

1) En el script se hizo lo siguiente.

- Cargo las imágenes.
- Las llevo a 1 y -1.
- Construyo matriz de pesos sinápticos.
- Verifico que haya aprendido las imágenes al menos.

Para esto lo que hago es ver que al hacerlo evolucionar 1 iteración en forma sincrónica, partiendo del estado guardado, el resultado tiene que ser el mismo patrón.

El programa calcula la cantidad de errores en cada uno y los imprime de la siguiente manera:

```
Errores en las imagenes:
```

```
errores =
```

```
0      0      0      0
```

Donde cada cero indica los errores en el aprendizaje de cada una de las 4 imágenes. Puede verse que las ha aprendido.

- Finalmente el programa hace una estadística basada en 1000 iteraciones de, partiendo de un estado inicial aleatorio, ver a donde converge. Para esto lo primero que hace es generar un estado inicial aleatorio. Luego lo hace evolucionar en forma sincrónica hasta que el estado se estabiliza. Luego contabiliza la cantidad de veces que terminó en alguna de las imágenes o en un estado espurio. Dividiendo las cantidades por la cantidad de iteraciones estima la probabilidad de ocurrencia de cada caso.

Un ejemplo de corrida del programa emite el siguiente resultado:

```
Probabilidades de caer en los estados aprendidos
```

```
probabilidades_imagenes =
```

```
0.0630    0.0460    0.1050    0.0690
```

```
Probabilidad de caer en estados espurios
```

```
probabilidad_espurio =
```

```
0.7170
```

Puede observarse que la probabilidad de caer en alguno de los estados espurios es mucho mayor. Por otro lado se ve que la probabilidad de caer en alguna de las imágenes es distinta, para una estimación basada en 1000 iteraciones.

2) En el script se realizó lo siguiente:

- Construyo la matriz de pesos sinápticos basada en un único patrón.
- Lo hago evolucionar en forma sincrónica y guardo el resultado.
- Lo hago evolucionar en forma asincrónica y guardo el resultado.

El programa emite el siguiente resultado:

Resultado evolucion sincronica:

resultado_sync =

1
-1
1
1

Resultado evolucion asincronica

resultado_asinc =

1
1
1
-1

Puede observarse que en la evolución asincrónica el resultado es que cayó en el único estado guardado (En algunas corridas cayó en el inverso del mismo). Sin embargo en la evolución sincrónica, cayó en el estado inicial. Para el caso de la evolución sincrónica, el resultado es que el sistema quedará entre dos estados, sin converger. Puede verse que:

$$\text{signo}(W \cdot \text{estado}) = \text{signo} \left(\begin{bmatrix} +0 & +1 & +1 & -1 \\ +1 & +0 & +1 & -1 \\ +1 & +1 & +0 & -1 \\ -1 & -1 & -1 & +0 \end{bmatrix} \begin{bmatrix} +1 \\ -1 \\ +1 \\ +1 \end{bmatrix} \right) = \begin{bmatrix} -1 \\ +1 \\ -1 \\ -1 \end{bmatrix}$$

Y en la siguiente iteración volverá al estado original:

$$\text{signo}(W \cdot \text{estado}) = \text{signo} \left(\begin{bmatrix} +0 & +1 & +1 & -1 \\ +1 & +0 & +1 & -1 \\ +1 & +1 & +0 & -1 \\ -1 & -1 & -1 & +0 \end{bmatrix} \begin{bmatrix} -1 \\ +1 \\ -1 \\ -1 \end{bmatrix} \right) = \begin{bmatrix} +1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

Se ve que vuelve al estado inicial, en un ciclo infinito, cosa que en la actualización asincrónica no pasa. La razón es que en la evolución asincrónica solo cambia un bit a la vez, hasta que se llega a un mínimo, mientras que en la sincrónica esto puede no ser cierto.

Archivos entregados:

- 'ej_1.m'
- 'ej_2.m'
- 'evolucionar.m'
- 'signo.m'
- 'padrino.bmp'
- 'panda.bmp'
- 'perro.bmp'
- 'v.bmp'