

MODELOS COMPUTACIONALES


PRÁCTICA 4: MEMORIA ASOCIATIVA BIDIRECCIONAL

- 1) Diseña un script denominado BAM1.m que asocie los patrones vistos en las transparencias de clase, es decir, el patrón $[1 \ 1 \ 1; -1 \ 1 \ -1; -1 \ 1 \ -1] \rightarrow [1 \ -1 \ -1]$ y el $[1 \ -1 \ -1; 1 \ -1 \ -1; 1 \ 1 \ 1] \rightarrow [-1 \ -1 \ 1]$. Ayúdate de la función sign para la regla de actualización
 - a) Utiliza el comando imagesc y reshape para dibujar el patrón de entrada y el reconocido. (ver Nota 1)
 - b) Explica en el código para qué sirve cada una de las variables que utilices.

- 2) A partir del anterior script, crea uno que se llame BAM2.m que use como patrón a memorizar las matrices que hay en el fichero Matrices.zip en el campus virtual. Se debe asociar así: barco \rightarrow TextoBarco y a coche \rightarrow TextoCoche (ver Nota 1)
 - a) Si le introduces el patrón barco, ¿se estabiliza en el patrón correcto?
 - b) Prueba a añadir ruido gaussiano al patrón barco y úsala como entrada ¿La red se estabiliza en el patrón correcto sin ruido? (ver Nota 2)
 - c) Modifica el script para que muestre la entrada y el patrón reconocido, ayúdate de este código:

```
subplot(3,1,1)
imshow(reshape(Entrada,size(barco,1),size(barco,2)))
subplot(3,1,2)
imshow(reshape(salidaY,size(textoBarco,1),size(textoBarco,2)))
subplot(3,1,3)
imshow(reshape(salidaX,size(barco,1),size(barco,2)))
```

donde Entrada es el vector de entrada, salidaY el estado de la segunda capa de neuronas y salidaX el estado de la primera capa.

- d) Guarda la salida mediante el comando gráfico  y pégala en el documento de respuestas.
 - e) Explica en el código para qué sirve cada una de las variables que utilices.
- 3) Sube los scripts BAM1.m y BAM2.m

*Nota 1: Utiliza la función reshape para convertir de vector a matriz o viceversa. Por ejemplo:

```
B=Reshape(A, 1,30*40) %convierte la imagen A de 30x40 píxeles a un vector de 1x(30*40)
```

```
C=Reshape(B, 30, 40) %convierte el vector B a una matriz de 30 por 40
```

*Nota 2: Para añadir ruido gaussiano a una matriz usa la función imnoise, por ejemplo:

```
matrizBipolarGausiano=imnoise(VectorEntradaX(1,:), 'gaussian', 0, 0.5)*2-1;
```

Recuerda que debes comprobar que los valores de entrada que estás usando están en el intervalo $[-1 \ 1]$, imnoise devuelve una matriz en el intervalo $[0 \ 1]$, por eso se multiplica por 2 y se resta 1 en el código de ejemplo.