



**UNIVERSIDAD  
DE GRANADA**

# Modelo de Hopfield de Red Neuronal

Física Computacional: Grado en Física

*Caleb Porter*

*7 de Julio de 2023*

# 1 ÍNDICE

---

2	Introducción .....	3
3	Resultados y Discusión .....	5
3.1	Único Patrón .....	5
3.1.1	Estado Inicial Aleatorio .....	5
3.1.2	Estado inicial: Patrón deformado .....	7
3.2	Varios Patrones .....	10
3.2.1	Estado Inicial Aleatorio .....	10
3.2.2	Estado inicial: Patrón deformado .....	12
3.3	Memoria en Función del Número de Patrones .....	14
4	Conclusiones .....	15
5	Referencias.....	15

## 2 INTRODUCCIÓN

El modelo de Hopfield fue descrito por Shun'ichi Amari en 1972 y popularizado por John Hopfield en 1982.<sup>4</sup> Aunque es basada en el modelo de Ising, un modelo matemático del ferromagnetismo, la red de Hopfield trata de simplificar matemáticamente una neurona, y así, permitir el análisis del comportamiento de redes neuronales de escala grande. Caracteriza el efecto de cambios a unidades individuales sobre una propiedad de la arquitectura neuronal, denominada la energía. La red de Hopfield es diseñada para almacenar patrones para que se puedan recuperar a partir de estados iniciales nublados o parciales. La red es capaz de recordar los patrones por algo en el que podemos pensar como una superficie energética. En la superficie energética, los patrones son unas depresiones y la configuración una partícula que la atraviesa. Mientras el sistema busca una configuración de energía baja, la partícula se cae en las depresiones, que representan mínimas en la energía.<sup>3</sup>

La red se compone de  $N \times N = N^2$  nodos y cada uno de ellos  $(i, j)$  (con  $i, j = 1, 2, \dots, N$ ) pueden estar o activa ( $s_{i,j} = +1$ ), o inactiva ( $s_{i,j} = 0$ ), así representando neuronas. La llamada energía del sistema, aquí representado usando el formalismo Hamiltoniano toma la forma

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

Dentro de esta ecuación, vemos más propiedades de la red. El peso sináptico,  $\omega_{ij,kl}$  representa la conexión entre los nodos, los cuales están conectados con todos los otros nodos, salvo consigo mismo. Esto permite las interacciones de largo alcance.

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

Con

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

Este peso sináptico viene dado en términos de  $P$  patrones  $\xi^{\mu} = \{\xi_{ij}^{\mu}\}$  donde  $\xi_{ij}^{\mu} \in \{0,1\}$ . También son simétricos, es decir que  $\omega_{ij,kl} = \omega_{kl,ij}$ . El término  $a$  en esta ecuación representa la proporción de neuronas activas en los patrones.

Por otro lado, al término  $\theta_{i,j}$  en la ecuación de la energía se le llama umbral de disparo y viene dado por los pesos sinápticos.

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

El *input* que recibe los nodos  $(i, j)$  en la red, como vemos en la ecuación de la energía viene dado por la suma de los pesos de los nodos activos de la red. Eso, en combinación con el umbral de disparo, hace que la red suele caminar hacia un estado igual o parecido a uno de los patrones.

Para llevar este modelo al cabo, vamos a usar el algoritmo de Metropolis. El algoritmo de Metropolis es un algoritmo Montecarlo que usa números aleatorios para cambiar el estado de un sistema. En este

caso, si elegimos un nodo  $(k, l)$  aleatoriamente, calculamos la cantidad  $\Delta H$  que sería el cambio en la energía del sistema si ese nodo  $(k, l)$  cambiara de estado. De allí, calculemos  $p(s \rightarrow s') = \min(1, e^{-\frac{\Delta H}{T}})$ . Si este valor  $p$  es mayor a un número entre 0 y 1 elegido al azar, el cambio se hace, mientras que, si  $p$  es menor que el número aleatorio, el cambio no se hace.

Este método va mucho más rápido usando una ecuación simplificada de  $\Delta H = H' - H$ . Usando las propiedades vistas más arriba, se puede llegar a la siguiente forma, donde  $(k, l)$  es un punto aleatorio que sería el único cambio entre  $H'$  y  $H$ .

$$\Delta H(s) = (1 - 2s_{k,l}) \left( \left( \sum_{i=1, i \neq k}^N \sum_{j=1, j \neq l}^N \omega_{ij,kl} s_{i,j} \right) - \theta_{k,l} \right) \quad (2)$$

## 3 RESULTADOS Y DISCUSIÓN

---

### 3.1 ÚNICO PATRÓN

#### 3.1.1 Estado Inicial Aleatorio

Empezando nuestra investigación del modelo con solo un patrón, veremos los resultados que se salen para diferentes temperaturas. Ejecutamos el programa *hopfield.c* completando 40 pasos Montecarlo y partiendo de un estado inicial aleatorio de la red. Notamos que la semilla que se cambiaba entre las ejecuciones del programa haciendo que el estado inicial aleatorio fuera el mismo a las distintas temperaturas.

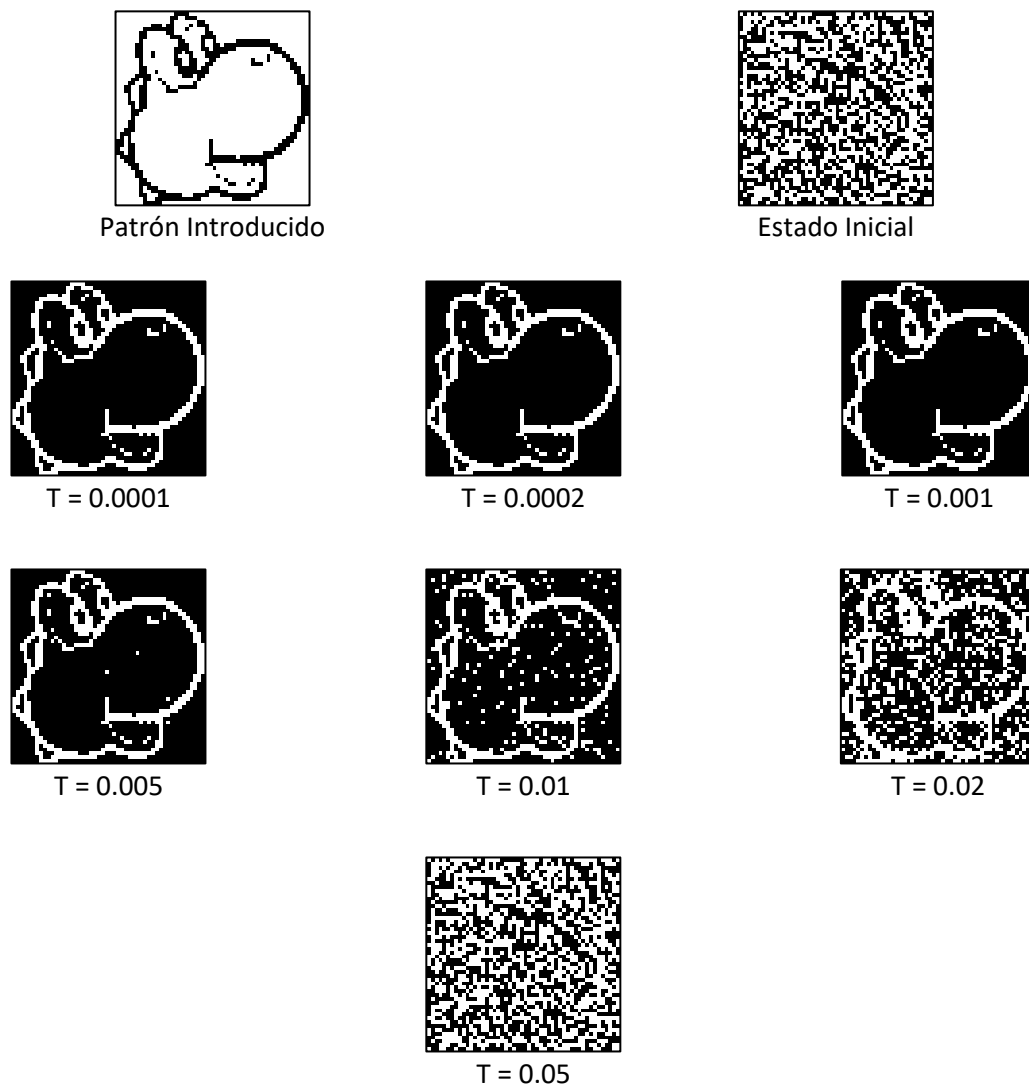


Figura 1: Resultados al cabo de 40 pasos Montecarlo para diferentes temperaturas

Ya sabíamos que el patrón era un mínimo en  $H$ , y esperábamos que el algoritmo condujera hacia ello. Sin embargo, en estos resultados vemos otro mínimo de energía, el antipatrón. El patrón fue

introducido con la figura en negro (ceros) sobre un fondo blanco (unos), y la red conduce a la misma figura, pero con el fondo en negro y la figura blanca. A estos estados los podemos llamar “estados espúreos” (Wikipedia), y aunque se llaman así cualquier mínimo de energía que no sea un patrón almacenado, en este caso nos salen la negación del patrón.

Aparte de los estados espúreos, podemos comenzar a entender la dependencia del modelo en la temperatura. A las temperaturas más bajas, la aproximación al patrón está muy cerca a la perfección. Sin embargo, al subirla, el estado final se va alejando del patrón introducido hasta que se quede sin semejanza con él.

Para poder comparar esta aproximación, o convergencia, sería útil cuantificarla. Eso lo hacemos por definir el solapamiento con el patrón como:

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

El mismo programa también nos calcula este valor para diferentes temperaturas, como se ve en la figura 2.

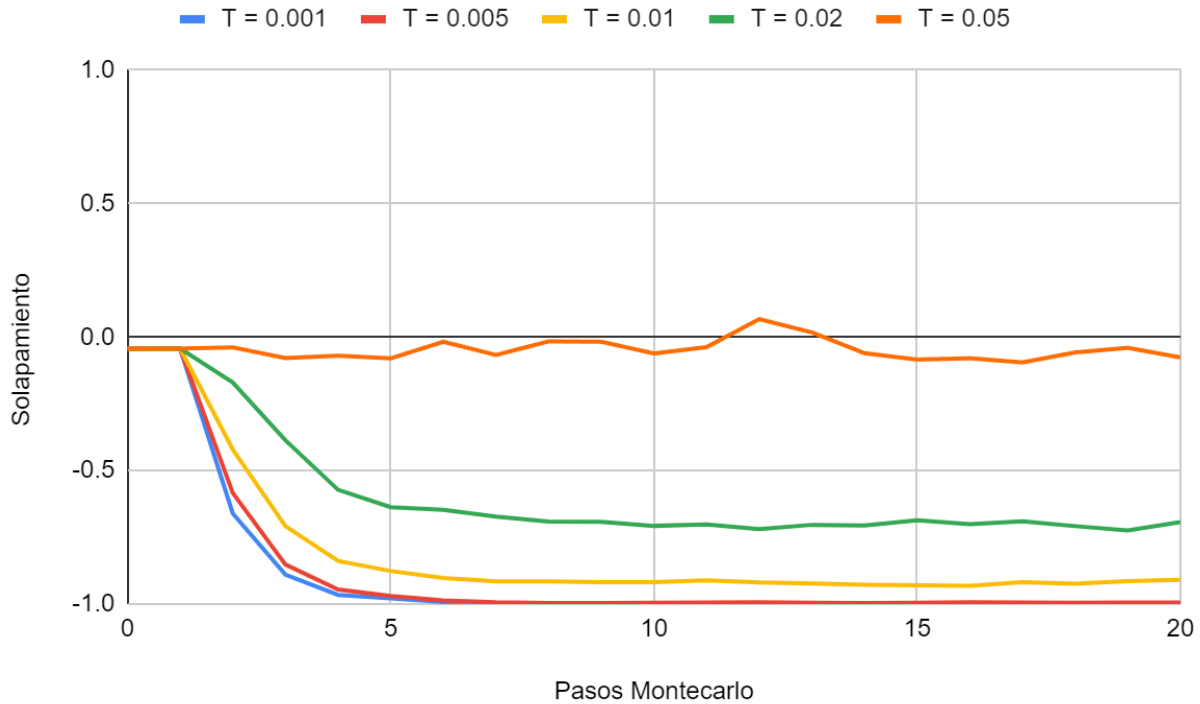


Figura 2: El solapamiento en función del tiempo para diferentes temperaturas

Notamos que no se representan las temperaturas más bajas con las que hemos generado resultados arriba. Eso es por su similitud a la temperatura  $T = 0.001$ , y todas se solapan de una forma parecida. Los valores posibles del solapamiento son del intervalo  $[-1, 1]$  siendo  $m(s) = 1$  cuando el patrón se recupera perfectamente,  $m(s) = -1$  cuando se recupera el antipatrón y  $m(s) = 0$  cuando el estado final de la red está desordenado en comparación con el patrón relevante.

Utilizando este conocimiento del solapamiento, en concordancia con las imágenes generadas, vemos que a las temperaturas bajas el antipatrón es completamente recuperado. Subiendo la temperatura, el algoritmo se queda a medias, oscilando alrededor de valores que representan una recuperación parcial del patrón (o antipatrón). Al final, a  $T = 0.05$ , al igual que antes, el estado final está desordenado, sin haberse ido hacia un valle en la superficie energética.

En la siguiente figura, hemos usado el programa *hopfieldtemp.c* para comparar el solapamiento en función de la temperatura, al cabo de 20 pasos Montecarlo.

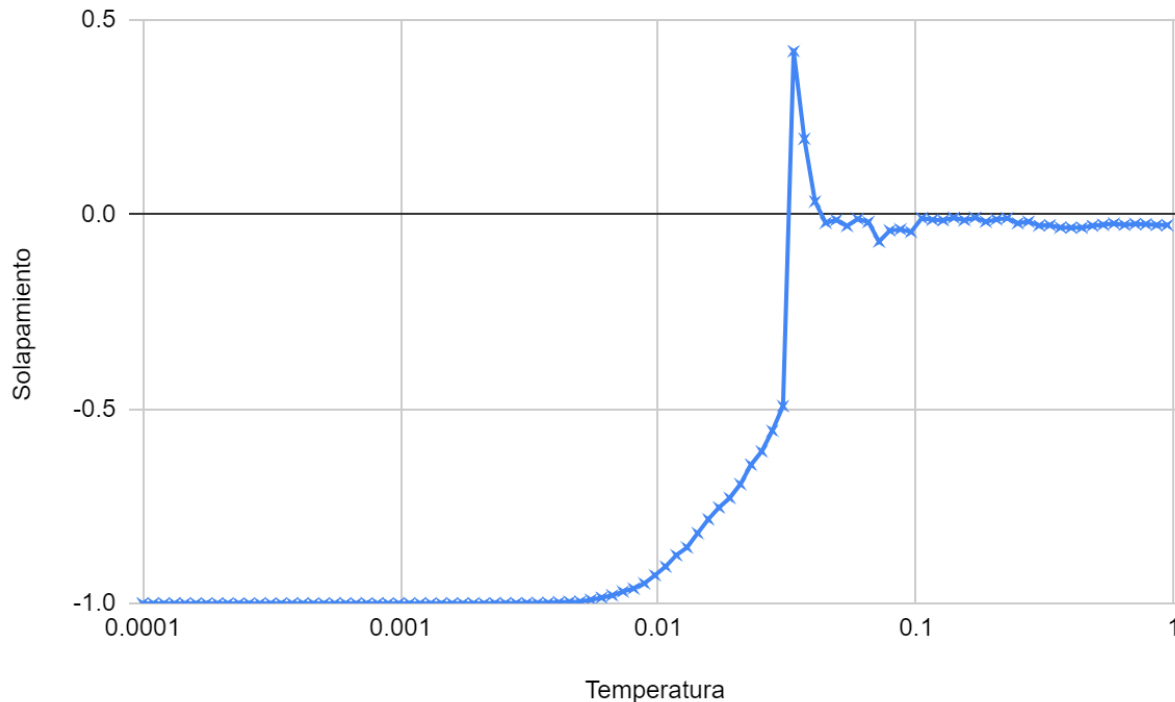


Figura 3: El solapamiento en función del tiempo al cabo de 20 pasos Montecarlo

En este gráfico, se ve que aumentando la temperatura hasta aproximadamente  $T = 0.009$  no despegamos el resultado de una recuperación perfecta del antipatrón. En la zona entre  $T = 0.009$  y  $T = 0.06$  el solapamiento va aumentándose hasta quedarse muy cerca de cero, una configuración desordenada. Es decir, que a partir de  $T = 0.06$ , la red ya no recupera el patrón. Notamos también el pico que vemos sobre  $T = 0.04$ . Aquí, la red llega parcialmente al patrón en cambio al antipatrón. Este puede deberse al carácter aleatorio del sistema. Mientras la temperatura se aumenta, aumenta la posibilidad de que haya cambios que no conduzcan hacia abajo en la superficie energética. Por eso, aunque empiece más cerca al antipatrón, un cambio aleatorio puede empujar el sistema hacia otro valle en la superficie energética y así llegamos al solapamiento positivo.

### 3.1.2 Estado inicial: Patrón deformado

En este apartado veremos qué ocurrirá si no partimos de un estado inicial aleatorio, sino del mismo patrón que introducimos, pero deformado. Aleatoriamente cambiaremos el estado del 30% de los nodos para que la configuración inicial de la red sea el patrón introducido deformado ~30%. Cabe destacar aquí

que no descartamos los nodos que se cambian una vez cambiado, es decir, que en el algoritmo que deforma el patrón, un nodo puede cambiar más de una vez, haciendo que la deformación total sea un poco menos que 30%. Sin embargo, para nuestro propósito, una investigación básica del modelo, nos vale.

Obtenemos los siguientes resultados usando el programa *hopfielddef.c*.

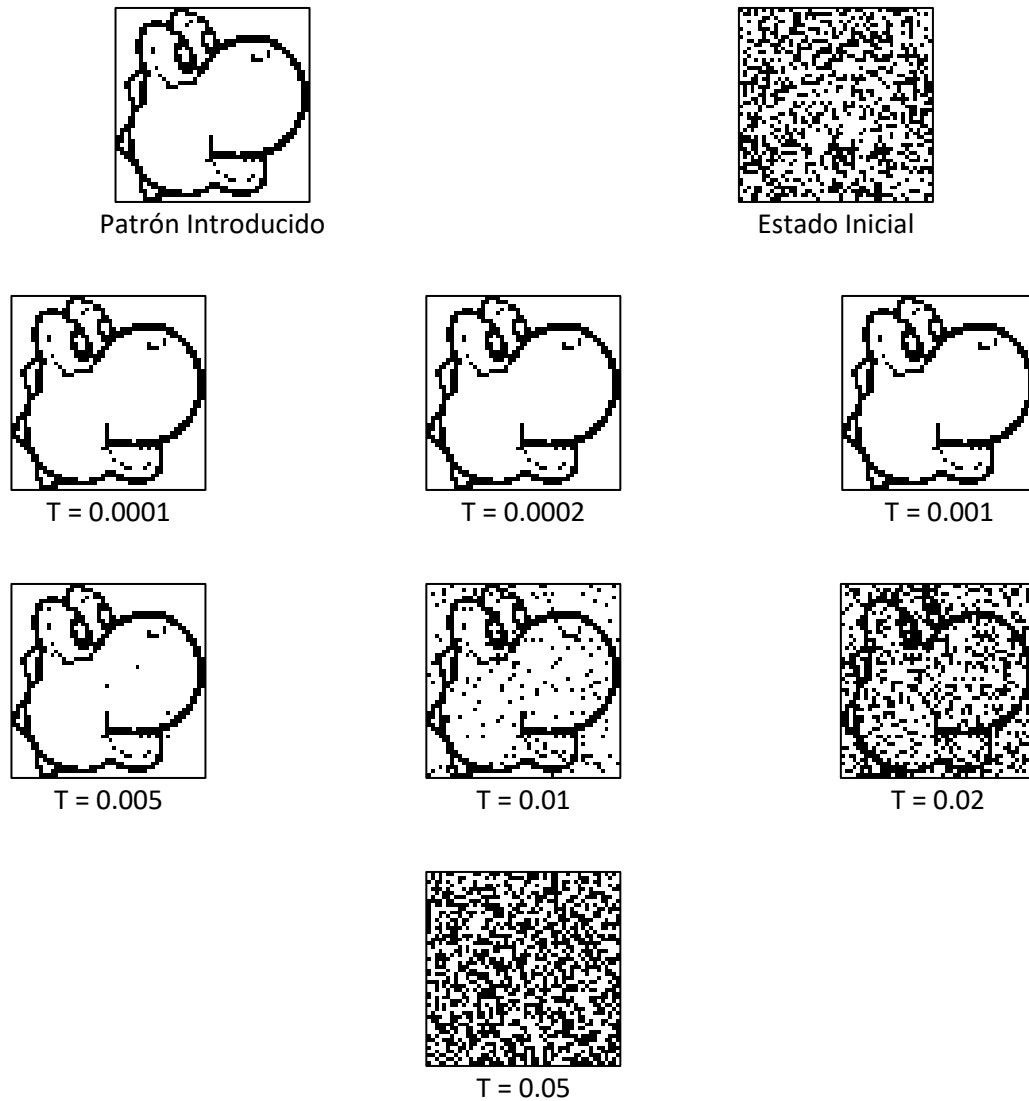


Figura 4: Estados resultantes partiendo del patrón deformado a varias temperaturas



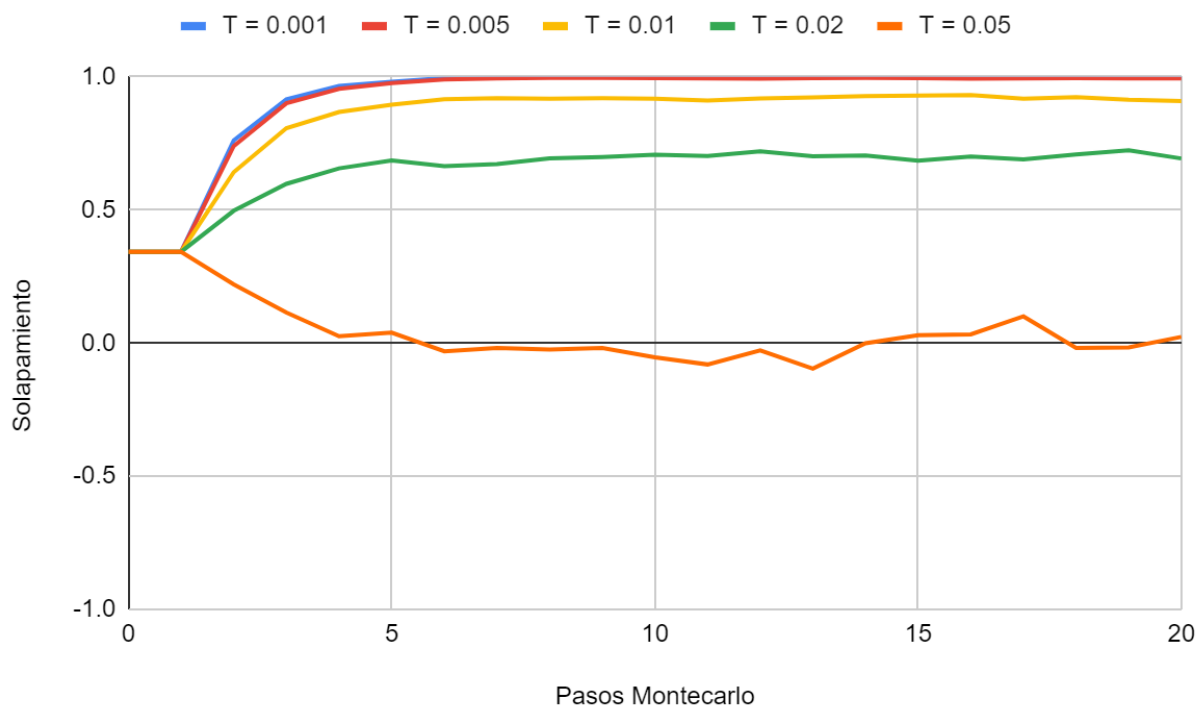


Figura 5: Solapamiento en función de pasos Montecarlo partiendo del patrón deformado

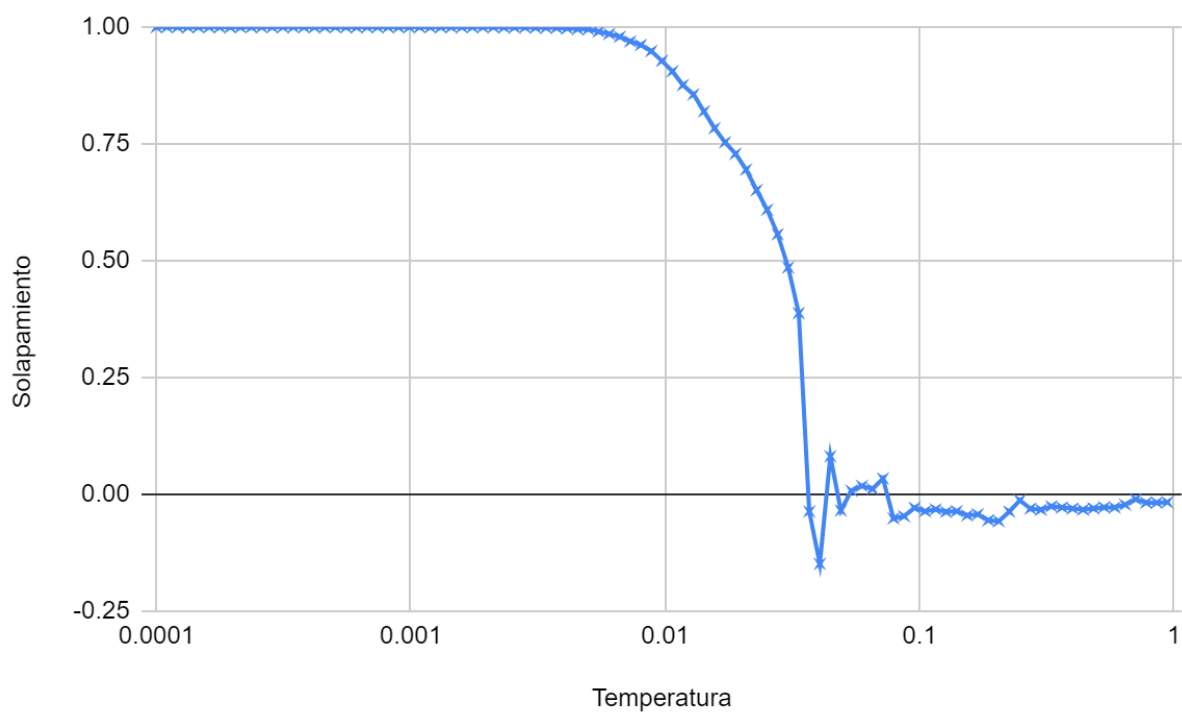


Figura 6: Solapamiento en función de la temperatura, partiendo del patrón deformado

Aquí vemos que el sistema partiendo del patrón deformado se comporta de manera muy similar al anterior, excepto que llega al patrón en cambio al estado espúreo. Sin embargo, lo importante es que aunque empecemos con una configuración cercana al patrón, veamos que a temperaturas altas la red todavía no lo recupera.

### 3.2 VARIOS PATRONES

Investigaremos en este apartado lo que ocurrirá cuando introducimos varios patrones a la red de Hopfield a la vez. Esto lo estudiaremos de la misma manera que hemos estudiado el caso del único patrón. No obstante, hemos de comentar que con varios patrones ( $P$ ), tendremos un número igual de solapamientos, lo que definimos al igual que hemos hecho en el primer apartado.

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

Vamos a usar los cuatro patrones que hay en la Figura 7, que notarán que todos tienen el fondo blanco y la letra están en partes diferentes del cuadro. Lo hemos hecho así para que los patrones sean muy distinguibles entre ellos. Podemos entender porque esto es necesario refiriéndonos a la analogía de la superficie energética. Los patrones representan depresiones en la superficie y la red se cae en una de ellas. Si los patrones son demasiado similares, las depresiones se superpondrán y no se podrá ver de manera visual en que consiste el estado final.

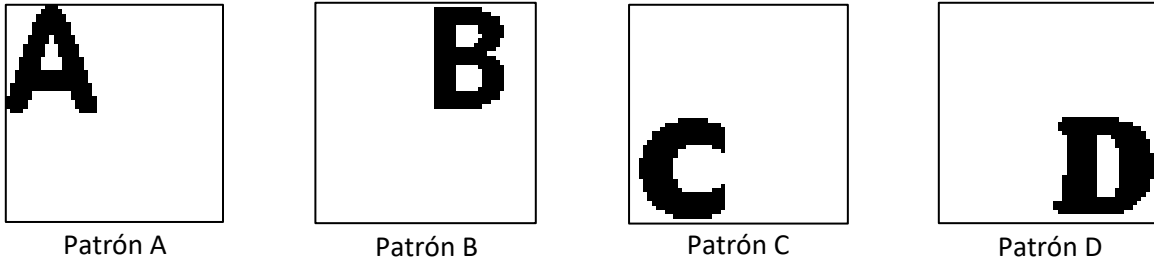


Figura 7: Los patrones introducidos en la red de Hopfield

#### 3.2.1 Estado Inicial Aleatorio

En la primera prueba, se ha usado una condición inicial aleatoria y se han realizado 20 pasos Montecarlo para diferentes temperaturas.

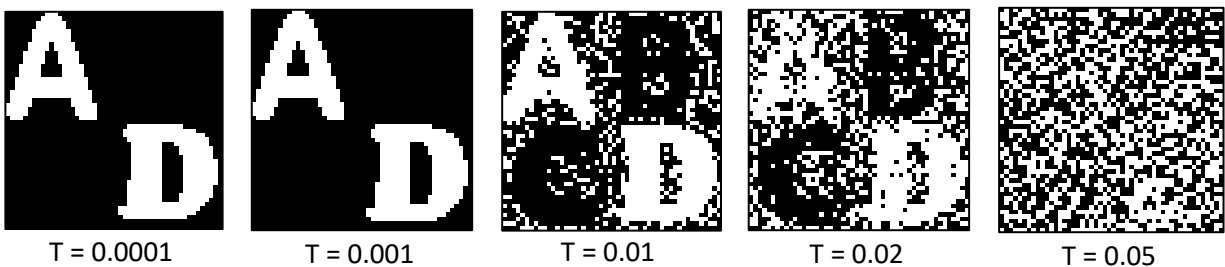


Figura 8: Resultados partiendo de un estado inicial aleatorio con varios patrones

Se observa que después de realizar 20 pasos Montecarlo la red acaba en una superposición de los antipatrones del Patrón A y Patrón D, y luego, con la subida de la temperatura, vemos que aparecen los

Patrones C y D también. A continuación, se presenta el solapamiento frente al número de pasos Montecarlo (a  $T = 0.0001$ ) y a la temperatura para cada uno de los patrones (tras 20 pasos Montecarlo).

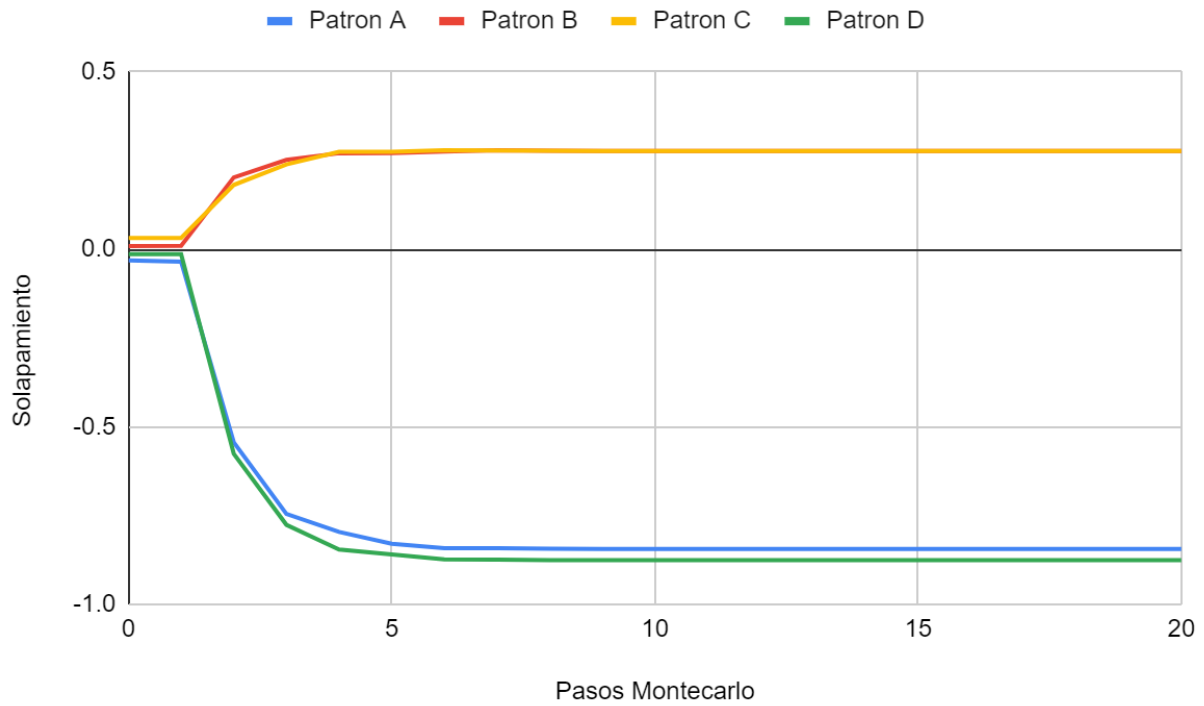


Figura 1: Solapamiento frente a pasos Montecarlo a partir de un estado inicial aleatorio y  $T = 0.0001$

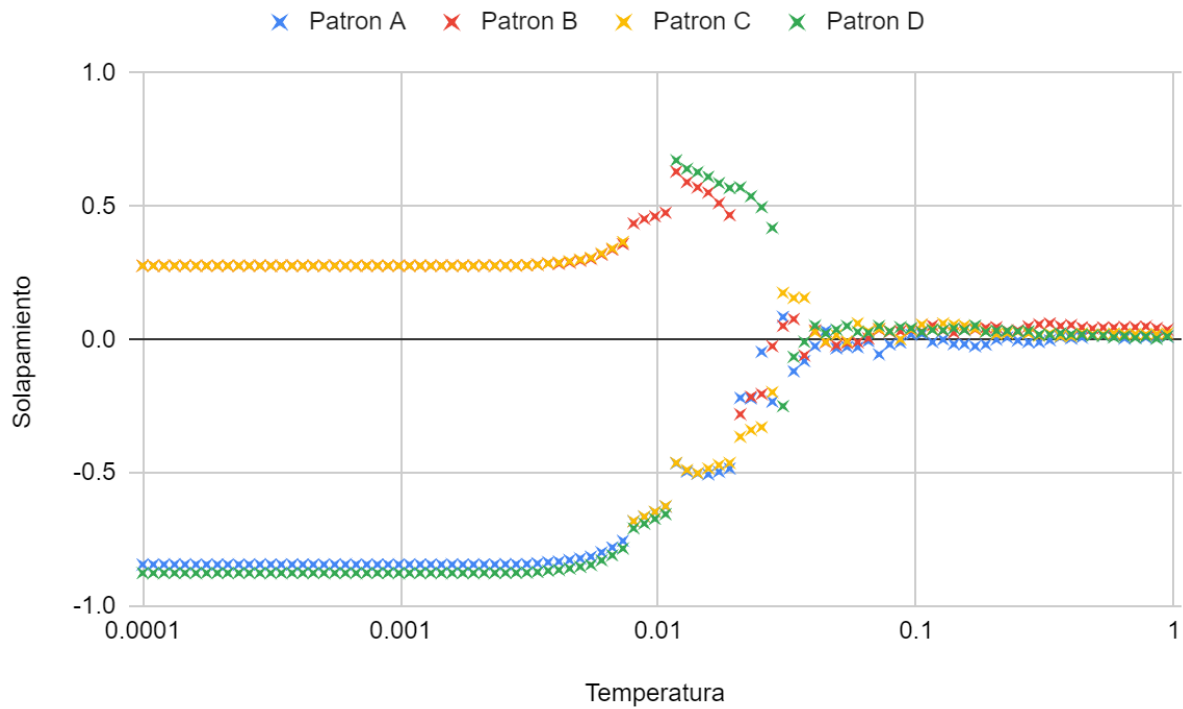
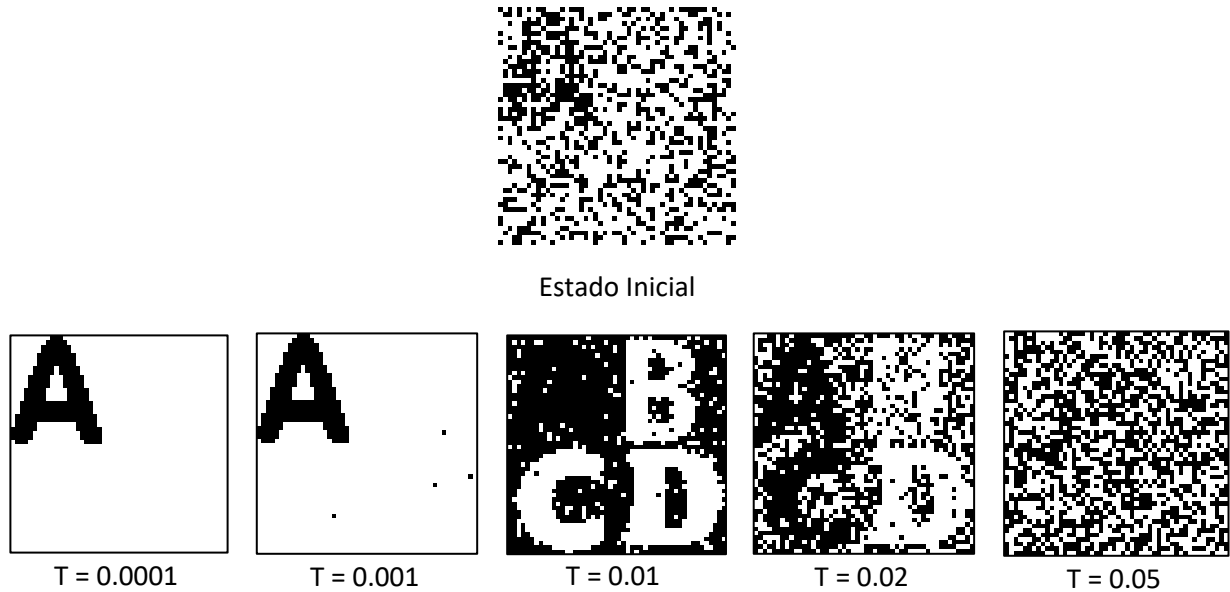


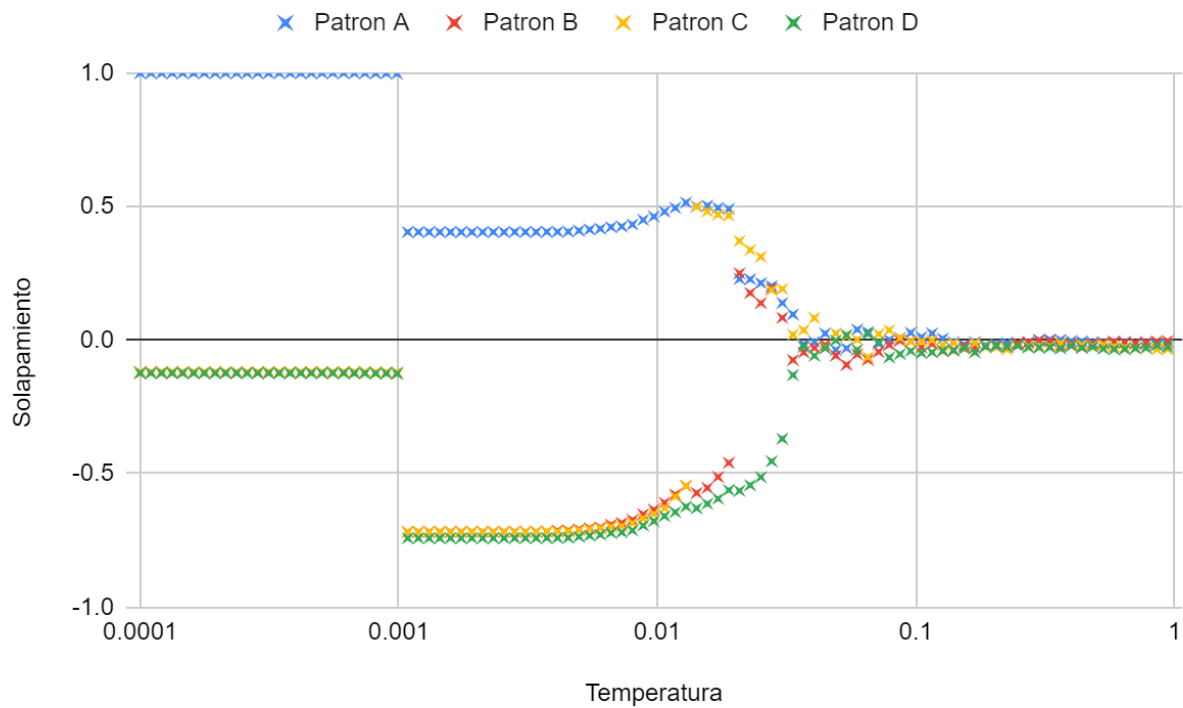
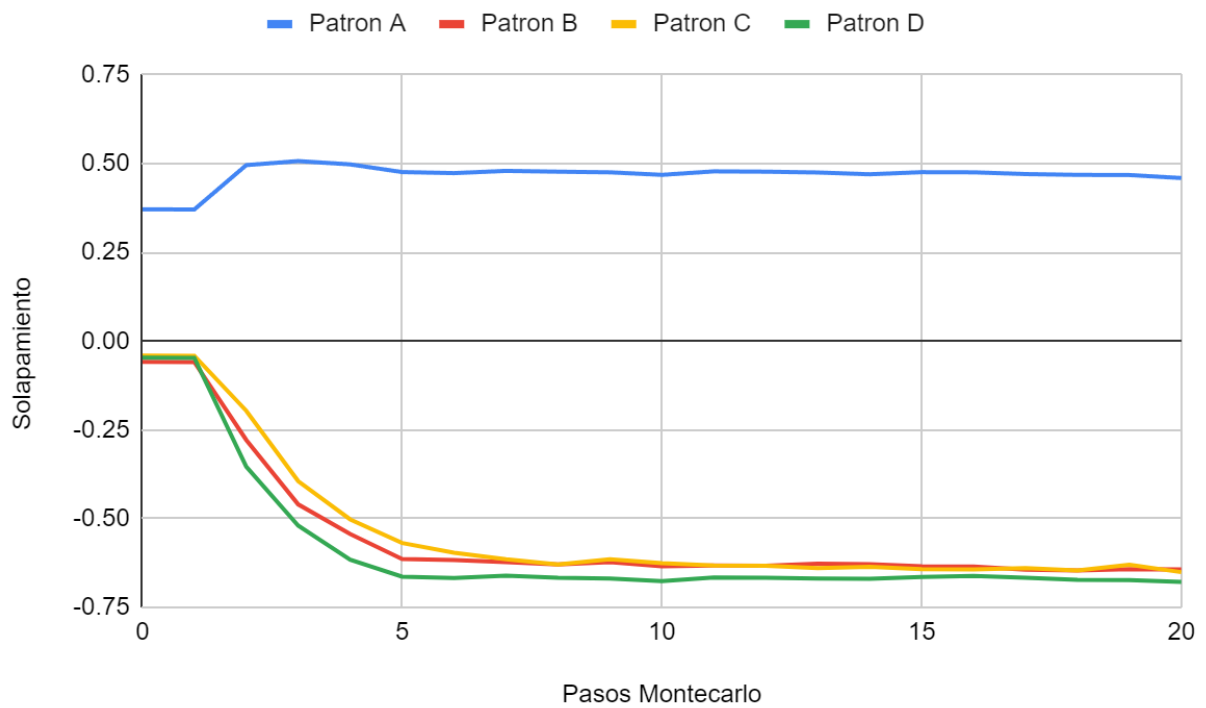
Figura 2: Solapamiento frente a la temperatura a partir de un estado inicial aleatorio, 20 pasos Montecarlo

En estas figuras podemos ver que la red converge a un estado que es la superposición de los patrones almacenados y que, como en el caso del único patrón, a partir de  $T = 0.009$ , entramos en una zona de transición donde los patrones no se recuperan perfectamente, pero siguen recuperándose. Por último, a temperaturas altas, no hay recuperación.

### 3.2.2 Estado inicial: Patrón deformado

Usando los programas *Hopmultdef.c* y *hopmultdeftemp.c* generamos lo mismo que antes, pero con un estado inicial del patrón A deformado por 30% (al igual que el caso de un patrón)





Los resultados son esperados. A bajas temperaturas, la red genera el patron A, pero mientras sube la temperatura, recuperan los otros patrones.

### 3.3 MEMORIA EN FUNCIÓN DEL NÚMERO DE PATRONES

En este apartado investigaremos la relación entre el número total de patrones almacenados y la memoria del sistema. Lo haremos a una temperatura baja ( $T = 0.0001$ ), con un red de  $20 \times 20$  y todos los patrones se generarán aleatoriamente.

Usamos el programa *hopfieldpatrons.c*. Este programa empieza con un patrón y calcula el solapamiento medio en los pasos Montecarlo 20 -40. De ahí, vaya repitiéndose, cada vez con un patrón más hasta 100 patrones. Denominamos un patrón recordado si el valor absoluto del solapamiento es mayor a 0.75, es decir, si  $|m^\mu(s)| > 0.75$  podemos decir es  $\mu$  es recordado por la red. Con el aumento del número de patrones, veremos cuantos la red es capaz de recordar. A continuación, vemos la figura (INSERT).

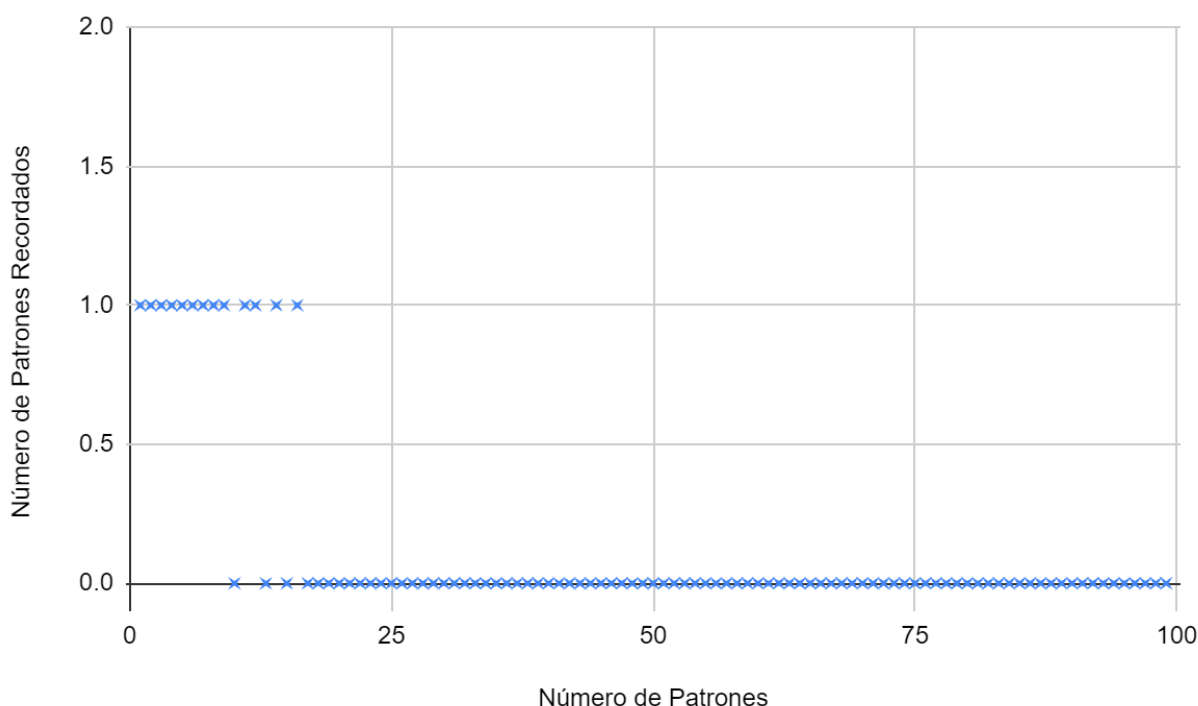


Figura 3: El número de patrones recordados en función al número de patrones aleatorios,  $T = 0.0001$

Vemos en la figura (INSERT) que la red solo es capaz de recordar un solo patrón y cuando llega a los 16 patrones, ya no es capaz de recordar ninguno. También veremos la fracción máxima que la red puede almacenar  $a_c = P_c / N^2$  de forma que todos los patrones se puedan recordar. Como el único caso en el que se recordó todos los patrones fue el caso de un solo patrón,  $a_c$  será 0.0025 o  $1/400$ .

Por último, notamos que en el apartado anterior la red era capaz de recordar varios patrones con el valor absoluto de sus solapamientos mayor a 0.75. Sin embargo, esos patrones no eran patrones aleatorios y compartían un fondo blanco, que hace que sus letras pudieran aparecer sin bajar demasiado el solapamiento de los otros patrones. Aquí, con patrones aleatorios, es muy poco probable que encontremos patrones tan similares para que eso suceda.

## 4 CONCLUSIONES

---

De estos resultados podemos sacar algunas conclusiones interesantes. En primer lugar, la red de Hopfield es capaz de recordar patrones almacenados a bajas temperaturas. En esto, vemos que puede recordar tanto el patrón en sí, como el antipatrón, o estado espúreo. Sin embargo, si la temperatura se aumenta, pierde esta habilidad y solo recupera ruido. Este efecto de las temperaturas altas ocurre igual cuando el estado inicial es muy similar al patrón.

La red también es capaz de recordar varios patrones, pero solo si son lo suficientemente similar, es decir, que un único mínimo en la energía puede considerarse “recordar” varios patrones. A temperaturas más altas, la red no llega a ningún patrón sino a un estado entre patrones, que recuerde ninguno, pero lleva elementos de varios.

Para más investigación, sería interesante seguir con estas mismas cuestiones, pero variando el tamaño de la red. En el apartado de los varios patrones, almacenamos cuatro patrones pro el tamaño de la red. Así cabían cuatro figuras para que su pudieran ver todas. Con una red más grande, la capacidad de recordar podría aumentar y sería capaz de recordar información más compleja.

## 5 REFERENCIAS

---

1. Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8), 2554–2558.
2. Hopfield, J. J. (1986). Neural networks and physical systems with emergent collective computational abilities. In *World Scientific Lecture Notes in Physics* (pp. 411–415). WORLD SCIENTIFIC.
3. Dennis, Simon (1997). The Hopfield Network: Descent on an Energy Surface.  
<https://staff.itee.uq.edu.au/janetw/cmc/chapters/Hopfield/>
4. Wikipedia contributors. (n.d.) Hopfield network. Retrieved July 8, 2023, from Wikipedia, The Free Encyclopedia website: [https://en.wikipedia.org/w/index.php?title=Hopfield network&oldid=1023261754](https://en.wikipedia.org/w/index.php?title=Hopfield_network&oldid=1023261754)