

## Voluntario 2: Modelo de Hopfield de red neuronal

**Usar el algoritmo de Metrópolis utilizado en el modelo de Ising para simular y estudiar el comportamiento emergente de una red neuronal de Hopfield para cualquier temperatura  $T$ .**

Como en el caso del modelo de Ising, consideraremos una red cuadrada bidimensional con  $N \times N = N^2$  nodos y condiciones de contorno periódicas, donde cada nodo de la red  $(i, j)$  (con  $i, j = 1, 2, \dots, N$ ) representa una neurona que está activa ( $s_{i,j} = +1$ ) o inactiva ( $s_{i,j} = 0$ ). En este caso el Hamiltoniano para cada configuración del sistema  $\mathbf{s} = \{s_{i,j}\}$  se expresa como

$$H(\mathbf{s}) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N \omega_{ij,kl} s_{i,j} s_{k,l} + \sum_{i=1}^N \sum_{j=1}^N \theta_{i,j} s_{i,j} \quad (1)$$

donde la interacción entre la neurona  $(i, j)$  y la neurona  $(k, l)$ , es decir  $\omega_{ij,kl}$ , viene dada en términos de  $P$  configuraciones de la red previamente almacenadas  $\xi^\mu = \{\xi_{i,j}^\mu\}$  donde  $\xi_{i,j}^\mu \in \{0, 1\}$  (con  $i, j = 1, 2, \dots, N$  y  $\mu = 1, 2, \dots, P$ ). Estos son los llamados patrones, que a su vez minimizan el Hamiltoniano (1) y corresponden a los mínimos de la energía libre para  $T = 0$ . De esta forma tenemos que,

$$\omega_{ij,kl} = \begin{cases} \frac{1}{N^2} \sum_{\mu=1}^P (\xi_{i,j}^\mu - a^\mu)(\xi_{k,l}^\mu - a^\mu) & , \quad \text{si } (i, j) \neq (k, l), \\ 0 & , \quad \text{si } (i, j) = (k, l), \end{cases} \quad (2)$$

con

$$a^\mu = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \xi_{i,j}^\mu.$$

Las interacciones entre neuronas  $\omega_{ij,kl}$  representan el estado de las sinapsis o conexiones entre neuronas y se denominan pesos sinápticos. Vemos que la red de Hopfield asume interacciones de largo alcance, siendo en particular una red totalmente conectada, pero donde las autoconexiones no están permitidas, es decir  $\omega_{ij,ij} = 0$ , como vemos de la ecuación (2).

Por otro lado, el término  $\theta_{i,j}$  en la ecuación (1) se denomina umbral de disparo y viene dado en términos de los pesos sinápticos,

$$\theta_{i,j} = \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega_{ij,kl}. \quad (3)$$

Por tanto a partir de las ecuaciones (1),(2) y (3) podemos implementar la dinámica del sistema utilizando el algoritmo de Metrópolis dados  $P$  patrones  $\xi^\mu$  (con  $\mu = 1, 2, \dots, P$ ) previamente almacenados.

A continuación se enumeran las actividades a realizar:

1. Ver cómo la red es capaz de recordar un solo patrón almacenado<sup>I</sup> (de al menos  $N = 30$ , es decir, de 900 neuronas) para temperatura<sup>II</sup>  $T = 10^{-4}$  partiendo de (i) una condición inicial aleatoria y (ii) del patrón deformado. Calcular también el solapamiento en función del tiempo (en pasos Monte Carlo) para cuantificar cómo la red se aproxima al patrón a medida que el tiempo aumenta. El solapamiento de la configuración  $\mathbf{s} = \{s_{i,j}\}$  con el patrón  $\xi_{i,j}^\mu$  para un solo patrón ( $\mu = 1$ ) viene dado por:

$$m^1(\mathbf{s}) = \frac{1}{N^2 a^1 (1 - a^1)} \sum_{i=1}^N \sum_{j=1}^N (\xi_{i,j}^1 - a^1)(s_{i,j} - a^1).$$

2. Hacer lo anterior pero para diferentes temperaturas, de manera que se pueda obtener la curva de solapamiento frente a temperatura.
3. Hacer lo mismo que en el punto 1 y 2 pero para varios patrones, partiendo de (i) una condición inicial aleatoria y de (ii) algunos de los patrones deformados. En este caso el solapamiento de la configuración  $\mathbf{s} = \{s_{i,j}\}$  con cada patrón será:

$$m^\mu(\mathbf{s}) = \frac{1}{N^2 a^\mu (1 - a^\mu)} \sum_{i=1}^N \sum_{j=1}^N (\xi_{i,j}^\mu - a^\mu)(s_{i,j} - a^\mu).$$

Utilizar diferentes tipos de patrones prefijados, como por ejemplo los dígitos del 0 al 9, algunas letras o un conjunto de símbolos.

Notad que el solapamiento es una cantidad acotada  $m^\mu(\mathbf{s}) \in [-1, 1]$  siendo  $m^\mu(\mathbf{s}) = 1$  cuando se recupera el patrón perfectamente, es decir,  $s_{i,j} = \xi_{i,j}^\mu, \forall i, j$  ó  $m^\mu(\mathbf{s}) = -1$  cuando recuperamos el antipatrón, es decir, cuando  $s_{i,j} = 1 - \xi_{i,j}^\mu, \forall i, j$  que también corresponde a un mínimo del Hamiltoniano (1), y es lo que se conoce como estado espúreo.

4. Asumiendo que un patrón dado “se recuerda” sin apenas error cuando el solapamiento es mayor que 0.75 ( $m^\mu(\mathbf{s}) > 0.75$ ), calcular cómo decae la recuperación de la memoria en función del número de patrones almacenados, es decir, cómo disminuyen el número de patrones recordados en función del número de patrones almacenados considerando una red de  $N = 20$  (es decir de 400 neuronas) y  $T = 10^{-4}$ . Calcular también la fracción máxima  $\alpha_c = P_c/N^2$  de patrones que la red podrá almacenar de forma que todos los patrones se puedan recordar. En este caso es mejor trabajar con patrones aleatorios.

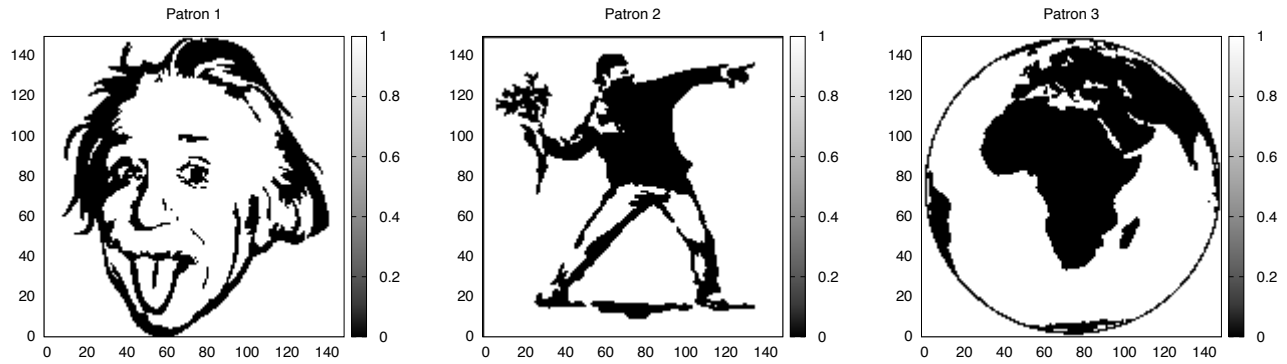
Consejo:

- Para que el programa no vaya demasiado lento a la hora de implementar la dinámica de Metrópolis es recomendable expresar  $\Delta H = H(\mathbf{s}') - H(\mathbf{s})$  de forma inteligente (en el modelo de Ising así lo hemos hecho), de manera que no tengamos que calcular  $H(\mathbf{s})$  y  $H(\mathbf{s}')$  en cada iteración del algoritmo.

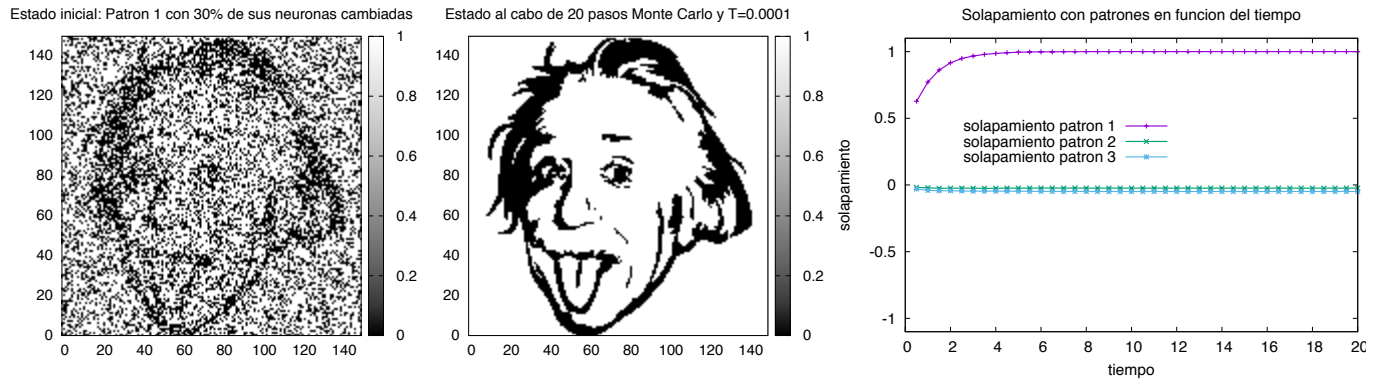
<sup>I</sup>Para almacenar patrones prefijados tales como imágenes de números, letras o símbolos existen diversas herramientas que convierten dichas imágenes en ficheros binarios (de “unos” y “ceros”). Ver por ejemplo este sitio web: <https://www.dcode.fr/binary-image>. Para ello se recomienda usar imágenes en blanco y negro.

<sup>II</sup>No tomamos  $T = 0$  ya que es un valor patológico en el algoritmo de Metrópolis: recordad que la probabilidad de transición de  $\mathbf{s}$  a  $\mathbf{s}'$  viene dada por  $p_{\mathbf{s} \rightarrow \mathbf{s}'} = \min(1, e^{-\Delta H/T})$ , con  $\Delta H = H(\mathbf{s}') - H(\mathbf{s})$ . En este caso  $\mathbf{s}'$  es la configuración que obtenemos cuando elegimos al azar un espín de la configuración  $\mathbf{s}$  (por ejemplo el espín  $(n, m)$  con valor  $s_{n,m}$ ) y cambiamos su valor a  $s_{n,m} = 1 - s_{n,m}$ .

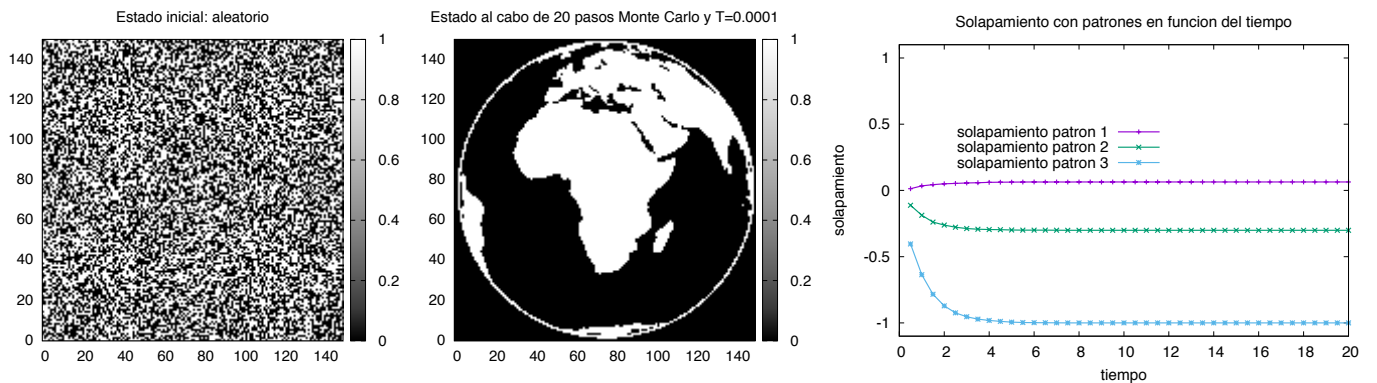
**Ejemplo para 3 patrones en una red de  $150 \times 150 = 22500$  neuronas y  $T = 10^{-4}$**



**Figure 1: Patrones iniciales**



**Figure 2: Evolución del estado inicial (patrón 1 difuminado) al cabo de 20 pasos Monte Carlo**



**Figure 3: Evolución del estado inicial aleatorio al cabo de 20 pasos Monte Carlo**