

# Práctica 7: Memorias Asociativas Bidireccionales

## Modelos de la Computación

Emilio Gomez Esteban

### 1. Enunciado de la práctica

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:
  - Utiliza el comando `imagesc` y `reshape` para dibujar el patrón de entrada y el reconocido (ver Nota 1).
  - 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).

- Si le introduces el patrón `barco`, ¿se estabiliza en el patrón correcto?
- 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).
- Modifica el script para que muestre la entrada y el patrón reconocido, ayúdate de este código:

```
1 subplot(3,1,1)
2 imshow(reshape(Entrada,size(barco,1),size(barco,2)))
3 subplot(3,1,2)
4 imshow(reshape(salidaY,size(textoBarco,1),size(
    textoBarco,2)))
5 subplot(3,1,3)
6 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.

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.

## 2. Resolución de la práctica

### Ejercicio 1

Implementación del script `BAM1.m` con comentarios en el código que dibuja el patrón de entrada y el reconocido:

```
1 clear all;
2
3 % Numero maximo de epocas para ejecutar el algoritmo
4 epocMax = 21;
5
6 % Matriz de patrones de entrada (cada fila es un patron)
7 X(1,:) = [1 1 1 -1 1 -1 -1 1 -1]; % Primer patron de entrada
8 X(2,:) = [1 -1 -1 1 -1 -1 1 1 1]; % Segundo patron de entrada
9
10 % Matriz de patrones de salida asociados a los patrones de
    entrada
11 Y(1,:) = [1 -1 -1]; % Salida asociada al primer patron
12 Y(2,:) = [-1 -1 1]; % Salida asociada al segundo patron
13
14 % Calculo de la matriz de pesos sinapticos utilizando la
    regla de Hebb
15 w = X' * Y; % Producto entre la transpuesta de X y Y
16
17 % Inicializacion de matrices para almacenar los estados de
    entrada y salida
18 S = zeros(size(X,2), epocMax); % Matriz para estados de
    entrada
19 S2 = zeros(size(Y,2), epocMax); % Matriz para estados de
    salida
20
21 % Inicializacion del vector de entrada (puede ser uno de los
    patrones en X)
22 sinit = [1 1 1 -1 1 -1 -1 1 -1]; % Patron inicial de entrada
23
24 % Estado inicial de salida, calculado con el patron inicial y
```

```

    la matriz de pesos
25 sinit2 = sign(sinit * w);
26
27 % Almacenar los estados iniciales en las matrices S y S2
28 S(:,1) = sinit;      % Primer columna: patron inicial de
    entrada
29 S2(:,1) = sinit2;    % Primer columna: patron inicial de salida
30
31 % Iteraciones para actualizar los estados de entrada y salida
32 for epoc = 2:1:epocMax
33     % Actualizar el estado del patron de entrada basado en el
        estado de salida previo
34     S(:,epoc) = sign(w * S2(:,epoc-1));
35
36     % Actualizar el estado del patron de salida basado en el
        estado de entrada actual
37     S2(:,epoc) = sign(S(:,epoc)' * w);
38
39     % Verificar la convergencia: si los estados no cambian
        entre epocas consecutivas
40     if (sum(S(:,epoc) == S(:,epoc-1)) == size(X,2)) && (sum(
        S2(:,epoc) == S2(:,epoc-1)) == size(Y,2))
41         % Mostrar los estados finales convergidos
42         S(:,epoc) % Estado convergido del patron de entrada
43         S2(:,epoc) % Estado convergido del patron de salida
44         epoc      % Numero de epocas necesarias para
            converger
45
46         % Asumiendo que las entradas son imagenes de 3x3
            pixeles para X y 1x3 para Y
47         rows_input = 3; % Numero de filas de la matriz de
            entrada
48         cols_input = 3; % Numero de columnas de la matriz de
            entrada
49
50         % Convertir el vector de entrada inicial sinit en
            matriz
51         input_matrix = reshape(sinit, rows_input, cols_input)
            ;
52
53         % Determinar las dimensiones del patron reconocido
54         % En este caso, las dimensiones corresponden a las
            dimensiones de entrada.
55         recognized_matrix = reshape(S(:, epoc-1), rows_input,
            cols_input);
56

```

```

57     % Visualizacion de los patrones
58     figure;
59
60     % Mostrar el patron de entrada inicial
61     subplot(1, 2, 1);
62     imagesc(input_matrix);
63     colormap gray;
64     axis equal;
65     title('Patron de entrada inicial');
66
67     % Mostrar el patron reconocido
68     subplot(1, 2, 2);
69     imagesc(recognized_matrix);
70     colormap gray;
71     axis equal;
72     title('Patron reconocido');
73
74     return % Terminar el algoritmo si hay
           convergencia
75 end
76 end

```

## Ejercicio 2

Si le introducimos el patrón barcoo el patrón barco, la red se estabiliza al patrón correcto:



(a) Ejemplo Barco



(b) Ejemplo Coche

Para añadir ruido, utilizamos las siguientes líneas de código:

```
1 % Anadir ruido gaussiano al patron de entrada "barco"
2 sinit_noisy = imnoise(reshape(X(1,:), size(barco.barco)), '
   gaussian', 0, 0.01); % Patron barco con ruido gaussiano
3 sinit_noisy = sinit_noisy(:)' * 2 - 1; % Convertir de bipolar
   [0, 1] a bipolar [-1, 1]
4
5 % Calcular el estado inicial de salida
6 s2init = sign(sinit_noisy * w);
7
8 % Almacenar los estados iniciales en las matrices
9 S(:,1) = sinit_noisy; % Patron barco con ruido
10 S2(:,1) = s2init; % Estado inicial de salida
```

Si la red converge, se muestra el patrón de entrada reconocido después de eliminar el ruido y el patrón de salida correspondiente reconocido por la red. Si no converge, el programa muestra un mensaje indicando que la red no logró estabilizarse. Si usamos el valor 0.01 como varianza del ruido gaussiano es moderado. Si usamos el valor 0.5 como varianza del ruido gaussiano es mayor.



(c) Ejemplo Barco Ruido 0.01



(d) Ejemplo Barco Ruido 0.5