Red de Hopfield

.

1) Utilizando el comando de Matlab << **loadlab1** >>, almacenar los patrones de los dígitos del 0 al 9 en vectores p0, p1, p2, p9.

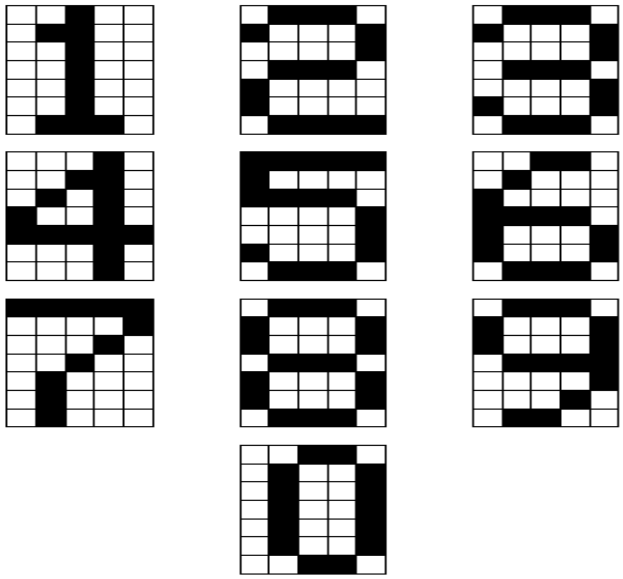

Imagen 1

2) Utilizar el comando << **disppat(p1)** >> para visualizar los patrones de 10x10 bits, en el caso del ejemplo el patrón 1.

3) Estudiar el comportamiento de una red de Hopfield cuando se intenta almacenar estos patrones. Comenzar almacenando 1, 2 ó 3 patrones de su elección y comprobar la estabilidad de los mismos.

a) Generar la matriz de pesos sinápticos utilizando el comando <<**W=genpesos ([p1 p3]) ;**>> (por ejemplo, para crear una matriz de pesos para los patrones 1 y 3).

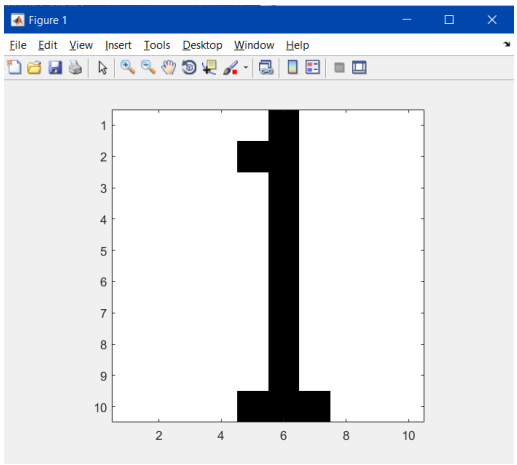

Imagen 2

b) Probar la estabilidad de cada patrón usando la función << **plothopd (W, betai, p1, etai, niter)** >> que mostrar la evolución de la red. "niter" es el número de iteraciones y un valor razonable es alrededor de 300. Compruebe que el inverso de un patrón (por ejemplo, usando el patrón **-p1**) es también estable.

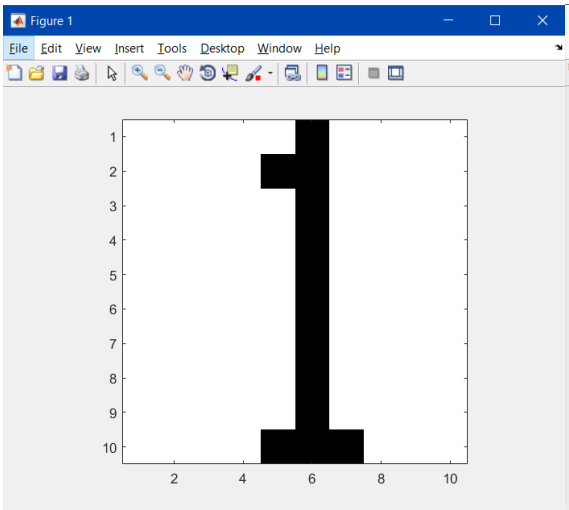

Imagen 3. Función con entrada p1

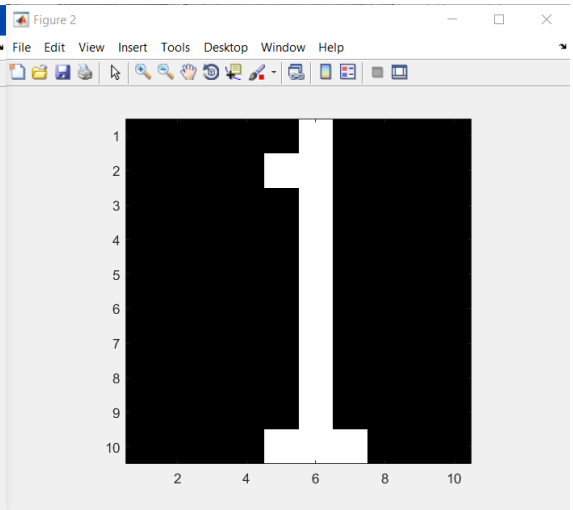

Imagen 4.Función con entrada -p1

Como vemos en las imágenes, la red se estabiliza, sin embargo, con la entrada p1 nos sale el patrón1 normal y con la entrada -p1, nos sale ese mismo pero inverso.

c) Mostrar qué sucede con patrones no almacenados. Probar con patrones aleatorios, que pueden ser generados usando << **pal=sign (rand (100,1)-0.5) ;>>**.

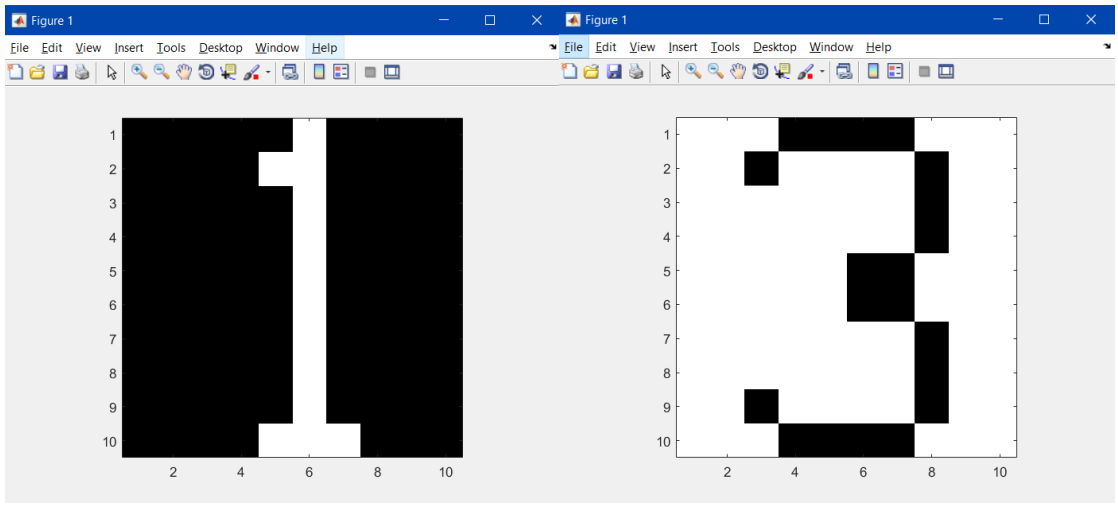

Imagen 5

Imagen 6

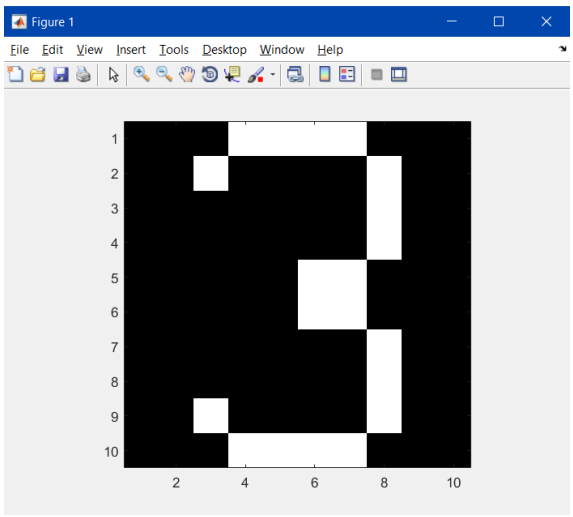

Imagen 7

En estas se han probado patrones aleatorios, los cuales la mayoría han llegado a un estado estable, independientemente si es el patrón normal o inverso. Sin embargo, en las pruebas no se ha llegado a obtener alguno en el cual no haya llegado a un estado estable.

Pregunta: ¿Cuántos patrones logra almacenar correctamente esta red? ¿Qué sucede si presentamos un patrón que no fue aprendido? Describa lo observado.

Correctamente sólo se puede almacenar dos patrones, los cuales deben ser lo mas ortonormales posibles entre sí, es decir, que al multiplicarlos de valor 0. Si presentamos cualquier patrón que no haya sido aprendido, intentara que coincida con alguno que se haya aprendido. En esta práctica el p0, p2 y p4 da como resultado el patrón p3, el resto da el patrón p1.

4. Tolerancia de la red de Hopfield frente al ruido.
Almacenar en la red 2 patrones de su elección, con la menor correlación posible entre ellos. **¿Por qué?**
Esto es para que los patrones sean lo más ortonormal posible, los cuales hace que esta red funcione mejor que si no lo fueran. Para este caso utilizare los patrones p1 y p8.

Añadir un nivel de ruido entre 0 y 1 y **comentar** los resultados analizando si la red es capaz de recuperar los patrones almacenados o no de acuerdo con el nivel de ruido elegido. (Ejemplo: para generar un patrón p1 con nivel de ruido igual a 0.2 utilizar el comando <<**p1r=noisevector (p1, 0.2)** >>)
Comentar los resultados obtenidos.

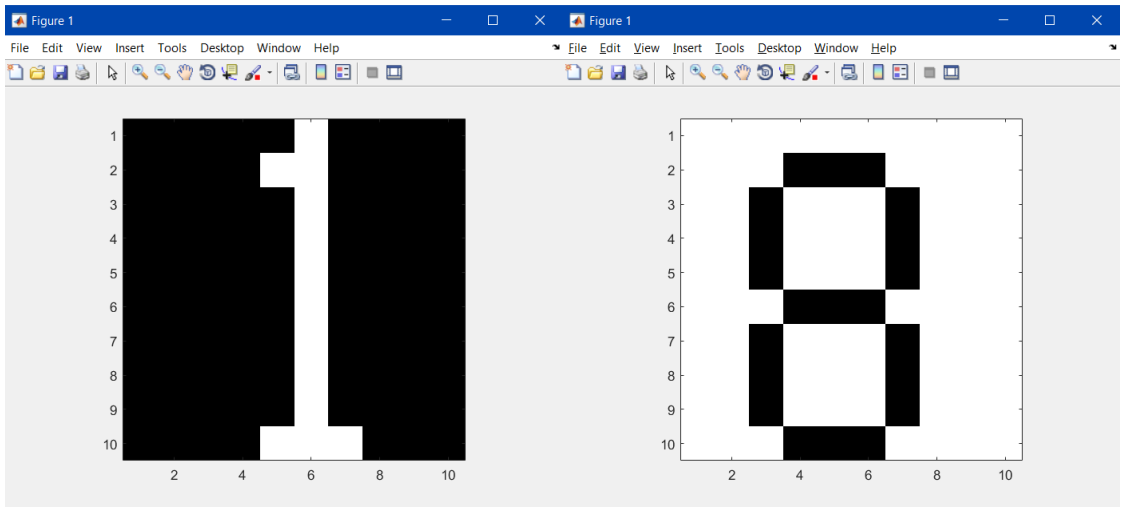

Imagen 8. Resultado de patrón 1 con ruido

Imagen 9. Resultado de patrón 8 con ruido

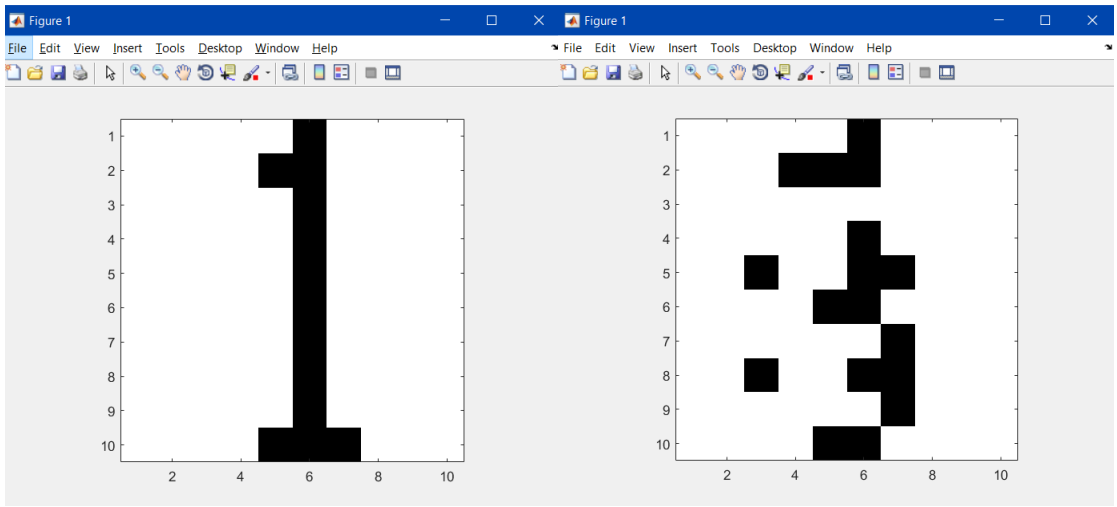

Imagen 10. Resultado de patrón 0 con ruido

Imagen 11. Resultado de cualquier entrada (menos p1 y p8) aleatorio o patrón con ruido

Como vemos, esta nueva red es algo inmune a los ruidos. Hay casos en que con patrones diferentes a p1 y p8 no llega a estabilizar la red y nos sale un resultado como la Imagen 11. También depende el nivel de ruido, ya que, si es bajo, funciona la mayor parte bien, pero si es alto, ya no funciona correctamente porque falla normalmente o da otro patrón al que es insertado y esta guardada en la red.

5. Una forma de mejorar la capacidad de almacenamiento de patrones en la red de Hopfield es utilizar patrones menos correlados, siendo el caso óptimo tener patrones ortogonales.

Utilizar en Matlab el comando sum **(p1. *p3)** para medir la correlación entre patrones y compruebe que los patrones originales (p0 …p9) son altamente correlados.

```
>> sum(p1.*p3)

ans =

    58
```

Como vemos en la imagen los patrones p1 y p3 no son para nada ortonormales, y, por lo tanto, están muy correlacionados.

Imagen 12. Correlación entre p1 y p3

Editor - D:\Users\franc\Google Drive\Ingenieria de Informatica\Curso2018-19\Modelos computacionales\Practica 2\Prac5.m

Prac5.m

```
1 -   val=[p0 p1 p2 p3 p4 p5 p6 p7 p8 p9];
2 - for i=1:10
3 -     for j=i+1:10
4 -         x=val(:,i);
5 -         y=val(:,j);
6 -         fprintf('p%u y p%u = %u \t',i-1,j-1,sum(x.*y));
7 -     end
8 -     fprintf('\n');
9 - end
```

Command Window

```
>> Prac5
p0 y p1 = 46    p0 y p2 = 36    p0 y p3 = 44    p0 y p4 = 38    p0 y p5 = 48    p0 y p6 = 44    p0 y p7 = 48    p0 y p8 = 42    p0 y p9 = 58
p1 y p2 = 50    p1 y p3 = 58    p1 y p4 = 44    p1 y p5 = 58    p1 y p6 = 58    p1 y p7 = 62    p1 y p8 = 52    p1 y p9 = 52
p2 y p3 = 76    p2 y p4 = 46    p2 y p5 = 40    p2 y p6 = 40    p2 y p7 = 44    p2 y p8 = 34    p2 y p9 = 34
p3 y p4 = 58    p3 y p5 = 44    p3 y p6 = 48    p3 y p7 = 48    p3 y p8 = 42    p3 y p9 = 46
p4 y p5 = 42    p4 y p6 = 46    p4 y p7 = 62    p4 y p8 = 40    p4 y p9 = 52
p5 y p6 = 68    p5 y p7 = 64    p5 y p8 = 74    p5 y p9 = 54
p6 y p7 = 52    p6 y p8 = 82    p6 y p9 = 42
p7 y p8 = 54    p7 y p9 = 54
p8 y p9 = 40
```

Imagen 13. Correlación dos a dos entre todos los patrones.

Como se ve en esta imagen entre si ningún patrón es lo mas cercano a lo ortonormal, por lo tanto, todos están correlacionados con todos. Esto nos quiere decir que estos patrones no sirven para hacer una correcta red de Hopfield.

6. A fin de obtener patrones con una menor correlación usaremos una transformación matricial de rotación (magicrot.m).
Ejecutar la función << **loaduncorrpat** >> y visualizar algunos de los patrones que estarán guardados en: **pu0, pu1, pu2, …, pu9.**
Comprobar que la correlación entre patrones ha disminuido de forma considerable.
Dé ejemplos de valores de la correlación obtenida entre los patrones originales y estos nuevos patrones:

Patrones originales: Ver Imagen 13

Patrones rotados:

```
Command Window
>> Prac6
pu0 y pu1 = -8.000000e+00   pu0 y pu2 = -8.000000e+00   pu0 y pu3 = -4.000000e+00   pu0 y pu4 = 1.000000e+01   pu0 y pu5 = -2.000000e+00   pu0 y pu6 = 6.000000e+00   pu0 y pu7 = 4.000000e
pu1 y pu2 = -8.000000e+00   pu1 y pu3 = 4.000000e+00   pu1 y pu4 = -1.400000e+01   pu1 y pu5 = 6.000000e+00   pu1 y pu6 = -1.000000e+01   pu1 y pu7 = 9.436896e-14   pu1 y pu8 = -8.000000
pu2 y pu3 = 1.600000e+01   pu2 y pu4 = -2.000000e+00   pu2 y pu5 = -3.400000e+01   pu2 y pu6 = 1.400000e+01   pu2 y pu7 = -1.200000e+01   pu2 y pu8 = -2.122746e-13   pu2 y pu9 = -8.000000
pu3 y pu4 = 2.000000e+00   pu3 y pu5 = -2.000000e+00   pu3 y pu6 = 6.000000e+00   pu3 y pu7 = 8.000000e+00   pu3 y pu8 = -8.000000e+00   pu3 y pu9 = -1.203482e-13
pu4 y pu5 = 8.000000e+00   pu4 y pu6 = 4.130030e-14   pu4 y pu7 = 2.000000e+00   pu4 y pu8 = 2.000000e+00   pu4 y pu9 = -1.400000e+01
pu5 y pu6 = 4.529710e-14   pu5 y pu7 = 2.000000e+00   pu5 y pu8 = -1.000000e+01   pu5 y pu9 = -1.400000e+01
pu6 y pu7 = 2.000000e+00   pu6 y pu8 = 6.000000e+00   pu6 y pu9 = -2.000000e+00
pu7 y pu8 = -2.460254e-13   pu7 y pu9 = -4.000000e+00
pu8 y pu9 = 4.000000e+00

pu0 y pu8 = 1.200000e+01    pu0 y pu9 = -8.000000e+00
pu1 y pu9 = -4.000000e+00
```

Imagen 14. Correlación dos a dos entre todos los patrones nuevos.

Como vemos en la imagen, los nuevos patrones son casi ortonormales entre sí, por lo tanto, están menos correlacionados que los patrones originales.

7. Estudiar la capacidad de almacenamiento de la red usando estos nuevos patrones. ¿Es ahora mejor? Sí, es mucho mejor
(Pruebe la estabilidad de la red con los patrones almacenados y con patrones contaminados con cierto nivel de ruido)
¿Cuántos patrones consigue almacenar correctamente? Podemos almacenar todos los patrones, sin embargo, esto le hace más sensible a ruidos muy altos. En contraste, con un par de patrones funciona muchísimo mejor y es casi inmune al ruido alto.
(Ayuda: Utilice el comando **"plothopduncorr (W, betai, noisevector(pu4,0.2), etai,50, R)"** para comprobar la recuperación de patrones contaminados con ruido)

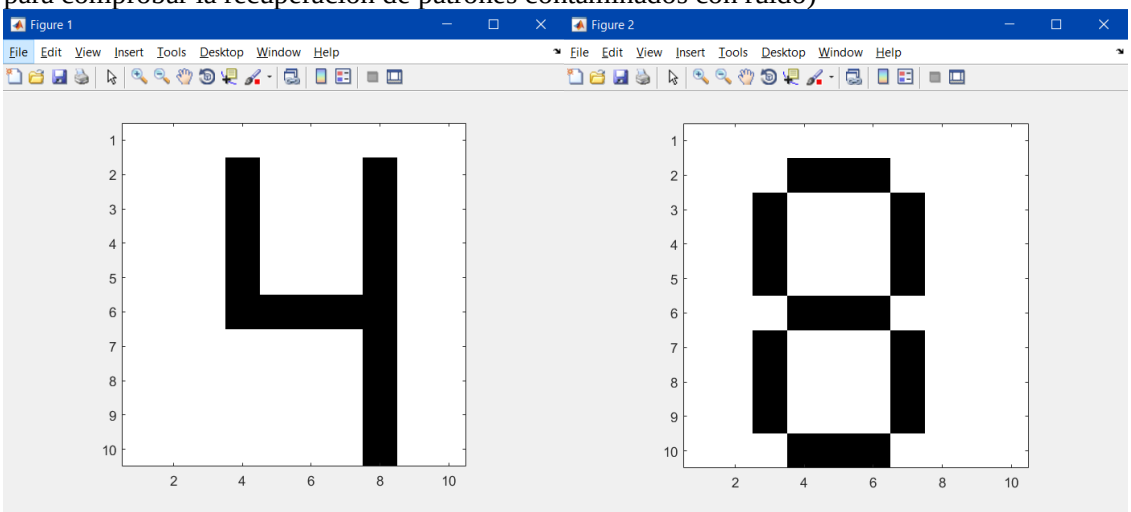

Imagen 15. Resultado de patrón pu4 con ruido

Imagen 15. Resultado de patrón pu8 con ruido

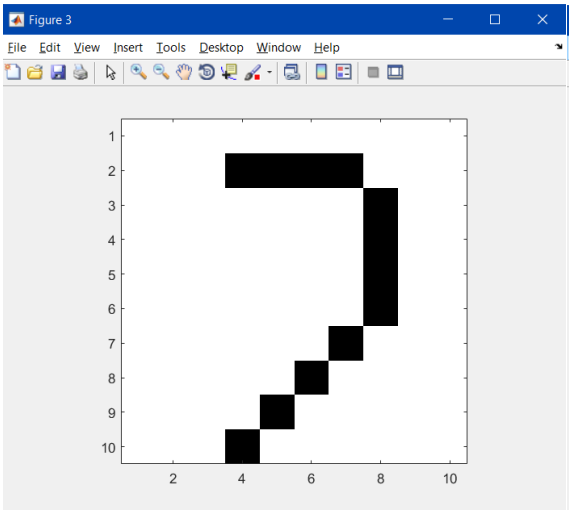

Imagen 16. Resultado de patrón pu7 con ruido

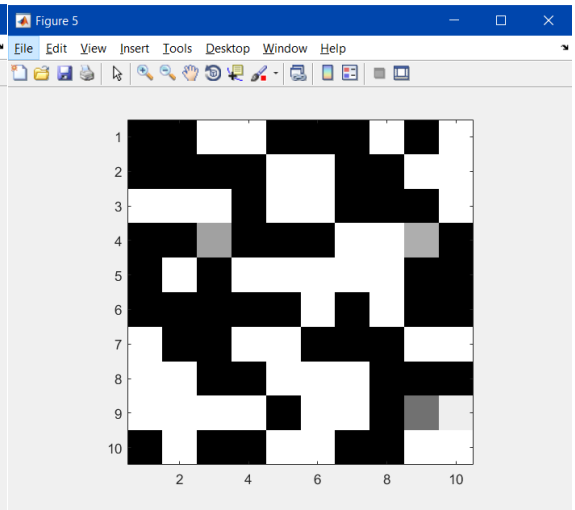

Imagen 17. Resultado de cualquier entrada aleatorio o patrón con ruido muy alto

8. De acuerdo con el valor de la capacidad teórica del modelo de Hopfield ¿Cuántos patrones cree que podrá almacenar esta red?

Nos dice un teorema que la capacidad máxima esta acotada con c=1/(4*ln(N)) y que la capacidad se calcula con c=p/N, siendo N el numero de neuronas y p el numero de patrones almacenados, pues nos quedaría que para calcular p:

$$\frac{p}{N} = \frac{1}{4lnN} \Rightarrow p = \frac{N}{4lnN}$$

Y como el numero de neuronas es igual que al numero de entradas y hay 100 entradas en cada patrón, pues p=100/(4*ln(100))=5.428, que aproximado serian 5, por lo que se podría almacenar teóricamente 5 patrones en la red.