08-08-19

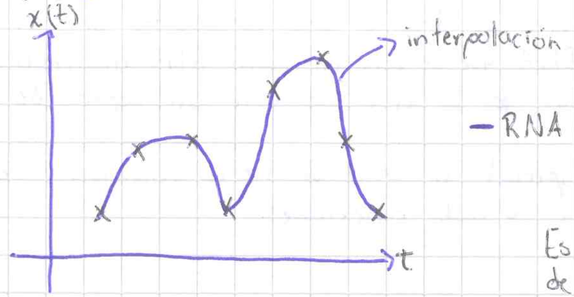
Neural Networks.

Una Red Neuronal Artificial (RNA) funciona de 2 modos:

a) Regresor: La RNA ajusta datos

b) Clasificador: La RNA separa datos

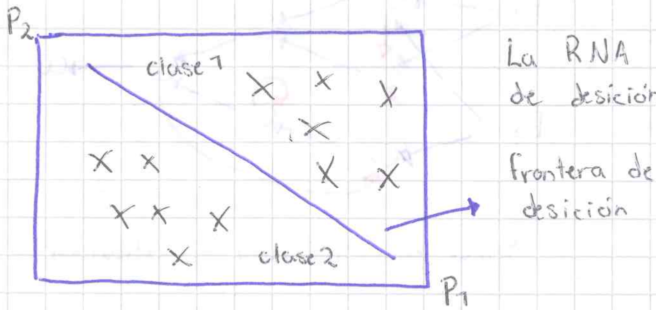
Regresor



* Interpolación: Unir 2 puntos conocidos mediante un modelo dentro del rango conocido del fenómeno.
* Extrapolación: Es pronosticar valores fuera del rango conocido del fenómeno.

Este modo se usa para tareas que requieren series de tiempo.

Clasificador

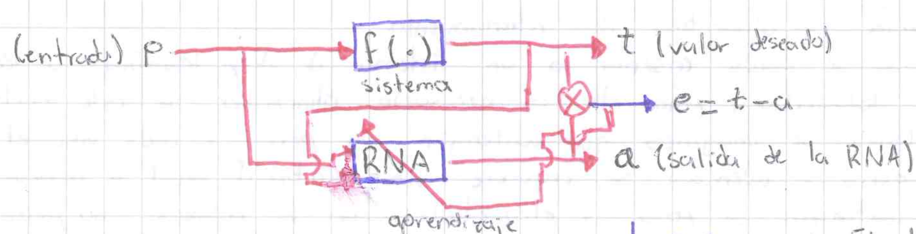


La RNA ajusta la frontera de desición.

Ejemplos modo regresor:

1.- Identificación de sistemas. Dado un sistema $f(t)$ del cuál se desconoce su modelo matemático, pero se tiene un conjunto de datos entrada-salida (p, t) que representan su comportamiento se puede entrenar una RNA que se comporte de manera muy similar, a.

entrada ↙ ↘ salida (target)



IMPORTANTE: El objetivo del aprendizaje es $e \rightarrow 0$

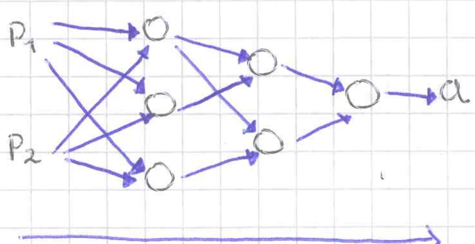
08-08-19

Se dice que una RNA es un aproximador universal, ya que es capaz de ajustarse al comportamiento de una enorme cantidad de sistemas/Fenómenos lineales o no. Esto fue demostrado por primera vez en el teorema de Cybenko.

NOTA: Si bien la RNA es un aproximador universal, funciona ajustando un sistema a la vez.

2.- Pronóstico de series de tiempo. Dado una serie de tiempo que representa un fenómeno físico, existe un tipo particular de RNA que se conocen como RNA recurrentes que son capaces de pronosticar valores futuros de la serie, es decir son capaces de extrapolar.
Las RNA recurrentes usan una arquitectura especial en donde se tienen conexiones en sentido contrario de la dirección entrada-salida ó auto conexiones.

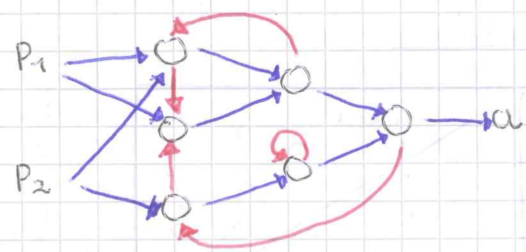
RNA feed-forward



Flujo
propagación hacia adelante

Sirve para interpolar

RNA recurrente



Sirve para interpolar y extrapolar

Tipos de aprendizaje.

Supervisado
Este requiere que se provea de un conjunto de entrenamiento muy bien estructurado y etiquetado para el algoritmo de entrenamiento.
Por ejemplo, se le puede enseñar a un sistema de deep learning a reconocer un perro en una fotografía, para ello es necesario alimentar a este sistema con miles o millones de imágenes que contengan un perro y cada una etiquetada con la palabra "perro", además se requiere una cantidad igual o mayor de imágenes que no contengan un perro y etiquetadas con la palabra "no perro".
Una vez que el sistema es entrenado, se puede dar como entradas nuevas imágenes, y el sistema dirá si es o no un perro.
El aprendizaje supervisado es por mucho la técnica más usada en los sistemas de IA, siendo usada para casi el 95% de las aplicaciones prácticas.
Sin embargo, un problema es que este tipo de aprendizaje requiere cantidades enormes de datos etiquetados.

16-08-19

Reforzado

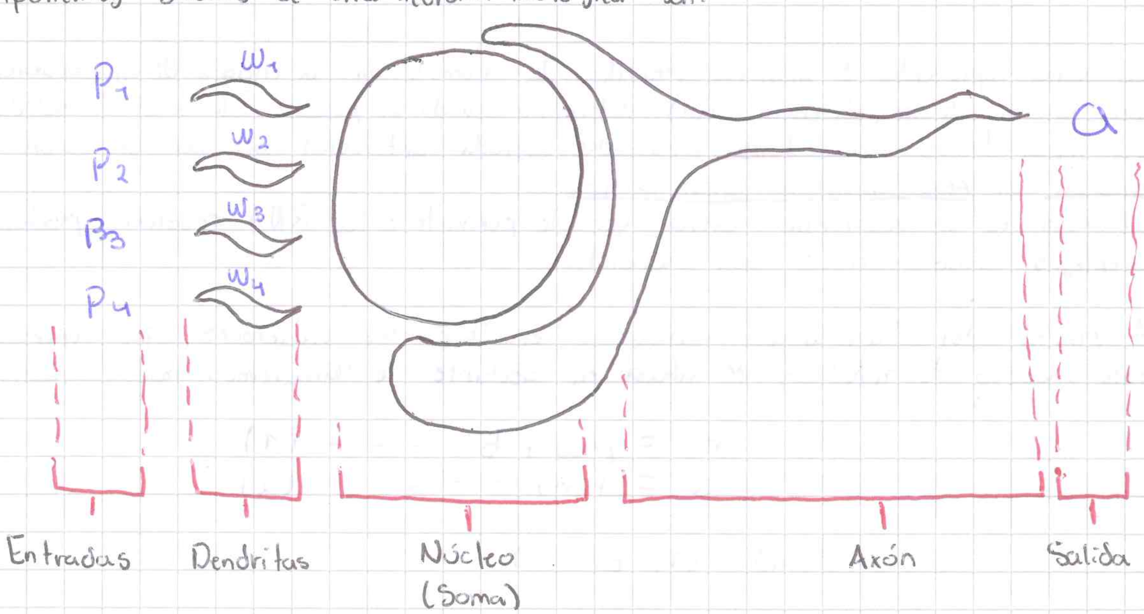
En escencia, este aprendizaje se obtiene através de la práctica de prueba y error. En vez de entrenar al sistema con ejemplos, se establece una pérdida donde el sistema trata de encontrar la solución por sí misma, y si tiene éxito recibe una recompensa.
Un ejemplo es la forma como se entrena a los perros.
El aprendizaje reforzado es especialmente poderoso para entrenar sistemas de IA que jueguen juegos.
El problema con el aprendizaje reforzado es que requiere un enorme número de corridas de práctica antes de que el algoritmo sea exitoso. Por esta razón, su uso principal es en los videojuegos o en tareas que puedan ser simuladas en una computadora a alta velocidad.

Neurona biológica.

Dado que una RNA es un modelo bioinspirado, veamos los componentes básicos de una neurona biológica.
Los sistemas biológicos son tan complejos que se han convertido en una excelente fuente de inspiración para diseñar sistemas artificiales que emulen algunas de sus características, entre ellas se encuentran las siguientes:

1) No siempre requieren de un módulo de referencia (aprendizaje supervisado).
2) Se adaptan fácilmente a nuevos ambientes
3) Se desempeñan exitosamente ante incertidumbres
4) Pueden procesar información de diversas fuentes en forma simultánea.

Los componentes básicos de una neurona biológica son:



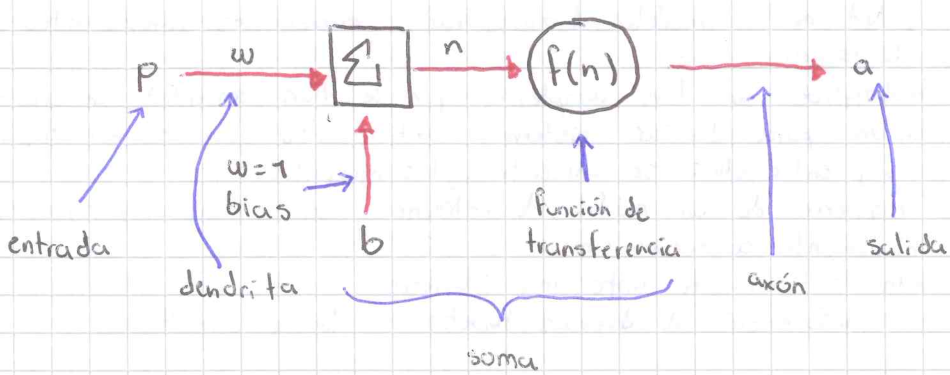
16-08-19

Las dendritas conectan las señales de entrada con el núcleo de la neurona. Cada una de éstas conexiones tiene asociado un peso sináptico. Los pesos sinápticos se usan para indicar el nivel de importancia de cada una de las entradas para un proceso en particular (por ejemplo: aprender a multiplicar 2 números).
El soma se encarga de acumular energía y determina cuándo se debe generar una señal de salida.
El axón es el encargado de transmitir la señal de salida hacia otras neuronas o al mundo exterior.

Componentes que forman a una RNA.

ARQUITECTURAS

a) Modelo de 1 neurona con 1 entrada



La suma representa de manera sencilla la acumulación de energía de una neurona biológica.
La función de transferencia f(n) determina cuándo se produce la señal de salida a.
La señal b se llama bias y es una entrada artificial que por definición tendrá siempre un peso sináptico igual a uno.
El bias es un parámetro extra que le permite a la RNA resolver problemas más complejos como veremos más adelante.

A partir del diagrama, vamos a escribir las ecuaciones que representan a éste modelo de RNA y de ahora en adelante le llamaremos modelo matemático.

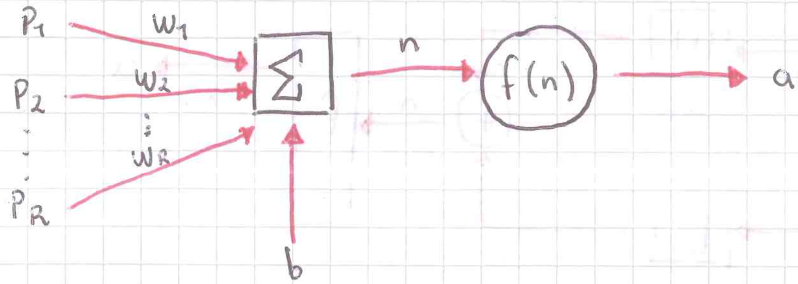
$$n = pw + b \quad --- (1)$$
$$a = f(n) \quad ----- (2)$$

Sustituyendo (1) en (2)

$$a = f(pw + b) \quad --- (3)$$

16-08-19

b) Una neurona con múltiples entradas



donde R es la dimensión del vector de entrada

Modelo matemático:

$$n = \sum_{i=1}^{R} w_i p_i + b \quad \text{- - - (4)}$$

$$a = f(n) \quad \text{- - - - - - - (5)}$$

Este modelo también se puede representar de forma matricial

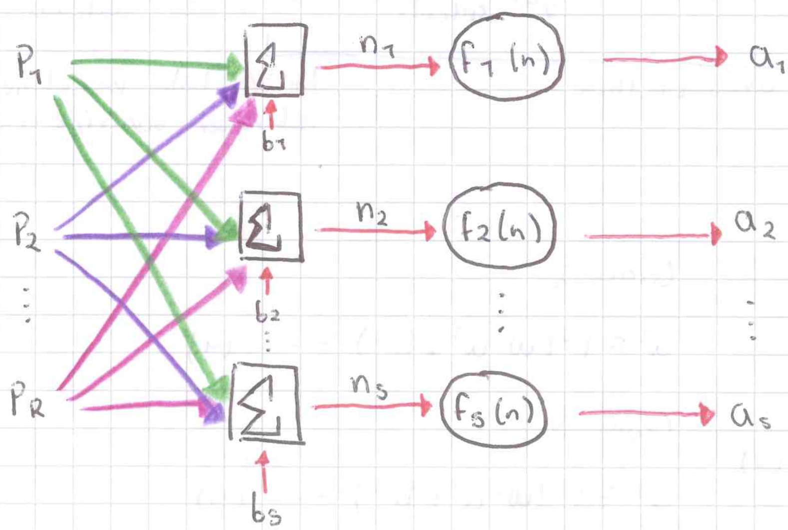
$$\underset{1\times1}{a} = f(\underbrace{\underset{1\times R}{W} \cdot \underset{R\times1}{p}} + \underset{1\times1}{b}) \quad \text{- - - (6)}$$

Esto es así para que los vectores se puedan multiplicar

Observa: Para cada neurona solo hay un bias y en este caso W es un vector fila.

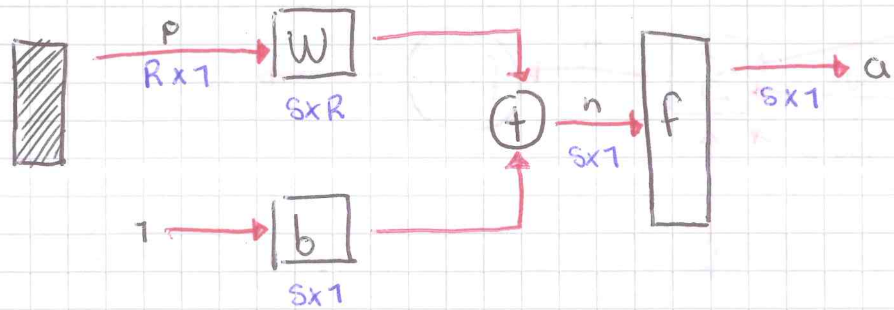
c) Múltiples entradas con múltiples neuronas.

* Arquitectura no matricial (completamente conectada)



- Capa de neuronas: Es un arreglo de neuronas en paralelo.

* Arquitectura matricial



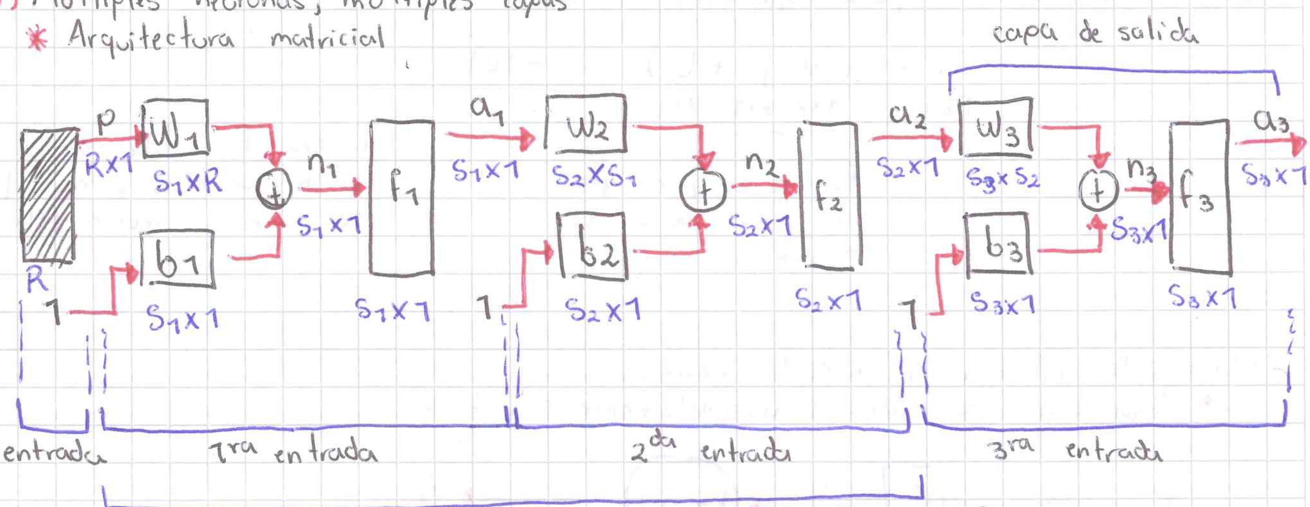
Modelo matemático matricial:

$$\underset{S\times 1}{a} = f(\underset{S\times R}{W}\ \underset{R\times 1}{p} + \underset{S\times 1}{b}) \quad --- \quad (7)$$

donde S es el número de neuronas en la capa

d) Múltiples neuronas, múltiples capas

* Arquitectura matricial



NOTA: Todos los subíndices deben ser superíndices

Modelo matemático

Capa 1

$a^1 = f^1(n^1) \quad ---- \quad (1)$

$n^1 = W^1 \cdot p + b^1 \quad -- \quad (2)$

Sustituyendo

$a^1 = f^1(W^1 \cdot p + b^1) \quad -- \quad (3)$

Capa 2

$a^2 = f^2(W^2 a^1 + b^2) \quad --- \quad (4)$

Capa 3

$a^3 = f^3(W^3 a^2 + b^3) \quad --- \quad (5)$

22-08-19

A este tipo de RNA se le conoce como MultiLayer Perceptron (MLP).
En el curso consideraremos una arquitectura completamente conectada, lo que quiere decir que todas las salidas de una capa se convierten en entradas de todas las neuronas de la siguiente capa.
Cada señal de MLP tiene asignado un peso sináptico.
Un MLP se llama feed forward (propagación hacia adelante) si sus conexiones van exclusivamente en el sentido de la entrada hacia la salida, por lo que no tiene conexiones hacia atrás, hacia arriba o hacia abajo.

Para describir un MLP completamente conectado y feed forward, usaremos 2 vectores:

1) Un vector que indica en sus elementos el número de capas, la dimensión de la entrada y el número de neuronas en cada capa.

$$V_1 : [\, R \; \overbrace{S^1 \quad S^2 \cdots S^M}^{\#\text{ de capas}} \,]$$

dimensión del vector de entrada

M es el # de capas

Ejemplo:

$$V_1 : [\, 3 \; \overbrace{\underbrace{2 \quad 4}_{\text{ocultas}} \quad 1}^{3\text{ capas}} \,]$$

R=3

neuronas en la capa

2) Otro vector que indica el tipo de función de transferencia de cada capa del MLP, considerando que se usa un solo tipo de función de transferencia en cada capa.

Ejemplo: Considere que se usa la siguiente lista de f. de t.

1. purelin(·)
2. logsig(·)
3. tansig(·)

$$V_2 : [\, 2 \quad 3 \quad 1 \,]$$

capa oculta 1 usa una f. de t. logsig(·)

Ejemplo: Dados los vectores V_1 y V_2, obtenga el modelo matemático del MLP.

$V_1 : [\,5 \quad 3 \quad 2 \quad 4 \quad 1\,]$

$V_2 : [\,3 \quad 2 \quad 3 \quad 1\,]$

$$\underset{3\times1}{a^1} = \text{tansig}(\underset{3\times5}{W^1} \cdot \underset{5\times1}{p} + \underset{3\times1}{b^1})$$

$$\underset{2\times1}{a^2} = \text{logsig}(\underset{2\times3}{W^2} \cdot \underset{3\times1}{a^1} + \underset{2\times1}{b^2})$$

$$\underset{4\times1}{a^3} = \text{tansig}(\underset{4\times2}{W^3} \cdot \underset{2\times1}{a^2} + \underset{4\times1}{b^3})$$

$$\underset{1\times1}{a^4} = \text{purelin}(\underset{1\times4}{W^4} \cdot \underset{4\times1}{a^3} + \underset{1\times1}{b^4})$$

23-08-19

¿Cómo seleccionar el tamaño de la arquitectura de un MLP?

1) La dimensión del vector de entrada (R) será igual a la dimensión de la serie de tiempo a aproximar o al número de rasgos que representan el objeto a clasificar.
2) El número de neuronas a la salida también está definido por el problema a resolver en el caso cuando se tiene un dataset supervisado (p, t).
datos de entrada → p; ← t neuronas de salida

3) El tipo de función de transferencia a usar depende de la codificación del dataset y de que tipo de tarea se realizará: regresor o clasificador.
4) En cuanto al número de capas ocultas y el número de neuronas en cada capa es un problema abierto, sin embargo es común [usar entre 1 y 3 capas ocultas con menos de 30 neuronas (aproximadamente) c/u para problemas de regresión.] [Para problemas más complejos, como lo es procesar una imagen como entrada, es necesario usar arquitecturas profundas con 3 o más capas ocultas y decenas o centenas de neuronas.] [MLP Vanilla]

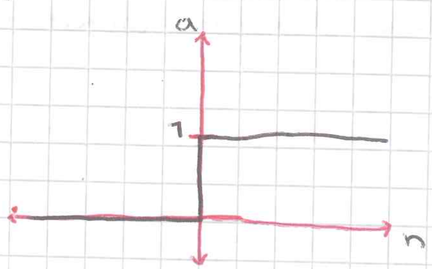
[ ] Sólo se usan números
MLP clásico

[ ] Deep Learning

Funciones de transferencia.

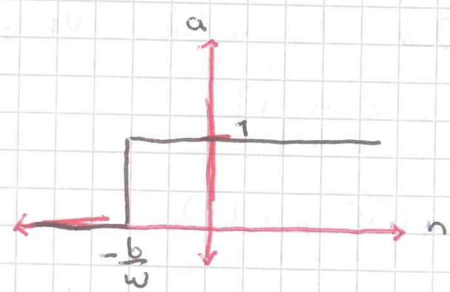
Una función de transferencia puede ser lineal o no lineal y se elige para satisfacer alguna especificación del problema que la RNA está tratando de aprender.

a) Función Hard Limit
Esta función se usa para clasificar ya que genera un <u>0</u> a la salida si el valor de "n" es menor a 0 y genera un <u>1</u> si el valor de "n" es mayor o igual a 0.



$a = \text{hardlim}(n)$
$a = \text{hardlim}(\underbrace{wp})$

neurona sin bias



$a = \text{hardlim}(n)$
$a = \text{hardlim}(\underbrace{wp+b})$

neurona con bias

El bias sirve para desplazar la función de transferencia del origen y así poder resolver una gama más grande de problemas.

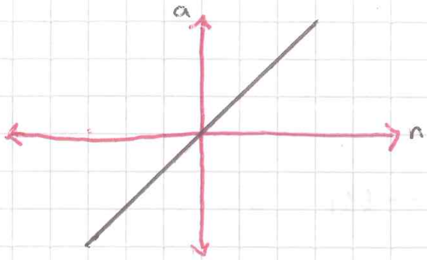
Existe una variante de esta función que se llama Symmetrical Hard Limit (Hardlims) que es similar a Hardlim pero sus valores de salida son +1 y −1.

b) Función lineal

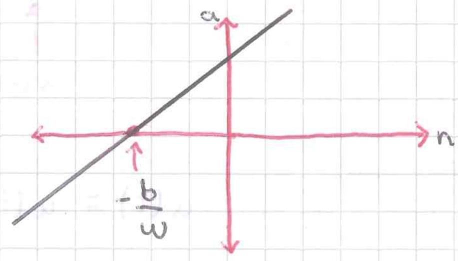
Esta función se usa comunmente para problemas de tipo regresor y su función es

$$a = n \quad --- \quad (1)$$

Por ejemplo, este tipo de función de transferencia se usa en la RNA ADALINE.



$a = purelin(wp)$



$a = purelin(wp+b)$

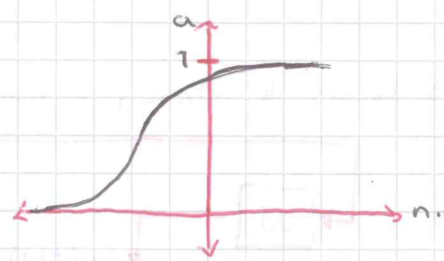
c) Función Logsigmoid

Esta función toma los valores de entrada y los reduce a un rango de salida entre 0 y 1 de acuerdo a la siguiente expresión:

$$a = \frac{1}{1+e^{-n}} \quad --- \quad (2)$$

Ésta es una función no lineal que se usa comunmente en las capas ocultas de un MLP cuyo aprendizaje se realizará usando backpropagation. Esto es debido a que es una función diferenciable. Otra función similar es tansig(·).

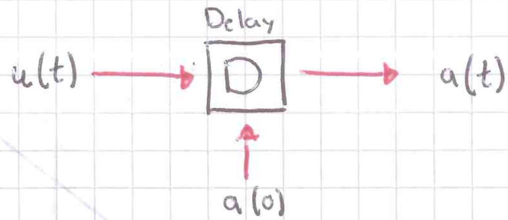
a logsig (wp)



a logsig (wp+b)

Redes Neuronales Recurrentes

Para construir este tipo de RNA se requiere de un bloque llamado "retraso en tiempo" que se muestra a continuación:



Cuya ecuación es:

$$a(t) = u(t-1) \quad --- (1)$$

Este bloque se encarga de generar un retraso en la señal de entrada $u(t)$.

Ejemplo: Dada la señal $u(t)$ obtenga los valores correspondientes de $a(t)$ para $t \geq 0$.

$$u(t) = \begin{array}{cccc} t=0 & t=1 & t=2 & t=3 \\ \end{array}$$
$$u(t) = [1.1 \quad 3.2 \quad 0.5 \quad 7.8]$$
$$a(0) = 4.3$$

Usando la ecuación (1) tenemos que

$a(0) = u(0-1)$

Como no tenemos valores definidos para $u(t)$ en $t<0$, usamos la condición inicial.

$a(0) = 4.3$

$a(1) = u(0) = 1.1$

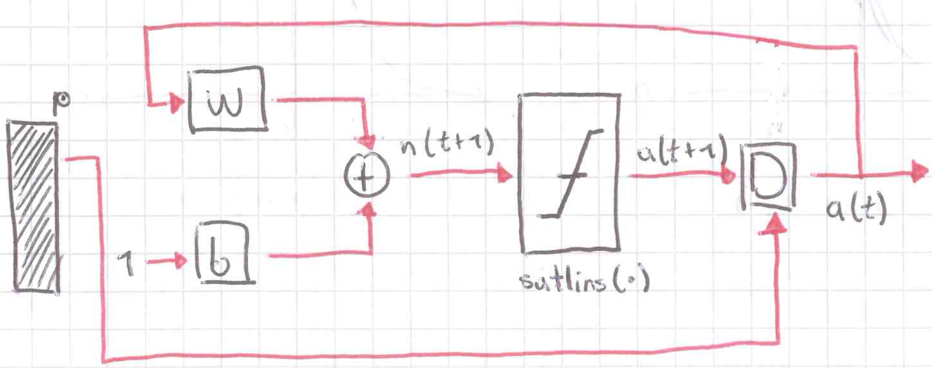
$a(2) = u(1) = 3.2$

$a(3) = u(2) = 0.5$

$a(4) = u(3) = 7.8$

$a(5)$ en adelante ya no tenemos valores definidos

Usando los bloques de retraso, se pueden diseñar RNA recurrentes, como la siguiente:



Modelo matemático:

$a(0) = p$

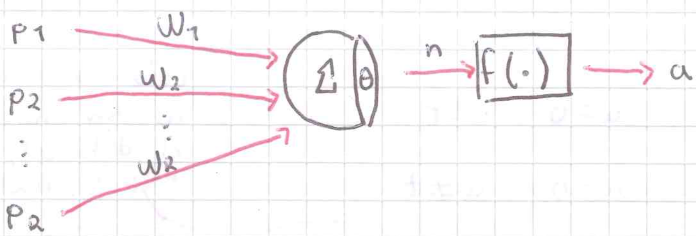
$a(t+1) = \text{satlins}(w \cdot a(t) + b)$

$n(t+1) = w \cdot a(t) + b$

26-09-19

Célula de McCulloch-Pitts

Esta célula modela por primera vez a una neurona biológica como un dispositivo de dos estados: apagado (0) y encendido (1). La arquitectura es:



El modelo matemático es:

$$n = \sum_{i=1}^{R} w_i p_i = w \cdot p$$

$$a = f(n)$$

donde $f(\cdot)$ es:

$$a = \begin{cases} 1, & \text{si } n > \theta \\ 0, & \text{en otro caso} \end{cases}$$

En esta RNA el valor θ se usa para activar la señal de salida de la célula.

- Propagación hacia adelante: Es aplicar como entrada cada uno de los datos del conjunto de entrenamiento y obtener su valor de salida correspondiente. Los datos se pueden usar en cualquier orden.
- Conjunto de entrenamiento: Dado un dataset (100), se toma un subconjunto (70%-80% del dataset) y se usa para llevar a cabo el aprendizaje de la RNA.
- Iteración: Propagar hacia adelante uno de los datos del conjunto de entrenamiento.
- Época (Epoch): Propagar hacia adelante todos los datos del conjunto de entrenamiento.
- Aprendizaje supervisado: Es cambiar el valor de los parámetros internos de la RNA para minimizar el error entre la salida de la RNA (a) y el valor deseado (t).
- Regla de aprendizaje: Realizar un ajuste automático de los parámetros internos de la RNA.

En esta célula, la regla de aprendizaje es mediante prueba y error. El aprendizaje termina cuando todos los datos del conjunto de entrenamiento estén bien clasificados, es decir $a = t$.
Esta célula mostró por primera vez que una RNA es capaz de realizar cómputo universal, lo que implica que debe poder comportarse como las 3 compuertas básicas: NOT, AND y OR.

Ejercicio: Realizar el aprendizaje de una célula de McCulloch-Pitts para las tres compuertas básicas.
dataset: (p, t)

a) NOT

p	t
0	1
1	0

b) AND

p_1	p_2	t
0	0	0
0	1	0
1	0	0
1	1	1

c) OR

p_1	p_2	t
0	0	0
0	1	1
1	0	1
1	1	1

Solución:

a) Paso 1: Obtener dataset

Paso 2: Modelo matemático

$\boxed{n = w \cdot p}$, $a = f(n)$

Paso 3: Aprendizaje. Inicio con valores aleatorios (num. enteros)

$w = 1$, $\theta = 1$

Época 1.

Iteración 1 (d_1)

$n = 1(1) = 1$ $n \ngtr \theta$ $a = 0$ $a = t$

Iteración 2 (d_2)

$n = (1)(0) = 0$ $n \ngtr \theta$ $a = 0$ $a \neq t$

Esto es debido a que no hay más datos en el data set, y a que no todos los resultados fueron correctos.

Como falló se cambian valores de w y θ y se sigue con la Época 2. Se realizan cambios en los valores de los parámetros de la célula (w, θ) a lo que llamaremos que todos los datos están bien clasificados (criterio de finalización).

Época 2. $w = -1$ $\theta = -1$

Iteración 3 (d_1)

$n = (-1)(0) = 0$ $n > \theta$ $a = 1$ $a = t$

Iteración 4 (d_2)

$n = (-1)(1) = -1$ $n \ngtr \theta$ $a = 0$ $a = t$

Dado que en esta última época todos los datos están bien clasificados con un único conjunto de valores (w, θ) se llega a un aprendizaje exitoso.

b) $n = w_1 \cdot p_1 + w_2 \cdot p_2$

$a = f(n)$

Época 1. $w_1 = w_2 = \theta = -1$

Iteración 1. (d_1)

$n = (-1)(1) + (-1)(1) = -2$ $n \ngtr \theta$ $a \neq t$

Iteración 2. (d_2) $w_1 = w_2 = \theta = 1$

$n = (1)(1) + (1)(0) = 1$ $n \ngtr \theta$ $a = t$

Iteración 3 (d_3)

$n = (1)(0) + (1)(1) = 1$ $n \ngtr \theta$ $a = t$

Iteración 4 (d_4)

$n = (1)(0) + (1)(0) = 2$ $n \ngtr \theta$ $a = t$

Como hubo fallo se realiza otra época.

Época 2

Iteración 5 (d_1)

$n = (1)(1) + (1)(1) = 2$ $n > \theta$ $a = t$

It 6 $d_2 \longrightarrow a = t$

It 7 $d_3 \longrightarrow a = t$

It 8 $d_4 \longrightarrow a = t$

05-09-19

Problema de Clasificación.

En el capítulo 3 se presentan tres tipos de RNA para resolver un sencillo problema de clasificación, estas RNA's son:

1) RNA Feed Forward (perceptron)
2) RNA competitiva (Hamming)
3) RNA que es una memoria asociativa recurrente (Hopfield)

Problema de clasificación.
Un granjero desea construir una máquina para clasificar fruta de acuerdo a su tipo. En particular, desea separar entre manzanas y naranjas. Para ello usará los siguientes 3 sensores:

a) Sensor de forma: La salida de este sensor es 1 si la fruta es redonda y -1 si es elíptica.

b) Sensor de textura: La salida de este sensor es 1 si la superficie de la fruta es suave y -1 si es rugosa.

c) Sensor de peso: La salida será 1 si la fruta pesa más de una libra y -1 si es menor a una libra.

NOTA: En términos de propagación de información es más útil usar una salida binaria con +1 y -1 que usar 0 y 1. Esto es debido a que si se usara 0, al propagar la información, siendo que esta sea un 0, se cancelaría el peso sináptico de la neurona a la que se propaga.

$$p \triangleq \begin{bmatrix} \text{forma} \\ \text{textura} \\ \text{peso} \end{bmatrix}$$

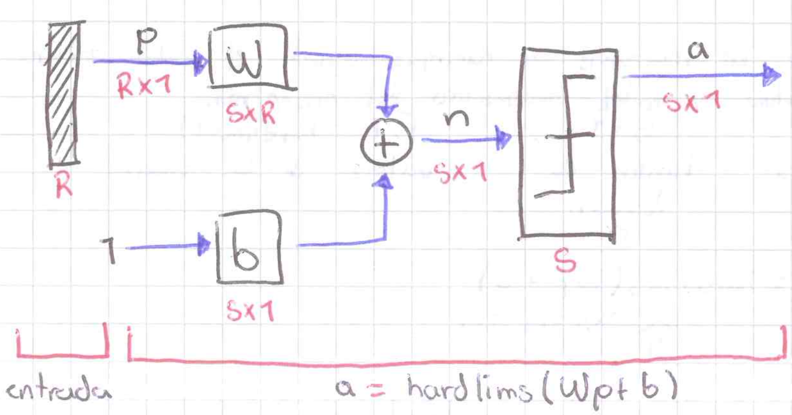
Los vectores prototipo para cada fruta son:

$$P_1 = \begin{bmatrix} +1 \\ -1 \\ -1 \end{bmatrix} \qquad P_2 = \begin{bmatrix} +1 \\ +1 \\ -1 \end{bmatrix}$$

naranja   manzana

a) Perceptrón de R-entradas

Un perceptrón consta de una sola capa y usa como funciones de transferencia hardlim() ó hardlims() y se muestra en la siguiente figura:



Este perceptrón puede clasificar a los vectores de entrada en dos clases, usando la siguiente configuración:

$$n = w_{11} p_1 + w_{12} p_2 + b \quad --- \quad (1)$$

Si se usa $w_{11} = -1$ y $w_{12} = +1$ y se sustituye en (1) se obtiene:

$$n = [-1 \quad +1] \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + b \quad --- \quad (2)$$

A continuación, la ecuación (2) se sustituye en la ecuación de salida del perceptrón:

$$a \quad \text{hardlims}\left([-1 \; +1] \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + b \right) \quad --- \quad (3)$$

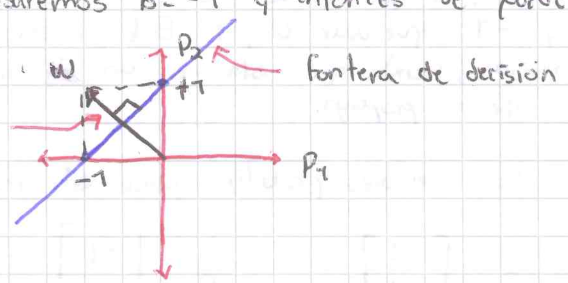
En la ecuación (3) podemos ver que:

i) Si $wp \geq -b$ la salida será $+1$

ii) Si $wp < -b$ la salida será -1

A manera de ejemplo usaremos $b = -1$ y entonces se puede dibujar la frontera de decisión.

Vector de pesos sinápticos



$$w = [-1 \quad +1]$$

El perceptrón solo es capaz de hacer clasificación lineal.

▶ Características del vector de pesos:

1) Parte del origen
2) Es perpendicular a la frontera de decisión
3) Apunta a la clase $+1$
4) Lo importante es su orientación, no su magnitud.

Resolución del problema de las manzanas y naranjas usando el método gráfico del perceptrón.

1.- Obtenemos la arquitectura y modelo matemático del perceptrón.

$R = 3$	$S = 1$	F.T. = hardlims()
sensores	Frontera de desición	valores de los sensores

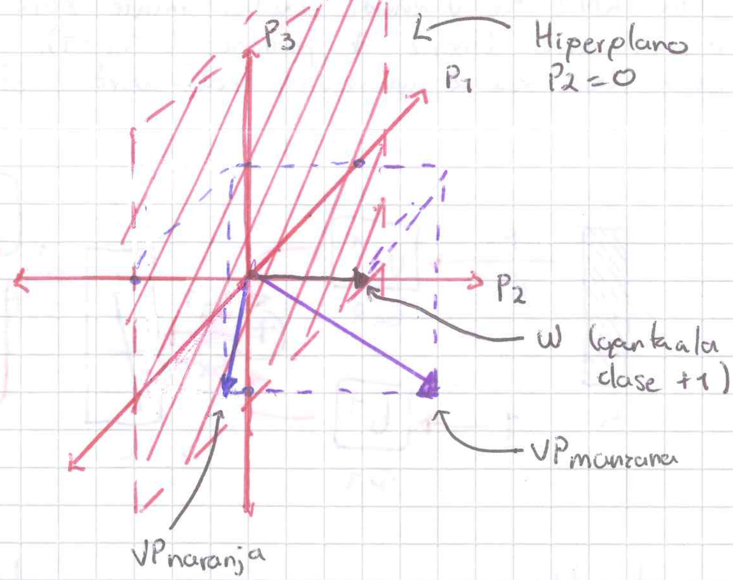
$$\underset{1\times1}{a} = \text{hardlims}(\underset{1\times3}{W}\,\underset{3\times1}{p} + \underset{1\times1}{b})$$

$$\Updownarrow$$

$$a = \text{hardlims}\left([w_1 \, w_2 \, w_3] \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} + b \right)$$

2.- Obtener los valores de W y b mediante el método gráfico (frontera de decisión). Este método grafica ambos vectores prototipo y visualmente se busca una frontera de decisión.

	naranja	manzana
P_1	+1	+1
P_2	-1	+1
P_3	-1	-1



Visualmente se elige una frontera de decisión que separe correctamente a ambas clases y se escribe la ecuación de dicho plano.

$$P_2 = 0 \quad --- \ (1)$$

Para obtener los alores de W y b se representa la ecuación del plano usando la estructura de la señal n del perceptrón:

$$n = [W_1 \ W_2 \ W_3] \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} + b = 0 \quad --- \ (2)$$

Proponiendo valores para que (1) = (2)

$$n = [0 \ 1 \ 0] \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} + 0 = 0$$

$$P_2 = 0 \quad --- \ (3)$$

3.- Se valida el resultado aplicando el dataset a la RNA usando los valores de W y b que se obtuvieron gráficamente.

VPnaranja:

$$a = \text{hardlims}\left([0 \ 1 \ 0] \begin{bmatrix} +1 \\ -1 \\ -1 \end{bmatrix} + 0 \right) = -1$$

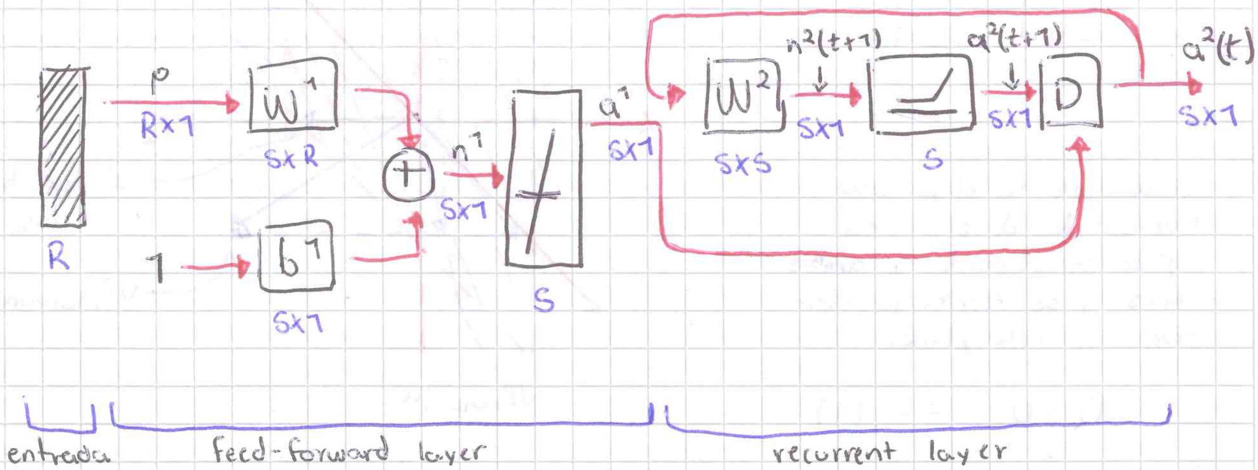
VPmanzana

$$a = \text{hardlims}\left([0 \ 1 \ 0] \begin{bmatrix} +1 \\ +1 \\ -1 \end{bmatrix} + 0 \right) = +1$$

09-09-19

Red Hamming

Esta RNA fue diseñada explícitamente para ser implementada en problemas de reconocimiento binario de patrones (+1, -1).
Esta RNA usa 2 capas: una feedforward y otra recurrente.



$$a^1 = \text{purelin}(\underbrace{W^1 p + b^1}_{n^1})$$

$$a^2(0) = a^1; \quad a^2(t+1) = \text{poslin}(\underbrace{W^2 a^2(t)}_{n^2(t+1)})$$

Observaciones:

* El número de neuronas es el mismo en ambas capas.
* La capa recurrente no tiene bias.
* La recurrencia se lleva acabo usando la señal de salida a^2 como entrada a los pesos W^2.

Capa feed-forward

Esta capa calcula la correlación o producto interno entre cada uno de los vectores prototipo. Con este objetivo, las filas de la matriz de pesos W^1 serán cada uno de los vectores prototipo. Por ejemplo, para el caso de clasificar fruta, la matriz queda como:

$$\underset{S\times R}{W^1} = \begin{bmatrix} P_1^T \\ P_2^T \end{bmatrix} = \underset{2\times 3}{\begin{bmatrix} +1 & -1 & -1 \\ +1 & +1 & -1 \end{bmatrix}} \begin{matrix} \text{naranja} \\ \text{manzana} \end{matrix} \qquad \therefore S = 2$$

NOTA: El número de vectores prototipo es igual al número de neuronas.

El valor del bias b^1 es igual a R, en nuestro ejemplo:

$$\underset{\substack{S\times 1 \\ 2\times 1}}{b^1} = \begin{bmatrix} R \\ R \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

NOTA: Al agregar el valor R, se garantiza que el valor de salida de esta capa no sean valores negativos, lo cual es necesario para el correcto funcionamiento de la capa recurrente.

Capa recurrente

Las neuronas de esta capa se inicializan con las salidas de la capa anterior. En esta capa las neuronas compiten entre ellas para determinar a la ganadora. La matriz de pesos W^2 es cuadrada y tiene 1's en la diagonal principal y un valor $-\varepsilon$ en el resto de la matriz, donde ε es un valor menor a: $\boxed{\frac{1}{S-1} \; ; \; \varepsilon > 0}$

En nuestro ejemplo:

$$W^2 = \begin{bmatrix} 1 & -\varepsilon \\ -\varepsilon & 1 \end{bmatrix}$$

Ejercicio: Clasificación de manzanas y naranjas.

1) Dado W^1 y b^1, se obtiene la salida a^1 para $p = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$

$$a^1 = \begin{pmatrix} +1 & -1 & -1 \\ +1 & +1 & -1 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} + \begin{pmatrix} 3 \\ 3 \end{pmatrix}$$

$$a^1 = \text{purelin}\,(W^1 p + b^1)$$

$$a^1 = \begin{pmatrix} +1 \\ -1 \end{pmatrix} + \begin{pmatrix} 3 \\ 3 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

2) En la capa recurrente se aplica la recurrencia hasta que sólo una de las neuronas sea diferente de cero.
Se obtiene el valor de ε:

$$\frac{1}{S-1} = \frac{1}{2-1} = 1 \rightarrow 0 < \varepsilon < 1$$

usemos $\varepsilon = 0.7$

a) Primera iteración
$t = 0$
$a^2(0) = a^1 = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$

$$a^2(1) = \text{poslin}\left[\begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 2 \end{pmatrix}\right] = \begin{pmatrix} 2.6 \\ 0 \end{pmatrix}$$

Para comprobar que la RNA ya convergió, se hace otra iteración y si da el mismo resultado, se finaliza el proceso.

09-09-19

b) Segunda iteración

$t=1$

$$a^2(2) = \text{poslin}\left[\begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix}\begin{pmatrix} 2.6 \\ 0 \end{pmatrix}\right] = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$$

c) Tercera iteración

$t=2$

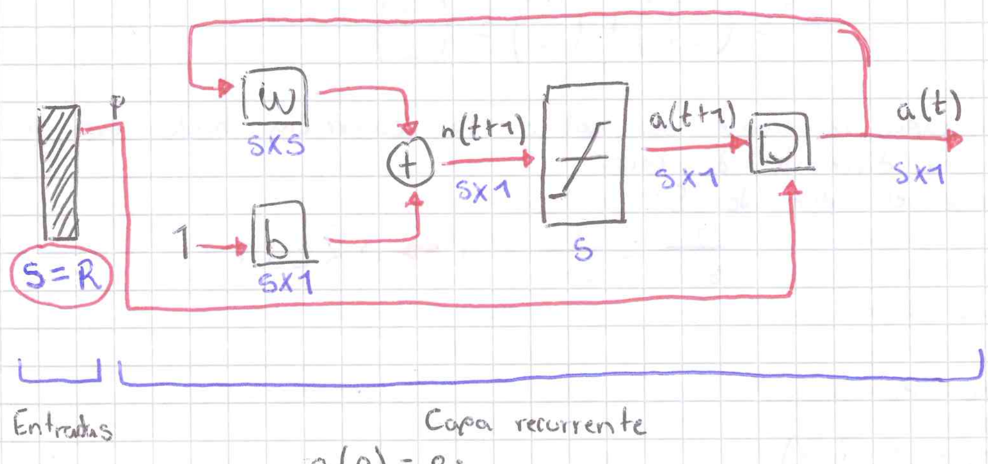
$$a^2(3) = \text{poslin}\left[\begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix}\begin{pmatrix} 3 \\ 0 \end{pmatrix}\right] = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$$

$\therefore$ El vector $p = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$ es una naranja

Red de Hopfield

Esta red es recurrente y con sólo una capa es capaz de realizar las operaciones de la red Hamming, hay que recordar que esta RNA no siempre converge y esto sucede si 2 o más vectores prototipo son equidistantes al vector de entrada. Esto no sucede en la RNA de Hopfield.

Su arquitectura es:



$$a(0) = p;$$
$$a(t+1) = \text{satlins}(W \cdot a(t) + b)$$

Diferencias con respecto a la RNA Hamming

1) La dimensión del vector de entrada (R) es igual al número de neuronas (S).
2) Sólo tiene una capa
3) Usa satlins() como función de transferencia
4) Tiene bias en la capa recurrente
5) Es una memoria asociativa recurrente
6) Siempre converge pero no necesariamente a la respuesta correcta

12-09-19

NOTAS: - Esta RNA realiza iteraciones hasta que converge a uno de los vectores prototipo.
- La forma de obtener los valores W y b se encuentra en el capítulo 21 del libro "Neural Networks".

Ejemplo: Dado el siguiente vector de entrada, use una RNA de Hopfield para clasificarlo como manzana o naranja, $p^T = [-1\ -1\ -1] \rightarrow R=3$

1) Obtener el modelo matemático y el valor de W y b.

$$\underset{3\times1}{a(0)} = \underset{3\times1}{p} \qquad \underset{3\times1}{a(t+1)} = \text{satlins}\,(\underset{3\times3}{W} \cdot \underset{3\times1}{a(t)} + \underset{3\times1}{b})$$

para el caso de manzanas y naranjas

$$W = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} \qquad b = \begin{bmatrix} 0.9 \\ 0 \\ -0.9 \end{bmatrix}$$

2) Se realizan iteraciones hasta que la RNA converja.

- $t=0$

$$\begin{bmatrix} a_1(1) \\ a_2(1) \\ a_3(1) \end{bmatrix} = \text{satlins}\left(\begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 0.9 \\ 0 \\ -0.9 \end{bmatrix} \right)$$

$$= \text{satlins}\left[\begin{pmatrix} -0.2 \\ -1.2 \\ -0.2 \end{pmatrix} + \begin{pmatrix} 0.9 \\ 0 \\ -0.9 \end{pmatrix} \right] = \text{satlins}\left[\begin{pmatrix} 0.7 \\ -1.2 \\ -1.1 \end{pmatrix} \right] = \begin{pmatrix} 0.7 \\ -1 \\ -1 \end{pmatrix}$$

- $t=1$

$$\begin{pmatrix} a_1(2) \\ a_2(2) \\ a_3(2) \end{pmatrix} = \text{satlins}\left[\begin{pmatrix} 0.2 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0.2 \end{pmatrix} \begin{pmatrix} 0.7 \\ -1 \\ -1 \end{pmatrix} + \begin{pmatrix} 0.9 \\ 0 \\ -0.9 \end{pmatrix} \right]$$

$$= \text{satlins}\left[\begin{pmatrix} 1.04 \\ -1.2 \\ -1.1 \end{pmatrix} \right] = \begin{pmatrix} +1 \\ -1 \\ -1 \end{pmatrix}$$

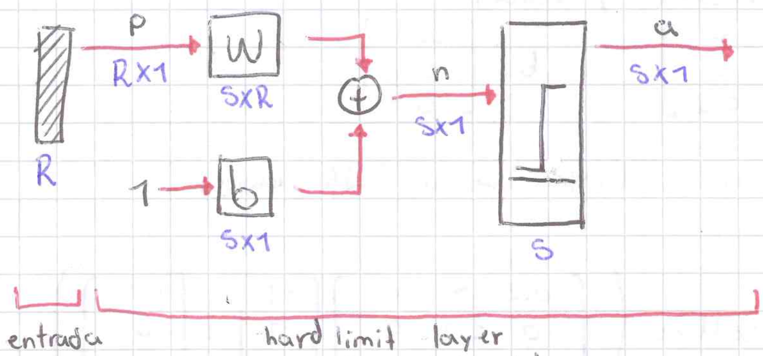
Se finaliza porque la RNA convergió al vector prototipo de la naranja.

Regla de aprendizaje del perceptrón.

Una regla de aprendizaje es un procedimiento que de manera automática modifica los valores de pesos y bias de una RNA con el propósito de aprender una tarea dada.
En el caso del aprendizaje supervisado, se cuenta con un conjunto de ejemplos que indican cual debe ser el comportamiento correcto de la RNA para un caso dado y se representa de la siguiente manera:

$$\{P_1, t_1\}, \{P_2, t_2\}, \{P_3, t_3\}, \cdots, \{P_Q, t_Q\}$$

Arquitectura del perceptrón:



$a = \text{hardlim}(Wp + b)$

Con el objetivo de desarrollar la regla de aprendizaje del perceptrón es útil tener una referencia individual a cada uno de los parámetros de la RNA. Primero consideraremos la matriz de pesos:

$$W = \begin{array}{cl} \begin{bmatrix} w_{11} & w_{12} & w_{13} & \cdots & w_{1R} \\ w_{21} & w_{22} & w_{23} & \cdots & w_{2R} \\ \vdots & \vdots & \vdots & & \vdots \\ w_{S1} & w_{S2} & w_{S3} & \cdots & w_{SR} \end{bmatrix} & \begin{array}{l} \leftarrow \text{neurona } 1 \\ \leftarrow \text{neurona } 2 \\ \\ \leftarrow \text{neurona } S \end{array} \end{array}$$

Se puede definir un vector compuesto de los elementos de la i-ésima fila de W:

$$w_i = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}$$

Ahora se puede representar la matriz de pesos de la siguiente manera:

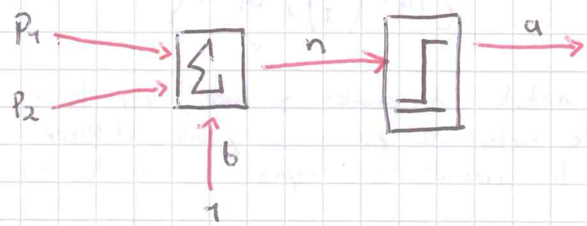
$$W = \begin{bmatrix} {}_1w^T \\ {}_2w^T \\ \vdots \\ {}_Sw^T \end{bmatrix}$$

Esto nos permite escribir el i-ésimo elemento del vector de salida como:

$$a_i = \text{hardlim}({}_iW^T p + b_i)$$

RECUERDA: Cada neurona de la RNA divide el espacio de entrada en 2 regiones.

Caso para 1 neurona, con 2 entradas



$n = W_1 \cdot P_1 + W_2 \cdot P_2 + b$
$a = \text{hardlim}(n)$

Otras representaciones son:
$a = \text{hardlim}(Wp + b)$
$a = \text{hardlim}({}_1W^T p + b)$

ANÁLISIS DE LA FRONTERA DE DECISIÓN

Esta frontera se determina por los valores para las cuales la señal "n" es igual a cero:

$$W_1 \cdot P_1 + W_2 \cdot P_2 + b = 0 \quad --- \ (1)$$

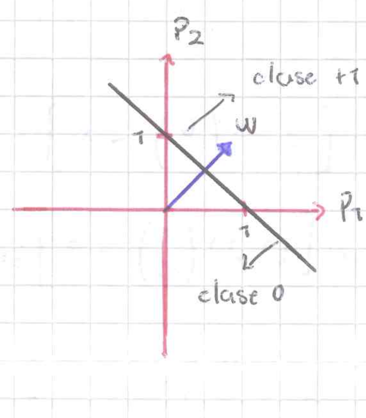
Veamos un ejemplo concreto, tomado de los siguientes valores:

$$W_1 = W_2 = 1 \ ; \quad b = -1 \quad --- \ (2)$$

Sustituyendo (2) en (1):

$$P_1 + P_2 - 1 = 0 \quad --- \ (3)$$

La ecuación (3) define una línea en el espacio, en donde, de un lado de la línea la salida del perceptrón será 0 y del otro lado +1.
Para obtener esta línea es necesario obtener los puntos dondo intersecta a los ejes P_1 y P_2.
Haciendo $P_1 = 0 \rightarrow P_2 = 1$
Haciendo $P_2 = 0 \rightarrow P_1 = 1$



27-09-19

Ejemplo Perceptrón

Diseñe un perceptrón que se comporte como la compuerta AND de 2 entradas.

Conjunto de entrenamiento:

$$\left\{ p_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix};\ t_1 = 0 \right\} \quad \left\{ p_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix};\ t_2 = 0 \right\} \quad \left\{ p_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix};\ t_3 = 0 \right\}$$

$$\left\{ p_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix};\ t_4 = 1 \right\}$$

Solución: Se usará el método de prueba y error, por lo que se propone usar $_1w^T = (2, 2)$. Con este vector de pesos se puede obtener el valor de bias correspondiente usando la siguiente ecuación que representa a la frontera de decisión:

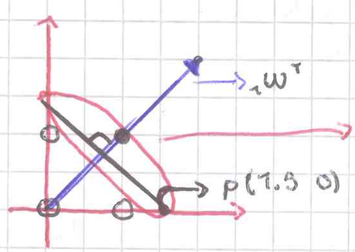
$$_1w^T p + b = 0 \quad --- (1)$$

A continuación se toma un valor "p" que esté sobre la frontera de decisión, como por ejemplo $p = (1.5 \ \ 0)^T$, y se sustituye en la ecuación (1) para obtener el valor del bias

$$(2 \ \ 2)\begin{pmatrix} 1.5 \\ 0 \end{pmatrix} + b = 0 \quad --- (2)$$

$$b = -3$$

p se obtuvo de la siguiente manera



NOTA: Se puede elegir cualquier punto sobre la recta

Una vez obtenidos estos valores es necesario hacer una comprobación numérica.
Se propagan hacia adelante todos los datos del conjunto de entrenamiento, fijando los valores de w y b

$$P_1 \rightarrow a = \text{hardlim}\left[(2 \ \ 2)\begin{pmatrix} 0 \\ 0 \end{pmatrix} - 3\right]$$

$$= 0$$

$$P_2 \rightarrow a = \text{hardlim}\left[(2 \ \ 2)\begin{pmatrix} 0 \\ 1 \end{pmatrix} - 3\right] = 0$$

$$P_3 \rightarrow a = \text{hardlim}\left[(2 \ \ 2)\begin{pmatrix} 1 \\ 0 \end{pmatrix} - 3\right] = 0$$

$$P_4 \rightarrow a = \text{hardlim}\left[(2 \ \ 2)\begin{pmatrix} 1 \\ 1 \end{pmatrix} - 3\right] = 1$$

30-09-19

Entrenamiento de Perceptrón con Múltiples Neuronas

Sabemos que un perceptrón con una neurona puede clasificar a un vector de entrada en 2 clases. Un perceptrón con múltiples neuronas puede clasificar a los vectores de entrada en múltiples clases.
Dado que a la salida del perceptrón sólo se tiene 1 o 0, existe un total de 2^S categorías posibles, donde S es el número de neuronas.

Ejemplo: Indique cuántas neuronas se requieren para representar 8 clases a la salida

$$\text{Usando } 2^S \Longrightarrow S=?\ ;\ S=3$$
$$\therefore\ 2^3 = 8$$

NOTA: El perceptrón solo es capaz de resolver problemas linealmente separables.

La regla de aprendizaje en forma matricial del perceptrón es:

$$W^{new} = W^{old} + e p^T \quad --- (1)$$
$$b^{new} = b^{old} + e \quad --- (2)$$

NOTA: Revisar en el libro la deducción de estas reglas de aprendizaje.

Ejemplo: Considere una vez más el problema de clasificación de las naranjas y las manzanas.

$$\underbrace{\left\{ p_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix},\ t_1 = [0] \right\}}_{\text{naranja}} \quad \underbrace{\left\{ p_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix},\ t_2 = [1] \right\}}_{\text{manzana}}$$

Note que en los valores deseados usamos 0 en vez de -1 debido al uso de hardlim.

Paso 1: Determinar arquitectura y modelo matemático de la RNA.

\# de neuronas $\Longrightarrow S = 1$

$$\underset{1\times1}{a} = \text{hardlim}(\underset{1\times3}{W}\ \underset{3\times1}{p} + \underset{1\times1}{b})$$

Paso 2: Se establecen valores iniciales aleatorios pequeños para W y b.

$$W = [0.5 \quad -1 \quad 0.5],\quad b = 0.5$$

Paso 3: Se realizan épocas de aprendizaje hasta que se cumpla alguno de los siguientes criterios de finalización:

a) Que se cumpla el máximo de épocas a realizar. ej. max epoch = 3
b) Que todos los datos del conjunto de entrenamiento sean bien clasificados.

Época 1

Iteración 1

$$d_1 \longrightarrow a = \text{hardlim}\left([0.5 \ -1 \ -0.5] \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + 0.5 \right) = \text{hardlim}(2.5) = 1$$

$$e = t_1 - a$$
$$= 0 - 1 = -1$$

Ahora se aplican las reglas de aprendizaje:

$$W^{new} = W^{old} + e p^T$$
$$= [0.5 \ \ -1 \ \ -0.5] + (-1)[1 \ \ -1 \ \ -1] = [-0.5 \ \ 0 \ \ 0.5]$$

$$b^{new} = b^{old} + e$$
$$= 0.5 + (-1) = -0.5$$

Iteración 2

$$d_2 \longrightarrow a = \text{hardlim}\left([-0.5 \ \ 0 \ \ 0.5]\begin{bmatrix}1\\1\\-1\end{bmatrix} + (-0.5)\right) = \text{hardlim}(-1.5) = 0$$

$$e = t_2 - a$$
$$= 1 - 0 = 1$$

$$W^{new} = [-0.5 \ \ 0 \ \ 0.5] + [1 \ \ 1 \ \ -1] = [0.5 \ \ 1 \ \ -0.5]$$
$$b^{new} = -0.5 + 1 = 0.5$$

Época 2

Iteración 3

$$d_1 \longrightarrow a = \text{hardlim}\left([0.5 \ \ 1 \ \ -0.5]\begin{bmatrix}1\\-1\\-1\end{bmatrix} + 0.5\right) = \text{hardlim}(0.5) = 1$$

$$e = t_1 - a$$
$$= 0 - 1 = -1$$

$$W^{new} = [0.5 \ \ 1 \ \ -0.5] + (-1)[1 \ \ -1 \ \ -1] = [-0.5 \ \ 2 \ \ 0.5]$$
$$b^{new} = 0.5 - 1 = -0.5$$

Iteración 4

$$d_2 \longrightarrow a = \text{hardlim}\left([-0.5 \ \ 2 \ \ 0.5]\begin{bmatrix}1\\1\\-1\end{bmatrix} - 0.5\right) = \text{hardlim}(0.5) = 1$$

$$e = t_2 - a$$
$$= 1 - 1 = 0 \qquad W^{new} = W^{old} \qquad b^{new} = b^{old}$$

Época 3

Iteración 5

$$d_1 \longrightarrow a = \text{hardlim}\left([-0.5 \ \ 2 \ \ 0.5]\begin{bmatrix}1\\-1\\-1\end{bmatrix} - 0.5\right) = \text{hardlim}(-3.5) = 0$$

$$e = t_1 - a$$
$$= 0 - 0 = 0 \qquad W^{new} = W^{old} \qquad b^{new} = b^{old}$$

Iteración 6 $\equiv$ Iteración 4

En la época 3 se obtuvo un aprendizaje exitoso debido a que un único conjunto de valores de W y b clasifican correctamente a todos los datos del conjunto de entrenamiento.

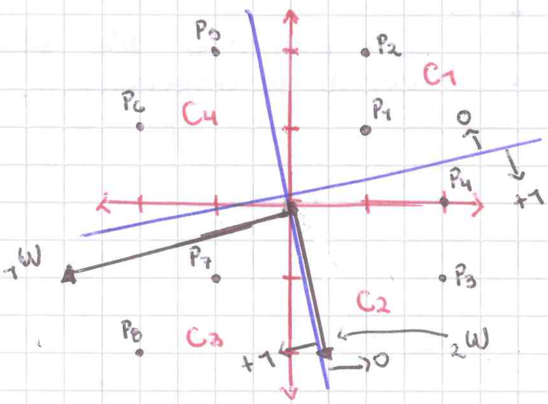
03-10-19

Ejercicio: Use el perceptrón para resolver el siguiente problema de clasificación que requiere separar los datos en 4 clases.

clase 1 = $\left\{ P_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix};\ P_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}$; clase 2 = $\left\{ P_3 = \begin{bmatrix} 2 \\ -1 \end{bmatrix};\ P_4 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right\}$

clase 3 : $\left\{ P_5 = \begin{bmatrix} -1 \\ 2 \end{bmatrix};\ P_6 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \right\}$; clase 4 = $\left\{ P_7 = \begin{bmatrix} -1 \\ -1 \end{bmatrix};\ P_8 = \begin{bmatrix} -2 \\ -2 \end{bmatrix} \right\}$

1) Primero se debe asignar el valor deseado a cada uno de los datos, para ello se grafican para poder dibujar las fronteras de decisión.



NOTA: Recordar que w siempre parte del origen.
w debe ser perpendicular a la frontera de decisión.

$P_1, P_2 \longrightarrow t = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$; $P_3, P_4 \longrightarrow t = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$; $P_5, P_6 \longrightarrow t = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$;

$P_7, P_8 \longrightarrow t = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$${}_1W = \begin{bmatrix} -3 \\ -1 \end{bmatrix} \qquad {}_2W = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

Ahora se obtienen los valores de los bias mediante la siguiente ecuación:

$$n = W^T p + b = 0 \quad --- (1)$$

$${}_1b = -{}_1W^T p + b = -\begin{bmatrix} -3 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1$$

$${}_2b = -{}_2W^T p + b = -\begin{bmatrix} 1 & -2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0$$

NOTA: Para obtener el bias el valor de p debe ser tomado de cualquier punto por el que pasa la frontera de la cual se está calculando el bias.

Finalmente se valida el resultado propagando hacia adelante c/u de los datos del conjunto de entrenamiento, usando los valores obtenidos de w y b.

$$d_1 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -3 & -1 \\ 1 & -2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$d_2 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -3 & -1 \\ 1 & -2 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$d_3 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -3 & -1 \\ 1 & -2 \end{bmatrix}\begin{bmatrix} 2 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$d_4 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -3 & -1 \\ 1 & -2 \end{bmatrix}\begin{bmatrix} 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$\vdots$

Este proceso se debe realizar con todo el conjunto de entrenamiento.

Tarea 5.

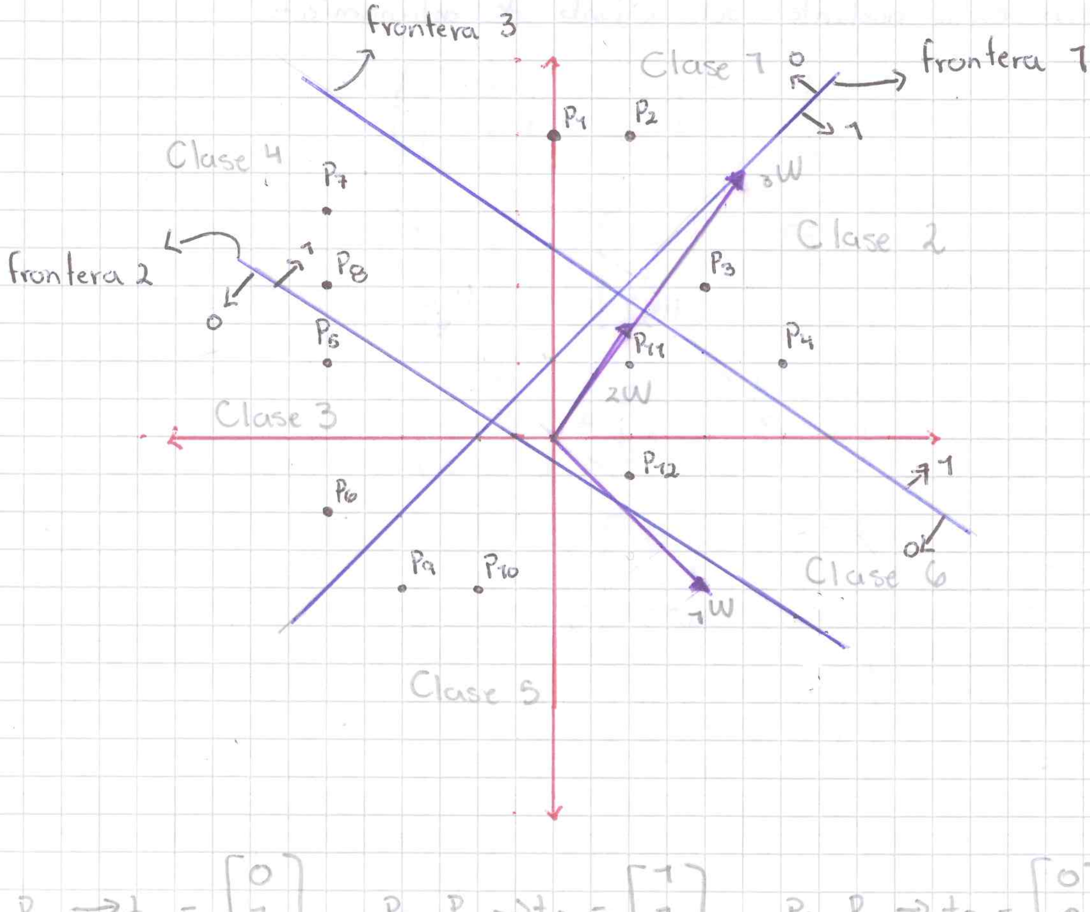
- Dataset. Se debe separar en 6 clases.

$P_1 = \begin{bmatrix} 0 \\ 8 \end{bmatrix}$ t_1 $\quad P_2 = \begin{bmatrix} 2 \\ 8 \end{bmatrix}$ t_1 $\quad P_3 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$ t_2 $\quad P_4 = \begin{bmatrix} 6 \\ 2 \end{bmatrix}$ t_2 $\quad P_5 = \begin{bmatrix} -6 \\ 2 \end{bmatrix}$ t_3

$P_6 = \begin{bmatrix} -6 \\ -2 \end{bmatrix}$ t_3 $\quad P_7 = \begin{bmatrix} -6 \\ 6 \end{bmatrix}$ t_4 $\quad P_8 = \begin{bmatrix} -6 \\ 4 \end{bmatrix}$ t_4 $\quad P_9 = \begin{bmatrix} -4 \\ -4 \end{bmatrix}$ t_5 $\quad P_{10} = \begin{bmatrix} -2 \\ -4 \end{bmatrix}$ t_5

$P_{11} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ t_6 $\quad P_{12} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$ t_6

- Graficación de puntos y fronteras de decisión.



$P_1, P_2 \to t_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ $\quad P_3, P_4 \to t_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ $\quad P_5, P_6 \to t_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

$P_7, P_8 \to t_4 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ $\quad P_9, P_{10} \to t_5 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ $\quad P_{11}, P_{12} \to t_6 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

- Obtención de valores de pesos y bias.

$$ {}_1w = \begin{bmatrix} 4 \\ -4 \end{bmatrix}, \quad {}_2w = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad {}_3w = \begin{bmatrix} 5 \\ 7 \end{bmatrix} $$

$$ {}_1b = -{}_1w^T p = -\begin{bmatrix} 4 & -4 \end{bmatrix} \begin{bmatrix} -2 \\ 0 \end{bmatrix} = 8 $$

$$ {}_2b = -{}_2w^T p = -\begin{bmatrix} 2 & 3 \end{bmatrix} \begin{bmatrix} -4 \\ 2 \end{bmatrix} = 2 $$

$$ {}_3b = -{}_3w^T p = -\begin{bmatrix} 5 & 7 \end{bmatrix} \begin{bmatrix} 6 \\ 1 \end{bmatrix} = -42 $$

- Propagación hacia adelante del conjunto de entrenamiento

$$ d_1 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \begin{bmatrix} 0 \\ 8 \end{bmatrix} + \begin{bmatrix} -8 \\ 2 \\ -42 \end{bmatrix} \right) $$

$$ = \text{hardlim}\left(\begin{bmatrix} -32 \\ 24 \\ 56 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} $$

$$ d_2 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \begin{bmatrix} 2 \\ 8 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix} \right) $$

$$ = \text{hardlim}\left(\begin{bmatrix} -24 \\ 28 \\ 66 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} $$

$$ d_3 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \begin{bmatrix} 4 \\ 4 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix} \right) $$

$$ = \text{hardlim}\left(\begin{bmatrix} 0 \\ 20 \\ 48 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} $$

$$ d_4 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \begin{bmatrix} 6 \\ 2 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix} \right) = \text{hardlim}\left(\begin{bmatrix} 16 \\ 18 \\ 44 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} $$

$$d_5 \to a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix}\begin{bmatrix} -6 \\ 2 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -32 \\ -6 \\ -16 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$d_6 \to a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix}\begin{bmatrix} -6 \\ -2 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -16 \\ -18 \\ -44 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$d_7 \to a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix}\begin{bmatrix} -6 \\ 6 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -48 \\ 6 \\ 12 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

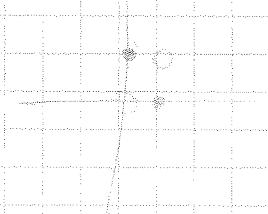
$$d_8 \to a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix}\begin{bmatrix} -6 \\ 4 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -40 \\ 0 \\ -2 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$d_9 \to a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix}\begin{bmatrix} -4 \\ -4 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 0 \\ -20 \\ -48 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$d_{10} \to a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix}\begin{bmatrix} -2 \\ -4 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 8 \\ -16 \\ -38 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

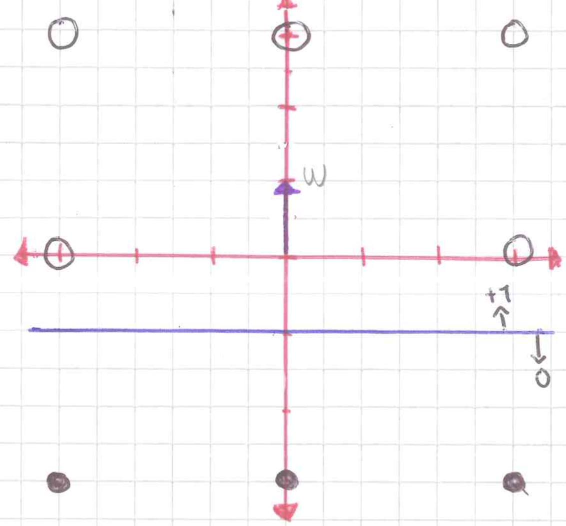
$$d_{11} \to a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix}\begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 0 \\ 10 \\ 24 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$$d_{12} \to a = \text{hardlim}\left(\begin{bmatrix} 4 & -4 \\ 2 & 3 \\ 5 & 7 \end{bmatrix}\begin{bmatrix} 2 \\ -1 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 12 \\ 1 \\ 3 \end{bmatrix} + \begin{bmatrix} 8 \\ 2 \\ -42 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$



Naranjo Ferrara Guillermo #19 10-10-19

1B)



Clase1 $\rightarrow t_1 = [1]$ Clase2 $\rightarrow t_2 = [0]$ $W = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Calculando el bias

$b = -W^T p = -[0 \ \ 1]\begin{bmatrix} 0 \\ -1 \end{bmatrix} = 1$

Validación

$P = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \rightarrow a = \text{hardlim}\left([0 \ \ 1]\begin{bmatrix} 3 \\ 3 \end{bmatrix} + 1\right) = \text{hardlim}(3+1) = 1$ ✓

$P = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \rightarrow a = \text{hardlim}\left([0 \ \ 1]\begin{bmatrix} 3 \\ 0 \end{bmatrix} + 1\right) = \text{hardlim}(0+1) = 1$ ✓

$P = \begin{bmatrix} 0 \\ -3 \end{bmatrix} \rightarrow a = \text{hardlim}\left([0 \ \ 1]\begin{bmatrix} 0 \\ -3 \end{bmatrix} + 1\right) = \text{hardlim}(-3+1) = 0$ ✓

$P = \begin{bmatrix} -3 \\ -3 \end{bmatrix} \rightarrow a = \text{hardlim}\left([0 \ \ 1]\begin{bmatrix} -3 \\ -3 \end{bmatrix} + 1\right) = \text{hardlim}(-3+1) = 0$ ✓

2) $\left\{P_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; t_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right\} \left\{P_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}; t_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right\} \left\{P_3 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}; t_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right\}$

$\left\{P_4 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}; t_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right\} \left\{P_5 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}; t_5 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right\} \left\{P_6 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}; t_6 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right\}$

$\left\{P_7 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}; t_7 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right\} \left\{P_8 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}; t_8 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right\}$

$W(0) = \begin{bmatrix} -2 & -1 \\ 3 & -4 \end{bmatrix}; \quad b(0) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

Época 1

- **Iteración 1**

$$d_1 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -2 & -1 \\ 3 & -4 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -3 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} -2 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$e = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

$$W(1) = \begin{bmatrix} -2 & -1 \\ 3 & -4 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix}\begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} -2 & -1 \\ 3 & -4 \end{bmatrix} - 1 = \begin{bmatrix} -3 & -2 \\ 2 & -5 \end{bmatrix}$$

$$b(1) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

- **Iteración 2**

$$d_2 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -3 & -2 \\ 2 & -5 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -7 \\ -8 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} -6 \\ -7 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$e = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0, \quad W(2) = W(1), \quad b(2) = b(1)$$

- **Iteración 3**

$$d_3 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -3 & -2 \\ 2 & -5 \end{bmatrix}\begin{bmatrix} 2 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -4 \\ 9 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} -5 \\ 10 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$e = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0, \quad w(3) = w(2), \quad b(3) = b(2)$$

- **Iteración 4**

$$d_4 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -3 & -2 \\ 2 & -5 \end{bmatrix}\begin{bmatrix} 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -6 \\ 4 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} -5 \\ 5 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$e = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0, \quad w(4) = w(3), \quad b(4) = b(3)$$

- **Iteración 5**

$$d_5 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -3 & -2 \\ 2 & -5 \end{bmatrix}\begin{bmatrix} -1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -1 \\ -12 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} 0 \\ -11 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$e = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$w(5) = \begin{bmatrix} -3 & -2 \\ 2 & -5 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} -1 & 2 \end{bmatrix} = \begin{bmatrix} -3 & -2 \\ 2 & -5 \end{bmatrix} - 1 = \begin{bmatrix} -4 & -3 \\ 1 & -6 \end{bmatrix}$$

$$b(5) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

- Iteración 6

$$d_6 \to a = \text{hardlim}\left(\begin{bmatrix} -4 & -3 \\ 1 & -6 \end{bmatrix}\begin{bmatrix} -2 \\ 1 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 5 \\ -8 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} 7 \\ -7 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$e = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0 \ , \quad w(6) = w(5) \ , \quad b(6) = b(5)$$

- Iteración 7

$$d_7 \to a = \text{hardlim}\left(\begin{bmatrix} -4 & -3 \\ 1 & -6 \end{bmatrix}\begin{bmatrix} -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 7 \\ 5 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} 9 \\ 6 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$e = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0 \ , \quad w(7) = w(6) \ ; \quad b(7) = b(6)$$

- Iteración 8

$$d_8 \to a = \text{hardlim}\left(\begin{bmatrix} -4 & -3 \\ 1 & -6 \end{bmatrix}\begin{bmatrix} -2 \\ -2 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 14 \\ 10 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} 16 \\ 11 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$e = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0 \ , \quad w(8) = w(7) \ , \quad b(8) = b(7)$$

Época 2

- Iteración 9

$$d_1 \to a = \text{hardlim}\left(\begin{bmatrix} -4 & -3 \\ 1 & -6 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -7 \\ -5 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} -5 \\ -4 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$e = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0 \ , \quad w(9) = w(8) \ , \quad b(9) = b(8)$$

- Iteración 10

$$d_2 \to a = \text{hardlim}\left(\begin{bmatrix} -4 & -3 \\ 1 & -6 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -10 \\ -11 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} -8 \\ -10 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$e = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0 \ , \quad w(10) = w(9) \ , \quad b(10) = b(9)$$

- Iteración 11

$$d_3 \to a = \text{hardlim}\left(\begin{bmatrix} -4 & -3 \\ 1 & -6 \end{bmatrix}\begin{bmatrix} 2 \\ -1 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -5 \\ 8 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} -3 \\ 9 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$e = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \ , \quad w(11) = w(10) \ , \quad b(11) = b(10)$$

- Iteración 12

$$d_4 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -4 & -3 \\ 1 & -6 \end{bmatrix}\begin{bmatrix} 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -8 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} -6 \\ 3 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$e = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0, \quad w(12) = w(11), \quad b(12) = b(11)$$

- Iteración 13

$$d_5 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -4 & -3 \\ 1 & -6 \end{bmatrix}\begin{bmatrix} -1 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} -2 \\ -13 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} 0 \\ -12 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$e = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$w(13) = \begin{bmatrix} -4 & -3 \\ 1 & -6 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} -1 & 2 \end{bmatrix} = \begin{bmatrix} -4 & -3 \\ 1 & -6 \end{bmatrix} - 1 = \begin{bmatrix} -5 & -3 \\ 0 & -7 \end{bmatrix}$$

$$b(13) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

- Iteración 14

$$d_6 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -5 & -3 \\ 0 & -7 \end{bmatrix}\begin{bmatrix} -2 \\ 1 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 7 \\ -7 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} 10 \\ -6 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$e = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0, \quad w(14) = w(13), \quad b(14) = b(13)$$

- Iteración 15

$$d_7 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -5 & -3 \\ 0 & -7 \end{bmatrix}\begin{bmatrix} -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 8 \\ 7 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} 11 \\ 8 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$e = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0, \quad w(15) = w(14), \quad b(15) = b(14)$$

- Iteración 16

$$d_8 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -5 & -3 \\ 0 & -7 \end{bmatrix}\begin{bmatrix} -2 \\ -2 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 16 \\ 14 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}\right)$$

$$= \text{hardlim}\left(\begin{bmatrix} 19 \\ 15 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$e = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0, \quad w(16) = w(15), \quad b(16) = b(15)$$

Época 3

- **Iteración 17**

$d_1 \rightarrow a = \text{hardlim}\left(\begin{bmatrix}-5 & -3\\ 0 & -7\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix}-8\\-7\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right)$

$= \text{hardlim}\left(\begin{bmatrix}-5\\-6\end{bmatrix}\right) = \begin{bmatrix}0\\0\end{bmatrix}$

$e = \begin{bmatrix}0\\0\end{bmatrix} - \begin{bmatrix}0\\0\end{bmatrix} = 0, \quad w(17) = w(16), \quad b(17) = b(16)$

- **Iteración 18**

$d_2 \rightarrow a = \text{hardlim}\left(\begin{bmatrix}-5 & -3\\ 0 & -7\end{bmatrix}\begin{bmatrix}1\\2\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix}-7\\-14\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right)$

$= \text{hardlim}\left(\begin{bmatrix}-4\\-13\end{bmatrix}\right) = \begin{bmatrix}0\\0\end{bmatrix}$

$e = \begin{bmatrix}0\\0\end{bmatrix} - \begin{bmatrix}0\\0\end{bmatrix} = 0, \quad w(18) = w(17), \quad b(18) = b(17)$

- **Iteración 19**

$d_3 \rightarrow a = \text{hardlim}\left(\begin{bmatrix}-5 & -3\\ 0 & -7\end{bmatrix}\begin{bmatrix}2\\-1\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix}-7\\7\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right)$

$= \text{hardlim}\left(\begin{bmatrix}-4\\8\end{bmatrix}\right) = \begin{bmatrix}0\\1\end{bmatrix}$

$e = \begin{bmatrix}0\\1\end{bmatrix} - \begin{bmatrix}0\\1\end{bmatrix} = 0, \quad w(19) = w(18), \quad b(19) = b(18)$

- **Iteración 20**

$d_4 \rightarrow a = \text{hardlim}\left(\begin{bmatrix}-5 & -3\\ 0 & -7\end{bmatrix}\begin{bmatrix}2\\0\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix}-10\\0\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right)$

$= \text{hardlim}\left(\begin{bmatrix}-7\\1\end{bmatrix}\right) = \begin{bmatrix}0\\1\end{bmatrix}$

$e = \begin{bmatrix}0\\1\end{bmatrix} - \begin{bmatrix}0\\1\end{bmatrix} = 0, \quad w(20) = w(19), \quad b(20) = b(19)$

- **Iteración 21**

$d_5 \rightarrow a = \text{hardlim}\left(\begin{bmatrix}-5 & -3\\ 0 & -7\end{bmatrix}\begin{bmatrix}-1\\2\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix}-1\\-14\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right)$

$= \text{hardlim}\left(\begin{bmatrix}2\\-13\end{bmatrix}\right) = \begin{bmatrix}1\\0\end{bmatrix}$

$e = \begin{bmatrix}1\\0\end{bmatrix} - \begin{bmatrix}1\\0\end{bmatrix} = 0, \quad w(21) = w(20), \quad b(21) = b(20)$

- **Iteración 22**

$d_6 \rightarrow a = \text{hardlim}\left(\begin{bmatrix}-5 & -3\\ 0 & -7\end{bmatrix}\begin{bmatrix}-2\\1\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix}7\\-7\end{bmatrix}+\begin{bmatrix}3\\1\end{bmatrix}\right)$

$= \text{hardlim}\left(\begin{bmatrix}10\\-6\end{bmatrix}\right) = \begin{bmatrix}1\\0\end{bmatrix}$

$e = \begin{bmatrix}1\\0\end{bmatrix} - \begin{bmatrix}1\\0\end{bmatrix} = 0, \quad w(22) = w(21), \quad b(22) = b(21)$

- Iteración 23

$d_7 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -5 & -3 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 8 \\ 1 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}\right)$

$= \text{hardlim}\left(\begin{bmatrix} 11 \\ 2 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$e = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0, \quad w(23) = w(22), \quad b(23) = b(22)$

- Iteración 24

$d_8 \rightarrow a = \text{hardlim}\left(\begin{bmatrix} -5 & -3 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} -2 \\ -2 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}\right) = \text{hardlim}\left(\begin{bmatrix} 16 \\ 14 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}\right)$

$= \text{hardlim}\left(\begin{bmatrix} 19 \\ 15 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$e = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0, \quad w(24) = w(23), \quad b(24) = b(23)$

Se termina el entrenamiento en la época 3 siendo exitoso.

14-10-19

Red Adaline

En el caso del perceptrón de una sola neurona es posible separar a los datos de entrada en 2 clases. Para separar estos datos en más de dos clases usando una sola neurona, se usa una RNA conocida como Adaptative Linear Neuron (ADALINE).

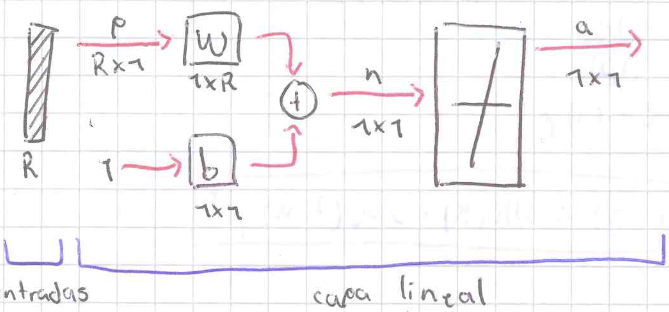
Descripción del modelo

En 1960, Widrow y Hoff proponen esta RNA con una regla de aprendizaje que toma en cuenta la señal del error, usando la siguiente definición:

$$e = (t-a)^2 \quad \text{---} \quad (1)$$

La ecuación (1) se eleva al cuadrado para considerar solamente la magnitud de este valor.

Arquitectura



$$a = \text{purelin}(Wp + b)$$

Regla de aprendizaje

Esta regla actualiza automáticamente los valores de W y b de la RNA haciendo uso de la señal de error.

A manera de ejemplo, considere una red ADALINE sin bias, por lo que solo será necesario obtener reglas de aprendizaje para los pesos W.

Se parte de la siguiente ecuación que representa el formato base para la regla de aprendizaje:

$$W_i(k+1) = W_i(k) - \Delta W_i \quad \text{---} \quad (2)$$

El diseño de una regla de aprendizaje consiste en proponer una forma para ΔW_i. En el caso de ADALINE se diseña un ΔW_i que hace uso de la regla delta, por lo que tenemos lo siguiente:

$$\Delta W_i = \alpha \frac{\partial e}{\partial W_i} \quad \text{---} \quad (3)$$

La ecuación (3) es una regla de aprendizaje supervisada.

donde:

- α: Factor de aprendizaje y toma valores $[0, 1]$

Para obtener de manera explícita una expresión para ΔW_i, observe que en la ecuación del error no aparece de forma explícita el término W_i, por lo que se usará la regla de la cadena para obtener la expresión deseada. Para ello se parte de las siguientes ecuaciones:

$$e = (t-a)^2$$
$$a = purelin(Wp+b)$$

A continuación, se aplica la regla de la cadena para resolver la derivada parcial de la ecuación (3):

$$\frac{\partial e}{\partial w_i} = \frac{\partial e}{\partial a} \cdot \frac{\partial a}{\partial f} \cdot \frac{\partial f}{\partial w_i}$$

$$\frac{\partial e}{\partial a} = -2(t-a) \qquad \frac{\partial a}{\partial f} = 1 \qquad \frac{\partial f}{\partial w_i} = p$$

Uniendo todos los resultados:

$$W_i(k+1) = W_i(k) - \alpha \frac{\partial e}{\partial w_i}$$
$$= W_i(k) + 2\alpha e(1) p^T$$

$$\boxed{W_i(k+1) = W_i(k) + 2\alpha(t-a)\, p^T}$$

Ahora para el bias:

$$\Delta W_i = -\alpha \frac{\partial e}{\partial b} \qquad \frac{\partial e}{\partial b} = \frac{\partial e}{\partial a} \frac{\partial a}{\partial f} \frac{\partial f}{\partial b}$$

$$\boxed{b(k+1) = b(k) + 2\alpha(t-a)(1)}$$

Ejercicio de regresión: Realice el aprendizaje de un codificador de binario a decimal de 3 bits.

- Data set

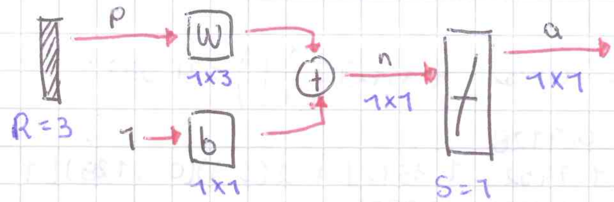
	P_1	P_2	P_3	t
d_1	0	0	0	0
d_2	0	0	1	1
d_3	0	1	0	2
d_4	0	1	1	3
d_5	1	0	0	4
d_6	1	0	1	5
d_7	1	1	0	6
d_8	1	1	1	7

- Datos proporcionados por el usuario
  - epochmax = 5
  - Eepoch = 0.01
  - $\alpha = 0.3$

Solución

1.- Modelo matemático y arquitectura

NOTA: Para regresión solo se usa una neurona.



$$a = \text{purelin}(Wp + b)$$

2.- Pedir datos al usuario, los cuales ya tenemos

3.- $W = [0.84 \quad 0.39 \quad 0.78]$

NOTA: Para el modo regresión puede o no usarse bias. Para el caso del codificador no usaremos bias debido a que es un modelo lineal.

4.- Iniciar una época de aprendizaje

$$d_1 \rightarrow a = \text{purelin}\left([0.84 \quad 0.39 \quad 0.78]\begin{bmatrix}0\\0\\0\end{bmatrix}\right) = 0$$

$$e = t - a = 0$$
$$W(1) = W(0)$$

$$d_2 \rightarrow a = \text{purelin}\left([0.84 \quad 0.39 \quad 0.78]\begin{bmatrix}0\\0\\1\end{bmatrix}\right) = 0.78$$

$$e = 1 - 0.78 = 0.22$$
$$W(2) = W(1) + 2\alpha e p^T$$
$$= [0.84 \quad 0.39 \quad 0.78] + 2(0.3)(0.22)[0 \quad 0 \quad 1]$$
$$= [0.84 \quad 0.39 \quad 0.912]$$

$$d_3 \rightarrow a = \text{purelin}\left([0.84 \quad 0.39 \quad 0.912]\begin{bmatrix}0\\1\\0\end{bmatrix}\right) = 0.39$$

$$e = 2 - 0.39 = 1.61$$
$$W(3) = [0.84 \quad 0.39 \quad 0.912] + 2(0.3)(1.61)[0 \quad 1 \quad 0]$$
$$= [0.84 \quad 1.356 \quad 0.912]$$

$$d_4 \rightarrow a = \text{purelin}\left([0.84 \quad 1.356 \quad 0.912]\begin{bmatrix}0\\1\\1\end{bmatrix}\right) = 2.268$$

$$e = 3 - 2.268 = 0.732$$
$$W(4) = [0.84 \quad 1.356 \quad 0.912] + 2(0.3)(0.732)[0 \quad 1 \quad 1]$$
$$= [0.84 \quad 1.7952 \quad 1.3512]$$

$$d5 \rightarrow a = purelin\left(\begin{bmatrix} 0.84 & 1.7952 & 1.3512 \end{bmatrix} \begin{bmatrix} 1 \\ 8 \end{bmatrix}\right) = 0.84$$

$$e = 4 - 0.84 = 3.16$$

$$w(5) = \begin{bmatrix} 0.84 & 1.7952 & 1.3512 \end{bmatrix} + 2(0.3)(3.16)\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2.736 & 1.7952 & 1.3512 \end{bmatrix}$$

$$d6 \rightarrow a = purelin\left(\begin{bmatrix} 2.736 & 1.7952 & 1.3512 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}\right) = 4.0872$$

$$e = 5 - 4.0872 \quad 0.9128$$

$$w(6) = \begin{bmatrix} 2.736 & 1.7952 & 1.3512 \end{bmatrix} + 2(0.3)(0.9128)\begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 3.2836 & 1.7952 & 1.8988 \end{bmatrix}$$

$$d7 \rightarrow a = purelin\left(\begin{bmatrix} 3.2836 & 1.7952 & 1.8988 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}\right) = 5.0788$$

$$e = 6 - 5.0788 = 0.9212$$

$$w(7) = \begin{bmatrix} 3.2836 & 1.7952 & 1.8988 \end{bmatrix} + 2(0.3)(0.9212)\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 3.8363 & 2.3479 & 1.8988 \end{bmatrix}$$

$$d8 \rightarrow a = purelin\left(\begin{bmatrix} 3.8363 & 2.3479 & 1.8988 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\right) = 8.083$$

$$e = 7 - 8.083 = -1.083$$

$$w(8) = \begin{bmatrix} 3.8363 & 2.3479 & 1.8988 \end{bmatrix} + 2(0.3)(-1.083)\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 3.1865 & 1.6981 & 1.249 \end{bmatrix}$$

5.- Calcular eepoch

$$eepoch = \frac{1}{8}\left|\sum_{i=1}^{8} e_i\right| = \frac{1}{8}\left|(0 + 0.22 + 1.61 + 0.732 + 3.16 + 0.9128 + 0.9212 - 1.083)\right|$$

$$= 0.8091$$

6.- Criterios de finalización

i) No es un entrenamiento correcto

ii) $0.8091 < 0.01$? No

iii) Se llegó a epochmax? No

Dado que ningún criterio se cumple, regresamos al paso 4.

Backpropagation.

Un MLP con regla de aprendizaje Backpropagation se describe con las siguientes ecuaciones:

a) Modelo matemático o propagación hacia adelante

$$a^0 = p$$

$$a^{m+1} = f^{m+1}(W^{m+1} a^m + b^{m+1})$$

$\hookrightarrow$ para $m = 0, 1, \ldots, (M-1)$

$$a = a^M$$

donde: M – Es el número de capas

b) Backpropagation

En este algoritmo de aprendizaje supervisado, primero se calculan las sensitividades y luego se usan para las reglas de aprendizaje.

b1) Sensitividades

$$s^M = -2\dot{F}^M(n^M)(t-a)$$

$$s^m = \dot{F}^m(n^m)(W^{m+1})^T s^{m+1}$$

$\hookrightarrow$ para $m = (M-1), \ldots, 2, 1$

NOTA: Se debe obtener una ecuación de sensitividad por cada capa del MLP, donde:

$$\dot{F}^m(n^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \cdots & 0 \\ 0 & \dot{f}^m(n_2^m) & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{f}^m(n_{S^m}^m) \end{bmatrix}$$

– $\dot{F}(\cdot)$ solo tiene valores en la diagonal principal.

– Es una matriz cuadrada cuya dimensión está dada por el número de neuronas en esa capa.

b2) Reglas de aprendizaje

$$W^m(k+1) = W^m(k) - \alpha s^m (a^{m-1})^T$$

$$b^m(k+1) = b^m(k) - \alpha s^m$$

04-11-19

Ejemplo: Dado el siguiente MLP obtenga sus ecuaciones de propagación hacia adelante y backpropagation.

$$V_1 : [4 \quad 2 \quad 3 \quad 1]$$
$$V_2 = [2 \quad 3 \quad 1]$$

1) Propagación hacia adelante

- Se establece el valor de $M \longrightarrow M = 3$

$$m = 0, 1, \underset{M-1}{2}$$

$$\underset{4\times1}{a^0} = \underset{4\times1}{P}$$

$$m=0 \longrightarrow \underset{2\times1}{a^1} = \text{logsig}(\underset{2\times4}{W^1}\ \underset{4\times1}{a^0} + \underset{2\times1}{b^1})$$

$$m=1 \longrightarrow \underset{3\times1}{a^2} = \text{tansig}(\underset{3\times2}{W^2}\ \underset{2\times1}{a^1} + \underset{3\times1}{b^2})$$

$$m=2 \longrightarrow \underset{1\times1}{a^3} = \text{purelin}(\underset{1\times3}{W^3}\ \underset{3\times1}{a^2} + \underset{1\times1}{b^3})$$

$$\underset{1\times1}{a} = \underset{1\times1}{a^3}$$

2) Backpropagation

- Sensitividades

$$m = (M-1), \ldots, 2, 1 \longrightarrow m = 2, 1$$

$$M=3 \longrightarrow s^3 = -2\dot{F}^3(n^3)(t-a) \qquad \underset{1\times1}{\dot{F}^3(n^3)} = [\underbrace{\dot{f}^3(n^3)}_{\text{purelin}(\cdot)}] = 1$$
$$s^3 = -2(t-a)$$

$$m=2 \longrightarrow s^2 = \dot{F}^2(n^2)(W^3)^T s^3$$

$$\dot{F}^2(n^2) = \begin{bmatrix} \dot{f}^2(n_1^2) & 0 & 0 \\ 0 & \dot{f}^2(n_2^2) & 0 \\ 0 & 0 & \dot{f}^2(n_3^2) \end{bmatrix} = \begin{bmatrix} 1-(a_1^2)^2 & 0 & 0 \\ 0 & 1-(a_2^2)^2 & 0 \\ 0 & 0 & 1-(a_3^2)^2 \end{bmatrix}$$

$$m=1 \longrightarrow s^1 = \dot{F}^1(n^1)(W^2)^T s^2$$

$$\dot{F}^1(n^1) = \begin{bmatrix} \dot{f}^1(n_1^1) & 0 \\ 0 & \dot{f}^1(n_2^1) \end{bmatrix} = \begin{bmatrix} a_1^1(1-a_1^1) & 0 \\ 0 & a_2^1(1-a_2^1) \end{bmatrix}$$

NOTA: Realizando el cálculo, tenemos que

$$\frac{d}{dn}\text{tansig}(a) = \frac{2(ne^{2n-1} + ne^{-2n-1})}{(e^n + e^{-n})^2}$$

- Finalmente se obtienen las reglas de aprendizaje por cada capa.

Ejercicio: Obtenga por separado las reglas de aprendizaje de c/u de los parámetros del siguiente MLP.

V1: [1 2 1] V2: [2 1]

a) Propagación hacia adelante

$$\underset{1\times1}{a^0} = \underset{1\times1}{P}$$

$$\underset{2\times1}{\begin{bmatrix} a_1^1 \\ a_2^1 \end{bmatrix}} = \text{logsig}\left(\underset{2\times1}{\begin{bmatrix} w_{11}^1 \\ w_{21}^1 \end{bmatrix}} \underset{1\times1}{[a^0]} + \underset{2\times1}{\begin{bmatrix} b_1^1 \\ b_2^1 \end{bmatrix}}\right)$$

$$\underset{1\times1}{a^2} = \text{purelin}\left(\underset{2\times1}{[w_{11}^2 \;\; w_{12}^2]} \underset{1\times2}{\begin{bmatrix} a_1^1 \\ a_2^1 \end{bmatrix}} + \underset{1\times1}{[b_1^2]}\right)$$

Entonces tenemos que:

$$a_1^1 = \text{logsig}(w_{11}^1 \, a^0 + b_1^1)$$
$$a_2^1 = \text{logsig}(w_{21}^1 \, a^0 + b_2^1)$$
$$a^2 = \text{purelin}(w_{11}^2 \, a_1^1 + w_{12}^2 \, a_2^1 + b_1^2)$$

b) Backpropagation

b1) Sensitividades

$M=2$

$$S^2 = -2\dot{F}^2(n^2)(t-a) \quad - - - (1)$$

$$\underset{1\times1}{\dot{F}^2(n^2)} = [\dot{f}^2(n^2)] = 1 \quad - - - (2)$$

Sustituyendo 2 en 1

$$S^2 = -2(t-a)$$

$m=1$

$$\underset{2\times1}{S^1} = \underset{2\times2}{\dot{F}^1(n^1)} \underset{2\times1}{(W^2)^T} \underset{1\times1}{S^2}$$

$$\underset{2\times2}{\dot{F}^1(n^1)} = \begin{bmatrix} \dot{f}^2(n_1^1) & 0 \\ 0 & \dot{f}^2(n_2^1) \end{bmatrix} = \begin{bmatrix} a_1^1(1-a_1^1) & 0 \\ 0 & a_2^1(1-a_2^1) \end{bmatrix}$$

$$\begin{bmatrix} S_1^1 \\ S_2^1 \end{bmatrix} = \begin{bmatrix} a_1^1(1-a_1^1) & 0 \\ 0 & a_2^1(1-a_2^1) \end{bmatrix} \begin{bmatrix} w_{11}^2 \\ w_{12}^2 \end{bmatrix} (-2(t-a))$$

14-11-19

$$S^1_1 = a^1_1(1-a^1_1)\, W^2_{11}\,(-2(t-a))$$

$$S^1_2 = a^1_2(1-a^1_2)\, W^2_{12}\,(-2(t-a))$$

b2) Reglas de aprendizaje

$m=2$

$$\underset{1\times 2}{\left[W^2_{11}(k+1)\quad W^2_{12}(k+1)\right]} = \underset{1\times 2}{\left[W^2_{11}(k)\quad W^2_{12}(k)\right]} - \alpha\, \underset{1\times 1}{\left[-2(t-a)\right]}\, \underset{1\times 2}{(a^1)^T}$$

$$W^2_{11}(k+1) = W^2_{11}(k) + 2\alpha(t-a)\, a^1_1$$

$$W^2_{12}(k+1) = W^2_{12}(k) + 2\alpha(t-a)\, a^1_2$$

$$b^2_1(k+1) = b^2_1(k) - \alpha\left[-2(t-a)\right] = b^2_1 + 2\alpha(t-a)$$

Se hace lo mismo para $m=1$

$m=1$

$$\underset{2\times 1}{\begin{bmatrix} W^1_{11}(k+1) \\ W^1_{21}(k+1) \end{bmatrix}} = \underset{2\times 1}{\begin{bmatrix} W^1_{11}(k) \\ W^1_{21}(k) \end{bmatrix}} - \alpha\, \underset{2\times 1}{S^1}\, \underset{2\times 1}{(a^0)^T}$$

$$= \begin{bmatrix} W^1_{11}(k) \\ W^1_{21}(k) \end{bmatrix} - \alpha \begin{bmatrix} a^1_1(1-a^1_1) & 0 \\ 0 & a^1_2(1-a^1_2) \end{bmatrix} \begin{bmatrix} W^2_{11} \\ W^2_{12} \end{bmatrix} \left[-2(t-a)\right](a^0)^T$$