

Algoritmo de Retropropagacion. Ejemplo

Francisco José Ribadas Pena

INTELIGENCIA ARTIFICIAL

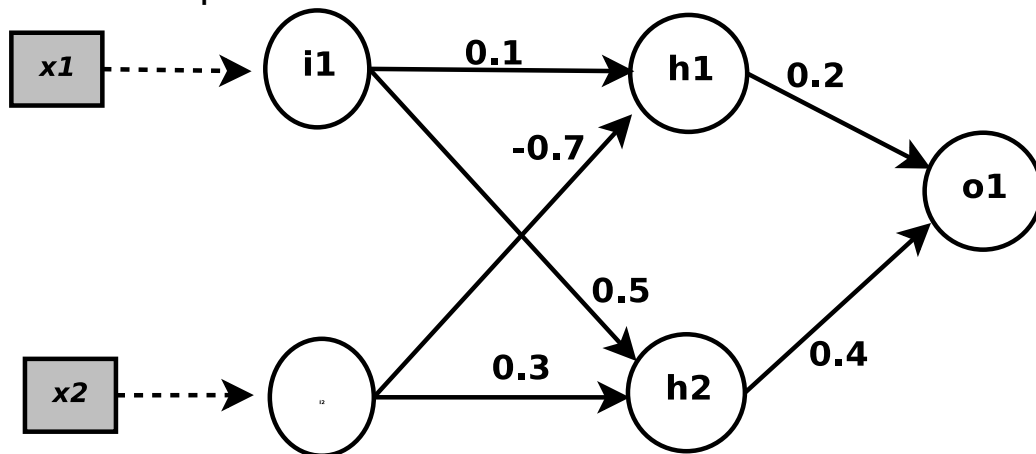
5º Informática

`ribadas@uvigo.es`

12 de diciembre de 2005

Descripción del ejemplo

- Entrenamiento de un perceptron multicapa para realizar la operacion XOR
- Descripción de la red.
 - 1 capa oculta
 - 2 neuronas en capa de entrada (i_1, i_2)
 - 2 neuronas en capa oculta (h_1, h_2)
 - 1 neurona en capa de salida (o_1)
- Red inicial con pesos aleatorios



$$W_1 = \begin{pmatrix} 0,1 & 0,5 \\ -0,7 & 0,3 \end{pmatrix} \quad W_2 = \begin{pmatrix} 0,2 \\ 0,4 \end{pmatrix}$$

- Conjunto de entrenamiento

	Entradas		Salida
	x_1	x_2	t_1
e_1	0	1	1
e_2	1	0	1
e_3	1	1	0
e_4	0	0	0

- Tasa de aprendizaje: $\alpha = 0,25$

Propagación hacia adelante del ejemplo e_1

■ Entradas

- $x_1 = 0, x_2 = 1$
- Salida esperada: $t_1 = 1$

■ Capa oculta

- Neurona h_1 :

$$\begin{aligned} \text{Entrada: } & 0,1 * 0 + (-0,7) * 1 = -0,7 \\ \text{Salida: } & \frac{1}{1+e^{0,7}} = \mathbf{0,332} \end{aligned}$$

- Neurona h_2 :

$$\begin{aligned} \text{Entrada: } & 0,5 * 0 + 0,3 * 1 = 0,3 \\ \text{Salida: } & \frac{1}{1+e^{-0,3}} = \mathbf{0,574} \end{aligned}$$

■ Capa de salida

- Neurona o_1 :

$$\begin{aligned} \text{Entrada: } & 0,2 * 0,332 + 0,4 * 0,574 = 0,296 \\ \text{Salida: } & \frac{1}{1+e^{-0,296}} = \mathbf{0,573} \end{aligned}$$

Abreviado en "notación vectorial"

- Salida capa oculta:

$$\vec{H} = g(\vec{X} \cdot W_1) = g\left((0, 1) \cdot \begin{pmatrix} 0,1 & 0,5 \\ -0,7 & 0,3 \end{pmatrix}\right) = (0,468, 0,574)$$

- Salida capa de salida:

$$\vec{O} = g(\vec{H} \cdot W_2) = g\left((0,468, 0,574) \cdot \begin{pmatrix} 0,2 \\ 0,4 \end{pmatrix}\right) = (0,573)$$

Ajuste de pesos por retropropagacion del error

(1) Pesos de la capa de salida

■ Neurona o_1 :

- Error real obtenido en neurona o_1 : $t_1 - o_1 = 1 - 0,573 = 0,427$
- Nuevos pesos para neurona o_1 : $W_{2jk} = W_{2jk} + \alpha h_j \Delta_k$

$$\begin{aligned}\Delta_k &= g'(ent_k) * (T_k - o_k) = o_k * (1 - o_k) * (T_k - o_k) \\ \Delta_1 &= 0,573 * (1 - 0,573) * 0,427 = 0,1044\end{aligned}$$

$$W_{211} = W_{211} + \alpha h_1 \Delta_1 = 0,2 + 0,25 * 0,332 * 0,1044 = 0,2086$$

$$W_{221} = W_{221} + \alpha h_2 \Delta_1 = 0,4 + 0,25 * 0,574 * 0,1044 = 0,4149$$

(2) Pesos de la capa oculta

■ Formulas de ajuste:

- Error estimado en neurona h_j :

$$\Delta_j = g'(ent_j) \sum_{k=1}^p (W_{2jk} \Delta_k) = h_j (1 - h_j) \sum_{k=1}^p (W_{2jk} \Delta_k)$$

OJO: W_{2jk} se refiere a los pesos de la capa de salida antiguos (antes del ajuste anterior)

- Nuevos pesos para neurona h_j : $W_{1ij} = W_{1ij} + \alpha i_i \Delta_j$

■ Neurona h_1 :

- Error estimado:

$$\begin{aligned} \Delta_1 &= h_1 * (1 - h_1) * (W_{211} * 0,1044) = \\ &= 0,332 * (1 - 0,332) * (0,2 * 0,1044) = 0,046 \end{aligned}$$

- Nuevos pesos:

$$\begin{aligned} W_{111} &= W_{111} + \alpha i_1 \Delta_1 = 0,1 + 0,25 * 0 * 0,046 = 0,1 \\ W_{121} &= W_{121} + \alpha i_2 \Delta_1 = -0,7 + 0,25 * 1 * 0,046 = -0,684 \end{aligned}$$

■ Neurona h_2 :

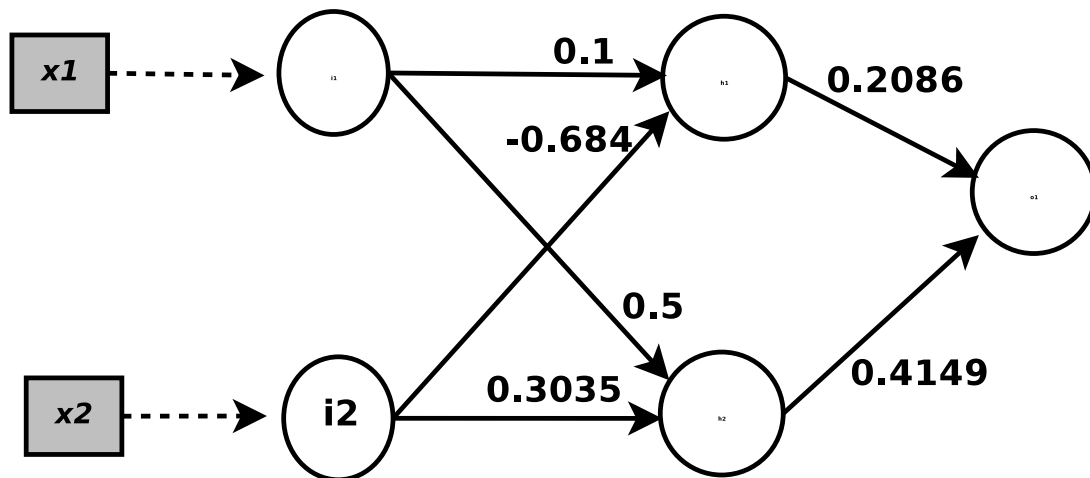
- Error estimado:

$$\begin{aligned} \Delta_2 &= h_2 * (1 - h_2) * (W_{221} * 0,1044) = \\ &= 0,574 * (1 - 0,574) * (0,4 * 0,1044) = 0,0102 \end{aligned}$$

- Nuevos pesos:

$$\begin{aligned} W_{112} &= W_{112} + \alpha i_1 \Delta_2 = 0,5 + 0,25 * 0 * 0,0102 = 0,5 \\ W_{122} &= W_{122} + \alpha i_2 \Delta_2 = 0,3 + 0,25 * 1 * 0,0102 = 0,3025 \end{aligned}$$

Nueva red



Para el mismo ejemplo e_1 , la salida será un poco mejor (más cercana al objetivo)

- Salida capa oculta:

$$\vec{H} = g(\vec{X} \cdot W1) = g\left((0, 1) \cdot \begin{pmatrix} 0,1 & 0,5 \\ -0,684 & 0,3035 \end{pmatrix}\right) = (0,335, 0,575)$$

- Salida capa de salida:

$$\vec{O} = g(\vec{H} \cdot W2) = g\left((0,335, 0,575) \cdot \begin{pmatrix} 0,2086 \\ 0,4149 \end{pmatrix}\right) = (0,576)$$