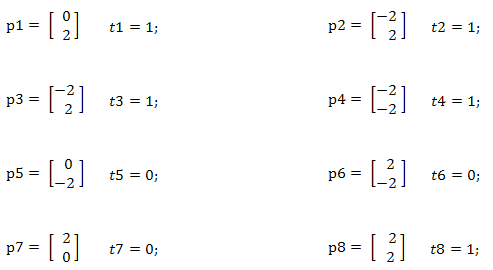
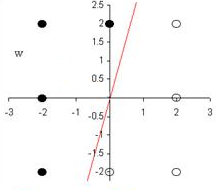
1. Usando la red neuronal del Perceptron, clasificar la siguiente información:



* 1. Trace la frontera de decisión

Se traza sobre la gráfica una línea que separa los patrones que arrojan la salida 1 (puntos negros) de los patrones cuya salida deseada sea 0 (puntos blancos).



* 1. Encuentre la matriz de pesos W y el umbral de activación b
  2. Valide el comportamiento del Perceptron con la información de entrada

De la gráfica se obtiene el vector de pesos W= [-2,1]

Dado que la frontera de decisión atraviesa el origen, el umbral de activación es cero b= 0

Primer par de entrada/salida

P1= [0 2] t1 = 1

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((-2) (0) +(1) (2) +0)

a = hardlim (2) =1

Paso 2. Se calcula el error:

e= t1 –a

e=1 - 1

e=0

Segundo par de entrada/salida

P2 = [-2 2] t2 = 1

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((-2) (-2) +(1) (2) +0)

a = hardlim (6) =1

Paso 2. Se calcula el error:

e= t2 –a

e=1 - 1

e=0

Tercer par de entrada/salida

P3= [-2 2] t3 = 1

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((-2) (-2) +(1) (0) +0)

a = hardlim (4) =1

Paso 2. Se calcula el error:

e= t3 –a

e=1 - 1

e=0

Cuarto par de entrada/salida

P4= [-2 -2] t4 = 1

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((-2) (-2) +(1) (-2) +0)

a = hardlim (2) =1

Paso 2. Se calcula el error:

e= t4 –a

e=1 - 1

e=0

Quinto par de entrada/salida

P5= [0 -2] t5 = 0

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((-2) (0) +(1) (-2) +0)

a = hardlim (-2) =0

Paso 2. Se calcula el error:

e= t5 –a

e=0 - 0

e=0

Sexto par de entrada/salida

P6= [2 -2] t6 = 0

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((-2) (2) +(1) (-2) +0)

a = hardlim (-6) =0

Paso 2. Se calcula el error:

e= t6 –a

e=0 - 0

e=0

Séptimo par de entrada/salida

P7= [2 0] t7 = 0

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((-2) (2) +(1) (0) +0)

a = hardlim (-4) =0

Paso 2. Se calcula el error:

e= t7 –a

e=0 - 0

e=0

Octavo par de entrada/salida

P8= [2 2] t8= 1

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((-2) (2) +(1) (2) +0)

a = hardlim (-2) =0

Paso 2. Se calcula el error:

e= t8 –a

e=0 - 0

e=0

* 1. Conclusión: se concluye que el problema es linealmente separable y este puede ser solucionado por un Perceptron simple, el cual introduce una sola frontera de decisión separando el espacio de los patrones en dos regiones

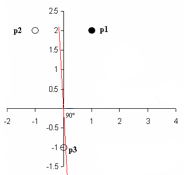
1. Usando la red neuronal del Perceptron, clasificar la siguiente información:

P1= [1, 2] t1 = 1

P2= [-1, 2] t2 = 0

P3= [0, -1] t3 = 0

* 1. Trace la frontera de decisión



* 1. Encuentre la matriz de pesos W y el umbral de activación b

W = [1 -8]

b = 0

a =

Primer par de entrada/salida

P1= [1 2] t1 = 1

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((1) (1) +(-0.8) (2) +0)

a = hardlim (-0.6) =0

Paso 2. Se calcula el error:

e= t1 –a

e=1 - 0

e=1

Paso 3. Usando la regla de aprendizaje del Perceptron, se modifica el vector de pesos:

Wnuevo = [1 -0.8] + (1) [1 2]

Wnuevo = [1 -0.8] + [1 2] = [1+1 -0.8+2]

Wnuevo = [2 1.2]

Paso 4. Para el umbral

bnuevo = b + e

bnuevo = 0 + 1 = 1

Segundo par de entrada/salida

P2= [-1 2] t2 = 0

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((2) (-1) +(1.2) (2) +1)

a = hardlim (1.4) =1

Paso 2. Se calcula el error:

e= t2 –a

e=0 - 1

e=-1

Paso 3. Usando la regla de aprendizaje del Perceptron, se modifica el vector de pesos:

Wnuevo = [2 1.2] + (-1) [-1 2]

Wnuevo = [2 1.2] + [1 -2]

Wnuevo = [3 -0.8]

Paso 4. Para el umbral

bnuevo = b + e

bnuevo = 1 - 1 = 0

Tercer par de entrada/salida

P1= [0 -1] t3 = 0

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((3) (0) +(-0.8) (-1) +0)

a = hardlim (0.8) =1

Paso 2. Se calcula el error:

e= t3 –a

e=0 - 1

e=-1

Paso 3. Usando la regla de aprendizaje del Perceptron, se modifica el vector de pesos:

Wnuevo = [3 -0.8] + (-1) [0 -1]

Wnuevo = [3 -0.8] + [0 1]

Wnuevo = [3 0.2]

Paso 4. Para el umbral

bnuevo = b + e

bnuevo = 0 - 1 = -1

* 1. Valide el comportamiento del Perceptron con la información de entrada

Primer par de entrada/salida

P1= [1 2] t1 = 1

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((3) (1) +(0.2) (2) +0)

a = hardlim (3.4) =1

Paso 2. Se calcula el error:

e= t1 –a

e=1 - 1

e=0

Segundo par de entrada/salida

P1= [-1 2] t2 = 0

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((3) (-1) +(0.2) (2) +0)

a = hardlim (-2.6) =0

Paso 2. Se calcula el error:

e= t2 –a

e=0 - 0

e=0

Tercer par de entrada/salida

P1= [0 -1] t1 = 0

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((3) (0) +(0.2) (-1) +0)

a = hardlim (-0.2) =0

Paso 2. Se calcula el error:

e= t3 –a

e=0 - 0

e=0

Por lo tanto, los valores finales de W y b son:

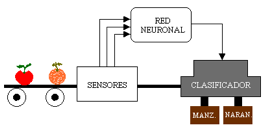
W = [3 0.2]

b = 0

* 1. Conclusión

se concluye que el problema es linealmente separable y este puede ser solucionado por un Perceptron simple, Y como la frontera de decisión pasa por el origen, se deduce que el valor del umbral b es cero

1. Diseñe una red neuronal que clasifique dos frutas:



Las entradas para el Perceptron tienen la siguiente forma:

P: [forma, textura, peso]

Los valores para cada parámetro son:

Forma:

1, si la fruta es redonda

-1, si la fruta es elíptica

Textura:

1, su la superficie de la fruta es suave.

-1, si la superficie es rugosa.

Peso:

1, 1 libra.

-1, < 1 libra

**VECTORES DE ENTRADA**

**Naranja:**

P1= [1, -1, -1] t1 = 0

**Manzana:**

P2= [1, 1, -1] t2 = 1

b = 0.5

* 1. Calcule W y b aplicando el algoritmo de entrenamiento del Perceptron

Primera época

Primer par de entrada/salida (primera iteración)

P1= [1 -1 -1] t1=0

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((0.5) (1) +(-1) (-1) +(0.5) (-1) +0.5)

a = hardlim (2.1) =1

Paso 2. Se calcula el error:

e= t1 –a

e=0 - 1

e=-1

Paso 3. Usando la regla de aprendizaje del Perceptron, se modifica el vector de pesos:

Wnuevo = [0.5 -1 -0.5] + (-1) [1 -1 -1]

Wnuevo = [0.5 -1 -0.5] + [-1 1 1]

Wnuevo = [-0.5 0 0.5]

Paso 4. Para el umbral

bnuevo = b + e

bnuevo = 0.5 -1 = -0.5

Segundo par de entrada/salida (segunda iteración)

P2= [1 1 -1] t2 =1

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((-0.5) (1) +(0) (1) +(0.5) (-1) -0.5)

a = hardlim (-1.5) =0

Paso 2. Se calcula el error:

e= t2 –a

e=1 - 0

e=1

Paso 3. Usando la regla de aprendizaje del perceptrón, se modifica el vector de pesos:

Wnuevo = [-0.5 0 0.5] + (1) [1 1 -1]

Wnuevo = [-0.5 0 0.5] + [1 1 -1]

Wnuevo = [0.5 1 -0.5]

Paso 4. Para el umbral

bnuevo = b + e

bnuevo = -0.5 +1 = 0.5

Segunda época

Primer par de entrada/salida (Primera iteración)

P1= [1 -1 -1] t1=0

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((0.5) (1) +(1) (-1) +(-0.5) (-1) +0.5)

a = hardlim (0.5) =1

Paso 2. Se calcula el error:

e= t1 –a

e=0 - 1

e=-1

Paso 3. Usando la regla de aprendizaje del perceptrón, se modifica el vector de pesos:

Wnuevo = [0.5 1 -0.5] + (-1) [1 -1 -1]

Wnuevo = [0.5 1 -0.5] + [-1 1 1]

Wnuevo = [-0.5 2 0.5]

Paso 4. Para el umbral

bnuevo = b + e

bnuevo = 0.5 -1 = -0.5

Segundo par de entrada/salida (segunda iteración)

P1= [1 1 -1] t2 =1

Paso 1. Se calcula la salida utilizando la función de activación propia de la red:

a = hardlim (Wp + b)

a = hardlim ((-0.5) (1) +(2) (1) +(0.5) (-1) -0.5)

a = hardlim (0.5) =1

Paso 2. Se calcula el error:

e= t2 –a

e=1 - 1

e=0

* 1. Conclusión:

De acuerdo a la regla de aprendizaje del Perceptron se ha alcanzado un mínimo por lo que se obtienen valores estables para la matriz de pesos y el umbral