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.



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.

1. 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).
2. 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.

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

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

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?**

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 al 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.**

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.

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: …………. Patrones rotados ………………

7. Estudiar la capacidad de almacenamiento de la red usando estos nuevos patrones. ¿Es ahora 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? …….

(Ayuda: Utilice el comando **“plothopduncorr(W,betai,noisevector(pu4,0.2),etai,50,R)”** para comprobar la recuperación de patrones contaminados con ruido)

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