



MODELOS SUPERVISADOS

REDES NEURONALES MLP

JAIME MIRANDA

(jmiranda@dii.uchile.cl)

Departamento de Ingeniería Industrial

Universidad de Chile

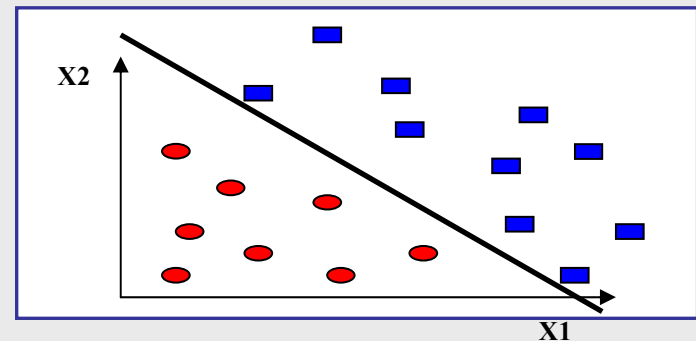
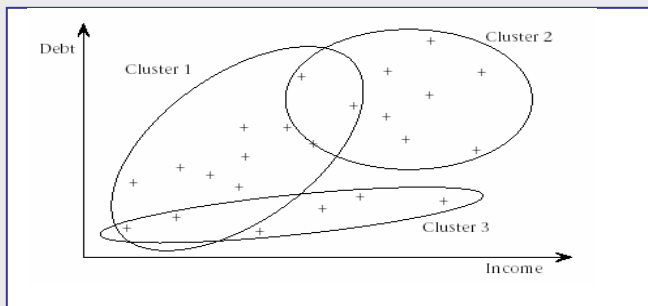
CONCEPTOS BÁSICOS

→ Medida de distancia

- Prototipo o centro de clase más cercana
- Entre más cerca mayor pertenencia de a la clase

→ Hipersuperficies

- Clasificación de acuerdo a si los objetos están a uno u otro lado de una hipersuperficie o conjunto de hiperplanos



Regla de Bayes

$$p(C_K / X_L) = \frac{p(X_L / C_K) * p(C_K)}{p(X_L)}$$

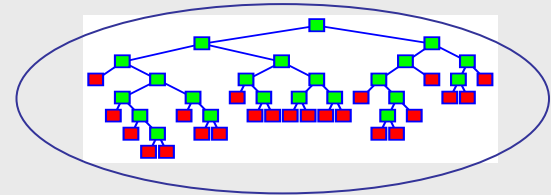
Donde:

C_k denota la clase k del atributo elegido como base.

X_L son los atributos a los cuales se condicionara en probabilidad.

MÉTODOS SUPERVISADOS

- Redes neuronales
- Árboles de decisión
- SVM



MÉTODOS NO SUPERVISADOS

- Fuzzy C-means (Cluster)
- Mapas Kohonen

NUEVAS TÉCNICAS

- MCS (Multiclassifier Systems)
- Clamping (Heurística selección de atributos)

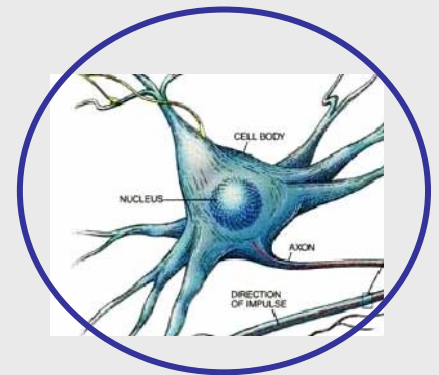


DIAGRAMA DE APRENDIZAJE SUPERVISADO

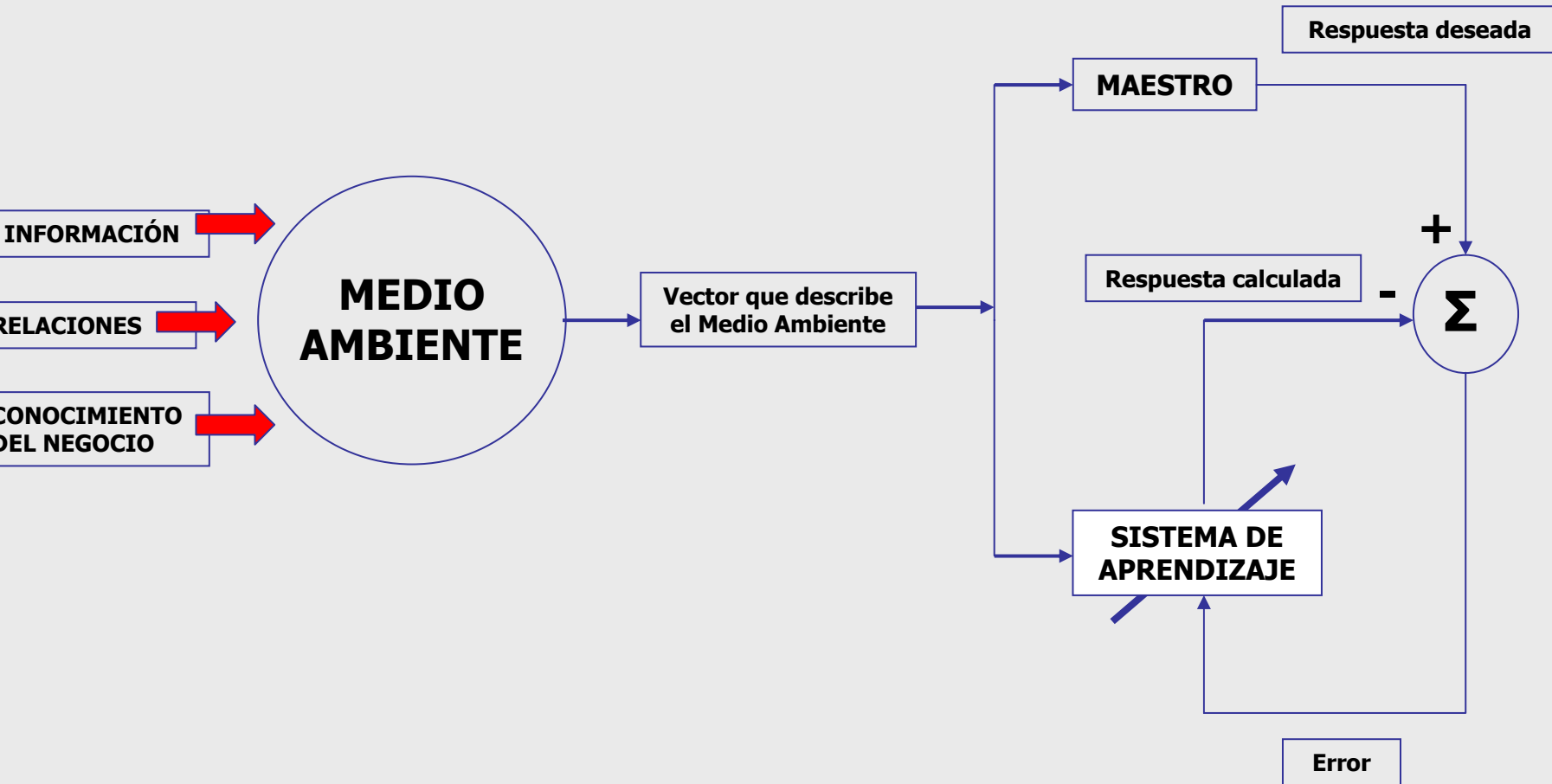
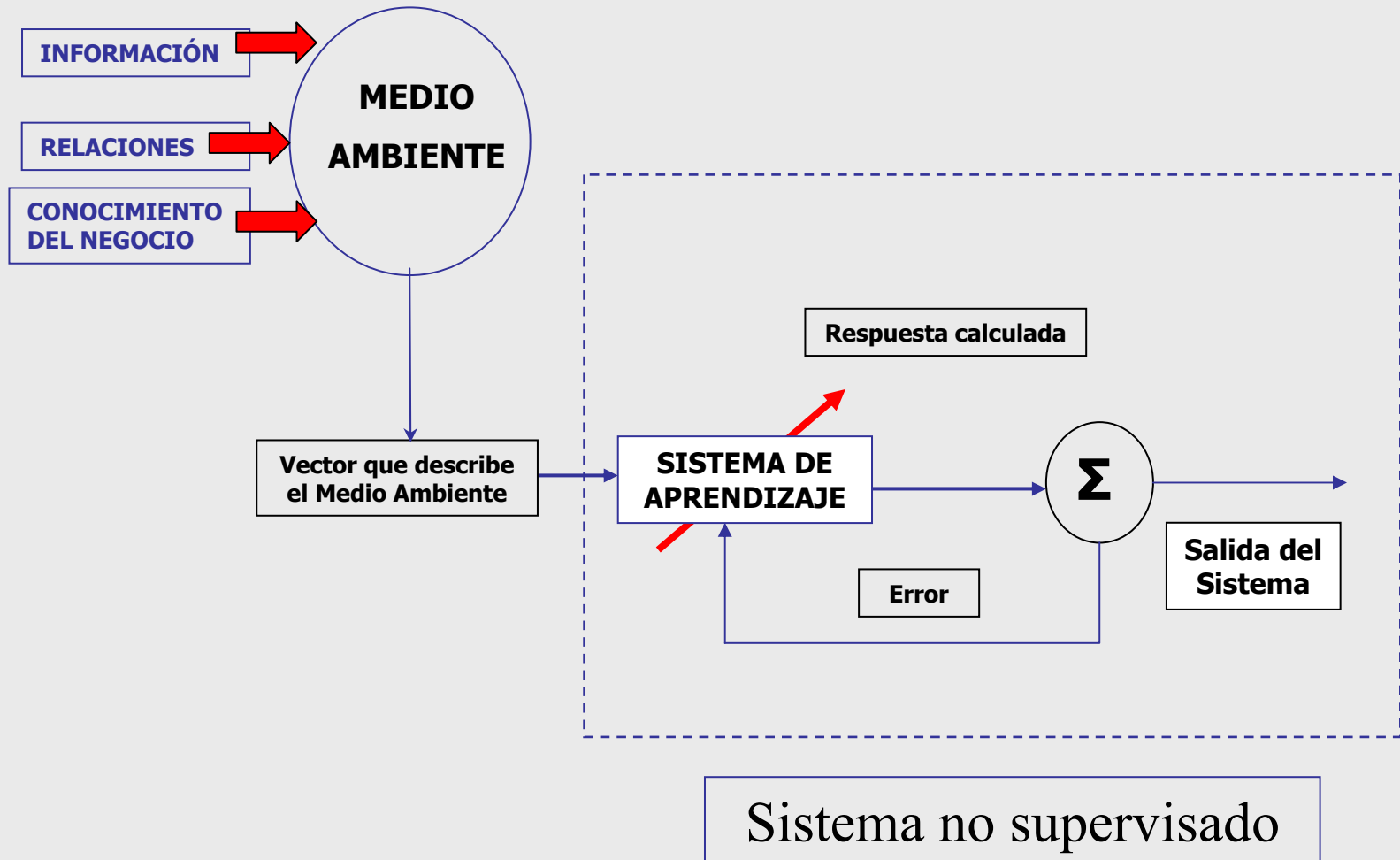


DIAGRAMA DE APRENDIZAJE NO SUPERVISADO

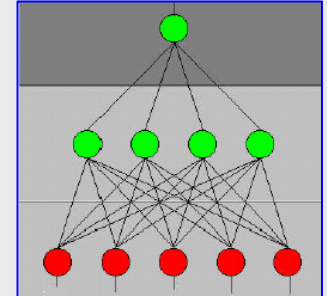


Scoring de riesgo crediticio para mutuos hipotecarios

→ Técnica de minería usada : Red neuronal multicapa

Modelo predictivo de fugas de cuentacorrentistas

→ Técnica de minería usada: SVM



Modelo predictivo de fugas de cuentacorrentistas

→ Técnica de minería usada: Red neuronal multicapa especializada

Modelo predictivo de ofertas focalizadas

→ Técnica de minería usada: Red neuronal multicapa – Árbol de decisión

Modelo detección de fraudes en transacciones electrónicas de fondos

→ Técnica de minería usada: MCS

- SVM
- Árboles de decisión
- Mapas de Kohonen

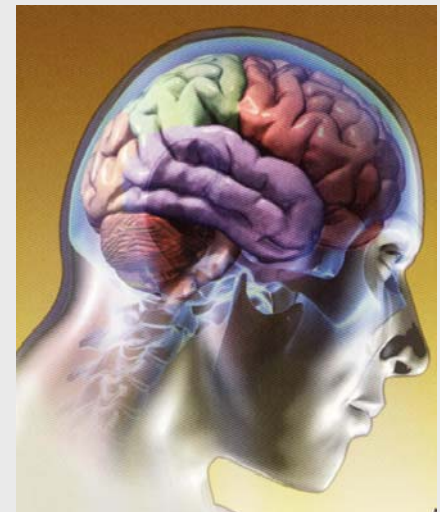
PROBLEMA: ALTO NUMERO DE FRAUDES EN TARJETAS DE CRÉDITO

- Construir un modelo de predicción de fraudes en tarjetas de crédito
- La empresa posee un Datawarehouse con la información transaccional y de caracterización (perfil) de los clientes
- La base consta de 2.000 clientes descritos por 15 variables
- Se desea encontrar un patrón característico que describa el comportamiento de fraude



APRENDIZAJE

- “El aprendizaje es una habilidad de la que disponen gran parte de los sistemas naturales para **adaptarse** al entorno en el que vive”.
- “Adquisición de conocimiento de un proceso por medio del análisis, ejercicio o **experiencia**”.
- “Un proceso por el cual los parámetros libres del sistema se **adaptan a través de un proceso continuo** de estimulación a partir del entorno en el que el sistema está inmerso”.



ENFOQUE BAYESIANO

$$p(C_K / X_L) = \frac{p(X_L / C_K) * p(C_K)}{p(X_L)}$$

DONDE:

$$\sum_i P(C_i) = 1$$

$$\sum_k P(X_k) = 1$$

C_k denota la clase k del atributo elegido como base.

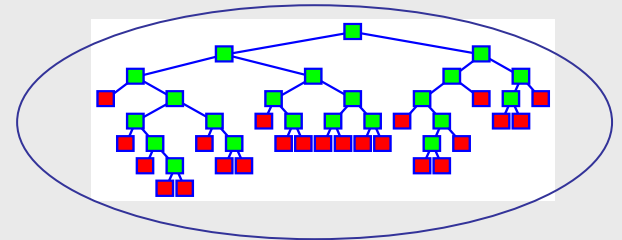
X_L son los atributos a los cuales se condicionara en probabilidad

REGLA DE DECISIÓN DE BAYES

X_k pertenece a la clase C_j si y solo si:

$$g(x_k) > g(x_k)$$

$$P(C_j / x_k) > P(C_i / x_k)$$



$$f: \mathbf{X} \in \mathbb{R}^n \rightarrow \mathbb{R}$$

$$\mathbf{X} = (x_1, x_2, \dots, x_n)$$

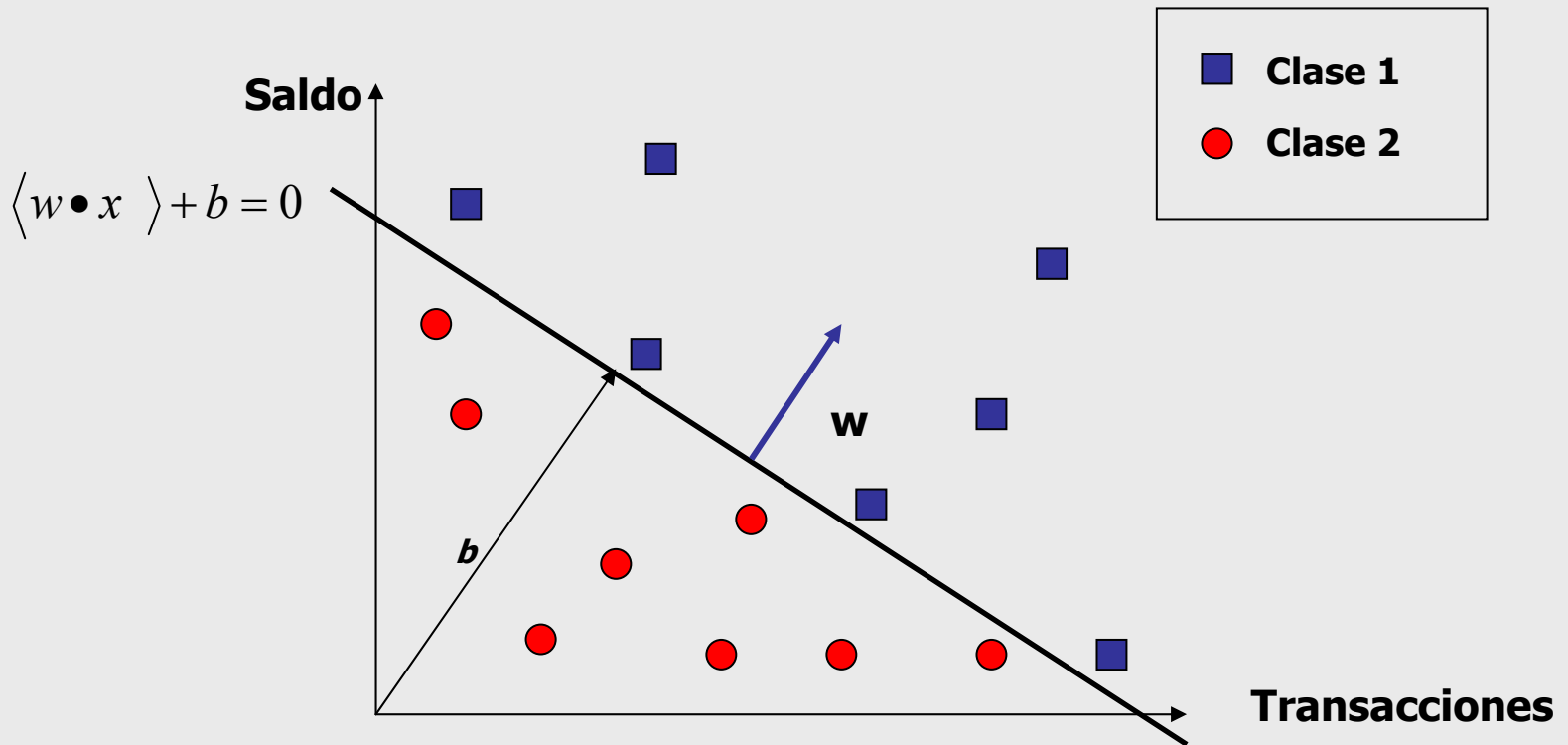
EJEMPLO: Clasificación binaria

$$f(\mathbf{X}) \geq 0 \quad \rightarrow \quad \text{Clase 1}$$

$$f(\mathbf{X}) \leq 0 \quad \rightarrow \quad \text{Clase 2}$$

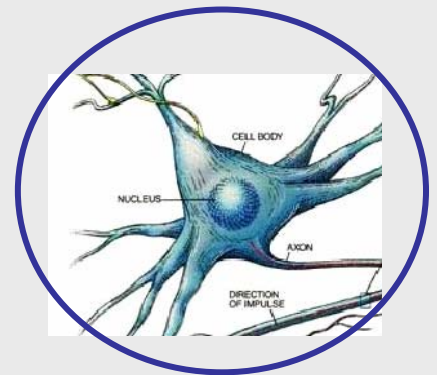
CLASIFICACIÓN LINEAL (2)

$$f(X) = \langle w \bullet x \rangle + b$$



ALGUNAS NOCIONES

- Propuesto en 1956 por Frank Rosenblatt.
- Fue objeto de gran interés a comienzos de los 60's.
- Primer algoritmo iterativo para clasificación lineal.
- Este algoritmo garantiza encontrar un hiperplano separador de clases, para datos linealmente separables.
- Unidad básica de la arquitectura de las redes neuronales.
- Se basa en una representación neuronal biológica.



MODELO

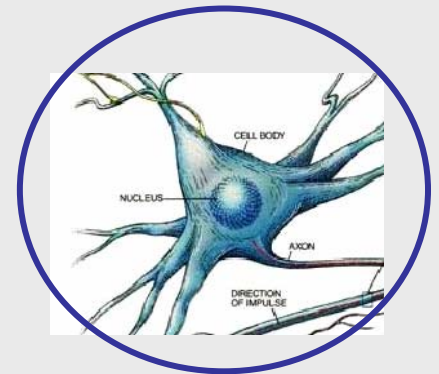
- Una neurona con pesos sinápticos y nivel umbral ajustable.
- Neurona biológica.

PROPÓSITO

- Clasificar estímulos externos de un objeto respecto a una clase.

APRENDIZAJE

- Determinar el vector de pesos óptimo (w) que clasifique bien a cada objeto.



UNIDADES DE ENTRADA

- Elementos que estimulan la red.
- Atributos o variables de entrada.

CONEXIONES

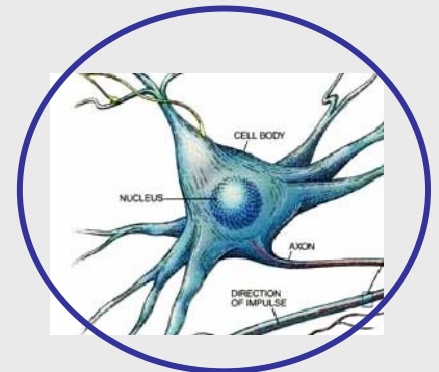
- Por las que se propaga la señal que conforma el patrón.
- Pesos w_i indican que tan fuerte es el atributo.

COMBINADOR LINEAL

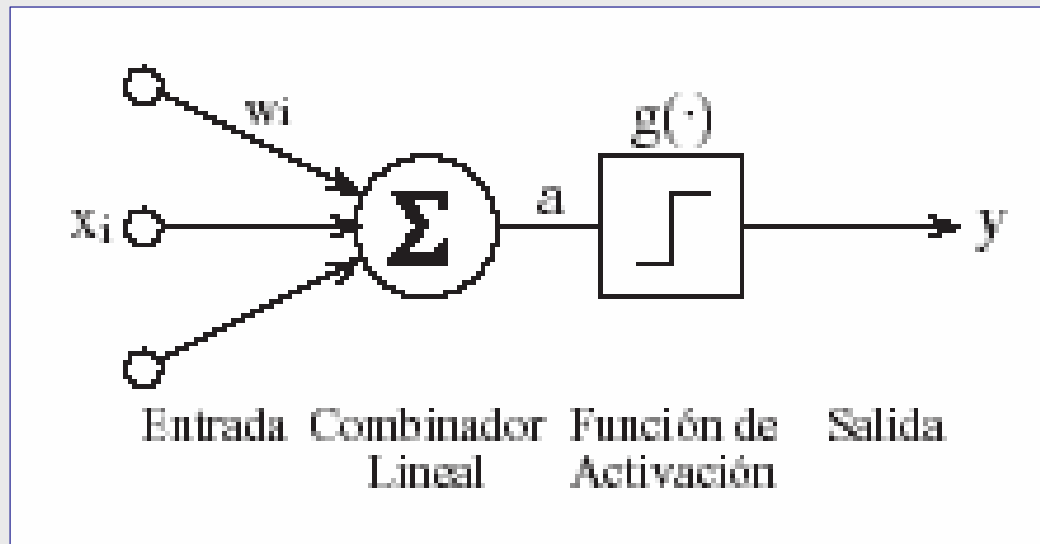
- Capta las señales y las combina en una sola.

FUNCIÓN DE ACTIVACIÓN

- Activa o no la señal.



MODELO GENERAL



DEFINICIONES

→ Vector de pesos

$$W = \langle w_0, w_1, \dots, w_n \rangle$$

→ Conjunto de ejemplos

$$E = \langle (x^1, t^1), (x^2, t^2), \dots, (x^p, t^p) \rangle$$

→ Salidas del modelo

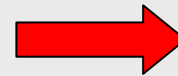
$$t^u = +1 \quad t^u = -1$$

ALGORITMO PERCEPTRON

1. Inicializar $W=(w_1,...w_n) = 0$.
2. Seleccionar un ejemplo $X^u=(x^u,t^u)$ en orden cíclico o al azar.
3. Si W clasifica correctamente a X^u , es decir:

$$\langle W \bullet X^u \rangle > 0 \quad \text{y} \quad t^u = +1$$

$$\langle W \bullet X^u \rangle < 0 \quad \text{y} \quad t^u = -1$$



No tomar ninguna acción

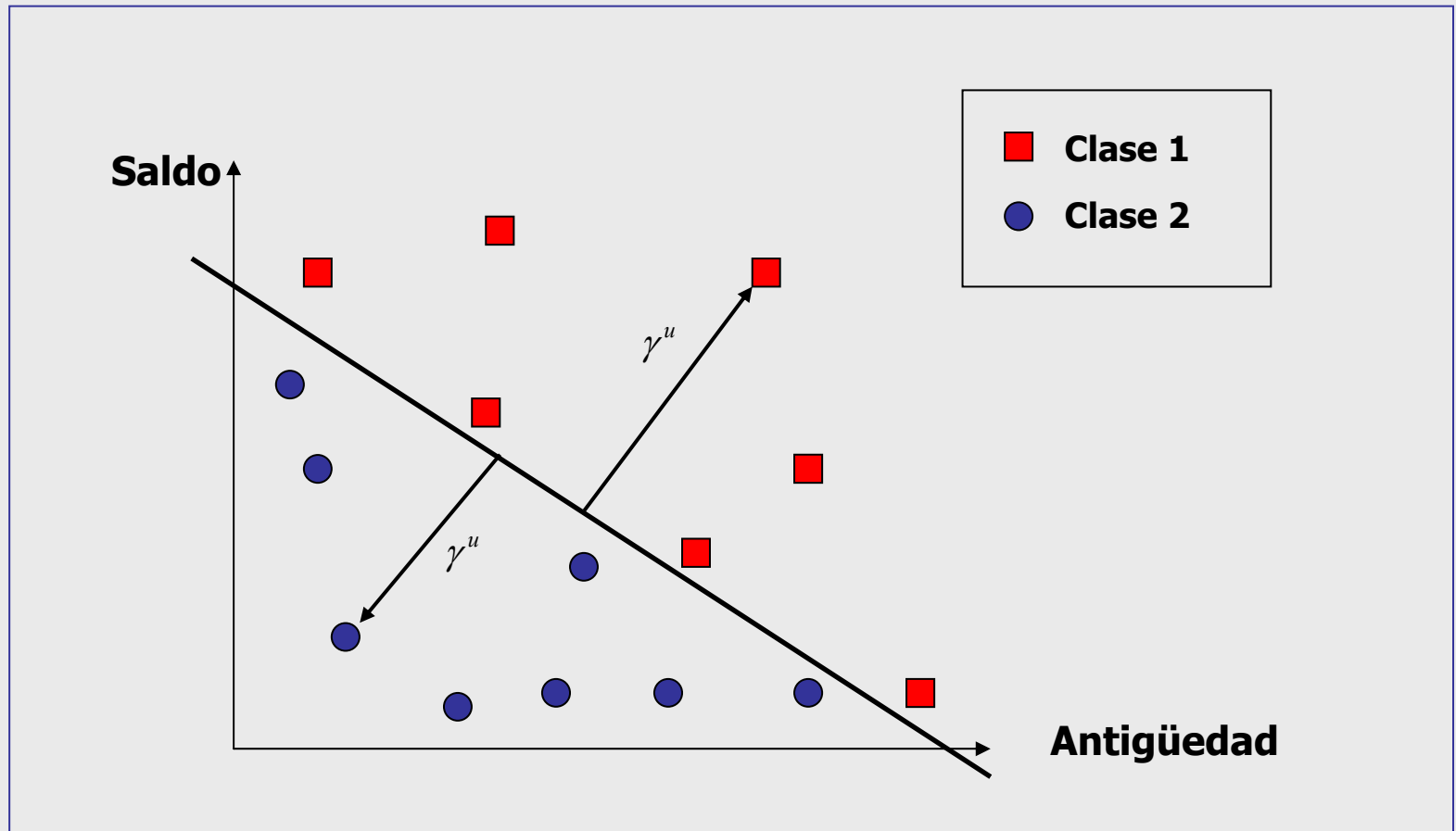
4. Si clasifica mal:

$$W'' = W + t^u x^u$$

5. Volver a 2.

MARGEN EN FORMA GRÁFICA

$$\gamma^u = t^u * \langle (w \bullet x^u + b) \rangle$$

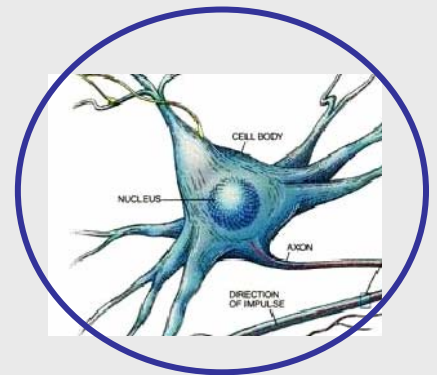


CONVERGENCIA

- El algoritmo perceptron converge siempre en tiempo finito, para un conjunto separable y finito de ejemplos

PESOS CÍCLICOS ACOTADOS

- El conjunto de pesos que algoritmo recorre para cualquier problema, sea éste separable o no, es acotado



EN TIEMPO FINITO EL ALGORITMO PRODUCIRÁ:

- Un vector de pesos que satisfaga todos los ejemplos
 - Conjunto linealmente separable
- Volverá a visitar un vector de pesos
 - Conjunto no linealmente separable

Test de separabilidad lineal

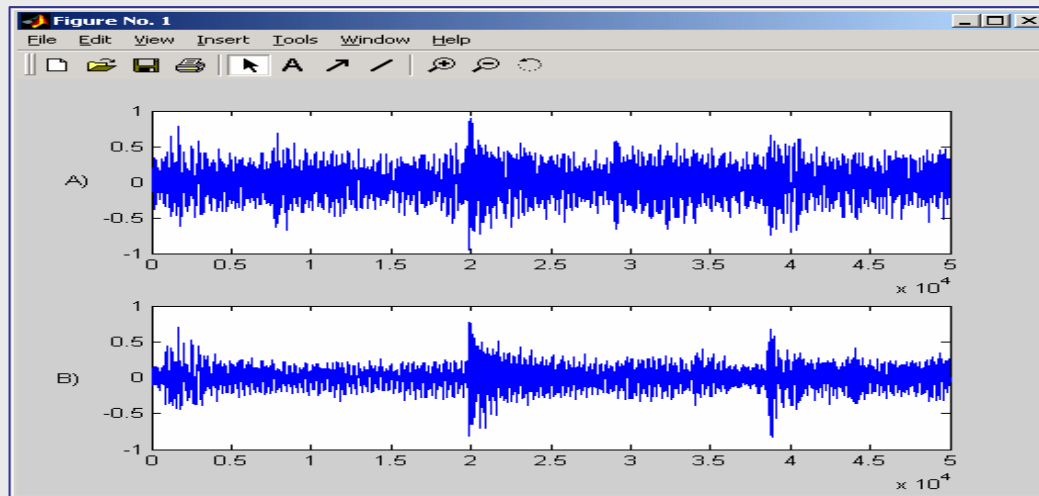
PROBLEMAS:

No se conoce cota de tiempo

Costoso: almacenar pesos pasados

CANCELADORES DE RUIDO (ADALINE)

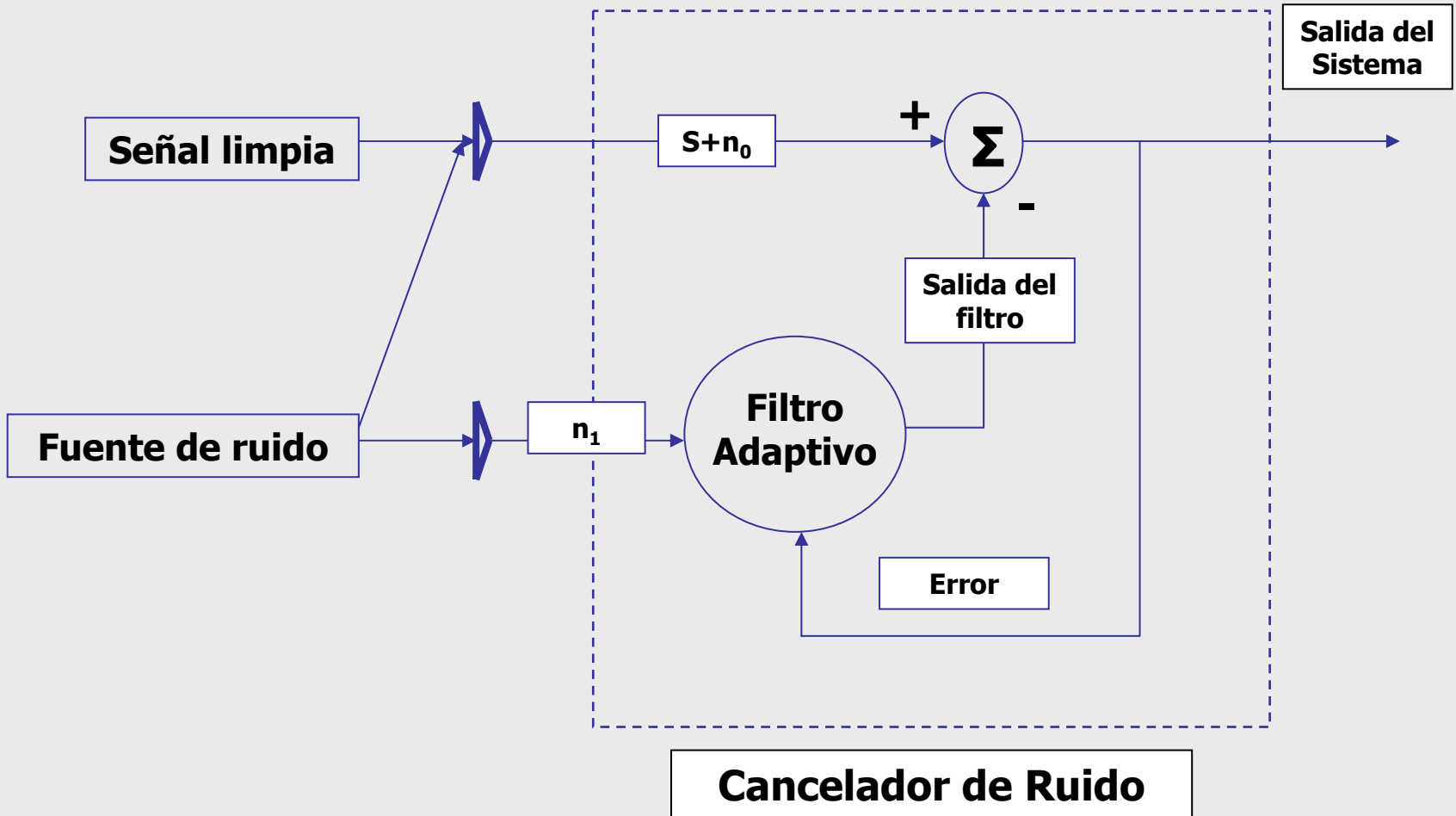
- Fueron introducidos por Widrow
- Limpieza de una señal acústica que posee ruido
- Uso de una estimación de una serie temporal



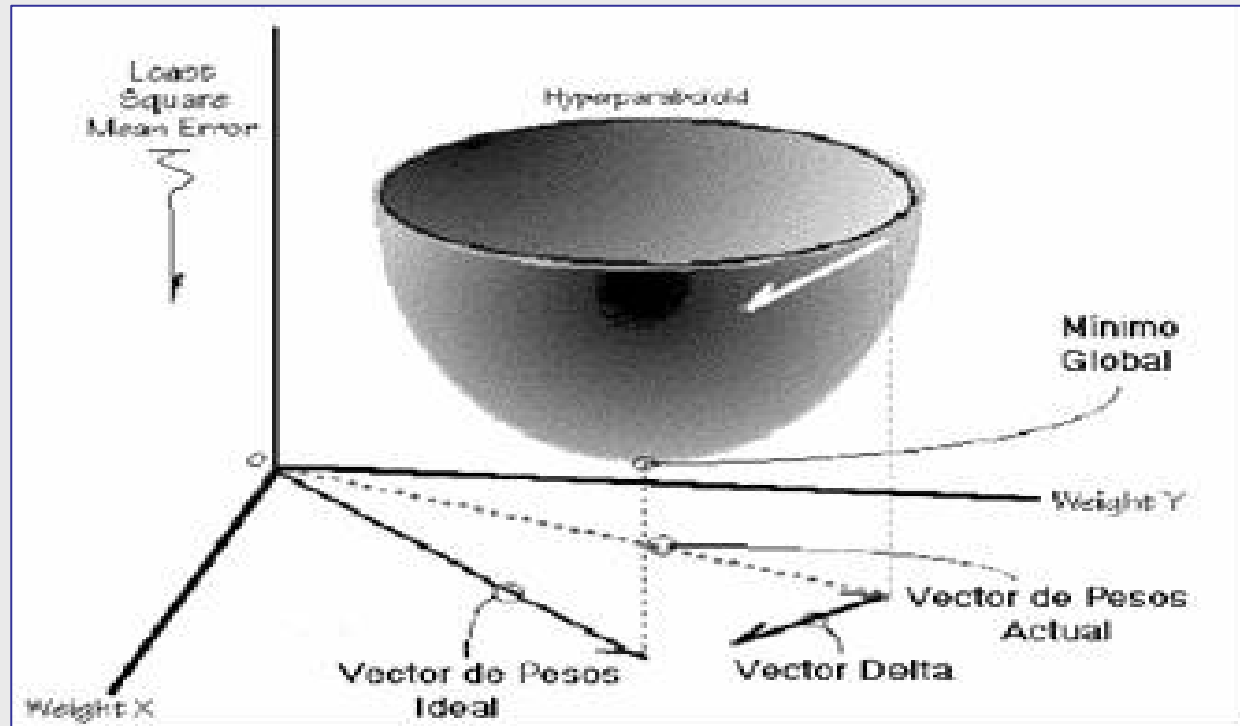
Señal sucia

Señal limpia

ESQUEMA GENERAL CANCELADOR DE RUIDO

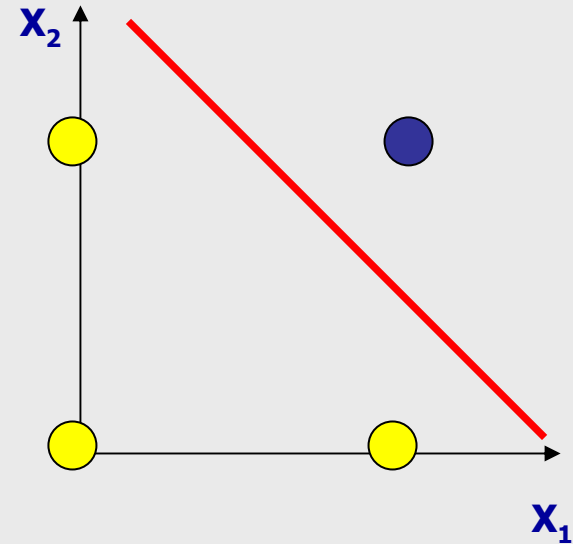


BÚSQUEDA DE PESOS ÓPTIMOS



SEPARABILIDAD EN FUNCIONES LINEALES

FUNCION BOOLEANAS		
X1	X2	AND
1	1	1
0	1	0
1	0	0
0	0	0

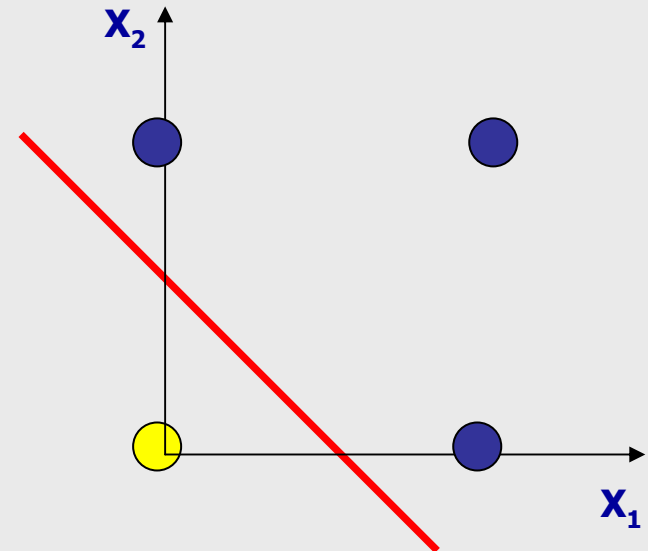


Los valores amarillos valen uno (TRUE)

Los valores azules valen cero (FALSO)

SEPARABILIDAD EN FUNCIONES LINEALES (2)

FUNCION BOOLEANAS		
X1	X2	OR
1	1	1
0	1	1
1	0	1
0	0	0

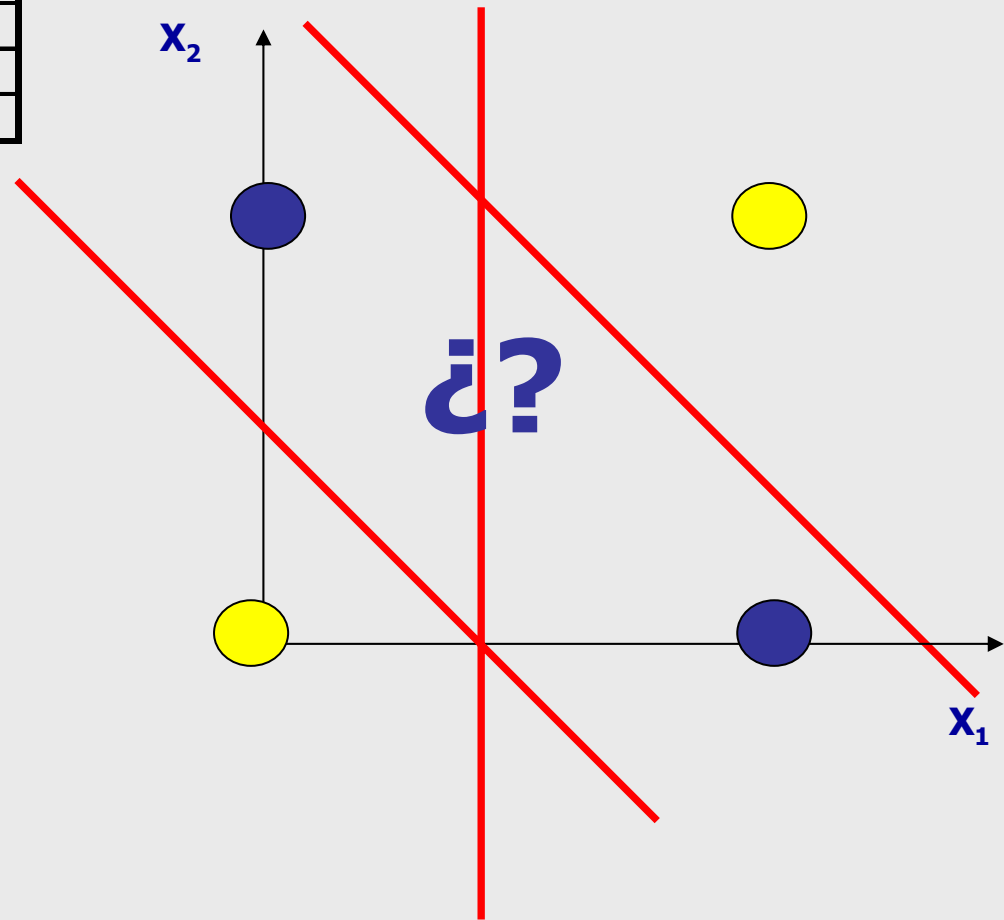


Los valores amarillos valen uno (TRUE)

Los valores azules valen cero (FALSO)

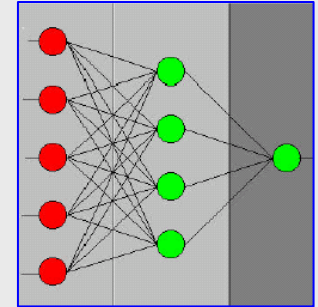
UN PEQUEÑO PROBLEMA

FUNCION BOOLEANAS		
X1	X2	XOR
1	1	0
0	1	1
1	0	1
0	0	0



Aplicaciones en la industria

- Retención o fuga de clientes
- Detección de fraudes
- Scoring



Fortalezas

- Fuerte en lo referente a la modelación no lineal
- Trabaja tanto con variables categóricas como continuas
- Alta aplicabilidad (variadas áreas de estudio)

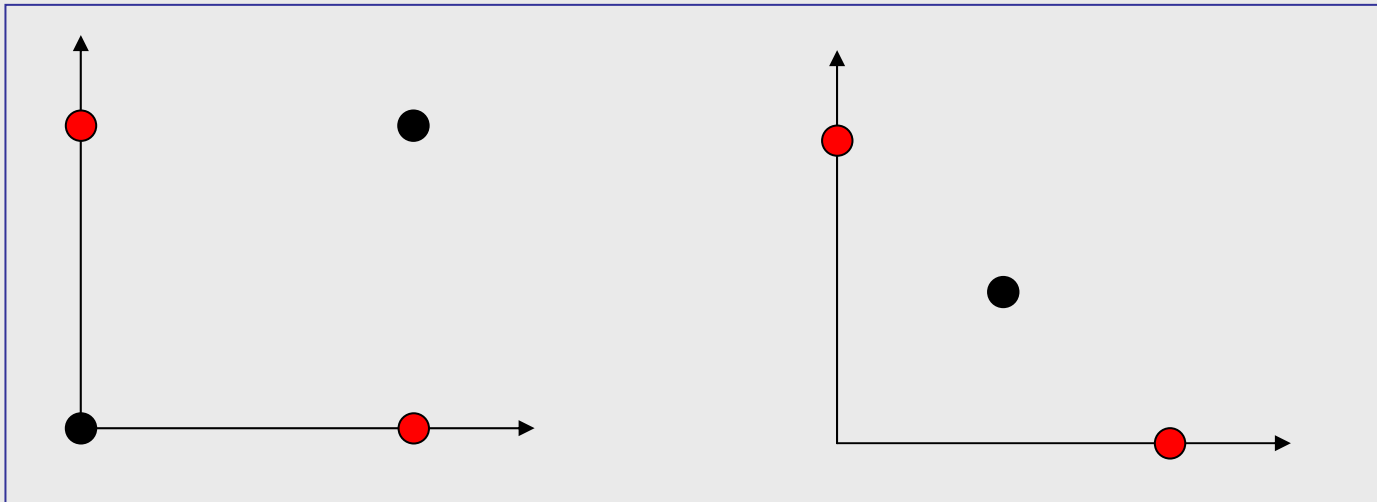
Debilidades

- Difícil interpretación de las relaciones entre las variables (Heurísticas)
- Sobreajuste

La gran mayoría de problemas no son linealmente separables

- No es posible usar el modelo perceptron. **Modelo limitado**
- No existe ningún hiperplano separador.

Se buscan funciones no lineales que definen las hipersuperficies



REPRESENTACIÓN DE FUNCIONES

- Cualquier función booleana de un número finito de entradas puede representarse en forma exacta por un patrón multicapas

HETCH-NIELSEN

- Cualquier función continua dentro de un cubo n -dimensional puede implementarse en forma exacta por una red con una capa oculta

HORNIK

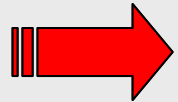
- La función puede ser representada con una red multicapa, siempre y cuando tenga un número adecuado de **unidades escondidas**

KOLGOMOROV

- Una capa oculta es suficiente para la aproximación de cualquier función

ARQUITECTURA RED NEURONAL MLP

Atributos



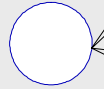
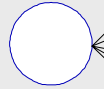
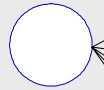
x_1

x_2

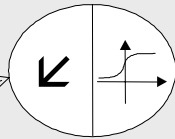
...

x_i

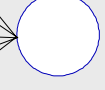
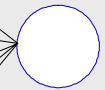
Capa de entrada



Capa oculta



Capa de salida



Clases



1

0

Construyen las distintas dimensiones de los polígonos (fronteras).

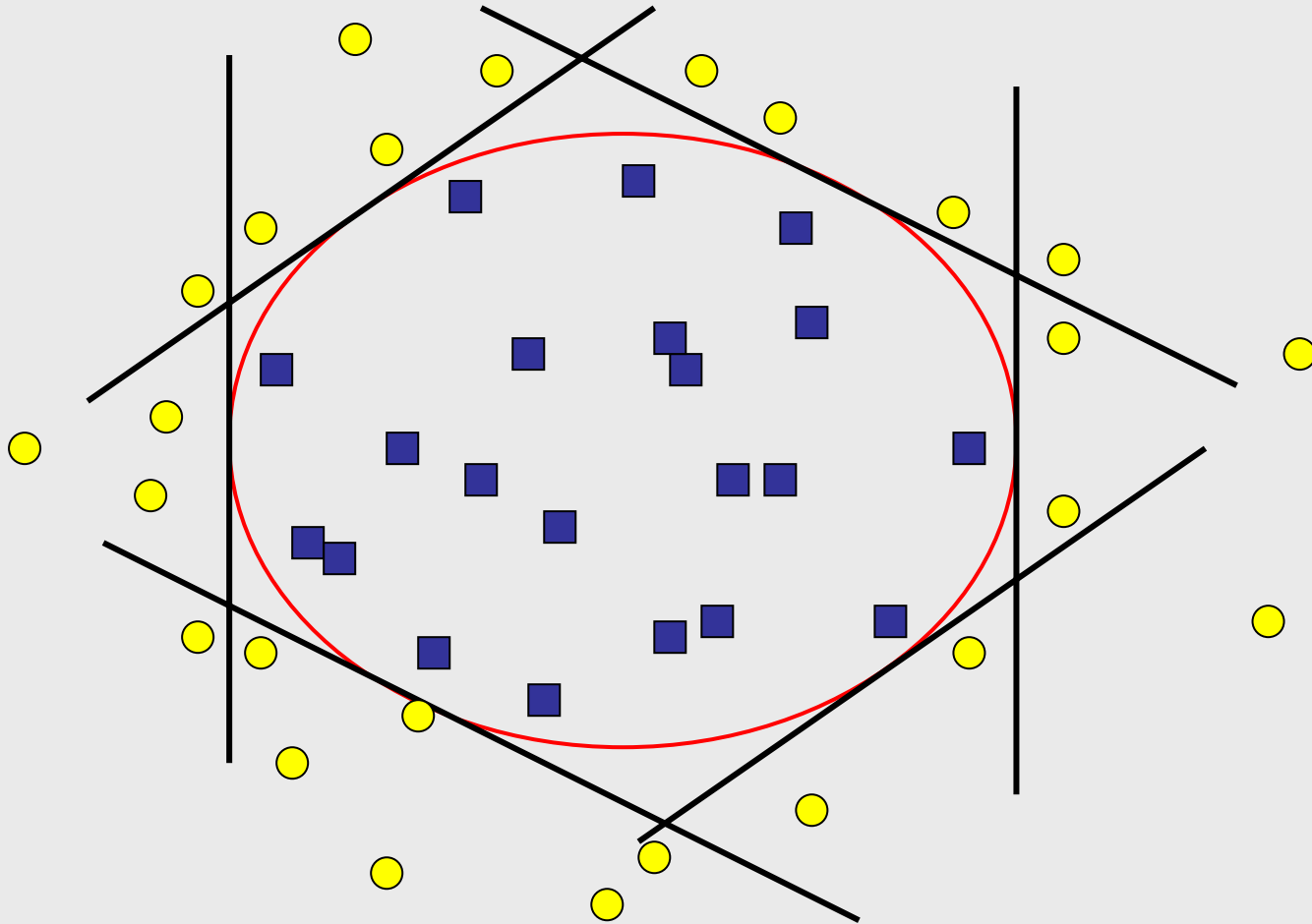
Si el polígono posee una forma muy complicada necesita más capas escondidas.

Muchas capa produce sobreajuste.

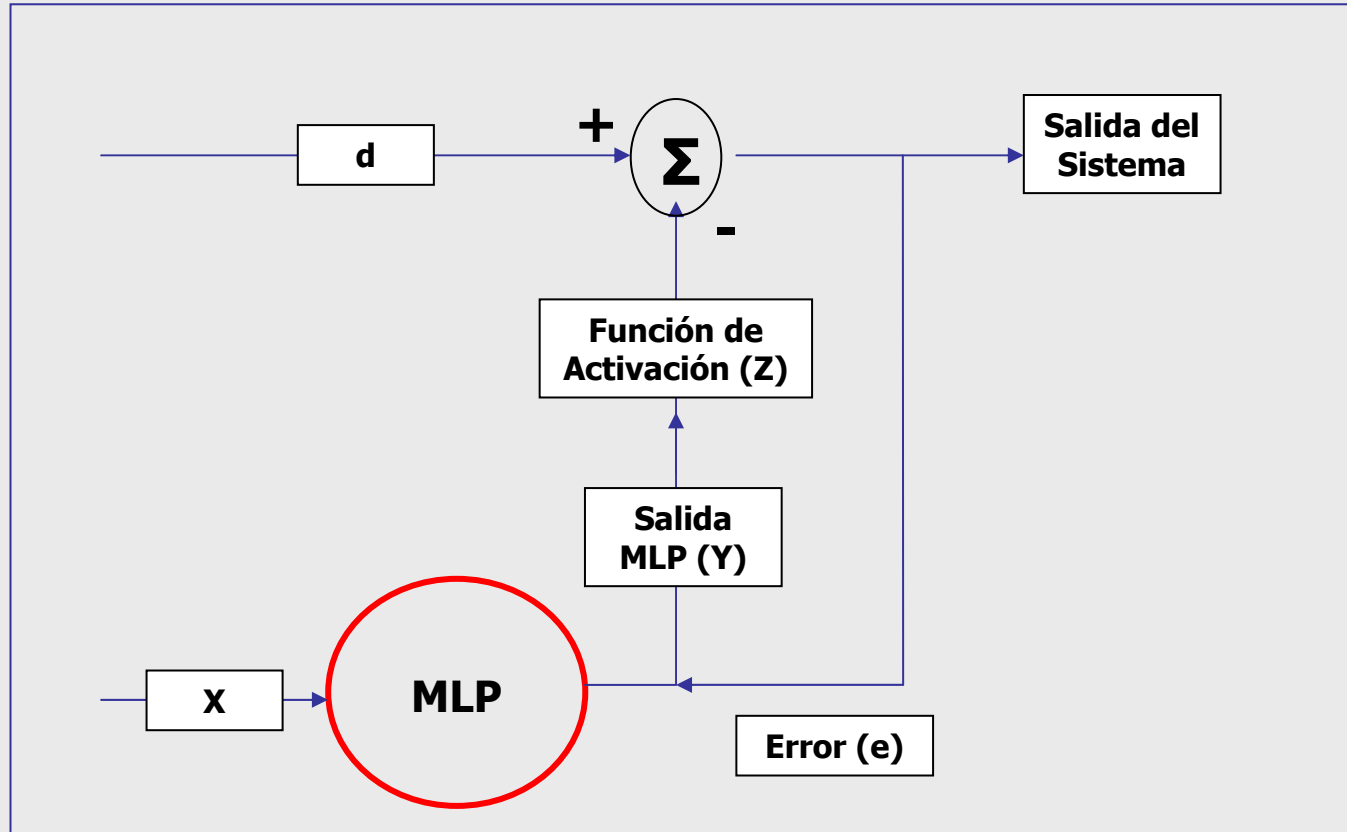
Pocas capas no puede ser modelado el problema.

“Regla Aproximada” ~ Usar el promedio de las neuronas de entrada y de salida

ROL DE LAS CAPAS OCULTAS: FORMA GRAFICA



ESQUEMA GENERAL: FUNCIONES DE ACTIVACIÓN



FUNCIONES SIGMOIDES

→ Logística (0,1)

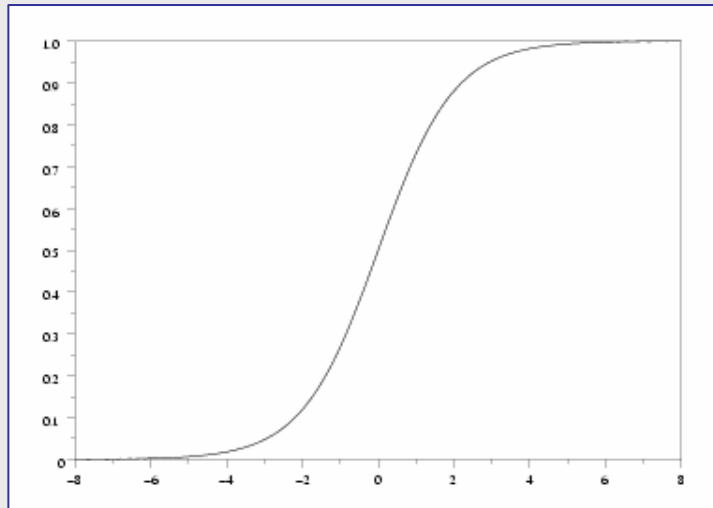
$$\text{sigm}(x) = \frac{1}{1 + e^{-x}}$$

→ Tangente hiperbólica (-1,1)

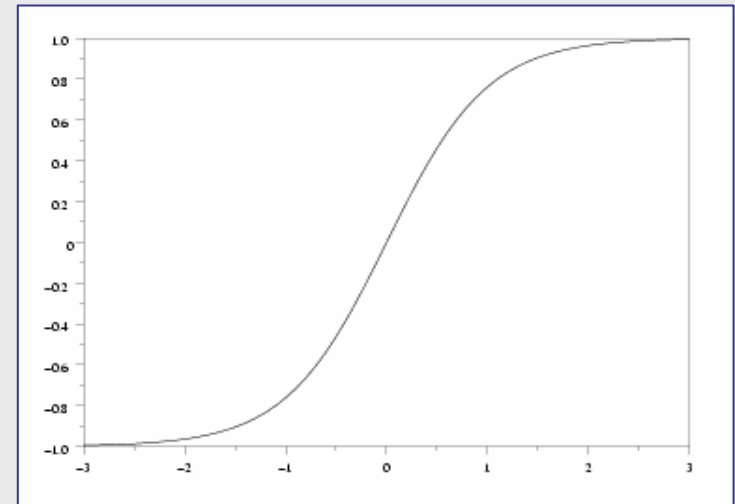
$$\text{sigm}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

FUNCIONES DE ACTIVACIÓN: FORMAS GRÁFICAS

Logística



Tangente Hiperbólica



RETROPORPAGACIÓN DEL ERROR

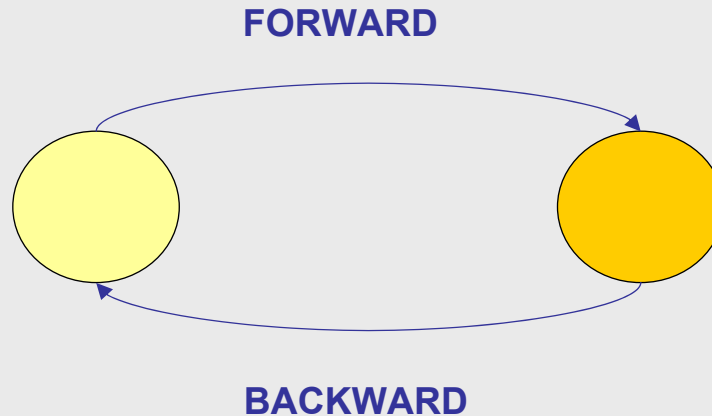
- Se explora el grado de influencia de los pesos en el error
- Estos se ajustan desde la capa de salida hacia la capa de entrada
- Necesita que las funciones de activación sean diferenciables

$$\text{Error} \rightarrow \boxed{\varepsilon_k = (d_k - z_k)}$$

\uparrow
Salida Deseada
 \uparrow
Salida Calculada

RETROPORPAGACIÓN EN DOS PASADAS

- **FORWARD:** Con los pesos fijos se calcula la respuesta de las distintas unidades (capa oculta y de salida) y se determina el error.
- **BACKWARD:** La señal del error es propagada hacia atrás usando los pesos de la red y ajustando los pesos.



TASA DE APRENDIZAJE (μ)

- Es la encargada de la velocidad en que son modificados los pesos en cada una de las iteraciones del algoritmo
- Toma valores entre 0 y 1
- Valor alto
 - Rápida minimización del error
 - Soluciones poco precisas e inestables
- Valor bajo
 - Mayor precisión en la búsqueda de la minimización del error
 - Mayor cantidad de épocas para ajustar el modelo
 - Sobreajuste por alto número de iteraciones del algoritmo

$$\Delta w(k) = \mu \cdot f'(e) \cdot x(k)$$

Tasa de aprendizaje

MOMENTUM

- Trata de aumentar la tasa de aprendizaje sin producir inestabilidad
- Trata de aumentar la velocidad de convergencia
- Ayuda en los “baches” provocado por los mínimos locales
- Amortigua las oscilaciones del error e durante el aprendizaje

Momentum

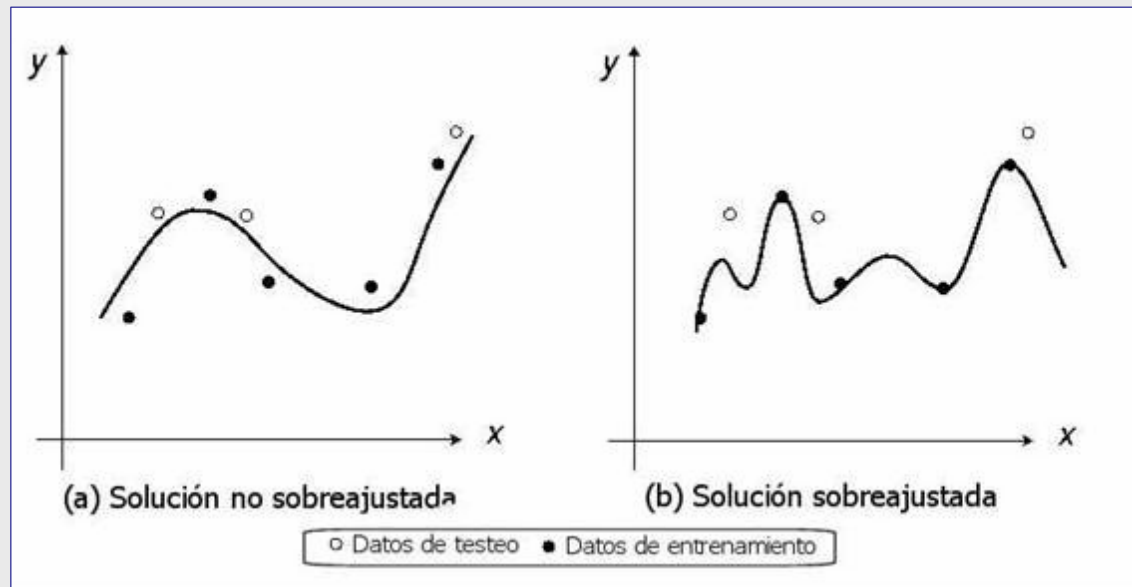
$$\Delta w(k) = \mu \cdot f'(e) \cdot x(k) + \alpha \cdot \Delta w(k-1)$$

SOBREAJUSTE DE LA RED

Ocurre cuando la red se aprende los patrones “de memoria”

Produce una mala generalización del modelo a nuevos ejemplos

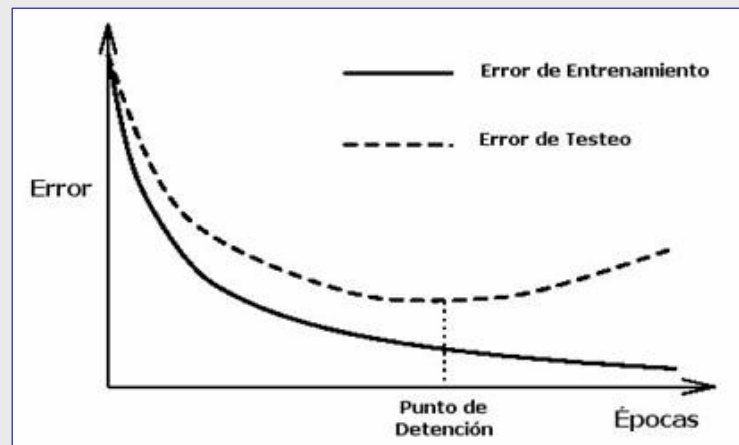
Se ajusta sólo a los datos de entrenamiento



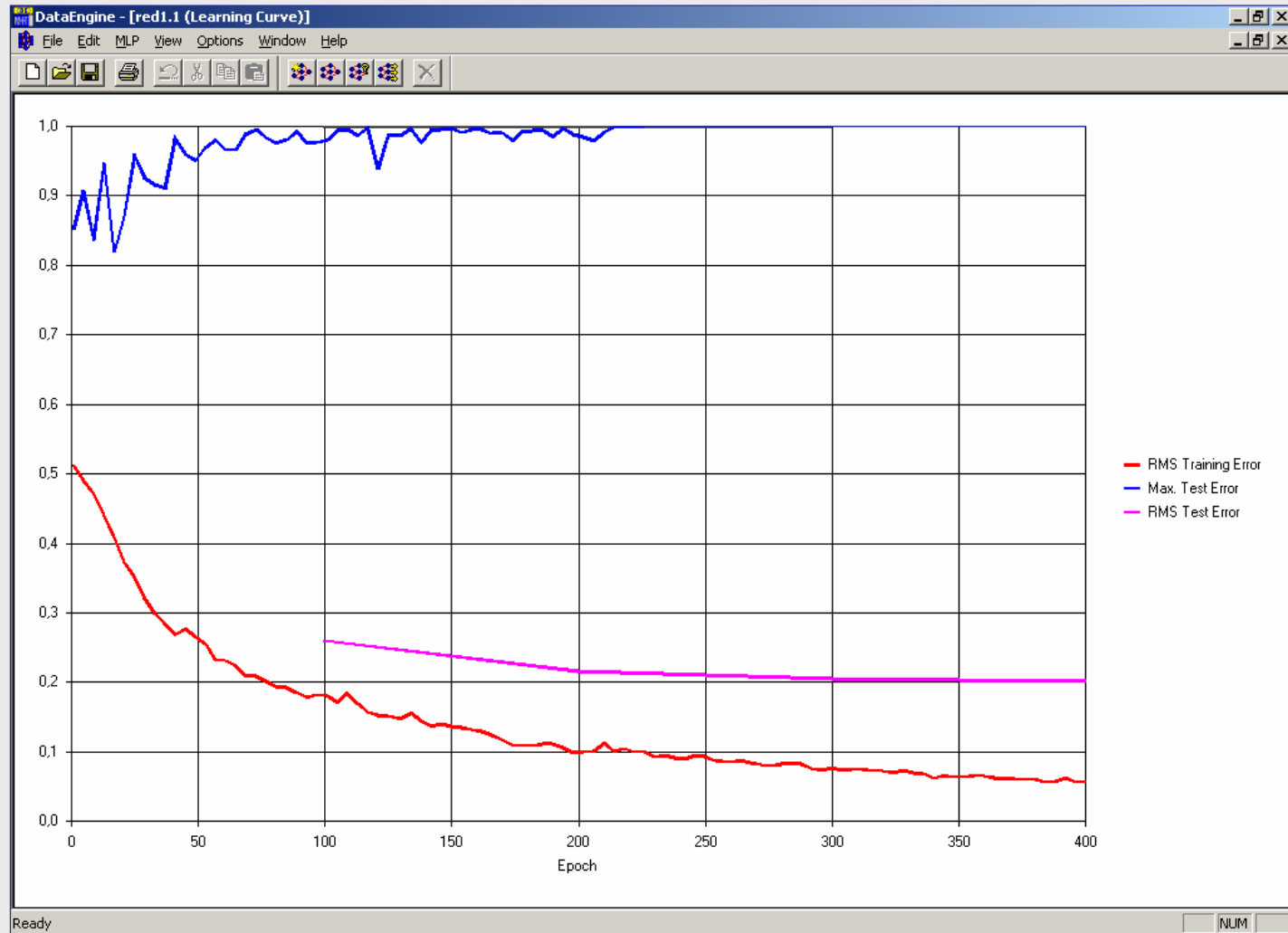
SOLUCIÓN: DETENCIÓN TEMPRANA

Busca encontrar el punto de equilibrio entre el número de épocas usado para el entrenamiento y el sobreajuste de la red

Lo mide a través de una comparación entre el error en el conjunto de entrenamiento y el conjunto de test

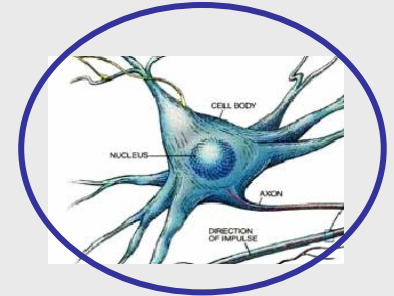


SOBREAJUSTE



Tiene una alta aplicabilidad

- Data Mining supervisado y no supervisado
- Variadas áreas de estudio
 - Industrial
 - Geología
 - Biotecnología y genética



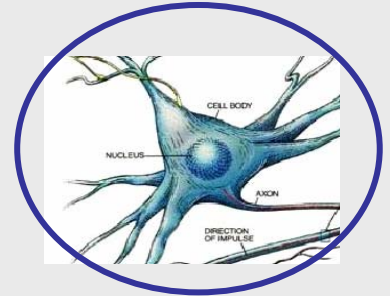
Aplicabilidad a soluciones complejas

- Modelación no lineal
- Gran adaptabilidad a los inputs de la red

Trabaja tanto con variables categóricas (transformadas) como continuas

Los datos deben ser preprocesados

- Uso de normalizaciones a intervalos definidos
- EJ: Normalización intervalo $[0,1]$



Los resultados son valores continuos

- Es difícil introducir variables categóricas a pesar de ser transformadas

Poseen una enorme cantidad de parámetros

- Hay enormes cantidades de combinaciones
- Problema combinatorial respecto al número de combinaciones posibles entre los valores de los parámetros de su arquitectura
- Diferencias **pequeñas** en la información de entrada pueden causar **enormes** cambios en la salida



MODELOS SUPERVISADOS

REDES NEURONALES MLP

JAIME MIRANDA

(jmiranda@di.uchile.cl)

Departamento de Ingeniería Industrial

Universidad de Chile