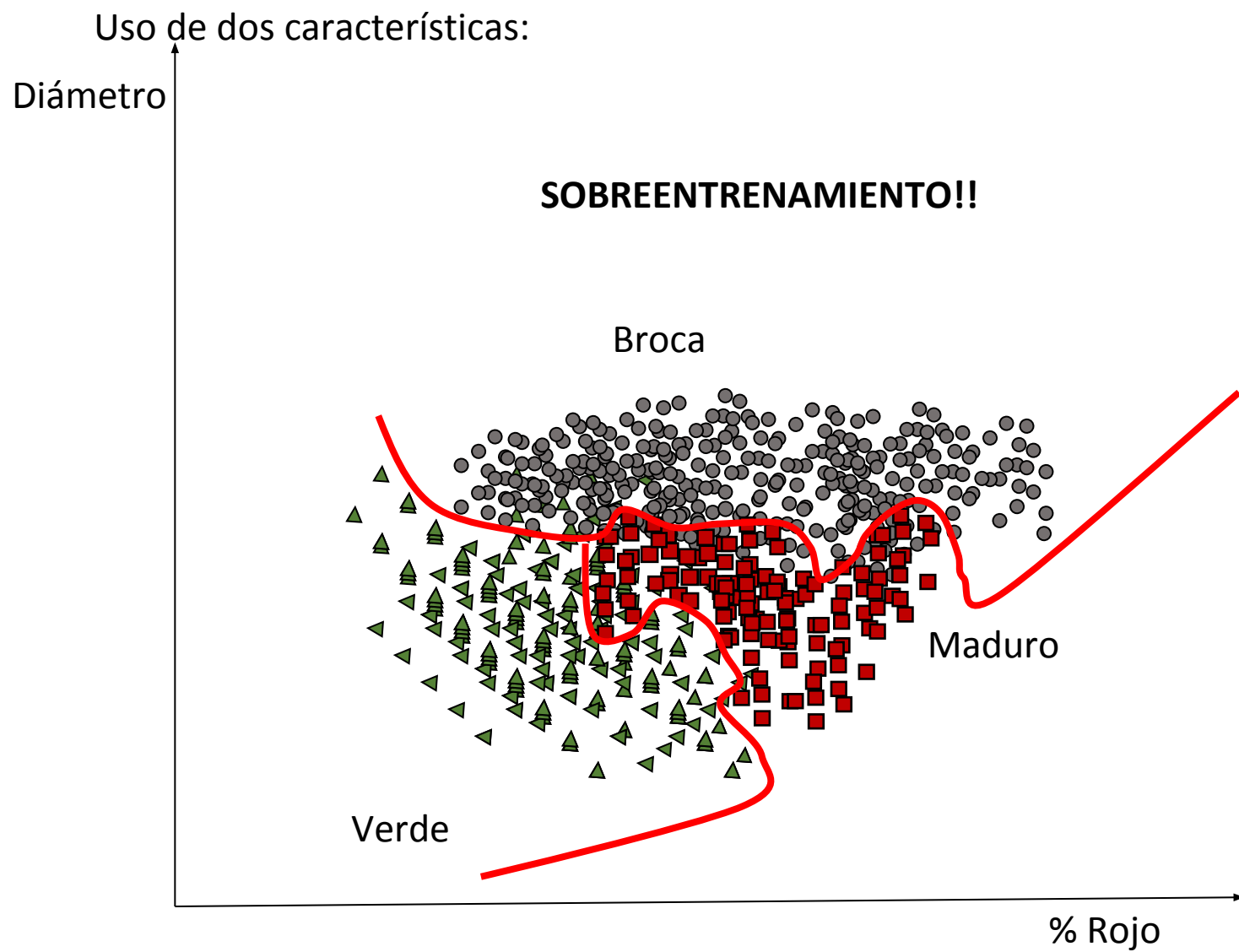
Histogramas:

Una característica



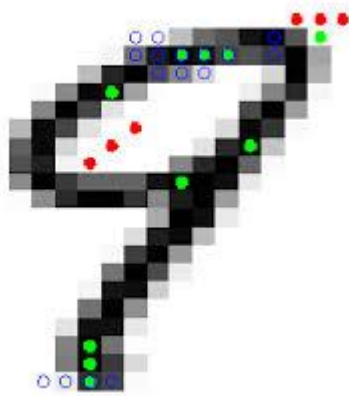




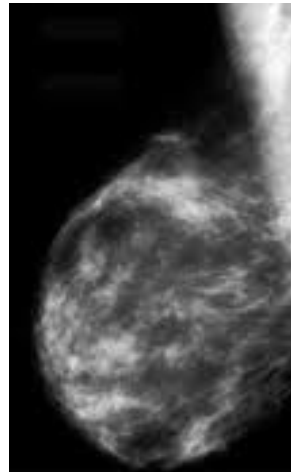


EXTRACCIÓN DE CARACTERÍSTICAS

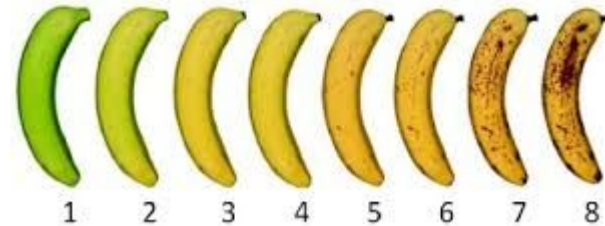
Después del proceso de segmentación es necesario extraer características de la señal 1D/2D, que permitan representarla inequívocamente. Para esto se obtienen características de color, textura, geométricas, topológicas, entre otras en el caso de imágenes



Geométricas y
Topológicas



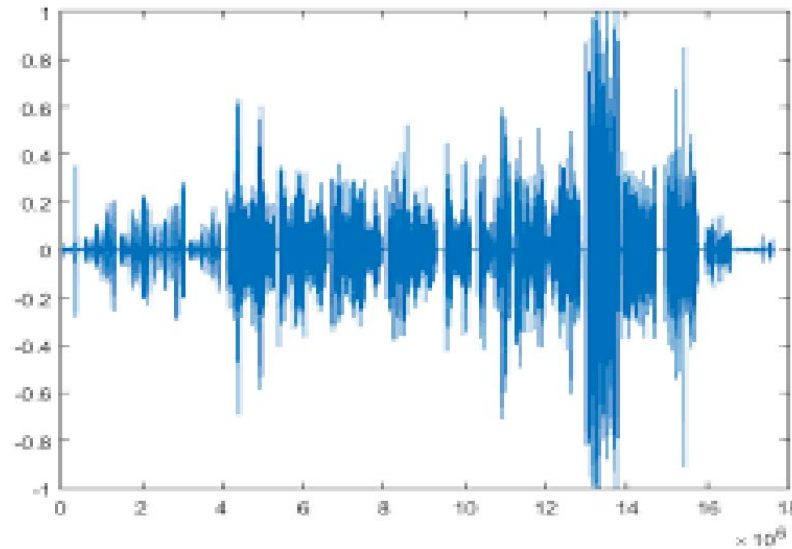
Textura



Color

EXTRACCIÓN DE CARACTERÍSTICAS

Mel Frequency Cepstral Coefficients (MFCC), Linear Predictive Coding (LPC) Coefficients, Perceptual Linear Prediction (PLP), Relative Spectral Transform Analysis (RASTA), GammaTone Log Energies (GTE), entre otros en el caso de audio.



EXTRACCIÓN DE CARACTERÍSTICAS

Sin importar el tipo de características estas deben cumplir con las siguientes condiciones:

Discriminantes: Maximizan la diferencia entre clases.

Independientes: Que no haya una relación entre los descriptores, de tal forma que cuando uno varié, no haya una variación por culpa de éste.

EXTRACCIÓN DE CARACTERÍSTICAS

Suficientes: De tal manera que permitan delimitar correctamente el objeto que están representando, sin dar lugar a representar otro objeto con las mismos valores de los descriptores.

Robustos: Deben ser, lo mayor posible, inmunes a diferentes factores como los cambios de iluminación, ruido, translación, rotación, escalado, deformación, falta de información.

Rápidos: Dependiendo del tipo de aplicación, en muchos casos es necesario que el calculo de los descriptores debe ser lo más rápido posible, con el fin de generar una respuesta del sistema en tiempo real.

APLICACIONES DE MACHINE LEARNING

Image Recognition

Speech Recognition

Social Media Services

Virtual Personal Assistants

Medical Diagnosis

Product Recommendations

Email Spam and Malware Filtering

INTRODUCCIÓN A LAS REDES NEURONALES ARTIFICIALES

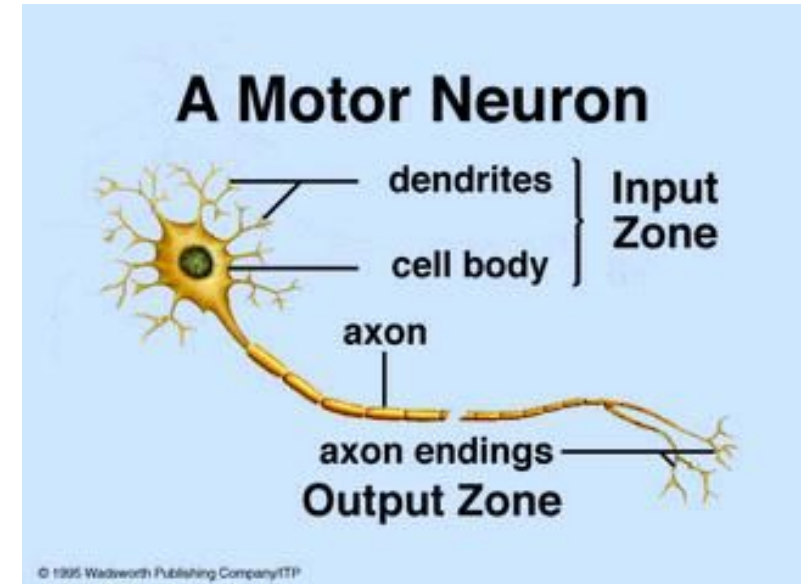
RED NEURONAL BIOLÓGICA

Cuerpo: Soma celular

Axon: Ramificación de la célula.

Dendritas: Ramificaciones de entrada

Sinapsis: Recoge información electro-química.



HISTORIA DE LAS REDES NEURONALES

- 1943: McCulloch y Pitts propusieron el primer modelo de neurona artificial.
- 1949: Hebb desarrolló un procedimiento matemático de aprendizaje.
- 1957: Rosenblatt propuso el modelo PERCEPTRÓN.
- 1960: Widrow propuso el modelo ADALINE.
- 1973: Anderson propuso un modelo llamado BSB *Brain state-in-a-Box*.
- 70s: Kohonen, hace una descripción y análisis de un conjunto de reglas adaptativas y propone el principio de aprendizaje competitivo.
- 70s: Grossberg, propone la clase de redes ART *Adaptive Resonance Theory*.
- 1982: Hopfield, inventó el algoritmo Backpropagation, devolviendo el interés en la computación neuronal.
- 2012: Aplicaciones industriales de DeepLearning

ENTRENAMIENTO DE LAS 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.

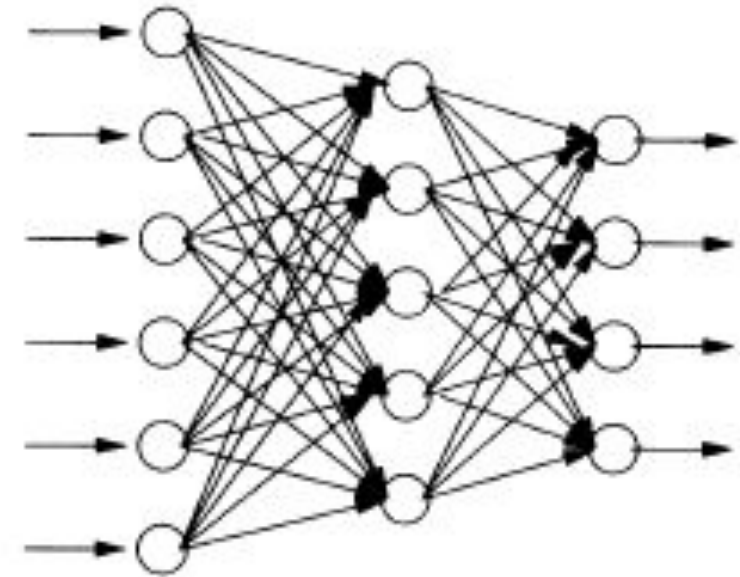
Representativo: Conjunto de aprendizaje diverso.

La convergencia en el entrenamiento se da por:

Número fijo de ciclos.

Error descienda de cierta cantidad.

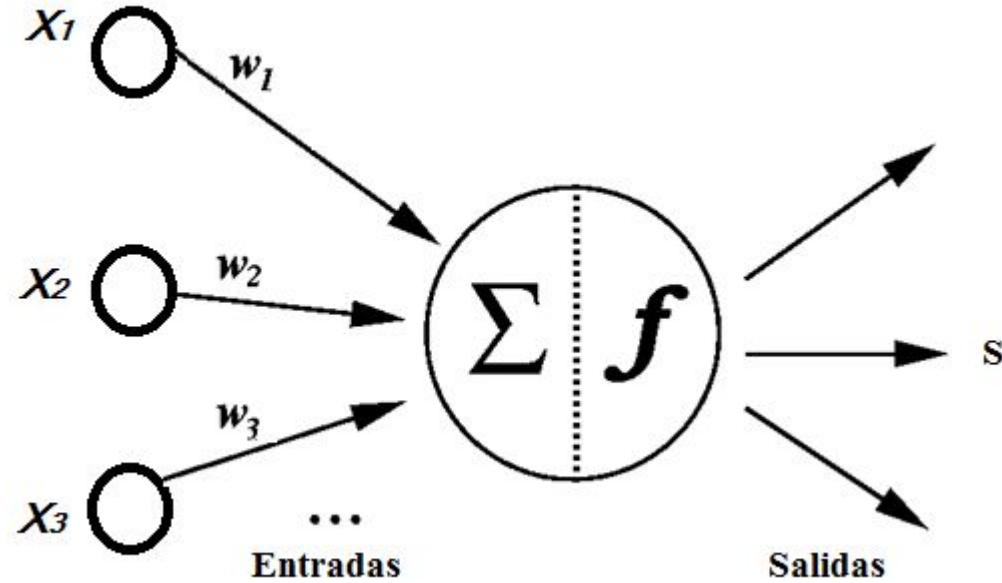
Modificación de los pesos sea irrelevante.



APLICACIONES DE LAS REDES NEURONALES ARTIFICIALES

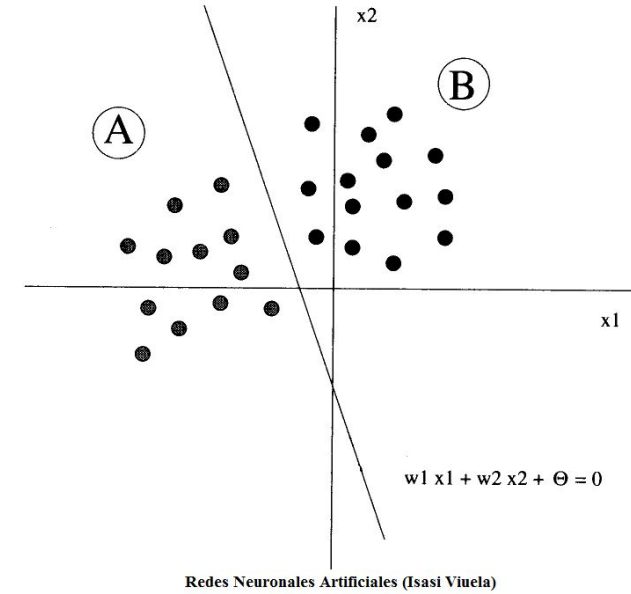
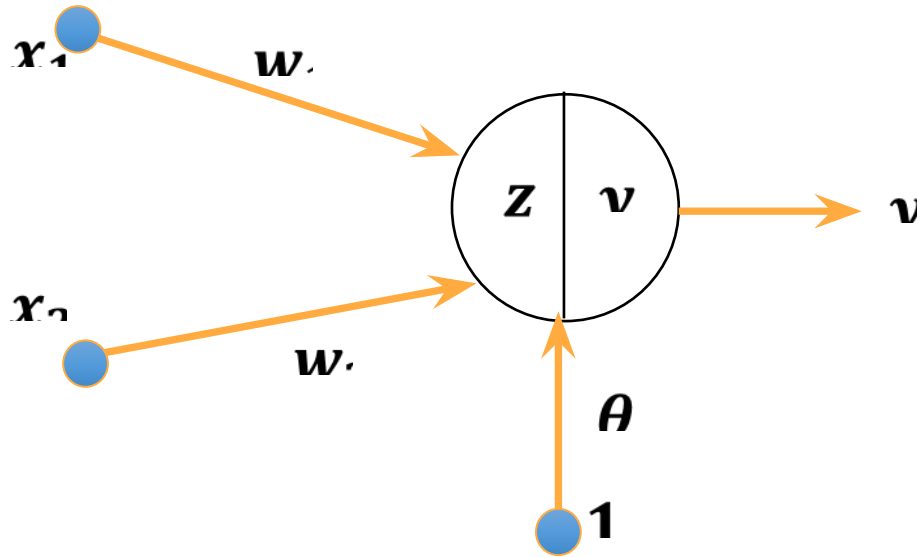
Aplicación	Tipo
Predicción	Adaline y Madaline Digital Neural Network Architecture (DNNA) Networks Perceptron Networks
Clasificación	Back-Propagation Counter-Propagation Delta-Bar-Delta (DBD) Networks Directed Random Search (DRS) Networks Functional-link Networks (FLN) Learning Vector Quantization (LVQ) Networks
Asociación	Bi-Directional Associative Memory (BAM) Networks The Boltzmann Machine Brain-State-in a Box Cascade-Correlation-Networks Hamming Networks Probabilistic Neural Network (PNN) Spatio-Temporal-Pattern Recognition (SPR)
Conceptualización	Adaptive Resonance Theory Networks (ART) Self-Organizing Maps (SOM)
Filtrado	Recirculation Networks
Optimización	Hopfield Networks

NEURONA McCulloch-PITTS



$$Y = \begin{cases} 1, & \sum_i w_i x_i > \theta \\ 0, & \text{en otros casos} \end{cases}$$

PERCEPTRON SIMPLE



$$z = \sum_{i=1}^n w_i x_i + \theta$$

Nivel de activación de la célula

$$y = f(z, \theta) \quad y = f\left(\sum_{i=1}^n w_i x_i + \theta\right)$$

Salida o Activación

Función de Activación

$$f(z) = \begin{cases} 1, & z > 0 \\ -1, & \text{en otros casos} \end{cases}$$

PROCESO DE APRENDIZAJE DEL PERCEPTRON SIMPLE

Siendo X cada uno de los patrones de entrenamiento y $d(x)$ su clase asociada con valores de $(1, -1)$, el proceso de aprendizaje tiene los siguientes pasos.

1. Dar valores aleatorios a los pesos w_i y al umbral θ .
2. Seleccionar un vector de entrada $\bar{X} = x_1, x_2, x_3, \dots, x_n$ del conjunto de ejemplos de entrenamiento.
3. Si $y \neq d(x)$ se genera una respuesta incorrecta de la red. Se debe modificar w_i en Δw_i de acuerdo a:

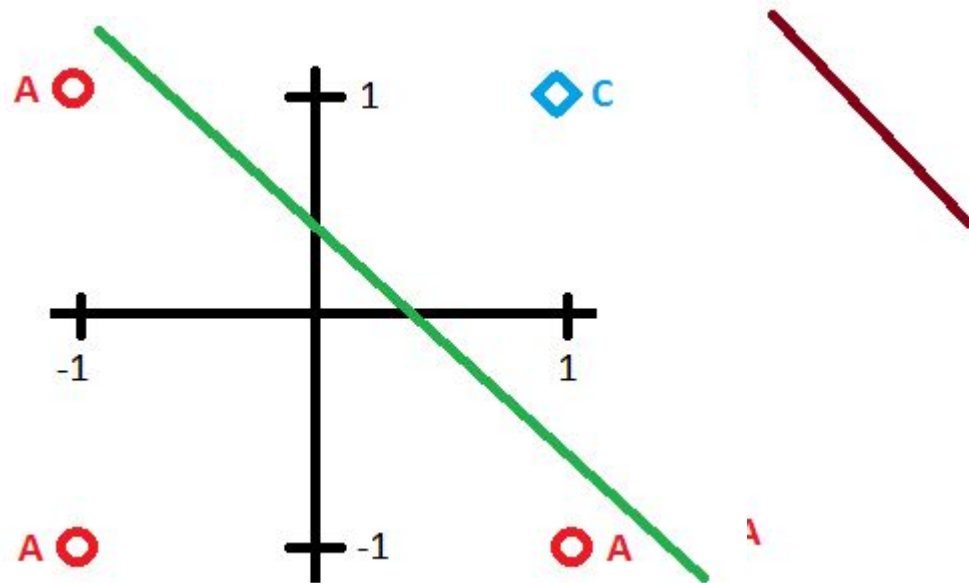
$$\Delta w_i = d(x)x_i$$

4. Si no se ha cumplido el criterio de terminación, retornar a 2.

TAREA

Realizar un programa para entrenar una red neuronal del tipo perceptrón simple para las funciones booleanas AND y OR de 2 entradas.

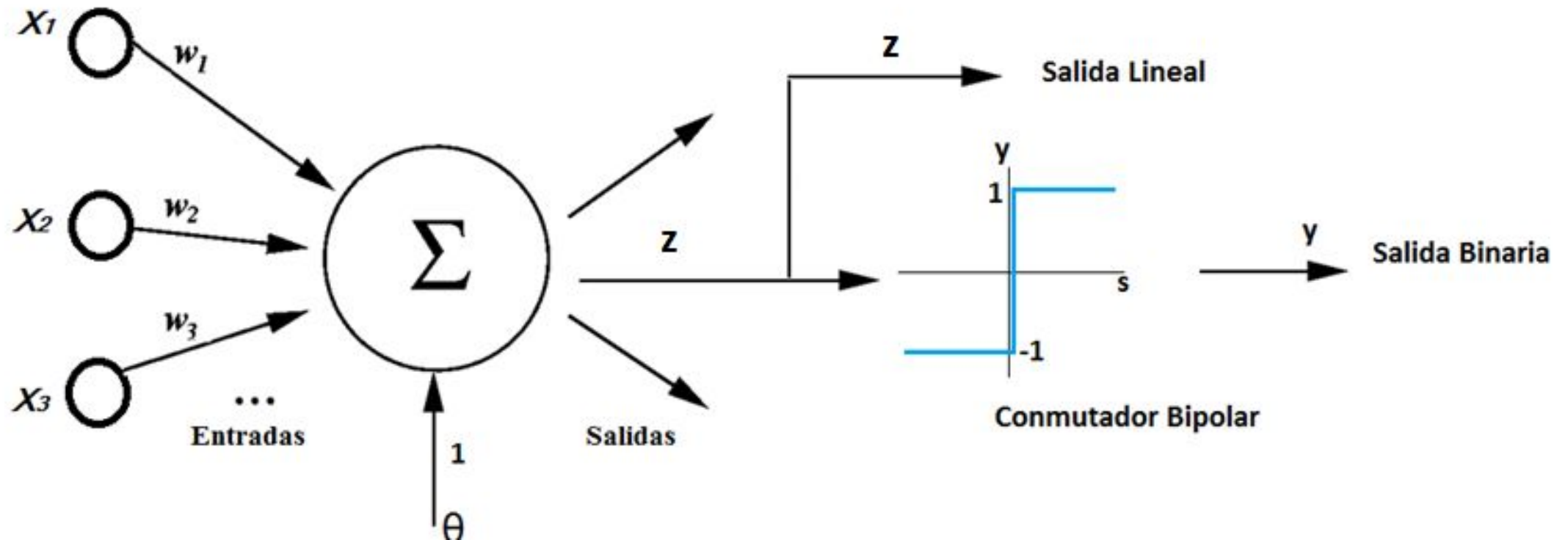
SOLUCIÓN DEL PERCEPTRON SIMPLE



ADALINE (ADAPTIVE LINEAR NEURON)

- 1960: Widrow Y Hoff propusieron el modelo ADALINE.
- ADALINE: Neurona Lineal Adaptativa.
- El objetivo del algoritmo de entrenamiento es minimizar el error cuadrático medio entre la salida real y la deseada, llamada **regla Delta**.
- MADALINE: Varias Neuronas en la Salida.
- Las aplicaciones de las redes ADALINE Y MADALINE, han estado asociadas al procesamiento de señales digitales, como filtros para la eliminación de ruido.
- La gran diferencia con el perceptrón simple es que este utiliza la salida de la función umbral para el aprendizaje, en cambio la regla delta utiliza directamente la activación de la célula, sin pasarla por ninguna función.

ARQUITECTURA ADALINE



REGLA DE APRENDIZAJE

$$E = \sum_{p=1}^m E^p = \frac{1}{2} \sum_{p=1}^m (d^p - y^p)^2$$

Error Cuadrático Medio

$$\Delta_p w_j = -\gamma \frac{\partial E^p}{\partial w_j}$$

Regla de Descenso del Gradiente

$$\frac{\partial A}{\partial x} = \frac{\partial A}{\partial y} \frac{\partial y}{\partial x}$$

Regla de la Cadena

$$y = \sum_{j=1}^n w_j x_j + \theta$$

$$\frac{\partial E^p}{\partial w_j} = \frac{\partial E^p}{\partial y^p} \frac{\partial y^p}{\partial w_j}$$



$\frac{\partial y^p}{\partial w_j} = x_j$
$\frac{\partial E^p}{\partial y^p} = -(d^p - y^p)$



$$\Delta_p w_j = \gamma (d^p - y^p) x_j$$



Factor de Aprendizaje

PROCESO DE APRENDIZAJE DEL ADALINE

Siendo X cada uno de los patrones de entrenamiento y $d(x)$ su clase salida asociada, el proceso de aprendizaje tiene los siguientes pasos.

1. Dar valores aleatorios a los pesos w_i y al umbral θ .
2. Seleccionar un vector de entrada $\bar{X} = x_1, x_2, x_3, \dots, x_n$ del conjunto de ejemplos de entrenamiento.
3. Calcular la diferencia entre la salida deseada y la real. $(d^p - y^p)$ y aplicar $\Delta_p w_j = \gamma (d^p - y^p) x_j$
4. Modificar los pesos. $w_j = w_j + \Delta_p w_j$
5. Si no se ha cumplido el criterio de terminación, retornar a 2.

EJERCICIO ADALINE

Realizar un programa para entrenar una red neuronal adaline, que haga las veces de decodificador binario a decimal.

X3	X2	X1	Y
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7