



Mejoramiento de la capacidad de almacenamiento de una red neuronal autoasociativa a través de la conectividad selectiva

Facundo Emina

Tesis de Licenciatura en Ciencias Físicas

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

Julio de 2021

TEMA: Mejoramiento de la capacidad de almacenamiento de una red neuronal autoasociativa a través de la conectividad selectiva

ALUMNO: Facundo Emina

L.U. N°: 210/15

LUGAR DE TRABAJO: Laboratorio de Fisiología y Algoritmos del Cerebro
Fundación Instituto Leloir

DIRECTOR DEL TRABAJO: Emilio Kropff

CODIRECTOR: -

FECHA DE INICIACION: Abril de 2020

FECHA DE FINALIZACION: Julio de 2021

FECHA DE EXAMEN: 3 de Agosto de 2021

INFORME FINAL APROBADO POR:

Autor

Jurado

Director

Jurado

Profesor(a) de Tesis de Licenciatura

Jurado

Dedicado a mamá, papá, Vicky, Abu, Rayo y Rita

Índice

Agradecimientos	III
Resumen	v
1. Introducción	1
1.1. Motivaciones	1
1.2. El Modelo de Hopfield	2
1.3. El Algoritmo de Almacenamiento de Información	3
1.4. Recuperación de los Patrones	4
1.5. Introducción de Conectividad en la Red	5
1.5.1. Restricciones en la Conectividad	6
1.6. Capacidad de Almacenamiento	6
1.6.1. Análisis Señal-Ruido	7
1.6.2. Tensor de Ruido	8
1.6.3. Trabajos Previos	9
1.7. Cuenca de Atracción	9
1.8. Patrones de Baja Actividad Media	10
1.9. Algoritmos de Optimización Metaheurísticos	13
1.9.1. Enfriamiento Simulado	13
1.9.2. Algoritmo Genético	14
2. Métodos	16
2.1. Problema de Optimización: Minimización del Ruido	16
2.2. Algoritmos Implementados	18
2.2.1. Adaptación del Enfriamiento Simulado	18
2.2.2. Adaptación del Algoritmo Genético	19
2.3. Comparación entre Algoritmos	19
3. Optimización de la Red de Hopfield	22
3.1. Búsqueda Binaria	22
3.2. Capacidad de Almacenamiento	23
3.3. Estudio de la Relación entre los Pesos Sinápticos y la Conectividad	24
3.4. Análisis de la Señal y el Ruido	26
3.5. Cuenca de Atracción	29
3.6. Breve Comentario acerca de la Función de Costo	31
4. Modificación de la Función de Costo: Refuerzo de la Señal	32
4.1. Optimización con $\epsilon = \alpha$	32
4.2. Capacidad de Almacenamiento	32
4.3. Estudio de la Relación entre los Pesos Sinápticos y la Conectividad	34
4.4. Análisis de la Señal y el Ruido	37
4.5. Cuenca de Atracción	39

5. Introducción de Elementos Biológicamente Plausibles	40
5.1. Múltiples Conexiones entre Pares de Neuronas	40
5.2. Reducción de la Actividad Media de la Red	42
5.3. Resultados de la Optimización	43
6. Algoritmo Online	44
6.1. Descripción del Algoritmo	44
6.2. Función de Costo	47
6.3. Variación de la Conectividad Media	48
6.4. Capacidad de Almacenamiento	49
7. Conclusiones	52
8. Apéndice A: Optimización en el Modelo Clásico de Hopfield	55
8.1. Valor Óptimo del Parámetro ϵ	55
9. Apéndice B: Algoritmo Online	58
9.1. Funciones de Costo Alternativas	58
9.2. Valor Óptimo del Parámetro ϵ	59
9.3. Capacidad de Almacenamiento	60
Bibliografía	62

Agradecimientos

‘Y si pudiere entretrejer tristeza con tristeza, hacer la red que pesque ratos de alegría en la memoria.’

- Hernán Izurieta

No podría dar comienzo a esta tesis sin antes dar las gracias a todas las personas que colaboraron con ella de alguna u otra manera, ya sea ayudándome con cuestiones relacionadas con el trabajo como acompañándome con un mate en el río en cualquier momento durante la carrera. Este trabajo se dio en el período más turbulento de mi vida y haber llegado al punto de escribir estos agradecimientos habría sido imposible sin el apoyo y compañía de tantas personas.

Primero de todo, quiero agradecerle a mi director, Emilio, por haberme guiado durante este tiempo, siempre con paciencia, comprensión y calidez, haciendo que la virtualidad y el contexto de trabajo nunca dejase de ser ameno. Gracias por la confianza que me tuviste, por dejarme explorar con libertad muchos de los caminos que se fueron abriendo durante el trabajo y por abrirme la puerta a un mundo que no conocía y me encantó (y que por suerte queda una infinidad por explorar). La motivación nunca faltó, a pesar de que el contexto externo no ayudara. Sin duda aprendí muchísimo y disfruté del recorrido.

No me alcanzan las palabras para nombrar todo lo que tengo que agradecerle a mi familia y cualquier texto se va a quedar corto. Todo paso que di en esta vida fue gracias a ustedes, los amo y los admiro. Mamá, me pudiste sostener incluso dando la pelea más grande de tu vida. Ojalá pueda tener algún día una fracción de tu fuerza. Papá, cuando parecía que el mundo se iba a caer, fuiste el pilar que lo sostuvo todo. Gracias a los dos por siempre empujarme a seguir adelante, por haber despertado mi curiosidad y entusiasmo por la ciencia, música y deporte y haberme impulsado a explorarlo todo con pasión. Pero más que nada, gracias por constantemente creer en mí, contenerme y apoyarme. Vicky, de vos aprendo todos los días, y si pude crecer es porque te tuve a mi lado. Que nunca nos falten las risas y bromas básicas como siuviésemos 5 años. Gracias por dejar que te explique de temas random, como si en realidad supiera de algo, pero especialmente, por escucharme, contenerme y compartirlo todo. Abu, gracias por apoyarme en todo lo que hice y hago, por buscar entenderme cuando no me entendés, por las risas en los juegos de cartas y por los viajes que me abrieron la cabeza. Admiro enormemente tu fuerza y entusiasmo por la vida.

También quiero agradecer al resto de mi familia, por estar siempre presentes y por habernos acompañado en este último tiempo tan duro. Particularmente, darle las gracias a Inés, por abrirnos sus brazos cuando más lo necesitamos y hospedarnos en su casa en el medio de la tormenta. Fue ahí donde di los primeros pasos en este trabajo. También darle las gracias a Beba y Magui por haber estado tan presentes durante nuestra estadía interminable en el hospital, trayendo risas en momentos tan difíciles. A Fabi, Erica y Matute, su amor, contención y presencia nunca nos faltó. Y por lo mismo le agradezco a las innumerables personas que nos acompañaron incondicional-

mente, relevándonos a Vicky y a mí de responsabilidades para que pudiéramos seguir estudiando.

Agradezco a mis amigos, que son la familia que elijo todos los días. A Serra, Tobi, Manu, Agus, Casti, Santi, Juanso, Juanma y Cande (todo Gordobardo). Podría escribir otra tesis entera contando todo lo que me dan, así que me quedo corto con decir lo importantes que son para mí. Los viajes, las comilonas, juntadas de todo tipo y recitales de La Experiencia fueron parte de este camino. Jamás me faltó su aliento ni compañía.

A Manu y Mati, con quienes nunca dejamos de compartir momentos ni divertirnos con las mismas pavadas que hacíamos a los 12 años. Particularmente, agradecerle a Mati por haber sido mi Matlab Dr. cuando necesité ayuda, por motivarme a investigar en el área de la neurociencia y por las infinitas charlas en temas variados de ciencia.

A Nico, por las meditaciones, recetas, entrenamientos y charlas sobre la vida que tan bien me hacen. Y también por los inolvidables asados junto con Nixon y Lucho. A pesar de la distancia, los siento igual de cerca.

Quiero agradecer a Alejo y Tomi, por haberme bancado con mis preguntas acerca de algoritmos y complejidad computacional. Hoy puedo decir que entiendo un poquito más.

Gracias a mis amigos y compañeros de la facu. Siempre me sentí cómodo en Exactas y me hubiese sido imposible seguir la carrera sin un grupo humano tan lindo. Gracias Ale, por haberme cuidado, motivado, animado y acompañado tanto; tu cariño fue infinito y esencial.

Gracias a todo el grupo de Desayunos Terapéuticos, por la calidez, contención y cuidado que recibí este último tiempo. Aprendí (y aprendo) muchísimo con y de ustedes.

A todas las personas que mencioné y me faltó mencionar, ojalá les haya podido aportar al menos una parte de todo lo que me dieron a mí.

Cualquier viaje, después de todo, es sólo tan bueno como la compañía.

Resumen

En esta tesis se estudió el límite de la capacidad de almacenamiento de redes neuronales autoasociativas con modificaciones sinápticas basadas en una regla de aprendizaje Hebbiano. A diferencia de los modelos clásicos, donde se utilizan conectividades aleatorias, el aspecto novedoso de este trabajo es que se modificó la matriz de conexiones de manera tal de optimizar la *performance* de la red. Para ello, en base a un análisis de la señal y el ruido presente en el campo de actividad que recibe cada neurona, se propuso una función de costo a minimizar mediante un algoritmo de enfriamiento simulado. La minimización de dicha función implicó una reconfiguración de las conexiones presinápticas de cada neurona, mejorando globalmente la capacidad de almacenamiento de la red. Asimismo, se diseñó un algoritmo biológicamente plausible que dio cuenta de un proceso simultáneo de generación y eliminación de conexiones capaz de mejorar la capacidad de la red. Uno de los resultados más interesantes fue que, al optimizar una red mediante este algoritmo, la misma tendió a estabilizarse en una conectividad fija. Para conectividades iniciales menores a este valor de equilibrio, la dinámica de su conectividad media presentó una tendencia similar a la observada en muchas regiones del cerebro de algunos mamíferos (como el ser humano), teniendo una primera etapa donde predomina la generación de conexiones, hasta alcanzar un pico y pasar a una segunda etapa donde predomina la eliminación.

1. Introducción

1.1. Motivaciones

La biología provee constantemente de evidencia que resalta la importancia de las características topológicas de las redes neuronales en los cerebros animales. Particularmente, dos factores fuertemente determinantes de la dinámica de estas redes son la distribución de pesos sinápticos y la estructura del cableado de la red. Un claro ejemplo está en la funcionalidad del hipocampo, que es capaz de almacenar y recuperar patrones de actividad neuronal. Se puede modelar su forma de cómputo siguiendo una dinámica de atractores [1], como se describe en el modelo simplificado de Hopfield [2]. En dicho modelo, los estados atractores surgen como consecuencia de configurar los pesos sinápticos siguiendo una regla de aprendizaje Hebbiano [3], donde la sinapsis entre dos neuronas se refuerza si ambas se activan juntas.

Si bien existen muchos trabajos en modelos teóricos que describen la funcionalidad del hipocampo y otras áreas del cerebro mediante una dinámica de atractores [1, 2, 4, 5], éstos utilizan una conectividad estructural definida aleatoriamente. Sin embargo, diversas pistas provenientes de observaciones experimentales dan cuenta de ciertas limitaciones en este tipo de modelos. Para empezar, en una red neuronal autoasociativa, cálculos teóricos y simulaciones indican que la capacidad de almacenamiento de la red (es decir, cuántos patrones de actividad es capaz de almacenar la red y recuperar de forma efectiva) es, a primer orden, proporcional a la conectividad con un factor que va desde ~ 0.1 para redes neuronales muy artificiales a ~ 0.01 , o menos, para redes que incorporan elementos biológicamente plausibles [1, 4, 5, 6]. Si se tiene en cuenta que el cerebro humano tiene 10^{11} neuronas, pero en promedio solamente 10^4 conexiones por neurona, las redes autoasociativas implementadas en el mismo podrían almacenar entre 100 y 1000 memorias, un límite que llama la atención por lo reducido. Otros elementos, como la actividad limitada, pueden elevar este límite un orden de magnitud, pero sigue siendo llamativamente bajo.

Por otro lado, es posible ver mediante el estudio de imágenes, cómo la formación de conexiones neuronales en el sistema nervioso central es un proceso altamente dinámico, que consta de eventos simultáneos de eliminación y formación de conexiones. Evidencia funcional sugiere que una regla común en gran parte del sistema nervioso central consiste en que células post-sinápticas reciben inicialmente un gran número de entradas, y la poda de las mismas mejora la respuesta neuronal durante la maduración [7]. Esto se refleja en la variación temporal de la densidad sináptica a medida que madura el cerebro. Esta evolución presenta un primer estadio donde abunda la formación de conexiones, alcanzando un pico a partir del cual comienza a predominar un proceso de poda de conexiones, para luego estabilizarse en la edad adulta [8, 9]. La diferencia porcentual entre la máxima densidad sináptica alcanzada en las primeras etapas del desarrollo y la densidad sináptica en la adultez varía según la región del cerebro y especie animal. Por ejemplo, en la corteza frontal humana se ha reportado un pico cerca de los dos años de edad un 50% mayor, mientras que en el caso de ratas se ha llegado a reportar un pico a los 35 días de vida (es decir, a medio camino de la adultez) sólo un 10% mayor [8]. Asimismo, existen trabajos que sugieren que el proceso de recableado constante observado en la corteza del adulto podría estar vinculado con el aprendizaje y almacenamiento de información [7, 10].

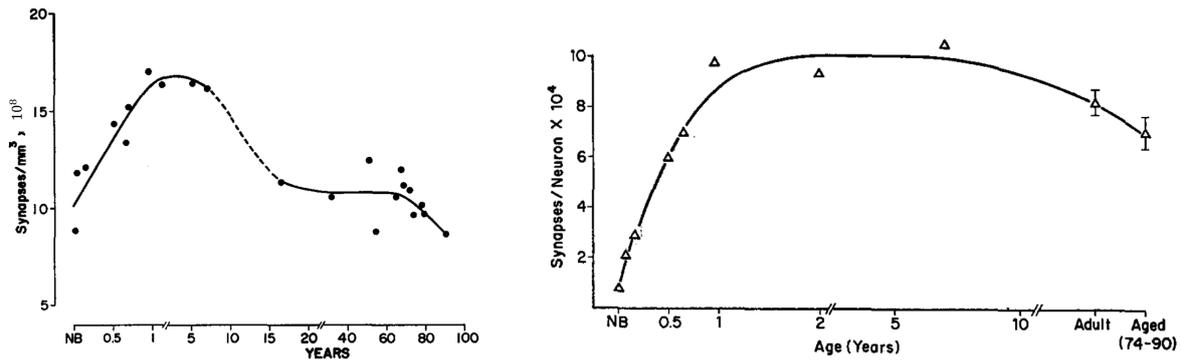


Figura 1: Ambas figuras muestran resultados observacionales correspondientes al giro frontal medio humano. En la izquierda, se muestra la densidad sináptica (sinapsis por $mm^3 \times 10^8$) en función de la edad. A la derecha, el número de conexiones por neurona (sinapsis por neurona $\times 10^4$) en función de la edad. Figuras tomadas del trabajo de Huttenlocher *et al.* [8].

Otras preguntas relacionadas con la arquitectura de conexiones surgen al comparar los cerebros de distintas especies animales. A lo largo de la evolución, el cerebro de los mamíferos creció varios órdenes de magnitud en cuanto al número de neuronas, pero permaneció estable en cuanto a la densidad de conexiones neuronales [11, 12]. En el cerebro humano, la materia blanca (conexiones de largo alcance) representa aproximadamente la mitad de la masa. Incrementar el número de conexiones implicaría aumentar enormemente el volumen cerebral (o bien, sacrificar neuronas), por lo tanto, es curioso entender qué mecanismos le permitieron al cerebro no incrementar su volumen de conexiones.

Todo esto desemboca en una serie de interrogantes muy interesantes que se abordarán en las siguientes páginas. Entre ellas, la principal pregunta es si es posible mejorar la capacidad de almacenamiento de una red neuronal autoasociativa mediante una elección especial de la estructura de sus conexiones. Si es posible mejorarla, ¿hasta cuánto? ¿Cuál debe ser la regla de cableado óptima? ¿Existirá algún mecanismo biológicamente plausible que de cuenta de esta mejora? El presente trabajo apunta a responder estas preguntas.

1.2. El Modelo de Hopfield

En analogía con sistemas físicos donde fenómenos colectivos emergen como resultado de la interacción entre sus componentes elementales (como orientaciones magnéticas estables o patrones de vórtices en un flujo de fluido), J. J. Hopfield estudió cómo el almacenamiento y la recuperación de memorias puede pensarse como una propiedad colectiva emergente de un conjunto grande de neuronas [2]. Básicamente, propuso una arquitectura de red neuronal como memoria autoasociativa, capaz de almacenar información en forma de patrones de actividad, y de recuperarlos en el futuro si el estado inicial de la red representa a una de estas memorias, pero corrompida.

El modelo de Hopfield discreto consiste en N neuronas interconectadas entre sí, sin autoconecciones, por lo que el número de conexiones por neurona es de $c = N - 1$. La actividad de la red evoluciona de a pasos temporales discretos Δt , y cada neurona puede encontrarse en sólo uno de

dos estados posibles a un dado tiempo. Estos estados son $s_i = 1$ ó $s_i = -1$, que representan el potencial de acción de la neurona i -ésima medido en un intervalo de tiempo Δt , indicando si la neurona está activa o inactiva, respectivamente.

El modelo se basa en una red recurrente donde la actividad de una neurona queda determinada por la actividad del resto en un tiempo anterior. Cada neurona i recibe un campo de actividad h_i dado por

$$h_i(t) = \sum_{\substack{j=1 \\ j \neq i}}^N J_{ij} s_j(t) \quad (1)$$

donde se introduce J_{ij} como la matriz de adyacencia de la red. Cada uno de sus elementos indica el peso sináptico de la conexión entre la neurona presináptica j y postsináptica i (en caso de ser nula, indica la ausencia de dicha conexión). Como ya se mencionó, en el modelo de Hopfield no existen autoconexiones, lo que significa que $J_{ii} = 0 \forall i = 1, \dots, N$. La actividad resultante de la neurona dependerá del valor del campo,

$$s_i(t + \Delta t) = \text{sgn}(h_i(t)) \quad (2)$$

donde U es un umbral de actividad (mientras más alto, más difícil es excitar a la red, y viceversa) y $\text{sgn}(x)$ devuelve el signo del argumento x .

1.3. El Algoritmo de Almacenamiento de Información

La tarea de la red consiste en almacenar p patrones distintos etiquetados con la variable μ tal que $1 \leq \mu \leq p$. Cada patrón se define como una configuración dada por $\{\xi_i^\mu = \pm 1; 1 \leq i \leq N\}$. En la versión más simple del modelo (que es con la que se trabajó, a menos que se indique lo contrario), los patrones no están correlacionados entre sí y cada uno está compuesto por $N/2$ neuronas activadas y $N/2$ desactivadas, de forma tal que la actividad media de la red es siempre nula ($\sum_{j=1}^N \xi_j^\mu = 0$). Para estos patrones, el umbral de actividad óptimo presente en la regla de recurrencia de la ecuación 2 es $U = 0$. Por lo tanto, siempre se tomó este umbral a lo largo del trabajo a menos que se indique lo contrario.

El algoritmo de almacenamiento consiste en una regla de aprendizaje Hebbiano en el que el peso sináptico entre dos neuronas se modifica de acuerdo a su coactivación en los distintos patrones. Particularmente, en este modelo los pesos W_{ij} son iguales a la diferencia entre el número de patrones donde ambas neuronas poseen igual actividad y el número de patrones donde cada neurona se encuentra en un estado distinto [1, 2]

$$W_{ij} = \sum_{\mu=1}^p \xi_i^\mu \xi_j^\mu \quad (3)$$

Los elementos de matriz W_{ij} pueden modelarse como el resultado de una caminata aleatoria de p pasos (salvo por los elementos diagonales, donde $W_{ii} = p$). Esto se debe a que todo par de variables ξ_i^μ y ξ_j^ν son independientes entre sí. Los elementos de matriz de W pertenecen al conjunto $\{2k - p \mid k \in \mathbb{Z}; 0 \leq k \leq p\}$, y la probabilidad de que $W_{ij} = 2k - p$ coincide con la de una distribución binomial de p eventos con probabilidad $1/2$ de éxito o fracaso

$$\mathcal{P}(W_{ij} = 2k - p) = \mathcal{B}(x = k | p, 1/2) = \binom{p}{k} \frac{1}{2^p} = \frac{p!}{k!(p-k)!} \frac{1}{2^p} \quad (4)$$

A partir de esta matriz de pesos, en una red de conectividad completa la matriz de adyacencia se define como

$$J_{ij} = \frac{1}{N} W_{ij} (1 - \delta_{ij}) \quad (5)$$

con δ_{ij} la función delta de Kronecker.

1.4. Recuperación de los Patrones

Se dice que la red recupera óptimamente el patrón ν si sucede que éste es un punto fijo de la dinámica, es decir $s_i(t + \Delta t) = s_i(t) = \xi_i^\nu$. En una simulación donde una red de Hopfield recupera un patrón, se define un estado inicial de la red $\bar{s}(t_0)$ de forma tal que éste contenga información parcial (o corrompida) del patrón ν , es decir $\bar{s}(t_0) - \bar{\xi}^\nu = \bar{\Delta}^\nu (\neq \bar{0})$. Siguiendo la ecuación 2 que define la dinámica del sistema, se deja evolucionar a la red hasta alcanzar, idealmente, el punto fijo deseado. Para poder seguir la evolución temporal, conviene definir un parámetro que mida la similaridad entre el estado de la red y el patrón que se desea recuperar. Se define entonces el solapamiento (*overlap*) de la red con el patrón ν

$$m^\nu(t) = \frac{1}{N} \sum_{j=1}^N \xi_j^\nu s_j(t) \quad (6)$$

El solapamiento de la red con el patrón ν está acotado entre -1 y 1, donde $m^\nu = 1$ indica que la red ha recuperado de forma óptima el patrón (es decir, $\bar{s} = \bar{\xi}^\nu$). Por otro lado, si $m^\nu = 0$ entonces no hay correlación entre el estado de la red y el patrón a recuperar. Por último, si $m^\nu = -1$

indica que $\bar{s} = -\bar{\xi}^\nu$; a estos estados se los denomina espurios ya que, al almacenar un patrón $\bar{\xi}^\nu$, se almacena también $-\bar{\xi}^\nu$ como consecuencia de la simetría de reflexión de la actividad de la red (en la bibliografía se explica más en detalle este fenómeno [1, 13]). Si bien el caso donde $m^\nu = 1$ representa la recuperación óptima, puede suceder que el solapamiento se estanque en algún valor menor a 1. Más aún, puede suceder que se estanque en un ciclo límite, donde la red transita un número finito de estados periódicamente. En la práctica, es necesario definir la recuperación de una memoria a través de un límite en el solapamiento. En estos casos, si el *overlap* es siempre mayor a algún umbral predeterminado, se puede decir que la red ha recuperado de forma parcial (o no óptima) el patrón. La convención utilizada en este trabajo es que una red es capaz de recuperar un patrón (por más que sea de forma parcial) si la misma converge a un atractor o ciclo límite tal que $m^\nu > 0.7$. En la figura 2 se muestra la evolución temporal de la red mediante el solapamiento de varios patrones, pudiéndose notar cómo recupera a algunos. Es importante aclarar que cada curva se corresponde a un evento de recuperación distinto (no recupera los patrones de forma simultánea). Por otro lado, se muestra también un ejemplo de ciclo límite en el que puede estancarse la red.

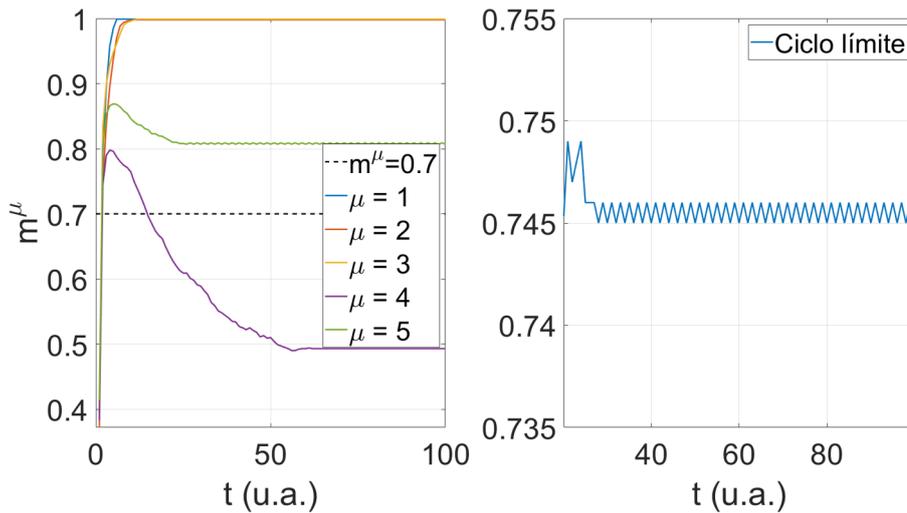


Figura 2: Izquierda: se grafica el *overlap* (m^μ) en función del tiempo (unidades arbitrarias) correspondiente a distintos patrones almacenados en una red de 2000 neuronas. En los 5 casos, la red se inicializó en un estado con $m^\mu(t=0) \sim 0.4$. Se observa que todas las curvas convergen a un estado estacionario después de un tiempo. Sin embargo, el patrón $\mu = 4$ no es recuperado ya que, según el criterio establecido, $m^4 < 0.7$. El resto de los patrones sí son recuperados. Derecha: se muestra un ciclo límite en el que se estanca la misma red para otro patrón. En este caso, el patrón es recuperado ya que su solapamiento es siempre mayor a 0.7.

1.5. Introducción de Conectividad en la Red

En el modelo simplificado de Hopfield, la conectividad es completa, es decir que cada neurona se conecta con las $N - 1$ restantes. Sin embargo, esto no es biológicamente plausible ya que la conectividad en los cerebros de los mamíferos es baja (particularmente, en el ser humano $c/N \sim 10^{-7} \ll 1$ [10]). La forma de definir una conectividad no completa en la red es mediante una variable aleatoria c_{ij} que indique la existencia o no de una conexión dirigida desde la neurona presináptica j a la postsináptica i . Para simplificar el tratamiento analítico, se suele utilizar la

siguiente distribución, asumiendo que las variables c_{ij} son independientes entre sí y donde c es el valor medio de la conectividad por neurona

$$P(c_{ij}) = \begin{cases} \frac{c}{N}\delta(c_{ij} - 1) + (1 - \frac{c}{N})\delta(c_{ij}) & \text{si } i \neq j \\ \delta(c_{ij}) & \text{si } i = j \end{cases} \quad (7)$$

La matriz de adyacencia de la red, con información de pesos sinápticos y existencia o no de conexión entre par de neuronas, se expresa de la siguiente forma

$$J_{ij} = \frac{1}{c}c_{ij} \sum_{\mu=1}^p \xi_i^\mu \xi_j^\mu = \frac{1}{c}c_{ij}W_{ij} \quad (8)$$

donde se normaliza usando el valor de la conectividad media por convención (más adelante se dará sentido a esta normalización).

1.5.1. Restricciones en la Conectividad

Si se considera que la matriz de conectividad representa la conexión física entre dos neuronas, entonces una modificación biológicamente plausible es relajar la restricción $c_{ij} \in \{0, 1\}$, permitiéndole tomar valores discretos mayores a 1. Existen estudios que muestran la presencia de conexiones multi-sinápticas, siendo la distribución del número de conexiones entre par de neuronas bimodal, con un primer pico en el cero (ya que la conectividad en el cerebro es baja) y otro pico entre las 3 y 8 conexiones [14, 15]. Durante el trabajo, se utilizó la restricción $c_{ij} \in \{0, 1\}$, salvo en la subsección 5.1, donde se investigaron los efectos que trae relajar esta condición.

1.6. Capacidad de Almacenamiento

El número de memorias que puede almacenar una red y recuperar exitosamente no es infinito. En redes de conectividad aleatoria, el límite es aproximadamente proporcional al número de conexiones por neurona. El parámetro relevante que describe la capacidad de almacenamiento es la razón entre el número de bits almacenados y el número de sinapsis en la red $\alpha = (pN)/(cN) = p/c$. Existe un valor crítico $\alpha_c = p_c/c$ a partir del cual la red sufre una transición hacia una fase amnésica, donde pierde la capacidad de recuperar la información almacenada [1, 4]. Se puede demostrar que α_c es una función monótonamente decreciente con c/N [1] y particularmente, el caso correspondiente a redes de conectividad completa, $\alpha_c \sim 0.138$. En el panel izquierdo de la figura 3 se compara el incremento del número de patrones críticos (p_c) que predice la teoría en redes de conectividad completa con simulaciones para este mismo tipo de redes, variando su tamaño. A su vez, se grafican los distintos valores de p_c que una red de 10000 neuronas puede almacenar al variar c . En el panel derecho se grafican los valores de α_c obtenidos de dicha simulación. Es posible resolver analíticamente un modelo de Hopfield ultra-diluido, donde $c \ll \ln(N)$,

y obtener que la capacidad de almacenamiento crítica alcanza un valor de $2/\pi$ [16]. Muchas de las características de estas redes ultra-diluidas pueden estudiarse mediante simulaciones en redes diluidas ($c \ll N$), cuyo tamaño no es exponencialmente mayor a su conectividad [17].

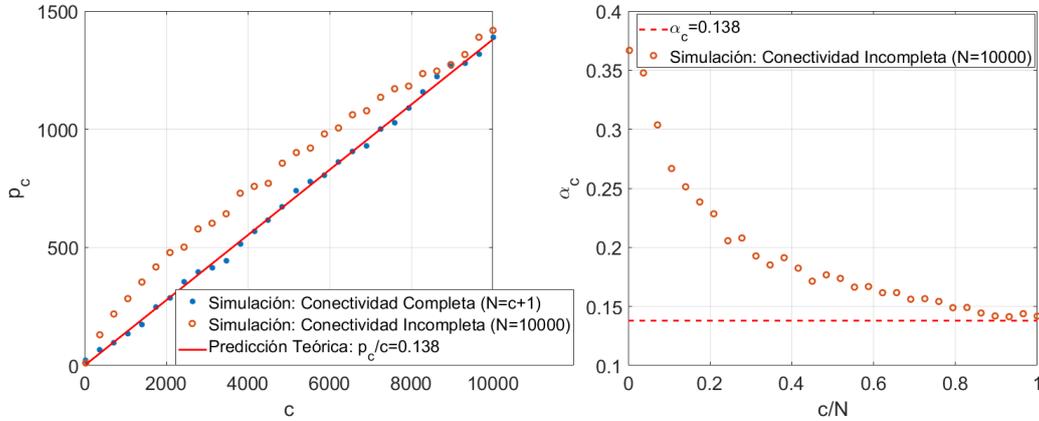


Figura 3: Izquierda: se grafica el número de patrones máximos que puede almacenar una red (p_c) en función del número de conexiones por neurona (c). Derecha: Se grafica $\alpha_c = p_c/c$ en función de c/N obtenido de las mismas simulaciones. La línea roja se corresponde con la predicción teórica para redes de conectividad completa (es decir que $c = N - 1$), cuya pendiente es 0.138; en azul los resultados de simulaciones de redes de conectividad completa (cada punto representa una red de tamaño $N = c + 1$); y en naranja los resultados de simulaciones para una red de $N = 10000$ variando el número de conexiones por neurona.

1.6.1. Análisis Señal-Ruido

La transición de fase que sufre la red después de sobrepasar su límite crítico de carga puede ser explicado a partir de un análisis de la señal y el ruido presente en el campo de actividad (h_i) que recibe cada neurona. A medida que se cargan patrones a la red, la probabilidad de que una neurona esté activa en más de un patrón aumenta. Como consecuencia, cuando la red trata de recuperar una memoria específica, cada neurona tiende a activar a aquellas neuronas con las que comparte patrones, sin importar que no se activen en el patrón que se quiere recuperar. Por lo tanto, se tiene interferencia entre patrones, y la misma aumenta conforme más patrones se almacenan en la red.

Suponiendo que se quiere recuperar el patrón $\bar{\xi}^\nu$, para que éste sea un punto fijo de la dinámica, basta pedir que, si se inicializa la red en el mismo patrón, es decir $s_i(t_0) = \xi_i^\nu$, entonces $s_i(t_0 + \Delta t) = s_i(t_0)$, para todo i (cuando el estado de la red a tiempo t_0 es idéntico al patrón ν , conviene usar la notación $h_i(t_0) = h_i^\nu$ para describir el campo de actividad que recibe cada neurona). Usando la regla de recurrencia indicada en la ecuación 2 se obtiene

$$\begin{aligned}
s_i(t_0 + \Delta t) &= \text{sgn}(h_i^\nu) = \text{sgn} \left(\sum_{\substack{j=1 \\ j \neq i}}^N J_{ij} \xi_j^\nu \right) \\
&= \text{sgn} \left(\frac{1}{c} \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{\mu=1}^p \xi_i^\mu \xi_j^\mu \xi_j^\nu c_{ij} \right) = \text{sgn} \left(\begin{array}{c} \boxed{\xi_i^\nu} \\ \text{Señal } (S_i^\nu) \end{array} + \begin{array}{c} \boxed{\frac{1}{c} \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^p \xi_i^\mu \xi_j^\mu \xi_j^\nu c_{ij}} \\ \text{Ruido } (R_i^\nu) \end{array} \right) \\
\rightarrow s_i(t_0 + \Delta t) &= \text{sgn}(\xi_i^\nu + R_i^\nu)
\end{aligned} \tag{9}$$

Entonces, si $\text{sgn}(\xi_i^\nu + R_i^\nu) = \xi_i^\nu$ (o equivalentemente $1 + R_i^\nu \xi_i^\nu > 0$) para toda neurona i , el patrón es recuperado de forma óptima. Por lo tanto, la estabilidad del patrón queda determinada por la probabilidad de que el ruido (R_i^ν) no haga que el campo (h_i^ν) tome un signo opuesto a la señal (ξ_i^ν). Se puede ver que el término de ruido no es más que el resultado de una caminata aleatoria en una dimensión de $c(p-1)$ pasos de largo $1/c$. Esto se debe a que ξ_i^ν y ξ_j^μ son variables independientes entre sí con probabilidad $1/2$ de valer 1 ó -1 , y por lo tanto, el producto $\xi_i^\mu \xi_j^\mu \xi_j^\nu$ también es una variable aleatoria con probabilidad $1/2$ de valer 1 ó -1 . Asimismo, la suma sobre N neuronas se transforma en una suma sobre c neuronas al azar ya que c_{ij} es una variable aleatoria independiente de los patrones. Si el número de patrones almacenados es alto, entonces se puede aproximar al resultado de dicha caminata aleatoria por el de una variable aleatoria proveniente de una distribución gaussiana con media nula ($\langle R_i^\nu \rangle = 0$) y desviación estándar $\sigma = \sqrt{(p-1)/c} \sim \sqrt{\alpha}$. En la bibliografía suele ser más común hablar de la distribución del ruido-señal $R_i^\nu \xi_i^\nu$ [1, 13], pero como la conectividad es aleatoria y los patrones no están correlacionados, ambas distribuciones son equivalentes. Como la desviación estándar de la distribución del ruido crece monótonamente con el número de patrones, existe un punto crítico a partir del cual las fluctuaciones de R_i^ν son lo suficientemente grandes como para desestabilizar a todos los patrones e impedir que puedan ser recuperados.

1.6.2. Tensor de Ruido

Una forma de representar al término de ruido es mediante un tensor de rango 3 al que se llamó *tensor de ruido* (Ω_{ij}^μ). Nuevamente, si se inicializa la red en el patrón que se desea recuperar, se puede reescribir al ruido de la siguiente forma

$$\begin{aligned}
R_i^\nu &= \sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\nu c_{ij} \\
\Omega_{ij}^\nu &= \frac{1}{c} \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^p \xi_i^\mu \xi_j^\mu \xi_j^\nu = \frac{1}{c} (W_{ij} - \xi_i^\nu \xi_j^\nu) \xi_j^\nu
\end{aligned} \tag{10}$$

Esto es simplemente un cambio en la notación, pero deja expuesta de forma más explícita una de las preguntas fundamentales en este trabajo. De la ecuación 9 se puede ver que si $R_i^\nu \xi_i^\nu \geq 0$ entonces se satisface la condición de punto fijo ¿Qué sucede si la conectividad, en vez de ser aleatoria, es tal que $(\xi_i^\nu \sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\nu c_{ij}) \geq 0$?

1.6.3. Trabajos Previos

Existen autores que exploraron conectividades no aleatorias con el fin de mejorar las propiedades de las redes autoasociativas. Montemurro y Tamarit presentaron un trabajo [18] donde propusieron una regla de poda de conexiones capaz de mejorar la capacidad de almacenamiento de la red. Dicha regla consistió en podar todas las conexiones presinápticas de cada neurona a excepción de las c sinapsis con máximo módulo de peso. Para entender la regla, se puede suponer un caso donde las c conexiones más fuertes por neurona tienen peso sináptico $W_{ij} = \pm p$. Entonces, dejando sólo estas conexiones, al recuperar un patrón $\bar{\xi}^\nu$, el término de ruido siempre refuerza la señal, ya que $\text{sgn}(W_{ij}) = \text{sgn}(\xi_i^\nu \xi_j^\nu)$

$$h_i^\nu \xi_i^\nu = \sum_{\substack{j=1 \\ j \neq i}}^N W_{ij} \xi_j^\nu \xi_i^\nu c_{ij} = \frac{p}{c} \sum_{\substack{j=1 \\ j \neq i}}^N c_{ij} = p > 0 \longrightarrow \text{sgn}(h_i^\nu) = \xi_i^\nu \text{ (es punto fijo)} \tag{11}$$

Siendo que los elementos de la matriz W_{ij} provienen de una distribución binomial (ec. 4), basta pedir que $c \ll N/2^{p-1}$ para garantizar que existan c sinapsis con peso $W_{ij} = \pm p$ por neurona. Utilizando las herramientas analíticas introducidas por Derrida *et. al.* [16], los autores demostraron que bajo la restricción $c \ll \min\{\ln(N), N/2^{p-1}\}$, la red carece de punto crítico donde deje de operar como memoria asociativa. Asimismo, bajo condiciones menos restrictivas ($c \ll N$), mostraron mediante simulaciones que limitando las conexiones de cada neurona a las c más fuertes (donde no necesariamente se satisface $|W_{ij}| = p$), la capacidad de almacenamiento mejora sustancialmente.

1.7. Cuenca de Atracción

Una de las características fundamentales de los atractores es su cuenca de atracción (o *basin of attraction*). Ésta representa el entorno de una memoria en el cual se puede encontrar el estado

inicial de la red y converger a la misma. En otras palabras, es una medida de qué tanta tolerancia tiene la red a errores en su estado inicial al momento de recuperar un patrón. Mediante simulaciones numéricas, es posible ver que en el modelo simplificado de Hopfield, la fuerza de los atractores depende del factor c/N . En el caso de redes diluidas $c/N \ll 1$, todos los patrones almacenados mantienen una tolerancia cercana al 50 % hasta alcanzar el punto crítico [17], donde la estabilidad sufre un salto abrupto a una tolerancia nula. A medida que se aumenta la conectividad en la red, lo que sucede es que la cuenca de atracción comienza a debilitarse un poco más suavemente, pero conserva una región donde se observa un salto abrupto en la estabilidad. En la figura 4 se ejemplifican 3 casos simulados en redes de 2000 neuronas.

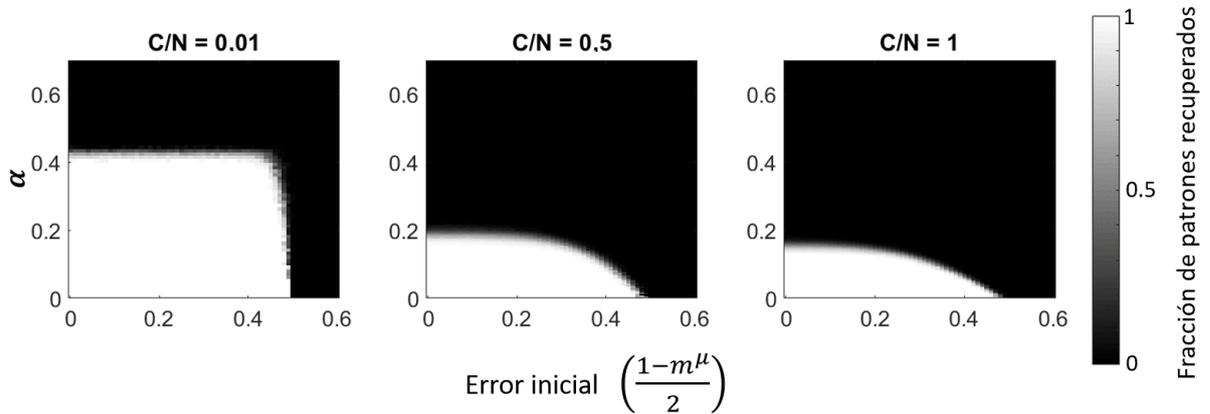


Figura 4: Se grafican la cuencas de atracción para redes de 2000 neuronas con distinta conectividad, representada con el factor c/N , obtenidas de simulaciones numéricas. El eje x indica la fracción de error inicial de la red con respecto al patrón que se desea recuperar. Dicha fracción se escribe en función del solapamiento del estado inicial de la red $m^\mu(t_0)$, con t_0 el tiempo inicial, de la forma $[1 - m^\mu(t_0)]/2$. Para ejemplificar, si la red busca recuperar el patrón $\bar{\xi}^\nu$, un 0 % de error inicial quiere decir que el estado inicial es idéntico a $\bar{\xi}^\nu$, y por lo tanto el *overlap* inicial es $m^\nu(t_0) = 1$; en cambio, un 50 % implica que la mitad de las neuronas de la red tienen el signo *contrario* al que les correspondería tener en dicho patrón, y por lo tanto $m^\nu(t_0) = 0$; porcentajes de error inicial mayores al 50 % implican $m^\nu(t_0) < 0$. El eje y indica el parámetro de carga de patrones en la red $\alpha = p/c$. La escala de grises indica la fracción de patrones del total almacenado que la red pudo recuperar exitosamente (es decir, que tuvo un *overlap* final $m^\nu(t_f) > 0.7$, con t_f el tiempo final).

1.8. Patrones de Baja Actividad Media

Existen muchos aspectos biológicamente implausibles en el modelo de Hopfield. Entre ellos, está el hecho de que los patrones de actividad almacenados requieren que un 50 % de las neuronas de la red estén activas. Datos neurofisiológicos indican que en las regiones de la corteza donde se detectan funcionalidades asociativas, las neuronas activas en un patrón de actividad dado corresponden a un 4 % – 5 % de la red [1]. Normalmente, las modificaciones más efectivas del modelo de Hopfield que dan cuenta de esta característica describen a los patrones de actividad a almacenar como variables discretas $\eta_i^\mu \in \{0, 1\}$ (en vez de $\xi_i^\mu \in \{-1, 1\}$) provenientes de la siguiente distribución [1, 19, 20]

$$P(\eta_i^\mu) = a\delta(\eta_i^\mu - 1) + (1 - a)\delta(\eta_i^\mu) \quad (12)$$

donde el parámetro $0 < a \leq 0.5$ indica la fracción de neuronas que se encuentran activas en el patrón $\bar{\eta}^\mu$. Al introducir este *sesgo* en los patrones junto con el cambio de variables, la actividad media de la red pasa a ser $\sum_{j=1}^N \eta_j^\mu = Na$. Asimismo, este cambio hace desaparecer los estados espurios simétricos mencionados en la sección 1.4. Por último, en el caso de patrones no sesgados ($a = 0.5$), el cambio de variables $\xi_i^\mu = 2\eta_i^\mu - 1 \rightarrow \xi_i^\mu \in \{-1, 1\}$ lleva al modelo original.

Esta modificación conlleva a cambios en algunas de las definiciones presentadas previamente. Antes que nada, los estados neuronales discretos ahora están dados por $s_i \in \{0, 1\}$, por lo tanto, la regla de recurrencia debe modificarse de la siguiente forma

$$s_i(t + \Delta t) = \frac{1}{2} [\text{sgn}(h_i(t) - U) + 1] \quad (13)$$

Por otro lado, si se construye la matriz de pesos sinápticos mediante la prescripción clásica ($W_{ij} = \sum_{\mu=1}^p \eta_i^\mu \eta_j^\mu$) se obtiene que el valor medio del ruido deja de ser nulo

$$\langle R_i^\nu \rangle \propto \left\langle \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^p \eta_i^\mu \eta_j^\mu \eta_j^\nu c_{ij} \right\rangle = ca^2 \left\langle \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^p \eta_i^\mu \right\rangle = ca^3(p - 1) \neq 0 \quad (14)$$

donde se utilizó que $\left\langle \sum_{\substack{j=1 \\ j \neq i}}^N \eta_j^\mu \eta_j^\nu c_{ij} \right\rangle = ca^2$ y $\left\langle \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^p \eta_i^\mu \right\rangle = a(p - 1)$ (y se hizo uso de la independencia estadística entre las variables η_i^μ). Dado que el valor medio del ruido es proporcional al número de patrones almacenados ($\langle R_i^\nu \rangle \propto p$), la capacidad de almacenamiento de la red disminuye de forma abrupta. La siguiente modificación sobre cada elemento W_{ij} soluciona este problema, anulando el valor medio del ruido. En consecuencia, la capacidad de almacenamiento queda determinada, nuevamente, por la magnitud de las fluctuaciones del ruido (es decir, su desviación estándar σ)

$$W_{ij} = \sum_{\mu=1}^p (\eta_i^\mu - a)(\eta_j^\mu - a) \longrightarrow \begin{cases} \langle R_i^\nu \rangle & = 0 \\ \sigma & = \sqrt{a\alpha} \end{cases} \quad (15)$$

A partir de la modificación introducida en la ecuación 15, se redefinen la matriz de adyacencia, tensor de ruido y solapamiento de la siguiente forma

$$\begin{aligned}
J_{ij} &= \frac{1}{ca(1-a)} c_{ij} W_{ij} \\
\Omega_{ij}^\nu &= \frac{1}{ca(1-a)} \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^p (\eta_i^\mu - a)(\eta_j^\mu - a) \eta_j^\nu c_{ij} \\
m^\nu &= \frac{1}{Na(1-a)} \sum_{j=1}^N (\eta_j^\nu - a) s_j
\end{aligned} \tag{16}$$

Por lo tanto, el campo de actividad local que recibe cada neurona al inicializar la red en el patrón $\bar{\eta}^\nu$ es

$$\begin{aligned}
h_i^\nu - U &= \frac{1}{ca(1-a)} \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{\mu=1}^p (\eta_i^\mu - a)(\eta_j^\mu - a) \eta_j^\nu c_{ij} - U \\
&= \boxed{\frac{1}{ca(1-a)} (\eta_i^\nu - a) \sum_{\substack{j=1 \\ j \neq i}}^N (\eta_j^\nu - a) \eta_j^\nu c_{ij}} + \boxed{\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\nu c_{ij}} - U \\
&\quad \text{Señal (S}_i^\nu\text{)} \qquad \qquad \qquad \text{Ruido (R}_i^\nu\text{)}
\end{aligned} \tag{17}$$

La señal presente en el campo de actividad ya no toma valores discretos ± 1 sino que proviene de una distribución bimodal, con un pico en $-a$ si $\eta_i^\nu = 0$ y otro en $(1-a)$ si $\eta_i^\nu = 1$

$$S_i^\nu \sim \frac{1}{ca(1-a)} (\eta_i^\nu - a) \left\langle \sum_{\substack{j=1 \\ j \neq i}}^N (\eta_j^\nu - a) \eta_j^\nu c_{ij} \right\rangle = \eta_i^\nu - a \tag{18}$$

Como la estabilidad de la memoria depende de la actividad de las neuronas más débiles, es necesario ecualizar las contribuciones de los dos picos de la señal. Para ello, se elige usar el siguiente umbral (notar que en el caso $a = 0.5$, se recupera $U = 0$)

$$U = \frac{1}{2} - a \tag{19}$$

y de esta forma, los dos estados neuronales posibles satisfacen

$$|S_i^v - U| = \frac{1}{2} \quad (20)$$

Uno de los resultados más importantes de esta modificación es que la capacidad de almacenamiento escala como $\alpha_c \propto \frac{1}{a \ln(a)}$. En la figura 5 se muestra cómo aumenta la capacidad de almacenamiento a medida que decrece a , manteniendo constante c/N .

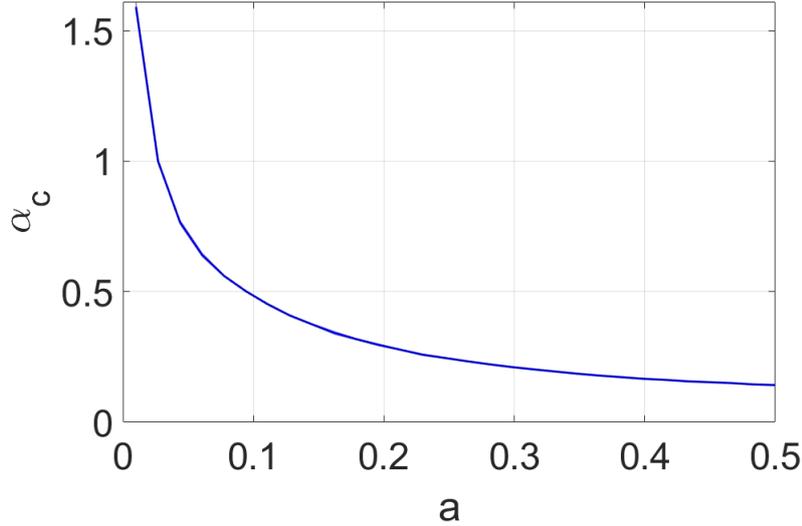


Figura 5: Se grafica el parámetro de almacenamiento crítico ($\alpha_c = p_c/c$) en función de la fracción de neuronas activas en un patrón (a). Los puntos fueron obtenidos de promediar los resultados de 10 simulaciones realizadas para redes de $N = 10000$ y $c = 2000$ (se grafica la banda de la desviación estándar si bien no se aprecia).

1.9. Algoritmos de Optimización Metaheurísticos

A continuación, se introducen dos algoritmos de optimización discreta metaheurísticos para minimizar una función objetivo. La idea es dejar plasmado su funcionamiento, dado que en este trabajo fue necesario resolver un problema de optimización duro. Este tipo de problemas no puede ser resuelto de forma óptima mediante un algoritmo de optimización exacto en un tiempo razonable. Para poder hallar soluciones satisfactorias, puede utilizarse algoritmos metaheurísticos. Éstos se caracterizan por hacer uso de recursos estocásticos (generar variables aleatorias) y suelen estar inspirados en procesos de la naturaleza (físicos y biológicos).

1.9.1. Enfriamiento Simulado

El enfriamiento simulado [21] imita el proceso de recocido en el procesamiento de materiales cuando se requiere que un metal se enfríe y alcance un estado sólido ‘bien ordenado’ de mínima energía, evitando los estados ‘metaestables’ propios de mínimos locales. Lo que busca hacer el algoritmo es minimizar una función de costo $f(x)$, análoga a la energía del metal, a partir de un protocolo de enfriamiento, que en realidad, es la reducción de un parámetro T que se asocia a una temperatura.

El algoritmo comienza eligiendo una solución al azar e inicializando la temperatura $T = T_0$. Luego, en cada iteración se selecciona una solución x' en un entorno cercano a la solución actual x . La solución x' es aceptada de acuerdo al parámetro T y a la variación de la función de costo. Si la función de costo decrece, es decir $\Delta f = f(x') - f(x) < 0$, entonces se acepta el cambio. Por otro lado, si no decrece, el cambio se acepta con una probabilidad de transición $P_{tr} = \exp(-\Delta f/T)$. En el caso de que las soluciones sean vectores de variables discretas \bar{X} , una forma de generar un cambio en un entorno cercano a la solución es eligiendo uno de sus elementos al azar y sumando o restándole una unidad $X_j \rightarrow X_j \pm 1$. A este procedimiento se lo llama *single-spin-flip dynamics* [22].

Durante el proceso de búsqueda, la temperatura se va disminuyendo de acuerdo a un protocolo establecido. Uno de los protocolos más usados es $T_{k+1} = \gamma T_k$ (donde el subíndice k indica el número de iteración), con $0.7 \leq \gamma \leq 0.99$, si bien existen muchas variantes [23, 24, 25]. El programa de enfriamiento debe ser lo suficientemente lento como para garantizar una mejor convergencia al mínimo absoluto. Una forma de conseguir esto es incluir entre cada cambio de temperatura, iteraciones donde se generan cambios en la solución (a temperatura constante), permitiéndole al sistema alcanzar un *equilibrio térmico*.

Por último, es necesario determinar una temperatura inicial T_0 correcta. Debe ser lo suficientemente alta como para permitir una exploración amplia del espacio de soluciones en un estadio inicial (notar que cuando $T \rightarrow 0$ entonces $P_{tr} \rightarrow 0$), sin enlentecer demasiado el proceso de minimización. En este trabajo se utilizó el siguiente método, si bien existen muchas formas de hacer esto [23, 26]

$$T_0 = -\frac{\text{máx}(\Delta f)}{\ln(P_0)} \quad (21)$$

donde $\text{máx}(\Delta f)$ se encuentra numéricamente, realizando muchas transiciones aleatorias, y P_0 es la probabilidad de transición que se desea tener para la temperatura inicial. A medida que el protocolo de enfriamiento avanza, la temperatura desciende, disminuyendo la probabilidad de transición, y por lo tanto, refinando la búsqueda de la solución óptima. El algoritmo se detiene cuando la temperatura alcanza un umbral cercano al cero (T_f), o cuando la solución se estabiliza (no sufre cambios).

1.9.2. Algoritmo Genético

Este algoritmo está inspirado en la mecánica de selección natural y busca optimizar una función de costo mediante el uso de *operadores genéticos*, entre ellos, cruza, mutación y selección [27]. El algoritmo procede de a pasos llamados generaciones. En cada generación, una nueva población es creada a partir de las soluciones anteriores mediante el uso de los operadores genéticos. A medida que se avanza de generación, las soluciones van minimizando la función de costo (se van *adaptando*) hasta alcanzar el objetivo deseado.

El esquema del algoritmo es el siguiente. Se comienza con una población inicial de M soluciones generadas aleatoriamente, lo suficientemente grande como para cubrir una fracción considerable del espacio de búsqueda. De esta población, se seleccionan $m < M/2$ parejas de soluciones que minimicen la función de costo. Una vez seleccionadas, se aplica el operador de cruce generando K crías por pareja. En esta cruce se combinan elementos de las dos soluciones padres de forma aleatoria. Por lo general, las poblaciones consisten en vectores binarios, por lo que se seleccionan aleatoriamente sus elementos para formar un nuevo vector, como se muestra en la figura 6. En dicha cruce puede aplicarse un operador de mutación, que perturbe leve y aleatoriamente algunos elementos de las crías. Luego, se seleccionan las $k < K$ crías, por cada pareja, que minimicen la función de costo, y éstas pasan a formar la nueva población. El proceso se repite hasta que la mínima solución de cada generación se estanca en un valor por un determinado número de iteraciones, o hasta que se supera un umbral deseado para el objetivo buscado [27, 28].

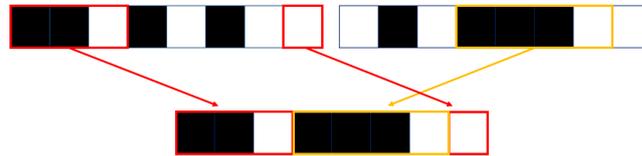


Figura 6: Esquema de cómo se cruzan dos vectores binarios padres aleatoriamente para dar origen a una cría. En este caso, los casilleros negros y blancos representan 0 y 1, respectivamente.

2. Métodos

2.1. Problema de Optimización: Minimización del Ruido

Como ya se mencionó previamente, la inestabilidad de las memorias en una red de Hopfield puede explicarse como un aumento sustancial de las fluctuaciones aleatorias del término de ruido, que impiden que el campo de actividad que reciben algunas neuronas conserve el signo de la señal. Dado que el término de ruido depende directamente de la conectividad presináptica de cada neurona, la pregunta fundamental en este trabajo es si es posible anular el aporte del ruido que recibe cada neurona reconfigurando su conectividad presináptica. Y de ser posible, ¿llevaría esto a una mejora en la capacidad de almacenamiento de la red? La respuesta a esta pregunta podría hallarse resolviendo el siguiente problema:

Dada una red de Hopfield de N neuronas con p patrones almacenados, encontrar para cada neurona i , una configuración de conectividad presináptica c_{ij} tal que para todo patrón μ se satisfaga

$$\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^{\mu} c_{ij} = 0 \quad (22)$$

Si $\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^{\mu} c_{ij} \equiv R_i^{\mu} = 0$ para todo patrón μ y neurona i (con $c_{ii} = 0$), se garantiza la estabilidad de las memorias. El problema tiene una complejidad alta ya que $c_{ij} \in \{0, 1\}$ y el número total de conexiones por neurona está limitado a c (sin esta última restricción, la solución al problema sería la trivial $\bar{c}_i = \bar{0}$). Es importante notar que si c_{ij} no tuviese restricciones, el problema sería trivial y se reduciría a encontrar el espacio nulo de la matriz $\bar{\bar{\Omega}}_i^1$ para cada neurona.

Una forma de relajar este problema para obtener alguna solución que aproxime el resultado deseado es pedir que el módulo de cada término de ruido $|R_i^{\mu}|$ se minimice en vez de anularse. Se llega entonces a un problema de minimización multiobjetivo, donde, para cada neurona, la configuración óptima de c_{ij} debe minimizar p variables simultáneamente (una por cada patrón almacenado). En esta tesis se decidió llevar la resolución de este problema de optimización multiobjetivo a la de un único objetivo: una *función de energía* o de costo. Para ello, una función de costo admisible, cuya minimización implica la minimización simultánea de cada término $|R_i^{\mu}|$, es la suma

$$\sum_{\mu=1}^p |R_i^{\mu}| \quad (23)$$

Sin embargo, es posible obtener una función de costo con más significado físico. Cada neurona

¹Cuando se habla de la matriz $\bar{\bar{\Omega}}_i$, corresponde a la componente i (que es una matriz en el espacio $\mathbb{R}^{p \times N}$) del tensor Ω_{ij}^{μ} de rango 3.

i recibe p aportes de ruido R_i^μ . Si la conectividad es aleatoria, el valor medio de dicha distribución es nulo ($\langle R_i^\mu \rangle = 0$, ver sección 1.6.1). Por lo tanto, la desviación estándar de la distribución de ruido por neurona es

$$\sigma_i^2 = \frac{1}{p} \sum_{\mu=1}^p (R_i^\mu - \langle R_i^\mu \rangle)^2 = \frac{1}{p} \sum_{\mu=1}^p \left(\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu c_{ij} \right)^2 \quad (24)$$

Entonces, minimizar la suma de los cuadrados del ruido puede interpretarse como minimizar las fluctuaciones de ruido, asumiendo que su valor medio será nulo. Si se consigue una conectividad c_{ij} tal que $\sigma_i = 0$ (valor mínimo posible), entonces también se satisface que el ruido se anula ($\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu c_{ij} = 0$) para todo patrón μ , que es lo que se busca en la ecuación 22. Está claro entonces que cualquier algoritmo de optimización que busque minimizar σ_i tratará de llevarlo al valor más cercano a cero posible.

Con el objetivo de poder comparar las características de redes optimizadas con redes de conectividad aleatoria, resultó conveniente utilizar como parámetro de control la conectividad presináptica por neurona. Si se quiere optimizar una red con c conexiones por neurona, el problema de optimización sujeto a restricciones puede expresarse de la siguiente forma

Minimizar:

$$E_i^0 = \sum_{\mu=1}^p \left(\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu c_{ij} \right)^2$$

Sujeto a:

$$\begin{aligned} c_{ij} &\in \{0, 1\} \\ \sum_{\substack{j=1 \\ j \neq i}}^N c_{ij} &= c \end{aligned}$$

donde se nombró E_i^0 a la función de costo a minimizar.

El problema de optimización planteado pertenece a la familia de problemas de optimización cuadrática binaria con restricciones. Existen muchos métodos con los cuales abordarlo. Particularmente, es posible clasificar a estos métodos en dos grupos principales: exactos y heurísticos. De forma resumida, los métodos exactos permiten resolver el problema encontrando el mínimo absoluto, pero con un costo computacional muy grande. Algunos ejemplos son la adaptación del método simplex a su forma binaria o métodos de optimización cuadrática binaria [29, 30]. Por otro lado, se tienen los métodos de optimización heurísticos [31], que implican un costo y

tiempo computacional mucho menor que los métodos exactos, pero, por lo general, no hallan un mínimo absoluto sino una aproximación del mismo. Qué tan buena es la solución que encuentre el algoritmo depende del método y problema a resolver.

La razón por la cual se recurrió a métodos heurísticos y no exactos radica en que la complejidad del problema de optimización a resolver incrementa rápidamente con el tamaño de la red. Se puede definir la multiplicidad del problema Γ como el número de configuraciones posibles de la conectividad presináptica de cada neurona

$$\Gamma = \binom{N-1}{c} \quad (25)$$

Esta multiplicidad determina el tamaño del espacio de búsqueda en el que un algoritmo debe hallar la solución óptima. A c fijo, Γ incrementa como N^{c-1} y el tiempo empleado por algoritmos de optimización exactos se hace muy grande para los valores de N con los que se trabajó en esta tesis.

En la introducción se describieron a los dos algoritmos que se contemplaron en este trabajo: algoritmo genético y enfriamiento simulado. En la subsección siguiente se comparan algunos resultados entre el rendimiento de ambos algoritmos para justificar la elección del enfriamiento simulado como método utilizado por el resto del trabajo.

2.2. Algoritmos Implementados

Tanto el algoritmo genético como el de enfriamiento simulado fueron implementados en MATLAB desde primeros principios. La forma de implementar los algoritmos consistió en reconfigurar para cada neurona su conectividad presináptica minimizando E_i^0 .

2.2.1. Adaptación del Enfriamiento Simulado

Se adaptó el algoritmo de enfriamiento simulado ya introducido (ver sección 1.9) modificando levemente la dinámica *single-spin-flip dynamics* para imponer la restricción de conectividad constante. En vez de generar cambios aleatorios de conexiones de forma independiente, se generaron cambios de a pares de conexiones: para cada iteración se seleccionó un par de neuronas y se intercambiaron su estado de conexión. En otras palabras, dados dos estados de conexión aleatorios $c_{ik} = 1$ y $c_{il} = 0$ (siempre tomando $k \neq l \neq i$), se realizó la siguiente permutación

$$\begin{array}{ccc} c_{ik} = 1 & \longrightarrow & c_{ik} = 0 \\ c_{il} = 0 & & c_{il} = 1 \end{array} \quad (26)$$

Intercambio

Dicho intercambio se aceptó sólo frente a una disminución de la función de costo, o en caso contrario, con una probabilidad $e^{-\Delta E_i^0/T}$.

2.2.2. Adaptación del Algoritmo Genético

Este algoritmo no requirió de mayores adaptaciones con respecto a su forma de operar explicada en la sección 1.9. Al momento de realizar la crua entre dos vectores de conectividad, se definió la fracción g de elementos con respecto al total $(N - 1)$ que la solución con menor energía ($E_i^{0,\text{mín}}$) transmitiría a su herencia

$$g = 1 - \frac{E_i^{0,\text{mín}}}{2E_i^{0,\text{máx}}} \quad (27)$$

Esta fue una forma de asegurarse de que aquel padre que tuviese una menor energía tuviese una mayor contribución sobre la configuración de la conectividad *cría*. En los casos donde la función de energía fue la misma para ambos padres, ambos contribuyeron equitativamente en la herencia de su cría. Por ejemplo, si se retoma la figura 6, el caso correspondería a dos padres con mismo valor de su función de costo, ya que $g = 1/2$.

La segunda adaptación tuvo que ver con la imposición de la restricción de conectividad constante. Para cumplirla, se analizaba cada cría de forma individual. Si el número de conexiones era mayor a c , entonces se cortaban conexiones de forma aleatoria hasta conseguir la conectividad deseada. En cambio, si el número de conexiones era menor, se seleccionaban conexiones de los padres de forma aleatoria hasta alcanzar el valor deseado.

2.3. Comparación entre Algoritmos

Para comparar el rendimiento de ambos algoritmos, se buscó para cada uno el conjunto de parámetros a definir, de forma tal que el funcionamiento del algoritmo fuese óptimo (es decir, que maximizara la calidad de la solución total y minimizara el tiempo de cómputo requerido). En el caso del enfriamiento simulado, se determinó la temperatura inicial T_0 de forma numérica utilizando la ecuación 21 mediante el método explicado en la sección 1.9.1; y la temperatura final se la fijó en $T_f = 10^{-10}$. Se eligió el factor de enfriamiento $\gamma = 0.99$, correspondiente al protocolo indicado en la misma sección. Elegir un γ aún más alto hubiese implicado un enlentecimiento del algoritmo sin una sustancial mejora en la calidad de solución, de acuerdo con simulaciones de prueba realizadas. Para garantizar una mejor convergencia al mínimo, se estableció un número de 100 iteraciones de estabilización térmica. Por último, se le indicó frenar al algoritmo una vez que el valor de la función de costo se estancase por 500 iteraciones.

En cuanto al algoritmo genético, se eligió el número de la población inicial de soluciones aleatorias igual al tamaño de la red N (se verificó que este parámetro no tuvo mayor importancia en el rendimiento del algoritmo siempre que fuera lo suficientemente grande). La elección de algunos parámetros fue compleja, dado que presentaron dependencias con N , c y p . Mediante simulaciones, se observó que la mejor elección fue tomar como número de parejas a un décimo de la población inicial total ($N/10$) y un número de 5 crías por pareja (éstos son los parámetros que permitieron obtener buenas soluciones en un tiempo considerablemente corto, pero su elección fue subjetiva). Asimismo, no se hizo uso de operadores de mutación, dado que no introdujeron cambios significativos en los resultados, que se verificó en simulaciones de prueba. Por último, nuevamente se le indicó frenar al algoritmo una vez que el valor de la función de costo se estancase por 500 iteraciones, para que fuese comparable con el enfriamiento simulado.

Para comparar los tiempos de cómputo entre algoritmos, se optimizó la conectividad de neuronas individuales, pertenecientes a redes con distintos valores de N , c y p . Se obtuvo que la tasa de crecimiento del tiempo empleado por el algoritmo genético en función de N fue mayor que la del enfriamiento simulado, como se muestra en la figura 7.

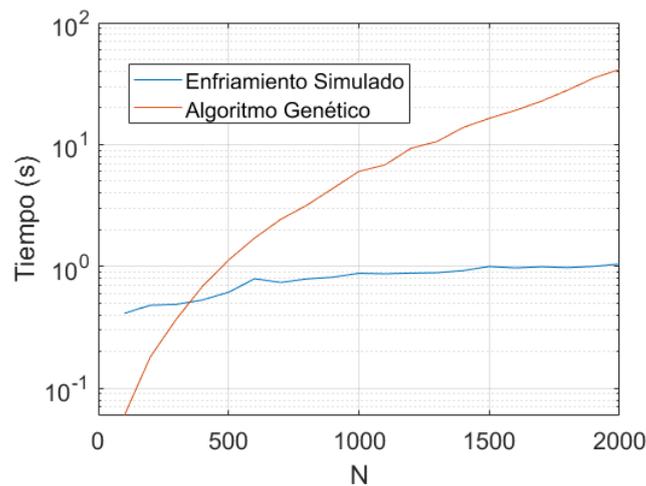


Figura 7: Curvas del tiempo implementado por cada algoritmo al optimizar la conectividad de una neurona (mediante la minimización de E_i^0) perteneciente a una red con 300 patrones almacenados, variando N y manteniendo constante la fracción $c/N = 0.5$.

Se comparó también la calidad de las soluciones entre ambos algoritmos para distintas combinaciones de N , c y p . Se obtuvo que el enfriamiento simulado minimizó mejor a la función de costo, en comparación con el algoritmo genético, como se muestra en el gráfico izquierdo de la figura 8. En el gráfico derecho de la misma figura se compara la evolución de la función de costo en una optimización para una neurona elegida al azar, dando cuenta del orden de magnitud de una solución hallada al azar (correspondiente a los tiempos iniciales) y la solución óptima alcanzada (tiempo final).

En resumen, el algoritmo de enfriamiento simulado consiguió minimizar mejor la función de costo para distintas combinaciones de N , c y p . Simultáneamente, el mismo implementó menos tiempo que el algoritmo genético para resolver el problema. Por lo tanto, se decidió utilizar el algoritmo de enfriamiento simulado como método de optimización por el resto del trabajo.

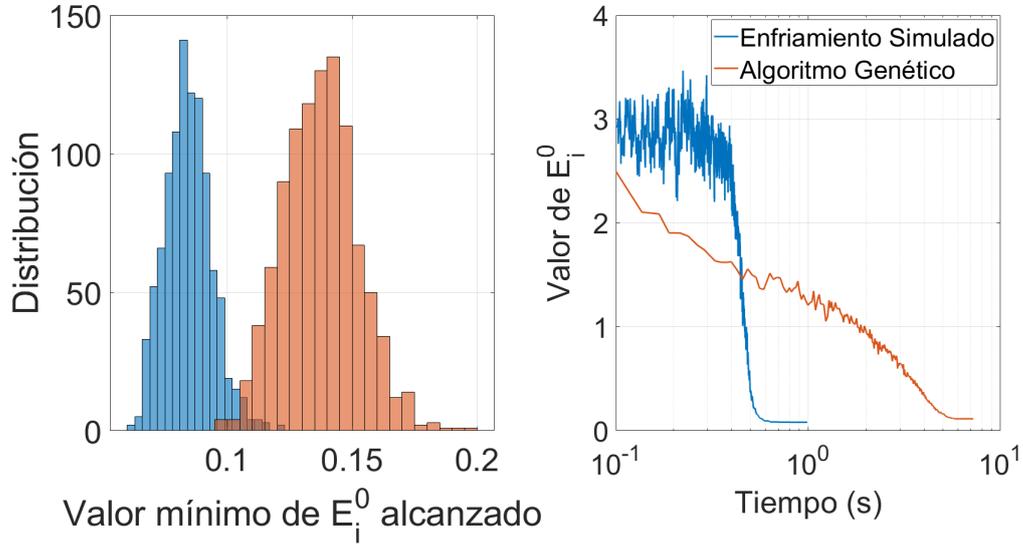


Figura 8: Izquierda: histograma de soluciones obtenidas al optimizar la conectividad de una neurona a partir de la minimización de la función de costo E_i^0 , mediante los métodos de enfriamiento simulado (azul) y algoritmo genético (naranja). Derecha: valor de la función de costo E_i^0 (correspondiente a una neurona elegida al azar) en función del tiempo (tomando como tiempo inicial $t = 0.01s$) al ser minimizada por cada uno de los algoritmos implementados. Los datos de ambas figuras corresponden a la optimización de una red de 1000 neuronas, 500 conexiones por neurona y 70 patrones almacenados.

3. Optimización de la Red de Hopfield

3.1. Búsqueda Binaria

Para encontrar el límite máximo (p_{opt}) de la capacidad de almacenamiento de una red de N neuronas y c conexiones por neurona mediante una optimización de enfriamiento simulado se realizó una búsqueda binaria. Este tipo de algoritmo permite encontrar la posición de un valor dentro de un vector ordenado. Para realizar la búsqueda, se propuso una carga de memorias inicial, que fuese el doble de la capacidad de almacenamiento de una misma red con conectividad aleatoria, desde ahora en más p_0 (esta elección fue puramente arbitraria). Se estableció al intervalo $[0, N]$ como límite inicial de la búsqueda. Para esta propuesta inicial, se inicializó una red de conectividad aleatoria, se le cargaron $2p_0$ patrones y se optimizó la conectividad presináptica de cada neurona de forma tal que se minimizara la función de energía E_i^0 , mediante el algoritmo de enfriamiento simulado

$$E_i^0 = \sum_{\mu=1}^p \left(\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^{\mu} c_{ij} \right)^2 \quad (28)$$

Una vez optimizadas todas las conectividades neuronales, se estudió la estabilidad de los patrones almacenados. Para ello, se inicializó a la red en cada patrón ($\bar{s}(t_0) = \bar{\xi}^{\mu}$) y se la dejó evolucionar en el tiempo (por 100 iteraciones o hasta alcanzar un estado estacionario en el que el *overlap* m^{μ} no variara por 10 iteraciones). Los patrones considerados estables fueron aquellos cuyo solapamiento con el estado final de la red fue mayor a 0.7. Una vez analizada la estabilidad, se redefinieron los límites de búsqueda de la siguiente manera: si todos los patrones fueron estables, entonces se tomó a la carga $2p_0$ como límite inferior de la búsqueda; en cambio, si al menos un patrón fue inestable, entonces la carga $2p_0$ pasó a tomarse como límite superior de la búsqueda. Una vez redefinido el nuevo intervalo, el procedimiento se repitió para una carga igual al valor medio entre los nuevos límites de búsqueda. El algoritmo iteró hasta alcanzar una diferencia menor o igual a 1 entre el límite superior e inferior. Como se esquematiza en la figura 9, esto permitió encontrar p_{opt} evaluando en un mínimo de puntos entre 0 y N .

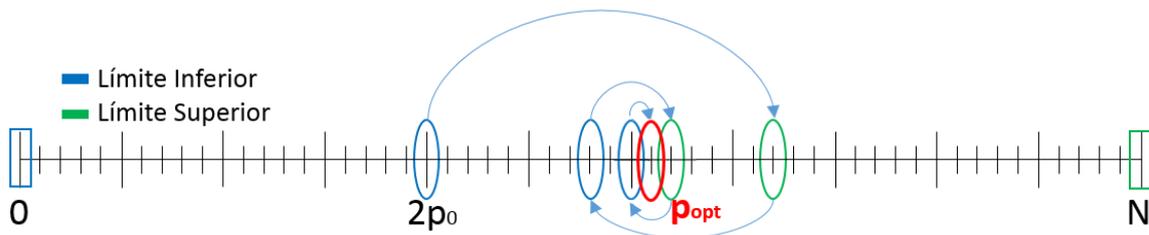


Figura 9: Esquema de una búsqueda binaria. La búsqueda comienza en $2p_0$, con límites iniciales de búsqueda $[0, N]$. Se van redefiniendo sucesivamente los límites de exploración (en azul los límites inferiores, y en verde los superiores) hasta encontrar el valor óptimo p_{opt} .

3.2. Capacidad de Almacenamiento

A partir de la búsqueda binaria utilizando el algoritmo de enfriamiento simulado, se obtuvo una mejora en la capacidad de almacenamiento en función de la conectividad de la red. Se observa en el gráfico de la izquierda de la figura 10 que esta mejora no está relacionada con efectos de tamaño finito ya que persiste para distintos tamaños de red. Esto es importante porque resultados de simulaciones con redes chicas y menos costosas computacionalmente pueden extrapolarse a redes de mayor tamaño. Cabe destacar que la mejora se cuantifica mediante el cociente entre el número máximo de patrones que puede almacenar una red de conectividad optimizada y el número máximo de patrones que puede almacenar la misma red pero con una configuración aleatoria de conexiones (p_{opt}/p_0).

A medida que $c/N \rightarrow 1$, la razón de mejora p_{opt}/p_0 tiende a 1. Esto es esperable ya que los grados de libertad para elegir la matriz de conectividad tienden a cero. El coeficiente de mejora alcanza un máximo en un entorno cercano a $c/N \sim 0.2$. Para $c/N \sim 0$ se observa que, si bien los grados de libertad para elegir la conectividad se reducen (al igual que $c/N \sim 1$), el cociente p_{opt}/p_0 se mantiene por encima de 3.

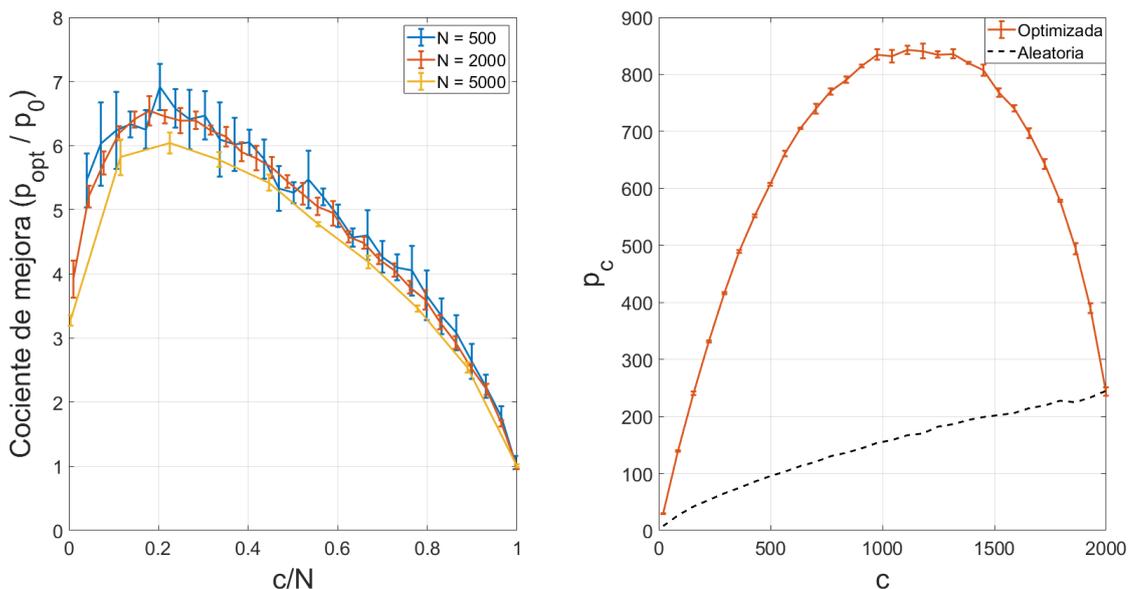


Figura 10: Mejora en la capacidad de almacenamiento de una red de Hopfield mediante la optimización de su matriz de conexiones a partir de la minimización de E_i^0 . Izquierda: se grafica la fracción de mejora (p_{opt}/p_0) en función del factor de conectividad c/N para redes de 500, 2000 y 5000 neuronas. Derecha: se grafica el número máximo de patrones que puede almacenar una red optimizada de 2000 neuronas en función de la conectividad por neurona. La línea punteada negra representa la capacidad de almacenamiento de una red de conectividad aleatoria de 2000 neuronas. El símbolo p_c representa la capacidad crítica de la red, ya sea optimizada (p_{opt}) o aleatoria (p_0). Todos los gráficos se obtuvieron de promediar las capacidades de almacenamiento provenientes de 5 optimizaciones distintas para cada N . Los errores provienen de la desviación estándar de dicho promedio (en el gráfico derecho), y en el caso del gráfico izquierdo, la correspondiente propagación de error.

Otro resultado importante de estas simulaciones es que, mientras en una red de Hopfield tra-

dicional la capacidad de almacenamiento aumenta monótonamente con el número de conexiones por neurona, siendo la red de conectividad completa la de mejor *performance*, en una red optimizada esto no se cumple. Se observa que para una red de N neuronas, la máxima capacidad de la misma se da para una conectividad en el rango $0.4 \leq c/N \leq 0.7$ (gráfico derecho de la figura 10).

3.3. Estudio de la Relación entre los Pesos Sinápticos y la Conectividad

Resulta interesante preguntarse de qué forma puede este proceso de optimización correlacionar la elección de conexiones con los pesos sinápticos. Repasando lo visto en la introducción, dada una red de N neuronas con c conexiones por neurona y p patrones almacenados, los elementos de matriz de W están restringidos al conjunto $\{2k - p \mid k \in \mathbb{Z}; 0 \leq k \leq p\}$. En este caso, la probabilidad de que $W_{ij} = 2k - p$ coincide con la de una distribución binomial $\mathcal{B}(x = k|p, 1/2)$. Por lo tanto, los elementos de la matriz de adyacencia ($J_{ij} = c_{ij}W_{ij}/c$) también provendrán de dicha distribución si la conectividad es aleatoria, es decir $\mathcal{P}\left(J_{ij} = \frac{2k-p}{c}\right) = \mathcal{B}(x = k|p, 1/2)$. Sin embargo, ¿qué sucede en una red optimizada? ¿Existe alguna correlación entre los pesos sinápticos y la probabilidad de que exista o no una conexión con dicho peso?

En esta sección, se abordan dos formas complementarias de estudiar los cambios en la matriz J de las redes optimizadas. Dicho análisis se realizó sobre los resultados de simulaciones de distintas redes de 5000 neuronas (este número extenso de neuronas permitió obtener una mejor estadística a la hora de contar las distintas sinapsis) cuya conectividad fue optimizada. Por un lado, en la figura 11 se comparan las distribuciones de pesos en la matriz de adyacencia J de redes optimizadas con las de redes de conectividad aleatoria para distintas conectividades. Éstas son las distribuciones de probabilidad de que un elemento de matriz J_{ij} tome el valor $\frac{2k-p}{c}$. Se observa que las distribuciones correspondientes a redes de conectividad optimizada tienen una tendencia bimodal (línea roja) para valores de c tal que $c/N < 0.5$. Esto difiere completamente de la distribución binomial que se observa en todas las redes aleatorias (línea negra). Este hecho sugiere que, en este rango de conectividad, la red tiende a cortar conexiones de pesos cercanos a cero. A medida que la conectividad crece, la red va recurriendo cada vez más a conexiones de peso cercano a cero. Una vez superado $c/N > 0.5$, la red recurre en exceso (comparando con el caso aleatorio) a seleccionar los pesos más bajos. Se observa que, en este rango de conectividad, las distribuciones de pesos son similares entre la red aleatoria y la optimizada, salvo en valores de pesos sinápticos cercanos a cero, donde difieren fuertemente. A medida que la conectividad crece, la distribución de pesos de la matriz de adyacencia tiende a una distribución binomial.

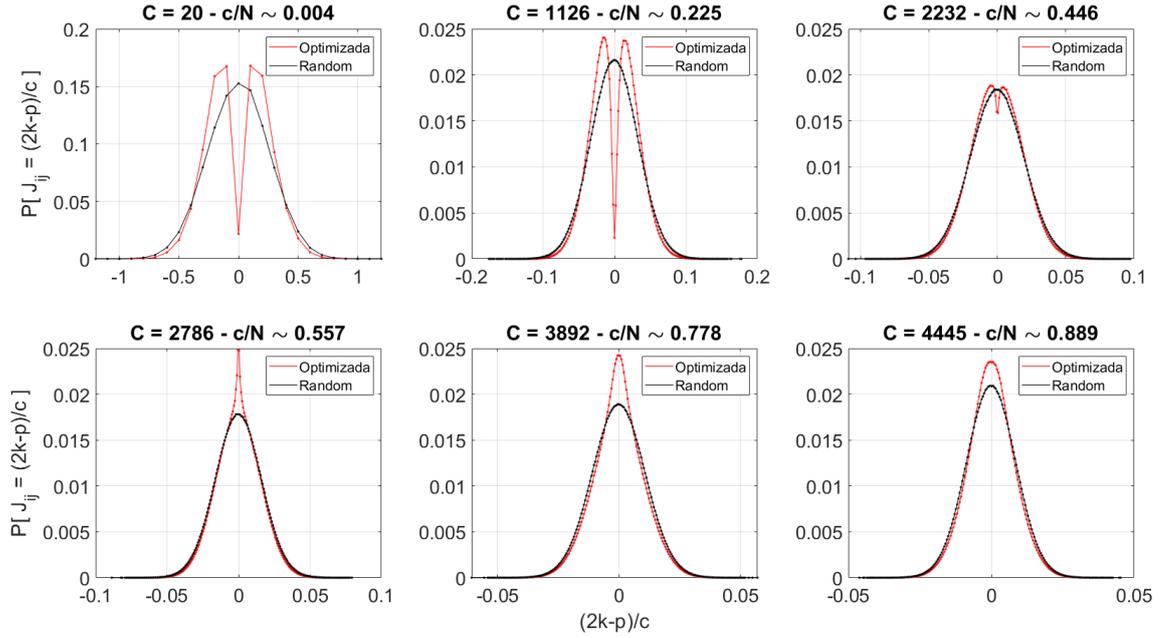


Figura 11: Se grafican las distribuciones de pesos $P[J_{ij} = (2k - p)/c]$ (con $k = 0, 1, \dots, p$) de la matriz de adyacencia J para redes de 5000 neuronas cuya conectividad fue optimizada a partir de la minimización de E_i^0 (llevando su capacidad al límite crítico). Cada panel corresponde a un número de conexiones por neurona distinto, indicado con C . Las curvas negras se corresponden con redes de conectividad aleatoria y las rojas con redes de conectividad optimizada.

Otra manera de mirar este problema es estimando la probabilidad de que, dado un peso sináptico W_{ij} entre dos neuronas, exista una conexión. La distribución de probabilidad condicional $\mathcal{P}(c_{ij} = 1 | W_{ij} = 2k - p)$ puede obtenerse a partir de los resultados de la simulación, como el cociente entre el número de elementos de matriz $J_{ij} = \frac{2k-p}{c}$ sobre el número de elementos de matriz $W_{ij} = 2k - p$. En la figura 12 se grafican estas distribuciones en función de W_k^M , definido como

$$W_k^M = \frac{2k - p}{\max\{|W_{ij}|\}} \quad (29)$$

En la ecuación 29 se normalizan los pesos sinápticos por el valor máximo que alcanza la matriz W (sin tener en cuenta los elementos de su diagonal) en una red particular. Esto permite comparar los pesos de redes de distinta conectividad, con distintas capacidades de almacenamiento. Para una red de conectividad aleatoria lo que se espera observar es que este cociente sea constante para todo valor de los pesos sinápticos. En particular, se espera que

$$\mathcal{P}(c_{ij} = 1 | W_{ij} = 2k - p) = c/N \quad (30)$$

En la figura 12 se muestra que para conectividades bajas, la red evita los pesos muy cercanos

a cero (conclusión a la que se llegó en el gráfico anterior), al mismo tiempo que existe una tendencia a evitar los pesos de módulo alto. Por otro lado, en redes de conectividad alta ($c/N > 0.5$), sigue estando esta tendencia a evitar pesos altos ($|W_k^M| \sim 1$), sumado a que la red comienza a hacer uso de las sinapsis con peso muy bajo ($|W_k^M| \sim 0$). Para dar una interpretación a estos resultados, conviene recordar que en la sección 3.2 se mostró que la conectividad que permite alcanzar la capacidad máxima de almacenamiento a una red de N neuronas se encuentra en el rango $0.4 \leq c/N \leq 0.7$. Lo que sugieren estas figuras es que la red intenta *imitar* la conectividad correspondiente a la de máxima capacidad. Cuando la conectividad de la red se encuentra por debajo de este rango, la forma de priorizar las conexiones es eligiendo a las de peso intermedio por encima de las de peso cercano a cero (es decir, optimiza la elección de sinapsis descartando las menos relevantes). En contraste, cuando la conectividad se encuentra por encima de este rango, la red tiene conexiones en exceso que distribuye sobre las sinapsis de peso más bajo para minimizar su contribución sobre el término de ruido.

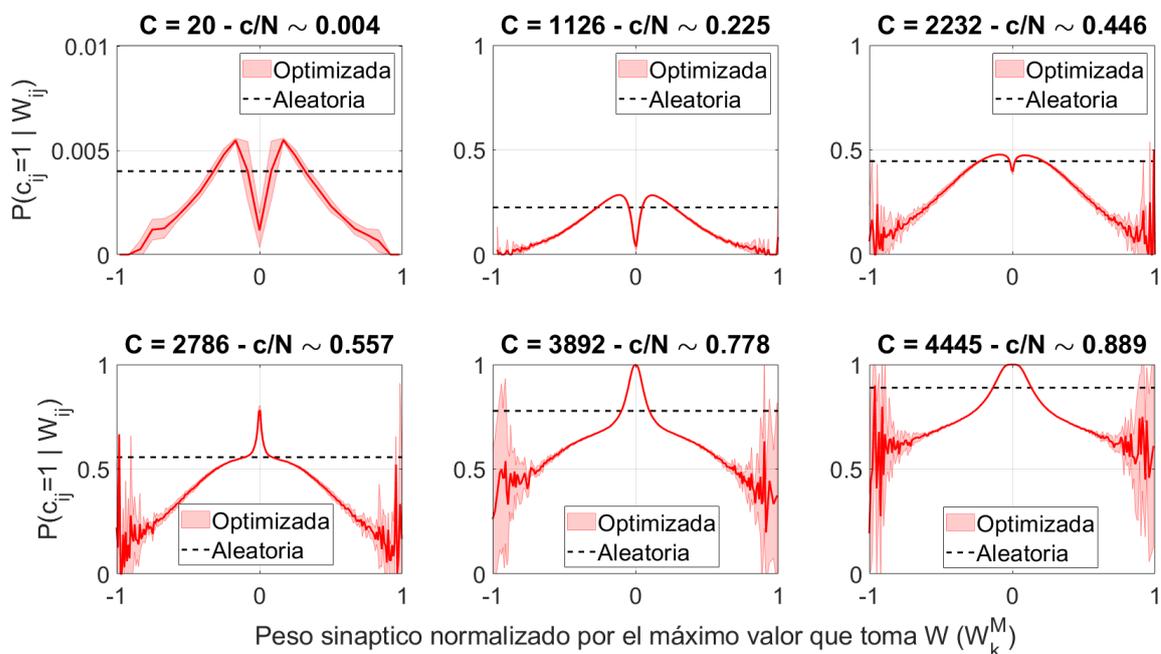


Figura 12: En las figuras se grafican las curvas de probabilidad $\mathcal{P}(c_{ij} = 1 | W_{ij} = 2k - p)$ en función de W_k^M (que son los pesos normalizados por el máximo elemento de W) para distintos valores de número de conexiones por neurona. Las curvas rojas indican las distribuciones correspondientes a redes de 5000 neuronas cuya conectividad fue optimizada mediante el algoritmo de enfriamiento simulado. Dichas curvas resultaron del promedio de 5 simulaciones por cada valor de c/N , y en rojo atenuado se indica la desviación estándar de dicho promedio. Se puede observar que la dispersión aumenta notablemente en los extremos de los gráficos. Esto se debe a que, a medida que el valor del peso sináptico aumenta, disminuye la frecuencia con la cual aparece en la matriz W (y por lo tanto, hacer estadística sobre valores de pesos altos conlleva a un aumento en el error). En negro punteado se indica el valor de la distribución correspondiente a una red de conectividad aleatoria, que es $\mathcal{P}(c_{ij} = 1 | W_{ij} = 2k - p) = c/N$.

3.4. Análisis de la Señal y el Ruido

¿A qué se debe la mejora en la capacidad de almacenamiento? ¿Cómo se modifica la distribución de ruido después de optimizar una red? Entender estas preguntas profundiza la comprensión

de los efectos que tiene el proceso de optimización sobre la dinámica de la red. El objetivo del algoritmo de enfriamiento simulado presentado es minimizar una función de costo E_i^0 . La solución ideal es llevar esta función al cero, si bien pocas veces es posible. Si la función de costo no se anula, entonces las neuronas reciben en promedio un aporte de ruido, que se espera que sea menor al del caso de una conectividad aleatoria. Para comprender los efectos que tiene el ruido sobre la red, conviene estudiarlo en función de su correlación con la señal. Se define al ruido-señal como el producto entre el ruido que recibe una neurona por el signo que la misma toma en la memoria que la red quiere recuperar

$$\xi_i^\mu R_i^\mu = \xi_i^\mu \sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu c_{ij} \quad (31)$$

Analizar los resultados en términos del ruido-señal permite discriminar al aporte del ruido que favorece a la señal ($\xi_i^\mu R_i^\mu > 0$) y al aporte del ruido que perjudica la recuperación del patrón ($\xi_i^\mu R_i^\mu < 0$). En la sección 1.6.1 de la introducción, se explicó que, en una red de conectividad aleatoria, la distribución del ruido-señal es igual a la del ruido, que puede aproximarse por una distribución gaussiana de media nula $\langle \xi_i^\mu R_i^\mu \rangle = 0$ y desviación estándar $\sigma_{\xi R} \sim \sqrt{p/c}$. Sin embargo, los resultados obtenidos para las redes optimizadas difieren de este comportamiento.

Se comparó la distribución de ruido por neurona en redes de conectividad aleatoria y optimizada para distintos valores de carga de patrones ($\alpha = p/c$). Se obtuvo una distribución simétrica en torno a la media $\langle \xi_i^\mu R_i^\mu \rangle$ para las redes de conectividad optimizada. Sin embargo, a diferencia de las redes de conectividad aleatoria, se observó que el valor medio de dicha distribución tiende a hacerse negativo al aumentar α . Este fue uno de los resultados más llamativos ya que significa que el proceso de optimización tiende a anticorrelacionar el ruido con la señal. En la figura 13.a se grafica la media de las distribuciones de ambos casos. Se puede observar que $\langle \xi_i^\mu R_i^\mu \rangle \rightarrow -1$ al incrementar α . De todos modos, el módulo de la media de la distribución en el caso optimizado no es significativamente mayor al de la media del caso aleatorio cerca del punto crítico (ver figura 13, paneles d.1 y d.2), que es el punto donde la fracción de patrones recuperados correctamente f cae abruptamente a 0 (fig. 13.c).

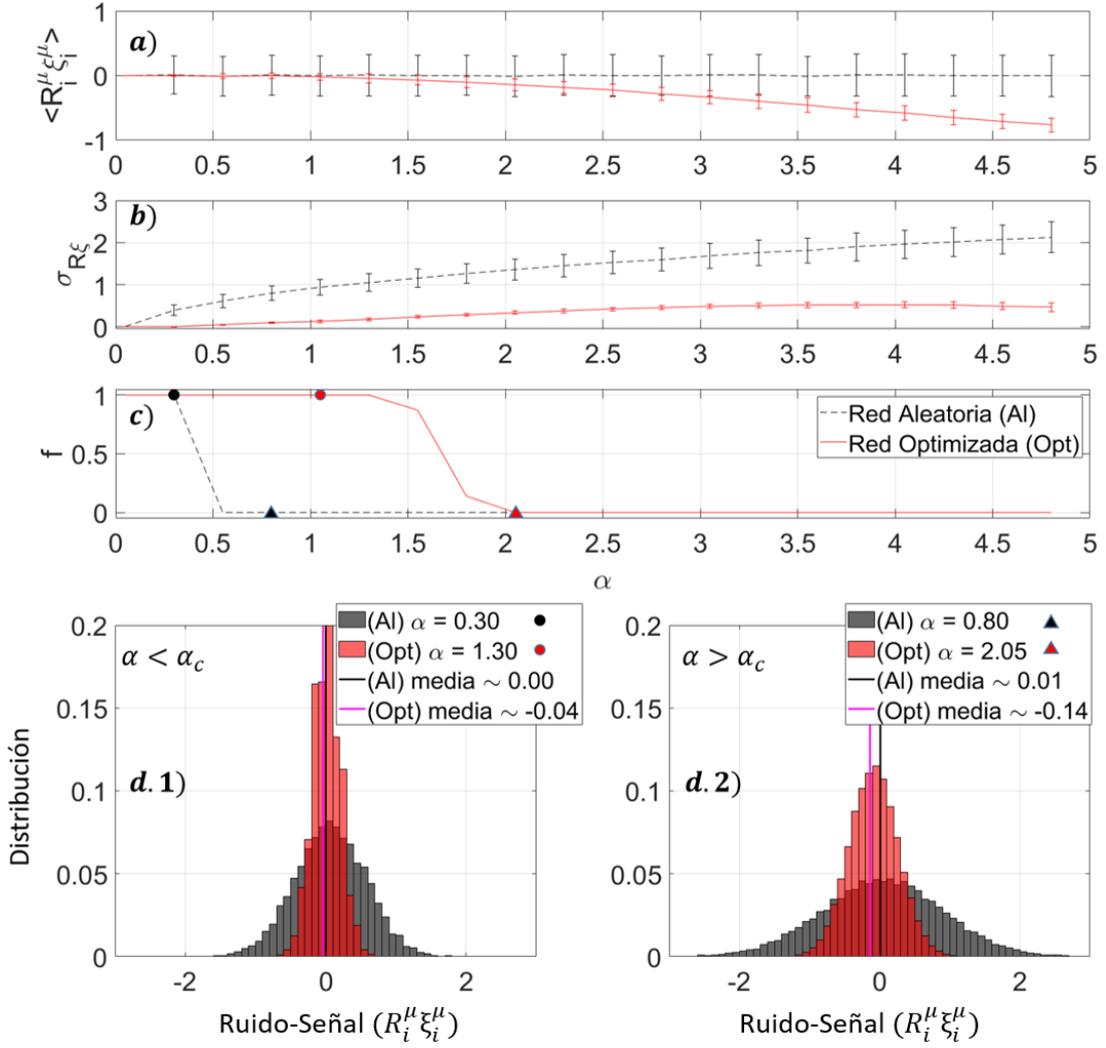


Figura 13: Comparación del ruido-señal entre red de conectividad aleatoria y optimizada mediante la minimización de E_i^0 . Se comparan los resultados obtenidos de simulaciones realizadas con redes de 2000 neuronas con 20 conexiones por neurona ($c/N = 0,01$). En rojo se representa la red optimizada mediante la minimización de la función E_i^0 y en negro, la red de conectividad aleatoria. Panel **a)**: se grafica el valor medio del término de ruido multiplicado por la señal ($\langle R_i^\mu \xi_i^\mu \rangle$) que recibe cada neurona en función del parámetro $\alpha = p/c$; las barras de error indican la desviación estándar del promedio de las medias. Panel **b)**: se grafica la desviación estándar del ruido-señal ($\sigma_{R\xi}$) que reciben las neuronas en función del parámetro α ; las barras de error indican la desviación estándar del promedio de las desviaciones estándar. Panel **c)**: se grafica la fracción de patrones (f) que la red puede recuperar de un total de p patrones almacenados en función de α . Panel **d.1)**: Se grafican las distribuciones del ruido-señal correspondientes a $\alpha = 0.30$ para una red de conectividad aleatoria y $\alpha = 1.30$ para una red de conectividad optimizada, junto con los valores de las medias de sus distribuciones. Estos dos puntos se encuentran marcados con un círculo negro y rojo, respectivamente, en el panel **c)**, con el objetivo de mostrar que ambas distribuciones corresponden a cargas de patrones cercanas pero previas al punto crítico. Panel **d.2)**: Se grafican las distribuciones del ruido señal correspondientes a $\alpha = 0.80$ para una red de conectividad aleatoria y $\alpha = 2.05$ para una red de conectividad optimizada, junto con los valores de las medias de sus distribuciones. Estos dos puntos se encuentran marcados con un triángulo negro y rojo, respectivamente, en el panel **c)**, con el objetivo de mostrar que ambas distribuciones corresponden a cargas de patrones cercanas pero posteriores al punto crítico.

Analizando la tasa de crecimiento de la desviación estándar, se observa que la correspondiente a la red optimizada es mucho menor a la de la red de conectividad aleatoria (ver figura 13.b). Este hecho explica la mejora obtenida en la capacidad de almacenamiento. Si bien el valor medio de la distribución de $R_i^\mu \xi_i^\mu$ es negativo, su ancho es lo suficientemente chico como para que las fluctuaciones del término de ruido no desestabilicen a los patrones en el límite clásico. Sin embargo, la desviación estándar correspondiente al punto crítico de la red optimizada es menor a la de la red aleatoria en su respectivo punto crítico. La anticorrelación entre el ruido y la señal podría estar vinculada con esta disminución en la tolerancia a las fluctuaciones del ruido.

Se puede concluir que el algoritmo de optimización permitió mejorar la *performance* de la red gracias a una disminución sustancial en la desviación estándar de la distribución del ruido-signal. Sin embargo, resulta intrigante que un proceso de optimización tendiente a disminuir el ruido lo lleve a anticorrelacionarse, en promedio, con la señal. Por razones que no se alcanzan a conocer en esta tesis, esto parece resultar beneficioso para la minimización de la función de costo.

3.5. Cuenca de Atracción

Un rasgo importante de las memorias autoasociativas, además de la capacidad de almacenamiento, es su robustez. Es importante que la red sea capaz de recuperar una memoria si el estado inicial de la misma representa al patrón corrompido. En esta sección se presentan los resultados del estudio de la cuenca de atracción de las redes optimizadas. Dada una red con una carga de patrones α , el método consistió en evaluar la estabilidad de cada uno de los patrones almacenados inicializando el estado de la red con un cierto porcentaje de error inicial. Es decir, dado un patrón ν a recuperar, se inicializó a la red con distintos valores de similaridad inicial $m^\nu(t_0)$ (con t_0 el tiempo inicial). La fracción del total de bits erróneos se la puede expresar como $[1 - m^\nu(t_0)]/2$ (por ejemplo, un porcentaje de un 50% de error inicial corresponde a un *overlap* inicial de $m^\nu(t_0) = 0$). Para cada patrón almacenado y porcentaje de error inicial, se dejó evolucionar la red por 100 iteraciones y se computó su similaridad final. En escala de grises se graficó la fracción de patrones del total almacenado que la red pudo recuperar exitosamente (es decir, que $m^\nu(t_f) > 0.7$, con t_f el tiempo final).

En el caso de redes de conectividad aleatoria, ya se comentó en la introducción que la cuenca de atracción depende de la conectividad. Para redes de conectividad diluida, es posible recuperar cualquier patrón almacenado hasta con un 50% de error inicial (siempre que $\alpha < \alpha_c$). En este caso, se observa un salto abrupto hacia la fase amnésica en el punto crítico (ver figura 4). A medida que la conectividad incrementa, sucede que el salto abrupto se *suaviza*, es decir que a partir de una determinada carga α , la tolerancia al error comienza a disminuir a medida que se cargan más patrones a la red, hasta llegar al punto crítico.

Se estudió la cuenca de atracción en las redes optimizadas (figura 21). Para conectividades bajas, a diferencia de las redes de conectividad aleatoria, la fuerza de las memorias tiende a disminuir de forma suave a medida que se aumenta el número de patrones almacenados. En el caso particular de $c/N = 0.01$ (red diluida) se puede ver que existe una primera región con

$0 \leq \alpha \leq 0.6$ donde se maximiza la estabilidad de las memorias, pudiendo éstas ser recuperadas con un error inicial entre un 40 % y 50 %. A partir de este punto, la cuenca de atracción comienza a disminuir significativamente hasta alcanzar el almacenamiento crítico. Lo que sucede entonces con redes diluidas optimizadas es que, si bien se alcanza un factor α elevado (entre 3 y 6 veces más grande que una red aleatoria, ver figura 10), los patrones almacenados son muy débiles cerca de este punto crítico, en el sentido de que sólo pueden recuperarse si el estado inicial tiene muy poco error. Esto último quiere decir que el *precio a pagar* por mejorar la capacidad de almacenamiento mediante la minimización de la función E_i^0 es reducir la robustez de las memorias.

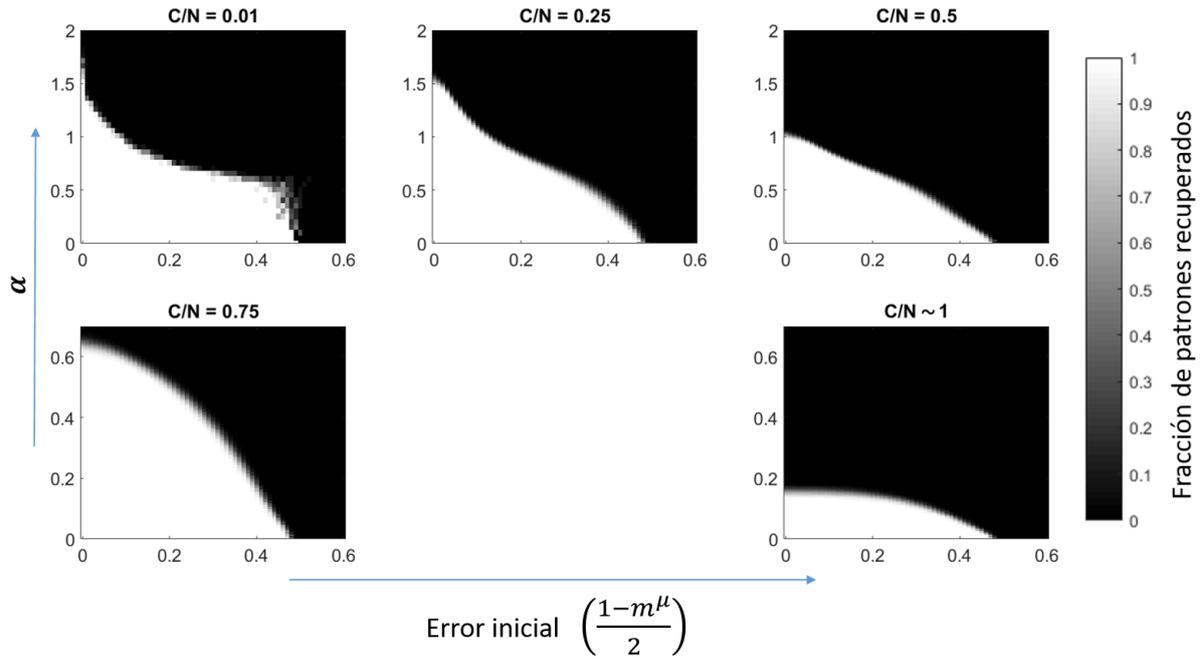


Figura 14: La figura ilustra el resultado de simulaciones en las que se estudió la fuerza de los patrones almacenados en una red de 2000 neuronas con 20, 500, 1000, 1500 y 1998 conexiones por neurona (en la figura se indica el factor c/N correspondiente). El eje vertical de la figura representa el factor de carga de la red ($\alpha = p/c$). El eje horizontal representa el error con el que se inicializa cada patrón almacenado en la red ($[1 - m^\mu(t_0)]/2$). En escala de grises se indica la fracción de los patrones almacenados que pudieron ser recuperados al inicializarse con el error correspondiente. Analizando todo el rango de conectividades, se observa un cambio en la concavidad de la curva que define el límite entre la región estable e inestable que depende de c/N . Asimismo, a medida que la conectividad va tendiendo a ser completa, se recupera el resultado conocido.

3.6. Breve Comentario acerca de la Función de Costo

Es importante tener en cuenta que la función de costo E_i^0 es una de las tantas funciones posibles que podrían ser utilizadas. Ya se mencionó que la suma $\sum_{\mu=1}^p |R_i^\mu|$ es una función admisible. De hecho, tanto esta suma como E_i^0 pertenecen a una misma familia de funciones admisibles para esta optimización

$$F_i^\beta = \sum_{\mu=1}^p |R_i^\mu|^\beta \quad (32)$$

donde $\beta > 0$ y $F_i^2 \equiv E_i^0$. Toda esta familia de funciones satisface que $F_i^\beta = 0 \rightarrow \sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu c_{ij} = 0$ para todo patrón μ . Si se optimiza una red de parámetros N , c y p para distintos valores de β se puede ver que los resultados de la optimización dependen de β . En la figura 15 se muestra un ejemplo. La conclusión de esta observación es que

- el rendimiento de este algoritmo de optimización depende de la función de costo a minimizar;
- los resultados de las optimizaciones obtenidas no representan la capacidad de almacenamiento máxima posible (si bien el límite podría estar cerca);
- podría existir otra familia de funciones que genere mejores resultados.

Sin embargo, la función utilizada E_i^0 cumple el objetivo principal de este trabajo, que es mostrar que es posible mejorar la capacidad de almacenamiento de la red.

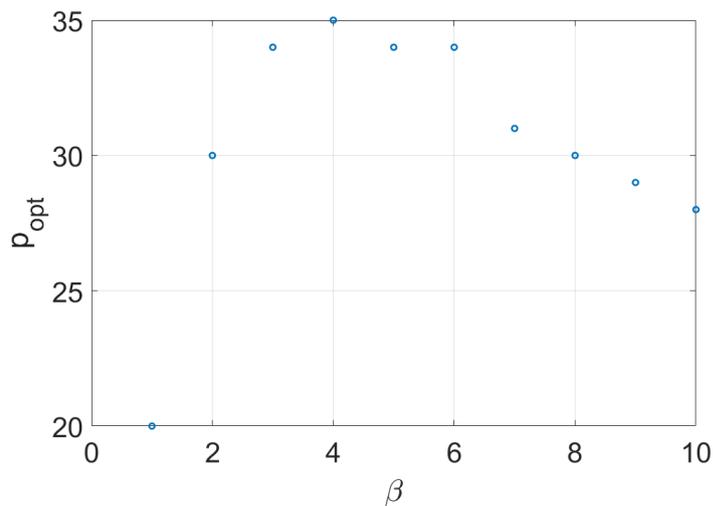


Figura 15: Se grafican las capacidades máximas (p_{opt}) obtenidas al optimizar la conectividad de una red de $N = 2000$ y $c = 20$ minimizando F_i^β , para distintos valores del parámetro β .

4. Modificación de la Función de Costo: Refuerzo de la Señal

Hasta el momento, la hipótesis de base fue que una red neuronal debería configurar su conectividad de forma tal que minimice el ruido para mejorar su capacidad de almacenamiento. Sin embargo, los resultados obtenidos indican que esta optimización anticorrelaciona la señal y el ruido (figura 13). Surgió entonces la pregunta de si sería posible utilizar una función de costo que, por el contrario, correlacionara la señal y el ruido, con el fin de obtener un incremento en la *performance* de la red incluso mayor al obtenido en la sección anterior. En otras palabras, la idea fue encontrar una configuración c_{ij} tal que para cada patrón μ suceda que

$$\text{sgn}(\xi_i^\mu) = \text{sgn}\left(\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu c_{ij}\right) \quad (33)$$

Puesto en términos de la función de costo, se propuso la siguiente función a minimizar mediante el algoritmo de enfriamiento simulado

$$E_i^\epsilon = \sum_{\mu=1}^p \left(\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu c_{ij} - \epsilon \xi_i^\mu \right)^2 \quad (34)$$

donde $\epsilon > 0$ es un parámetro que regula qué tan fuerte debe ser el refuerzo de la señal. Al minimizar esta función, se pide para cada patrón μ que $\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu c_{ij} \rightarrow \epsilon \xi_i^\mu$.

4.1. Optimización con $\epsilon = \alpha$

En las siguientes subsecciones se analizan los resultados obtenidos después de haber optimizado diversas redes neuronales (variando N , c y p) utilizando como función de costo E_i^ϵ (ec. 34). Se fijó al parámetro $\epsilon = \alpha$ ($= p/c$), ya que se vio que esta elección mejora los resultados de la optimización en la región de conectividad diluida (ver Apéndice A).

4.2. Capacidad de Almacenamiento

A partir de la minimización de la función de costo E_i^ϵ mediante el algoritmo de enfriamiento simulado, se obtuvo una mejora en la capacidad de almacenamiento de las redes de Hopfield. Al igual que en las optimizaciones con $\epsilon = 0$, los resultados obtenidos para redes de distinta dimensión colapsan sobre una misma curva siempre que se describa su dinámica en términos de α y c/N . En la optimización previa ($\epsilon = 0$), se mostró que redes de conectividad baja ($c/N \leq 0.5$) pueden aprovechar mejor el proceso de optimización pero su máximo rendimiento se alcanza

cerca de $c/N \sim 0.2$. Utilizando $\epsilon = \alpha$, se obtuvo que la fracción de mejora p_{opt}/p_0 tiene una tendencia monótonamente decreciente con el parámetro c/N , alcanzando su mejora máxima cerca de la conectividad más baja ($c/N < 0.1$). Para la región $c/N \leq 0.4$, se consiguió una mejora superior a la de la optimización tomando $\epsilon = 0$ (ver figura 16). Estos resultados sugieren que redes de conectividad diluida (como las del cerebro de los mamíferos) son las que más aprovecharían una optimización donde la conectividad permita que el término de ruido refuerce la señal.

Si bien la capacidad de almacenamiento lograda es superior para bajas conectividades, en la región de conectividades altas sucede lo inverso. Se puede ver que la optimización con $\epsilon = 0$ alcanza un máximo levemente superior en la región $0.4 < c/N < 0.7$, mientras que la optimización $\epsilon = \alpha$ alcanza su máximo de almacenamiento en la región $0.35 < c/N < 0.55$ (gráfico derecho de la figura 16).

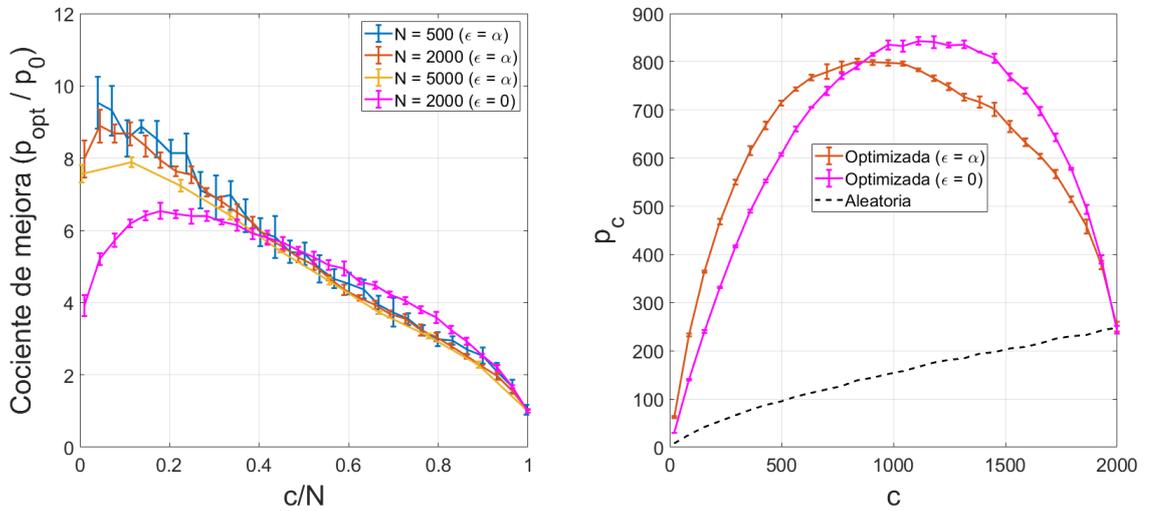


Figura 16: Mejora en la capacidad de almacenamiento de una red de Hopfield mediante la optimización de su matriz de conexiones a partir de la minimización de E_i^ϵ tomando $\epsilon = \alpha$ ($= p/c$). Izquierda: se grafica la fracción de mejora p_{opt}/p_0 en función del factor c/N para redes de 500, 2000 y 5000 neuronas. Asimismo, se grafica en magenta el resultado de una optimización tomando $\epsilon = 0$ para una red de 2000 neuronas. Derecha: se grafica el número máximo de patrones que puede almacenar una red optimizada de 2000 neuronas en función del número de conexiones por neurona. Se ilustran las curvas para el caso de una red de conectividad aleatoria (negro rayado), red de conectividad optimizada tomando $\epsilon = \alpha$ (naranja) y red de conectividad optimizada tomando $\epsilon = 0$ (magenta). Todos los gráficos se obtuvieron de promediar las capacidades de almacenamiento provenientes de 5 optimizaciones distintas para cada N . Los errores provienen de la desviación estándar de dicho promedio (en el gráfico derecho), y en el caso del gráfico izquierdo, la correspondiente propagación de error.

Por último, se comparan las curvas de α_c en función de c/N . Gran parte de la información contenida en este gráfico fue brindada en el previo análisis del cociente de mejora. Sin embargo, en la figura 17 se puede ver explícitamente los valores que adquiere el parámetro α_c para cada optimización. Particularmente, se puede notar que en la conectividad $c/N = 0.01$ correspondiente a $\epsilon = \alpha$, la capacidad crítica es $\alpha_c \sim 3.15$ mientras que el caso $\epsilon = 0$, la capacidad está cercana a la mitad, $\alpha_c \sim 1.49$.

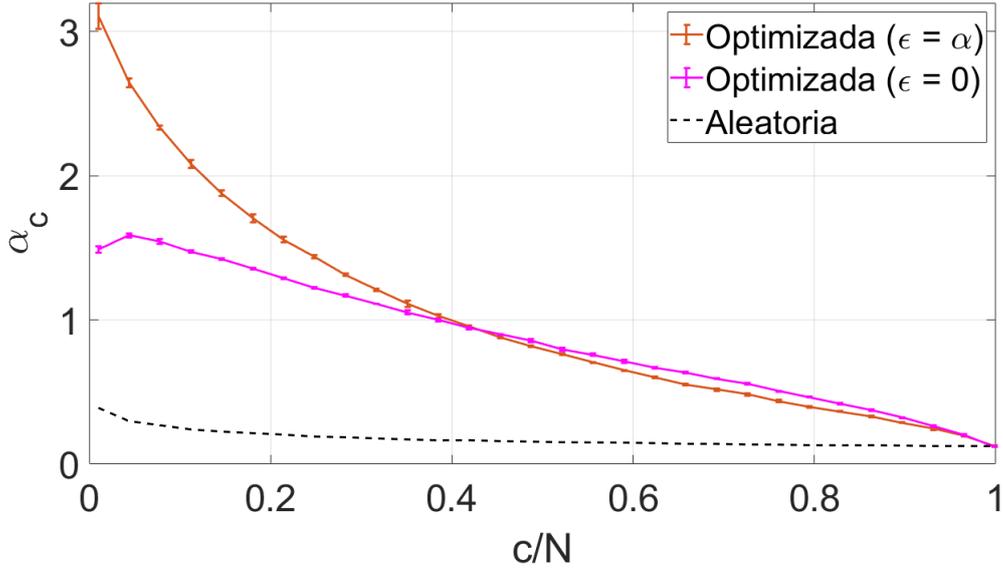


Figura 17: Se grafica la capacidad de almacenamiento crítica $\alpha_c = p_c/c$ en función del factor c/N . Se comparan los resultados correspondientes a redes con conectividad optimizada tomando $\epsilon = \alpha$ (naranja), redes con conectividad optimizada tomando $\epsilon = 0$ (magenta) y redes de conectividad aleatoria (negro rayado). Las simulaciones fueron hechas con redes de tamaño $N = 2000$ y los resultados provienen de un promedio de 5 simulaciones realizadas para cada N , indicando con las barras de error la desviación estándar correspondiente.

4.3. Estudio de la Relación entre los Pesos Sinápticos y la Conectividad

Se analizaron los efectos que tuvo la optimización sobre la elección de pesos sinápticos de la red, al igual que se hizo en la sección 3.3. Se obtuvo que la distribución de pesos en la matriz J es bimodal para el rango $c/N \leq 0.6$. Esto significa que, al igual que en el caso con $\epsilon = 0$, la optimización tiende a cortar pesos sinápticos de módulo muy bajo (figura 18). Por otro lado, en el rango de conectividad $c/N > 0.6$, nuevamente se observa un uso en exceso de las conexiones de módulo cercano a cero.

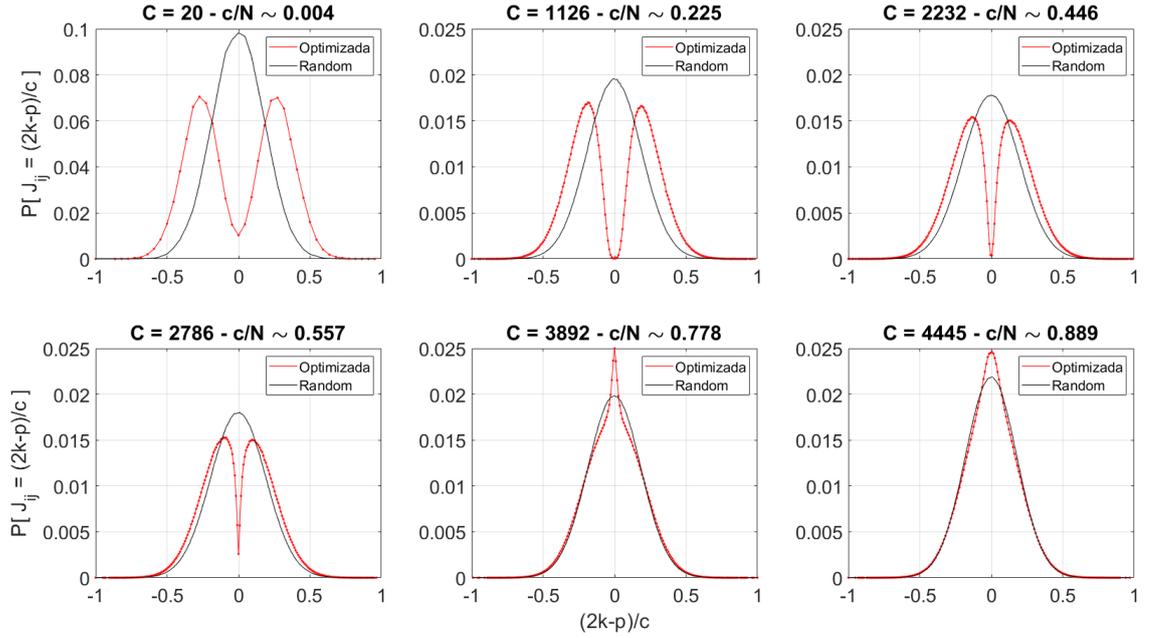


Figura 18: Se grafican las distribuciones de pesos $P[J_{ij} = (2k - p)/c]$ (con $k = 0, 1, \dots, p$) de la matriz de adyacencia J para redes de 5000 neuronas cuya conectividad fue optimizada a partir de la minimización de E_i^ϵ , con $\epsilon = \alpha$ (llevando su capacidad al límite crítico) para distinto número de conexiones por neurona, indicado con C . Las curvas negras se corresponden con redes de conectividad aleatoria (distribución binomial) y las rojas con redes de conectividad optimizada. Se observa que a medida que $c/N \rightarrow 1$, la distribución de la red optimizada tiende a la distribución binomial.

Como ya se explicó en la sección 3.3, la información que brindan las distribuciones de pesos debe ser complementada con la distribución de probabilidad de que exista una conexión entre dos neuronas dado un peso sináptico $\mathcal{P}(c_{ij} = 1 | W_{ij} = 2k - p)$ para obtener un panorama completo de los cambios en la matriz de adyacencia optimizada. Resulta sumamente interesante comparar las distribuciones mostradas en la figura 19 con aquellas mostradas en la figura 12. Se observa que, a diferencia de la optimización con $\epsilon = 0$, la red tiene una tendencia marcada a quedarse con los pesos más altos. Por otro lado, la probabilidad de que existan conexiones con peso bajo es prácticamente nula cuando c/N es bajo, y ésta aumenta a medida que incrementa c/N .

Se puede interpretar los resultados obtenidos de forma análoga a como se hizo con la optimización $\epsilon = 0$. Como se mostró en la figura 16 (gráfico derecho), la capacidad de almacenamiento máxima para una red de N neuronas (al optimizar su conectividad minimizando E_i^ϵ , con $\epsilon = \alpha$) se consigue cuando su conectividad se encuentra en el rango $0.35 < c/N < 0.55$. Al optimizar una red con una conectividad por debajo de este rango, el algoritmo tiende a *imitar* a la configuración de conexiones correspondiente a la máxima capacidad, priorizando las sinapsis de módulo más alto, ya que son las más relevantes en la distribución $\mathcal{P}(c_{ij} = 1 | W_{ij} = 2k - p)$. Por otro lado, si la conectividad se encuentra por encima de este rango, el algoritmo tiende a utilizar las conexiones *en exceso* en los pesos sinápticos de menor módulo, ya que son los que menos contribuyen en el campo de actividad de las neuronas.

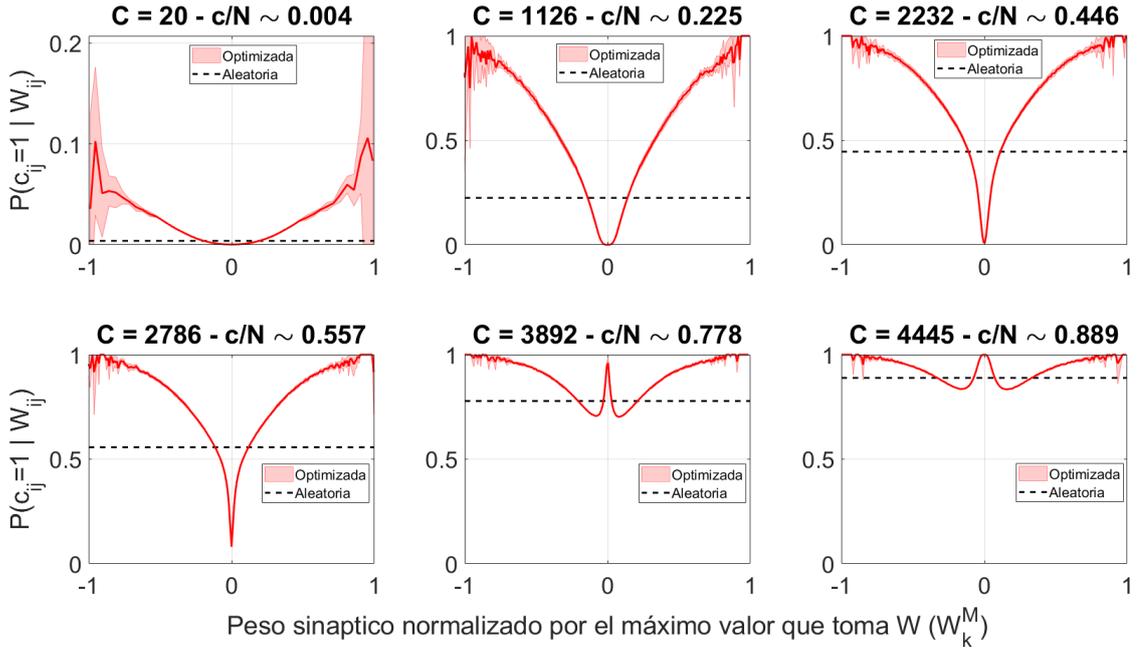


Figura 19: En las figuras se grafican las curvas de probabilidad $\mathcal{P}(c_{ij} = 1 | W_{ij} = 2k - p)$ en función de W_k^M (que son los pesos normalizados por el máximo elemento de W) para distintos valores de número de conexiones por neurona. Las curvas rojas indican los casos correspondientes a redes de 5000 neuronas cuya conectividad fue optimizada mediante el algoritmo de enfriamiento simulado (minimizando E_i^ϵ con $\epsilon = \alpha$). Dichas curvas resultaron del promedio de 5 simulaciones por cada valor de c/N , y en rojo atenuado se indica la desviación estándar de dicho promedio. En negro punteado se indica el valor de la distribución correspondiente a una red de conectividad aleatoria, que es $\mathcal{P}(c_{ij} = 1 | W_{ij} = 2k - p) = c/N$.

La distribución condicional $P(c_{ij} = 1 | W_{ij} = 2k - p)$ obtenida en el rango de baja conectividad posee una tendencia monótonamente creciente con el módulo del peso sináptico, y como ya se mencionó, a medida que disminuye la conectividad, mayor es la mejora sobre la capacidad de almacenamiento. Esto es coherente con el resultado obtenido por Montemurro y Tamarit [18], quienes mostraron que mediante una regla de poda de conexiones en la que la red conserve sólo las c conexiones más fuertes por neurona, y respetando la condición de red diluida $c \ll N$, la capacidad mejora sustancialmente (ver introducción, sección 1.6.3). Una explicación posible es que la red busca *consenso* entre neuronas. Cada elemento de matriz de W es proporcional a la diferencia entre el número de patrones en que ambas neuronas se activan juntas y el número de patrones donde ambas neuronas poseen actividad contraria. Luego, un peso sináptico de módulo alto implica una correlación (o anticorrelación) fuerte entre la actividad de dos neuronas al recuperar cualquier patrón arbitrario. En un caso así, la actividad de una casi siempre contribuirá positivamente (o negativamente) sobre el campo de actividad de la otra, sin importar qué patrón se esté recuperando, por lo que se puede decir que hay *consenso* en su actividad (las neuronas ya ‘saben’ cuál es el signo de la contribución más probable de su vecina antes de recuperar cualquier patrón). En otras palabras, tomando un patrón μ al azar, la probabilidad de que $W_{ij} \xi_i^\mu \xi_j^\mu > 0$ es alta y por lo tanto, en promedio el campo-sígnal es positivo.

4.4. Análisis de la Señal y el Ruido

En esta optimización, la mejora en la capacidad de almacenamiento se debe a que el ruido tiende a correlacionarse con la señal. Como fue de esperar, al graficar el ruido-señal en función de los patrones cargados a la red, la media $\langle \xi_i^\mu R_i^\mu \rangle$ es siempre positiva y crece monótonamente con el parámetro α en el caso de la red optimizada (ver figura 20.a). Se verifica entonces que la función de costo propuesta cumple el objetivo deseado. Este resultado es compatible con el análisis de la distribución de pesos sinápticos dada en la subsección anterior, ya que verifica el hecho de que los pesos sinápticos de módulo alto permiten que, en promedio, el ruido presente en el campo de actividad de cada neurona refuerce la señal del patrón a recuperar.

Asimismo, este refuerzo en la señal hace más resiliente a la red frente a fluctuaciones aleatorias del ruido. Por un lado, la desviación estándar del ruido-señal ($\sigma_{R\xi}$) crece monótonamente con α (fig. 20.b), al igual que $\langle \xi_i^\mu R_i^\mu \rangle$. Sin embargo, el valor que toma $\sigma_{R\xi}$ en el punto crítico es mayor al que toma en la red de conectividad aleatoria en su respectivo punto crítico (fig. 20.d). Esto quiere decir que hacen falta más fluctuaciones para desestabilizar a la red optimizada.

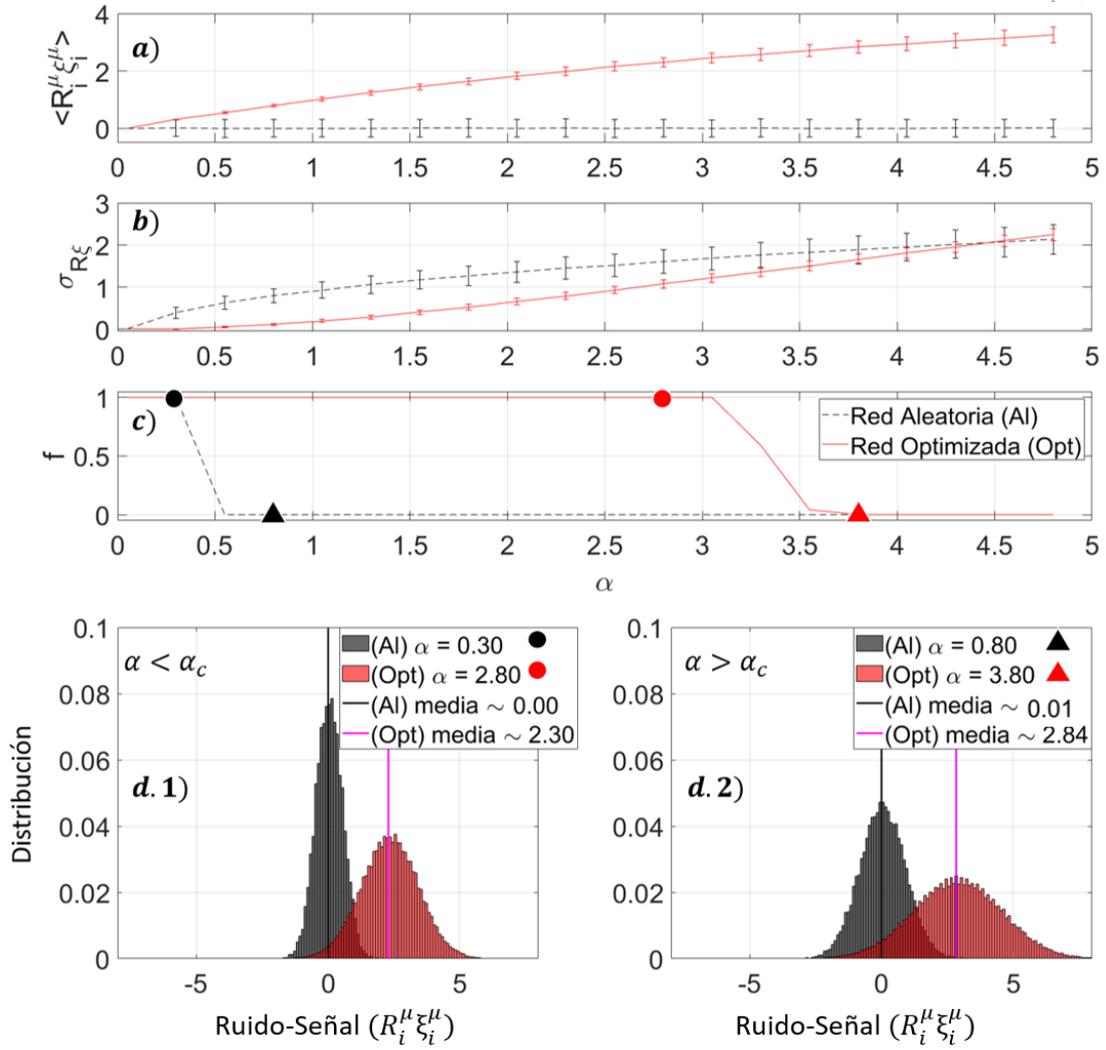


Figura 20: Comparación del ruido-señal entre red de conectividad aleatoria y optimizada mediante la minimización de E_i^ϵ tomando $\epsilon = \alpha$ ($= p/c$). Se comparan los resultados obtenidos de simulaciones realizadas con redes de 2000 neuronas con 20 conexiones por neurona ($c/N = 0.01$). En rojo se representa la red optimizada mediante la minimización de la función E_i^ϵ ($\epsilon = \alpha$) y en negro, la red de conectividad aleatoria. Panel **a)**: se grafica el valor medio del término de ruido multiplicado por la señal ($\langle R_i^\mu \xi_i^\mu \rangle$) que recibe cada neurona en función del parámetro α ; las barras de error indican la desviación estándar del promedio de las medias. Panel **b)**: se grafica la desviación estándar del ruido-señal ($\sigma_{R\xi}$) que reciben las neuronas en función del parámetro α ; las barras de error indican la desviación estándar del promedio de las desviaciones estándar. Panel **c)**: se grafica la fracción de patrones (f) que la red puede recuperar de un total de p patrones almacenados en función de α . Panel **d.1)**: Se grafican las distribuciones del ruido-señal correspondientes a $\alpha = 0.30$ para una red de conectividad aleatoria y $\alpha = 2.80$ para una red de conectividad optimizada, junto con los valores de las medias de sus distribuciones. Estos dos puntos se encuentran marcados con un círculo negro y rojo, respectivamente, en el panel **c)**, con el objetivo de mostrar que ambas distribuciones corresponden a cargas de patrones cercanas pero previas al punto crítico. Panel **d.2)**: Se grafican las distribuciones del ruido señal correspondientes a $\alpha = 0.80$ para una red de conectividad aleatoria y $\alpha = 3.80$ para una red de conectividad optimizada, junto con los valores de las medias de sus distribuciones. Estos dos puntos se encuentran marcados con un triángulo negro y rojo, respectivamente, en el panel **c)**, con el objetivo de mostrar que ambas distribuciones corresponden a cargas de patrones cercanas pero posteriores al punto crítico.

4.5. Cuenca de Atracción

Se estudiaron las cuencas de atracción de los patrones almacenados en las redes con conectividad optimizada mediante la minimización de E_i^c ($\epsilon = \alpha$). Por un lado, vuelve a suceder que la fuerza de los atractores disminuye de forma relativamente continua en el rango de conectividad baja (a diferencia de la red de conectividad aleatoria). Sin embargo, la primera diferencia con la optimización $\epsilon = 0$ es que las memorias en el rango de conectividad $c/N < 0.4$ son mucho más robustas. Este hecho es notable ya que uno de los problemas con la optimización $\epsilon = 0$ es que el incremento en la capacidad de almacenamiento conlleva a una reducción considerable de la tolerancia de la red a tener errores en el estado inicial, lo cuál es biológicamente implausible. Por otro lado, se observa que a medida que $c/N \rightarrow 1$ se recupera el resultado conocido, al igual que en la otra optimización.

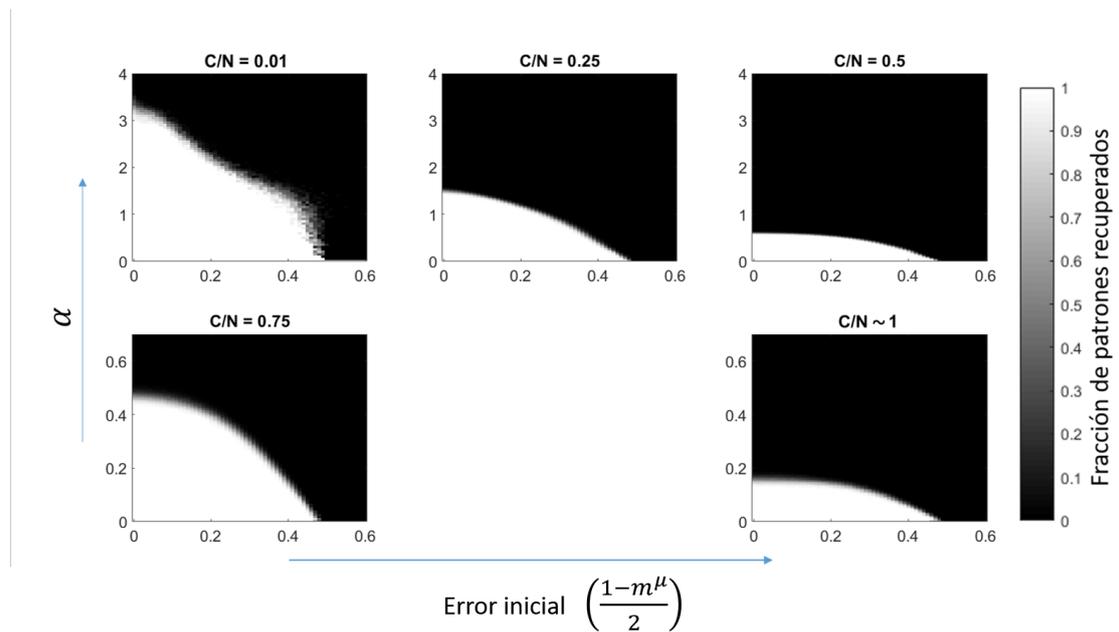


Figura 21: La figura ilustra el resultado de simulaciones en las que se estudió la cuenca de atracción de los patrones almacenados en una red de 2000 neuronas con 20, 500, 1000, 1500 y 1998 conexiones por neurona (en la figura se indica el factor c/N correspondiente). El eje vertical de la figura representa el parámetro de carga de la red (α). El eje horizontal representa el error con el que se inicializa cada patrón almacenado en la red ($[1 - m^\mu(t_0)]/2$). En escala de grises se indica la fracción de los patrones almacenados que pudieron ser recuperados al inicializarse con el error correspondiente.

5. Introducción de Elementos Biológicamente Plausibles

La idea de esta sección es investigar si la introducción de características biológicas en el modelo de redes autoasociativas puede tener efectos positivos al momento de optimizar la conectividad utilizando el enfriamiento simulado. Particularmente, una de las modificaciones consiste en relajar la restricción $c_{ij} \in \{0, 1\}$, ya que experimentalmente se observa la existencia de múltiples conexiones entre neuronas [14, 15]. Por otro lado, una segunda modificación tiene que ver con la introducción de patrones de baja actividad media en redes de actividad asimétrica ($s_i = 0$ ó $s_i = 1$).

5.1. Múltiples Conexiones entre Pares de Neuronas

Se volvieron a optimizar redes de Hopfield mediante el algoritmo de enfriamiento simulado, relajando las restricciones sobre la matriz de conectividad. Se propuso que $c_{ij} \in \{0, 1, \dots, n\}$, para distintos valores de $n \in \mathbb{N}$. Para ello, se realizó una leve modificación en el algoritmo, generalizando las permutaciones entre estados de conectividad por intercambios de conexiones. Es decir, dados dos estados de conectividad aleatorios c_{ik} y c_{im} (siempre siendo $k \neq m \neq i$), entonces

1. si $c_{ik} = c_{im} = n$ ó $c_{ik} = c_{im} = 0$ entonces no se realiza operación;
2. si $c_{ik} = 0$ y $0 < c_{im} < n$ entonces se modifican de la siguiente forma $c_{ik} = 1$ y $c_{im} = c_{im} - 1$;
3. si $c_{ik} = n$ y $0 < c_{im} < n$ entonces se modifican de la siguiente forma $c_{ik} = n - 1$ y $c_{im} = c_{im} + 1$;
4. si $0 < \{c_{ik}, c_{im}\} < n$ entonces se elije de forma aleatoria la opción $+$ ó $-$:
 $c_{ik} = c_{ik} \pm 1$ y $c_{im} = c_{im} \mp 1$.

En la figura 22 se muestran los resultados obtenidos para distintos valores de n , optimizando la conectividad mediante la minimización de E_i^0 , tomando $\epsilon = 0$ (los resultados obtenidos para $\epsilon > 0$ fueron análogos). Lo primero que se destaca es que en el rango de conectividad baja ($c < N/2$) no existe una mejora en la capacidad de almacenamiento con respecto a la optimización con la restricción original $c_{ij} \in \{0, 1\}$. Por otro lado, para el rango $c > N/2$ se puede ver una mejora abrupta en la capacidad de almacenamiento a medida que $c \rightarrow N$. Esto se puede explicar a partir del hecho de que relajar la restricción $c_{ij} \in \{0, 1\}$ aumenta enormemente la cantidad de configuraciones de conectividad posibles, haciendo que en el límite $c = N$, los grados de libertad no sean cero. Más aún, se observa que la capacidad de almacenamiento no solo mejora en este límite sino que crece monótonamente con c .

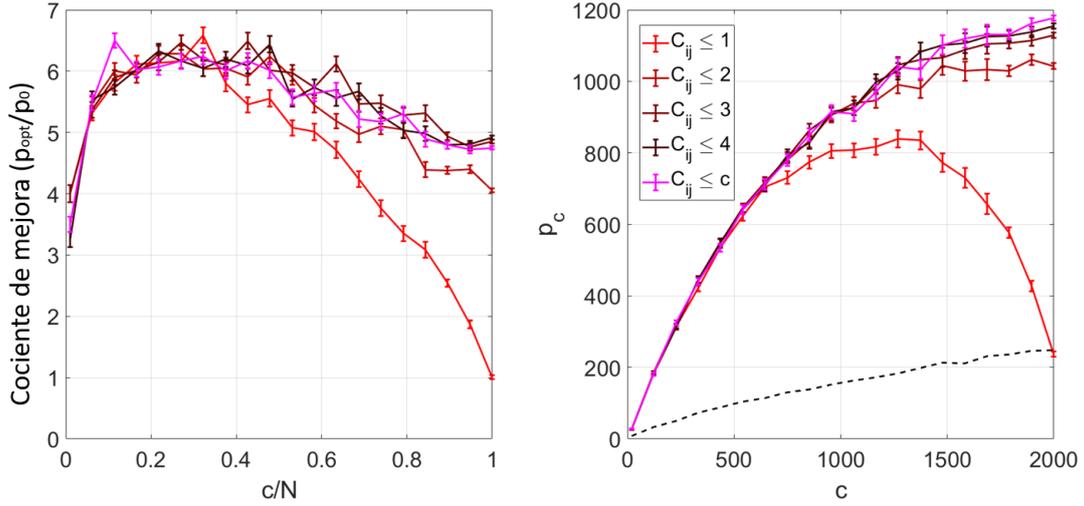


Figura 22: Mejora en la capacidad de almacenamiento de una red de Hopfield mediante la optimización de su matriz de conexiones a partir de la minimización de E_i^0 , relajando la restricción $c_{ij} \in \{0, 1\}$. Se muestran los resultados de optimizaciones a partir de la minimización de E_i^0 ($\epsilon = 0$) imponiendo la relajación $c_{ij} \in \{0, 1, \dots, n\}$ para distintos valores de n (en la leyenda se lo indica como $c_{ij} \leq n$ ya que n es el máximo valor posible que puede tomar cada elemento de la matriz de conexiones). El caso $c_{ij} \leq c$ corresponde a la máxima relajación posible para una red de c conexiones por neurona. Izquierda: se grafica la fracción de mejora (p_{opt}/p_0) en función de c/N . Gráfico derecho: Se grafica la capacidad de almacenamiento crítica en función del número de conexiones por neurona y en negro rayado se grafica la capacidad de almacenamiento de una red de conectividad aleatoria con $c_{ij} \in \{0, 1\}$. En el gráfico derecho, las barras de error indican la desviación estándar (provenientes de promediar los resultados en 3 simulaciones para cada punto) y en el izquierdo las barras de error provienen de la propagación correspondiente de dicha desviación.

Otro hecho destacable es que a medida que se relaja más la restricción (es decir que aumenta n , el valor máximo posible de conexiones entre dos neuronas), la capacidad de almacenamiento parece converger a una misma curva. De hecho, las curvas correspondientes a $c_{ij} \in \{0, 1, 2, 3, 4\}$ y $c_{ij} \in \{0, 1, \dots, c\}$ son muy similares, siendo la última la máxima relajación posible sobre una red de c conexiones por neurona. Esto puede explicarse a partir de la distribución de número de conexiones entre pares de neuronas que se muestra en la figura 23. Para la restricción $c_{ij} \in \{0, 1, \dots, c\}$, se obtuvo una distribución similar a una ley de potencias, lo que implica que números de conexiones grandes entre pares de neuronas se hacen muy improbables. Por lo tanto, aumentar mucho n lleva a tener contribuciones despreciables a partir de algún $n_0 < n$. De más está decir que esta distribución se encuentra muy alejada de lo observado experimentalmente.

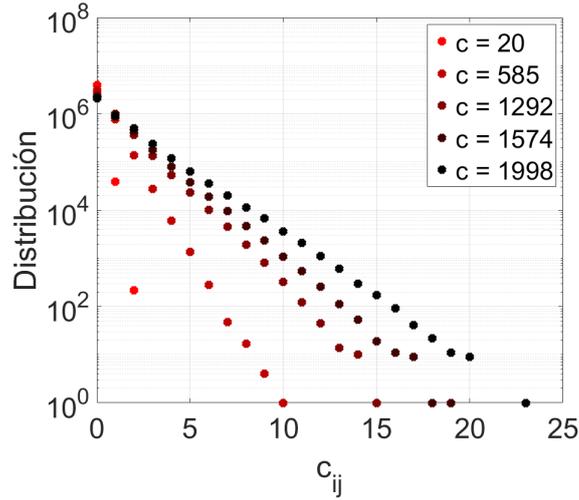


Figura 23: Distribución de número de conexiones entre par de neuronas (c_{ij}) para redes de distinta conectividad, utilizando la restricción $c_{ij} \in \{0, 1, \dots, c\}$.

5.2. Reducción de la Actividad Media de la Red

Como se explicó en la introducción, sección 1.8, es posible modificar el modelo clásico de Hopfield de forma tal de mejorar su plausibilidad biológica introduciendo patrones de actividad donde el número de neuronas activas sea menor a un 50% de la red. En tal caso, la regla de aprendizaje Hebbiano se debe reescribir como

$$W_{ij} = \frac{1}{ca(1-a)} \sum_{\mu=1}^p (\eta_i^{\mu} - a) (\eta_j^{\mu} - a); \quad \eta_i^{\mu} = \begin{cases} 1 \\ 0 \end{cases} \quad (35)$$

donde a es la fracción de neuronas activas en cada memoria, $\bar{\eta}^{\mu}$ es la nueva variable con la que se identifican los patrones almacenados y ahora los estados posibles de las neuronas son $s_i \in \{0, 1\}$. También cabe destacar que, al introducir esta modificación en el modelo, es necesario modificar la regla de recurrencia introduciendo un umbral de actividad U . En esta sección se trabajó siempre tomando el valor óptimo del umbral $U = 1/2 - a$,

$$s_i(t + \Delta t) = \frac{1}{2} [1 + \text{sgn}(h_i(t) - U)] \quad (36)$$

Al momento de optimizar la conectividad de la red en este modelo modificado, fue necesario revisar la formulación de la función de costo E_i^c , ya que pedir que el ruido tenga el mismo signo que la señal es pedirle que valga 0 ó 1. Para los patrones donde el ruido de una neurona deba llevarse a cero, la optimización tendería a llevarlo a algún valor en un entorno más o menos cercano al cero. En consecuencia, algunas de las contribuciones del ruido resultarían en contra de la señal, y otras, a favor. Un pedido más fuerte para evitar que el ruido vaya en contra de la se-

ñal nula es pedirle que tenga signo negativo. De esta forma, se reformuló E_i^ϵ de la siguiente manera

$$E_i^\epsilon = \sum_{\mu=1}^p \left[\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu c_{ij} - 2\epsilon \left(\eta_i^\mu - \frac{1}{2} \right) \right]^2 \quad (37)$$

Así, cuando $\eta_i^\mu = 1$, la minimización de la función lleva el ruido a ϵ , y cuando $\eta_i^\mu = 0$ lo lleva a $-\epsilon$.

5.3. Resultados de la Optimización

Se muestran los resultados obtenidos para las optimizaciones utilizando el parámetro $\epsilon = 0$. Se obtuvieron resultados análogos (que no se muestran) para optimizaciones utilizando un parámetro $\epsilon > 0$. La primera consecuencia interesante de esta modificación es que la fracción de mejora (p_{opt}/p_0) aumenta con respecto a la del modelo original. Tomando $a = 0.5$, se esperaría recuperar el resultado ya obtenido en las secciones anteriores. Sin embargo, en la figura 24 se aprecia que el cociente de mejora, si bien tiene la misma forma funcional con c/N , alcanza el valor máximo de p_{opt}/p_0 cercano a 9 (cuando en el modelo clásico se alcanzó una mejora cercana a 7). Sin embargo, otro resultado llamativo es que la eficacia de la optimización empeora a medida que se reduce la fracción a .

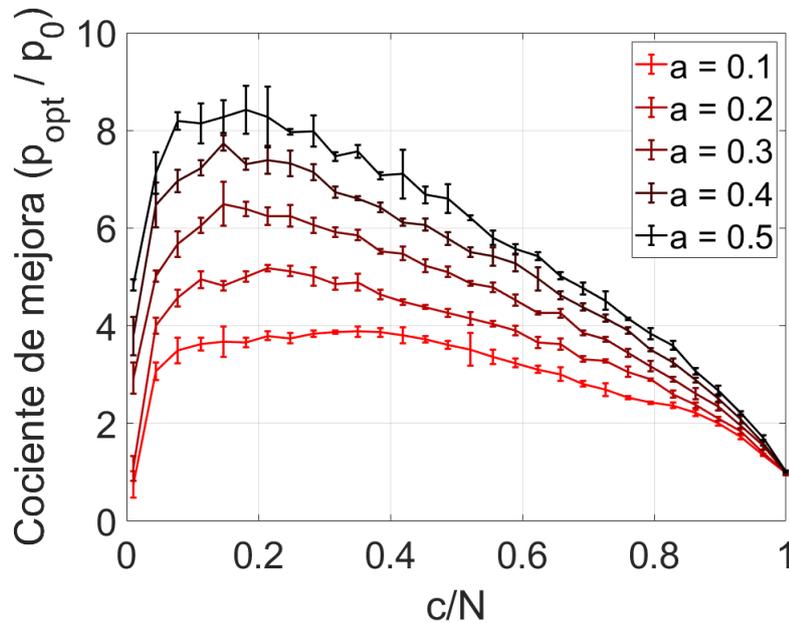


Figura 24: Se grafica la fracción de mejora (p_{opt}/p_0) en función de c/N de las optimizaciones de conectividad para distintos valores de actividad media a . Los datos fueron obtenidos de optimizar redes de 2000 neuronas utilizando el modelo modificado de Hopfield (donde la actividad neuronal toma los valores $s_i \in \{0, 1\}$ y el número de neuronas activas en cada patrón es Na) mediante la minimización de E_i^0 ($\epsilon = 0$), ver ec. 37. Los puntos del gráfico vienen de promediar los resultados de 4 simulaciones y las barras de error provienen de la propagación de la desviación estándar de la capacidad de almacenamiento óptima y aleatoria obtenidas de las simulaciones.

6. Algoritmo Online

Hasta el momento, se demostró que es posible mejorar la capacidad de almacenamiento de redes autoasociativas optimizando su conectividad. Sin embargo, el método de optimización utilizado carece de plausibilidad biológica. En las secciones anteriores se buscó la capacidad máxima mediante una búsqueda binaria, y en cada paso de la búsqueda se sometió a cada neurona a un programa de enfriamiento. Sin embargo, la red debería ser capaz de ir almacenando de forma progresiva la información, modificando las sinapsis neuronales de forma simultánea (una optimización en paralelo, o bien *online*). Por otro lado, ya se habló de cómo el ser humano produce durante los primeros años de desarrollo muchas más conexiones de las que utiliza en su etapa adulta. A lo largo de su crecimiento, el cerebro genera y poda conexiones de forma simultánea [7]. Este hecho motivó a construir un algoritmo de optimización que siguiera una dinámica de generación y eliminación de sinapsis (y no de permutaciones implausibles). Por último, cabe destacar que el enfriamiento simulado parte de un principio de funcionamiento básico que es la minimización de una función de costo. Las funciones utilizadas hasta el momento sólo contienen información local de la neurona, es decir, información contenida en el campo de actividad que recibe la neurona al querer recuperar un patrón. Por lo tanto, este fue un buen punto de partida, pero se requirieron modificaciones.

En esta sección, se explica cómo se dio lugar a un algoritmo que incorpora los elementos mencionados, capaz de mejorar la capacidad de almacenamiento de la red. Si bien este algoritmo no pretende explicar datos experimentales de forma cuantitativa, la idea es mostrar cómo a partir de la minimización de una función de costo con información local se obtiene un proceso de generación y eliminación de conexiones que optimiza la conectividad. Las simulaciones fueron realizadas sobre redes de 500 neuronas, dado el alto costo computacional que implicó el algoritmo, siguiendo el modelo clásico de Hopfield, es decir, $s_i \in \{-1, 1\}$, $c_{ij} \in \{0, 1\}$, la actividad media de los patrones nula ($N/2$ neuronas activas y $N/2$ neuronas inactivas en cada patrón $\bar{\xi}^\mu$) y umbral de actividad $U = 0$.

6.1. Descripción del Algoritmo

Se parte de una red con conectividad aleatoria de N neuronas y c_{init} conexiones presinápticas por neurona. Se carga un número inicial de patrones p_{init} de forma tal que la red sea capaz de recuperarlos. Dadas estas condiciones iniciales, se comienza a cargar patrones a la red de a pasos Δp . Lo que se quiere es que en cada paso, la red optimice su conectividad de forma global hasta poder estabilizar la mayoría de los patrones.

Para optimizar la conectividad, el algoritmo debe recorrer todas las neuronas. La dinámica que sigue es análoga a un mecanismo de prueba y error, donde se se generan cambios aleatorios en la conectividad pero se conservan sólo aquellos que reduzcan el valor de la función de costo. Dada una neurona i , se selecciona aleatoriamente alguna neurona presináptica j (al igual que en el enfriamiento simulado). Si existe conexión, entonces el algoritmo la corta, sólo si al hacerlo, la función de costo a minimizar disminuye (en caso contrario, no hace nada). De forma análoga, si no existe conexión, entonces el algoritmo genera una, sólo si al hacerlo, la función de costo

disminuye. Se realizan muy pocas iteraciones de este proceso por cada neurona para simular una optimización en línea (dados los recursos computacionales disponibles en el laboratorio, no fue posible correr el algoritmo de forma paralela en las $N = 500$ neuronas). Si la red es capaz de recuperar un 90% de los patrones almacenados, entonces se cargan Δp patrones más (en este trabajo se usó $\Delta p = 10$), y se vuelve a realizar el mismo recorrido sobre cada neurona. De otra forma, el recorrido se repite hasta que un 90% de los patrones sean estables.

Al implementar este algoritmo se observó que la conectividad media de la red tiende a estabilizarse. Por lo tanto, se determinó un límite de iteraciones para finalizar el proceso de optimización: el algoritmo frena si la conectividad media no varía por más de 50 iteraciones.

El algoritmo propuesto es muy similar al del enfriamiento simulado. Sin embargo, existen algunas grandes diferencias:

1. Desaparece el concepto de enfriamiento (ya no hay un programa que descienda una temperatura que defina una probabilidad de transición).
2. Se abandona la idea de permutación y se incorpora la noción de generación y eliminación de conexiones. Ya no se impone la restricción $\sum_{\substack{j=1 \\ j \neq i}}^N c_{ij} = c$.
3. Ya no se realiza una búsqueda binaria, sino que se cargan patrones de forma progresiva, y la red busca recuperarlos en cada paso (simulando una situación *más biológica* donde la red incorpora información en el tiempo).
4. El algoritmo no busca que cada neurona alcance un mínimo de energía, sino que el proceso de minimización se de en paralelo hasta que los patrones puedan ser recuperados.

Algoritmo 1: Algoritmo Online**Inicializar:**

Generar red de N neuronas con matriz de conectividad inicial C_{ij} tal que

$$\sum_{\substack{j=1 \\ j \neq i}}^N C_{ij} = c_{init} \quad \forall i/1 \leq i \leq N$$

Almacenar $p = p_{init}$ patrones

Calcular f fracción de patrones que se pueden recuperar

Iterar:

mientras $f > 0.9$ **hacer**

$$p = p + \Delta p$$

Cargar patrones a la red

Calcular f

mientras $f < 0.9$ **hacer**

(si no se recupera un 90 % del total, entonces se itera sobre las N neuronas)

para $i=1:N$ **hacer**

\bar{C}_i (vector de conexiones presinápticas de la neurona i con las $N-1$ restantes)

Calcular F_i^0 (función de costo para neurona i con conectividad \bar{C}_i)

para $k=1:Iteraciones$ **hacer**

$$\bar{C}^f = \bar{C}_i$$

Elegir una neurona j al azar tal que $j \neq i$

si $C_j^f == 0$ **entonces**

| $C_j^f = 1$ (generar conexión)

en otro caso

| $C_j^f = 0$ (cortar conexión)

fin

Calcular F_i^f (función de costo para neurona i con conectividad \bar{C}^f)

si $F_i^f - F_i^0 < 0$ **entonces**

| *(si la función de costo decrece, se acepta el cambio)*

$$F_i^0 = F_i^f$$

$$\bar{C}_i = \bar{C}^f$$

fin

fin

fin

Calcular f

si $\left\langle \sum_{\substack{j=1 \\ j \neq i}}^N C_{ij} \right\rangle$ no cambia por 50 iteraciones **entonces**

| **romper iteración**

fin

fin

fin

6.2. Función de Costo

Además de que el algoritmo esté estructurado de forma tal que pueda representar un proceso de optimización biológico, también es importante definir una función de costo a minimizar que tenga sentido biológico. Una de las características más importantes es que la función posea únicamente información local de la neurona, lo cual se satisface en la función E_i^ϵ (ec. 34). Al utilizar esta función en el algoritmo online, la dinámica de optimización cambia notablemente si se la compara con el algoritmo de enfriamiento simulado, ya que se elimina la restricción de conectividad constante ($\sum_{\substack{j=1 \\ j \neq i}}^N c_{ij} = c$). Para este nuevo algoritmo, la constante de normalización $1/c$ pasa a ser una variable que depende de la conectividad c_{ij} de cada neurona en cada iteración

$$E_i^\epsilon = \sum_{\mu=1}^p \left(\frac{\sum_{\substack{j=1 \\ j \neq i}}^N \tilde{\Omega}_{ij}^\mu c_{ij}}{\sum_{\substack{j=1 \\ j \neq i}}^N c_{ij}} - \epsilon \xi_i^\mu \right)^2 ; \quad \boxed{\tilde{\Omega}_{ij}^\nu = \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^p \xi_i^\mu \xi_j^\mu \xi_j^\nu} \quad (38)$$

Tensor de ruido sin normalizar

Sin embargo, en caso de que el algoritmo haga que la red tienda a una configuración de conectividad constante, es válido utilizar esta conectividad estacionaria como el valor de normalización a lo largo de todo el proceso de optimización. Esto se debe a que en el límite de la capacidad de almacenamiento, ésta será la conectividad que definirá la magnitud de la señal en el campo de actividad. En la sección 6.3 se muestra que, efectivamente, el algoritmo online hace tender la conectividad de la red a un valor estacionario c_{est} . Por lo tanto, se puede hacer que ϵ absorba esta constante y redefinir la función de costo como

$$\tilde{E}_i^\epsilon = \sum_{\mu=1}^p \left(\sum_{\substack{j=1 \\ j \neq i}}^N \tilde{\Omega}_{ij}^\mu c_{ij} - \epsilon \xi_i^\mu \right)^2 \quad (39)$$

Sin embargo, el punto débil de esta función es que requiere de la identificación de la señal y el ruido ya que se asume que la neurona es capaz de discriminar los aportes de cada uno dentro del campo de actividad que recibe. Es fácil ver que el término de ruido puede reescribirse en función del campo de actividad total (dándole un carácter más biológicamente aceptable a la expresión de la función), pero la identificación de la señal es una característica propia de esta función de costo

$$\begin{aligned}
\tilde{E}_i^\epsilon &= \sum_{\mu=1}^p \left(\sum_{\substack{j=1 \\ j \neq i}}^N \tilde{\Omega}_{ij}^\mu c_{ij} - \epsilon \xi_i^\mu \right)^2 = \sum_{\mu=1}^p \left[\sum_{\substack{j=1 \\ j \neq i}}^N W_{ij} \xi_j^\mu c_{ij} - \left(\sum_{\substack{j=1 \\ j \neq i}}^N c_{ij} + \epsilon \right) \xi_i^\mu \right]^2 \\
&= \sum_{\mu=1}^p \left(\tilde{h}_i^\mu[c_{ij}] - \tilde{\epsilon}[c_{ij}] \xi_i^\mu \right)^2
\end{aligned} \tag{40}$$

donde $\tilde{\epsilon}[c_{ij}] = \sum_{\substack{j=1 \\ j \neq i}}^N c_{ij} + \epsilon$ es el nuevo parámetro de regulación y $\tilde{h}_i^\mu[c_{ij}]$ el campo de actividad sin normalizar, ambos con una dependencia funcional de la conectividad c_{ij} de la neurona. Existen muchas formas alternativas de plantear la función de costo y algunas de ellas se exploraron en el Apéndice B. De todos modos, todas requieren de que las neuronas sean capaces de reconocer a qué estado debe llevarlas su conectividad. Queda abierta la pregunta para futuros trabajos de cómo sería posible construir una función de costo con una mayor plausibilidad biológica.

En otro orden de ideas, al momento de utilizar la función de la ecuación 39, fue necesario definir el parámetro ϵ . En las optimizaciones realizadas mediante el algoritmo simulado, se observó la existencia de un parámetro óptimo ϵ_c capaz de maximizar el número de patrones donde el ruido refuerza la señal. Particularmente, se encontró una dependencia de este parámetro con α , y en el caso de redes de conectividad diluida se observó que $\epsilon_c \sim \alpha$ (ver Apéndice A). A la hora de estudiar el algoritmo online, se decidió simplificar la variabilidad de ϵ dejándolo constante (es decir, sin depender del número de patrones almacenados ni de la conectividad). En el Apéndice B se muestra la existencia de un parámetro ϵ_c capaz de mejorar el rendimiento del algoritmo. En los resultados que se muestran a continuación se utilizó dicho parámetro ϵ_c .

6.3. Variación de la Conectividad Media

Uno de los resultados más interesantes obtenidos al implementar el algoritmo online fue el efecto que tuvo sobre la conectividad de la red. Como ya se mencionó previamente, el algoritmo permite que cada neurona corte o genere conexiones, según se minimice la función de costo. Se observó que la conectividad media tiende a estabilizarse en un valor $c_{est}(\epsilon)$, de acuerdo al ϵ que se decida utilizar, independientemente de la conectividad inicial c_{init} . Para redes con conectividad inicial por encima de $c_{est}(\epsilon)$ se obtuvo que su conectividad media decrece monótonamente, es decir, predomina un proceso de poda de conexiones. Para redes de conectividad inicial por debajo de $c_{est}(\epsilon)$ se observó una dinámica que parece coincidir cualitativamente con lo observado experimentalmente: se tiene un primer período donde predomina la generación de conexiones, y tras alcanzar un pico, empieza a predominar la poda de conexiones hasta alcanzar la estabilidad (ver figura 25).

Es importante notar que el paralelismo presentado en el párrafo anterior está vinculando

la evolución de redes numéricas a medida que se le cargan patrones con la evolución de redes biológicas a lo largo del tiempo. Una razón para justificar esta relación establecida es suponer que en el modelo computacional, la red va almacenando patrones a lo largo del tiempo, por lo tanto, existe una relación directa entre el tiempo transcurrido y la cantidad de información almacenada. De hecho, es de esperar que una memoria biológica vaya acumulando información conforme transcurre el tiempo.

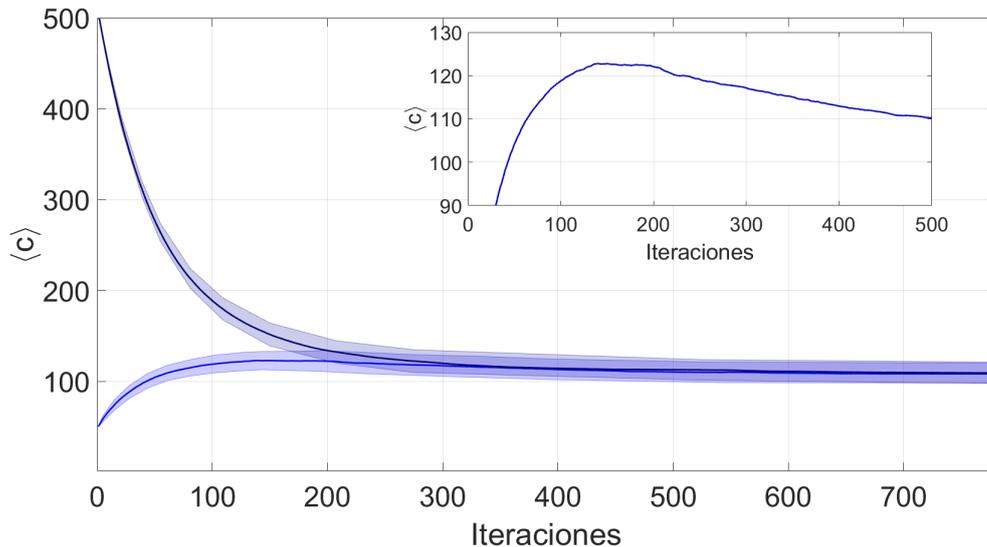


Figura 25: Resultados obtenidos al implementar el algoritmo online (minimizando \tilde{E}_i^ϵ con $\epsilon = \epsilon_c$) sobre redes de 500 neuronas con conectividades iniciales $c_{init} = 50$ y 499. Se muestra la conectividad media en función de las iteraciones. En color atenuado se grafica la desviación estándar de la conectividad media. En el subpanel en el margen superior derecho se selecciona una parte de la curva correspondiente a $c_{init} = 50$ para destacar el máximo global que alcanza la conectividad.

6.4. Capacidad de Almacenamiento

El algoritmo online permitió mejorar sustancialmente la capacidad de almacenamiento de las redes autoasociativas. La mejora obtenida fue comparable con la mejora dada por el algoritmo de enfriamiento simulado. Se definió al parámetro $\alpha_{eff} = p_{eff}/c$ como la capacidad de almacenamiento efectiva de la red, siendo p_{eff} el número de patrones que la red puede recuperar del total p almacenado ($0.9p \leq p_{eff} \leq p$). Se estudió la dinámica de la capacidad de almacenamiento de la red a partir de la evolución de p_{eff} a lo largo de las iteraciones. En la figura 26 se observa que en cada paso donde se agregan Δp patrones, el valor de p_{eff} cae abruptamente. Esto se debe a que se le están agregando memorias a la red para las cuales su conectividad no se encuentra optimizada. A partir de esta caída, la red comienza a reoptimizar su conectividad, donde tiene en cuenta este nuevo conjunto de patrones, hasta alcanzar un punto de estabilidad, repitiendo el ciclo, nuevamente.

Es posible notar que la red alcanza un máximo de almacenamiento, y al repetir el ciclo, no vuelve a recuperar el 90% de los patrones. En cambio, se estabiliza en un p_{eff} cercano al máximo, representando un número mayor al 70% de los patrones almacenados. Se puede interpretar

que alcanzado un punto crítico de patrones, la red ya no es capaz de de optimizar de ninguna forma su conectividad para almacenar más memorias. Esto es consistente con lo que sucedió en las optimizaciones mediante el algoritmo de enfriamiento simulado, donde la optimización pudo mejorar de forma finita la capacidad de almacenamiento.

La máxima capacidad de almacenamiento efectiva que alcanza la red es aproximadamente independiente de la conectividad inicial, si bien depende fuertemente del parámetro ϵ , como se muestra explícitamente en el Apéndice B. Este hecho sugiere que el algoritmo siempre converge a una configuración de conectividad que representa un mínimo global en el espacio de soluciones (ver figura 27).

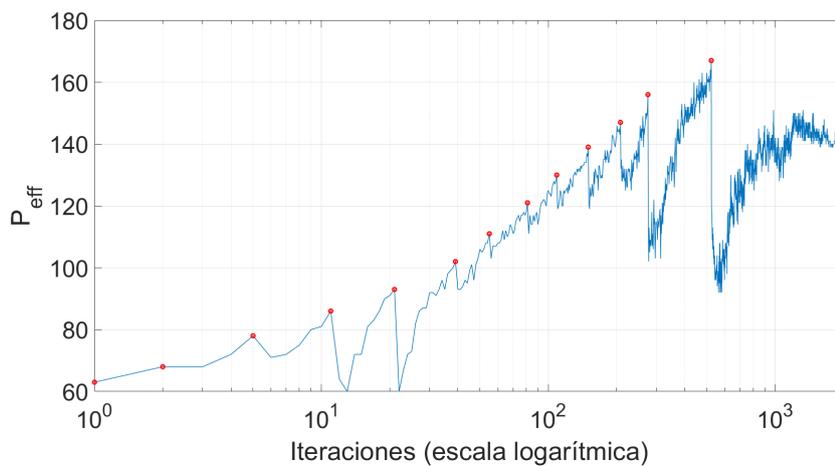


Figura 26: Evolución del número de patrones recuperados efectivamente por la red (p_{eff}) a lo largo de las iteraciones del algoritmo. En rojo se marcan los puntos donde la red alcanza a recuperar al menos un 90% de los patrones almacenados, y en el siguiente paso se cargan 10 patrones más (generando la caída abrupta que se observa en el número efectivo de patrones que la red puede recuperar). Se utiliza la escala logarítmica en las iteraciones para mejorar la resolución en los saltos de estabilidad que tiene la red en las primeras iteraciones.

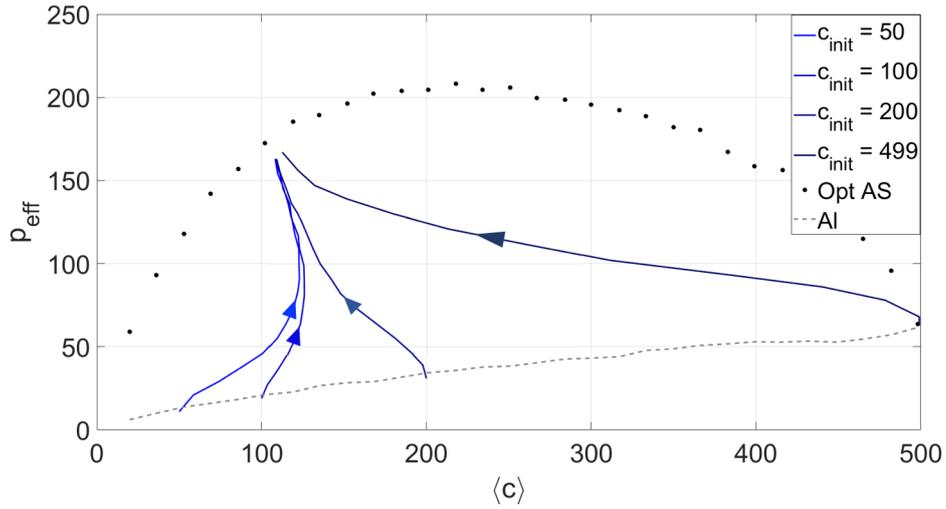


Figura 27: Se muestran las trayectorias que siguen las redes en el espacio dado por el número de patrones que la red puede recuperar de forma efectiva (p_{eff}) y la conectividad media $\langle c \rangle$ (la dirección se marca con una flecha sobre cada curva) al ser optimizadas mediante el algoritmo online (minimizando E_i^ϵ con $\epsilon = \epsilon_c$). Cada curva corresponde a una red con distinta conectividad inicial aleatoria (c_{init}). Se observa que la capacidad efectiva inicial de todas las redes se encuentra sobre la curva dada por la capacidad de almacenamiento de una red de conectividad aleatoria (indicada mediante una línea gris rayada, nombrada como ‘AI’ en la leyenda). Las máximas capacidades efectivas alcanzadas por todas las redes convergen a puntos muy cercanos entre sí, que, a su vez, se encuentran próximos a la curva dada por la capacidad de almacenamiento óptima obtenida mediante el algoritmo de enfriamiento simulado utilizando el parámetro $\epsilon = \alpha$ (indicada mediante puntos negros, nombrada como ‘Opt ES’ en la leyenda).

7. Conclusiones

En este trabajo se consiguió demostrar que es posible mejorar la capacidad de almacenamiento de una red neuronal autoasociativa reconfigurando su conectividad estructural en función de los patrones de actividad almacenados en la misma. Para ello, se planteó un problema de optimización binaria para cada neurona proponiendo una función de costo a minimizar, donde las variables a optimizar fueron sus conexiones presinápticas. A dicho problema se le impuso la restricción de conectividad presináptica constante (c) para poder utilizarla como parámetro de control. Para resolver este problema, se adaptó un algoritmo de enfriamiento simulado que se corrió sobre cada neurona, siguiendo un protocolo de enfriamiento para determinar la probabilidad de permutación entre dos estados de conexión presináptica. La función de costo propuesta fue la siguiente suma cuadrática $E_i^\epsilon = \sum_{\mu=1}^p (\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu \cdot c_{ij} - \epsilon \xi_i^\mu)^2$ tal que su mínimo absoluto representara un caso donde $\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu \cdot c_{ij} = \epsilon \xi_i^\mu$ para todo patrón μ . Para la elección del parámetro $\epsilon = 0$, se observó que la optimización tiende a anticorrelacionar el signo del ruido con la señal. Sin embargo, una reducción efectiva en el ancho de la distribución de ruido permite mejorar la capacidad de almacenamiento, alcanzando una mejora cerca de 7 veces la capacidad de una red de conectividad aleatoria para $c/N \sim 0.2$. Para la elección del parámetro $\epsilon = \alpha (= p/c)$, se observó una mejora en la capacidad de almacenamiento que alcanza su máximo en las conectividades más bajas estudiadas ($c/N < 0.1$), llegando a conseguir una capacidad entre 8 y 9 veces mayor a la de una red de conectividad aleatoria. Esta optimización resultó superior a la de $\epsilon = 0$ en el rango $c/N < 0.4$. Este resultado sugiere que en las redes de baja conectividad de los mamíferos, un proceso de optimización sobre la conectividad que tienda a hacer que el ruido presente en el campo de actividad refuerce la señal podría ser sumamente beneficioso. Se estudió la correlación del ruido con la señal y se observó que, efectivamente, en esta optimización el signo del ruido tiende a ser igual al de la señal. Este efecto permite que la red tenga una mayor tolerancia a las fluctuaciones aleatorias del ruido, mejorando su capacidad de almacenamiento. Para una fracción $c/N = 0.01$ se obtuvo un $\alpha_c \sim 3.15$ en el caso $\epsilon = \alpha$, mientras que para $\epsilon = 0$ se obtuvo en la misma conectividad $\alpha_c \sim 1.49$. Ambas optimizaciones tuvieron resultados similares para la región $c/N > 0.4$.

Luego, se estudiaron los cambios en la matriz de adyacencia de la red (J) debidos a las optimizaciones realizadas sobre su conectividad estructural. Se observó que en el caso $\epsilon = 0$, si $c/N < 0.5$ la red tiende a eliminar conexiones de pesos sinápticos de módulo muy alto y muy bajo, quedándose con los pesos intermedios. A medida que la conectividad aumenta, la red comienza a recurrir a pesos de módulo cercano a cero. Conforme $c/N \rightarrow 1$, la red agota las sinapsis más chicas y recurre progresivamente a utilizar sinapsis más altas. Por otro lado, en el caso $\epsilon = \alpha$, si la conectividad es baja, la red recurre primero a los pesos más altos. Este resultado es coherente con el trabajo de Montemurro y Tamarit (2001), donde muestran que elegir las c conexiones con peso sináptico más alto en una red de conectividad diluida mejora la capacidad de almacenamiento de la misma. A medida que $c/N \rightarrow 1$, nuevamente, la red recurre a los pesos sinápticos de módulo más bajo, y al agotarlos, va tomando progresivamente sinapsis de módulo más alto.

Para complementar el análisis de la mejora en la capacidad de almacenamiento, se estudiaron las cuencas de atracción de las redes optimizadas. Se observó que en el caso $\epsilon = 0$, si la conectividad es baja, la robustez de los atractores disminuye de forma continua a medida que el número de patrones almacenados se acerca al límite crítico. La consecuencia de este resultado es que los patrones en el punto crítico no pueden ser fácilmente recuperados por la red, ya que poseen una tolerancia muy baja a errores en el estado inicial de la misma. Por otro lado, en las optimizaciones con $\epsilon = \alpha$ los atractores almacenados mostraron una mayor robustez en esta conectividad, si bien también disminuye de forma continua con la carga de la red. Esto hace que una optimización de este tipo tenga una ventaja biológica por encima de la minimización del ruido (optimización con $\epsilon = 0$).

En otro orden de ideas, se estudió el efecto que tienen ciertas modificaciones biológicamente plausibles del modelo de Hopfield sobre las optimizaciones realizadas. Particularmente, se relajó la restricción de conectividad binaria $c_{ij} \in \{0, 1\}$ permitiéndole a cada elemento de la matriz de conectividad tomar valores menores o iguales a algún umbral n (es decir, $c_{ij} \in \{0, 1, \dots, n\}$). Se observó que relajar esta restricción mejora notablemente la optimización en un rango de conectividades altas ya que aumenta el número de configuraciones posibles de la red. Sin embargo, no trae efectos sobre las conectividades bajas, que es el rango de interés biológico. Por otro lado, una segunda modificación fue reducir la actividad media de los patrones de actividad a una fracción a de la red, realizando el cambio de variables $\xi_i^\mu \in \{-1, 1\} \rightarrow \eta_i^\mu \in \{0, 1\}$. Al hacer esto, se obtuvo que el rendimiento de la optimización decrece a medida que disminuye a . Los resultados obtenidos muestran que las modificaciones exploradas no benefician al proceso de optimización realizado. Sin embargo, se sugiere revisar la implementación del algoritmo de enfriamiento simulado en estos casos (como por ejemplo, la propuesta de una función de costo alternativa), ya que podría ser posible realizar cambios que se beneficien de los elementos biológicamente plausibles introducidos.

Por último, se diseñó un algoritmo con características biológicamente plausibles capaz de llevar la capacidad de almacenamiento de la red por encima del límite teórico de conectividad aleatoria (en el modelo clásico de Hopfield). El algoritmo consiste en una optimización en paralelo (de allí su nombre *algoritmo online*) donde cada neurona intenta minimizar una función de costo que sólo contenga información local. Para minimizar esta función, cada neurona genera y elimina conexiones siempre que produzca una diferencia negativa entre el valor final e inicial de la función de costo (similar al algoritmo de enfriamiento simulado). Una vez que la red puede recuperar un 90% de los patrones almacenados, se le cargan nuevos y se repite el proceso. Al implementar el algoritmo utilizando una versión de la función de costo E_i^ϵ sin normalizar, se encontró que éste lleva a la red a un punto de estancamiento donde se estabiliza su conectividad media. Asimismo, la máxima capacidad de almacenamiento que alcanza la red se encuentra muy cercana a la capacidad obtenida mediante el algoritmo de enfriamiento simulado, y la misma no depende de las condiciones iniciales. Se observó que de acuerdo a la conectividad inicial (c_{init}), si ésta es muy alta, predomina un proceso de poda de conexiones hasta que la conectividad se estabiliza. Si c_{init} es lo suficientemente baja, la red atraviesa un primer período donde predomina la generación de conexiones, y alcanzado un pico, atraviesa un segundo período donde predomina la poda de conexiones hasta estabilizarse. Este resultado surge naturalmente de aplicar principios de

minimización locales a la formación y eliminación de sinapsis y muestra similitud con la dinámica de generación y eliminación de conexiones observada en algunos mamíferos (como el ser humano y las ratas). De todos modos, el punto más débil en el algoritmo online, en cuanto a su plausibilidad biológica, se encuentra en la función de costo utilizada. Esta función asume que la neurona *conoce* el estado al que debe llevarla su conectividad. Se deja abierta la puerta a futuras modificaciones de esta función que permitan mejorar su plausibilidad biológica.

8. Apéndice A: Optimización en el Modelo Clásico de Hopfield

8.1. Valor Óptimo del Parámetro ϵ

Al momento de optimizar la conectividad de una red utilizando la función de costo

$$E_i^\epsilon = \sum_{\mu=1}^p \left(\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu c_{ij} - \epsilon \xi_i^\mu \right)^2 \quad (41)$$

es necesario determinar el valor de ϵ . En principio, no se conoce hasta qué magnitud es posible llevar al ruido de forma tal que conserve el signo de la señal. Podría suceder que la optimización mejore siempre que se aumente el valor de ϵ , como también que exista un valor crítico a partir del cual la optimización deje de ser eficaz. Si se optimiza la conectividad de una neurona minimizando la ecuación 41, lo que se esperaría obtener en el caso ideal es que para todo patrón μ suceda que

$$\xi_i^\mu \sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\mu c_{ij} > 0 \quad (42)$$

Por lo tanto, un buen estimador de la eficacia de la optimización es la fracción de patrones (del total p almacenado) que satisface la ecuación 42. Mediante simulaciones (para distintos valores de N , c y p), se realizó un barrido del parámetro ϵ , optimizando en cada valor la conectividad de múltiples neuronas, y se contabilizó dicho estimador. Se obtuvo que para todo número de patrones cargados, existe un valor crítico ϵ_c para el cual se maximiza la fracción de patrones donde el ruido refuerza la señal. Para $\epsilon > \epsilon_c$, la optimización tiende a estabilizarse (ver figura 28).

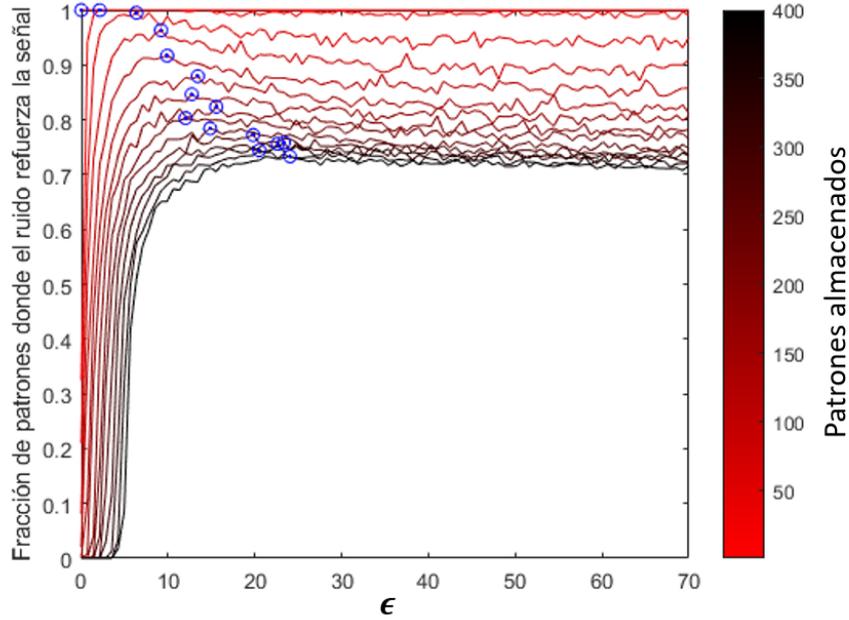


Figura 28: Se grafica a la fracción de patrones almacenados en los que el ruido refuerza a la señal para una neurona en función del parámetro ϵ . Los resultados fueron obtenidos luego de optimizar la conectividad de una red de 2000 neuronas con 20 conexiones por neurona (minimizando la función E_i^ϵ). La escala de colores indica el número de patrones para el cual la red fue optimizada, yendo de 2 (rojo) a 400 (negro) patrones. En azul se marcan los puntos correspondientes al valor crítico ϵ_c , el cual maximiza el estimador para cada conjunto de patrones cargados a la red. Se observa que a medida que se aumenta el número de patrones cargados a la red, la optimización empeora (la cantidad de patrones donde el ruido refuerza la señal es cada vez menor en promedio), lo cual es esperable dado que se aproxima (y supera) al límite crítico de almacenamiento.

Asimismo, se observó que ϵ_c tiene una dependencia monótona con el parámetro $\alpha = p/c$. Esta tendencia monótona depende también de la fracción c/N . En los primeros 4 gráficos de la figura 29 se muestran las curvas de ϵ_c vs. α para distintos valores de c/N . Para $c/N \geq 0.8$, la curva tiene mucha dispersión, dejando de ser clara la tendencia monótona. Ajustando las curvas ϵ_c vs. α por un modelo lineal de ordenada al origen nula, se obtuvo que la tasa de crecimiento aumenta con la fracción c/N . En el gráfico inferior de la figura 29 se grafican los resultados de las pendientes obtenidas al ajustar las curvas con un modelo lineal, y su respectivo intervalo de confianza (95%). Para valores de conectividad bajos, las pendientes están cercanas a la unidad, es decir que $\epsilon_c \sim \alpha$. Para valores de conectividad altos $c/N \geq 0.8$ se ve un aumento sustancial en los intervalos de confianza de las pendientes ya que que el modelo lineal deja de ser una buena aproximación.

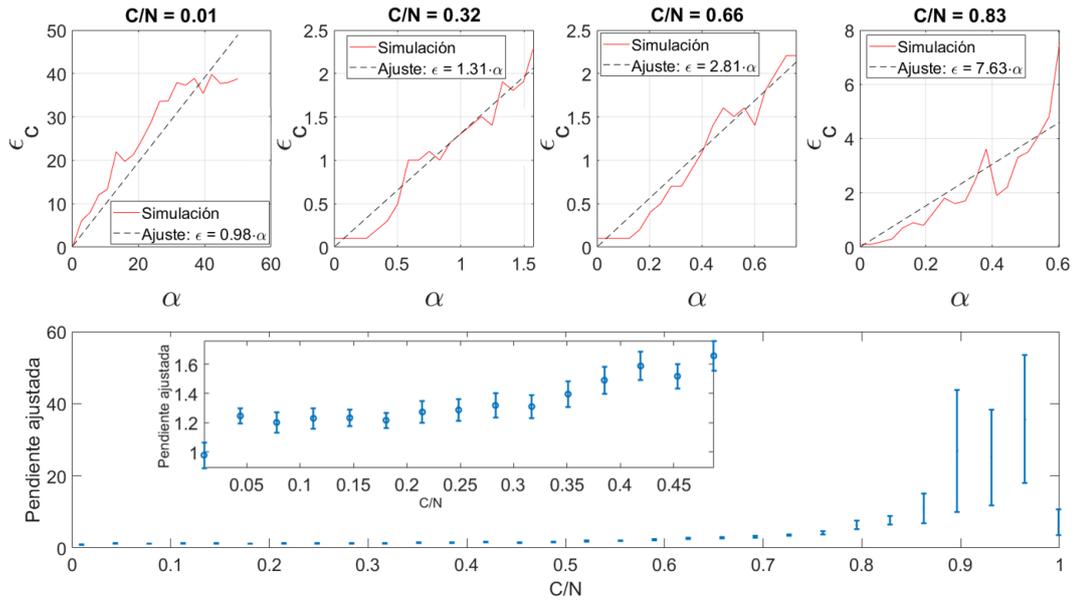


Figura 29: Primeros cuatro gráficos superiores: se grafica la dependencia de ϵ_c con $\alpha = p/c$ para distintos valores de c/N . Gráfico inferior: se grafican las pendientes de los modelos lineales con ordenada al origen nula ajustados a las curvas $\epsilon_c(\alpha)$, en función de la fracción c/N . Las barras de error indican el intervalo de confianza de 95 % del parámetro calculado mediante el ajuste. En zoom se muestra la región $c/N \leq 0.5$ donde las pendientes están cercanas a 1. Particularmente, las simulaciones de la figura fueron hechas con redes de 2000 neuronas

9. Apéndice B: Algoritmo Online

9.1. Funciones de Costo Alternativas

Partiendo de la función de costo E_i^ϵ , es posible plantear distintas alternativas que no modifiquen sus elementos biológicamente plausibles. Por un lado, al momento de utilizar el algoritmo online, se debe decidir qué normalización del campo de actividad corresponde utilizar. En la sección 6.2 se indicó que la normalización a usar sería la conectividad estacionaria (c_{est}) que alcanza la red en el límite de almacenamiento. En este caso, la normalización es sólo una constante que no modifica la dinámica de la función de costo y que puede ser absorbida en el parámetro ϵ . Esto permite definir la función utilizada en dicha sección

$$\tilde{E}_i^\epsilon = \sum_{\mu=1}^p \left(\sum_{\substack{j=1 \\ j \neq i}}^N \tilde{\Omega}_{ij}^\mu c_{ij} - \epsilon \xi_i^\mu \right)^2 ; \quad \tilde{\Omega}_{ij}^\nu = \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^p \xi_i^\mu \xi_j^\mu \xi_j^\nu \quad (43)$$

Esta función es equivalente a optimizar el campo de actividad total variando el parámetro $\tilde{\epsilon}[c_{ij}]$ de forma proporcional con la conectividad

$$\tilde{E}_i^\epsilon = \sum_{\mu=1}^p \left[\tilde{h}_i^\mu[c_{ij}] - \underbrace{\left(\sum_{\substack{j=1 \\ j \neq i}}^N c_{ij} + \epsilon \right)}_{\tilde{\epsilon}[c_{ij}]} \right] \xi_i^\mu ; \quad \tilde{h}_i^\mu[c_{ij}] = \sum_{\substack{j=1 \\ j \neq i}}^N W_{ij} \xi_j^\mu c_{ij} \quad (44)$$

Sin embargo, otra función de costo admisible puede resultar de proponer que la señal esté modulada por un parámetro ϵ constante. Es decir, proponer una función que al ser minimizada, lleve a todo el campo de actividad a un valor constante $\epsilon \xi_i^\mu$

$$\tilde{H}_i^\epsilon = \sum_{\mu=1}^p \left(\tilde{h}_i^\mu[c_{ij}] - \epsilon \xi_i^\mu \right)^2 \quad (45)$$

En estas dos funciones (\tilde{E}_i^ϵ y \tilde{H}_i^ϵ) se normaliza el campo por una conectividad constante (c_{est}), pero esto es sólo una convención. Como alternativa, es plausible proponer una optimización donde el campo se renormalice en cada iteración. Si este es el caso, sería posible utilizar la función original E_i^ϵ o bien la versión *normalizada* de \tilde{H}_i^ϵ

$$E_i^\epsilon = \sum_{\mu=1}^p \left(\frac{\sum_{\substack{j=1 \\ j \neq i}}^N \tilde{\Omega}_{ij}^\mu c_{ij}}{\sum_{\substack{j=1 \\ j \neq i}}^N c_{ij}} - \epsilon \xi_i^\mu \right)^2 ; \quad H_i^\epsilon = \sum_{\mu=1}^p \left(\frac{\sum_{\substack{j=1 \\ j \neq i}}^N W_{ij} \xi_j^\mu c_{ij}}{\sum_{\substack{j=1 \\ j \neq i}}^N c_{ij}} - \epsilon \xi_i^\mu \right)^2 \quad (46)$$

En las siguientes subsecciones, se muestran los resultados obtenidos al optimizar redes mediante el algoritmo online utilizando cada una de las 4 funciones de costo E_i^ϵ , \tilde{E}_i^ϵ , H_i^ϵ y \tilde{H}_i^ϵ . Se obtuvo que el resultado de la optimización de la conectividad es sensible a la elección de la función de costo.

9.2. Valor Óptimo del Parámetro ϵ

Se corrió el algoritmo online sobre redes de 500 neuronas para distintos valores de ϵ variando su conectividad inicial. Se encontró que, para cada función de costo, la mejora en la capacidad de almacenamiento efectiva varía de acuerdo a la elección del parámetro ϵ . Se obtuvo la existencia de un valor crítico ϵ_c que maximiza α_{eff} . Sin embargo, ϵ_c no necesariamente maximiza p_{eff} , como se observa en la figura 30. El parámetro crítico ϵ_c no presentó dependencia con la conectividad inicial de la red.

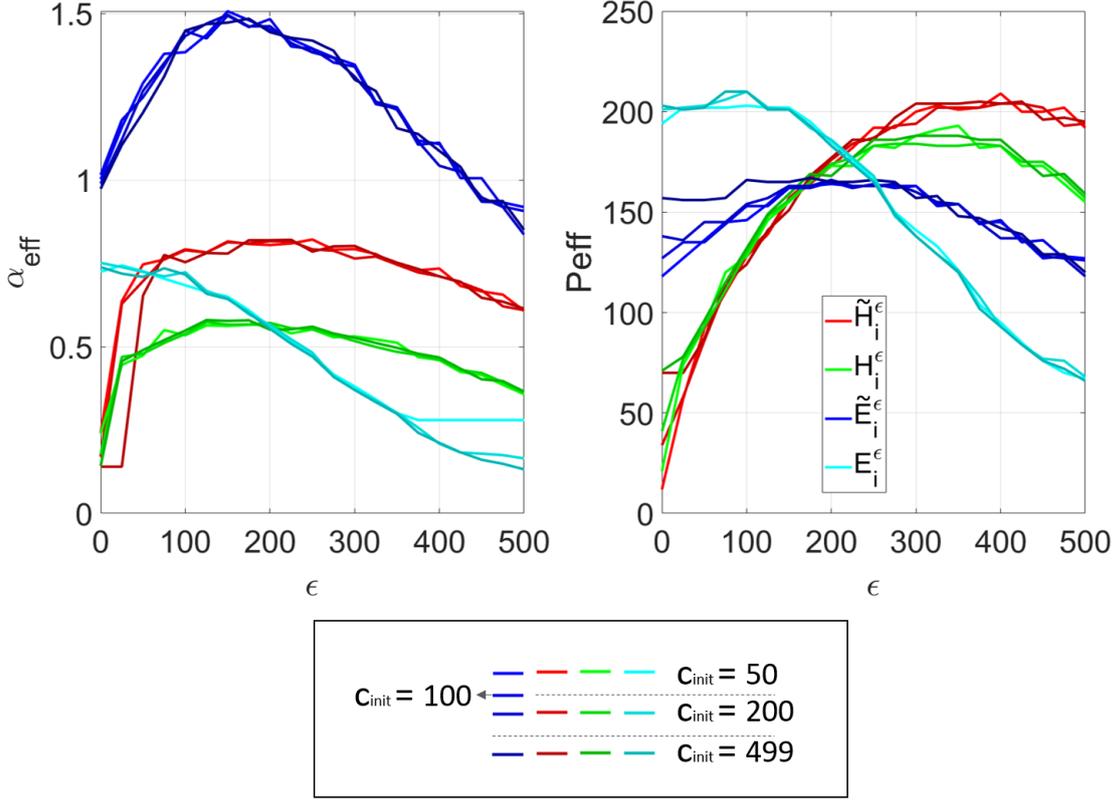


Figura 30: Se grafica α_{eff} (gráfico izquierdo) y p_{eff} (gráfico derecho) en función de ϵ para las 4 funciones de costo propuestas (cada color indica una), inicializando la red en distintas conectividades c_{init} (cada tono de color distinto) al implementar la optimización mediante el algoritmo online. En ambos gráficos, α_{eff} y p_{eff} corresponden al máximo valor alcanzado durante la optimización. Se observa que el máximo de las curvas no muestra dependencia con la conectividad inicial, pero depende de la función de costo utilizada. Además, los máximos de α_{eff} y p_{eff} no coinciden.

	\tilde{H}_i^ϵ	H_i^ϵ	\tilde{E}_i^ϵ	E_i^ϵ
ϵ_c	250	125	150	0
α_{eff}	0.82	0.57	1.51	0.75
p_{eff}	192	148	163	201

Tabla 1: Se muestran los valores de ϵ_c que maximizan α_{eff} , junto con los correspondientes valores de p_{eff} , para cada función de costo utilizada. El ϵ_c correspondiente a la función \tilde{E}_i^ϵ fue el utilizado en la sección 6.2.

9.3. Capacidad de Almacenamiento

$$sgn(\xi_i^\nu) = sgn\left(\sum_{\substack{j=1 \\ j \neq i}}^N \Omega_{ij}^\nu c_{ij}\right) \quad (47)$$

Se obtuvo que todas las funciones de costo fueron capaces de mejorar la capacidad de almacenamiento de redes inicializadas con distinta conectividad aleatoria inicial. En general, al utilizar

el valor ϵ_c , la máxima capacidad de almacenamiento efectiva alcanzada por las redes converge a un punto cercano a la curva dada por la capacidad de almacenamiento óptima lograda por el algoritmo de enfriamiento simulado (minimizando E_i^ϵ con $\epsilon = \alpha$), como se muestra en la figura 31. Cada uno de estos puntos es distinto, según la función de costo utilizada, lo que significa que la dinámica del algoritmo online depende fuertemente de las características de la función a minimizar. Salvo por la función H_i^ϵ , el estado final de la red (en cuanto a su conectividad y capacidad de almacenamiento), es independiente del estado inicial.

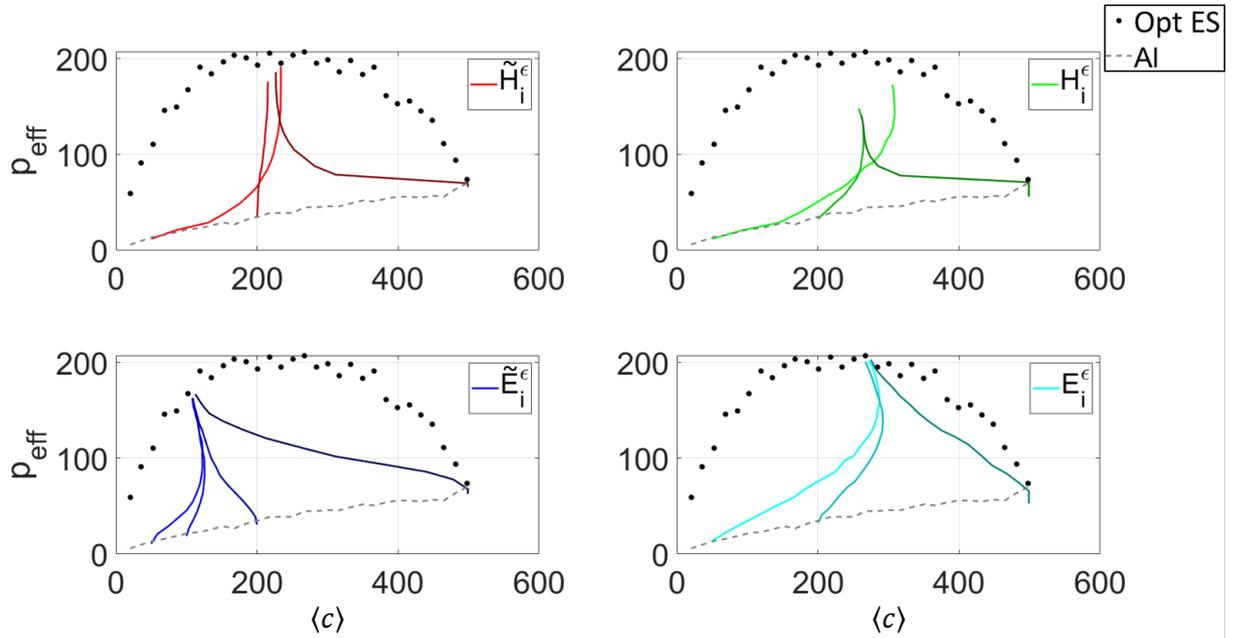


Figura 31: En cada figura se grafican las curvas p_{eff} vs. $\langle c \rangle$ que siguen las redes al ser optimizadas mediante el algoritmo online utilizando el parámetro de regulación ϵ_c hallado en la subsección 9.2 de este apéndice. Cada panel corresponde a una de las 4 funciones de costo presentadas. Para cada función, se inicializaron redes con distinta conectividad aleatoria c_{init} (en los paneles se utiliza el mismo código de color utilizado en la figura 30). El número de patrones que inicialmente puede almacenar cada red se encuentra sobre la curva gris rayada, correspondiente a la capacidad de almacenamiento de una red de conectividad aleatoria (indicado en la leyenda como ‘AI’). El número máximo de patrones que alcanzan a almacenar y recuperar de forma efectiva las redes (p_{eff}) se encuentra cercano a la curva de puntos, que indican la capacidad de almacenamiento de una red optimizada mediante el algoritmo de enfriamiento simulado, utilizando $\epsilon = \alpha$ (indicado en la leyenda como ‘Opt ES’).

Bibliografía

- [1] Amit, DJ. *Modelling Brain Function*. Cambridge Univ. Press, Nueva York, 1989.
- [2] Hopfield John J. “Neural networks and physical systems with emergent collective computational abilities”. En: *Proceedings of the national academy of sciences* 79.8 (1982), págs. 2554-2558.
- [3] Hebb, DO. *Organization of behavior*. Nueva York: Wiley, 1949.
- [4] Kropff, Emilio y Treves, Alessandro. “Uninformative memories will prevail: the storage of correlated representations and its consequences”. En: *HFSP journal* 1.4 (2007), págs. 249-262.
- [5] Treves, Alessandro y Rolls, Edmund T. “What determines the capacity of autoassociative memories in the brain?” En: *Network: Computation in Neural Systems* 2.4 (1991), págs. 371-397.
- [6] Roudi, Yasser y Latham, Peter E. “A balanced memory network”. En: *PLoS Comput Biol* 3.9 (2007), e141.
- [7] Hua Jackie Yuanyuan y Smith Stephen J. “Neural activity and the dynamics of central nervous system development”. En: *Nature neuroscience* 7.4 (2004), págs. 327-332.
- [8] Huttenlocher, Peter R y col. “Synaptic density in human frontal cortex—developmental changes and effects of aging”. En: *Brain Res* 163.2 (1979), págs. 195-205.
- [9] Huttenlocher, Peter R, de Courten, Christian, Garey, Laurence J y Van der Loos, Hendrik. “Synaptogenesis in human visual cortex—evidence for synapse elimination during normal development”. En: *Neuroscience letters* 33.3 (1982), págs. 247-252.
- [10] Chklovskii, Dmitri B, Mel, BW y Svoboda, K. “Cortical rewiring and information storage”. En: *Nature* 431.7010 (2004), págs. 782-788.
- [11] Herculano-Houzel, Suzana, Mota, Bruno, Wong, Peiyan y Kaas, Jon H. “Connectivity-driven white matter scaling and folding in primate cerebral cortex”. En: *Proceedings of the National Academy of Sciences* 107.44 (2010), págs. 19008-19013.
- [12] Assaf, Yaniv, Bouznach, Arieli, Zomet, Omri, Marom, Assaf y Yovel, Yossi. “Conservation of brain connectivity and wiring across the mammalian class”. En: *Nature neuroscience* 23.7 (2020), págs. 805-808.
- [13] Wulfram Gerstner, Werner M. Kistler, Richard Naud y Liam Paninski. “Neuronal Dynamics: From single neurons to networks and models of cognition and beyond”. En: Septiembre 2014. Cap. 17.
- [14] Fauth, Michael, Wörgötter, Florentin y Tetzlaff, Christian. “The formation of multi-synaptic connections by the interaction of synaptic and structural plasticity and their functional consequences”. En: *PLoS Comput Biol* 11.1 (2015), e1004031.
- [15] Markram, Henry, Lübke, Joachim, Frotscher, Michael, Roth, Arnd y Sakmann, Bert. “Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex.” En: *The Journal of physiology* 500.2 (1997), págs. 409-440.
- [16] Derrida, Bernard, Gardner, Elizabeth y Zippelius, Anne. “An exactly solvable asymmetric neural network model”. En: *EPL (Europhysics Letters)* 4.2 (1987), pág. 167.

- [17] Arenzon, JJ y Lemke, N. “Simulating highly diluted neural networks”. En: *Journal of Physics A: Mathematical and General* 27.15 (1994), pág. 5161.
- [18] Montemurro, Marcelo A y Tamarit, Francisco A. “An efficient dilution strategy for constructing sparsely connected neural networks”. En: *Physica A: Statistical Mechanics and its Applications* 294.3-4 (2001), págs. 340-350.
- [19] Buhmann, J, Divko, R y Schulten, K. “Associative memory with high information content”. En: *Physical Review A* 39.5 (1989), pág. 2689.
- [20] Tsodyks, Mikhail V y Feigel'man, Mikhail V. “The enhanced storage capacity in neural networks with low activity level”. En: *EPL (Europhysics Letters)* 6.2 (1988), pág. 101.
- [21] Kirkpatrick, Scott, Gelatt, C Daniel y Vecchi, Mario P. “Optimization by simulated annealing”. En: *science* 220.4598 (1983), págs. 671-680.
- [22] Newman, M y Barkema, G. *Monte carlo methods in statistical physics chapter 1-4*. Vol. 24. Oxford University Press: Nueva York, USA, 1999.
- [23] Yang, Xin-She. *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.
- [24] Yang, Xin-She. “Optimization algorithms”. En: *Computational optimization, methods and algorithms*. Springer, 2011, págs. 13-31.
- [25] Boussaïd, Ilhem, Lepagnot, Julien y Siarry, Patrick. “A survey on optimization metaheuristics”. En: *Information sciences* 237 (2013), págs. 82-117.
- [26] Ben-Ameur, Walid. “Computing the initial temperature of simulated annealing”. En: *Computational optimization and applications* 29.3 (2004), págs. 369-385.
- [27] Manoj Kumar, Mohamed Husain, Naveen Upreti y Deepti Gupta. “Genetic algorithm: Review and application”. En: *Available at SSRN 3529843* (2010).
- [28] Tello, Eduardo Arturo Rodríguez. “Optimización genética de problemas SAT”. Tesis doct. Tecnológico de Monterrey, 1999.
- [29] Wolfe Philip. “The simplex method for quadratic programming”. En: *Econometrica: Journal of the Econometric Society* (1959), págs. 382-398.
- [30] Billionnet Alain , Elloumi Sourour y Lambert Amélie. “Linear reformulations of integer quadratic programs”. En: *International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences*. Springer. 2008, págs. 43-51.
- [31] Yang, Xin-She. “Optimization and metaheuristic algorithms in engineering”. En: *Metaheuristics in water, geotechnical and transport engineering* (2013), págs. 1-23.

Tesis disponible bajo Licencia Creative Commons Reconocimiento – No Comercial – Sin Obra
Derivada (by-nc-nd) 2.5 Argentina
Buenos Aires, 2021