



**REDES
NEURONALES
ADAPTABLES**

**Unidad 3: Redes neuronales artificiales
y modelos de entrenamiento**

SubTemas

3.2 Perceptron simple

- Arquitectura
- Regla delta
- Multi Layer Perceptrón

3.3 Redes Neuronales Adaptables

- En línea
- Fuera de línea

2

Tema 3.2 Perceptrón simple

■ La neurona de McCulloch-Pitts (1/3)

El modelo de una neurona real se debe a Warren McCulloch (neurofisiólogo) y Walter Pitts (matemático) de 1943. Este se originó como un modelo simple. Introdujo los conceptos de pesos que implicaban una influencia continua de la sinapsis en la neurona; así mismo, la idea de pesos negativos y su influencia inhibidora.

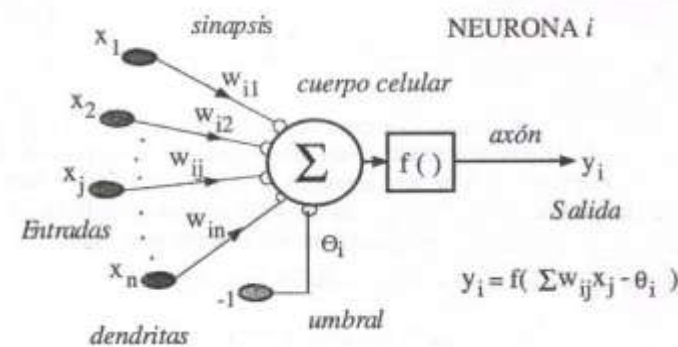
Después de esto hubo varias contribuciones al modelo, principalmente en la manera de 'entrenar' la red.

3

Tema 3.2 Perceptrón simple

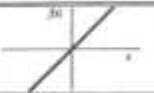
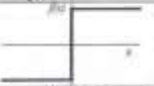
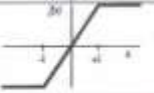

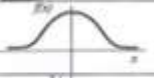

■ La neurona de McCulloch-Pitts (2/3)

Notar que w_{ij} y θ_i son parámetros escalares ajustables de la neurona. Éstos, son ajustados por medio de una regla de aprendizaje.



4

La neurona de McCulloch-Pitts (3/3)

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq l \\ +1, & \text{si } x > l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(ax + \varphi)$	$[-1, +1]$	

5

Tema 3.2 Perceptrón simple

■ Inicios del entrenamiento de RNA

Contribuciones importantes se deben a el Biólogo Donald **Hebb** en 1949 en aspectos de bases biológicas; Bernard **Widrow** en 1960 propuso una regla de aprendizaje importante conocida como la [regla Widrow-Hoff](#) (regla delta)

Una prueba de convergencia importante para el modelo McCulloch-Pitts fue desarrollada y publicada por Frank **Rosenblatt** en 1958 y 1962 en su obra 'Perceptrons'. Este teorema se llama el teorema de convergencia del perceptron.

6

Tema 3.2 Perceptrón simple

■ Limitantes del perceptron

Este trabajo impulsó mucho el estudio de las RNAs en sus inicios porque además se presentó como un método de reconocimiento de patrones con logros importantes.

Una de las características de esta red es existe un grupo importante de problemas que no puede resolver. Por esta razón este tipo de redes está casi en desuso ahora pero no se debe menospreciar la contribución de Rosenblatt. **La gran limitante del perceptron es que sólo se puede usar para problemas de clasificación que sean separables linealmente en la salida.**

7

Tema 3.2 Perceptrón simple

■ Características del perceptron

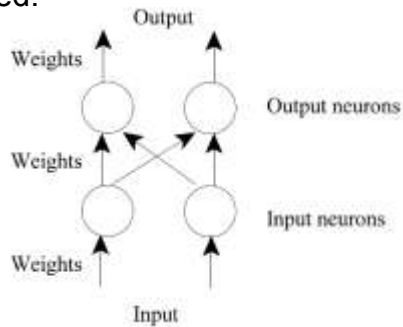
- Sistema capaz de realizar tareas de clasificación de forma automática.
- A partir de un número de ejemplos conocidos, el sistema determina la ecuación del plano discriminante (Aprendizaje supervisado).
- Puede usarse como neurona dentro de otro tipo de red de neurona artificial. Es un modelo unidireccional compuesto por dos capas de neuronas.
- Las neuronas de entrada son discretas en el entrenamiento, y la función de activación de las de salida es de tipo escalón.

8

Tema 3.2 Perceptrón simple

■ Familiarizándose con las RNA (1/5)

Dibuja una red que este constituida por 4 neuronas artificiales, las primeras dos neuronas reciben la señal de entrada a la red, y las restantes generan la salida de la red.



9

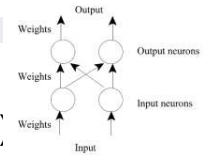
Tema 3.2 Perceptrón simple

■ Familiarizándose con las RNA (2/5)

Hay pesos (weights) asignados a cada flecha, los cuales representan el flujo de la información. Estos pesos se multiplican por los valores que van a lo largo de la flecha, para darle más fuerza o disminuir la intensidad de la señal que transmiten.

Las neuronas de esta red solo suman sus entradas.

Ya que las neuronas de entrada, solo tienen 1 sola entrada. La salida es la entrada multiplicada por la W . ¿Qué pasa si sus pesos son negativos? ¿Qué pasa si sus pesos son cero?



10

M.C. ENRIQUE MARTINEZ PEÑA

Tema 3.2 Perceptrón simple

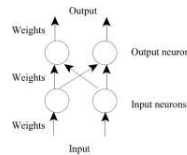
■ Familiarizándose con las RNA (3/5)

Las neuronas en la capa de salida reciben las salidas de ambas neuronas multiplicadas por sus respectivos pesos, y los suman. Ellas dan una salida la cual es multiplicada por otro peso.

Ahora

Asígnales a todos los pesos un valor de uno. Esto significa que la información fluirá sin afectaciones. Calcula las salidas de la red para los siguientes pares de entrada:

(1, 1)	(1, 0)	(0, 1)	(0, 0)	(-1, 1)	(-1, -1)
(2, 2)	(1, 1)	(1, 1)	(0, 0)	(0, 0)	(-2, -2)



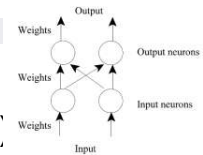
Tema 3.2 Perceptrón simple

■ Familiarizándose con las RNA (4/5)

Muy bien! Ahora supón que queremos una red como la que tenemos, de tal manera que las salidas deben ser las entradas en orden inverso.

Modifica las w necesarias.

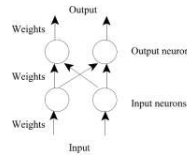
Ejemplo $(0.3, 0.7) \rightarrow (0.7, 0.3)$.



Tema 3.2 Perceptrón simple

■ Familiarizándose con las RNA (5/5)

Excelentísimo, otra red fácil sería aquella donde la salida fuese el doble que las entradas.



13

Clave: 1-umbral-2

Tema 3.2 Perceptrón simple

■ Agregando umbral (thresholds) a la neurona

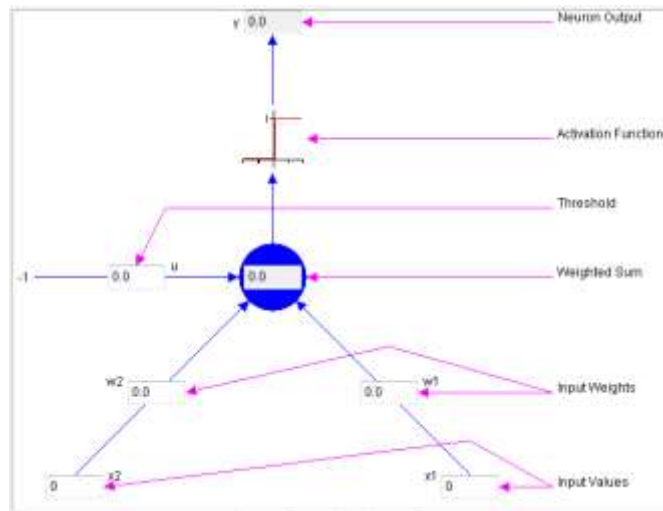
Esto es, si la salida previa de la neurona (suma ponderada de las entradas) es mayor o igual que el umbral de la neurona, la salida de la neurona será **uno**, y **cero** en caso contrario. Esto es, función de activación en escalón.

Además considera que la red solo recibirá entradas con unos y ceros, Ajusta los pesos y umbrales de manera que la salida se comporte como el operador AND, y OR.

14

<http://lcn.epfl.ch/tutorial/english/aneuron/html/index.html>

Tema 3.2 Perceptr3n simple

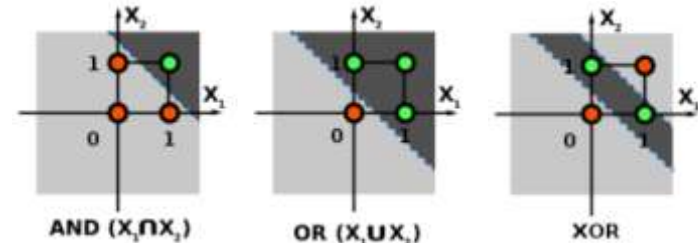


15

Tema 3.2 Perceptr3n simple

■ Comentarios finales

Intenta encontrar los pesos y umbral necesarios para ejecutar la funci3n X-OR.



La funci3n XOR no puede ser aprendida por un 3nico perceptr3n puesto que requiere al menos de dos l3neas para separar las clases (0 y 1). Debe utilizarse al menos una capa adicional de perceptrones para permitir su aprendizaje. MLP MultiLayerPercep.

Tema 3.2 Perceptrón simple

■ Ecuación del hiperplano

La ecuación que separa las dos clases se obtiene así:

$$x_1 w_1 + x_2 w_2 - \theta = 0$$

Despejamos para x_2

$$x_2 = \frac{\theta - w_1 x_1}{w_2}$$

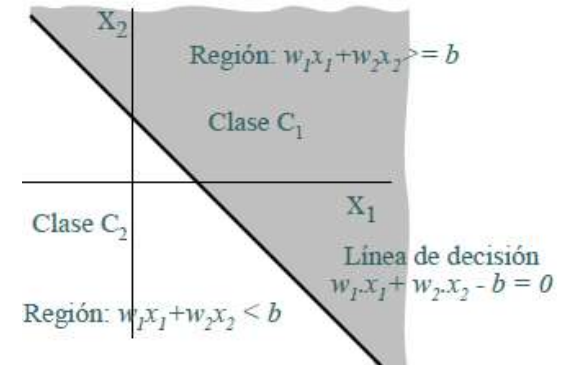
$$x_2 = -\frac{w_1}{w_2} x_1 + \frac{\theta}{w_2}$$

17

Tema 3.2 Perceptrón simple

■ Ecuación del hiperplano

El perceptrón simple separa dos clases con un hiperplano.



18

Tema 3.2 Perceptrón simple

■ Aprendizaje supervisado

Los patrones de entrenamiento están constituidos por pares de valores: (vector de entrada y salida deseada). La regla Delta utiliza la diferencia entre la salida producida para cada patrón y la deseada (E).

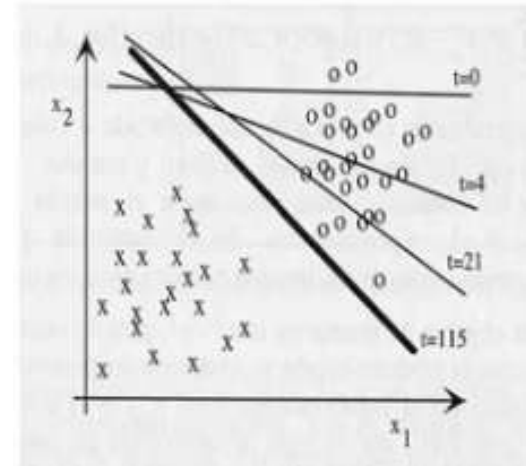
La regla busca el conjunto de pesos que minimiza la función de error, esto se realiza mediante un proceso iterativo donde se presentan los patrones y se van modificando los parámetros de la red mediante la regla del gradiente descendente.

$$w(j)' = w(j) + \alpha(\delta - y)$$

19

Tema 3.2 Perceptrón simple

■ Evolución de las reglas de decisión



20

M.C. ENRIQUE MARTINEZ PEÑA

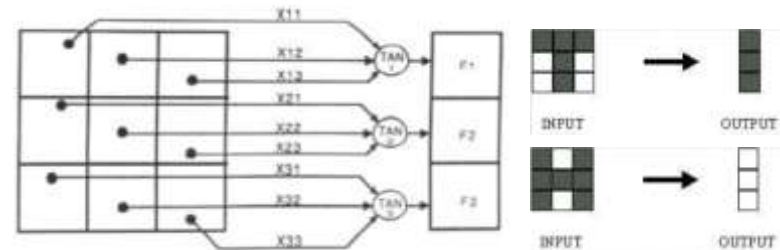
Learning		Input			Initial		Output						Final	
Threshold	Rate	Sensor values		Desired output	Weights		Calculated		Sum	Network	Error	Correction	Weights	
TH	LR	X1	X2	Z	w1	w2	C1	C2	S	N	E	R	W1	W2
							X1 x w1	X2 x w2	C1+C2	(S>TH,1,0)	Z-N	LR x E	R+w1	R+w2
0.5	0.2	0	0	0	0.1	0.3	0	0	0	0	0	0	0.1	0.3
0.5	0.2	0	1	1	0.1	0.3	0	0.3	0.3	0	1	0.2	0.3	0.5
0.5	0.2	1	0	1	0.3	0.5	0.3	0	0.3	0	1	0.2	0.5	0.7
0.5	0.2	1	1	1	0.5	0.7	0.5	0.7	1.2	1	0	0	0.5	0.7
0.5	0.2	0	0	0	0.5	0.7	0	0	0	0	0	0	0.5	0.7
0.5	0.2	0	1	1	0.5	0.7	0	0.7	0.7	1	0	0	0.5	0.7
0.5	0.2	1	0	1	0.5	0.7	0.5	0	0.5	0	1	0.2	0.7	0.9
0.5	0.2	1	1	1	0.7	0.9	0.7	0.9	1.6	1	0	0	0.7	0.9
0.5	0.2	0	0	0	0.7	0.9	0	0	0	0	0	0	0.7	0.9
0.5	0.2	0	1	1	0.7	0.9	0	0.9	0.9	1	0	0	0.7	0.9
0.5	0.2	1	0	1	0.7	0.9	0.7	0	0.7	1	0	0	0.7	0.9
0.5	0.2	1	1	1	0.7	0.9	0.7	0.9	1.6	1	0	0	0.7	0.9

21

Tema 3.2 Perceptrón simple

■ Reconocimiento de patrones

Se puede implementar usando una red neuronal hacia adelante, durante su entrenamiento, se asocia la salida de la red con patrones de entrada.

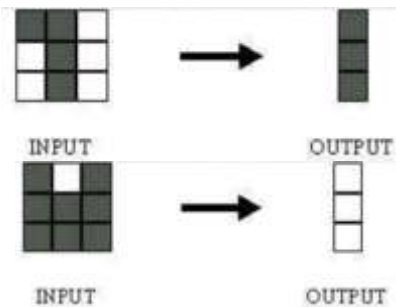


22

Tema 3.2 Perceptrón simple

■ Reconocimiento de patrones

El poder de las redes neuronales se materializa cuando se tienen patrones parecidos a los usados en el entrenamiento y las RNA tienen la capacidad de seguir discriminando.



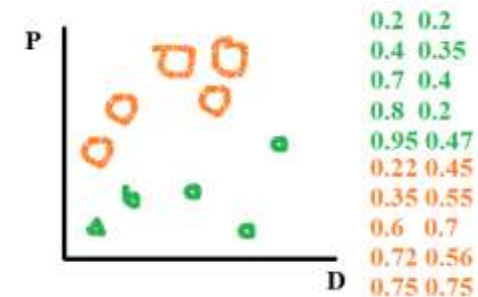
23

Tema 3.2 Perceptrón simple

■ Toolbox de Matlab para NN

El toolbox tiene diferentes redes, vamos a iniciar con la que conocemos, perceptron.

Para ello, vamos a generar un discriminador de limones – naranjas. Basados en el siguiente gráfico.



24

Tema 3.2 Perceptrón simple

■ Toolbox de Matlab para NN

Lo primero es dar de alta la red, diciendo que son dos entradas (indicando rango [min max] de cada entrada) y solo se requiere una neurona:

```
mipercep = newp([0 1; 0 1],1);
```

Ahora hay que definir la matriz P de vectores de entrada y el vector de salida deseado de la neurona:

```
P = [0.2 0.4 0.7 0.8 0.95 0.22 0.35 0.6 0.72 0.75;
     0.2 0.35 0.4 0.2 0.47 0.45 0.55 0.7 0.56 0.75];
T = [0 0 0 0 0 1 1 1 1 1];
```

25

Tema 3.2 Perceptrón simple

■ Toolbox de Matlab para NN

Lo siguiente es hacer el entrenamiento indicándole el número de épocas a entrenar:

```
mipercep.trainParam.epochs = 20;
mipercep = train(mipercep,P,T);
```

Listo! Ahora vamos a probar la salida del vector original P, con el que realizamos el entrenamiento:

```
Y = sim(mipercep,P)
```

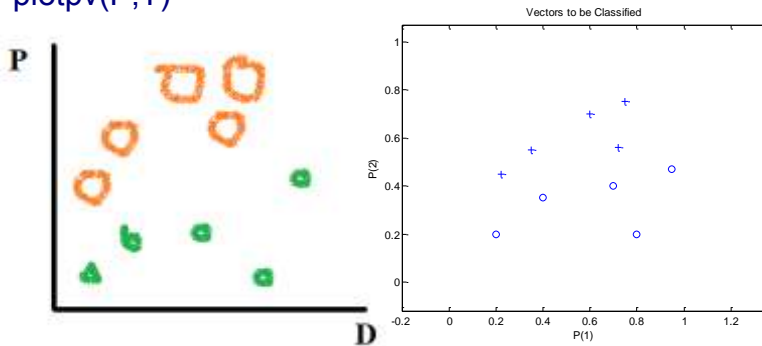
Debe quedar igualito a T, para estar seguros, vamos a buscar un punto en la gráfica donde no estuvieron los puntos de entrenamiento (0.4,0.8) y (0.9,0.2) ²⁶

Tema 3.2 Perceptrón simple

■ Toolbox de Matlab para NN

Graficando las entradas

`plotpv(P,T)`



27

Tema 3.2 Perceptrón simple

■ Toolbox de Matlab para NN

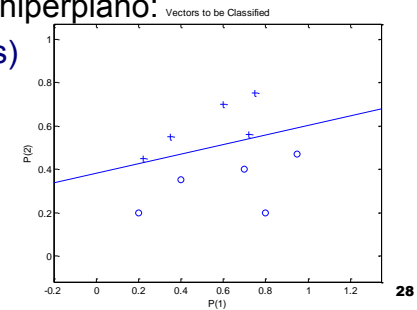
Para obtener los pesos y los umbrales:

`Pesos = mipercep.IW{1,1};`

`Umbrales = mipercep.b{1};`

Agregando la recta del hiperplano:

`plotpc(Pesos, Umbrales)`



28



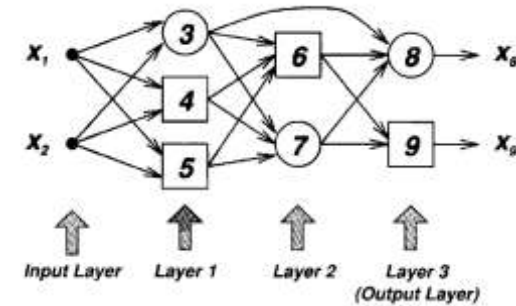

REDES NEURONALES ADAPTABLES

**Unidad 3: Redes neuronales artificiales
y modelos de entrenamiento**

Tema 3.2 Redes neuronales adaptables

■ Introducción

Una red adaptable se compone de nodos conectados, en donde cada uno ejecuta una función estática de sus entradas para generar una salida. Un cuadrado representa un nodo adaptable, un círculo un nodo fijo.



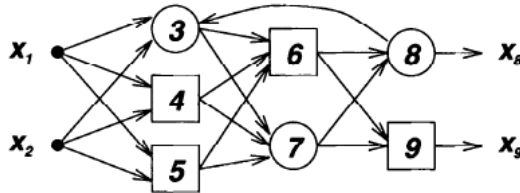
30

Tema 3.2 Redes neuronales adaptables

■ Tipos de redes adaptables

- hacia delante (feedforward)
- recurrente (recurrent)

Si existe una retroalimentación que forme una trayectoria circular en la red, ésta es recurrente, como el ejemplo de abajo.



31

Tema 3.2 Redes neuronales adaptables

■ The learning rule

Especifica como los parámetros de la red deberían ser actualizados para minimizar la medida de error prescrita, la cual es una expresión matemática que mide la diferencia entre la salida de la red actual y la salida deseada.

En otras palabras, una red adaptable se usa para la identificación de sistemas, y nuestro trabajo es encontrar una arquitectura de red adecuada y un set de parámetros que modelen lo mejor posible un sistema desconocido que se describe por un conjunto de datos de entrada-salida.

32



Tema 3.2 Redes neuronales adaptables

■ Steepest descent method

La regla de aprendizaje básica para la red adaptable es el método de paso descendente, en el cual el vector gradiente es derivado por la sucesiva invocación de la regla de la cadena.

En 1986, Rumelhart et al. Usaron el procedimiento de Werbos para encontrar el gradiente en una red neuronal multicapa, le llamaron regla de aprendizaje retropropagada (backpropagation).