REDES NEURALES

EJEMPLOS DE REDES NEURALES - PERCEPTRON

RESOLUCIÓN DE EJERCICIO TIPO PARA PERCEPTRON SIMPLE (MONOCAPA)

Un perceptron está formado por varias neuronas lineales para recibir las entradas a la red y una neurona de salida; es capaz de decidir cuando una entrada presentada a la red pertenece a una de las clases que es capaz de reconocer.

La única neurona de salida del Perceptron realiza la suma ponderada de las entradas y pasa el resultado a una función de transferencia. La regla de decisión es responder +1 si el patrón presentado pertenece a la clase A, o -1 si el patrón pertenece a la clase B.

Regla de Aprendizaje:
El algoritmo de aprendizaje es del tipo supervisado, lo cual requiere que sus resultados sean evaluados y se realicen las modificaciones del sistema si fuera necesario.

EJERCICIO 4.a

Dado un perceptron en una red neural de una capa, con dos entradas y una salida, ajuste los pesos asociados a cada entrada para que la salida de este sistema responda según la función AND de las entradas.

La función de activación o transferencia está dada por:

$$Y = \begin{cases} 1 \text{ si } u > 0 \\ 0 \text{ si } u \leq 0 \end{cases}$$

Siendo $u = \sum w_i \cdot x_i = w_1 * x_1 + w_2 * x_2 + w_3$

Factor de ganancia $\alpha = 1$

ENTRADAS		SALIDA DESEADA
X1	X2	
0	0	0
0	1	0
1	0	0
1	1	1

REDES NEURALES

EJEMPLOS DE REDES NEURALES - PERCEPTRON

PRIMER PASO: INICIALIZAR LOS PESOS

$w_3 = 1,5$ $w_1 = 0$ $w_2 = -2,5$

SEGUNDO PASO: PRESENTAR ENTRADAS

$x_3 = 1$ (umbral) $x_1 = 0$ $x_2 = 0$ salida deseada: 0

TERCER PASO: CALCULAR LA SALIDA ACTUAL Y ERROR

$Y_{(t)} = f\,[\sum w_{i\,(t)} \cdot x_{i\,(t)}]$

Y= f [(1,5* 1) + (0 * 0) + (- 2,5 * 0)]

Y = f [1,5] > *0*

Y = 1 como la salida deseada es 0 y la salida calculada es 1 debemos calcular el error y modificar los pesos.
(aprendizaje por corrección de error)

Error = Salida deseada – salida calculada

Error = $[\,d_{(t)} - y_{(t)}\,]$

- 1 = 0 – 1

REDES NEURALES

EJEMPLOS DE REDES NEURALES - PERCEPTRON

CUARTO PASO: ADAPTACIÓN DE LOS PESOS

$w_{i\ (t+1)} = w_{i\ (t)} + \alpha\ [\ d_{(t)} - y_{\ (t)}\ ]\ .\ x_{i\ (t)}$

$w_3 = 1{,}5 + 1\ .\ [\ -1\ ]\ .\ 1 = 0{,}5$

$w_1 = 0 + 1\ .\ [\ -1\ ]\ .\ 0 = 0$

$w_2 = -\ 2{,}5 + 1\ .\ [\ -1\ ]\ .\ 0 = -\ 2{,}5$

PRIMER PASO: INICIALIZAR LOS PESOS

$w_3 = 0{,}5$ $w_1 = 0$ $w_2 = -\ 2{,}5$

SEGUNDO PASO: PRESENTAR ENTRADAS

$x_3 = 1$ (umbral) $x_1 = 0$ $x_2 = 1$ salida deseada: 0

TERCER PASO: CALCULAR LA SALIDA ACTUAL Y ERROR

$Y = f\ [\ (0{,}5 * 1) + (0 * 0) + (-\ 2{,}5 * 1)]$

$Y = f\ [\ -\ 2\ ] \leq 0$

$Y = 0$

Error = Salida deseada – salida calculada

0 – 0 = 0 como no hay error los pesos no se actualizan y continúo con la entrada siguiente.

REDES NEURALES

EJEMPLOS DE REDES NEURALES - PERCEPTRON

PRIMER PASO: INICIALIZAR LOS PESOS

$w_3 = 0{,}5$ $w_1 = 0$ $w_2 = -2{,}5$

SEGUNDO PASO: PRESENTAR ENTRADAS

$x_3 = 1$ (umbral) $x_1 = 1$ $x_2 = 0$ salida deseada: 0

TERCER PASO: CALCULAR LA SALIDA ACTUAL Y ERROR

$Y = f\,[\,(0{,}5 * 1) + (0 * 1) + (-2{,}5 * 0)]$

$Y = f\,[\,0{,}5\,] > 0$

$Y = 1$

Error = Salida deseada – salida calculada

0 – 1 = - 1 hay error

CUARTO PASO: ADAPTACIÓN DE LOS PESOS

$w_3 = 0{,}5 + 1\,.\,[-1]\,.\,1 = -0{,}5$
$w_1 = 0 + 1\,.\,[-1]\,.\,1 = -1$
$w_2 = -2{,}5 + 1\,.\,[-1]\,.\,0 = -2{,}5$

REDES NEURALES

EJEMPLOS DE REDES NEURALES - PERCEPTRON

PRIMER PASO: INICIALIZAR LOS PESOS

w_3 = - 0,5 w_1 = - 1 w_2 = - 2,5

SEGUNDO PASO: PRESENTAR ENTRADAS

x_3= 1 (umbral) x_1= 1 x_2 = 1 salida deseada: 1

TERCER PASO: CALCULAR LA SALIDA ACTUAL Y ERROR

Y= f [(- 0,5* 1) + (- 1 * 1) + (- 2,5 * 1)]

Y = f [- 4] ≤ 0

Y = 0

Error = Salida deseada – salida calculada

1 – 0 = 1 hay error

Hay que seguir iterando………………

LA RED SE ESTABILIZÓ CON LOS PESOS w_3 = - 2,5 w_1 = 2 w_2 = 1,5

REDES NEURALES

EJEMPLOS DE REDES NEURALES - PERCEPTRON

PRIMER PASO: INICIALIZAR LOS PESOS

$w_3 = -2,5$ $w_1 = 2$ $w_2 = 1,5$

SEGUNDO PASO: PRESENTAR ENTRADAS

$x_3 = 1$ (umbral) $x_1 = 0$ $x_2 = 0$ salida deseada: 0

TERCER PASO: CALCULAR LA SALIDA ACTUAL Y ERROR

$Y = f\,[\,(-2,5 * 1) + (2 * 0) + (1,5 * 0)]$
$Y = f\,[-2,5\,] \leq 0$
$Y = 0$

Error = Salida deseada – salida calculada
0 – 0 = 0 no hay error

REDES NEURALES

EJEMPLOS DE REDES NEURALES - PERCEPTRON

PRIMER PASO: INICIALIZAR LOS PESOS

$w_3 = -2,5$ $w_1 = 2$ $w_2 = 1,5$

SEGUNDO PASO: PRESENTAR ENTRADAS

$x_3 = 1$ (umbral) $x_1 = 1$ $x_2 = 0$ salida deseada: 0

TERCER PASO: CALCULAR LA SALIDA ACTUAL Y ERROR

$Y = f\,[(-2,5 * 1) + (2 * 1) + (1,5 * 0)]$
$Y = f\,[-0,5] \leq 0$
$Y = 0$

Error = Salida deseada – salida calculada
0 – 0 = 0 no hay error

REDES NEURALES

EJEMPLOS DE REDES NEURALES - PERCEPTRON

PRIMER PASO: INICIALIZAR LOS PESOS

w_3 = - 2,5 w_1 = 2 w_2 = 1,5

SEGUNDO PASO: PRESENTAR ENTRADAS

x_3= 1 (umbral) x_1= 1 x_2 = 1 salida deseada: 1

TERCER PASO: CALCULAR LA SALIDA ACTUAL Y ERROR

Y= f [(- 2,5* 1) + (2 * 1) + (1,5 * 1)]
Y = f [1] > 0
Y = 1

Error = Salida deseada – salida calculada
1 – 1 = 0 no hay error

REDES NEURALES

EJEMPLOS DE REDES NEURALES - PERCEPTRON

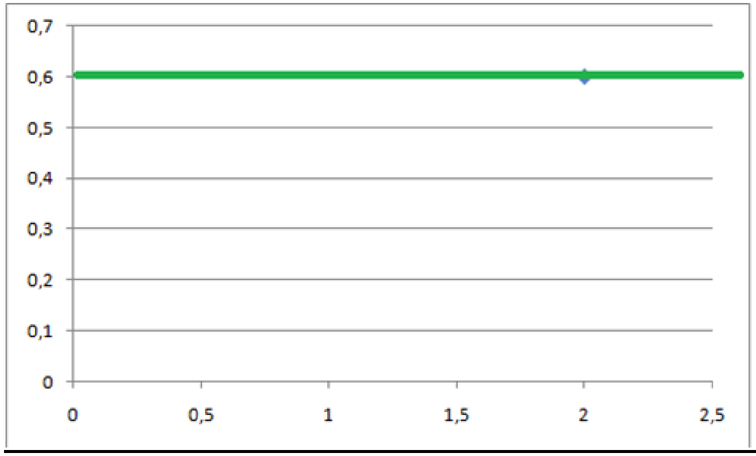
EVOLUCIÓN

Pesos iniciales:

$w_3 = 1{,}5$; $w_1 = 0$; $w_2 = -2{,}5$

$W_1\ X_1 + W_2\ X_2 + W_3 = 0$
$0\ X_1 - 2{,}5\ X_2 + 1{,}5 = 0$

Despejando $X_2 = (-1{,}5)/-2{,}5$
$X_2 = 0{,}6$

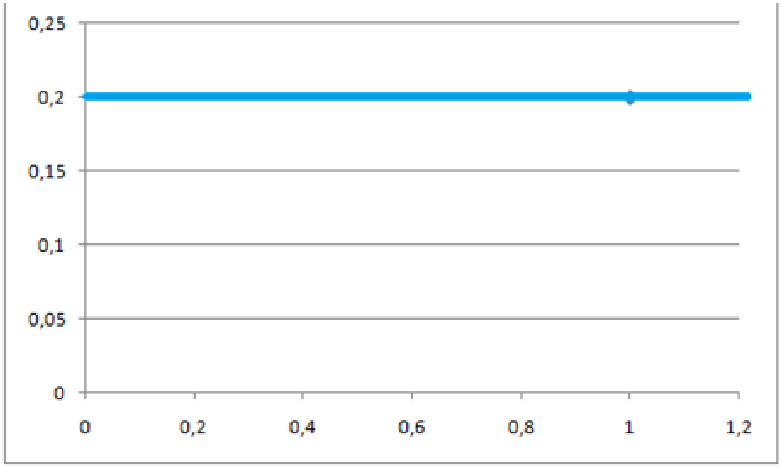


1° Pesos modificados

$w_3 = 0{,}5$; $w_1 = 0$; $w_2 = -2{,}5$

$W_1\ X_1 + W_2\ X_2 + W_3 = 0$
$0\ X_1 - 2{,}5\ X_2 + 0{,}5 = 0$

Despejando $X_2 = (-0{,}5/-2{,}5)$
$X_2 = 0{,}2$



REDES NEURALES

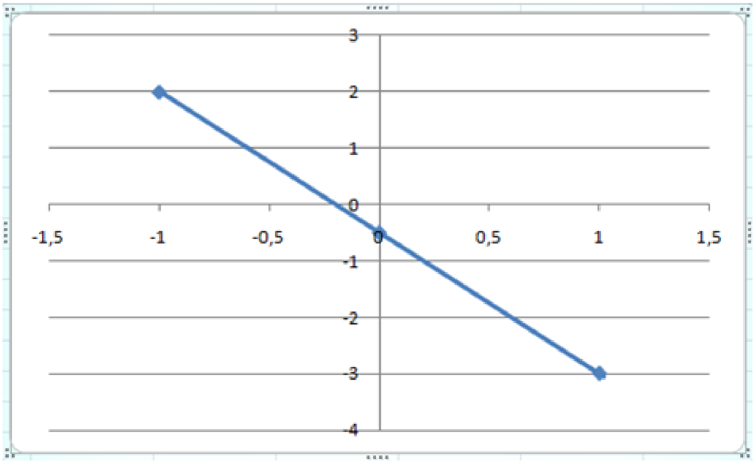
EJEMPLOS DE REDES NEURALES - PERCEPTRON

2° Pesos modificados

$w_3 = -0,5$; $w_1 = -1$; $w_2 = -2,5$

$W_1\ X_1 + W_2\ X_2 + W_3 = 0$
$-1\ X_1 - 2,5\ X_2 - 0,5 = 0$

Despejando $X_2 = (0,5 + X_1)/-2,5$

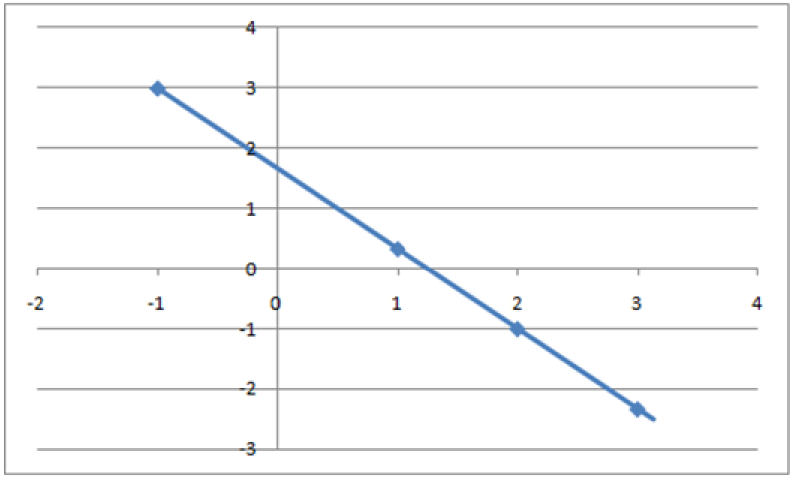


PESOS QUE ESTABILIZAN LA RED PERCEPTRON

$w_3 = -2,5$; $w_1 = 2$; $w_2 = 1,5$

$W_1\ X_1 + W_2\ X_2 + W_3 = 0$
$2X_1 + 1,5\ X_2 - 2,5 = 0$

Despejando $X_2 = (2,5 - 2\ X_1)/1,5$



REDES NEURALES

EJEMPLOS DE REDES NEURALES – RED ADALINE

EJERCICIO 5

Dados los siguientes dos patrones de entrada con salida -1 y 1 respectivamente desarrollar los pasos del algoritmo a fin de determinar si el error cuadrático medio es menor a 0.4.

```
0 0 x      x 0 0
0 x 0      0 x 0
x 0 0      0 0 x
  |          |
  V          V
0  0 1     1 0 0
0  1 0     0 1 0
1 0  0     0 0 1
  |          |
  V          V
Salida = - 1    Salida = 1
```

Estas salidas están fijadas de esta manera para señalar las dos salidas diferentes de la red.

Pasos del algoritmo:

1) Fijar error medio cuadrático aceptable 0.4

2) Fijar valor de $\alpha = 1$

3) Asignar valores aleatorios a los pesos:
   w1 = 0.5
   w2 =0.5
   w3 = - 1
   w4 = 1.5
   w5 = 0.5
   w6 = 1.5
   w7 = -1
   w8 = 1.5
   w9 = 0.5

REDES NEURALES

EJEMPLOS DE REDES NEURALES – RED ADALINE

4) Presentar vector de entrada:
0 0 1 0 1 0 1 0 0

5) Obtener la salida lineal (función rampa) de la red:

$$Sk = \sum_{j=0}^{N} wj\ xkj$$

S1= 0 (0.5) + 0 (0.5) + 1 (-1) + 0 (1.5) + 1 (0.5) + 0 (1.5) + 1 (-1) + 0 (1.5) + 0 (0.5) = -1.5

Calcular el error ε1 = Sd – Sr = (-1 – (-1.5)) = 0.5

6) Actualizar pesos:

$w(t+1) = w_{1(t)} + \alpha \,.\, \varepsilon_1 \,.\, x_1$
w1(t+1) = 0.5 + 1[0.5] 0 = 0.5
w2(t+1) = 0.5 + 1[0.5] 0 = 0.5
w3(t+1) = -1 + 1[0.5] 1 = -0.5
w4(t+1) = 1.5 1[0.5] 0 = 1.5
w5(t+1) = 0.5 + 1[0.5] 1 = 1
w6(t+1) = 1.5 + 1[0.5] 0 = 1.5
w7(t+1) = -1 + 1[0.5] 1 = -0.5
w8(t+1) = 1.5 + 1[0.5] 0 = 1.5
w9(t+1) = 0.5 + 1[0.5] 0 = 0.5

Con los pesos modificados probamos con la siguiente entrada

REDES NEURALES

EJEMPLOS DE REDES NEURALES – RED ADALINE

Con los pesos modificados probamos con la siguiente entrada

1) Entrada 1 0 0 0 1 0 0 0 1

2) Obtener la salida lineal (función rampa)

$$Sk = \sum_{j=0}^{N} wj\ xkj$$

S2= 1 (0.5) + 0 (0.5) + 0 (-0.5) + 0 (1.5) + 1 (1) + 0 (1.5) + 0 (-0.5) + 0 (1.5) + 1 (0.5) = 2

Calcular el error ε1 = Sd – Sr = (1 – 2) = -1 Hay error, se deben actualizar los pesos, pero antes verificar si no está dentro del error cuadrático medio aceptado.

Calcular el error cuadrático medio $< \varepsilon^2 > = 1/2L \sum^{L}_{K=1} \varepsilon^2_k$

$< \varepsilon^2 > = 1 / 2 (2) ((0.5)^2 + (-1)^2) = ¼ (0.25 +1) = 1.25/4= 0.3125$

Está dentro del error cuadrático medio aceptado, la red está estable.