NEURAL NETWORKS

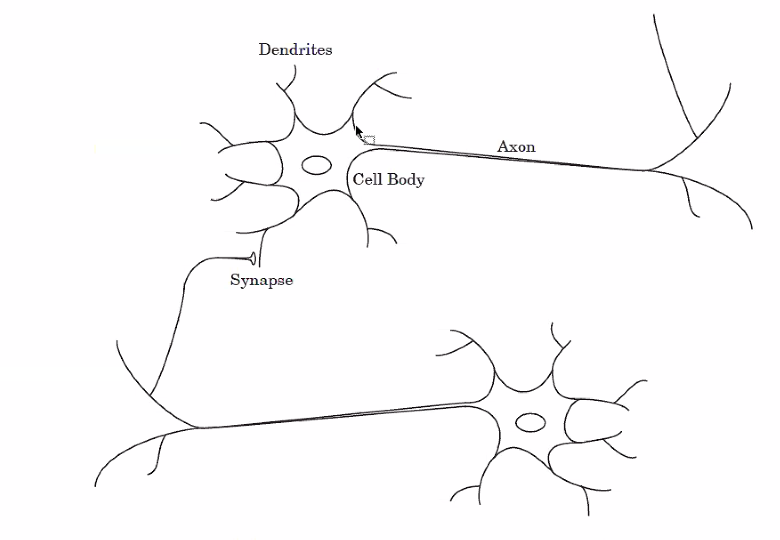
*Apuntes por Kevin David*

**Notación:**

* Escalares: Minúsculas en itálica *a,b,c*
* Vectores: Minúsculas en negritas **a,b,c**
* Matrices: Mayúsculas en negritas **A,B,C**

**Neuronas vistas biológicamente:**

Las redes neuronales son un modelo bioinspirado en las neuronas del cerebro humano.

****

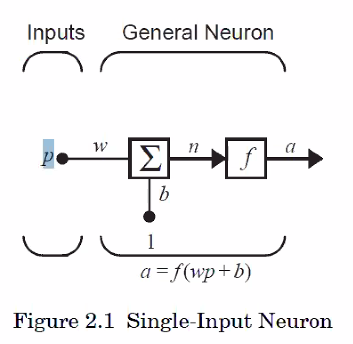
Las dendritas son las entradas.

El Axón es la salida de una neurona, y se convierte en la entrada de otras mediante la sinapsis.

Esta salida solo se produce cuando se supera el umbral que está determinado.

**La red neuronal mas pequeña que existe es de una sola neurona**

**Neuronas artificiales:**

****

p=Señal de entrada (Valor escalar)

La dendrita es la linea donde esta la w

w=peso sináptico, establece el nivel de importancia de cada una de las entradas de la neurona (calor escalar)

b=Es una entrada que no existe en las entradas biológicas, pero se decidió agregar (bias)

La sumatoria es el soma artificial.

La primera entrada del soma es w\*p, y la segunda es 1\*b, que viene del bias.

La señal de salida es n

n=w\*p+1\*p

Para decidir cuando esta neurona artificial genera una salida, se utiliza f, la cual se denomina señal de transferencia, y esta indica el valor que se generará a la salida (el valor a)

f=función de transferencia.

.: a=f(wp+b)

**Ejemplo:**

w=3, p=2 y b=-1.5

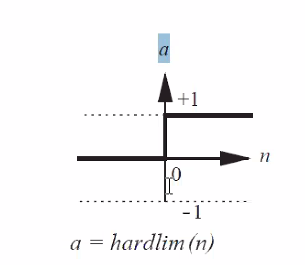
.: a=f(3(2)-1.5)=f(4.5)

Los únicos dos parámetros que se pueden cambiar es p y b

**Funciones de Transferencia:**

Al colocar funciones de transferencia se evita que se hagan multiplicaciones de números muy grandes, para evitar que estalle numéricamente la red neuronal, enviando valores muy pequeños.

Su segunda tarea de las funciones de transferencia, es la clasificación o la aproximación de señales, dependiendo de la función.

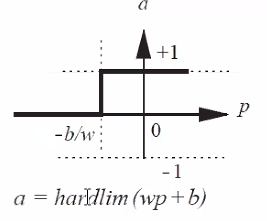


*El hardlim, no es más que como se llama la función en matlab, así será en todas las imágenes.*

**Hard Limit Transfer Function:**

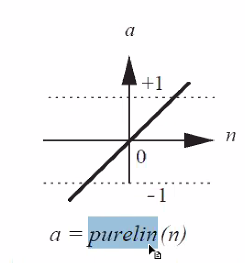
Si n<=0, n=0

Si n>0, n=1



**Single-input hardlim Neuron:**

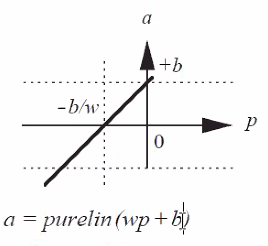
Al agregar el bias, se puede desplazar el valor de salida hacia delante o hacia atrás mediante el valor -b/w.



**Función de Transferencia lineal:**

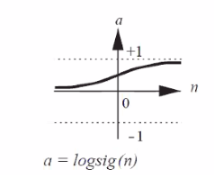
a=n

*.:Lo que se recibe como entrada, se pasa a la salida*



**Single-Input purelin Neuron:**

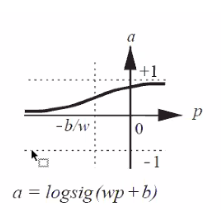
Con el bias se puede desplazar



**Log-Sigmoid Transfer Function**

Es una función continua, se hace una transición suave entre el 0 y el 1, a diferencia de Hard Limit que era un escalón.

Recordemos que “a”, es el valor de salida de la neurona.

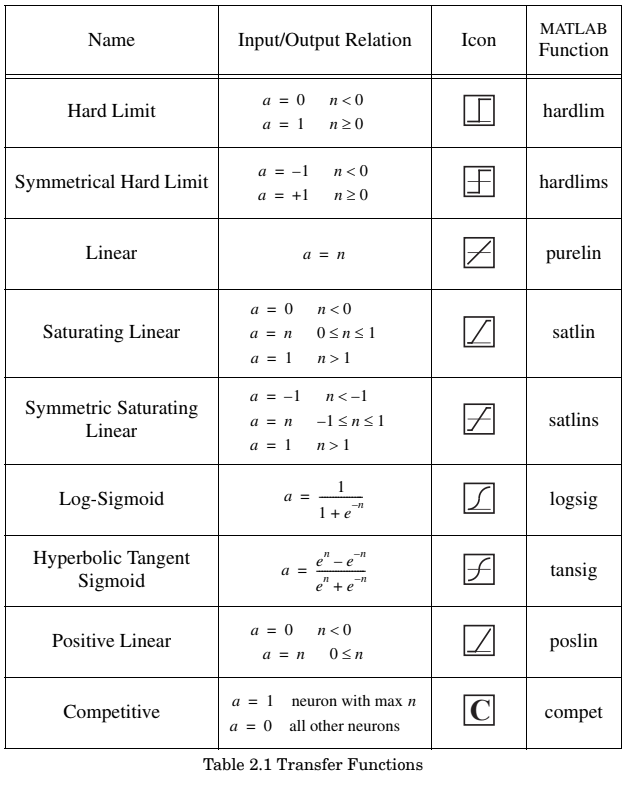


**Single input logsig Neuron**

Nos permite mover la señal hacia adelante o atrás, utilizando bias.

**Función Competitiva:**

Aunque se tengan “m” neuronas, la salida será 1 únicamente en la neurona que tenga el máximo n.



**Algunos ejemplos de uso:**

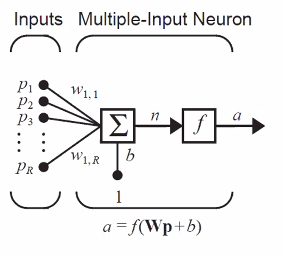
**Hard Limit:** Clasificación de objetos (Únicamente dos clases (Clase 0 y 1)).

**Symmetrical Hard Limit:** Clasificación de objetos (Únicamente dos clases (Clase -1 y +1)).

**Linear:** Aproximación de señales.

**Positive Linear:** Aproximación de señales o clasificación de objetos.

**Competitiva:** Clasificación de objetos.



Neurona con múltiples entradas

p deja de ser un escalar y se convierte en un vector.

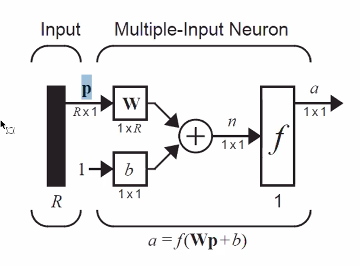
R es el tamaño del vector → .: si es 1, es un escalar.

W1,1 significa que es un peso sináptico que establece la importancia de la conexión de la primer neurona con la primer entrada

.: W1,R es la importancia de la conexión de la primer neurona con la entrada numero R

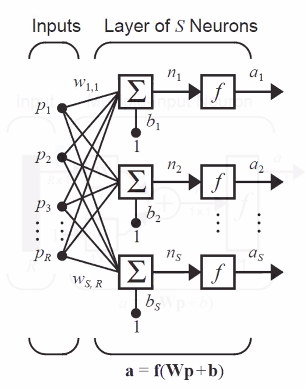


Esta expresión también se puede escribir en forma de matriz **Wp**+b



Notación Abreviada

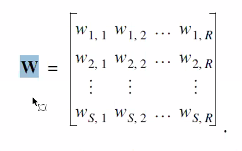
**Por cada neurona solamente se tendrá un bias**



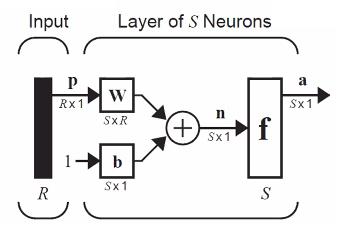
Capa de S neuronas

WS,R representa el peso sinaptico entre la neurona S con la entrada R

**a** = **f**(**Wp**+**b**)

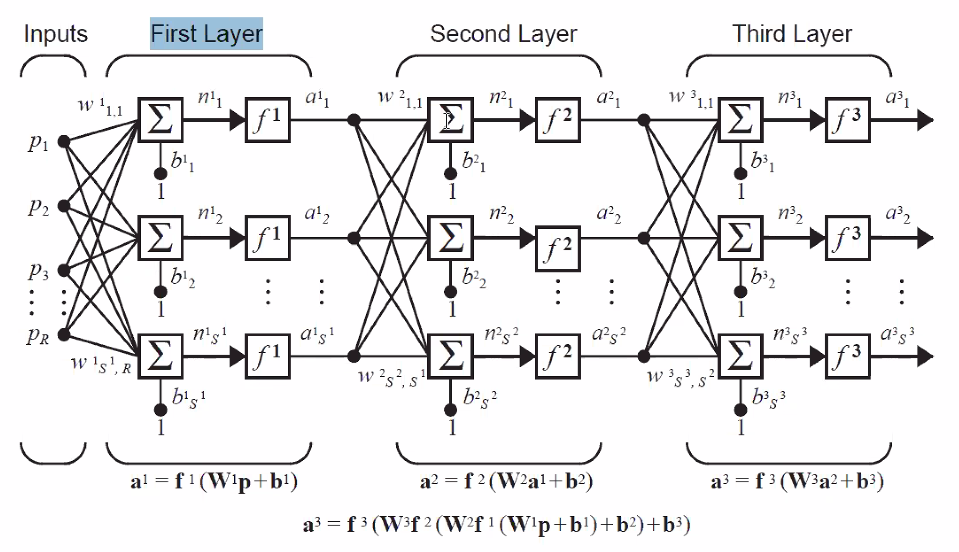


W es una matriz de pesos sinápticos

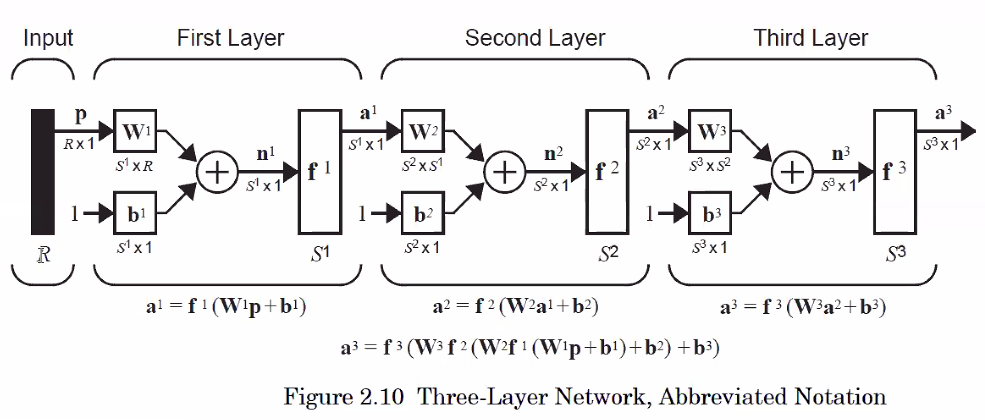


Notación abreviada

**a**=**f**(**Wp**+**b**)



Se conoce como una red fit forward, ya que solo tiene propagación hacia adelante.



Notación abreviada

MLP

Ejemplo: 3 capas

V1=[1 3 4 2]

R S1 S2 S3

Primer capa= a1=f1(w1p+b)

Se puede trabajar de dos modos basicos

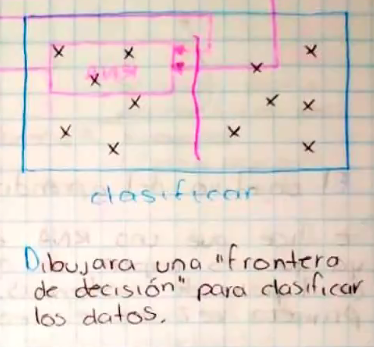
1.-Regresor: La red ajusta datos



*La linea de color rosa, se llama interpolar*

*Interpolar=Unir los datos con la linea de color rosa.*

2.-Clasificador: La red separa datos



*La línea se llama frontera de decisión*

