ADALINE Y PERCEPTRON

1. Adaline
    - 1.1. Características.
    - 1.2. Regla de Aprendizaje (LMS). Regla Delta.
    - 1.3. Aplicaciones.
        - 1.3.1. Clasificación de vectores.
        - 1.3.2. Tratamiento de Señales.
    - 1.4. Madaline.

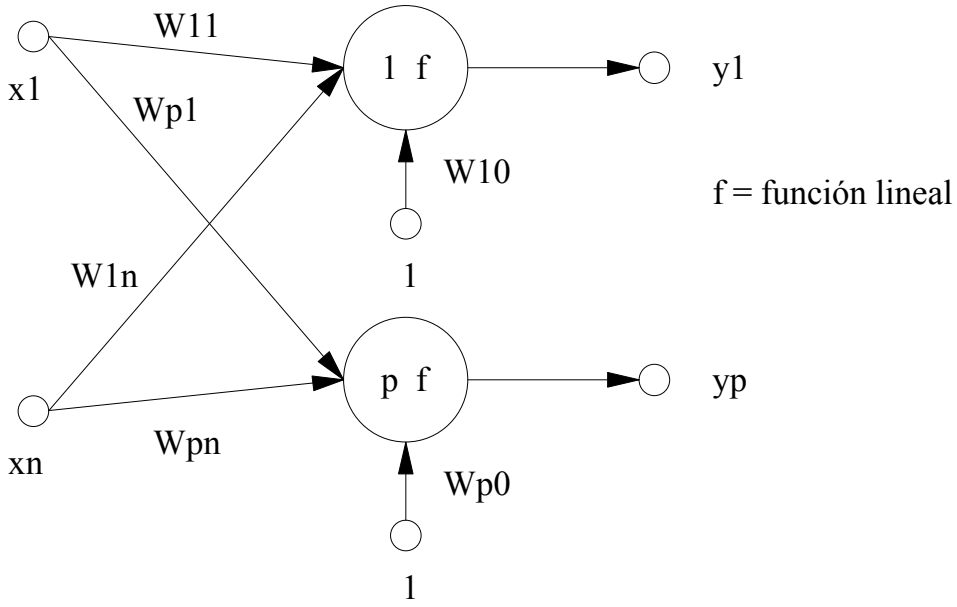
2. Perceptron
    - 2.1. Características.
    - 2.2. Regla de Aprendizaje.
    - 2.3. Problema XOR.
    - 2.4. Diferencias entre la convergencia entre el Adaline y el Perceptron.

ADALINE

(Freeman capt. 2)

Características

- Es un tipo de aprendizaje OFF Line.
- Se enmarca dentro del tipo de aprendizaje por corrección de error.
- Se utiliza para entrenar un Elemento Simple de Procesado, con una función de transferencia lineal.
- Se le conoce también con el nombre de Regla de ***Widrow-Hoff*****.** (Adaline: Adaptive Linear Elemento).
- Se pueden combinar un cierto nº de PEs en la capa de salida (estructura con un cierto grado de complejidad). La regla se aplica sobre cada uno de los PE de manera individual.



- Es aplicada a estructuras Lineales: $$y_j = w_{j0} + \sum_{i=1}^{n} w_{ji} * x_i$$

- **Idea:** Modificación de Pesos para tratar de reducir la diferencia entre la salida deseada y la actual (para cada patrón).
- Se denomina LMS: minimiza el ***Error Cuadrático Medio*** sobre todos los patrones de entrenamiento.

Cálculo de Pesos Óptimos

Sea el conjunto de entrenamiento: (X,D): patrones de entrada y salidas deseadas.
X ---- conjunto de L vectores de dimensión n.
D ---- conjunto de L vectores de dimensión m (en este caso m=1). Salida Deseada.
Y ---- conjunto de L vectores de dimensión m (en este caso m=1). Salida Obtenida

Se trata de minimizar: Sea Yk la salida obtenida para el patrón k.

El error para el patrón k $E_k = d_k - y_k$ entonces :

El Error Cuadrático Medio: $< E_k^2 >= \frac{1}{L}\sum_{k=1}^{L} E_k^2 = < (d_k - y_k)^2 >$

$$y_k = \sum_{j=0}^{n} w_j * x_j$$

$$< E_k^2 >= \frac{1}{L}\sum_{k=1}^{L} E_k^2 = < (d_k - y_k)^2 >=$$

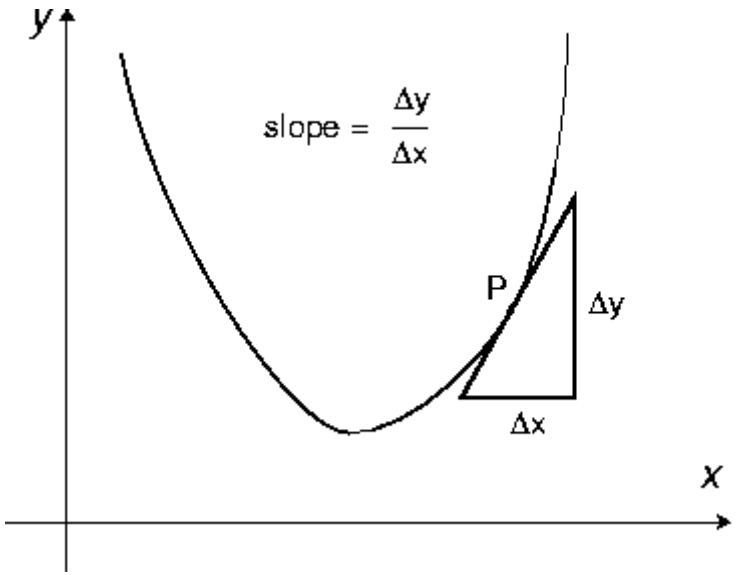
$$= < (d_k - w^t x_k)^2 >= < d_k^2 > + w^t < x_k x_k^t > w - 2 < d_k x_k^t > w$$

$$R = < x_k x_k^t > \quad P = < d_k x_k > \quad \xi = < E_k^2 >$$

Sea:

$$\xi = < d_k^2 > + w^t R w - 2 P^t w$$

Para hallar el mínimo derivamos con respecto a W y se hace igual a 0:



$$\frac{\nabla \xi(w)}{\nabla w} = 2Rw - 2P$$

$$2Rw^* - 2P = 0$$

$$w^* = R^{-1} P$$

Consecuencias:

- Posee un extremo.
- El extremo es mínimo

Ejercicio: Probar que el valor mínimo del error cuadrático medio se puede escribir de la manera siguiente:

$$\xi_{mínimo} = < d_k^2 > - P^t w^*$$

Demostración :

$$\xi = < d_k^2 > + w^t R w - 2 P^t w$$

$$\xi_{mínimo} \Rightarrow w^* = R^{-1} P \Rightarrow w^* R = P$$

$$w^{*t} P = (P^t w^*)^t = P^t w^*$$

$$\xi_{mínimo} = < d_k^2 > + w^{*t} R w^* - 2 P^t w^*$$

implicando:

$$\xi_{mínimo} = < d_k^2 > - P^t w^*$$

Cálculo de W*: Método de Gradiente Descendente.

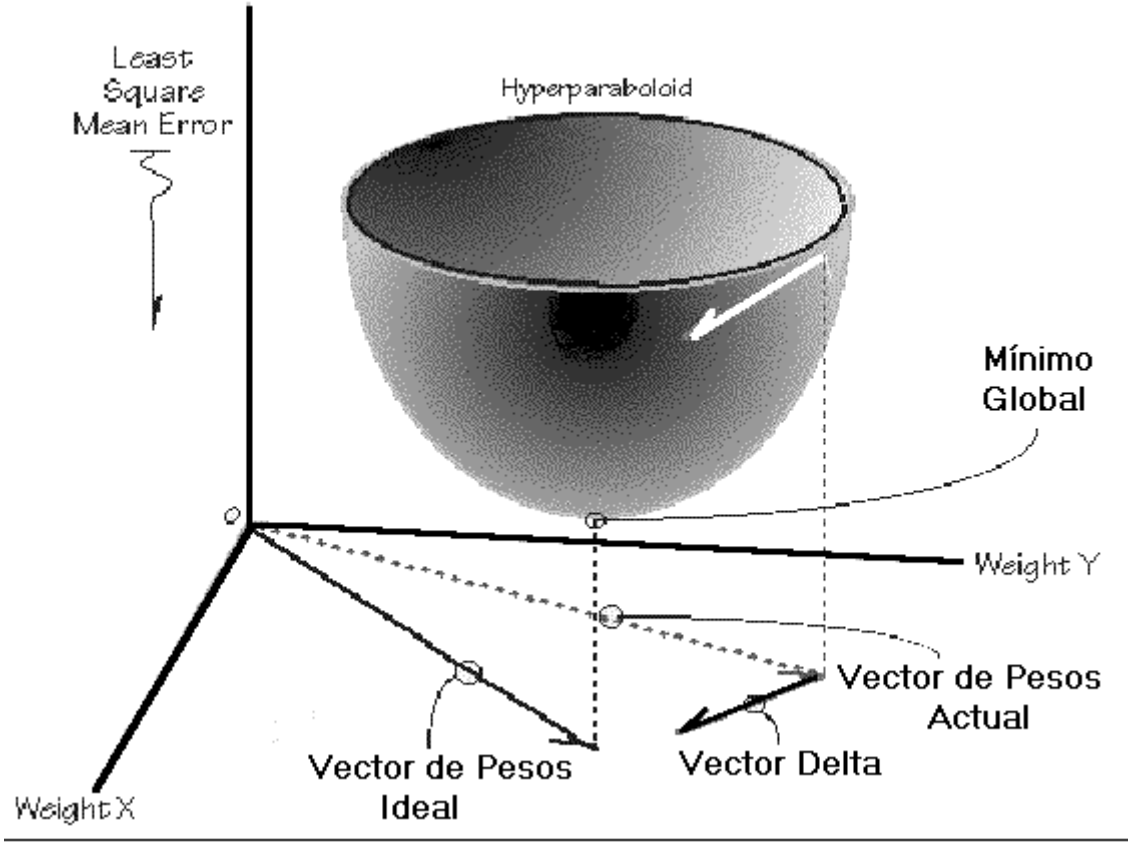
Diferentes Métodos:

- Buscar por todo el espacio de pesos hasta encontrar los que hiciesen el error mínimo.
- Realizar una búsqueda aleatoria.
- Realizar una búsqueda ***Dirigida***.

Método:

- Se inicializan los pesos aleatoriamente (pto. de partida).
- Se determina, la dirección de la pendiente más pronunciada en dirección hacia abajo.
- Se modifican los pesos para encontrarnos un poco más abajo en la superficie.

Sea W(t) el vector de pesos en el instante t. En t+1: $W(t+1)=W(t) + \Delta W(t)$



Dirección opuesta y más pronunciada ⇒ Dirección opuesta al gradiente de la Superficie $\nabla\xi(W(t))$

Para obtener la magnitud del cambio ⇒ $\mu\nabla\xi(W(t))$ Se realiza la siguiente aproximación:

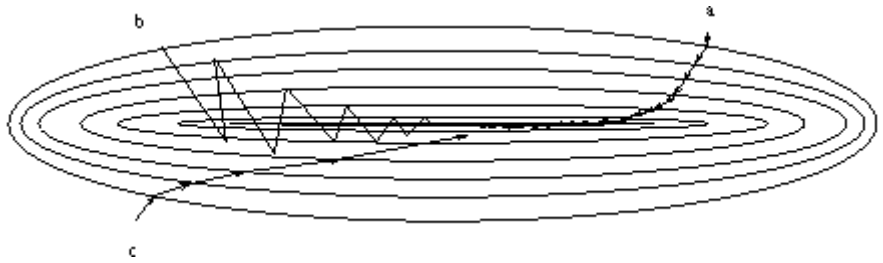
$$\xi_k(t) \approx < E_k^2 > \quad \Rightarrow \quad \nabla \xi_k = \nabla < E_k^2 > \approx \nabla E_k^2$$

Entonces como $E_k^2(t) = \left(d_k - w^t(t) x_k\right)^2$

Tenemos $\nabla \xi_k^2(t) = -2 E_k(t) x_k$

Obteniendo

$$w(t+1) = w(t) + 2\mu E_k x_k$$



a) μ pequeña.
b) μ alta.

Algoritmo:

1- Inicialización de pesos.

2- Se aplica un patrón de entrada (entradas y salida deseada).

3- Se computa la salida lineal que se optiene de la red.

4- Se calcula el error cometido para dicho patrón.

5- Se actualizan las conexiones mediante la ecuación obtenida anteriormente.

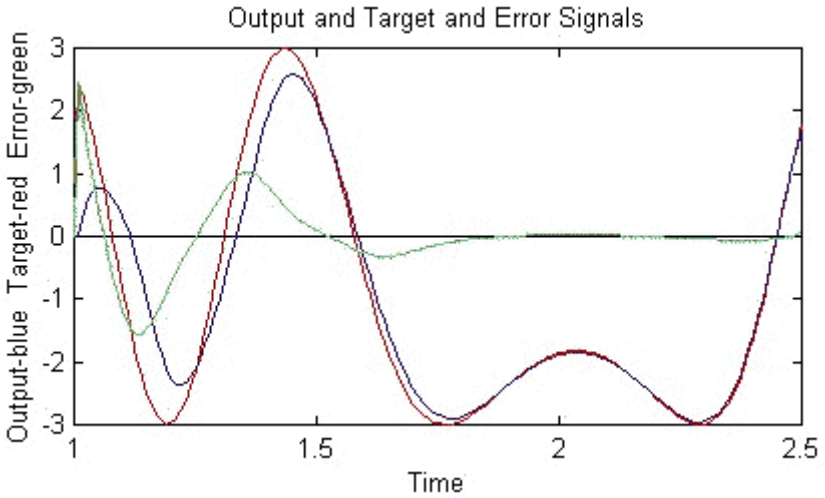
6- Se repiten los pasos del 2 al 5 para todos los patrones de entrenamiento.

7- Si el error cuadrático medio es un valor reducido aceptable, termina el proceso. Sino se vuelve al paso 2.

Aplicaciones:

La principal aplicación de las redes tipo Adaline se encuentra en el campo de procesamiento de señales. Concretamente en el diseño de filtros capaces de eliminar ruido en señales portadoras de información.

Otra aplicación es la de los ***filtros adaptativos***: Predecir el valor futuro de una señal a partir de su valor actual.



CONCLUSIONES

- Una simple capa de PE lineales pueden realizar aproximaciones a funciones lineales o asociación de patrones.

- Una simple capa de PE lineales puede ser entrenada con algoritmo LMS.

- Relaciones No Lineales entre entradas y salidas no pueden ser representadas exactamente por redes lineales. Dichas redes harán aproximaciones lineales. Otro tipo de redes abordarán la resolución de problemas no lineales.

REGLA DEL PERCEPTRÓN (Rosenblatt)

(Neural Computing, capt. 3)

- Supongamos ***un PE con una función de transferencia del tipo Hardlimiter*** y en donde las entradas son binarias o bipolares (mismo que Adaline pero con esas restricciones).

La regla que rige el cambio de pesos es:

$W_i(t+1) = W_i(t)$ Si la salida es correcta.

$W_i(t+1) = W_i(t) + X_i(t)$ Si la salida = -1 y debería de ser 1.

$W_i(t+1) = W_i(t) - X_i(t)$ Si la salida = 1 y debería de ser -1.

Sobre la regla anterior se han realizado diferentes modificaciones:

A)

$W_i(t+1) = W_i(t)$ Si la salida es correcta.

$W_i(t+1) = W_i(t) + \mu X_i(t)$ Si la salida = -1 y debería de ser 1.

$W_i(t+1) = W_i(t) - \mu X_i(t)$ Si la salida = 1 y debería de ser -1.

Con $\mu \in$ **[0,1],** término de control de ganancia y velocidad de aprendizaje.

B) Otra de las modificaciones propuestas fue sugerida por Widrow and Hoff. Ellos propusieron una regla basada en la regla Delta. (Es la más utilizada).

Los Pesos cambian según la formula: $$w(t+1) = w(t) + 2\mu E_k x_k$$

Tomando las entradas y salidas como bipolares tenemos que el cambio en los pesos se produce de la manera siguiente:

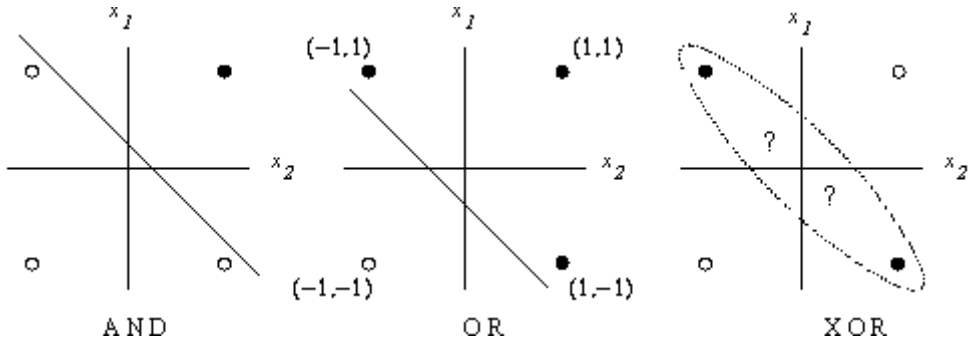
$$w_i(t+1) = w_i(t) + \mu(t)\, z_i \quad si \quad z_i\, w_i(t) \le 0$$

$$w_i(t+1) = w_i(t) \quad en \quad otro \quad caso$$

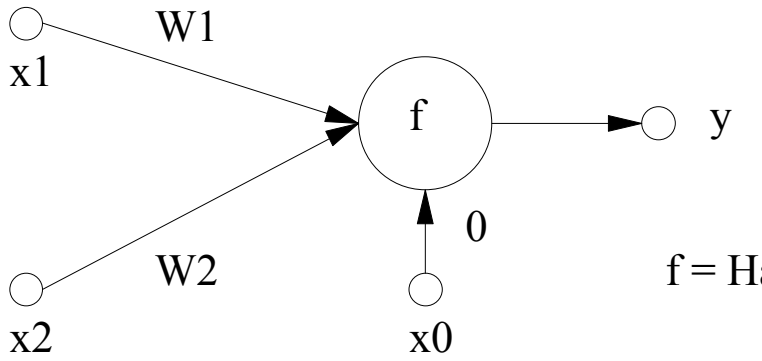
$$z_i = \begin{cases} +x_i & si \quad d_i = +1 \\ -x_i & si \quad d_i = -1 \end{cases}$$

Con esta regla de aprendizaje se obtiene una convergencia finita si el conjunto de entrenamiento es ***linealmente separable***.

EJEMPLO: XOR

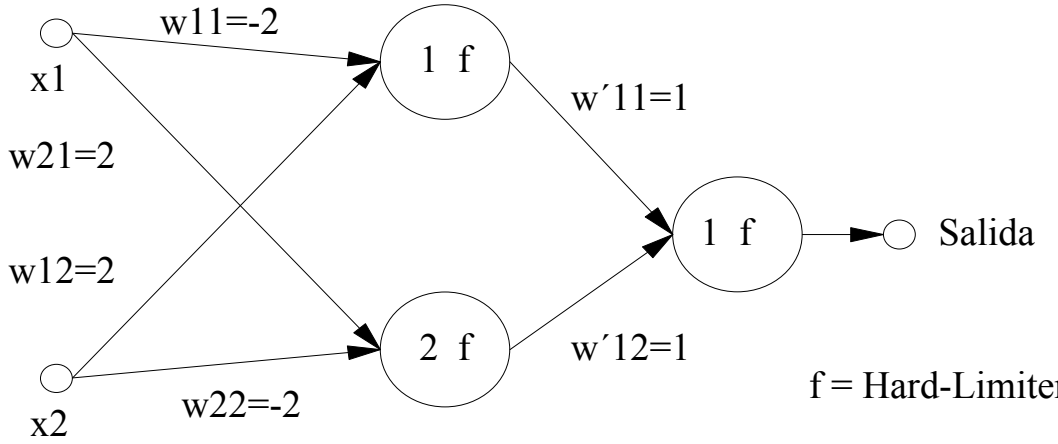


Solución con 2 Capas:



f = Hard-Limiter

$0 * W1 + 0 * W2 = 0$ W1,W2
$1 * W1 + 0 * W2 = 1$ W1 = 1
$0 * W1 + 1 * W2 = 1$ W2 = 1
$1 * W1 + 1 * W2 = 1$ No posible.



f = Hard-Limiter

PE sin Tendencia

Solución Tres Capas