uc3m | Universidad Carlos III de Madrid

OPENCOURSEWARE
REDES DE NEURONAS ARTIFICIALES
Inés M. Galván – José M. Valls

Tema 2

Primeros Modelos Computacionales

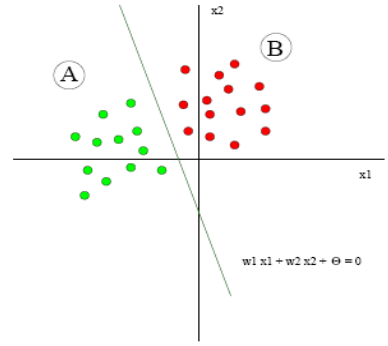
Primeros Modelos Computacionales

- Perceptron simple
- Adaline
- Clasificación y Regresión lineal
- Problemas no linealmente separables

Perceptron simple

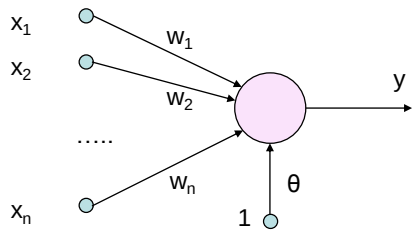
- Modelo propuesto por Rosenblatt en 1959
- Adaptación supervisada
- Sistema capaz de realizar tareas de clasificación de forma automática

A partir de un número de ejemplos etiquetados, el sistema determina la ecuación del hiperplano discriminante



Perceptron simple. Arquitectura

- Red monocapa con varias neuronas de entrada conectadas a la neurona de salida



$$y = \begin{cases} 1, \text{ si } w_1x_1+\ldots+w_nx_n+\theta > 0 \\ -1, \text{ si } w_1x_1+\ldots+w_nx_n+\theta \leq 0 \end{cases}$$

Perceptron simple. Arquitectura

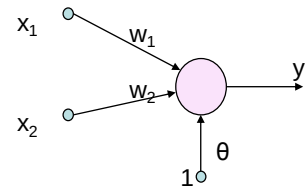
- El perceptron equivale a un hiperplano de dimensión n-1 capaz de separar las clases
  - Si la salida del perceptron es +1, la entrada pertenecerá a una clase (estará situada a un lado del hiperplano)
  - Si la salida es -1, la entrada pertenecerá a la clase contraria (estará situada al otro lado del hiperplano)
- La ecuación del hiperplano es:

$$w_1 x_1 + w_2 x_2 + ... + w_n x_n + \theta = 0$$

Perceptron simple. Arquitectura

Dimensión 2

$$y = \begin{cases} 1, \text{ si } & w_1 x_1 + w_2 x_2 + \theta > 0 \\ -1, \text{ si } & w_1 x_1 + w_2 x_2 + \theta \leq 0 \end{cases}$$



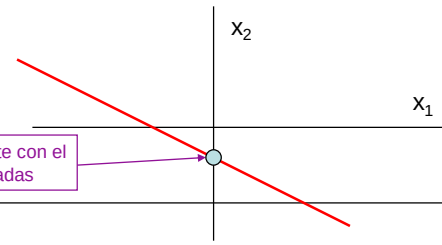
La ecuación del hiperplano es: $w_1 x_1 + w_2 x_2 + \theta = 0$

En dos dimensiones el hiperplano es una recta

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{\theta}{w_2}$$

Pendiente de la recta

Punto de corte con el eje de ordenadas



Perceptron simple. Aprendizaje

- Se dispone de un conjunto de observaciones (patrones, ejemplos, datos) de los que se sabe su categoría o clase
- Los ejemplos o datos son puntos en un espacio multidimensional

$$\Re^n : (x_1, x_2, ...x_n)$$

- Hay que determinar la ecuación del hiperplano que deja a un lado los ejemplos de una clase y a otro lado los de la otra clase
- La ecuación del hiperplano se deduce a partir de los ejemplos o datos

$$w_1x_1 + w_2x_2 + ... + w_nx_n + \theta = 0$$

Perceptron simple. Aprendizaje

- Proceso iterativo supervisado
- Modificación de los pesos y umbral hasta encontrar el hiperplano discriminante
- Número finito de iteraciones

Dado	Encontrar
Conjunto de patrones Vector de entrada: x=(x_1, x_2 , ..., x_n) Salida deseada: d(x) d(x)= 1 si x $\in$ A d(x)=-1 si x $\in$ B	Hiperplano discriminante (w_1, w_2 , ..., w_n, θ) tales que $x_1 \cdot w_1 + x_2 \cdot w_2 + ... + x_n \cdot w_n + \theta = 0$ separe las clases A y B

Perceptron simple. Aprendizaje

Paso 1: Inicialización aleatoria de los pesos y el umbral de la red $\{w_i(0)\}_{i=0,...,n}$ $\theta(0)\}$

Paso 2: Se toma un patrón o ejemplo entrada-salida $[x=(x_1,x_2, ..., x_n), d(x)]$

Paso 3: Se calcula la salida de la red: $y= f(x_1 \cdot w_1 + x_2 \cdot w_2 + ... + x_n \cdot w_n + \theta)$

Paso 4: Si $y = d(x)$ (clasificación correcta)

Si $y \neq d(x)$ (clasificación incorrecta) se modifican los pesos y el umbral:

$$w_i(t+1) = w_i(t) + d(x) \cdot x_i \quad \theta(t+1) = \theta(t) + d(x)$$

Ley de aprendizaje

Si $x \in A$, $d(x) = 1 \Rightarrow w_i(t+1) = w_i(t) + x_i \quad \theta(t+1) = \theta(t) + 1$

Si $x \in B$, $d(x) = -1 \Rightarrow w_i(t+1) = w_i(t) - x_i \quad \theta(t+1) = \theta(t) - 1$

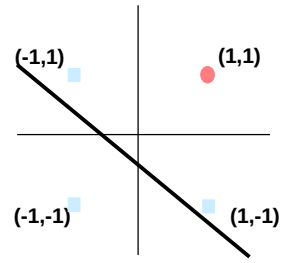
Paso 5: Se vuelve al paso 2 hasta completar el conjunto de patrones de entrenamiento

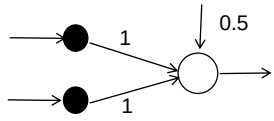
Paso 6: Se repiten los pasos 2, 3, 4 y 5 hasta alcanzar el criterio de parada

Perceptron simple. Ejemplo

Ejemplo: Función lógica AND

x_1	x_2	AND
-1	-1	-1
1	-1	-1
-1	1	-1
1	1	1





Perceptron simple. Ejemplo

Ejemplo: Función lógica AND

X=(-1,-1), d(x)=-1 ⇒ Y=f(-1.5)=-1 ⇒ Bien clasificado

X=(1,-1), d(x)=-1 ⇒ Y=f(0.5)=1 ⇒ Mal clasificado

Nuevos pesos

$w_1(1) = 1 - 1 = \mathbf{0}$

$w_2(1) = 1 - (-1) = \mathbf{2}$

$\theta(1) = 0.5 - 1 = \mathbf{-0.5}$

⇒ Y=f(-2.5)=-1 ⇒ Bien clasificado

X=(-1,1), d(x)=-1 ⇒ Y=f(1.5)=1 ⇒ Mal clasificado

Nuevos pesos

$w_1(1) = 0 - (-1) = \mathbf{1}$

$w_2(1) = 2 - 1 = \mathbf{1}$

$\theta(1) = -0.5 - 1 = \mathbf{-1.5}$

⇒ Y=f(-1.5)=-1 ⇒ Bien clasificado

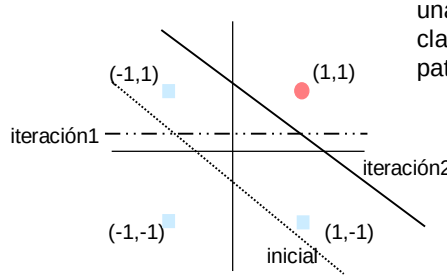
X=(1,1), d(x)=1 ⇒ Y=1 ⇒ Bien clasificado

Un hiperplano solución es: $\mathbf{x_1 + x_2 - 1.5 = 0}$

Perceptron simple. Ejemplo

Ejemplo: Función lógica AND

El hiperplano se mueve de una iteración a otra para clasificar correctamente los patrones



Perceptron simple. Proceso de Entrenamiento

Dado el conjunto de entrenamiento y el conjunto de validación (o test):

1. Se presentan los patrones de entrenamiento, modificando los pesos del perceptron para cada patrón si fuera necesario (transparencia 9) (un ciclo de aprendizaje).

2. Se evalúa el porcentaje de éxito del perceptron en el conjunto de entrenamiento con los últimos pesos obtenidos en el paso 1.

3. Se presentan los patrones de validación (o test) al perceptron y se calculan las salidas con los últimos pesos obtenidos en el paso 1. En este caso los pesos no se modifican, solo se calcula la salida el perceptron.

4. Se evalúa el porcentaje de éxito del perceptron en el conjunto de validación (o test).

5. Se repiten los pasos 1, 2, 3 y 4 hasta el criterio de parada

Siguiendo ese proceso se puede observar el comportamiento del perceptron en entrenamiento y validación (o test) a lo largo de los ciclos de aprendizaje:

• El porcentaje de éxito en entrenamiento tiende a aumentar.

• El porcentaje de éxito en validación (o test) también tiende a aumentar, pero podría ir disminuyendo a lo largo de los ciclos (sobre-aprendizaje, situación no deseada)

Perceptron simple. Parada Aprendizaje

Criterio de Parada: Perceptron Simple

Criterio 1: Fijar un número de ciclos máximo. Generalmente se elige el PS que proporciona mayor porcentaje de éxito en la clasificación de los patrones de entrenamiento. No necesariamente es el PS obtenido en el último ciclo

Criterio 2: Cuando el porcentaje de éxito en la clasificación de los patrones de entrenamiento no cambia durante x ciclos

El comportamiento de la red sobre el conjunto de test se puede visualizar a lo largo de los ciclos de aprendizaje, pero nunca se utilizará para decidir cuando parar el aprendizaje.

Criterio 3: Sí se puede utilizar un conjunto de validación, que correspondería con una porción aleatoria del conjunto de entrenamiento. En este caso, el criterio sería: cuando el mejor porcentaje de éxito sobre los patrones de validación no aumenta o se mantiene estable a lo largo de x ciclos.

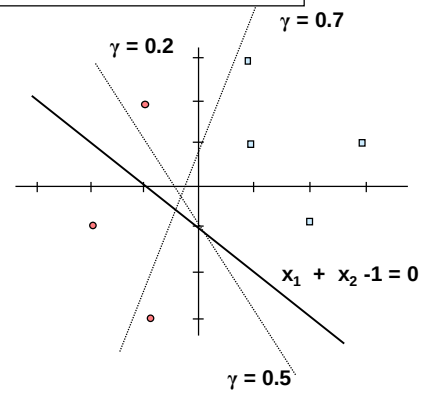
Perceptron simple

Ley de aprendizaje con **razón o tasa de aprendizaje**

$$w_i(t+1) = w_i(t) + \gamma \cdot d(x) \cdot x_i \quad \theta(t+1) = \theta(t) + \gamma \cdot d(x)$$

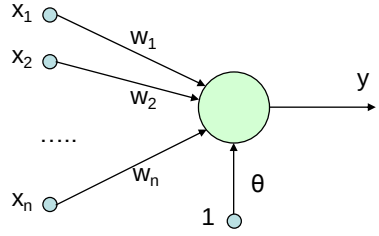
donde γ es un número real
$0 < \gamma < 1$

Controla el cambio que sufren los pesos de una iteración a otra



ADALINE:ADAptive LInear NEuron

- Desarrollado en 1960 por Widrow y Hoff
- Estructura prácticamente idéntica al perceptron, salida número real
- Elemento combinador adaptativo lineal, que recibe todas las entradas, las suma ponderadamente, y produce una salida



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

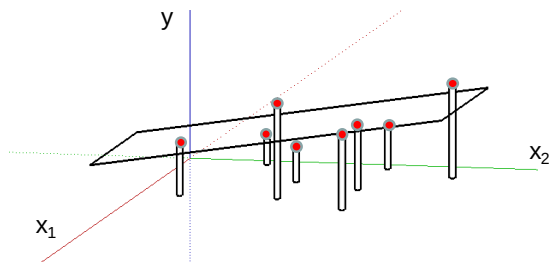
ADALINE

- La diferencia con el perceptron es la manera de utilizar la salida en la regla de aprendizaje
- El perceptron utiliza la salida de la funcion umbral (binaria) para el aprendizaje. Sólo se tiene en cuenta si se ha equivocado o no.
- En Adaline se utiliza directamente la salida de la red (real) teniendo en cuenta cuánto se ha equivocado.
- Se utiliza la diferencia entre el valor real esperado y la salida producida de la red.

  Para un patrón de entrada x^p, se tendrá en cuenta el error producido (d^p-y^p), siendo d^p la salida deseada e y^p la salida del ADALINE
- El objetivo es obtener una red tal que $y^p \approx d^p$ para todos los patrones *p*

ADALINE

- Será imposible conseguir una salida exacta porque ***y*** es una función lineal, pero se minimizará el error cometido para todos los patrones de entrenamiento
- Hay que elegir una medida de dicho error, p.ej.el error cuadrático
- La regla de aprendizaje es la **REGLA DELTA**



ADALINE. Regla Delta

- Los patrones de entrenamiento están formados por pares de valores (x^p, d^p). x^p vector de entrada; d^p su salida deseada (número real)
- EL aprendizaje se lleva a cabo utilizando la diferencia entre la salida producida para cada patrón (p) y la deseada

$$(d^p - y^p)$$

- Se utiliza una función de error para todo el conjunto de patrones. Generalmente el error cuadrático medio para los patrones de entrenamiento

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

Error global — Error cuadrático por patrón

ADALINE. Regla Delta

- La regla Delta busca el conjunto de pesos que minimiza la función de error
- Se hará mediante un proceso iterativo donde se van presentando los patrones uno a uno y se van modificando los pesos y el umbral de la red mediante la regla del descenso del gradiente

$$w_j(t+1) = w_j(t) + \Delta_p w_j$$

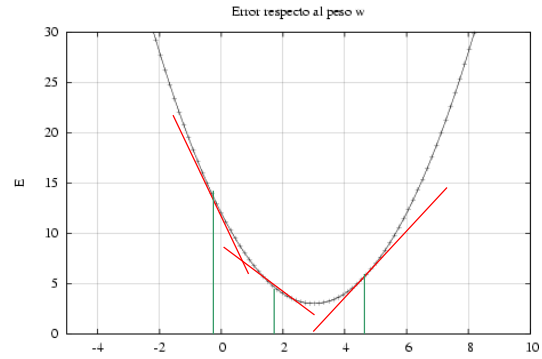
- La idea es realizar un cambio en cada peso proporcional a la derivada del error, medida en el patrón actual, respecto del peso:

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

siendo γ la tasa o razón de aprendizaje

ADALINE. Regla Delta

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



ADALINE. Regla Delta

- Aplicando la regla de la cadena queda:

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

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

$$y^p = w_1 x_1 + \ldots + w_j x_j + \ldots w_n x_n + \theta$$

$$2 \times \frac{1}{2}(d^p - y^p) \times (-1) = -(d^p - y^p)$$

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

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

ADALINE. Proceso de Aprendizaje

Paso 1: Inicializar los pesos y umbral de forma aleatoria

Paso 2: Presentar un patrón de entrada

Paso 3: Calcular la salida, compararla con la deseada y obtener la diferencia: (d^p - y^p)

Paso 4: Para todos los pesos y para el umbral, calcular:

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

Paso 5: Modificar los pesos y el umbral del siguiente modo:

$$w_j(t+1) = w_j(t) + \Delta_p w_j \qquad \theta(t+1) = \theta(t) + \Delta_p \theta$$

Paso 6: Repetir los pasos 2, 3, 4 y 5 para todos los patrones de entrenamiento (1 ciclo)

Paso 7: Repetir los pasos 2,3,4,5 y 6 tantos ciclos hasta cumplir el criterio de parada

Adaline. Proceso de Entrenamiento

Dado el conjunto de entrenamiento y el conjunto de validación (o test):

1. Se presentan los patrones de entrenamiento, modificando los pesos del Adaline para cada patrón (transparencia 23) (un ciclo de aprendizaje).

2. Se evalúa el error cuadrático medio (o el error medio) del Adaline en el conjunto de entrenamiento con los últimos pesos obtenidos en el paso 1.

3. Se presentan los patrones de validación (o test) al Adaline calculando la salida para estos patrones con los últimos pesos obtenidos en el paso 1. En este caso los pesos no se modifican, solo se calcula la salida el Adaline.

4. Se evalúa el error cuadrático medio (o el error medio) del Adaline en el conjunto de validación (o test).

5. Se repiten los pasos 1, 2, 3 y 4 hasta el criterio de parada

Siguiendo ese proceso se puede observar el comportamiento del Adaline en entrenamiento y validación (o test) a lo largo de los ciclos de aprendizaje:

•El error de entrenamiento debe ir disminuyendo.

•El error de validación (o test) también tiende a disminuir, pero podría aumentar a lo largo de los ciclos (sobre-aprendizaje, situación no deseada)

ADALINE. Parada Aprendizaje

Criterio de Parada: Adaline

Criterio 1: Fijar un número de ciclos máximo. Dicho número debe garantizar que el error cuadrático para los patrones de entrenamiento se haya estabilizado.

Criterio 2: Cuando el error cuadrático sobre los patrones de entrenamiento no cambia durante x ciclos

El error sobre el conjunto de test puede visualizarse a lo largo de los ciclos, pero nunca se utilizará para decidir cuando parar el aprendizaje.

Criterio 3: Sí se puede utilizar un conjunto de validación, que correspondería con una porción aleatoria del conjunto de entrenamiento. En este caso, el criterio sería: cuando el error cuadrático sobre los patrones de validación no aumenta o se mantiene estable a lo largo de x ciclos.

Perceptron Vs. Adaline

- En el Perceptron la salida es binaria, en el Adaline es real
- En Adaline existe una medida de cuánto se ha equivocado la red, en Perceptron sólo de si se ha equivocado o no
- En Adaline hay una razón de aprendizaje que regula lo que va a afectar cada equivocación a la modificación de los pesos. Es siempre un valor entre 0 y 1 para ponderar el aprendizaje
- En Perceptron también se podría utilizar la razón de aprendizaje, pero no pondera el error, solo amortigua el valor de al salida deseada, produciendo cambios menos bruscos de un ciclo a otro en el hiperplano
- El Perceptron sirve para resolver problemas de clasificación lineal, mientras el Adaline para problemas de regresión lineal

Adaline. Ejemplo

Decodificador binario-decimal

Aproximar con Adaline la función que realiza la decodificación binario-decimal

x1	x2	x3	d
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

Tasa de aprendizaje 0.3

Valores iniciales para los pesos:

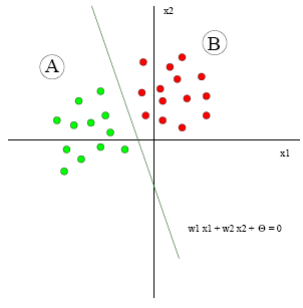
w1=0.84

w2=0.39

w3=0.78

Clasificación Lineal

•Dado un conjunto de **ejemplos o patrones**, determinar el **hiperplano** capaz de discriminar los patrones en dos clases



Dado

($x^1=(x^1_1, ..., x^1_n)$, Clase)

($x^2=(x^2_1, ..., x^2_n)$, Clase)

…

($x^m=(x^m_1, ..., x^m_n)$, Clase)

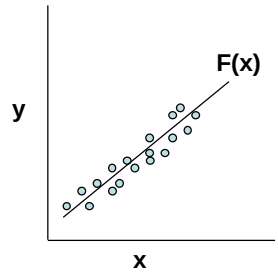
Encontrar

Hiperplano separador

Resolución con el Perceptron Simple

Regresión Lineal

•Dado un conjunto de **ejemplos o patrones**, determinar una función lineal que aproxime lo mejor posible a los valores deseados



Dado

$(x^1=(x^1_1, \ldots, x^1_n), y^1)$

$(x^2=(x^2_1, \ldots, x^2_n), y^2)$

…

$(x^m=(x^m_1, \ldots, x^m_n), y^m)$

Encontrar

Una función F(x) tal que $F(x^p) \approx y^p$

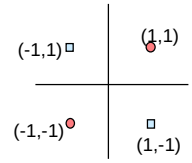
Resolución con el Adaline

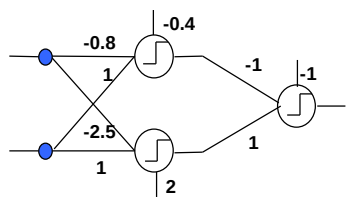
Problemas no linealmente separables

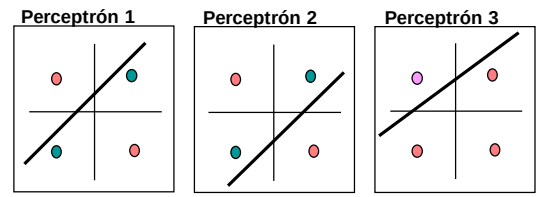
Función XOR (OR exclusivo): No existe un hiperplano.

Solución combinar varios Perceptrones

x_1	x_2	d(x)
-1	-1	1
-1	1	-1
1	-1	-1
1	1	1



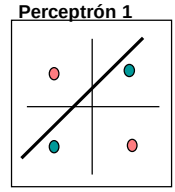




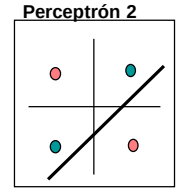
Problemas no linealmente separables

x_1	x_2	d(x)
-1	-1	1
-1	1	-1
1	-1	-1
1	1	1

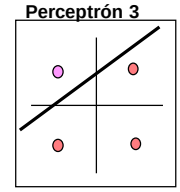
Perceptrón 1



Perceptrón 2



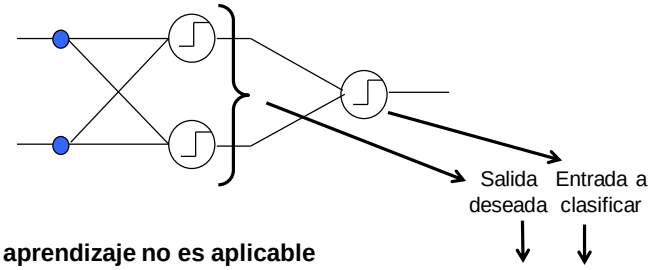
Perceptrón 3



	Perceptron1	Perpertron2	Perceptron3
(-1 -1)	-1	1	1
(-1 1)	1	1	-1
(1 -1)	-1	-1	-1
(1 1)	-1	1	1

Problemas no linealmente separables

Esta aproximación puede ser complicada de llevar a cabo en al práctica, pues la ley de aprendizaje no es aplicable y los pesos tendrían que ser determinados mediante un proceso manual



La ley de aprendizaje no es aplicable

$$w_i(t+1) = w_i(t) + d(x) \cdot x_i$$