



# Instituto Politécnico Nacional

## Escuela Superior de Cómputo

*Asignatura:*

Neural Networks

*Grupo:* 3CM2

### Práctica 2. **Red de Hamming**

*Alumno:*

Garcia Garcia Rafael

*Profesor:* Moreno Armendariz Marco Antonio

## Introducción:

Hemos estudiado diversas arquitecturas de redes neuronales cada una con su propia función como lo son las regresivas que mediante una extrapolación logran obtener valores fuera de rango de una función o las de aproximación que miden el error entre el comportamiento de datos y una función haciendo una interpolación.

En este documento buscamos ver a fondo una de los tipos de redes esenciales dentro de las RNA. Cuando las entradas de una RNA son binarias, las redes de Hamming son fundamentales. La red de Hamming fue desarrollada por Richard P. Lippmann a mediados de 1.987, al año siguiente lo publica como "An introduction to computing with neural nets". La red es muy similar a la de Hopfield, con la única diferencia de que ésta implementa un clasificador máximo de confianza, el aprendizaje también es similar.

## Marco teórico.

### Explicación de la arquitectura,

La red de Hamming consta de 2 capas una de ellas realiza la correlación entre el vector de entrada y los vectores prototipo, la segunda capa realiza la competición para determinar cuál de los vectores prototipo está más cercano al valor de entrada.

Esta RNA es interesante debido a que usa dos capas: una feed forward y otra recurrente como se muestra en la imagen de abajo (Hagan):

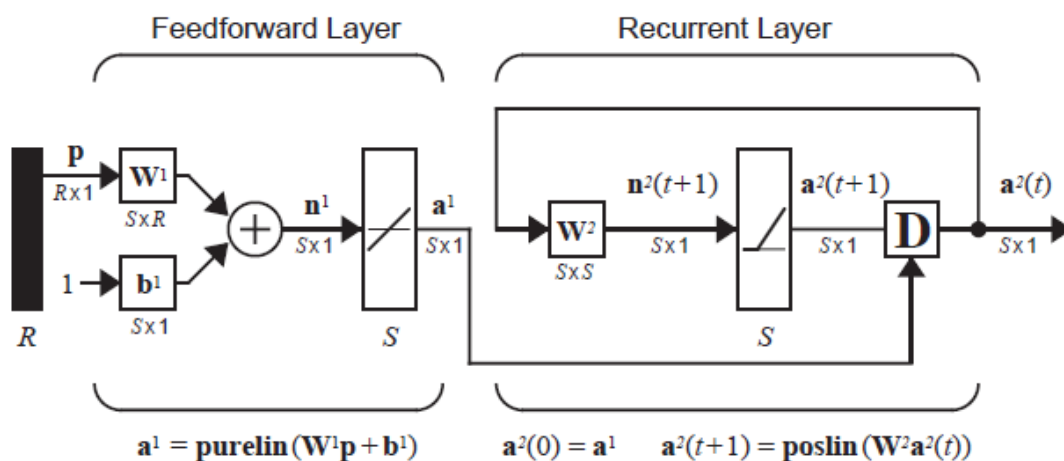


Figure 3.5 Hamming Network

### Capa Feed Forward

Esta capa calcula la correlación o producto interno de cada uno de los vectores prototipo y el patrón de entrada. Con este objetivo, las filas de la matriz de pesos  $\mathbf{W}^1$  serán uno de los vectores prototipo.

$$a^1 = W^1 p + b^1 = \begin{bmatrix} p_1^T p + R \\ p_2^T p + R \\ \vdots \\ p_Q^T p + R \end{bmatrix}$$

### Capa Recurrente

Las neuronas de esta capa se inicializan con la salida de la capa **feed forward**. En esta capa las neuronas compiten entre ellas para determinar a la ganadora. Al final de la competencia, sólo una neurona de esta capa tendrá un valor diferente de cero. Está es la neurona ganadora e indica a que clase pertenece el vector de entrada. (Hagan)

Las ecuaciones que describen esta capa son:

$$a^2(0) = a^1 \text{ --- (1)}$$

$$a^2(t+1) = \text{poslin}(W^2 a^2(t)) \text{ --- (2)}$$

$$W^2 = \begin{bmatrix} 1 & -\varepsilon \\ -\varepsilon & 1 \end{bmatrix} \text{ --- (3)}$$

donde:

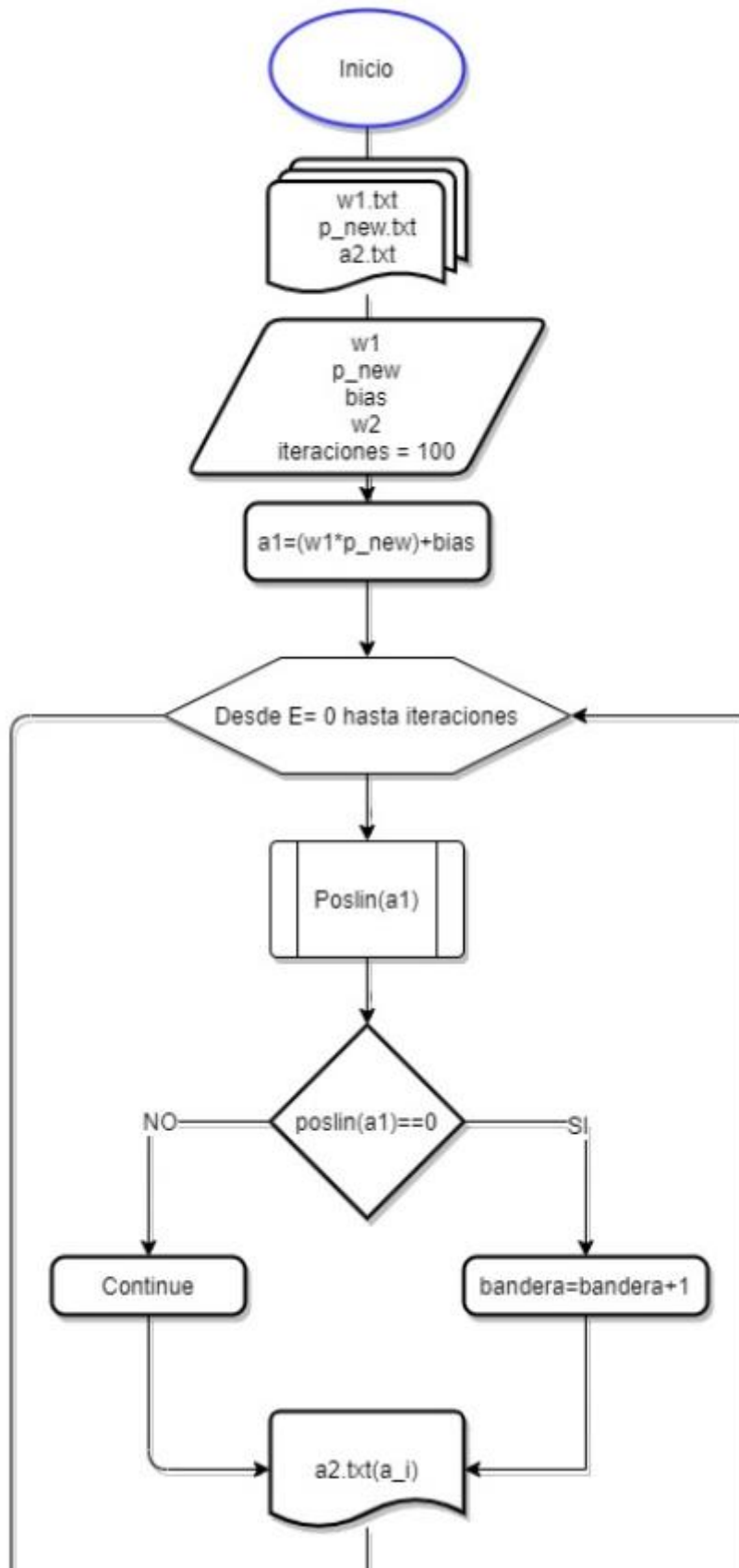
$\varepsilon$  es un valor menor a  $\frac{1}{s-1}$

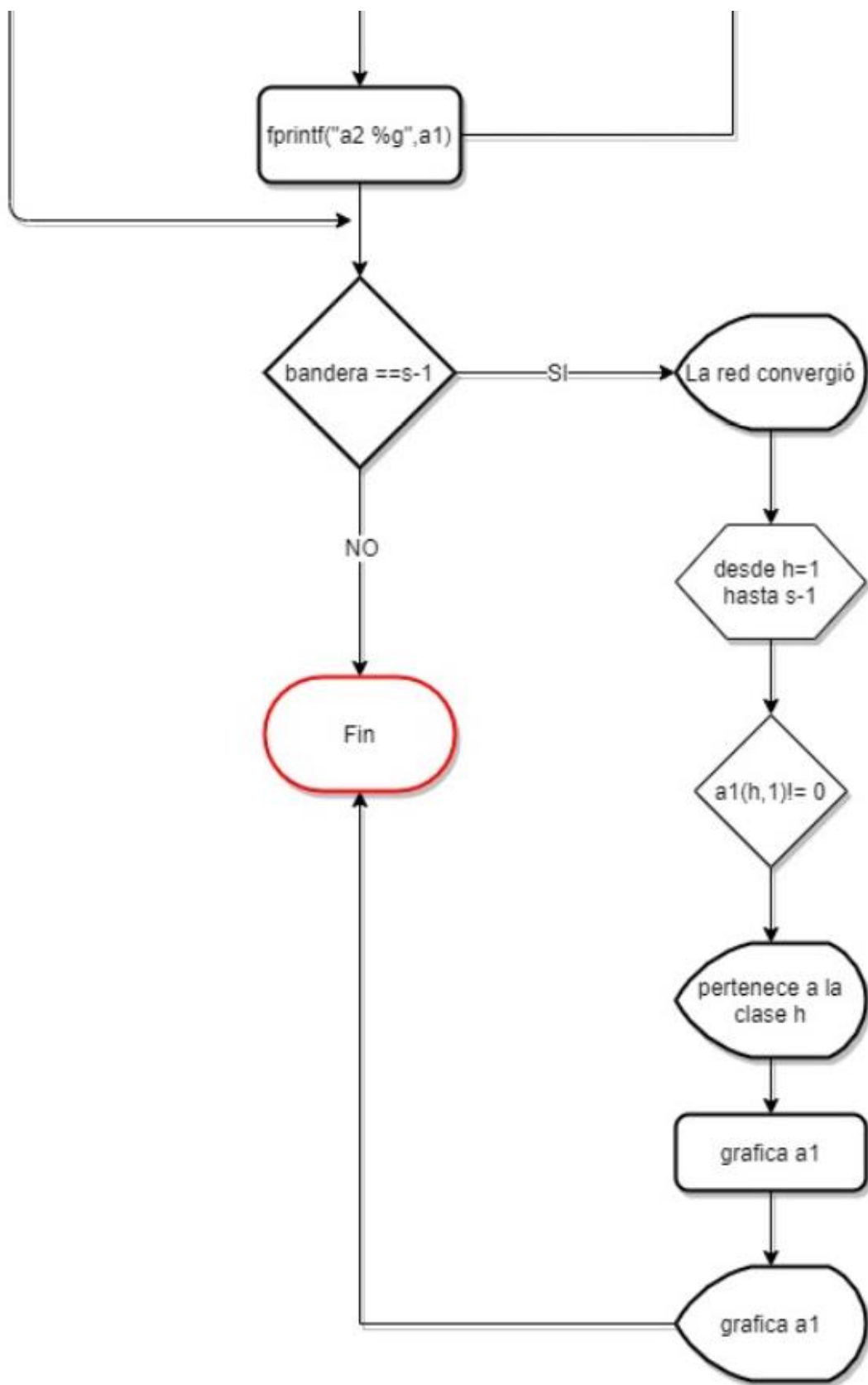
Nota: En genral está matriz tiene valor uno en la diagonal principal y  $-\varepsilon$  en el resto de la matriz.

Para nuestro ejemplo, las ecuaciones (1) y (2) quedan como :

$$a^2(0) = a^1; a^2(t+1) = \text{poslin} \left( \begin{bmatrix} 1 & -\varepsilon \\ -\varepsilon & 1 \end{bmatrix} \begin{bmatrix} a_1^2(t) \\ a_2^2(t) \end{bmatrix} \right)$$

## Diagrama de Flujo:





## Experimentos.

### Experimento 1:

El primer ejemplo para analizar en la practica será el siguiente caso: 3 clases con  $R=4$ .

Con los siguientes valores de entrada:

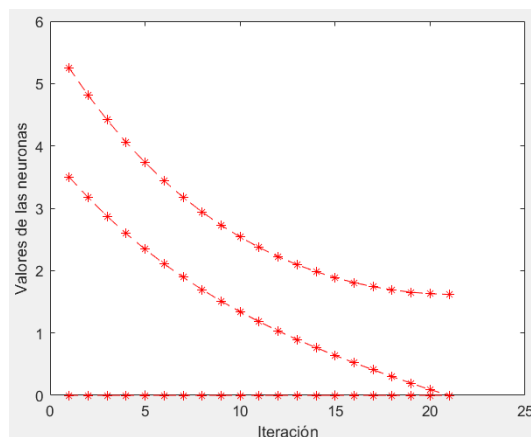
p_new.txt	+1	w1.txt
1 -1	1 1 1 -1 1	
2 1	2 -1 -1 1 -1	
3 1	3 1 1 -1 -1	
4 -1	4	

Al ejecutar nuestro programa podemos ver que nuestra red convergió al valor de la clase número 2 ya que es al que más se parece nuestro los valores en consola.

```
>> practica_2
2      6      4      La red de Hamming convergió\n
Pertenece a la clase: 2
Neuronas:
      0
    1.6181
      0
>>
```

practica_2_m	w1.txt	a2.txt
1		
2 0	5.25	3.5
3 0	4.8125	3.17188
4 0	4.41602	2.87109
5 0	4.05713	2.59509
6 0	3.73274	2.34152
7 0	3.44005	2.10823
8 0	3.17652	1.89322
9 0	2.93987	1.69469
10 0	2.72803	1.51095
11 0	2.53917	1.34045
12 0	2.37161	1.18175
13 0	2.22389	1.03352
14 0	2.0947	0.894529
15 0	1.98289	0.76361
16 0	1.88743	0.63968
17 0	1.80747	0.521715
18 0	1.74226	0.408748
19 0	1.69117	0.299857
20 0	1.65368	0.194159
21 0	1.62941	0.0908034
22 0	1.61806	0

Como logró converger nuestra red graficaremos nuestros valores, además observando los resultados podemos ver que nuestro vector de entrada realmente si se aproxima demasiado a la clase número dos de entrada.



## Experimento 2:

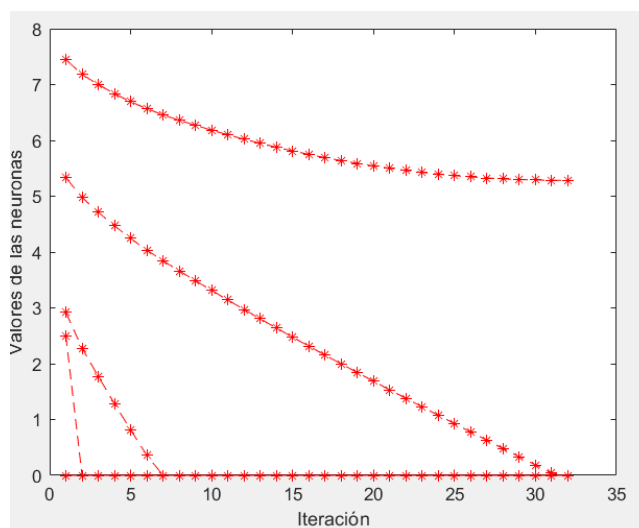
Ahora intentaremos hacer que nuestra red neuronal trabaje con datos y clases más complejos comparados con el experimento número los valores de entrada en este caso serán: 5 clases con  $R=6$ .

p_new.txt		+1	w1.txt
1	-1	1	1 1 -1 1 -1 1
2	1	2	-1 -1 1 -1 1 1
3	-1	3	1 1 -1 -1 -1 -1
4	-1	4	1 1 1 1 1 1
5	1	5	-1 -1 -1 -1 -1 -1
6	-1		

```
>> practica_2
4      6      8      4      8      La red de Hamming convergi n
Pertenece a la clase: 3
Neuronas:
      0
      0
5.2849
      0
      0
```

Como podemos observar en nuestra consola, el valor converge a la clase n mero 3 lo cual hace que las otras clases tengan un valor de 0, pero si observamos nuestra gr fica y archivo de salida podemos inferir que a mayor n mero de clases y caracter sticas a nuestra red le costar  m s converger.

Ahora anexo la la gr fica calculada por el programa no anexo la salida debido a que el archivo es algo grande, pero se anexa en el panel de entregables con el nombre de "a2\_2"



### Experimento 3:

Por último, pondremos a prueba a nuestra red neuronal con un ejercicio más complicado el cual constará de: *8 clases con  $R=8$* .

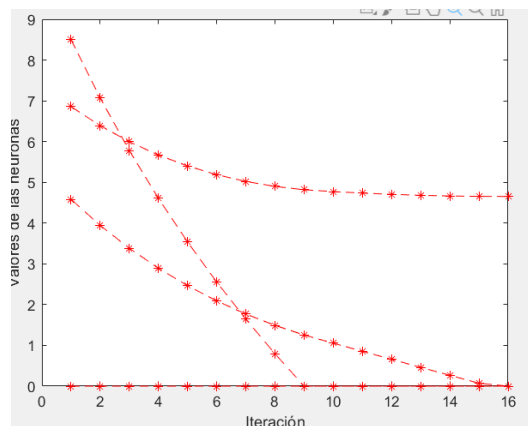
Se anexan los valores de entrada de nuestro vector  $W$  y  $p$  para el último experimento de la siguiente manera:

p_new.txt	+	practica_2.m	w1.txt	a2.b
1 -1		1 1 1 -1 1 -1 1 1 1		
2 1		2 -1 -1 1 -1 1 1 1 1		
3 -1		3 1 1 -1 -1 -1 -1 -1 1		
4 -1		4 1 1 1 1 1 1 -1 -1		
5 1		5 -1 -1 -1 -1 -1 -1 1 -1		
6 -1				
7 -1				
8 1				

Ahora si comparamos nuestros 2 vectores nos manda la respuesta con la clase 3 la es la clase que más se parece ya que solo hay 2 bits de diferencia

```
>> practica_2
4      6      8      4      8      La red de Hamming convergió\n
Pertenece a la clase: 3
Neuronas:
0
0
5.2849
0
0
```

Por último, en la práctica se anexan la imagen de la gráfica la cual tuvo un total de 16 iteraciones





## Discusión:

Al analizar con más detalle nuestros experimentos podemos observar que la red converge de una manera más rápida cuando nuestros valores de bits tienen una mayor discrepancia, es decir que se parecen menos a nuestro valor de entrada, además de que a cada clase le toco un valor de iteraciones similar, sin contemplar el número de clases o valores de R.

## Conclusiones

Haber elaborado esta práctica me gustó bastante, ya que pude aprender a mayor detalle el funcionamiento de las redes feed forward y recurrentes, además, de que al llevarlo a práctica noté las diferencias de implementación y de procesamiento de datos. Ahora cuando las combinamos podemos tener resultados bastantes precisos aunque depende del problema de la precisión que tendremos en los resultados, ahora, en este ejemplo en particular puedo rescatar que en la primera capa la red hace una aproximación de los valores de entrada a la red más parecida, para que al final la segunda capa pueda facilitarse la vida clasificando los resultado y logre converger.

## Bibliografía

Hagan, M. T. (s.f.). Neural Network Design. En M. T. Hagan, *Neural Network Design* (pág. 1010).

Pereira, U. T. (2000). <http://www.medicinaycomplejidad.org/pdf/redes/Competitivas.pdf>.

Obtenido de medicinaycomplejidad.