Redes neuronales

Cerebro: $\sim 10^{11}$ neuronas

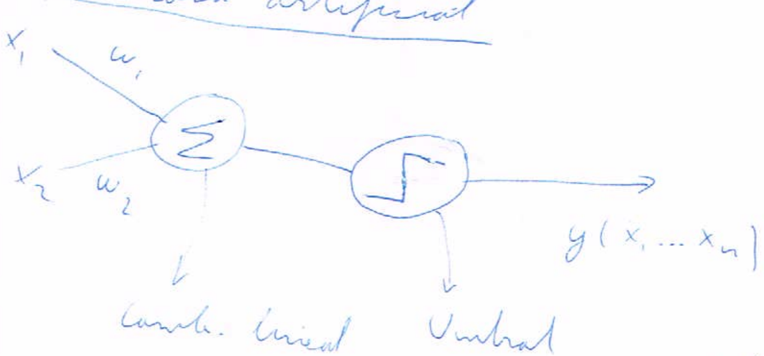
Neurona: de media 1.000 salidas

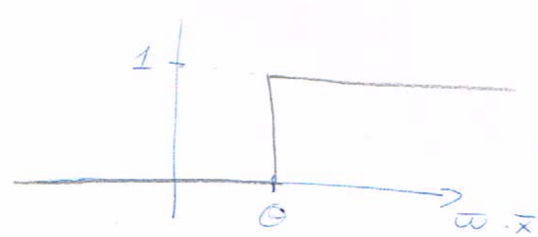
$\sim$1.000 sinapsis / neurona

Propiedades

- La neurona recibe impulsos (inhibidores o excitadores)
  ↳ Atributos
- Esta entrada tendrá distinta intensidad
  ↳ Pesos, parámetros
- El cuerpo de la neurona integra (suma) estos impulsos
  ↳ Combinación lineal
- Si esta suma está por encima de un umbral, entonces dispara el potencial de acción a la siguiente
  ↳ Salida

Neurona artificial





$$y(\bar{x}) = \begin{cases} 1 & \bar{x} \cdot \bar{w} \geq \theta \\ 0 & \bar{x} \cdot \bar{w} < \theta \end{cases}$$

Umbral: función de Heaviside

AND $w_1 = w_2 = 1$; $\theta = 1,5$

x_1	x_2	t	$w_1x_1 + w_2x_2$	y
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	1	2	1

OR $w_1 = w_2 = 1$ $\theta = 0,5$

x_1	x_2	t	$w_1x_1 + w_2x_2$	y
0	0	0	0	0
1	0	1	1	1
0	1	1	1	1
1	1	1	2	1

Integran el umbral

$$y(x_1 \dots x_n) = \begin{cases} 1 & \text{si} \quad w_0x_0 + w_1x_1 + \dots w_nx_n \geq 0 \\ 0 & \text{si} \quad w_0x_0 + \dots + w_nx_n < 0 \end{cases}$$

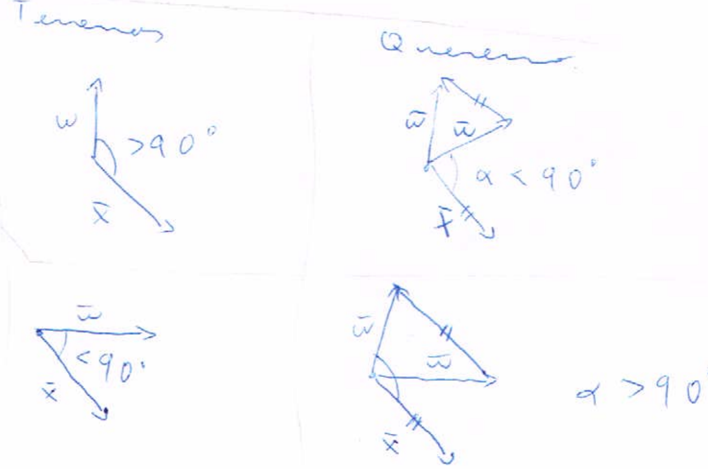
$w_0 = \theta$ y $x_0 = 1$ siempre

Perceptrón de Rosenblatt

$$y(\bar{x}) = \begin{cases} 1 & \text{si} \quad \bar{w}^T\bar{x} \geq 0 \implies \bar{x} \in C_1 \quad t = 1 \\ 0 & \text{si no} \implies \bar{x} \in C_2 \quad t = 0 \end{cases}$$

Tipos de error

- Si $y(\bar{x}) = 0$ y $\bar{x} \in C_1$
- Si $y(\bar{x})$ 1 y $\bar{x} \in C_2$



Con ángulo entre vectores

$-90 \leq \text{ang} \leq 90° \implies$ producto escalar > 0

Criterio de Rosemblatt

$$J(\bar{w}) = \sum_{n=1}^{N} \delta_n \bar{w}^T \bar{x}_n$$

donde

$$\delta_n = y(\bar{x}_n) - t(\bar{x}_n)$$

$J(\bar{w})$ es siempre positivo

Vamos a minimizar $J(\bar{w})$ derivando con respecto a $\bar{w}$

$$\frac{\partial J(\bar{w})}{\partial \bar{w}} = \sum_{n=1}^{N} \delta_n \bar{x}_n$$

regla de actualización → cte. aprendizaje

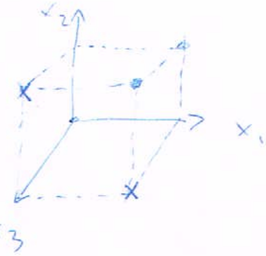
$$\bar{w} \to \bar{w} - \eta \, \delta_n \, \bar{x}_n$$

```
perceptron_Rosemblatt(Datos, η):
    w̄ = aleat_vals(-0,5 ; 0,5)
    Mientras errores:
        Para cada ejemplo:
            y = y(x̄);
            δ = y - t(x̄);
            w̄ = w̄ - η · δ · x̄
    return w̄
```

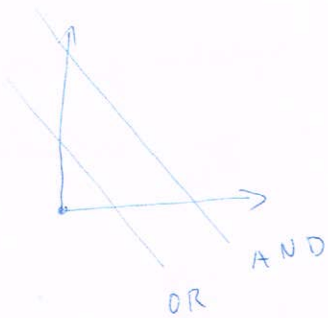
XOR

x_1	x_2	t
0	0	0
1	0	1
0	1	1
1	0	0

$$x_3 = (x_1 - x_2)^2$$



Vamos a encerrar una clase entre dos rectas

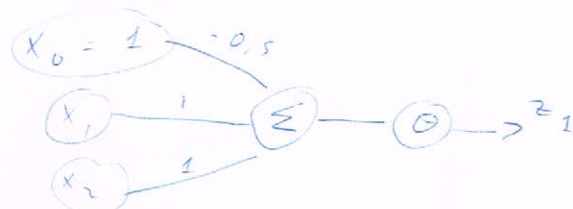


AND $x_1 + x_2 - 1,5 = 0$

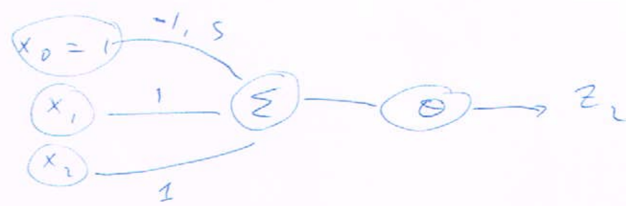
OR $x_1 + x_2 - 0,5 = 0$

Definimos los perceptrones

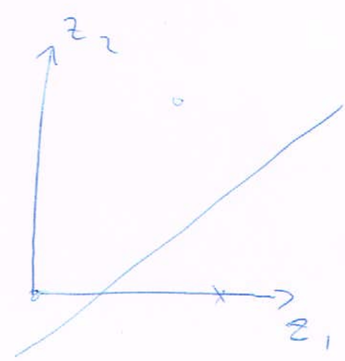
$$z_1 = \theta(x_1 + x_2 - 0,5)$$



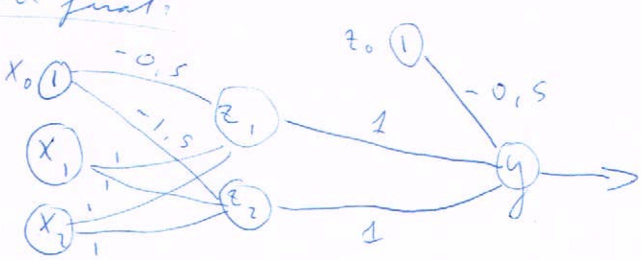
$$z_2 = \theta(x_1 + x_2 - 1,5)$$



x_1	x_2	t	z_1	z_2	y
0	0	0	0	0	0
0	1	1	1	0	1
1	0	1	1	0	1
1	1	0	1	1	0



Red final:



$w_0 = -0,5$

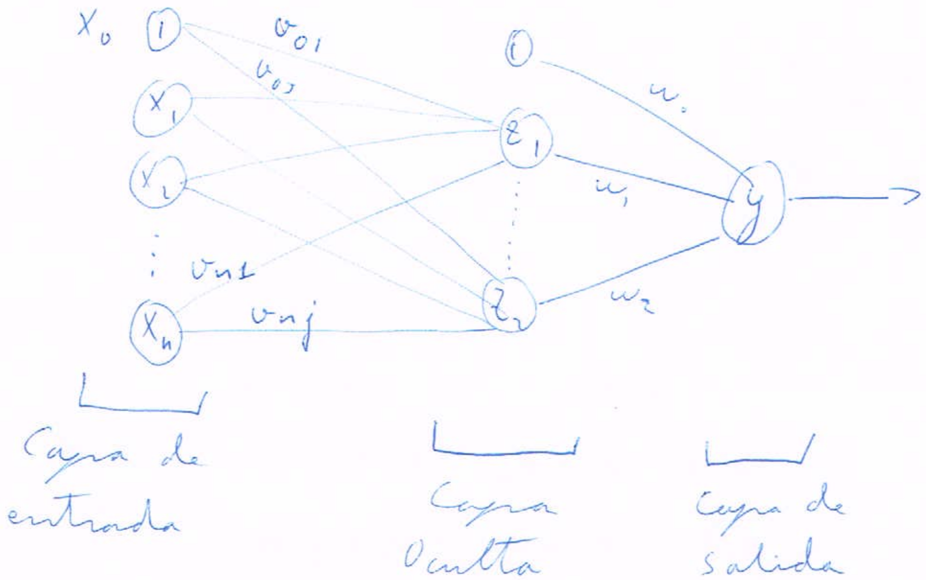
$w_1 = 1$

$w_2 = -1$

$y = \theta(z_1 - z_2 - 0,5$

Regla delta

Sustituir las f. de Heaviside θ por sigmoides σ



v_{dj}: peso entre neurona d (entrada y neurona j (salida

w_j: peso entre neurona j (oculta) e y (salida)

 # params. ajustar $= (J+1) + [(D+1) \times J]$

Calculemos el valor de y en función de $\bar{x}$

$$\bar{x} = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} \qquad z = \begin{pmatrix} 1 \\ z_1 \\ \vdots \\ z_J \end{pmatrix} \qquad z_j = \sigma\left(\sum_{d=0}^{D} x_d \cdot v_{dj}\right)$$

$$y(\bar{x}) = \sigma\left(\sum_{j=0}^{J} w_j \cdot z_j\right) = \sigma\left(\sum_{j=0}^{J} w_j \cdot \sigma\left(\sum_{d=0}^{D} x_d \cdot v_{dj}\right)\right)$$

Optimización de pesos v_{dj} y w_j por MV

$\{\bar{x}_n, t_n\}$ $n = 1 \ldots N$

$H = v_{dj} \; w_j$ con $d = 1 \ldots D$, $j = 1 \ldots J$

$D = t_1 \ldots t_N$ $\quad I = \bar{x}_1 \ldots \bar{x}_N$

MV

$$p(t_1 \dots t_N \mid V, \bar{w}, \bar{x}_1 \dots \bar{x}_N) =$$

con $V = \begin{pmatrix} v_{01} & v_{02} & \cdots & v_{03} \\ & \ddots & & \\ v_{D1} & & & v_{DS} \end{pmatrix}$

$$= \prod_{n=1}^{N} p(C_1 \mid \bar{x}_n, V, \bar{w})^{t_n} \cdot (1 - p(C_1 \mid \bar{x}_n, V, w))^{1-t}$$

aplicando ln

$$E(V, \bar{w}) = \log p(t_1 \dots t_N \mid V, \bar{w}, \bar{x}_1 \dots \bar{x}_N) =$$

$$= \sum_{n=1}^{N} E_n(V, \bar{w})$$

$$E_n(V, \bar{w}) = t_n \cdot \log p(C_1 \mid V, \bar{w}, \bar{x}_n) + (1-t_n) \log(1 - p(C_1), V$$

$$= t_n \log y(\bar{x}_n) + (1-t_n) \cdot \log(1 - y(\bar{x}_n))$$

derivada "comodín"

$$\frac{\partial E_n(V, \bar{w})}{\partial \beta}$$

$$\frac{\partial E_n}{\partial \beta} = -t \frac{1}{y(\bar{x}_n)} \cdot \frac{\partial y(\bar{x}_n)}{\partial \beta} + (1 - t_n) \frac{1}{1 - y(\bar{x}_n)} \cdot \frac{\partial y(\bar{x}_n)}{\partial \beta} =$$

$$\left[\frac{\partial y(\bar{x}_n)}{\partial \beta} = y(\bar{x}_n)\left(1 - y(\bar{x}_n)\right) \frac{\partial \bar{w}^T \bar{z}}{\partial \beta} \right.$$

$$= \left(-t_n\left(1 - y(\bar{x})\right) + (1 - t_n)\, y(\bar{x}_n)\right) \frac{\partial \bar{w}^T \bar{z}}{\partial \beta} =$$

$$= \left(y(\bar{x}_n) - t_n\right) \cdot \frac{\partial \bar{w}^T \bar{z}}{\partial \beta}$$

Ahora derivamos respecto a w_i, $(\beta = w_i$

$$\frac{\partial \bar{w}^T \bar{z}}{\partial w_i} = z_i \qquad \boxed{\frac{\partial E_n}{\partial w_i} = \left(y(\bar{x}_n) - t_n\right) \cdot z_i}$$

derivamos con respecto a v_{pq} $(\beta = v_{pq})$

$$\frac{\partial \bar{w}^T \bar{z}}{\partial v_{pq}} = \frac{\partial}{\partial v_{pq}} \cdot \sum_{j=0}^{J} w_j \cdot z_j = \sum_{j=0}^{J} w_j \frac{\partial}{\partial v_{pq}} \sigma\left(\sum_{d=0}^{D} x_d \cdot v_{dj}\right.$$

$$= \sum_{j=0}^{J} w_j\, z_j (1 - z_j) \frac{\partial}{\partial v_{pq}} \cdot \sum_{d=0}^{D} x_d \cdot v_{dj}$$

Delta de Kronecker

$$= \sum_{j=0}^{J} w_j \cdot z_j (1 - z_j) \sum_{d=0}^{D} x_d \frac{\partial v_{dj}}{\partial v_{pq}} = \delta_{dp}\, \delta_{jq}$$

si $d == p \Rightarrow 1$

si no $\Rightarrow 0$

$$= \sum_{j=0}^{J} w_j z_j (1 - z_j) \cdot \sum_{d=0}^{D} x_d \delta_{dp} \delta_{jq}$$

$$= \sum_{j=0}^{J} w_j z_j (1 - z_j) \cdot x_p \delta_{jq}$$

$$= w_q z_q (1 - z_q) x_p$$ → volvemos atrás para meterla

$$\boxed{\frac{\partial E_n}{\partial v_{pq}} = (y(\bar{x}_n) - t_n) w_q \cdot z_q (1 - z_q) x_p}$$

Algoritmo de retropropagación

Reglas de actualización

$$w_i \to w_i - \eta (y(\bar{x}_n) - t_n) z_i$$

$$v_{pq} \to v_{pq} - \eta (y(\bar{x}_n) - t_n) w_q z_q (1 - z_q) x_p$$

alg_retroprog(η, Datos, epocas):

- ini V, $\bar{w}$ con valores $[-0,5 ; 0,5]$
- for 1 → epocas — epoc
  - for 1 → N — N
    - $z_q = \sigma(\sum_{d=0}^{D} x_d v_{dq})$ — $J \times D$
    - $y = \sigma(\bar{w}^T \bar{z})$ — J } Ida
    - $\Delta q = (y - t_n) w_q z_q (1 - z_q)$ — J } Vuelta 3

$$w_q = w_q - \eta (y - t_i) z_q \quad J$$

$$v_{pq} = v_{pq} - \eta \Delta_q x_p \quad D \times J$$

vuelta

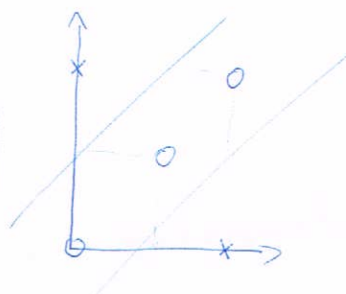
return $V, \bar{w}$

$$\boxed{\text{coste} = O(\text{epoc} \cdot N \cdot D \cdot J)}$$

Ejer:

Diseñar perceptrón multicapa con funciones escalón θ para resolver

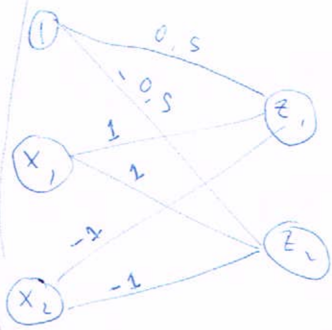
y	x_1	x_2	t	z_1	z_2
0	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	1	1
0	1	1	0	1	0
0	0,5	0,5	0	1	0



$$z_1 = x_1 - x_2 + 0,5$$

$$z_2 = x_1 - x_2 - 0,5$$

$$z = \theta(\bar{v}^T \bar{x})$$

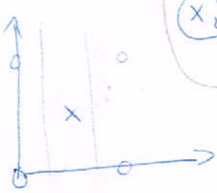


$$z_1 = \theta(x_1 - x_2 + 0,5)$$

$$z_2 = \theta(x_1 - x_2 - 0,5)$$

$$y = \theta(-z_1 + z_2 + 0,5)$$

Ejer:

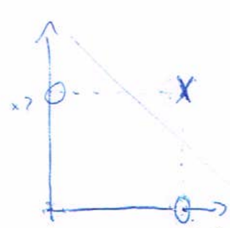


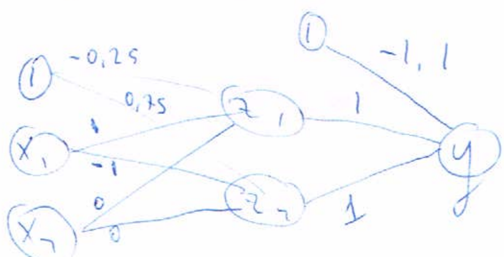
x_1	x_2	t	z_1	z_2	y
0	0	0	0	1	0
0	1	0	0	1	0
1	0	0	1	0	0
1	1	0	1	0	0
0,5	0,5	1	1	1	1

$$z_1 = \theta(x_1 - 0,25)$$

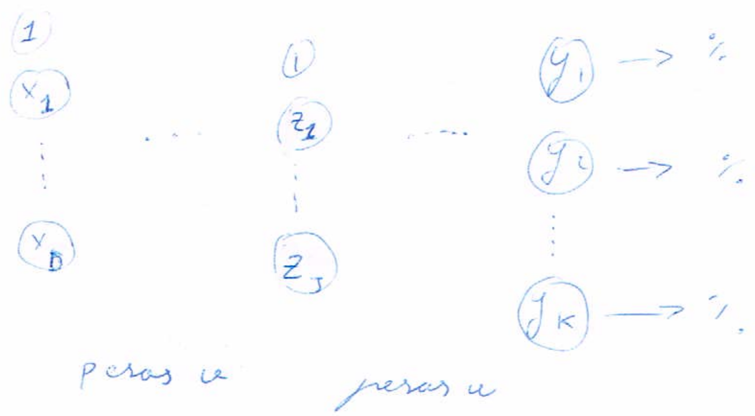
$$z_2 = \theta(-x_1 + 0,75)$$

$$y = \theta(z_1 + z_2 - 1,1)$$





Problemas multiclase



D = # atrib.

J = neuronas ocultas

K = # clases

$$y_K(\bar{x}_n) = p(C_K \mid v, w, \bar{x}_n)$$

por Bayes

$$p(C_K \mid \bar{x}) = \frac{p(\bar{x} \mid C_K) \cdot p(C_K)}{\sum_{K=1}^{K} p(\bar{x} \mid C_K) \cdot p(C_K)} = \frac{\exp(a_K)}{\sum \exp(a_K)}$$ softmax

con $a_K = \ln p(\bar{x} \mid C_K) \cdot p(C_K)$

en nuestro caso $a_K = \sum_{j=0}^{J} z_j \cdot w_{jK}$

Supongamos que a_K son las entradas a la capa de salida, esto es,

$$a_K = \ln p(\bar{x} \mid C) \cdot p(C_K) = \sum_{j=1}^{J} z_j \, w_{jK}$$

La salida final de la red es

$$y_{nK} = \frac{\exp(a_K)}{\sum_{K=1}^{K} \exp(a_K)} = \frac{\exp\left(\sum_{j=1}^{J} z_j \, w_{jK}\right)}{\sum_{K=1}^{K} \exp\left(\sum_{j=1}^{J} z_j \, w_{jK}\right)}$$

Disponemos de $D_{train} = (\{\bar{x}_n, \bar{t}_n\}\ n = 1 \dots N)$

$\bar{t}_n$ va a ser un vector de dimensión K (no. de clases) con todo 0 excepto la posición de K clase que será 1

$$\text{si: } \bar{x}_n \in K \Rightarrow \bar{t}_n\ (0, \dots 0, 1, 0 \dots 0)$$

$$\bar{t}_n = \begin{cases} t_{nK} = 1 \\ t_{nj} = 0 & j \neq K \end{cases}$$

$H = v, w$

$D = \bar{t}_1, \dots, \bar{t}_N \qquad I = \bar{x}_1 \dots \bar{x}_N$

Max por MV

$$p(D \mid H, I) = p(\bar{t}_1 \dots \bar{t}_N \mid \bar{x}_1 \dots \bar{x}_N, v, w) =$$

$$= \prod_{n=1}^{N} \underbrace{p(\bar{t}_n \mid \bar{x}_n, v, w)}_{= \prod_{K=1}^{K} y_{nK}^{t_{nK}}} = \prod_{n=1}^{N} \prod_{K=1}^{K} y_{nK}^{t_{nK}}$$

Aplicando el $-\ln$

$$E(v, w) = -\ln p(\bar{t}_1 \dots \bar{t}_N \mid \bar{x}_1 \dots \bar{x}_N, w, v) = -\sum_{n=1}^{N} \sum_{K=1}^{K} t_{nK} \ln y$$

y para el ejemplo n:

$$E_n(v, w) = -\sum_{K=1}^{K} t_{nK} \ln y_{nK}$$

Derivando

$$\frac{\partial E_n}{\partial w_{qr}} = \dots = (y_{nk} - t_{nr}) \cdot Z_{nq}$$

$$\frac{\partial E}{\partial v_{pq}} = \dots = \sum_{j=1}^{K} (y_{nj} - t_{nj}) w_{qj} \cdot Z_{nq} (1 - Z_{nq}) \cdot X_{np}$$

---

alg_retropropagación_mc (D_{train}, η, epocas)

Ini v, w entre [-0,5 ; 0,5]

for $i \to$ epocas $\quad \to$ epoc

for $n \to N$ $\quad \to N$

// Se calcula la red

$Z_{nj} = \sigma \left(\sum_{i=0}^{D} X_{ni} \cdot v_{ij} \right) \quad \to D \times J$

$a_{nK} = \sum_{j=0}^{J} z_{nj} \cdot w_{jK} \quad \to K \times J$

$y_{nK} = \exp(a_{nK}) / \sum_{n=1}^{K} \exp(a_{nK}) \quad \to K$

// Para cada unidad de salida

$\Delta_{nK} = y_{nK} - t_{nK} \quad \to K$

// Para cada neurona oculta

$$\Delta_{nj} = \sum_{k=1}^{K} \Delta_{nk} \, w_{jk} \, z_{nj} (1 - z_{nj}) \longrightarrow J$$

// Actualizamos peso

$$w_{jk} = w_{jk} - \eta \Delta_{nk} z_{nj} \longrightarrow J \times K$$

$$v_{ij} = v_{ij} - \eta \Delta_{nj} x_{ni} \longrightarrow D \times J$$

return v, w

$$O(\text{Epocas} \times N \times (DJ + JK))$$

Coste clasificar 1 ejemplo (test)

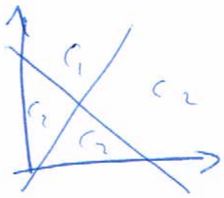
$$O(DJ + JK)$$

<u>Ejer:</u>

4 clases, 50 atribs, 10^6 ejemplos, 9 ocultas, 10 epocas

¿Actualizaciones de pesos?

$$\left.\begin{array}{l} 50 \times 9 \\ \quad 10 \times 4 \end{array}\right\} 490 + 10^6 \cdot 10'$$



$z_1 = \theta(\ )$

$z_2 = \theta \cdot (\ )$

¿Que pasa con la frontera si $\theta \to \nabla$?

Conjuntos de clasificadores

Entrenar conjunto de clasificadores y combinar decisiones.

Generalmente: la suma mejor que cada uno

- La clave es generar diversidad

Combinación

- Por mayoría
- Voto ponderado

Pongamos que tenemos T clasificadores

$$h_1(\bar{x}), h_2(\bar{x}) \dots h_T(\bar{x})$$

tal que $h(\bar{x}) = y$ con $y \in \{1, \dots c\}$

* Combinación por mayoría

$$H(\bar{x}) = \arg\max_j \left(\sum_{i=1}^{T} \mathbb{I}(h(\bar{x}) = j) \right)$$

donde $\mathbb{I}(\text{true}) = 1$ y $\underbrace{\mathbb{I}(\text{false}) = 0}_{\text{Función indicatriz}}$

* Combinación por voto ponderado.

$$H(\bar{x}) = \underset{j}{\mathrm{argmax}} \left(\sum_{i=1}^{T} \alpha_i \, \mathbb{I} \left(h_i(\bar{x}) = j \right) \right)$$

Peso para el clasificador h_i

Crear diversidad

1 - Combinación de clasificadores heterogeneos

$h_1 = KNN$ $\quad h_2 =$ reg. log. $\quad h_3 =$ red. neu.

2 - Modificar muestras de datos

- Bagging
- Boosting

3 - Aleatoriedad en los algoritmos

Bagging

```
muestra_bootstrap (D)
    for i = 1 -> N
        muestra [i] = D [rand (1, N)]
    return muestra;
```

ej: $D = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5, \bar{x}_6)$

muestra $= \{\bar{x}_6, \bar{x}_5, \bar{x}_1, \bar{x}_1, \bar{x}_4, \bar{x}_3\}$ Hay repeticiones y otras faltan.

bagging (D_{Train} , T)

for $i = 1 \longrightarrow T$

  muestra = muestra_bootstrap (D_{Train})

  h_i = entrena_modelo (muestra)

return $H(\cdot) = \arg\max_j \left(\sum_{i=1}^{T} \mathbb{I} (h_i(\cdot) = j) \right)$

Boosting

Modificar importancia de cada ejemplo en cada iteración

boosting (D_{Train} , T)

  $w_i = 1/N \quad (i = 1 \ldots i = N)$

  for $t = 1 \longrightarrow T$

    h_t = entrena_modelo (D, $\overline{w}$)

    error = $\sum_{i=1}^{N} (\mathbb{I} (h_t(\bar{x}_i) != y_i) w_i) / \sum_{i=1}^{N} w_i$

    $\alpha_t = \log \left(\frac{1 - error}{error} \right)$

    if (error $\geqslant$ 0,5)

      descarta h_t

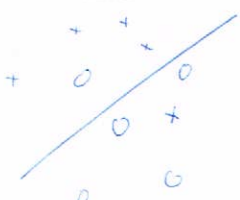
      break

```
if (error == 0)
    break
for i = 1 -> N
    if (h_t(x̄_i) == y_i)
        w_i = w_i / (2(1 - error))   // reduce peso
    else
        w_i = w_i / (2 error)        // aumenta peso
return H(·) = argmax_j Σ α_t 𝕀(h_t(·) == j)
```

$$H(\cdot) = \arg\max_j \sum \alpha_t \, \mathbb{I}(h_t(\cdot) == j)$$

ej:

ite 1



$w_i = \frac{1}{N} = \frac{1}{10}$

error $h_1 = \frac{2/10}{10/10} = 0,2$

$\alpha_1 = \log\left(\frac{0,8}{0,2}\right)$

Pesos bien clasf. h_1

$w_{bien} = \frac{1}{10 \cdot 2(1-0,2)} = 0,0625$

Pesos mal clasf. h_2

$w_{mal} = \frac{1}{10(2 \cdot 0,2)} = 0,25$

ite 2

Si sumas pesos de las mal = 0,5

Si sumas pesos de la bien = 0,5

$8 * 0,0625 = 0,5$

Mal: $2 * 0,25 = 0,25$

¿Que modelo utilizar?

- Se suelen utilizar árboles de decisión
- Se buscan modelos no demasiado estables

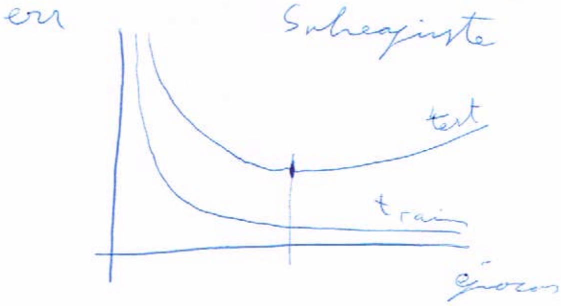
¿Tamaño?

$T \geq 100$ generalmente

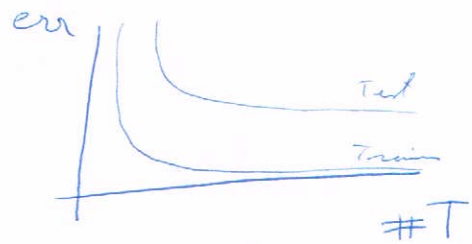
Los conjuntos no sobre ajustan

Red neu (u otros)

Sobreajuste

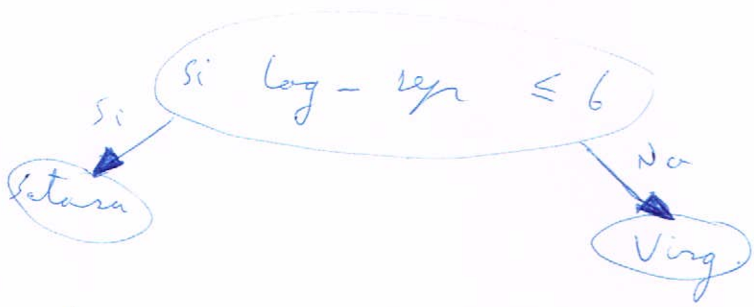


Conjuntos



Tocón de decisión (stumps)

Árbol de un solo nivel



¿Como busca pregunta?

Buscamos la división que reduzca más la impureza de entre todas las divisiones posibles

$$\Delta i = i(\text{raiz}) - P_I\, i(n_I) - P_D\, i(n_D)$$

$i(n)$ = impureza nodo n

P_I = prop. ej. nodo izq.

Criterio de Gini

$$i(\text{nodo}) = 2 \cdot p(C_1 \mid \text{Nodo}) \cdot p(C_2 \mid \text{Nodo})$$

$[0 ; 1]$; 0,5 la mas impura (mitad y mit

Ejem:

$X_1 \downarrow$	C_1
1	+
1,5	+
2	-
2,5	-
3	+
5	-
5,5	+
7	+
8	-
8	-

$$i(\text{raiz}) = 2 \cdot \frac{5}{10} \cdot \frac{5}{10} = 0,5$$

$X_1 \leq 1,75$

izq: 2+, 0-

der: 3+, 5-

$$P_I = \frac{2}{10}$$

$$P_D = \frac{8}{10}$$

Habría que hacerlo sobre todos y sobre X_2

$$\Delta i = 0,5 - \frac{2}{10} \cdot 2 \cdot \frac{2}{2} \cdot \frac{0}{2} - \frac{8}{10} \cdot 2 \cdot \frac{3}{8} \cdot \frac{5}{8} = \frac{1}{8}$$

$X_1 \leq 2,75$

izq: 2+, 2-

der: 3+, 3-

$\Delta i = 0$

$X_1 \leq 4$

izq: 3+, 2-

der 2+, 3-

$$\Delta i = 0,5 - \frac{5}{10} \cdot 2 \cdot \frac{3}{5} \cdot \frac{2}{5} - \frac{5}{10} \cdot 2 \cdot \frac{2}{5} \cdot \frac{3}{5} = 0,0$$

$h_1 \equiv (X_1 \leq 1,75)$

$$\alpha_1 = \log\left(\frac{1-err}{err}\right) = 0,37$$

$$\text{error } h_1 = \frac{3/10}{10/10} = 0,3$$

pesos

Mal clasificados

$$w_5 = \underbrace{\frac{1}{10} \cdot \frac{1}{2 \cdot 0{,}3}}_{\shortparallel} = 0{,}16\bar{6}$$

$w_7 = \ldots \quad \shortparallel$

$w_8 = \quad \shortparallel$

Bien clasificados

$$w_1 = \frac{1}{10} \cdot \frac{1}{2(1-0{,}3)} = 0{,}071$$

Igual para 2, 3, 4, 6, 9, 10

$x_2 \leq 6{,}25$

$$i(\text{raiz}) = 2 \cdot 0{,}355 \cdot (2 \times 0{,}071 + 3 \times 0{,}16\bar{6}) = 0{,}45$$

	+	-
Izq :	$0{,}071 * 2$	$0{,}071 * 4$
Der :	$0{,}5$	$0{,}071$

$$p_I = \frac{0{,}426}{0{,}426 + 0{,}571} = 0{,}426$$

$$p_D = 0{,}571$$

$$\Delta i = 0{,}45 - p_I \cdot i(n_I) - p_D \cdot i(n_d) =$$

$$= 0{,}45 - 0{,}426 \cdot 2 \cdot \frac{0{,}071 * 4}{0{,}071 * 6} \cdot \frac{0{,}071 * 2}{0{,}071 * 6}$$

$$- 0{,}571 \cdot 2 \cdot \frac{0{,}5}{0{,}5 + 0{,}071} \cdot \frac{0{,}071}{0{,}571} = 0{,}13632$$

42

Habría que probar el resto pero $h_2 \equiv (x_2 \le 6,25)$

error $h_2 = 0,213$

α_2 ...

modificar pesos.

...

Ejer:

h_1 $\quad x_1 \le 6,5 \Rightarrow +$ $\qquad \alpha_1 = 0,6$

else $\quad -$

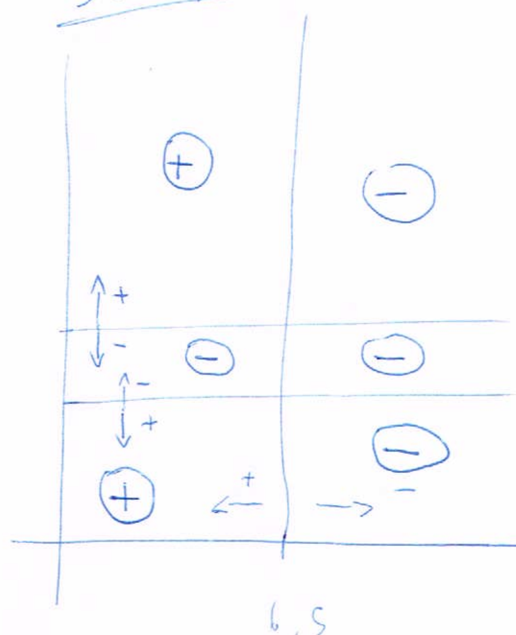
h_2 $\quad x_2 \le 3,25 \Rightarrow +$ $\qquad \alpha_2 = 0,48$

else $\quad -$

h_3 $\quad x_2 \le 4,7 \Rightarrow -$ $\qquad \alpha_3 = 0,14$

else $\quad +$

Sin pesos



Con pesos