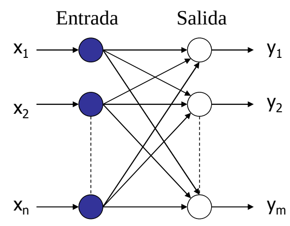
Perceptrón y Adaline

Teoría de Autómatas y Lenguajes Formales II. Curso 2008-09.
3º curso de Ingeniería Técnica en Informática de Sistemas
Escuela Técnica Superior de Ingeniería Informática
Universidad de Valladolid

Definición y funcionamiento

La misma arquitectura:

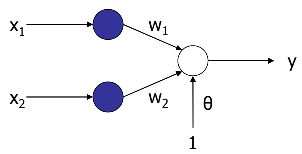


$$T : \Re^n \rightarrow \Re^m$$

- Las neuronas de entrada:
  - Propagan una componente del vector a todas las neuronas de salida
  - No realizan procesamiento alguno
- Las neuronas de salida:
  - Se ajustan al modelo de McCulloch-Pitts
  - Perceptrón simple:
    F(x)=sgn(x)
  - Adaline:
    F(x)=x

Interpretación geométrica

Separador lineal:

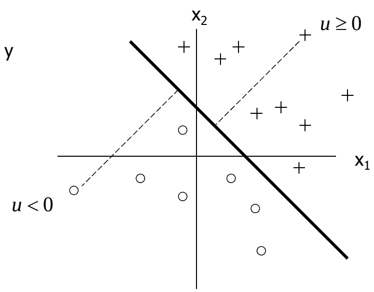


$$u = w_1 x_1 + w_2 x_2 + \Theta$$

$$F(x) = \mathrm{sgn}(x) \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

Activación:

distancia a un hiperplano



Aprendizaje del perceptrón

1. Pesos iniciales aleatorios
2. En cada iteración:
   $w(t+1) = w(t) + \Delta w(t)$
3. Para cada muestra de aprendizaje:
   - $Si\ y(x) = d(x) \Rightarrow \Delta w(t) = 0$
   - $Si\ y(x) \neq d(x) \Rightarrow \Delta w_i(t) = d(x) x_i$
     Adaptación de la Regla de Hebb:
     - Reforzar las conexiones, cuya entrada sea del mismo carácter que la salida: inhibidoras o excitadoras
     - $\Delta w_i(t)\ \alpha\ \ y(x) x_i$
4. ¿y(x)=d(x) para toda muestra de aprendizaje?
   - No: vuelta a 3
   - Sí: FINAL

Ejemplo de aprendizaje (perceptrón)

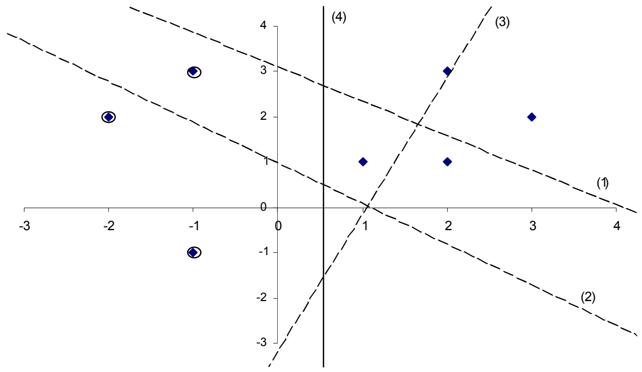
Iteraciones # 1

muestras	x_1	x_2	d(x)	u(x)	y(x)	w_1	w_2	θ
Pesos iniciales (1)						0.75	1	-3
A (2)	1	1	1	-1.25	-1	1,75	2	-2
B	2	1	1	3.5	1	1,75	2	-2
C	3	2	1	7.25	1	1,75	2	-2
D	2	3	1	7.5	1	1,75	2	-2
E	-1	-1	-1	-5.75	-1	1,75	2	-2
F	-2	2	-1	-1.5	-1	1,75	2	-2
G (3)	-1	3	-1	2.25	1	2,75	-1	-3

Iteraciones #2

muestras	x_1	x_2	d(x)	u(x)	y(x)	w_1	w_2	θ
A(4)	1	1	1	-1.25	-1	3.75	0	-2
B	2	1	1	5.5	1	3.75	0	-2
C	3	2	1	9.25	1	3.75	0	-2
D	2	3	1	5.5	1	3.75	0	-2
E	-1	-1	-1	-5.75	-1	3.75	0	-2
F	-2	2	-1	-9.5	-1	3.75	0	-2
G	-1	3	-1	-5.75	-1	3.75	0	-2
A	1	1	1	1.75	1	3.75	0	-2

Representación gráfica - ejemplo



Teorema de Convergencia del Perceptrón

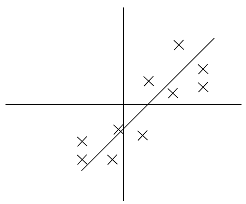
Si existe una transformación capaz de realizar:

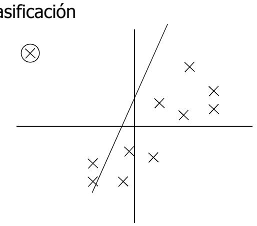
$$T : \Re^n \rightarrow \Re^m$$

para toda muestra de aprendizaje, entonces el algoritmo convergerá a una solución en un número finito de pasos. Es posible que esta solución sea diferente de la inicialmente supuesta.

Aprendizaje del Adaline

- La respuesta de la red es: $y_i^p = \sum_{j=1}^{n} w_{ij} x_j^p + \Theta_i$
- Aplicación del método de mínimos cuadrados
  - Distribuye el error para su minimización:
    - Problemas con puntos aislados
    - Malos resultados en clasificación





Regla Delta (I)

- Error cuadrático medio por muestra (p):

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

- El error total será: $E = \sum_p E^p$

- La modificación de pesos se basa en el algoritmo del gradiente muestra a muestra aplicado a E^p.

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

Regla Delta (II)

- Aplicación de la regla de la cadena:

$$\frac{\partial E^p}{\partial w_{ij}} = \frac{\partial E^p}{\partial y_i^p}\frac{\partial y_i^p}{\partial w_{ij}}$$

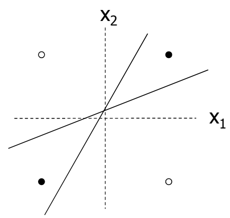
$$\frac{\partial E^p}{\partial y_i^p} = -(d_i^p - y_i^p) = -\delta_i^p \qquad \frac{\partial y_i^p}{\partial w_{ij}} = x_j^p$$

$$\Delta_p w_{ij} = \gamma \delta_i^p x_j^p$$

Problema del XOR (I)

- No es separable linealmente: una sola recta

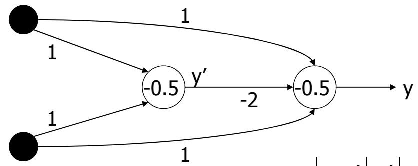
x_1	x_2	d
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1



- Minsky y Papert sugirieron una separación mediante combinación de varias rectas

Problema del XOR (II)

- Se propone añadir una neurona oculta



x_1	x_2	u′	y′	u	y	d
-1	-1	-2.5	-1	-0.5	-1	-1
-1	1	-0.5	-1	1.5	1	1
1	-1	-0.5	-1	1.5	1	1
1	1	1.5	1	-0.5	-1	-1