

Modelo de Hopfield de Red Neuronal

José Francisco Orta García

22/05/2021



**UNIVERSIDAD
DE GRANADA**

Grupo A - Subgrupo A1

Física Computacional

--

Grado en física

Universidad de Granada

Resumen

En esta práctica aplicaremos el algoritmo de Metrópolis para crear una red neuronal básica siguiendo el modelo de Hopfield. Estudiaremos su funcionamiento y comportamiento para diferentes temperaturas, tamaños y condiciones iniciales.

1. Introducción

El modelo de red neuronal de Hopfield, popularizado por John Hopfield en 1982, es un modelo de red neuronal artificial basado en el modelo de Ising, que posee una especie de "memoria asociativa" es capaz de recordar una serie de patrones. Posee un sistema de nodos que pueden tomar dos valores distintos, de manera que el sistema puede converger a un estado de "mínimo de potencial" que correspondería a uno de los patrones almacenados en la memoria de la red. [3]

1.1. Modelo de red

Como se mencionó anteriormente el modelo de red neuronal de Hopfield posee una serie de nodos binarios, que en este caso podrán tomar valores 1 y 0. Nombraremos a la red de N neuronas con la que trabajaremos s , de manera que s_{ij} hará referencia al nodo (i, j) de la red, donde $i, j = 0, 1, 2, \dots, N - 1$. El Hamiltoniano de la red que definirá la energía que posee una configuración concreta de la red es:

$$H(s) = -\frac{1}{2} \sum_{i=0}^{(N-1)} \sum_{j=0}^{(N-1)} \sum_{k=0}^{(N-1)} \sum_{l=0}^{(N-1)} \omega_{ij,kl} s_{ij} s_{kl} + \sum_{i=0}^{(N-1)} \sum_{j=0}^{(N-1)} \theta_{ij} s_{ij} \quad (1)$$

Donde vemos que existe interacción entre la neurona (i, j) y todas las demás. Estas interacciones vienen dadas por $\omega_{ij,kl}$, que es función de P , el número de patrones que almacenamos en la red. Sea $\xi^\mu = (\xi_{ij}^\mu)$ el patrón μ almacenado. Entonces

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

Y en caso de que $(i, j) = (k, l)$ entonces $\omega_{ij,kl} = 0$, ya que las neuronas no interaccionan consigo mismas. El parámetro a^μ viene dado por la expresión:

$$a^\mu = \frac{1}{N^2} \sum_{i=0}^{(N-1)} \sum_{j=0}^{(N-1)} \xi_{ij}^\mu \quad (3)$$

Podemos definir entonces ω como un tensor de cuatro dimensiones al que nos referiremos como tensor de pesos sinápticos.

El término θ_{ij} se denomina umbral de disparo y se define como:

$$\theta_{ij} = \frac{1}{2} \sum_{k=0}^{(N-1)} \sum_{l=0}^{(N-1)} \omega_{ij,kl} \quad (4)$$

Con esta definición de la energía del sistema podemos aplicar el algoritmo de Metrópolis tal y como se hace en el algoritmo de Ising. [1]

Llamamos solapamiento al parámetro que nos dice como de parecida es una configuración dada de la red s a uno de los patrones almacenados. Se define como:

$$m^\mu(s) = \frac{1}{N^2 a^\mu (1 - a^\mu)} \sum_{i=0}^{(N-1)} \sum_{j=0}^{(N-1)} (\xi_{ij}^\mu - a^\mu)(s_{ij} - a^\mu) \quad (5)$$

El valor de $m^\mu(s)$ irá desde 1 hasta -1 , siendo 1 el valor que tomará cuando la configuración de la red sea idéntica al patrón y -1 cuando sea inversa.

2. Resultados

Todo el material empleado en esta práctica, tanto códigos como animaciones se encuentra subido en el repositorio de GitHub indicado en la bibliografía. [2]

2.1. Solapamiento para un solo patrón

Para comenzar la práctica introduciremos un patrón de 1600 neuronas (40×40) y comprobaremos que nuestra red neuronal es capaz de memorizarlo. Partiremos de una temperatura de $T = 0,0001$ y dos tipos de condiciones iniciales, el patrón deformado y una matriz de espines aleatoria.

El patrón a memorizar se muestra en la **figura (1)**.

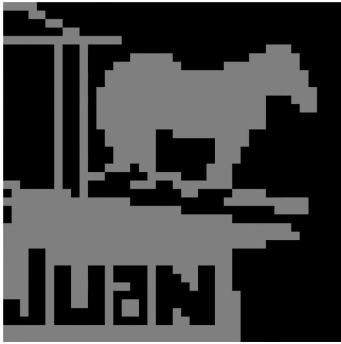


Figura 1: Patrón 1 de 40×40 neuronas que se introduce en la red.

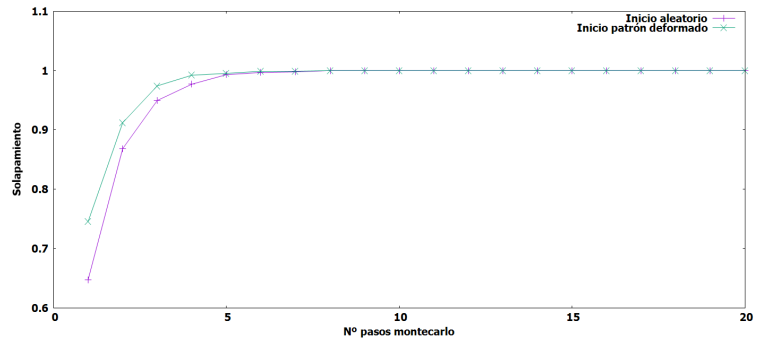


Figura 2: Solapamiento frente a n° de pasos Montecarlo realizados para $T = 0,0001$.

En la **figura (2)** podemos apreciar la representación gráfica del solapamiento de la configuración con el patrón dado por la **ecuación (5)** y el n° de pasos Montecarlo realizados. Observamos que aunque en el caso de tener una condición inicial aleatoria tenemos menor solapamiento al inicio, rápidamente estos valores se igualan, y en unos 5 pasos Montecarlo obtenemos un solapamiento de prácticamente 1 en ambos casos, recuperando así el patrón introducido.

Si variamos la temperatura podremos observar como el solapamiento cambia. Ejecutamos varias simulaciones de la red neuronal en las que variamos la temperatura, introduciendo los valores

$T = [0,001; 0,008; 0,02; 0,025; 0,05]$. Representamos los valores del solapamiento en el trigésimo paso Montecarlo, obteniendo:

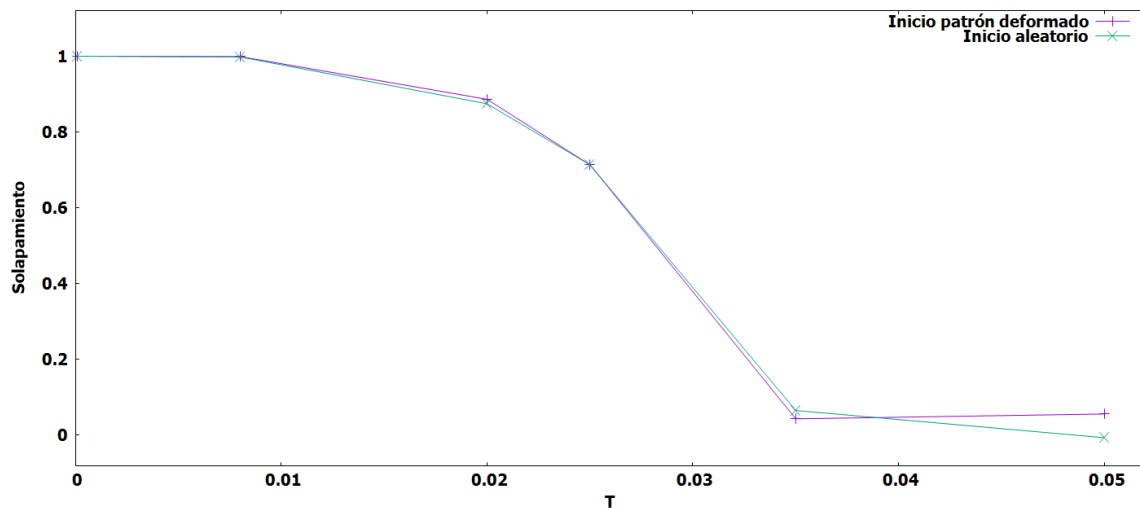


Figura 3: Solapamiento en el trigésimo paso Monte Carlo frente a temperatura

Apreciamos que el solapamiento disminuye inicialmente muy poco a poco con la temperatura, mientras que para temperaturas más altas, de unos 0.03, el solapamiento cae rápidamente hasta que llega un punto en el que oscila en torno a 0. Encontramos que a partir de $T = 0,025$ el patrón deja de ser reconocible a simple vista, encontrándose los valores de solapamiento por debajo de 0,75. Es destacable que no encontramos diferencias notables entre iniciar la simulación con un patrón aleatorio o con el patrón de la **figura (1)** deformado. Es probable que para redes con un mayor número de neuronas si que encontremos resultados distintos para cada caso.

2.2. Solapamiento para distintos patrones

A continuación realizaremos un análisis análogo al llevado a cabo, pero en este caso haremos que el programa memorice diversos patrones, 4 en total. El primero será el ya mostrado en la **figura (1)**, el resto serán:

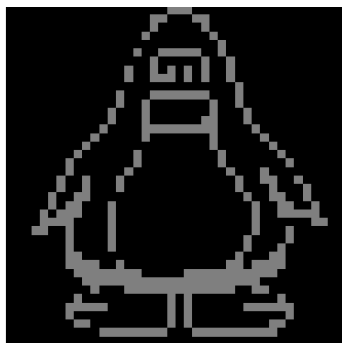


Figura 4: Patrón 2 introducido en el programa.

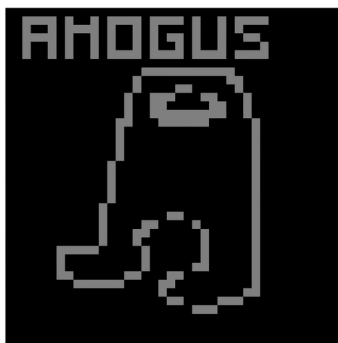


Figura 5: Patrón 3 introducido en el programa



Figura 6: Patrón 4 introducido en el programa

Veamos en primer lugar como cambia el patrón recuperado en función de las condiciones iniciales.

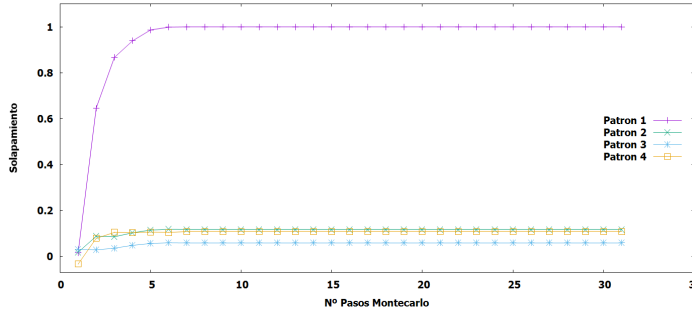


Figura 7: Solapamientos para condición inicial aleatoria a $T = 0,0001$

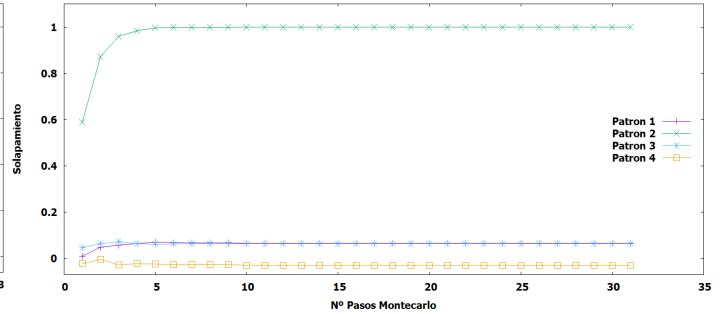


Figura 8: Solapamientos para condición inicial de patrón 2 deformado a $T = 0,0001$

En las **figuras (7) y (8)** se muestra la evolución del solapamiento frente a los pasos Montecarlo en los casos en los que la condición inicial es una red de espines aleatorios y en la que es el patrón 2 deformado. Apreciamos que, tal y como cabría esperar, en el caso en que la red inicial es un patrón deformado el sistema converge a dicho patrón. Únicamente mostramos el caso del patrón 2, puesto que los resultados son prácticamente idénticos para el resto de patrones.

El caso más interesante es el obtenido cuando la red inicial es aleatoria, pues en prácticamente todas las ejecuciones del código se converge al patrón 1. Esto puede deberse a que este patrón es el que presenta un mejor balance entre espines alineados en una dirección y en otra. Debido a que la condición inicial tenderá a tener la mitad de la red orientada en un sentido y la otra mitad en el otro sentido, es comprensible que el sistema tienda al caso en que el patrón es más equilibrado en este sentido.

Si ahora variamos la temperatura del sistema, dándole valores $T = [0,001; 0,008; 0,02; 0,025; 0,03; 0,05]$, podremos apreciar como el solapamiento varía. Tomaremos el solapamiento en el trigésimo paso Montecarlo para las diferentes condiciones iniciales. Los resultados obtenidos son:

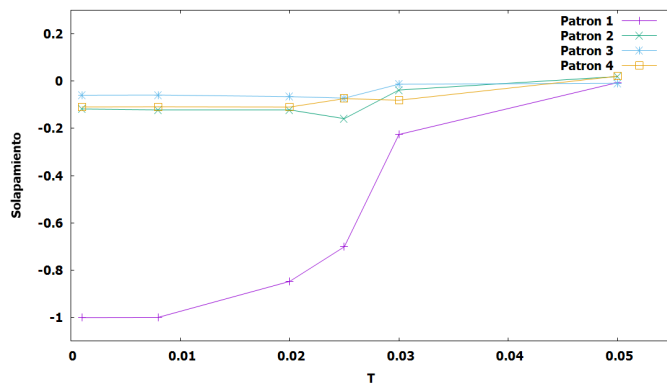


Figura 9: Solapamiento en función de la temperatura para matriz inicial aleatoria.

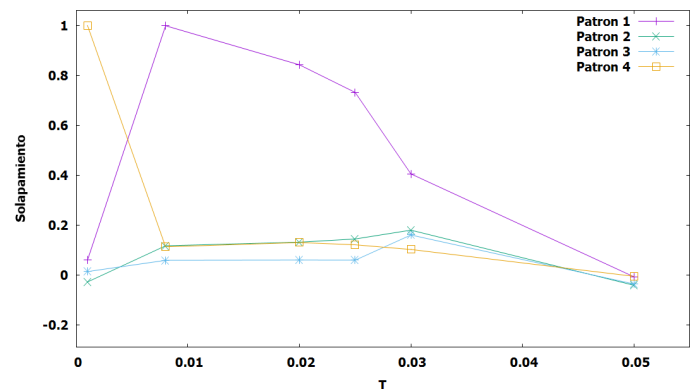


Figura 10: Solapamiento en función de la temperatura para condición inicial patrón 4 deformado.

Observamos en la **figura (9)** que en el caso en el que la condición inicial es aleatoria obtenemos el patrón 1 invertido para temperaturas bajas (esto cambia según la ejecución). Vemos que el resultado es muy similar al inverso del presentado en la **figura (2)**.

Cuando tratamos de establecer como condición inicial un patrón deformado distinto del patrón 1 obtenemos resultados similares a los expuestos en la **figura (10)**, donde para bajas temperaturas se converge al patrón esperado, pero al aumentar T el patrón 1 toma notoriedad y eclipsa al resto. Como sabemos, los patrones que se dan a la red neuronal queda fijados como estados de mínimo de potencial, a mayores temperaturas se exploran una mayor cantidad de mínimos, mientras que a temperaturas más bajas se tiene a caer en mínimos más locales. Esto nos lleva a pensar que el patrón 1 supone un mínimo de potencial mayor que el resto de patrones, por lo que se favorece la convergencia a este.

Dado que este último resultado no era el previsto, decidimos retirar el patrón 1 y operar con los patrones 2,3 y 4. Un ejemplo representativo de lo ocurrido es el caso en que comenzamos con el patrón 2 deformado.

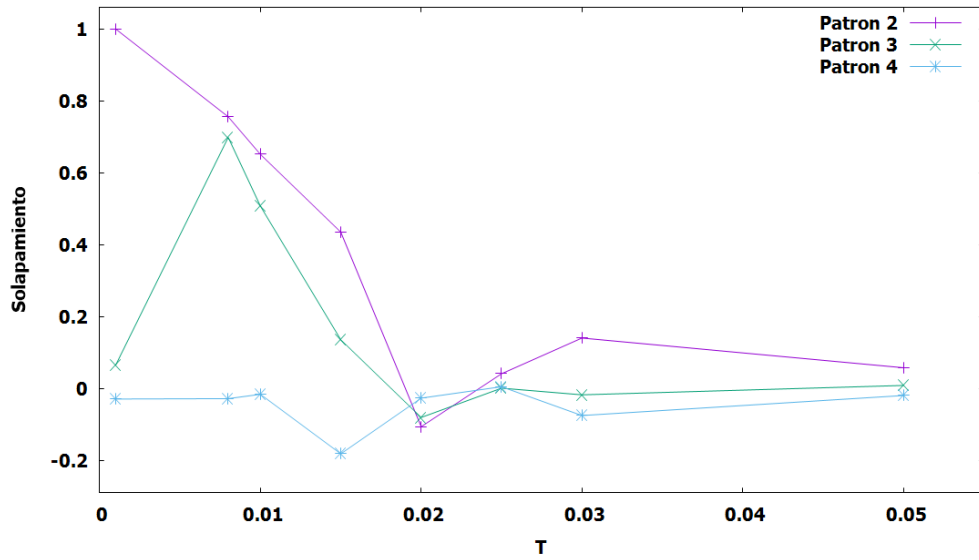


Figura 11: Solapamiento en función de la temperatura para condición inicial patrón 2 deformado.

En este caso $T = [0,001; 0,008; 0,01; 0,015; 0,02; 0,025; 0,03; 0,05]$, para operar con una mayor cantidad de puntos y mayor precisión. La **figura (11)** nos muestra que el solapamiento disminuye de forma más drástica en este caso que en el anterior, bajando m^H de 0,75 para valores de T próximos a 0,01. Además apreciamos que se tiene a un estado de mezcla cuando las temperaturas suben. En este caso entre los patrones 2 y 3, que son similares en su estructura. Es probable que en nuestro caso particular se tienda con facilidad al estado de mezcla para altas temperaturas porque el patrón 2 presenta bastantes similitudes con los patrones 3 y 4.

2.3. Decaimiento del solapamiento con el número de patrones almacenados

Almacenar una gran cantidad de patrones en la red neuronal debería provocar dificultades a la hora de recuperar un patrón dado. En este apartado trabajaremos con patrones de $N = 20$, es decir, de 400 neuronas y una temperatura $T = 0,0001$. Estos patrones serán generados aleatoriamente para tratar de evitar similitudes entre ellos que dificulten el procedimiento.

Variaremos el número de patrones almacenados y observaremos la capacidad de la red para recuperar cada uno de ellos. Para ello haremos uso de la llamada fracción máxima $c = \frac{P_c}{N^2}$, donde P_c es el número máximo de patrones almacenados para los que se recuperan todos los patrones. Esto significa que en $P_c + 1$ al menos uno de los patrones no se recuperaría con éxito.

Extraemos de la bibliografía un valor teórico de $\alpha_c = 0,138$, manteniéndose constante al variar N . [4]

En el caso descrito con anterioridad en el que $N = 20$ obtenemos $\alpha_c = 0,1375$, almacenándose un total de 55 patrones aleatorios, lo que nos da un error relativo $\varepsilon_r = 0,36 \%$, que nos indica que los resultados se ajustan perfectamente a los esperados.

Probando con $N = 40$ para observar si existen variaciones con N en nuestro caso obtenemos $\alpha_c = 0,125625$, que nos da un error relativo $\varepsilon_r = 8,97 \%$. Aunque se encuentra por encima del 5% , no se trata de un valor demasiado alto y se puede decir que nuestro modelo devuelve resultados lo suficientemente próximos a los teóricos.

3. Conclusión

En conclusión, en esta práctica pudimos comprobar el funcionamiento de una red neuronal de Hopfield básica con uno y varios patrones, aprendiendo en el proceso su funcionamiento con detalle, puesto que se elaboró un código propio para su funcionamiento.

En primer lugar comprobamos que con una media de 5 pasos Montecarlo era suficiente para que nuestro sistema convergiese a un patrón dado sin grandes diferencias entre el uso de unas condiciones iniciales u otras, comprobando además que el solapamiento del resultado obtenido tras 30 pasos Montecarlo disminuía con la temperatura de forma no lineal, reduciéndose poco a poco a temperaturas bajas y muy rápidamente una vez se alcanzaba cierta temperatura umbral o crítica, que en el caso del patrón uno estaba en torno a $T = 0,25$.

Seguimos un procedimiento completamente análogo en el caso en que se proveyeran 4 patrones distintos. Obtuvimos resultados distintos a los previstos, pues resultó que el patrón 1 generaba un mínimo de potencial más acusado que el resto, por lo que al subir la temperatura este patrón comenzaba a tomar notoriedad incluso cuando las condiciones iniciales llevaban a otro de los patrones. Además, se quiso observar el caso en que se retiraba el patrón problemático del sistema, obteniendo que para temperaturas no muy bajas se tendía al estado de mezcla en la mayoría de ocasiones, debiéndose esto probablemente a la similitud entre los patrones, que no serían completamente ortogonales.

Por último, trabajamos con la fracción máxima, comprobando que el solapamiento decae con la variación del número de patrones almacenado de manera que α_c oscila en torno al valor teórico $\alpha_c^{teo} = 0,138$. Con $N = 20$ obtuvimos un muy buen resultado, acercándonos mucho al valor teórico esperado. Con $N = 40$ este valor difería del esperado, pero no quedó demasiado lejos de este, por lo que resulta aceptable.

(Pequeño comentario recordando que tuvimos una tutoría para tratar de resolver el por qué se impone el patrón 1 en el apartado dos pero no hubo manera)

4. Apéndice

Para el cálculo eficiente de ΔH en el programa se hizo uso de la expresión obtenida de restar $H(s') - H(s)$ siguiendo la **ecuación (1)**, donde s es la red en un momento dado y s' esta misma red pero con la neurona (n, m) cambiada por $1 - s_{nm}$. Obtenemos:

$$\Delta H = \theta_{nm}(s'_{nm} - s_{nm}) - \frac{1}{2}s'_{nm} \sum_{k=0}^{(N-1)} \sum_{l=0}^{(N-1)} \omega_{nm,kl}s'_{kl} + \frac{1}{2}s_{nm} \sum_{k=0}^{(N-1)} \sum_{l=0}^{(N-1)} \omega_{nm,kl}s_{kl}$$

Lo que reduce drásticamente la cantidad de cálculos necesarios para obtener ΔH en cada iteración, y por ende el tiempo de ejecución del programa en su conjunto.´

Referencias

- [1] G. Galán-Marín y J. Muñoz-Pérez, *Design and Analysis of Maximum Hopfield Networks*. IEEE TRANSACTIONS ON NEURAL NETWORKS, Vol. 12, No. 2, Marzo 2001.
- [2] J.F.Orta-García, Material para la práctica de modelo de Hopfield.
https://github.com/Jorta18400/Repositorio_prueba/tree/main/Red%20Neuronal%20Hopfield
- [3] *Hopfield(RNA)*. Wikipedia, la enciclopedia libre, (15/02/2021).
https://es.wikipedia.org/wiki/Hopfield_%28RNA%29
- [4] J. Torres, *Tema 4. Redes Neuronales*. Marzo 2020.
http://www.ugr.es/~jtorres/Tema_4_redes_de_neuronas.pdf