

Memorias asociativas

Avetta, Gastón
Redes neuronales

10 de noviembre de 2023

1. Modelo de Hopfield sin ruido: regla de Hebb y dinámica determinista

Modelamos una red neuronal recurrente, es decir, donde todas las neuronas de la red están conectadas a todas las demás (salvo consigo misma, por convención). Dicha red está compuesta por N neuronas y el estado de las mismas (1 o -1) depende del estado de todas las demás en el instante de tiempo anterior de forma **determinista**. Existen dos maneras de llevar adelante esta dinámica: la forma **paralela**, donde el estado de cada neurona a tiempo $(t+1)$ es función del estado de todas las neuronas a tiempo t , como vemos en la expresión 1.1; o la forma **secuencial**, donde actualizo una neurona a la vez y, para las siguientes, tomo el nuevo valor de las neuronas ya actualizadas, tal como indica la expresión 1.2.

$$s_i(t+1) = \text{sgn}\left(\sum_{j=1}^N W_{ij} \cdot s_j(t)\right); \quad (1.1)$$

$$s_i(t+1) = \text{sgn}\left(\sum_{j=1}^i W_{ij} \cdot s_j(t+1) + \sum_{j=i+1}^N W_{ij} \cdot s_j(t)\right), \quad (1.2)$$

donde W_{ij} representa el peso de la conexión entre las neuronas i e j , $s_j(t)$ representa el estado de la neurona j a tiempo t y $\text{sgn}(\cdot)$ es la función signo.

Por otro lado, los pesos de las conexiones entre neuronas quedan determinados por los patrones a memorizar según la siguiente expresión, conocida como **regla de Hebb**,

$$w_{ij} = \begin{cases} \frac{1}{N} \sum_{\mu=1}^P x_i^{\mu} \cdot x_j^{\mu} & \text{si } i \neq j, \\ 0 & \text{si } i = j, \end{cases} \quad (1.3)$$

donde x_a^{μ} es el estado de la neurona a correspondiente al patrón μ de los P patrones que se desea memorizar en la red. Como dijimos en un comienzo, la conexión de una neurona consigo misma es nula.

Los patrones a memorizar por la red son definidos aleatoriamente, donde los estados 1 y -1 son equiprobables para todas las neuronas en todos los patrones.

1.1. Convergencia de las distintas dinámicas

Analizamos si la red converge o no a un **punto fijo** después de una cierta cantidad de pasos temporales, según utilicemos la dinámica paralela o la secuencial. Para esto, inicializamos la red con un patrón definido

aleatoriamente donde los estados de cada neurona son equiprobablemente 1 o -1 y dejamos evolucionar la red según alguna de las dinámicas propuestas, descritas por las expresiones 1.1 y 1.2.

Usando una red neuronal con 500 neuronas, definiendo 70 patrones para definir las conexiones W_{ij} ($\alpha = 0, 14$) y tomando cada uno de ellos como condición inicial de la red, obtenemos los siguientes resultados

- **Dinámica secuencial:** la red converge para todas las condiciones iniciales propuestas; en promedio tardó 3,4 pasos temporales para cada patrón y, en el peor caso, le tomó 25 pasos temporales converger.
- **Dinámica paralela:** de las 70 condiciones iniciales propuestas, la red convergió solo en 65 ocasiones. Cabe aclarar que, en los 5 casos restantes, lo que ocurre es que la red se queda atrapada en un ciclo de período 2: alterna infinitamente entre 2 estados, es decir, $s(t) = s(t + 2) = \dots = s(t + 2 \cdot n)$ y $s(t + 1) = s(t + 3) = \dots = s(t + 2 \cdot n + 1)$ para todo n natural.

Si bien estos son los resultados de un caso en particular y varían entre distintas realizaciones, además de depender de los parámetros N y P de la red neuronal, la conclusión es clara: la dinámica secuencial siempre converge a un punto fijo; mientras que la dinámica paralela, en algunas ocasiones, queda atrapada en ciclos de período 2, por lo cual la red no converge.

1.2. Overlap entre condición inicial y punto fijo

Como vimos recién, aplicando la dinámica secuencial a la red neuronal, esta siempre converge a un punto fijo. Sin embargo, este punto fijo no siempre coincide con la condición inicial impuesta a la red, dada por el patrón que se desea que la misma aprenda. Una forma de medir este desempeño es calculando el **overlap** entre el punto fijo obtenido y el patrón de aprendizaje deseado (condición inicial de la red), dado por la expresión 1.4,

$$m^\mu = \frac{1}{N} \sum_{i=1}^N s_i^\mu \cdot x_i^\mu. \quad (1.4)$$

Dado que este parámetro se calcula únicamente para un patrón, calculamos el overlap correspondiente a cada patrón de aprendizaje indicado y graficamos un histograma (normalizado) con los distintos valores obtenidos. En la figura 1.1 se pueden ver los histogramas obtenidos para distintas redes neuronales, donde en cada fila se muestran los resultados de redes con la misma cantidad de neuronas ($N = 500; 1000; 2000$ y 4000 , respectivamente) y, en cada columna, redes con la misma relación $\alpha = \frac{P}{N}$ ($\alpha = 0,12; 0,14; 0,16$ y $0,18$, respectivamente).

Cabe aclarar que, para las redes con $N = 4000$, se utilizaron solo la mitad de los patrones correspondientes para obtener los histogramas, debido al elevado tiempo de cómputo que requerían los mismos.

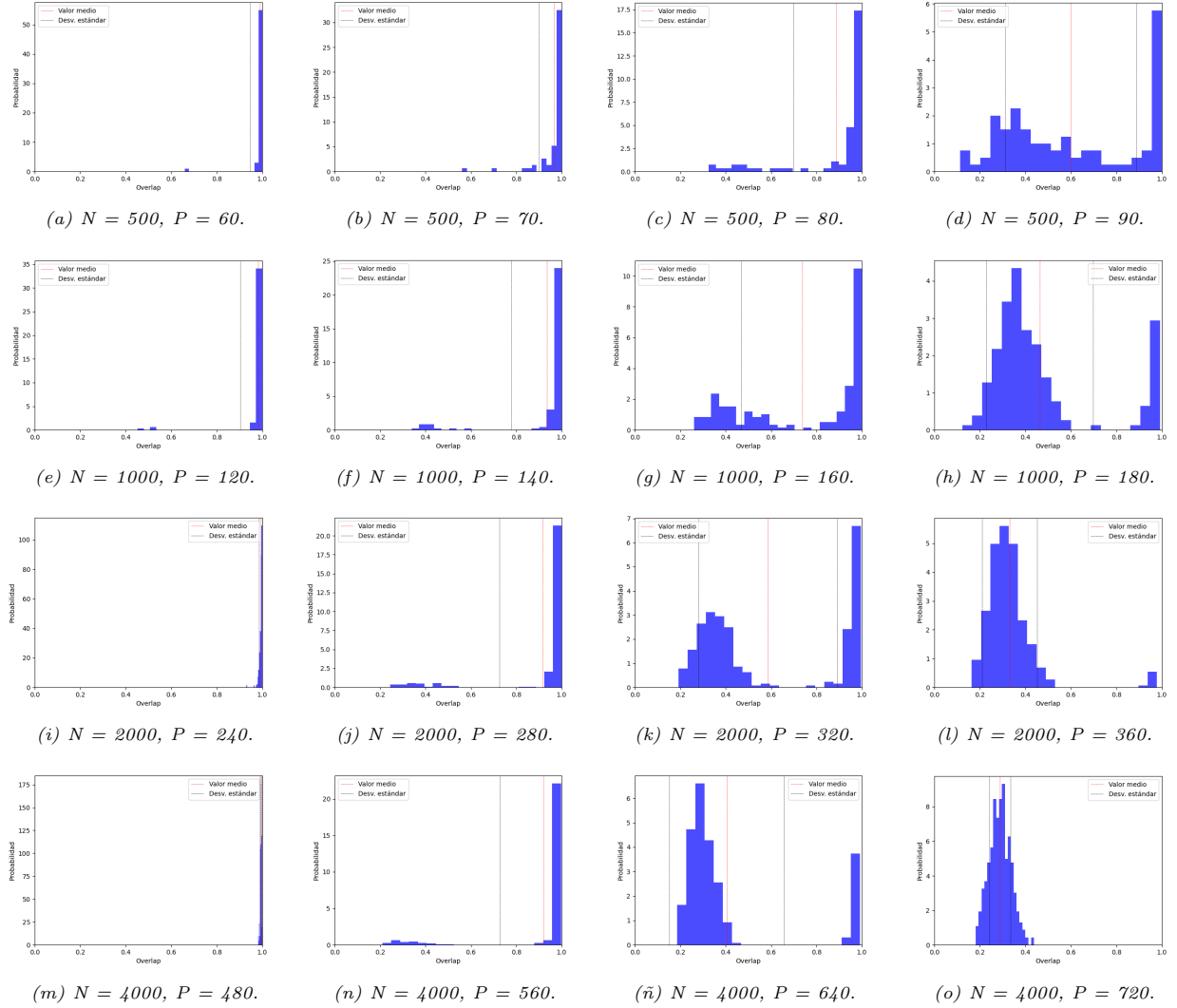


Figura 1.1: Histogramas normalizados de los valores de **overlap** obtenidos para redes neuronales recurrentes con distintas cantidades de neuronas (N) y distintas cantidades de patrones a memorizar (P). En líneas punteadas, el valor medio y su desviación estándar sumada/restada para cada distribución.

Podemos observar, si en cada fila analizamos de izquierda a derecha, la notable diferencia que hay entre los histogramas donde $\alpha \leq 0,14$ y donde $\alpha > 0,14$. Tal como vimos en clase, este valor es aproximadamente el límite entre tener un overlap no nulo o tener uno nulo para redes de tamaño (N) lo suficientemente grande. Vemos que, para $\alpha = 0,12$, todos los valores de overlap obtenidos son aproximadamente 1, siendo aún más notorio para redes de mayor tamaño (moviéndonos hacia abajo en la primera columna). Por otro lado, para $\alpha = 0,18$, ya se obtienen varios valores de overlap menores a 0,5; incluso, para la red de 4000 neuronas, parece que en ningún caso se obtiene un overlap mayor a dicho valor. En resumen, se puede concluir que, a medida que aumenta el parámetro α , el desempeño de la red empeora, teniendo un límite bastante marcado para $\alpha = 0,14$; por otro lado, a medida que aumenta la cantidad de neuronas, este límite se vuelve cada vez más claro: los overlaps tienden todos a 1 o difieren claramente de este valor según de qué lado del límite se encuentren, teniendo histogramas cada vez más concentrados a medida que crece la red.

En la figura 1.2 se puede ver resumida la información mostrada anteriormente: graficamos el valor medio de los overlaps obtenidos en función del parámetro α para los distintos tamaños de redes propuestos.

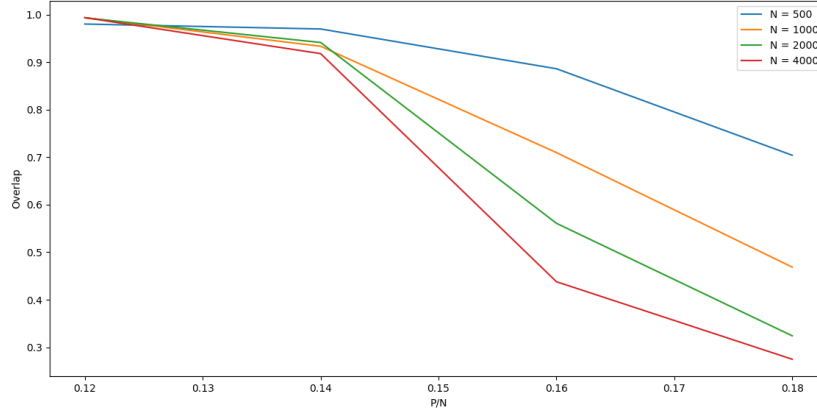


Figura 1.2: Valor medio de los overlaps obtenidos para cada una de las redes propuestas, graficados en función del parámetro α y separados según la cantidad de neuronas.

Aquí también podemos ver lo concluido anteriormente: a medida que aumenta α , los overlaps obtenidos son cada vez más pequeños, con una caída más notable a partir de 0,14 y más acentuada aún para redes de mayor tamaño.

2. Modelo de Hopfield con ruido: regla de Hebb y dinámica estocástica

Adoptando otra vez la dinámica secuencial, ahora, al modelo descrito en la sección anterior le aplicamos un pequeño cambio: si tenemos la expresión

$$h_i(t) = \sum_{j=1}^N W_{ij} \cdot s_j(t),$$

ahora el estado de la neurona i a tiempo $(t+1)$, $s_i(t+1)$, no depende únicamente del signo de dicha expresión, sino que queda determinado por una distribución de probabilidad que, lógicamente, depende, entre otras cosas, de $h_i(t)$, como podemos ver en la expresión 2.1,

$$P(s_i(t+1) = x) = \begin{cases} \frac{e^{\beta \cdot h_i(t)}}{e^{\beta \cdot h_i(t)} + e^{-\beta \cdot h_i(t)}} & \text{si } x = 1, \\ \frac{e^{-\beta \cdot h_i(t)}}{e^{\beta \cdot h_i(t)} + e^{-\beta \cdot h_i(t)}} & \text{si } x = -1, \end{cases} \quad (2.1)$$

donde $T = \frac{1}{\beta}$ se denomina la *temperatura* del sistema y da una noción de qué tanto difiere este modelo del modelo determinista. Podemos notar que, si $T = 0$, obtenemos el modelo determinista; mientras que, si $T \rightarrow \infty$, los estados 1 y -1 resultan equiprobables para cualquier neurona en cualquier instante de tiempo, sin importar el valor de $h_i(t)$.

Dado que ahora el sistema tiene un comportamiento estocástico, no converge a un punto fijo final, por lo cual hay que redefinir el **overlap** para tener una noción del parecido entre el patrón que se desea aprender y los estados que toma la red. La nueva definición de este parámetro está dada en la expresión 2.2,

$$m^\mu = \frac{1}{N} \sum_{i=1}^N \langle s_i^\mu \rangle \cdot x_i^\mu, \quad (2.2)$$

donde el operador $\langle \cdot \rangle$ representa la media sobre la dinámica (es decir, en tiempo) de la variable en cuestión (en este caso, el estado de la neurona i habiendo inicializado la red con el patrón μ).

Simulamos una red con 4000 neuronas y 40 patrones de aprendizaje ($\alpha = 0,01$) y calculamos el overlap de dichos patrones para distintos valores del parámetro T . En la figura 2.1 se pueden ver los resultados obtenidos para estas simulaciones: se grafican tanto el valor medio como la desviación estándar de los overlaps obtenidos para distintos valores de temperatura del sistema.

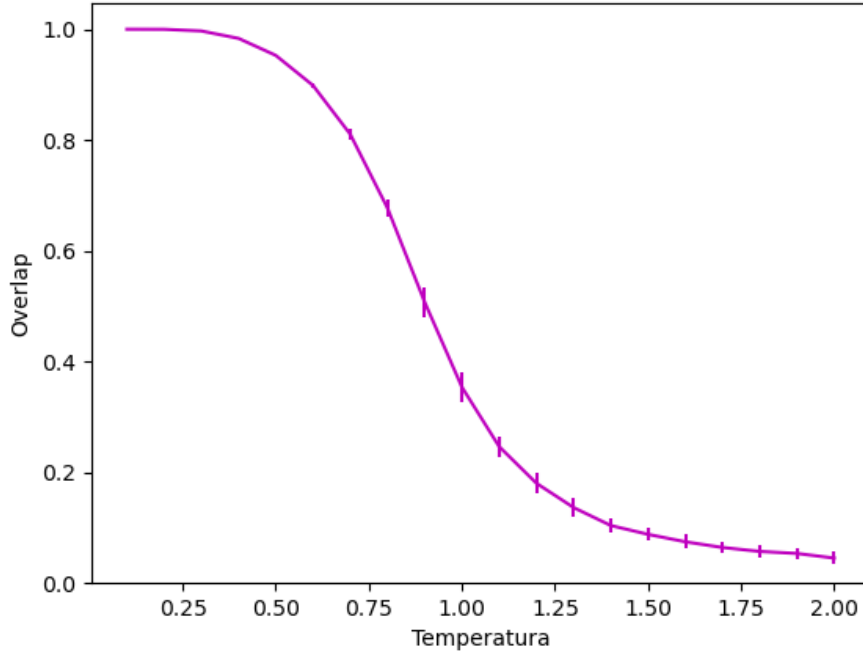


Figura 2.1: Valor medio y desviación estándar de los overlaps obtenidos en función de la temperatura del sistema, para una red de 4000 neuronas en la cual se desean memorizar 40 patrones.

Es fácil notar que la relación entre la cantidad de patrones a ser aprendidos por la red y la cantidad de neuronas que la conforman es un orden de magnitud menor a las consideradas en el ejercicio anterior. Teniendo en cuenta las conclusiones ya mencionadas, esto debería implicar un overlap muy cercano a 1. Esto resulta así para los valores más bajos de temperatura ($T < 0,25$), teniendo un overlap medio de prácticamente 1 y con una desviación estándar nula. Luego, a medida que aumenta la temperatura del sistema, empieza a predominar la lógica estocástica, por lo que, a pesar de tener un parámetro α muy pequeño, el desempeño de la red empeora considerablemente, decreciendo con una gran pendiente a partir de $T > 0,5$.

Luego, una vez que $T > 1$, el overlap medio parece saturar en un valor muy cercano a 0. Como fue mencionado en clase, esto es conocido como “efecto de tamaño finito”, debido a simular con redes con una cantidad finita de neuronas: si la red fuera de mayor tamaño, esta saturación se daría en 0. Lamentablemente, los tiempos de cómputo para $N > 4000$ se vuelven demasiado grandes, por lo cual no pude verificar este comentario.

Códigos utilizados

A continuación, se adjunta una carpeta con los códigos utilizados para resolver ambos ejercicios:

https://drive.google.com/drive/folders/1idY82U9stj80_VDdTMr8xp-GRJNd5HfA?usp=sharing.