



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN  
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y  
EN SISTEMAS

ESTUDIO DE AUTO-OPTIMIZACIÓN EN REDES  
NEURONALES DE HOPFIELD

T E S I S

QUE PARA OPTAR POR EL GRADO DE:  
MAESTRO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

PRESENTA:  
MARIO ALBERTO ZARCO LÓPEZ

Director de Tesis:  
DR. TOM FROESE  
IIMAS, UNAM

Ciudad Universitaria, CD. MX.      Octubre, 2017



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# Dedicatorias

A mis padres, infinitas gracias por sus incontables formas de apoyo. He llegado hasta aquí porque estoy sentado sobre sus hombros, ustedes son mis gigantes.

A mi abuela, si mañana muriera ten más que claro que fue un placer encontrarte en este mundo desolado. Mi último pensamiento se lo dedicaría a los que amo y justo antes de cerrar los ojos sólo vería tu abrazo.

A Jonathan, Enrique, Victor, Karen y Montserrat, mi familia extendida, gracias por escuchar.

A Oscar, Juan y Julio, gracias por ser grandes amigos antes y durante el posgrado.

A Leticia, Jorge, Héctor, Guillermo y Susana, gracias por las explicaciones y recomendaciones que recibí de ustedes.

# Agradecimientos

Agradezco a la Universidad Nacional Autónoma de México y al Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) por todo de lo que me dejó disponer.

De manera especial le doy las gracias al Dr. Tom Froese por su invaluable ayuda y paciencia en todo momento y en todos aspectos. Gracias por permitirme ser parte del *4E Cognition Group* y por brindarme la oportunidad de asistir a diferentes conferencias. El presente trabajo se realizó con el apoyo del proyecto IA104717 de UNAM-DGAPA-PAPIIT.

La agradezco al IIMAS por permitirme usar el Cluster de Alto Rendimiento, y de manera particular a Adrián Durán Chavesti por la ayuda proporcionada.



# Resumen

Las redes neuronales de Hopfield de tiempo discreto, cuya dinámica presenta múltiples atractores de punto fijo, han sido ampliamente usadas en dos casos: (1) memoria asociativa, basada en aprender un conjunto de patrones de entrenamiento los cuales son representados por atractores, y (2) optimización, basado en representar un problema de satisfacción de restricciones con la topología de la red de tal forma que los atractores sean soluciones de ese problema. En el último caso, la función de energía de la red debe tener la misma forma que la función a ser optimizada, de modo que los mínimos de la primera también sean mínimos de la segunda. Aunque se ha demostrado que los atractores de baja energía tienen un amplio dominio de atracción, la red usualmente queda atrapada en mínimos locales. Recientemente se demostró que las redes de Hopfield de tiempo-discreto pueden converger en atractores globalmente óptimos ampliando las mejores cuencas de atracción. La red combina el aprendizaje de sus propios atractores usando aprendizaje Hebbiano y la aleatorización de los estados neuronales una vez que la red ha reforzado su configuración actual. Dado que el óptimo global tiene subpatrones en común con muchos locales óptimos, el fortalecimiento de los atractores de alta energía a través del aprendizaje tiene el potencial de reforzar los atractores de baja energía incluso antes de que la red converja en estos últimos por primera vez. El denominado proceso de auto-modelado está restringido a ser aplicado a matrices de pesos simétricos sin conexiones auto-recurrentes. En este trabajo se relajan las restricciones de este proceso de auto-optimización mediante el uso de redes neuronales de Hopfield en tiempo continuo con matriz de pesos asimétricos y conexiones auto-recurrentes. El uso de una función de activación continua provoca que los atractores se muevan hacia la esquina del hipercubo del espacio de fase cuando se están reforzando los patrones aprendidos por la red. Aunque los atractores ya no son ni binarios ni están estabilizados de la misma manera que en el modelo discreto de Hopfield, la memoria asociativa permite a la red de Hopfield de tiempo continuo generalizar sobre los patrones aprendidos de tal manera que al reforzar óptimos locales también se refuerzan óptimos que son superiores con respecto a la satisfacción de restricciones. El proceso de auto-modelado puede explotar la estructura de la red para encontrar configuraciones globalmente óptimas, incluso si no se encontró una correlación positiva entre la energía de los atractores y el número de restricciones satisfechas.

## Participación en Conferencias

El objetivo y la motivación de este trabajo se presentaron en *The Fifteenth International Conference on the Synthesis and Simulation of Living Systems (ALIFE XV)* con sede en Cancún, México. El poster presentado se tituló *Can we incorporate sleep-like interruptions into evolutionary robotics?*, y el resumen se puede encontrar en *Late Breaking Abstracts Booklet - Artificial Life Conference 2016*.

Los primeros resultados de este trabajo se presentaron en *2017 Annual International Conference on Biologically Inspired Cognitive Architecture (BICA 2017)* con sede en Moscú, Rusia. El artículo se tituló *Self-modeling in Hopfield Neural Networks with Continuous Activation Function* y se puede encontrar en *BICA 2017 Proceedings*. El trabajo será publicado posteriormente en la revista de acceso abierto de Elsevier llamada *Procedia Computer Science*.

Por último, dos resúmenes fueron presentados en *Conference on Complex Systems 2017 (CCS 2017)* con sede en Cancún, México. El primero se tituló *Self-modeling in continuous-time Hopfield neural networks* y se presentó como poster en la sede principal de la conferencia. El segundo se tituló *A Review and Critique of Gabora's Models of Cultural Evolution* y se presentó como parte de las pláticas de la sesión satélite *Evolution of Cultural Complexity*.

# Índice

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Red de Hopfield . . . . .	1
1.2	Sistemas Dinámicos . . . . .	1
1.2.1	Estabilidad en el sentido de Lyapunov . . . . .	3
1.2.2	Estabilidad en el sentido de Hopfield . . . . .	4
1.3	Aplicaciones de la red de Hopfield . . . . .	5
1.3.1	Aprendizaje . . . . .	5
1.3.2	Optimización . . . . .	6
1.4	Auto-modelado en Redes de Hopfield . . . . .	6
1.5	Motivación . . . . .	7
1.6	Objetivos . . . . .	8
1.6.1	Objetivo General . . . . .	8
1.6.2	Objetivos Específicos . . . . .	8
1.7	Estructura del trabajo . . . . .	8
<b>2</b>	<b>Red Neuronal de Hopfield</b>	<b>10</b>
2.1	Red neuronal de Hopfield de tiempo discreto y estado discreto . .	10
2.2	Red neuronal de Hopfield de tiempo discreto y estado continuo .	10
2.3	Red neuronal de Hopfield de tiempo continuo y estado continuo .	11
<b>3</b>	<b>Memoria Asociativa y Optimización en Redes Neuronales de Hopfield</b>	<b>12</b>
3.1	Aprendizaje Hebbiano . . . . .	12
3.1.1	Regla de Hebb . . . . .	13
3.2	Memoria Asociativa . . . . .	13
3.3	Optimización . . . . .	15
<b>4</b>	<b>Auto-modelado en Redes Neuronales de Hopfield</b>	<b>17</b>
4.1	Auto-modelado en redes de Hopfield de tiempo discreto y estado discreto . . . . .	17
4.1.1	¿Por qué funciona el proceso de auto-modelado? . . . . .	18
4.1.2	Estado del arte . . . . .	19
4.2	Auto-modelado en redes de Hopfield de tiempo discreto y estado continuo . . . . .	20
4.3	Auto-modelado en redes de Hopfield de tiempo continuo y estado continuo . . . . .	21
4.4	Energía y satisfacción de restricciones en redes de Hopfield con conexiones asimétricas y auto-recurrentes. . . . .	22

<b>5</b>	<b>Auto-modelado en redes de Hopfield de tiempo discreto y estado discreto: Resultados y Análisis</b>	<b>25</b>
5.1	Resultados . . . . .	26
5.1.1	Ejemplo ilustrativo de un proceso de auto-modelado . . . . .	27
5.1.2	Resultados de 10000 procesos de auto-modelado . . . . .	29
5.1.3	Prueba $t$ de Student . . . . .	31
5.2	Análisis . . . . .	32
<b>6</b>	<b>Auto-modelado en redes de Hopfield de tiempo discreto y estado continuo: Resultados y Análisis</b>	<b>33</b>
6.1	Resultados . . . . .	34
6.1.1	Ejemplo ilustrativo de un proceso de auto-modelado . . . . .	35
6.1.2	Resultados de 10000 procesos de auto-modelado . . . . .	37
6.1.3	Prueba $t$ de Student . . . . .	39
6.2	Análisis . . . . .	40
<b>7</b>	<b>Auto-modelado en redes de Hopfield de tiempo continuo y estado continuo: Resultados y Análisis</b>	<b>43</b>
7.1	Resultados . . . . .	44
7.1.1	Ejemplo ilustrativo de un proceso de auto-modelado . . . . .	45
7.1.2	Resultados de 10000 procesos de auto-modelado . . . . .	47
7.1.3	Prueba $t$ de Student . . . . .	49
7.2	Análisis . . . . .	50
7.2.1	Auto-modelado como auto-optimización en redes de Hopfield de tiempo continuo y estado continuo . . . . .	52
<b>8</b>	<b>Conclusiones</b>	<b>54</b>
8.1	Trabajo a futuro . . . . .	55

# Capítulo 1

## Introducción

Las redes neuronales han sido estudiadas ampliamente desde diferentes aproximaciones, así como sus posibles aplicaciones. En este trabajo se investiga un proceso que relaciona dos aplicaciones de un modelo de red neuronal conocido como red de Hopfield. En este capítulo se presenta una breve introducción a la red de Hopfield como sistema dinámico, a las aplicaciones de este modelo, y el así llamado proceso de auto-modelado. Después se expone lo que motiva esta investigación, así como los objetivos planteados. Por último se describe la estructura del resto del trabajo.

### 1.1. Red de Hopfield

Una red neuronal es un conjunto interconectado de elementos simples de procesamiento llamados nodos. La capacidad de procesamiento de la red está basada en el valor de las conexiones entre unidades, usualmente llamados pesos, obtenidas mediante un proceso de aprendizaje de un conjunto de patrones de entrenamiento (Gurney, 1997). La red de Hopfield fue presentada por W. A. Little (Little, 1974) y popularizada por J. J. Hopfield (Hopfield, 1982). El modelo más general de la red neuronal de Hopfield se puede describir como una red neuronal recurrente, totalmente conectada y con conexiones auto-recurrentes (véase la Figura 1.1).

La neurodinámica estudia a las redes neuronales como sistemas dinámicos no lineales con especial atención al tema de estabilidad. Por lo tanto, a continuación se presenta una breve introducción a los conceptos básicos usados durante este trabajo.

### 1.2. Sistemas Dinámicos

Un sistema dinámico es aquel en el cual sus estados varían con el tiempo. El estado de un sistema dinámico es el conjunto de variables más pequeño, llamadas variables de estado, de forma que el conocimiento de éstas en  $t = t_0$ , junto con el conocimiento de la entrada para  $t \geq t_0$  determinan completamente el comportamiento del sistema en cualquier  $t \geq t_0$  (Ogata, 2001). El espacio de estados es el espacio  $N$ -dimensional cuyos ejes de coordenadas son las variables

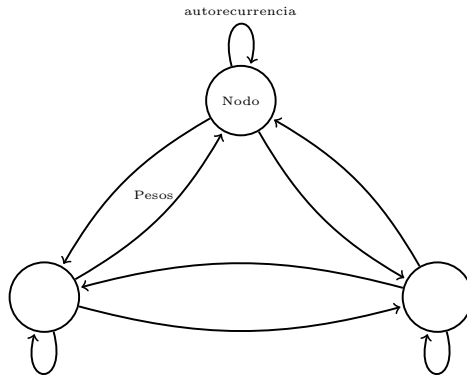


Figura 1.1: Ejemplo de una red de Hopfield totalmente conectada con conexiones auto-recurrentes

de estado. Los estados pueden ser representados como puntos en el espacio de estados.

Un sistema dinámico es entonces una tripleta  $\langle T, S, \phi_t \rangle$  consistente en un conjunto de tiempo ordenado  $T$ , un espacio de estado  $S$ , y un operador de evolución  $\phi_t : S \rightarrow S$  que transforma un estado inicial  $x_0 \in S$  en tiempo  $t_0 \in T$  a otro estado  $x_t \in S$  en tiempo  $t \in T$  (Beer, 2000).

En términos matemáticos, la dinámica de una gran clase de sistemas dinámicos no lineales puede ser descrito por un sistema de ecuaciones diferenciales de primer orden escritas usualmente como

$$\frac{d}{dt}x_j(t) = F_j(x_j), \quad j = 1, \dots, N \quad (1.1)$$

donde  $x_1(t), \dots, x_N(t)$  son las variables de estado (o configuración de estados),  $N$  es el orden del sistema y  $F_j$  es generalmente una función no lineal. Sea  $\mathbf{x} = [x_1, \dots, x_N]^T$  el vector de estados (o estado del sistema) y  $\mathbf{F}$  una función no lineal con argumento vectorial, la ecuación (1.1) puede expresarse como

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(\mathbf{x}). \quad (1.2)$$

Un sistema dinámico no lineal es autónomo si no tiene entradas externas; en caso contrario es no-autónomo (Haykin, 2009).

### Atractores

De acuerdo con Beer (2000), los estados de un sistema dinámico no lineal eventualmente convergen en un pequeño subconjunto del espacio de estado llamado conjunto límite. Entonces, una vez que el estado del sistema converge en un conjunto límite, la dinámica del sistema actuará para mantener el estado del sistema en dicho conjunto. En los conjuntos límite estables o atractores, todos los estados cercanos en el espacio de estado convergerán al conjunto límite. Por lo tanto, si el estado del sistema se aleja del ciclo límite debido a pequeñas perturbaciones, el estado convergería de nuevo en dicho ciclo límite. A la región

que rodea a un atractor, y donde ciertas condiciones iniciales convergen a dicho atractor, se llama dominio de atracción o cuenca de atracción. Por lo tanto, cada estado inicial del sistema está en el dominio de algún atractor. También existen conjuntos límite inestables en los cuales si el estado del sistema se aleja debido a pequeñas perturbaciones, el estado no convergería nuevamente en dicho conjunto límite.

Por ejemplo, el siguiente diagrama de fase de una red neuronal recurrente de tiempo continuo, es decir de una red neuronal de Hopfield de tiempo continuo, muestra conjuntos límite estables en azul y conjuntos límite inestables en rojo. Un punto de equilibrio (o atractor de punto fijo) es un conjunto límite el cual es un punto único en el espacio de estado que produce un comportamiento constante del sistema. Un ciclo límite es otro tipo de conjunto límite que se caracteriza por ser una trayectoria que cierra sobre sí misma, produciendo un comportamiento rítmico del sistema. El punto azul en el diagrama es un atractor de punto fijo, y la trayectoria cerrada azul es un ciclo límite. En verde se presenta un conjunto límite llamado *saddle limit set*, en el cual la trayectoria azul que converge hacia éste es estable y la trayectoria roja que se aleja es inestable. La dinámica de este punto ayuda a separar los dominios de atracción del ciclo límite (gris claro) del dominio de atracción del atractor de punto fijo (gris oscuro).

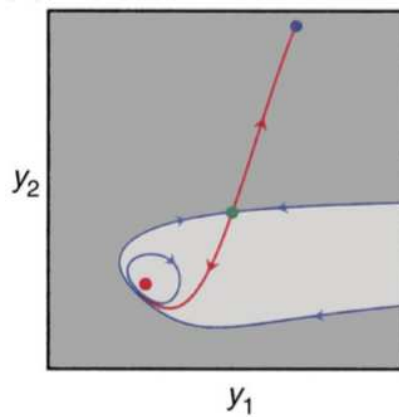


Figura 1.2: Diagrama de fase de una red neuronal de Hopfield de tiempo continuo con dos neuronas. Figura tomada de Beer (2000), pero se puede encontrar una descripción más amplia en Beer (1995).

### 1.2.1. Estabilidad en el sentido de Lyapunov

La presencia de estabilidad siempre implica alguna forma de coordinación entre las partes individuales del sistema. En el contexto de sistemas dinámicos no lineales, el término estabilidad se refiere usualmente a aquella definida por Aleksandr Lyapunov.

El teorema de Lyapunov sobre estabilidad del modelo de espacio de estado descrito por la ecuación (1.2) puede expresarse como sigue (Haykin, 2009):

**Teorema 1.** *El estado de equilibrio  $\mathbf{x}^*$  es estable si, en una pequeña vecindad*

de  $\mathbf{x}^*$ , existe una función positiva definida  $V(\mathbf{x})$  tal que su derivada con respecto al tiempo es negativa semidefinida en esa región.

Una función  $V(\mathbf{x})$ ,  $x \in \mathbb{R}^N$ , se dice que es una función de Lyapunov si existe un estado de equilibrio,  $\mathbf{x}^*$ , tal que las siguientes tres condiciones son satisfechas (Ansari & Hou, 1997):

1.  $V(\mathbf{x})$  es continua con respecto a todo los componentes de  $\mathbf{x}$ .
2.  $V(\mathbf{x})$  es positiva definida. Esto es,  $V(\mathbf{x}^*) = 0$ , y  $V(\mathbf{x}) > 0$  si  $\mathbf{x} \in \mu - \mathbf{x}^*$ , donde  $\mu$  es una pequeña vecindad alrededor de  $\mathbf{x}^*$ .
3. La derivada con respecto al tiempo,  $\dot{V}(\mathbf{x})$ , es negativa semidefinida. Esto es, la función decrece en el tiempo.

Entonces, si  $V(\mathbf{x})$  es una función de Lyapunov, el estado de equilibrio  $\mathbf{x}^*$  es estable si se cumple la condición

$$\frac{d}{dt}V(\mathbf{x}) \leq 0 \text{ para } \mathbf{x} \in \mu - \mathbf{x}^* \quad (1.3)$$

### 1.2.2. Estabilidad en el sentido de Hopfield

Una neurona es estable si su valor no cambia al seguir actualizando la red. La red de Hopfield es estable si todas las neuronas son estables. Se puede demostrar que la red de Hopfield es estable bajo la condición de simetría y sin conexiones auto-recurrentes. Bajo una dinámica asíncrona, independientemente del estado inicial, la red eventualmente alcanza un estado estable que es un mínimo de la función de energía. Se ha demostrado rigurosamente que la red es estable bajo dichas condiciones al mostrar que la función de energía es un función de Lyapunov (véase el Teorema 1).

No obstante, de acuerdo con (Liao, Wang, & Yu, 2007), la estabilidad de una red neuronal es diferente de la estabilidad en el sentido de Lyapunov. Para la estabilidad en el sentido de Lyapunov, los estados de equilibrio de un sistema dinámico son conocidos, y la función de Lyapunov tiene un signo definido en la vecindad del punto de equilibrio, mientras que la derivada de la función en las trayectorias del sistema tiene signo opuesto. Para las redes neuronales, los estado de equilibrio son desconocidos, y la función de energía puede variar de signo.

Una definición exacta de la estabilidad en una red neuronal sería la siguiente (Liao et al., 2007): el conjunto  $\Omega_E$  de estados de equilibrio  $u$  de la red neuronal se dice que es atractivo, si para cualquier  $K > 0$  y cualquier  $u_0 \in S_K := \{u, \|u\| < K\}$ , la solución de la red neuronal,  $u(t, t_0, u_0)$ , con el estado inicial  $u = u_0$ , satisface:

$$\rho\{u(t, t_0, u_0), \Omega_E\} \rightarrow 0 \text{ como } t \rightarrow \infty \quad (1.4)$$

donde  $\rho(u, \Omega_E)$  denota la distancia del punto  $u$  a  $\Omega_E$ . La red neuronal es Hopfield estable, si el conjunto de estados de equilibrio es atractivo.



## 1.3. Aplicaciones de la red de Hopfield

La red de Hopfield se ha utilizado principalmente en aprendizaje asociativo y en la solución de problemas de optimización debido a la forma en que sus estados estables pueden ser interpretados. A continuación se describen brevemente ambas aplicaciones.

### 1.3.1. Aprendizaje

La plasticidad es la habilidad de el cerebro y el sistema nervioso para cambiar estructural y funcionalmente como resultado de la interacción con el ambiente (Mouret & Tonelli, 2014). Los cambios en las sinapsis en la red de neuronas en el cerebro son llamados plasticidad sináptica o plasticidad funcional. Estos cambios son la base del aprendizaje. Es usual clasificar al aprendizaje en las siguientes tres categorías (Dayan & Abbott, 2000):

1. Aprendizaje no-supervisado: los cambios en los pesos sinápticos se realizan con base en la actividad neuronal actual. La red se auto-organiza con base en sus conexiones y su dinámica intrínseca dependiendo de la regla de plasticidad sináptica que se aplica y de las entradas presentadas durante el entrenamiento.
2. Aprendizaje supervisado: los cambios en los pesos sinápticos se realizan con base en la actividad neuronal requerida, es decir un conjunto deseado de relaciones de entrada-salida es impuesto a la red durante el entrenamiento.
3. Aprendizaje reforzado: la actividad neuronal es restringida a través de una evaluación que retroalimenta a la red. Esto es, el aprendizaje es facilitado (reforzado) ya sea mediante recompensa o castigo.

Usualmente se considera que el objetivo del aprendizaje no-supervisado es presentar a la red neuronal los datos sin procesar y dejar que la red haga su propia representación de éstos, esperando que la red almacene toda la información que se pueda considerar importante para la aplicación (Fyfe, 2005). El tipo de aprendizaje propuesto por D. O. Hebb es una forma de aprendizaje no-supervisado. Es usual simplificar la teoría del aprendizaje Hebbiano con la frase “*neurons that fire together wire together*”, es decir, las neuronas que se activan simultáneamente, refuerzan la conexión sináptica entre ellas. Las reglas de aprendizaje inspiradas por esta teoría son usadas comúnmente por las redes de Hopfield para construir una memoria de un conjunto de patrones.

Hopfield (1982) mostró como formar una memoria asociativa con este modelo de red neuronal: la red aprende un conjunto de patrones de entrenamiento al cambiar sus pesos usando una regla de aprendizaje (usualmente una regla Hebbiana). Los atractores definidos por la topología de la red neuronal corresponden a los patrones de entrenamiento, aunque existen atractores espurios que no corresponden con dichos patrones. En general, este último tipo de atractores no son deseados, pero pueden ser interpretados como una forma de clasificación o generalización de un conjunto de atractores con patrones similares en el espacio de estados.

### 1.3.2. Optimización

Un problema de optimización puede ser descrito como la selección de una solución que satisfaga los requerimientos del problema con un costo mínimo de realización. Para resolver un problema de optimización es necesario formular un modelo matemático de éste, el cual es generalmente definido por un conjunto de variables, una función objetivo que depende de dichas variables, y un conjunto de restricciones a ser satisfechas por la solución (Hérault, 2005).

Las redes de Hopfield han sido empleadas para resolver problemas de optimización. Esto se debe a que, bajo ciertas condiciones, la dinámica de la red con el tiempo resuelve más restricciones entre sus componentes de las que viola. Por lo tanto, es útil para resolver problemas de satisfacción de restricciones. Un problema de satisfacción de restricciones está compuesto de un conjunto finito de variables, cada una de las cuales está asociada con un dominio finito, y un conjunto de restricciones que restringe los valores que las variables pueden tomar simultáneamente (Tsang, 1996). Entonces, en el caso de la red de Hopfield, la evolución de la dinámica asigna valores a los estados de las neuronas para satisfacer todas o el mayor número de restricciones impuestas por las conexiones.

Hopfield y Tank (1985) mostraron como utilizar este modelo de red neuronal para resolver problemas de optimización: los pesos de la red definen un problema de satisfacciones de restricciones entre los componentes del problema representados por los estados de la red. Una función de energía es usada para determinar la estabilidad de la red, es decir, si la función existe, los estados de la red convergerán en un atractor independientemente de las condiciones iniciales. La convergencia de la red representa la forma en que se coordina la actividad de los componentes para satisfacer las restricciones. Un cambio de estado que reduce la energía resuelve más restricciones de las que viola. Los atractores son posibles soluciones de las restricciones internas de la red y mínimos de la función de energía. Así, el valor de la energía se interpreta como la cantidad de restricciones que permanecen insatisfechas, es decir mientras más restricciones insatisfechas mayor la energía.

## 1.4. Auto-modelado en Redes de Hopfield

Una red de Hopfield puede exhibir múltiples atractores y solamente una pequeña cantidad de ellos son globalmente óptimos. Por lo tanto, cuanto más compleja sea la red, más difícil será resolver las restricciones considerando que la red podría quedar atrapada en mínimos locales. Además, si la red presenta algún tipo de estructura (por ejemplo, una estructura modular), es decir no es totalmente aleatoria, los atractores en el espacio de estados formarán grupos (clusters). Posibles soluciones podrían ser formadas por la superposición de componentes independientes de varios grupos de atractores, y estas soluciones son menos probables de ser visitadas solamente actualizando la dinámica de la red.

Watson, Buckley y Mills desarrollaron un proceso para mejorar la capacidad de las redes neuronales de Hopfield de tiempo discreto para encontrar una configuración que minimice las restricciones no satisfechas, y por lo tanto la energía (Watson, Buckley, & Mills, 2011; Watson, Mills, & Buckley, 2011a, 2011b). El proceso está basado en que la red neuronal forma una memoria asociativa de sus

propios atractores – por lo tanto, “auto-modela” su dinámica previa. El proceso de auto-modelado usa el siguiente algoritmo iterativo: (1) los estados de la red son inicializados aleatoriamente, (2) los estados son actualizados para que la red converja a un atractor, y (3) una vez alcanzado el atractor, pequeños cambios en los pesos son aplicados usando aprendizaje Hebbiano.

La regla incremental de Hebb refuerza las conexiones que satisfacen las restricciones locales y debilita las conexiones que no lo hacen. El proceso de auto-modelado incrementa el tamaño del dominio de atracción del atractor más visitado debido al reforzamiento hecho por el aprendizaje Hebbiano. Adicionalmente, el aprendizaje puede causar una simple forma de generalización produciendo atractores que son nuevas combinaciones de patrones similares. Estos atractores espurios son posibles soluciones del problema de optimización original. Por lo tanto, los cambios en los pesos permite que la red puede converger en atractores que son poco probables de ser visitados o nuevos atractores formados a través del aprendizaje. Este proceso permite que los atractores compitan entre sí para encontrar una de las mejores soluciones.

## 1.5. Motivación

El proceso de auto-modelado es un procedimiento de auto-optimización para redes neuronales de Hopfield discretas que no requiere información heurística. Es un procedimiento útil dado que, de acuerdo con Vidyasagar (1998), dependiendo del problema, si una red binaria es usada para optimizar no hay garantía que un máximo global tenga un dominio de atracción no nulo. Es por esto que es importante investigar si es posible extender este procedimiento para ser utilizado en redes neuronales de Hopfield continuas, las cuales pueden presentar propiedades que incidan de manera diferentes en el proceso de auto-modelado. Además, no sería recomendable limitar la investigación bajo la condición de simetría impuestas para el algoritmo original, ya que las conexiones simétricas son muy restrictivas para ser biológicamente plausibles (Schürmann, Hollatz, & Ramacher, 1990). Por ejemplo, (Watson et al., 2011a) mencionan que las interacciones simétricas no son adecuadas para representar todos los tipos de interacciones, por ejemplo las interacciones depredador-presa, las cuales crean generalmente atractores cíclicos o caóticos. Por lo tanto, de acuerdo con Watson et al. (2011), es necesario investigar los efectos de las conexiones asimétricas. En este trabajo entonces se presenta una investigación sobre los efectos de relajar las restricciones de la red de Hopfield como sistema dinámico no lineal autónomo, es decir, cuando se usan conexiones asimétricas y auto-recurrentes, una función de activación continua, y una dinámica temporal continua en una red neuronal de Hopfield sin entradas externas.

El proceso de auto-modelado no se restringe a redes neuronales, sino que se ha aplicado a otros dominios como las redes sociales (Froese, Gershenson, & Manzanilla, 2014; Watson et al., 2011a) y las redes de regulación de genes (Watson, Buckley, Mills, & Davies, 2010). Además, existen áreas potenciales de aplicación como la evolución de la evolutividad (Watson & Szathmáry, 2016) o los ciclos rituales (Froese, 2013), por mencionar algunos. Este trabajo es el primer paso para posteriormente explorar las dinámicas que emergen del proceso cuando se utiliza en una red neuronal de Hopfield, con restricciones en las conexiones y sin éstas, como sistema dinámico no lineal no-autónomo, es de-

cir, con entredas externas. No obstante, la investigación actual del proceso de auto-modelado ha tenido lugar en redes artificiales aisladas, lo cual va en contra de décadas de investigación en robótica situada (Cliff, 1991). Por ejemplo, Woodward, Froese, y Ikegami (2015) mencionan que sería interesante dar una interpretación más conductual de la optimalidad de la configuración de estados al integrar la red neuronal en un robot móvil el cual requiera realizar una tarea. Es decir, desde un punto de vista dinámico, es importante analizar el efecto del proceso de auto-modelado cuando existe una interacción no lineal entre red, cuerpo y ambiente (Beer, 2000).

## 1.6. Objetivos

### 1.6.1. Objetivo General

Extender el proceso de auto-modelado para ser usado en redes neuronales de Hopfield de tiempo continuo y estado continuo con conexiones asimétricas y auto-recurrentes.

### 1.6.2. Objetivos Específicos

El objetivo general se cumplirá siguiendo los siguientes objetivos particulares:

1. Comprobar estadísticamente que el proceso de auto-modelado en redes de Hopfield de tiempo discreto y estado discreto funciona con conexiones asimétricas y auto-recurrentes.
2. Extender el proceso de auto-modelado en redes de Hopfield de tiempo discreto y función de activación continua con ganancia unitaria y sesgo nulo.
3. Comprobar estadísticamente que el proceso de auto-modelado en redes de Hopfield de tiempo discreto y estado continuo funciona con conexiones asimétricas y auto-recurrentes.
4. Analizar los resultados con el fin de entender qué fenómenos surgen al usar una función de activación continua.
5. Extender el proceso de auto-modelado en redes de Hopfield de tiempo continuo y función de activación continua donde los términos de ganancia y sesgo incidan en la dinámica de la red.
6. Comprobar estadísticamente que el proceso de auto-modelado en redes de Hopfield de tiempo continuo y estado continuo funciona con conexiones asimétricas y auto-recurrentes.
7. Analizar los resultados con el fin de entender qué fenómenos surgen al utilizar una dinámica temporal continua.

## 1.7. Estructura del trabajo

En el siguiente capítulo se presentan los modelos de la red neuronal de Hopfield en los cuales será aplicado el proceso de auto-modelado. En el Capítulo 3 se introducen los antecedentes necesarios relacionados con la formación de memoria asociativa y la resolución de problemas de optimización utilizando redes neuronales de Hopfield.

En el Capítulo 4 se describe el estado del arte del proceso de auto-modelado en redes neuronales de Hopfield de tiempo-discreto y estado discreto, así como se explica el proceso de auto-modelado en redes de Hopfield de tiempo discreto y estado continuo, y el proceso de auto-modelado en redes de Hopfield de tiempo continuo y estado continuo.

Los resultados del proceso de auto-modelado para los tres modelos de red neuronal de Hopfield son mostrados en los Capítulos 5, 6, y 7 utilizando gráficas y una prueba estadística. En cada uno de estos capítulos se analizan los resultados. Finalmente, las conclusiones y el trabajo futuro son presentados en el Capítulo 8.

## Capítulo 2

# Red Neuronal de Hopfield

Los modelos de la red neuronal de Hopfield tienen una dinámica determinista descrita por un conjunto de ecuaciones diferenciales no lineales que definen de manera exacta la evolución de los estados como función del tiempo. A continuación se presentan esas ecuaciones, así como las funciones de energía de cada modelo.

### 2.1. Red neuronal de Hopfield de tiempo discreto y estado discreto

Una red de Hopfield de tiempo discreto y estado discreto es una red completamente conectada, usualmente con una matriz de conexión simétrica y sin conexiones auto-recurrentes. La red consiste en  $N$  estados discretos  $s_i$ , con  $s_i \in \{-1, 1\}$ . Los estados son actualizados asincrónicamente de acuerdo con la siguiente ecuación (Watson et al., 2011):

$$s_i(t+1) = \theta_{HTF} \left[ \sum_j^N \omega_{ij} s_j(t) \right] \quad (2.1)$$

donde  $\omega_{ij}$  es el peso entre la neurona  $i$  y  $j$ , y  $\theta_{HTF}$  es una función umbral de Heaviside. Así, si  $x > 0$  entonces  $\theta_{HTF}(x) = 1$ ; de otro modo  $\theta_{HTF} = -1$ .

Dada la condición de simetría y de no auto-recurrencia en la matriz de conexiones, es decir  $\omega_{ij} = \omega_{ji}$  y  $\omega_{ii} = 0$ , Hopfield (1982) demostró la existencia de la función de energía mostrada en la ecuación (2.2), donde  $H_s(t)$  representa la configuración de estados en  $t$ .

$$E(H_s(t), \Omega(t)) = -\frac{1}{2} \sum_{ij}^N \omega_{ij} s_i s_j \quad (2.2)$$

### 2.2. Red neuronal de Hopfield de tiempo discreto y estado continuo

Una red de Hopfield de tiempo discreto y estado continuo es una red completamente conectada, usualmente con una matriz de conexión simétrica y con

conexiones auto-recurrentes no negativas. La red consiste en  $N$  estados discretos  $s_i$ , con  $s_i \in [-1, 1]$ . Los estados son actualizados asincrónicamente de acuerdo con la siguiente ecuación (Koiran, 1994):

$$s_i(t+1) = \sigma \left[ \sum_j^N \omega_{ij} s_j(t) \right] \quad (2.3)$$

donde  $\omega_{ij}$  es el peso entre la neurona  $i$  y  $j$ , y  $\sigma$  es una función continua. De acuerdo con Koiran (1994),  $\sigma$  debe ser una función continua estrictamente creciente en un intervalo  $\alpha < \beta$  con  $\alpha = -\infty$ ,  $\beta = +\infty$ , y  $\lim_{x \rightarrow \pm\infty} \sigma = \pm 1$ . Estas condiciones son cumplidas por la siguiente función sigmoide:

$$\sigma(x) = \frac{2}{1 + e^{-x}} - 1 \quad (2.4)$$

Dada la condición de simetría y de no auto-recurrencia en la matriz de conexiones, es decir  $\omega_{ij} = \omega_{ji}$  y  $\omega_{ii} = 0$ , Koiran (1994) demostró la existencia de la función de energía mostrada en la ecuación (2.5), donde  $H_s(t)$  representa la configuración de estados en  $t$ .

$$E(H_s(t), \Omega(t)) = -\frac{1}{2} \sum_{ij}^N \omega_{ij} s_i s_j + \sum_i^N \int_0^{s_i(t)} \sigma^{-1}(\xi) d\xi. \quad (2.5)$$

### 2.3. Red neuronal de Hopfield de tiempo continuo y estado continuo

Una red neuronal de Hopfield de tiempo continuo y estado continuo es una red completamente conectada, usualmente con una matriz de conexión simétrica y con conexiones auto-recurrentes. La red consiste en  $N$  estado continuos  $s_i$ . Los estados son actualizados asincrónicamente de acuerdo con la siguiente ecuación (Beer, 1995):

$$\tau_i \dot{s}_i = -s_i + \sum_{j=1}^N \omega_{ij} \sigma(g_j(s_j + \theta_j)) \quad (2.6)$$

donde  $\omega_{ij}$  es el peso entre la neurona  $i$  y  $j$ ,  $g_i$  es la ganancia de la neurona  $i$ ,  $\theta_i$  es el sesgo de la neurona  $i$  y  $\sigma$  es la siguiente función sigmoide (Hoinville, Siles, & Hénaff, 2011):

$$\sigma_i(x) = \frac{2}{1 + e^x} - 1. \quad (2.7)$$

Dada la condición de simetría y de no auto-recurrencia en la matriz de conexiones, es decir  $\omega_{ij} = \omega_{ji}$  y  $\omega_{ii} = 0$ , Hopfield (1984) demostró la existencia de la función de energía mostrada en la ecuación (2.8), donde  $V_j = \sigma(g_j(s_j + \theta_j))$  es la salida de la neurona  $j$  y  $H_V(t)$  representa la configuración de estados en  $t$  ( $V = (V_1, \dots, V_N)$  es usualmente llamado el vector de estados (Berg, 1996)).

$$E(H_V(t), \Omega(t)) = -\frac{1}{2} \sum_{ij}^N \omega_{ij} V_i(t) V_j(t) + \sum_i^N \frac{1}{\tau_i} \int_0^{V_i(t)} \sigma^{-1}(\xi) d\xi \quad (2.8)$$

## Capítulo 3

# Memoria Asociativa y Optimización en Redes Neuronales de Hopfield

Es común que la red neuronal de Hopfield sea interpretada como una memoria asociativa o sea aplicada a la solución de problemas de optimización. A continuación se presenta los antecedentes necesarios y una descripción general de estas aplicaciones.

### 3.1. Aprendizaje Hebbiano

Donald O. Hebb fue un psicólogo canadiense quien presentó una teoría sobre la manera en la que los estímulos pueden ser representados en el cerebro. De acuerdo con el profesor Richard Brown de la Universidad de Brown, Hebb incluyó un análisis del mecanismo de aprendizaje neuronal subyacente al condicionamiento Pavloviano en su tesis de maestría donde concluyó que (Milner, 2003):

“Una neurona excitada tiende a disminuir su descarga a las neuronas inactivas, y a aumentar esta descarga a cualquier neurona activa, y por lo tanto a formar una ruta hacia ella, ya sea que haya neuronas intermedias entre los dos o no. Con la repetición, esta tendencia es prepotente en la formación de las rutas neuronales.”

Más tarde, Hebb retomó esta idea y el siguiente postulado neurológico para explicar el aprendizaje de las representaciones visuales en el cerebro fue presentado en el libro ‘La organización del comportamiento’ (Hebb, 1949):

“Cuando el axón de una célula A está lo suficientemente cerca como para excitar a una célula B y repetidamente toma parte en la activación, ocurren procesos de crecimiento o cambios metabólicos en una o ambas células de manera que tanto la eficiencia de la célula A, como la capacidad de excitación de la célula B, son aumentadas.”



### 3.1.1. Regla de Hebb

La base del aprendizaje Hebbiano se puede expresar matemáticamente a través de la llamada regla de Hebb. Esta regla expresa la tasa de cambio de la conexión sináptica con eficacia  $\omega_{ij}$  de la siguiente forma (Gerstner, 2016):

$$\frac{d\omega_{ij}}{dt} = F(\omega_{ij}; v_{post}v_{pre}) \quad (3.1)$$

donde  $F$  es una función de la eficacia, de la actividad de la neurona pre-sináptica  $v_{pre}$  y de la actividad de la neurona post-sináptica  $V_{post}$ . Esta regla no depende de más variables para los modelos de redes neuronales basados en la tasa de actividad neuronal (por ejemplo, modelos de masa neuronal como los empleados en este trabajo). Kuriscak, Marsalek, Stroffek, y Toth (2015) resumen las propiedades más importantes de la regla de Hebb para su implementación:

1. Localidad: la regla se restringe a la entrada y salida de la neuronas de la sinapsis que se actualiza.
2. Cooperatividad: se requiere la actividad simultánea de ambas neuronas.
3. Limitación de los valores de peso: ésto evita su divergencia.
4. Competencia: algunas sinapsis se fortalecen a expensas de otras que se debilitan.
5. Estabilidad a largo plazo: necesaria para la estabilidad dinámica de la red neuronal.
6. Facilitación y depresión sináptica: es decir, el aumento y la disminución de peso.

Una de las aplicaciones más importantes de esta regla ha sido en el desarrollo de algoritmos para el almacenamiento asociativo en redes neuronales de Hopfield. Esto es, la red de Hopfield como memoria asociativa o memoria de contenido direccionable (CAM por sus siglas en inglés).

## 3.2. Memoria Asociativa

La red de Hopfield como memoria asociativa opera en dos fases: almacenamiento y recuperación. En la fase de almacenamiento, un conjunto de patrones, llamados memorias fundamentales, son aprendidos por la red a través de una regla de aprendizaje, usualmente Hebbiano. En la fase de recuperación, un patrón de prueba, usualmente una versión incompleta o con ruido, es impuesto como el estado de la red. Los estados de la red son entonces actualizados de acuerdo a la dinámica de la red hasta que ésta alcance el estado estable. Idealmente, este estado será el patrón almacenado que se deseaba recuperar aunque en ocasiones la red puede converger a estados espurios.

Hopfield (1982) mostró que las ‘propiedades colectivas del modelo’ de una red neuronal de tiempo discreto y estado discreto pueden ser utilizadas como una memoria asociativa. Posteriormente, Hopfield (1984) demostró que las redes neuronales de tiempo continuo y ‘respuesta gradual’ son capaces de operar también como una memoria de contenido direccionable.

La regla de Hebb utilizada por Hopfield (1982) al presentar su modelo discreto como una memoria asociativa fue la siguiente:

$$\omega_{ij} = \sum_{s=1}^M (2V_i^s - 1)(2V_j^s - 1) \quad (3.2)$$

donde  $V^s$ ,  $s = 1, \dots, M$  son los patrones a almacenar. Posteriormente, Dong y Hopfield (1992) presentaron la regla de Hebb continua de la ecuación (3.3) que fue utilizada con el modelo continuo de la red de Hopfield.

$$B_{ij} \frac{d\omega_{ij}}{dt} = \omega_{ij} + V_i V_j \quad (3.3)$$

donde  $B_{ij}$  es una constante de tiempo y  $V_i$  es la salida de la neurona  $i$ .

Las reglas de Hebb utilizados en la literatura son diversas dependiendo del modelo de la red y de la aplicación del aprendizaje. Por ejemplo, sea el conjunto de  $M$  patrones  $\{\mathbf{x}_p\}$ , donde  $\mathbf{x}_p = [x_{p,1}, \dots, x_{p,N}]^T$ . La regla de Hebb generalizada, también conocida como la regla del producto externo de almacenamiento, para almacenar patrones bipolares  $x_{p,i} = \pm 1$  en una red Hopfield de tiempo discreto y estado discreto es la siguiente (H. Wang, Wu, Zhang, & Du, 2011):

$$\omega_{ij} = \frac{1}{N} \sum_{p=1}^M x_{p,i} x_{p,j} \quad (3.4)$$

para todo  $i \neq j$ , donde  $N$  es la dimensión de los patrones (el número de neuronas de la red). Esta regla se puede expresar de manera incremental:

$$\omega_{ij}(t) = \omega_{ij}(t-1) + \delta x_{p,i} x_{p,j} \quad (3.5)$$

para todo  $i \neq j$ , donde  $\omega_{ij}(0) = 0$  y  $\delta$  es la tasa de aprendizaje. Esta última forma de la regla de Hebb no está limitada a patrones ortogonales discretos, ya que se puede utilizar con patrones donde  $x_{p,i} \in \mathbb{R}$  (Hoinville et al., 2011). El uso de estados bipolares, en lugar de estados binarios, es frecuente tanto en problemas de optimización (Šíma, Orponen, & AnttiPoika, 1999) como en el aprendizaje asociativo (H. Wang et al., 2011). Esto se debe a que la red presenta una mejor capacidad de almacenamiento, y los patrones bipolares tienen una mayor probabilidad de ser ortogonales. No obstante, existen varios trabajos donde se utilizan estados binarios. Sin embargo, el aprendizaje Hebbiano es generalmente inestable si solamente especifica un incremento. La mayoría de los modelos de aprendizaje en redes atractor <sup>1</sup> incluye una constante de decaimiento para lograr un balance entre el incremento y el decremento de la fuerza sináptica (Hua, Houk, & Mussa-Ivaldi, 1999). En el ejemplo anterior, la regla Hebbiana de la ecuación (3.5) considera de manera implícita el decaimiento en la fuerza de las conexiones. También es común que exista un límite máximo en el crecimiento de los pesos. El aprendizaje Hebbiano puede resultar ser inestable debido a un crecimiento no saturado. La red podría presentar problemas como el de interferencia catastrófica (Srivastava, Sampath, & Parker, 2014), en el cual olvida todos los patrones almacenados.

Vico y Jerez (2003) resumen correctamente las propiedades computacionales de una memoria asociativa formada a través de una regla incremental Hebbiana:

---

<sup>1</sup>Del inglés *attractor networks*

1. Un proceso de generalización tiene lugar cuando se presentan patrones pertenecientes a la misma clase (versiones con ruido o incompletos), activan el mismo atractor.
2. Las clases de patrones con correlación baja o nula se representan con diferentes atractores, provocando la reorganización de la matriz sináptica.
3. Las nuevas clases de patrones que se correlacionan con las clases previamente almacenadas implican un proceso competitivo-cooperativo entre atractores almacenados y nuevos conjuntos no estructurados de neuronas de red, resultando en la formación de nuevos atractores, mientras que las viejas representaciones son interferidas hasta cierto punto.

### 3.3. Optimización

Hopfield y Tank (1985) aplicaron por primera vez una red neuronal de Hopfield de tiempo continuo y estado continuo en la solución del problema de optimización combinatoria conocido como problema del agente viajero (TSP por sus siglas en inglés). La red de Hopfield es comúnmente utilizada para resolver problemas de optimización cuya solución pueda ser expresada por un conjunto de variables, las cuales pueden tomar uno de dos valores discretos,  $\{0, 1\}$  o  $\{-1, 1\}$ . Por su parte, las neuronas de la red deben tener una función de activación discreta (donde los estados sean elementos de  $\{0, 1\}$  o  $\{-1, 1\}$ ), o una función de activación continua (donde los estados sean elementos de  $[-1, 1]$ ) a condición de que el estado de equilibrio sea alcanzado en uno de los extremos del intervalo (Joya, Atencia, & Sandoval, 1997). Además, es común que la red tenga conexiones simétricas sin conexiones auto-recurrentes.

La función objetivo de un problema de optimización es generalmente expresada como la minimización del costo de la solución. Estos problemas pueden ser representados en una red de Hopfield cuyos mínimos de su función de energía son los mismos que los de la función objetivo. Estos mínimos de la función de energía con los estado estables alcanzados al actualizar la dinámica de la red, y por tanto soluciones al problema. Looi (1992) indica los siguientes pasos para mapear un problema en una red neuronal que utilice una función de energía:

- a. Elegir un esquema de representación el cual permita a la salida de las neuronas ser decodificada en la solución del problema.
- b. Derivar una función de energía cuyo valor mínimo corresponda a la “mejor” solución del problema a ser mapeado.
- c. Derivar de la función de energía las conexiones y el sesgo de entrada de la red.
- d. Establecer los valores iniciales para la entrada a las neuronas la cual determine completamente la salida estable de las neuronas de la red.

Una función de energía para optimización combinatoria es usualmente de la forma (Looi, 1992):

$$E = \sum_i A_i(\text{“violación de la restricción } i\text{”}) + B(\text{“costo”}) \quad (3.6)$$

donde  $A_i > 0$ ,  $B > 0$  y “costo” es una función de costo independiente de la violación de restricciones. Al minimizar la función de energía  $E$  se intenta

minimizar el costo mientras que al mismo tiempo se maximiza la satisfacción de las restricciones.

Joya et al. (1997) señalaron los principales errores y problemas al aplicar las redes neuronales de Hopfield a la solución de problemas de optimización:

- a. Incoherencia entre la dinámica de la red y la función de energía asociada a la red. Una situación común es usar una red neuronal con dinámica continua y asociarle una función de energía correspondiente a una red neuronal discreta.
- b. Error debido a la discretización de las ecuaciones dinámicas continuas causado por simulación en una computadora digital. Así, mientras más grande es el paso de simulación, más diferente es el comportamiento real y teórico de la red.
- c. La función de energía de la red tiene muchos mínimos locales, por lo que es probable que la red alcance un estado de equilibrio que no corresponda a una solución aceptable del problema.

## Capítulo 4

# Auto-modelado en Redes Neuronales de Hopfield

El proceso de auto-modelado nos ayuda a comprender mejor las condiciones bajo las cuales un proceso de auto-organización local puede mejorar la solución global de las restricciones entre componentes de un sistema adaptativo complejo (Watson et al., 2011). Por lo tanto, a continuación se presenta el procedimiento y las condiciones para los tres modelos descritos en el Capítulo 2, así como el estado del arte del proceso de auto-modelado.

### 4.1. Auto-modelado en redes de Hopfield de tiempo discreto y estado discreto

Watson et al. (2011) propusieron el siguiente algoritmo para el proceso de auto-modelado: (1) la configuración de estados es aleatorizada tal que  $R = \{-1|1\}^N$ ; (2) se permite que la red converja de la configuración aleatoria a un atractor; y (3) la red refuerza el atractor alcanzado utilizando aprendizaje Hebbiano. El proceso de convergencia a un atractor es llamado el periodo de relajación de la red y dura  $\tau$  pasos de tiempo. Suponiendo que durante el período de relajación la red esta la mayor parte del tiempo en un atractor, el proceso de aprendizaje para reforzar el atractor alcanzado puede realizarse

- (a) al final de periodo de relajación de acuerdo a la siguiente regla de aprendizaje:

$$\omega_{ij}(t+1) = \theta_{LTF} [\omega_{ij}(t) + \delta s_i(t)s_j(t)]; \quad (4.1)$$

- (b) o en cada paso de tiempo de acuerdo a la siguiente regla de aprendizaje (Watson et al., 2011b):

$$\omega_{ij}(t+1) = \theta_{LTF} \left[ \omega_{ij}(t) + \frac{\delta}{\tau} s_i(t)s_j(t) \right]. \quad (4.2)$$

Ambas reglas se aplican para todo  $\omega_{ij}$ , con  $i \neq j$ , donde  $\delta$  es la tasa de aprendizaje y  $\theta_{LTF}$  es una función umbral lineal. Así, si  $x > 0$  entonces  $\theta_{LTF}(x) = 1$ ;

si  $x < 0$  entonces  $\theta_{LTF}(x) = -1$ ; de otro modo  $\theta_{LTF}(x) = x$ . El efecto global de ambas formas de aprendizaje en la red neuronal es el mismo. Típicamente, el proceso iterativo continúa hasta que la red converge en un solo atractor desde cualquier configuración inicial.

La energía original,  $E^O$ , se utiliza para calcular el grado en el que una configuración de estados obtenida por el proceso de auto-modelado resuelve con éxito las restricciones originales. Las restricciones originales son satisfechas si  $\omega_{ij}(t=0)s_i s_j > 0$ . Esta energía se calcula con los pesos originales de la red y con los estados obtenidos con los pesos modificados, es decir

$$E^O(H(t), \Omega(t=0)) = - \sum_{ij}^N \alpha_{ij} s_i(t) s_j(t) \quad (4.3)$$

donde  $\alpha_{ij} \equiv \omega_{ij}(t=0)$  y  $H(t) = \langle s_1, \dots, s_N \rangle \in \{-1, 1\}^n$ . Si el proceso funciona correctamente, la energía de los atractores alcanzados será menor con el tiempo, es decir, los atractores alcanzados durante el proceso serán de menor energía, no porque algún atractor sea diferente de los originalmente definidos por la topología de la red, sino porque la distribución de atractores visitados es diferente.

De acuerdo con Watson et al. (2011), existen tres condiciones para que el proceso funcione: (C1) la dinámica inicial del sistema necesita exhibir múltiples atractores (C2) la configuraciones del sistema son repetidamente relajadas desde diferentes configuraciones iniciales tal que el sistema converja en muchos atractores diferentes en una escala de tiempo donde las conexiones cambien lentamente, y (C3) el sistema pasa la mayor parte de su tiempo en los atractores. También existen dos requisitos prácticos: (R1) la tasa de aprendizaje debe ser pequeña para que un gran número de atractores sean alcanzados y mínimos locales pobres no sean reforzados, y (R2) el tiempo para que el sistema converja en un atractor debe ser mucho menor que  $\tau$  para que la red pueda alcanzar consistentemente estados estables de atracción.

#### 4.1.1. ¿Por qué funciona el proceso de auto-modelado?

El proceso de auto-modelado se basa en dos propiedades de la red neuronal de Hopfield. Primero, existe una correlación positiva entre el ancho y la profundidad del dominio de atracción (Kryzhanovsky & Kryzhanovsky, 2008), por lo que cuanto más ancho el dominio de atracción, más probable es encontrar el atractor y por lo tanto mejor será la solución a las restricciones. En segundo lugar, la memoria asociativa permite generalizar al reforzar los atractores aprendidos por la red a través del aprendizaje Hebbiano, y por consiguiente aumentar su dominio de atracción. Esto es, el refuerzo de un mínimo local, al mismo tiempo refuerza mejores mínimos de energía dado que los sub-patronos que son comunes a muchos óptimos locales pueden ser comunes a óptimos superiores. Entonces, los mejores atractores se refuerzan con mayor frecuencia, incluso si algunos de ellos no han sido visitados previamente (Watson et al., 2011). Por lo tanto, la red aumentará con gran probabilidad un atractor globalmente óptimo ya que cuanto mejor sea el atractor, existe una mayor probabilidad de que sea visitado durante las relajaciones de la red, más se ampliará el dominio de atracción debido al aprendizaje, más será visitado durante el proceso, y así sucesivamente.

Watson et al. (2011a) explican que el reforzamiento de los atractores aprendidos por la red provoca la ampliación de la cuenca de atracción de atractores de baja energía a expensas de los atractores de alta energía a través dos formas de generalización:

- a. Generalización por ‘idealización’ o eliminación de ruido. En redes sin estructura, el óptimo global no es visitado mediante la relajación de la red, pero muchos óptimos locales en la vecindad del óptimo global son visitados. El aprendizaje provoca que la red generalice con base en el promedio de los patrones reforzados.
- b. Generalización por atractores espurios. En redes con estructura, el aprendizaje asociativo provoca que la red genere nuevos patrones a partir de nuevas combinaciones de las características comunes de los patrones reforzados. Estos atractores espurios pueden ser óptimos superiores. Entonces, al reforzar óptimos locales se refuerzan los atractores espurios con los que tengan sub-patrones en común, aún cuando la red no haya convergido en ellos anteriormente.

Por lo tanto, a medida que las conexiones cambian, el aprendizaje Hebbiano acelera la competencia entre atractores incompatibles. El resultado del proceso de auto-modelado es que los atractores globalmente óptimos superen a los atractores localmente óptimos, finalizando idealmente con un sólo atractor en el espacio de estados, el cual es un atractor de baja energía que satisface la mayor cantidad de restricciones posibles del problema de optimización representado por la topología de la red.

#### 4.1.2. Estado del arte

Watson, Buckley, y Mills (2009) presentaron en un informe técnico los primeros resultados del proceso de auto-modelado en una red neuronal de Hopfield con matriz de conexión simétrica y con conexiones autorecurrentes con valor igual a la unidad. Después, Watson et al. (2011) mostraron los resultados del proceso de auto-modelado en redes neuronales de Hopfield con una matriz de conexión simétrica y sin conexiones auto-recurrentes. Con base en este trabajo Watson et al. (2011a) publicaron un modelo basado en agentes para explicar como la auto-organización de agentes egoístas puede mejorar la adaptación global. Los agentes pueden modificar como son afectados por otros agentes para maximizar su utilidad individual. La utilidad individual es la suma ponderada de las interacciones entre el estado que adopta el agente y los estados de otros agentes en la red. Los cambios en las interacciones son continuos, es decir, pueden cambiar la fuerza y el signo de la conexión. La relación inter-agentes se alteran necesariamente de una manera homóloga con el aprendizaje Hebbiano. La utilidad total es calculada como la suma de las utilidades individuales. El resultado es la maximización de las utilidades individuales y por tanto de la utilidad global. En un trabajo posterior, Watson et al. (2011b) expusieron la diferencia entre el proceso de auto-modelado mencionado anteriormente y otro en el que las conexiones aprendidas se utilizan para crear clusters de variables de estado que una vez conectadas no pueden variar de manera independiente. Estos módulos cambian subconjuntos de variables simultáneamente, y estos cambios de estados múltiples serán evaluados con base en el beneficio que tienen en combinación, incluso si cada cambio en sus elementos es individualmente perjudicial.

En los artículos anteriormente descritos, el proceso de auto-modelado fue probado en matrices de conexiones con diferentes tipos de restricciones. Las diferentes restricciones son resumidas en la Tabla 4.1, donde  $N$  es el número de nodos.

Trabajo	Restricción	Definición ( $\alpha_{ij} \equiv \omega_{ij}(t=0)$ )
Watson et al. (2009)	Espacial	$\alpha_{ij} \in (0, e^{-d})$ , donde $d = \text{mod}( i-j , N)$ .
	Modular	$\alpha_{ij} = 1$ si $\lfloor \frac{i}{k} \rfloor = \lfloor \frac{j}{k} \rfloor$ ( $i \neq j, k = 5$ ); $\alpha_{ij} = p$ de otro modo.
Watson et al. (2011)	Aleatoria	$\alpha_{ij} \in \{-1, 1\}$ , y los valores se eligen con igual probabilidad.
	Modular	$ \alpha_{ij}  = 1$ si $\lfloor \frac{i}{k} \rfloor = \lfloor \frac{j}{k} \rfloor$ ( $i \neq j, k = 5$ ); $\alpha_{ij} = 0.01$ de otro modo.
Watson et al. (2011a)	Dispersa Aleatoria	$\alpha_{ij} \in \{-1, 0, 1\}$ , la densidad de no ceros es $\frac{k}{N}$ , con $k = 8$ , y se eligen con igual probabilidad.
	Modular	$ \alpha_{ij}  = 1$ si $\lfloor \frac{i}{k} \rfloor = \lfloor \frac{j}{k} \rfloor$ ( $i \neq j, k = 5$ ); $\alpha_{ij} = 0.01$ de otro modo.
Watson et al. (2011b)	Aleatoria (RA)	$\alpha_{ij} = 1$ si $i = j$ ; de otro modo $\alpha \in \{-p, p\}$ con igual probabilidad.
	Modular	$\alpha_{ij} = R \left( \lfloor \frac{i}{k} \rfloor, \lfloor \frac{j}{k} \rfloor \right)$ donde $R$ es una matriz RA.

Tabla 4.1: Tipos de restricciones en los artículos sobre auto-modelado de redes de Hopfield de tiempo discreto y estado discreto.

En trabajos relacionados anteriores se utilizaron otro tipo de funciones de activación para el proceso de auto-modelado. Watson et al. (2010) presentaron una red de regulación de genes usando una versión discretizada del modelo de red de Hopfield de tiempo continuo. Por su parte, Woodward et al. (2015) emularon una red neuronal de Hopfield usando una red neuronal de impulsos<sup>1</sup>. Una comparación de las funciones de activación, y el dominio de los estados y pesos utilizados en estos trabajos y en el trabajo original sobre auto-modelado se presentan en la Tabla 4.2.

Trabajo	Función de Activación	Estados	Pesos ( $\omega_{ij}$ )
Watson et al. (2011)	Función umbral de Heaviside	$\{-1, 1\}$	$-1 \leq \omega_{ij} \leq 1$
Watson et al. (2010)	Sigmoide ( $\tanh(x/10)$ )	$(-1, 1)$	$\omega_{ij}$ no limitado
Woodward et al. (2015)	Función lineal saturada	$[0, 1]$	$-1 \leq \omega_{ij} \leq 1$

Tabla 4.2: Tipos de funciones de activación usada en trabajos relacionados.

## 4.2. Auto-modelado en redes de Hopfield de tiempo discreto y estado continuo

Esta es una primera generalización de la red de Hopfield en la cual se utiliza una función de activación continua. Para este tipo de red se utiliza el siguiente procedimiento: (1) la configuración de estados es aleatorizada tal que

<sup>1</sup>Del inglés *spiking neural network*



$R = [-1, 1]^N$ ; (2) la red se relaja, durante  $\tau$  pasos de tiempo de la configuración aleatoria a un atractor (periodo de relajación); y (3) los pesos se actualizan al final de cada periodo de relajación de acuerdo a la siguiente regla de Hebb:

$$\omega_{ij}(t+1) = \theta_{LTF} [\omega_{ij}(t) + \delta s_i^r(t) s_j^r(t)] \quad (4.4)$$

donde  $\delta$  es la tasa de aprendizaje,  $\theta_{LTF}$  es la función umbral lineal definida anteriormente y  $s_i^r(t)$  representa a los atractores visitados durante el proceso, los cuales no son miembros del conjunto original definido por la topología original de la red. Esta regla de aprendizaje se aplica para todo  $\omega_{ij}$ , incluyendo las conexiones auto-recurrentes. Típicamente, el proceso iterativo continúa hasta que la red converge en un solo atractor desde cualquier configuración inicial.

La energía original,  $E^O$ , es calculada como antes con los pesos originales de la red, y con los estados de la red obtenidos con los pesos modificados, es decir

$$E^O(H(t), \Omega(t=0)) = -\frac{1}{2} \sum_{ij} \alpha_{ij} s_i(t) s_j(t) + \sum_i \int_0^{s_i(t)} \sigma^{-1}(\xi) d\xi. \quad (4.5)$$

donde  $\alpha_{ij} \equiv \omega_{ij}(t=0)$ , y  $H^r(t) = \langle s_1^r, \dots, s_N^r \rangle \in [-1, 1]^N$ . Finalmente, las tres condiciones, (C1), (C2) y (C3), así como los dos requerimientos prácticos, (R1) y (R2), explicados anteriormente son necesarios para que el proceso de auto-modelado funcione.

### 4.3. Auto-modelado en redes de Hopfield de tiempo continuo y estado continuo

La red de Hopfield de tiempo continuo generaliza al modelo de Hopfield utilizado en el proceso original de auto-modelado, tanto al utilizar una función de activación continua con términos de sesgo y de ganancia, como al utilizar un modelo de ecuaciones diferenciales para actualizar los estados. El procedimiento que se sigue es el siguiente: (1) la configuración de estados es aleatorizada tal que  $R = [-15, 15]^N$ ; (2) la red se relaja, durante  $\tau$  pasos de tiempo, con  $\tau_i$  como paso de integración, de la configuración aleatoria a un atractor (periodo de relajación); y (3) los pesos se actualizan al final de cada periodo de relajación de acuerdo a la siguiente regla de aprendizaje:

$$\omega_{ij}(t+1) = \theta_{LTF} [\omega_{ij}(t) + \delta V_i^r(t) V_j^r(t)] \quad (4.6)$$

donde  $\delta$  es la tasa de aprendizaje,  $\theta_{LTF}$  es la función umbral lineal definida anteriormente y  $V_i^r(t) = \sigma(g_i(s_i^r(t) + \theta_i))$  representa las salidas de las configuraciones de estados (atractores) visitados durante el proceso. Los atractores visitados no son miembros del conjunto original definido por la topología original de la red. Esta regla de aprendizaje se aplica para todo  $\omega_{ij}$ , incluyendo las conexiones auto-recurrentes.

La ecuación (2.6) es integrada a través del método explícito de Euler utilizando la siguiente ecuación discretizada:

$$s_i(t + \Delta t) = s_i(t) + \frac{\Delta t}{\tau_i} \left( -s_i(t) + \sum_{j=1}^N \omega_{ij} \sigma(g_j(s_j(t) + \theta_j)) \right) \quad (4.7)$$

donde  $\Delta t$  es el paso de integración. El método de integración es numericamente estable cuando  $\Delta t$  es menor que dos veces la constante de tiempo más pequeña en la red (Blynel & Floreano, 2002). No obstante, si existen conexiones auto-recurrentes el método de Euler puede ocasionalmente incrementar el valor de la energía (Feng & Douligeris, 2001).

En este trabajo se desea probar el proceso de auto-modelado de manera general utilizando el término de ganancia,  $g_i$ , y de sesgo,  $\theta_i$ . Por lo tanto,  $g_i$  puede tomar valores aleatorios positivos y  $\theta_i$  es calculado de la siguiente forma (Golos, Jirsa, & Daucé, 2016):

$$\theta_i = -\frac{1}{2} \sum_{j=1}^N \omega_{ij}. \quad (4.8)$$

Ya que la estabilidad de los atractores decrece con el número de patrones guardados en la red (Amit, Gutfreund, & Sompolinsky, 1985), Golos et al. (2016) demostraron la existencia de múltiples atractores de punto fijo para el modelo de Hopfield de tiempo continuo, en condición de alta ganancia, usando una matriz de conexiones normalizada basada en un conectoma de 998 nodos simétrico y de pesos positivos. Los autores usaron la ecuación  $\tau_x \frac{dx_i}{dt} = -x_i + \sum_{j=1}^N W_{ij} A_j$  con  $A_j = \frac{1}{2}(1 + \tanh(G(Px_i - \theta_i)))$ , donde  $x_i$  es el potencial del nodo,  $A_i$  la salida del nodo,  $W_{ij}$  la matriz de conexiones,  $\theta_i = \frac{1}{2} \sum_{j=1}^N W_{ij}$  el umbral,  $P$  un factor de escala y  $\tau_x$  la constante de tiempo. En su trabajo,  $\theta_i$  está acotada debido a que los elementos de la matriz de conexiones están normalizados, si  $P = 1$  el modelo corresponde al modelo original de Hopfield, y el número de atractores aumenta conforme aumenta el valor de  $g$ .

La energía original,  $E^O$ , es calculada con los pesos originales de la red y con las salidas de las neuronas obtenidas con los pesos modificados, es decir

$$E^O(H(t), \Omega(t=0)) = -\frac{1}{2} \sum_{ij} \alpha_{ij} V_i^r(t) V_j^r(t) + \sum_i \int_0^{V_i^r(t)} \sigma^{-1}(\xi) d\xi \quad (4.9)$$

donde  $\alpha_{ij} \equiv \omega_{ij}(t=0)$ , y  $H^r(t) = \langle V_1^r, \dots, V_N^r \rangle \in [-1, 1]^N$ . Finalmente, las tres condiciones, (C1), (C2) y (C3), así como los dos requerimientos prácticos, (R1) y (R2), explicados anteriormente son necesarios para que el proceso de auto-modelado funcione.

#### 4.4. Energía y satisfacción de restricciones en redes de Hopfield con conexiones asimétricas y auto-recurrentes.

El proceso de auto-modelado original exige que la matriz de conexiones sea simétrica y sin conexiones auto-recurrentes para garantizar la existencia de solamente atractores de punto fijo. Estas condiciones también aseguran que la función de energía  $E$  de la red sea una función de Lyapunov, con lo que se puede analizar la estabilidad de la red. Si la red es estable en el sentido de Lyapunov, entonces la energía siempre decrece al actualizar la dinámica de la red, ya que  $\frac{dE}{dt} \leq 0$ . De acuerdo con Z. Wang y Fan (2007), los modelos de Hopfield con conexiones

quasi-simétricas también tienen una función de energía decreciente, por lo en este trabajo se define una débil asimetría.

En una red de Hopfield de tiempo discreto y estado discreto, la energía siempre disminuirá al actualizar su dinámica si la función de energía es multilínea. La función de energía en la ecuación (4.3) es multilínea si y sólo si  $\omega_{ii} = 0$  para todo  $i$ . Si esta condición se mantiene, una matriz de conexiones asimétrica puede ser reemplazada por su parte simétrica  $(W + W^T)/2$  sin afectar el valor de la función de energía (Vidyasagar, 1998).

El valor de la función de energía es menor conforme más restricciones se resuelvan. Si existen conexiones auto-recurrentes, dado que  $s_i s_i = 1$ , la función de energía se puede expresar como:

$$E = -\frac{1}{2} \left[ \sum_{i=1, i \neq j}^N \omega_{ij} s_i s_j + \sum_{i=1}^N \omega_{ii} \right]. \quad (4.10)$$

El valor de la energía es afectado por un valor que desplazará negativamente el valor de la energía sin conexiones auto-recurrentes siempre que  $\sum_{i=1}^N \omega_{ii} > 0$ . Entonces su valor refleja correctamente la satisfacción de restricciones y se puede utilizar la ecuación (4.3) para evaluar el proceso de auto-modelado en redes de Hopfield de tiempo discreto y estado discreto.

Por otro lado, en el caso de los modelos de Hopfield con función de activación continua, el término integral es más complicado de analizar cuando existen conexiones auto-recurrentes. Ya que los pesos no tienen restricciones con respecto al signo, al actualizar la red no se puede determinar si las conexiones auto-recurrentes aumentan o disminuyen el valor de los estados  $s_i$  de las neuronas en la ecuación (2.3), o la salida  $V_i$  de las neuronas en la ecuación (2.6). Dado que no se sabe como varía el límite de  $s_i$  o  $V_i$ , no se puede determinar la magnitud del término integral. Entonces, aunque el término de suma discreta en la ecuación (4.10) disminuya, el término integral puede incrementar el valor de la energía. Por lo tanto, la energía no refleja correctamente el número de restricciones satisfechas. Este fenómeno ocurre tanto con conexiones simétricas como con conexiones asimétricas, e incluso sin la presencia de auto-recurrencia. En este trabajo se utilizan las ecuaciones (4.5) y (4.3) para medir la energía original de los modelos de Hopfield con estados continuos, y se calcula de manera independiente el número de restricciones satisfechas.

La forma de contar el número de restricciones satisfechas también es discutible. Los problemas usualmente resueltos por redes de Hopfield presentan soluciones discretas, por lo que éstas se encuentran en los vértices del hipercubo de espacio de estados en el caso de una red de Hopfield con estado discreto. Una red de Hopfield con estado continuo es poco probable que converja en algún estado estable en estos vértices, ya que existen muchos óptimos locales dentro del hipercubo (Joya et al., 1997). Entonces, estos óptimos en el interior no pueden ser solución de un problema de optimización ya que de manera estricta el dominio de las variables del problema es discreto. De manera formal, esto se debe a que la función de costo del problema no se puede asociar a la función de energía de la red (véase la Sección 3.3) a menos que la ganancia de la función sigmoide sea grande (región de alta ganancia) (Joya et al., 1997), donde el término integral es despreciable. Por lo tanto, es usual encontrar en la literatura que una red de Hopfield sea operada en la región de alta ganancia, en la cual

las trayectorias solución se aproximan a las esquinas del hipercubo donde se encuentran los estados estacionarios de la función de energía de la red de Hopfield discreta (Vidyasagar, 1998).

Considerar a las soluciones dentro del hipercubo podría resultar confuso en término de energía, es decir, un óptimo más cerca de alguna esquinas podría tener una mayor energía, aunque resuelva más restricciones, debido al término integral. Esto es, “hay siempre un contorno en el espacio de estados alrededor de  $V_i = 1$  o  $V_i = -1$  donde la energía es menor que aquella en  $V_i = 1$  o  $V_i = -1$ . Esto se debe a que el término integral siempre se incrementa cuando  $V_i$  se aproxima a 1 o  $-1$ , además el término de suma discreta varía linealmente con  $V_i$  e incluso si esta variación es negativa hay siempre un contorno alrededor de  $V_i = 1$  o  $V_i = -1$  donde el gradiente creciente del término integral es más grande que la disminución lineal del término de suma discreta” (Joya et al., 1997, p. 259-260). Entonces, los mínimos globales de la función de energía podrían no ser buenas soluciones del problema de optimización. El autor del presente trabajo desconoce si existe algún resultado en el que se haya demostrado que óptimos cercanos a los vértices del espacio de estados (en términos de la distancia de Hamming) resuelvan más restricciones que aquellos en el interior. No obstante, existen trabajos donde la manera de considerar válida una solución no es tan estricta. Por ejemplo, Molnár y Ercsey-Ravasz (2013); Molnár, Toroczka, y Ercsey-Ravasz (2012) consideran que existe una correspondencia uno a uno entre los atractores de punto fijo de una red recurrente de tiempo continuo (red de Hopfield de tiempo continuo) con una función de activación lineal acotada en  $[-1, 1]$  y las soluciones al problema de satisfacción booleana  $k - SAT$ . Esto lo consideran válido tanto para redes con conexiones simétricas como asimétricas. Con base en esto, y de manera análoga al trabajo de Froese et al. (2014), se considera que una restricción es satisfecha por una red de Hopfield de tiempo discreto y estado discreto si  $\alpha_{ij}s_i s_j > 0$ , con  $s_i \in -1, 1$ . De manera similar al trabajo de Zarco y Froese (en prensa), se considera que una restricción es satisfecha por una red de Hopfield de tiempo discreto y estado continuo si  $\alpha_{ij}s_i s_j > 0$ , con  $s_i \in [-1, 1]$ . En el caso de una red de Hopfield de tiempo continuo y estado continuo, se considera que una restricción es satisfecha si  $\alpha_{ij}V_i V_j > 0$ , donde  $V_i \in [-1, 1]$ .

## Capítulo 5

# Auto-modelado en redes de Hopfield de tiempo discreto y estado discreto: Resultados y Análisis

En este capítulo se presentan los resultados del proceso de auto-modelado para el modelo de Hopfield de tiempo discreto y estado discreto. El código fue desarrollado en C++ con ayuda de la biblioteca *Evolutionary Agents*<sup>1</sup> desarrollada por Randall Beer. Las gráficas fueron trazadas con ayuda de las bibliotecas *matplotlib* y *numpy* de Python. La prueba estadística *t* de Student para datos pareados se realizó con ayuda del software Mathematica.

En la Sección 5.1 se describen los tipos de restricciones que se utilizan en la matriz de conexiones para evaluar el proceso de auto-modelado, así como los parámetros de la red. Después se muestran ejemplos ilustrativos, los cuales muestran el resultado del proceso de auto-modelado. Éstos son presentados de manera análoga a los mostrados en el trabajo de Watson et al. (2011), es decir, son presentados en una figura con cuatro gráficas: (1) una gráfica con ejemplos de las trayectorias de la relajación de la red al estado estable antes del aprendizaje, (2) una gráfica con ejemplos de las trayectorias de la relajación de la red al estado estable después del aprendizaje, (3) una gráfica que muestra los atractores originales de la red, los atractores durante el proceso de auto-modelado y los atractores posteriores a éste, y (4) una gráfica que muestra el número de restricciones satisfechas antes y después del aprendizaje. Posteriormente se presentan los resultados de 10000 procesos de auto-modelado para cada tipo de restricción usando diagramas de caja que muestran la distribución de energía y la satisfacción de restricciones antes y después del aprendizaje. Para obtener estos datos el proceso de auto-modelado se aplicó usando 100 diferentes configuraciones de pesos originales y 100 diferentes configuraciones de estados iniciales por cada configuración de pesos, por lo tanto obteniendo 100 diferentes configuraciones de pesos modificados por cada configuración inicial. Así, desde cada una de las 100 configuraciones de estados iniciales para cada configuración de pesos iniciales la

---

<sup>1</sup>El código se puede descargar en <http://mypage.iu.edu/~rdbeer/>.

red fue relajada 100 veces (antes del aprendizaje); y para cada configuración de pesos modificados la red fue relajada 100 veces (después del aprendizaje). Entonces, la red se relajó 1000000 veces antes del aprendizaje y 1000000 después del aprendizaje. Al final de esta sección se presentan los resultados de la prueba  $t$  de Student hecha para cada tipo de restricción con respecto al número de restricciones satisfechas para saber si el proceso es estadísticamente significativo. Por último, en la Sección 5.2, se presenta el análisis de los resultados.

## 5.1. Resultados

El proceso de auto-modelado fue probado en este modelo de red de Hopfield para cada una de las siguientes restricciones

- **Restricciones Aleatorias Simétricas (RA Simétricas):** cada conexión inicial  $\alpha_{ij}$  toma  $-1$  o  $1$  con igual probabilidad de  $\{-1, 1\}$ .
- **Restricciones Aleatorias Asimétricas (RA Asimétricas):** a cada elemento de una matriz con restricciones aleatorias simétrica se le suma un valor aleatorio utilizando una distribución aleatoria uniforme entre  $[0, 0.25]$  si la  $|\alpha_{ij}| = -1$  o entre  $[-0.25, 0]$  si  $|\alpha_{ij}| = 1$ .
- **Restricciones Modulares Simétricas (RM Simétricas):** cada conexión inicial  $\alpha_{ij}$  toma el valor  $|\alpha_{ij}| = 1$  si  $\lfloor \frac{i}{k} \rfloor = \lfloor \frac{j}{k} \rfloor$ ; de otro modo  $|\alpha_{ij}| = 0.01$ . En este trabajo,  $k = 5$  y  $\alpha_{ij} > 0$  con probabilidad 0.8.
- **Restricciones Modulares Asimétricas (RM Asimétricas):** a cada elemento de una matriz con restricciones modulares simétrica se le suma un valor aleatorio utilizando una distribución aleatoria uniforme entre  $[-0.01, -0.001]$  si  $|\alpha_{ij}| < 0$  o entre  $[0.001, 0.01]$  si  $|\alpha_{ij}| > 0$ .

Las matrices de pesos tienen conexiones auto-recurrentes sin más condiciones en los signos que los especificados en cada tipo de restricción, es decir durante el aprendizaje puede cambiar el signo de la conexión. En este trabajo se utilizaron los siguientes valores:

- Número de neuronas:  $N = 100$
- Pasos de tiempo por relajación:  $\tau = 1000$
- Número de relajaciones:  $r = 1000$
- Tasa de aprendizaje:
  - $\delta_{RA \text{ Simétricas}} = 0.00075$
  - $\delta_{RM \text{ Simétricas}} = 0.00075$
  - $\delta_{RA \text{ Asimétricas}} = 0.00075$
  - $\delta_{RM \text{ Asimétricas}} = 0.00075$

Las tasas de aprendizaje  $\delta$  fueron definidas experimentalmente después de asegurar que el proceso de auto-modelado funcionaba correctamente en varias ejecuciones individuales usando las restricciones definidas. La actualización de los pesos se realiza al final de cada periodo de relajación utilizando la ecuación (4.1). Para contabilizar el número de restricciones satisfechas se consideró que una restricción se satisface si  $\alpha_{ij}s_i s_j > 0$ .

### 5.1.1. Ejemplo ilustrativo de un proceso de auto-modelado

#### Restricciones Aleatorias Simétricas

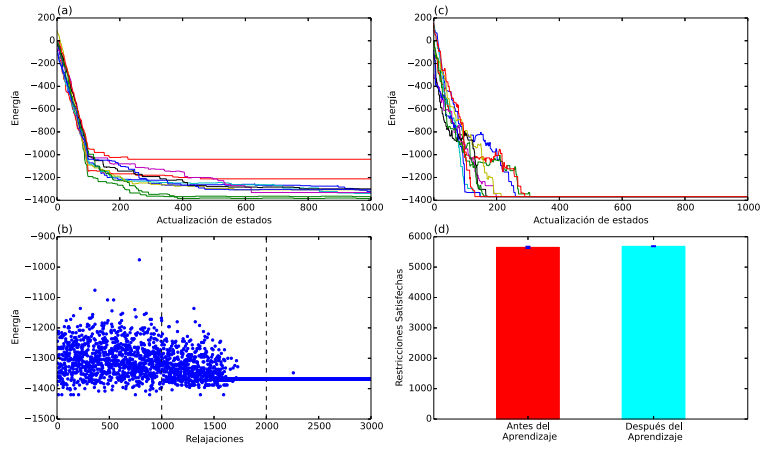


Figura 5.1: Auto-modelado en redes con restricciones aleatorias simétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relajaciones 1-1000), durante el aprendizaje (relajaciones 1000-2000), y después del aprendizaje (relajaciones 2000-3000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relajaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul). Las barras de error representan un error estándar.

#### Restricciones Aleatorias Asimétricas

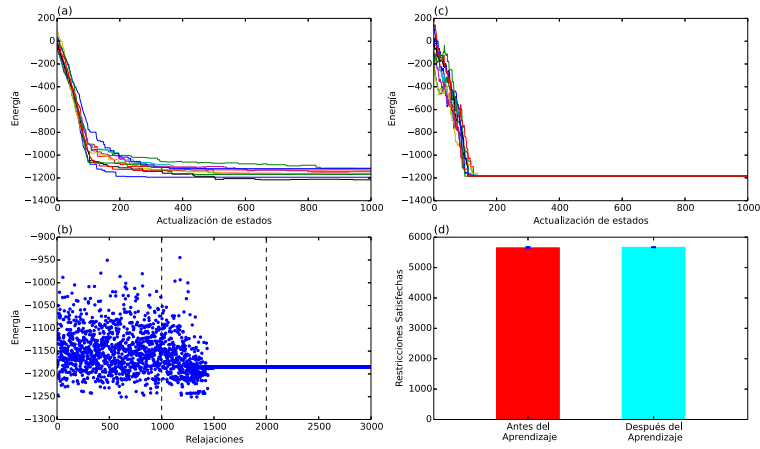


Figura 5.2: Auto-modelado en redes con restricciones aleatorias asimétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relajaciones 1-1000), durante el aprendizaje (relajaciones 1000-2000), y después del aprendizaje (relajaciones 2000-3000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relajaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul). Las barras de error representan un error estándar.

### Restricciones Modulares Simétricas

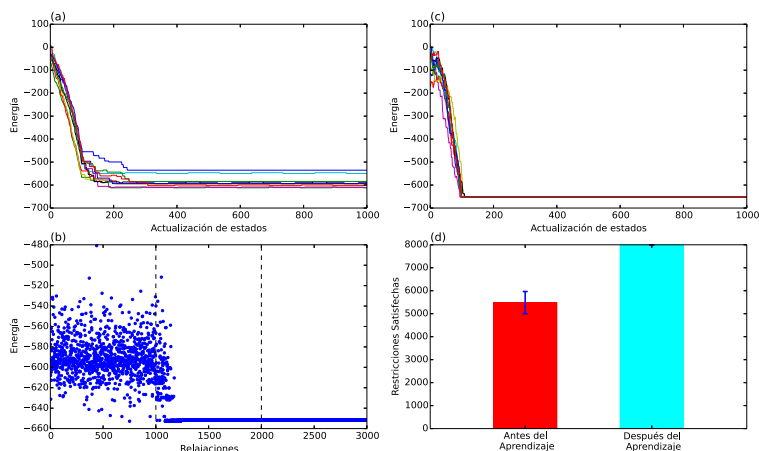


Figura 5.3: Auto-modelado en redes con restricciones modulares simétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relajaciones 1-1000), durante el aprendizaje (relajaciones 1000-2000), y después del aprendizaje (relajaciones 2000-3000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relajaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul). Las barras de error representan un error estándar.

### Restricciones Modulares Asimétricas

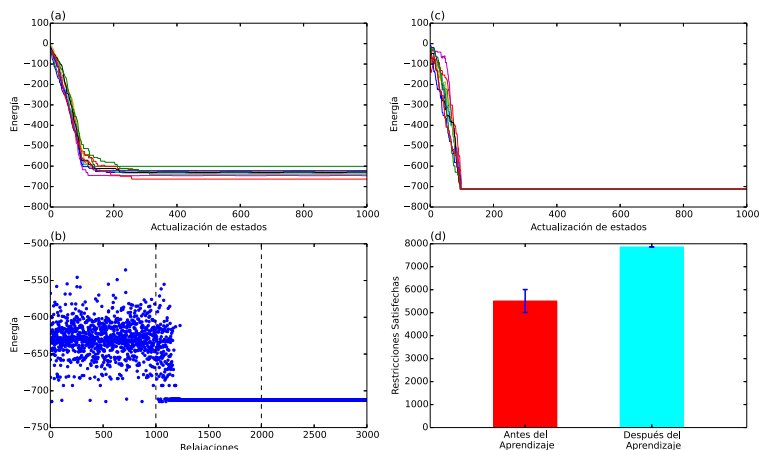


Figura 5.4: Auto-modelado en redes con restricciones modulares asimétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relajaciones 1-1000), durante el aprendizaje (relajaciones 1000-2000), y después del aprendizaje (relajaciones 2000-3000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relajaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul). Las barras de error representan un error estándar.



### 5.1.2. Resultados de 10000 procesos de auto-modelado Restricciones Aleatorias

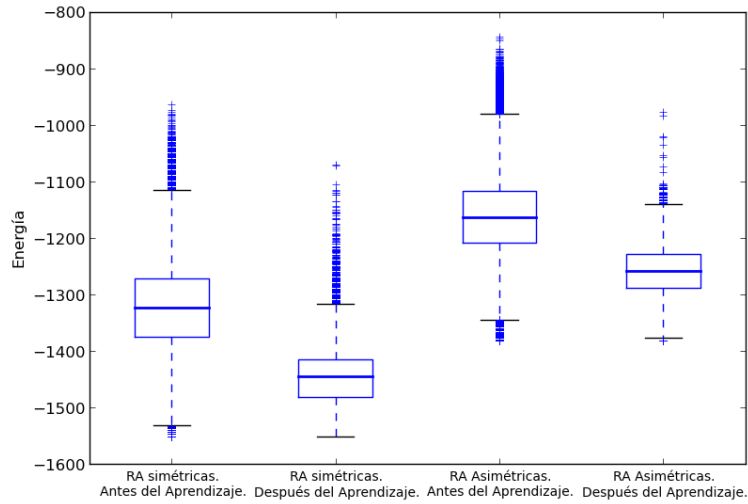


Figura 5.5: Distribución de la energía antes y después del proceso de auto-modelado en redes con restricciones aleatorias simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relajaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relajaciones.

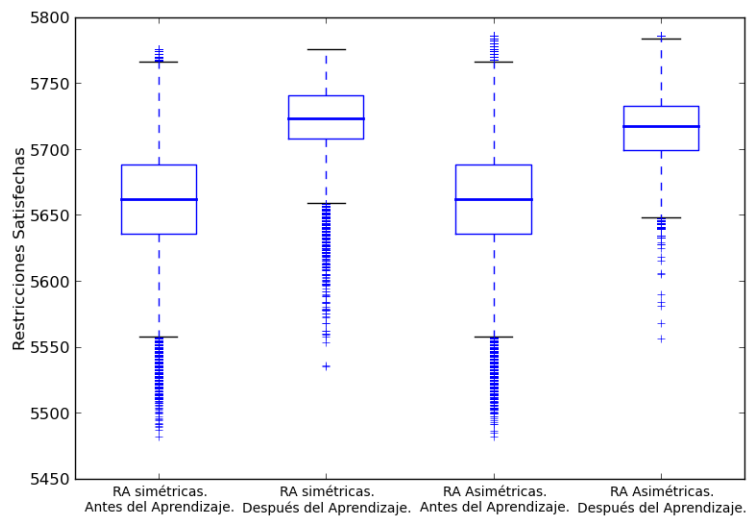


Figura 5.6: Restricciones satisfechas antes y después del proceso de auto-modelado en redes con restricciones aleatorias simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relajaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relajaciones.

## Restricciones Modulares

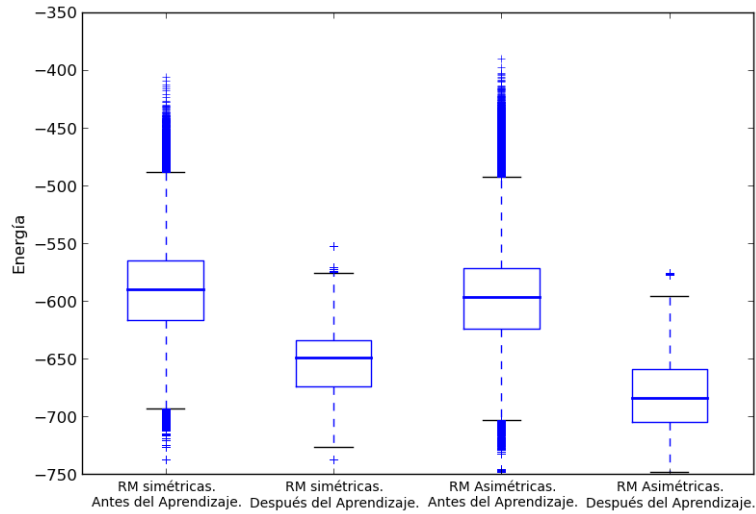


Figura 5.7: Distribución de la energía antes y después del proceso de auto-modelado en redes con restricciones modulares simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relaciones.

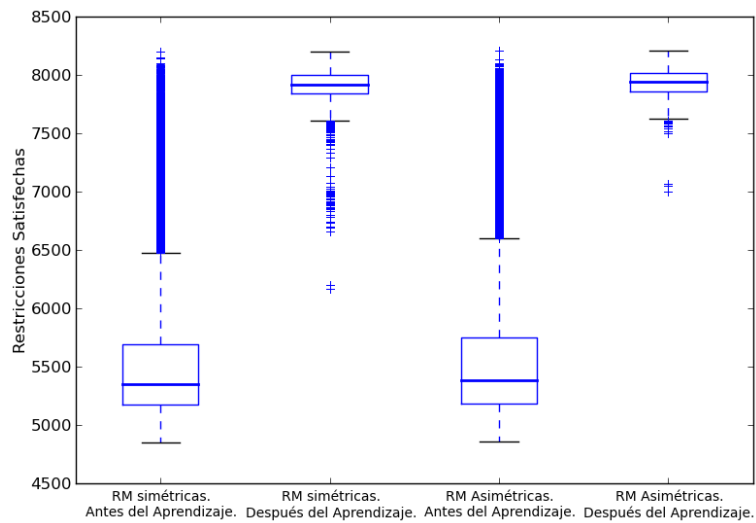


Figura 5.8: Restricciones satisfechas antes y después del proceso de auto-modelado en redes con restricciones modulares simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relaciones.

### 5.1.3. Prueba $t$ de Student

A continuación se muestran los resultados obtenidos de esta prueba estadística para saber si la diferencia entre los resultados antes y después del aprendizaje es estadísticamente significativa. La pregunta que se trata de resolver es: ¿el proceso de auto-modelado permite encontrar mejores soluciones al problema de satisfacción de restricciones en una red de Hopfield de tiempo discreto y estado discreto? Sea la media de las 100 relajaciones desde cada una de las 100 configuraciones de estados iniciales para una configuración de pesos iniciales. El conjunto de datos  $AA$  está definido por las medias de las medias calculadas de la forma anterior para cada una de las 100 configuraciones de pesos iniciales. Ahora, sea la media de las 100 relajaciones para cada una de las 100 configuraciones de pesos modificados obtenida a partir de una de las 100 configuraciones de pesos iniciales. El conjunto de datos  $DA$  está definido por las medias de las medias calculadas de la forma inmediata anterior para cada una de las 100 configuraciones de pesos iniciales. Por lo tanto, si se define la diferencia  $d = DA - AA$  para calcular los estadísticos, la hipótesis nula es  $H_0 : \mu_d \leq 0$ , y la hipótesis alternativa es  $H_a : \mu_d > 0$ , donde  $\mu_{DA}$  indica la media de  $DA$  y  $\mu_{AA}$  indica la media de  $AA$ .

#### Restricciones Aleatorias Simétricas

- Media: 61.222748
- Desviación estándar: 13.6059385984
- p:  $8.1403904688 \times 10^{-68}$

#### Restricciones Aleatorias Asimétricas

- Media: 54.757914
- Desviación estándar: 14.2536221525
- p:  $2.18224966801 \times 10^{-61}$

#### Restricciones Modulares Simétricas

- Media: 2385.311602
- Desviación estándar: 114.990789523
- p:  $1.51796959436 \times 10^{-132}$

#### Restricciones Modulares Asimétricas

- Media: 2378.365332
- Desviación estándar: 91.2164871583
- p:  $2.32484265465 \times 10^{-142}$

## 5.2. Análisis

El proceso de auto-modelado funciona correctamente de acuerdo con los resultados de la Sección 5.1. El proceso funciona debido a que los atractores de baja energía tienen un dominio de atracción grande, y a que la memoria asociativa tiene la capacidad de generalizar idealizando los patrones almacenados o generando nuevas combinaciones de sub-patrones Watson et al. (2011a). El aprendizaje Hebbiano crea una memoria asociativa del atractor que se refuerza, para lo que incrementa la fuerza de las conexiones que concuerdan con las combinaciones de estados en ese patrón y debilita las conexiones de las combinaciones que lo perturban. Entonces, ampliar el dominio de atracción de los atractores de baja energía no solo depende de la frecuencia con la que la red converge en éstos, sino también de los sub-patrones que los óptimos globales tengan en común con los óptimos locales. En general, el aprendizaje Hebbiano con facilitación y depresión sináptica no afecta los atractores de punto fijo en redes de Hopfield de tiempo discreto y estado discreto (Bibitchkov, Herrmann, & Geisel, 2002).

En los resultados mostrados en la Sección 5.1, dado que la matriz de conexiones es asimétrica y con conexiones auto-recurrentes, la energía de la red no siempre decrece durante el transitorio, pero la red es capaz de converger a un estado estable, como se muestra en la gráfica (a) de las Figuras 5.1, 5.2, 5.3, y 5.4. El aprendizaje Hebbiano fue aplicado al final del periodo de relajación, momento en el que se considera que la red ya ha alcanzado un estado estable. La gráfica (c) de estas figuras muestra como el transitorio para alcanzar el estado estable es más rápido después del aprendizaje, y puede presentar aumentos en la energía original,  $E^O$ . Esto se debe en parte a la presencia de conexiones auto-recurrente, y en parte a que las restricciones inter-modulares son violadas en favor de satisfacer las restricciones intra-modulares más fuertes, en el caso de la matriz con conexiones modulares (Watson et al., 2011b).

Por su parte, la gráfica (b) ejemplifica la distribución de la energía de los atractores para un ejemplo en particular antes, durante y después del proceso de auto-modelado. Se aprecia que en el caso de las restricciones aleatorias, problemas no estructurados, el atractor al final del aprendizaje es parte del conjunto al cual la red originalmente podía converger. Lo anterior se ve reflejado en la gráfica (d). Esto es, en el caso de las restricciones aleatorias, aunque los valores energía de los atractores parezcan ampliamente distribuidos, el número de restricciones satisfechas no parece variar mucho, por lo que el promedio parece no reflejar un aumento en el número de la satisfacción. No obstante, las Figuras 5.5 y 5.6 muestran de manera estadística que en promedio existe una disminución de energía y un aumento en las restricciones satisfechas, respectivamente. En el caso de las restricciones modulares, problemas con estructura, el atractor al final del aprendizaje es un atractor al cual la red no había convergido anteriormente, o al cual la red tenía pocas probabilidades de converger. En este caso los valores de energía de los atractores reflejan mejor el número de restricciones satisfechas, lo cual se aprecia de manera estadística en las Figuras 5.7 y 5.8. Los resultados en el caso de las restricciones modulares son superiores porque el proceso puede explotar la correlación entre óptimo locales y superiores, para que la red converga en atractores de menor energía.

## Capítulo 6

# Auto-modelado en redes de Hopfield de tiempo discreto y estado continuo: Resultados y Análisis

En este capítulo se presentan los resultados del proceso de auto-modelado para el modelo de Hopfield de tiempo discreto y estado continuo. El código fue desarrollado en C++ con ayuda de la biblioteca *Evolutionary Agents* desarrollada por Randall Beer. La integral de la función de energía se calculó usando la regla de punto medio con 1000 subintervalos. Las gráficas fueron trazadas con ayuda de las bibliotecas *matplotlib* y *numpy* de Python. La prueba estadística *t* de Student para datos pareados se realizó con ayuda del software Mathematica.

En la Sección 6.1 se describen los tipos de restricciones que se utilizan en la matriz de conexiones para evaluar el proceso de auto-modelado, así como los parámetros de la red. Después se muestran ejemplos ilustrativos, los cuales muestran el resultado del proceso de auto-modelado. Éstos son presentados en una figura con cuatro gráficas: (1) una gráfica con ejemplos de las trayectorias de la relajación de la red al estado estable antes del aprendizaje, (2) una gráfica con ejemplos de las trayectorias de la relajación de la red al estado estable después del aprendizaje, (3) una gráfica que muestra los atractores originales de la red, los atractores durante el proceso de auto-modelado y los atractores posteriores a éste, y (4) una gráfica que muestra el número de restricciones satisfechas antes y después del aprendizaje. Posteriormente se presentan los resultados de 10000 procesos de auto-modelado para cada tipo de restricción usando diagramas de caja que muestran la distribución de energía y la satisfacción de restricciones antes y después del aprendizaje. Para obtener estos datos el proceso de auto-modelado se aplicó usando 100 diferentes configuraciones de pesos originales y 100 diferentes configuraciones de estados iniciales por cada configuración de pesos, por lo tanto obteniendo 100 diferentes configuraciones de pesos modificados por cada configuración inicial. Así, desde cada una de las 100 configuraciones de estados iniciales para cada configuración de pesos iniciales la red fue relajada 100 veces (antes del aprendizaje); y para cada configuración de pesos modifica-

dos la red fue relajada 100 veces (después del aprendizaje). Entonces, la red se relajó 1000000 veces antes del aprendizaje y 1000000 después del aprendizaje. Al final de esta sección se presentan los resultados de la prueba  $t$  de Student hecha para cada tipo de restricción con respecto al número de restricciones satisfechas para saber si el proceso es estadísticamente significativo. Por último, en la Sección 6.2, se presenta el análisis de los resultados.

## 6.1. Resultados

El proceso de auto-modelado fue probado en este modelo de red de Hopfield para cada una de las siguientes restricciones en las conexiones:

- **Restricciones Aleatorias Simétricas (RAS):** cada conexión inicial  $\alpha_{ij}$  toma un valor aleatorio utilizando una distribución aleatoria uniforme entre  $[-1, 1]$ .
- **Restricciones Aleatorias Asimétricas (RAA):** a cada elemento de una matriz con restricciones aleatorias simétrica se le suma un valor entre  $[-0.25, 0.25]$  utilizando una distribución aleatoria uniforme.
- **Restricciones Modulares Simétricas (RMS):** cada conexión inicial  $\alpha_{ij}$  toma el valor  $|\alpha_{ij}| = 1$  si  $\lfloor \frac{i}{k} \rfloor = \lfloor \frac{j}{k} \rfloor$ ; de otro modo  $|\alpha_{ij}| = 0.01$ . En este trabajo,  $k = 5$  y  $\alpha_{ij} > 0$  con probabilidad 0.8.
- **Restricciones Modulares Asimétricas (RMA):** a cada elemento de una matriz con restricciones modulares simétrica se le suma un valor aleatorio utilizando una distribución aleatoria uniforme entre  $[-0.01, -0.001]$  si  $|\alpha_{ij}| < 0$  o entre  $[0.001, 0.01]$  si  $|\alpha_{ij}| > 0$ .

Las matrices de pesos tienen conexiones auto-recurrentes sin más condiciones en los signos que los especificados en cada tipo de restricción, es decir durante el aprendizaje puede cambiar el signo de la conexión. En este trabajo se utilizaron los siguientes valores:

- Número de neuronas:  $N = 100$
- Pasos de tiempo por relajación:  $\tau = 1000$
- Número de relajaciones:  $r = 1000$
- Tasa de aprendizaje:
  - $\delta_{RA}$  Simétricas = 0.00075
  - $\delta_{RM}$  Simétricas = 0.00075
  - $\delta_{RA}$  Asimétricas = 0.00001
  - $\delta_{RM}$  Asimétricas = 0.00001

Las tasas de aprendizaje  $\delta$  fueron definidas experimentalmente después de asegurar que el proceso de auto-modelado funcionaba correctamente en varias ejecuciones individuales usando las restricciones definidas. La actualización de los pesos se realiza al final de cada periodo de relajación utilizando la ecuación (4.4). Para contabilizar el número de restricciones satisfechas se consideró que una restricción se satisface si  $\alpha_{ij}s_i s_j > 0$ .

### 6.1.1. Ejemplo ilustrativo de un proceso de auto-modelado

#### Restricciones Aleatorias Simétricas

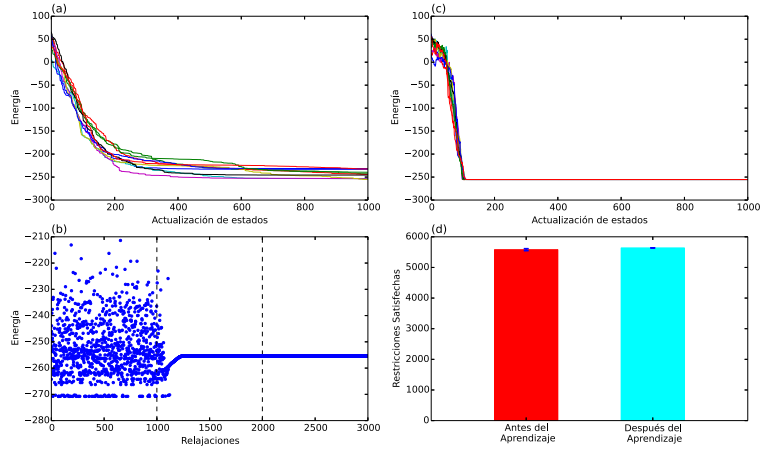


Figura 6.1: Auto-modelado en redes con restricciones aleatorias simétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relajaciones 1-1000), durante el aprendizaje (relajaciones 1000-2000), y después del aprendizaje (relajaciones 2000-3000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relajaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul). Las barras de error representan un error estándar.

#### Restricciones Aleatorias Asimétricas

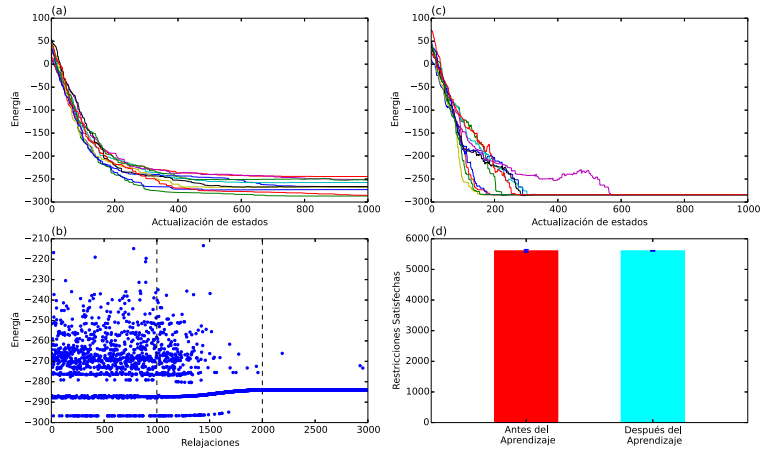


Figura 6.2: Auto-modelado en redes con restricciones aleatorias asimétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relajaciones 1-1000), durante el aprendizaje (relajaciones 1000-2000), y después del aprendizaje (relajaciones 2000-3000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relajaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul). Las barras de error representan un error estándar.

### Restricciones Modulares Simétricas

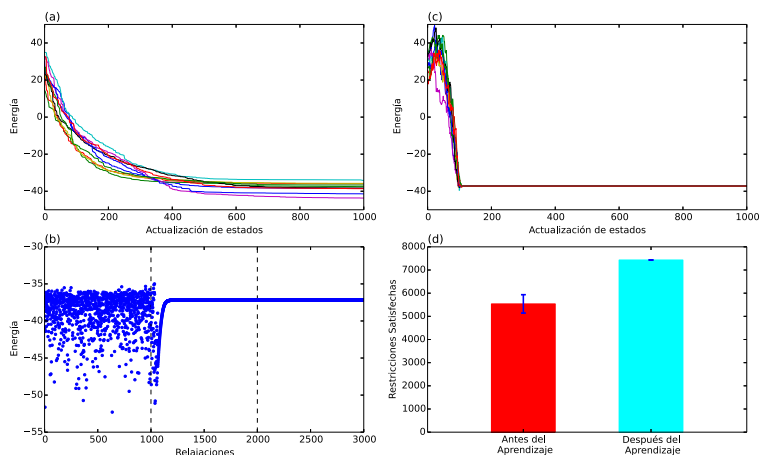


Figura 6.3: Auto-modelado en redes con restricciones modulares simétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relajaciones 1-1000), durante el aprendizaje (relajaciones 1000-2000), y después del aprendizaje (relajaciones 2000-3000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relajaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul). Las barras de error representan un error estándar.

### Restricciones Modulares Asimétricas

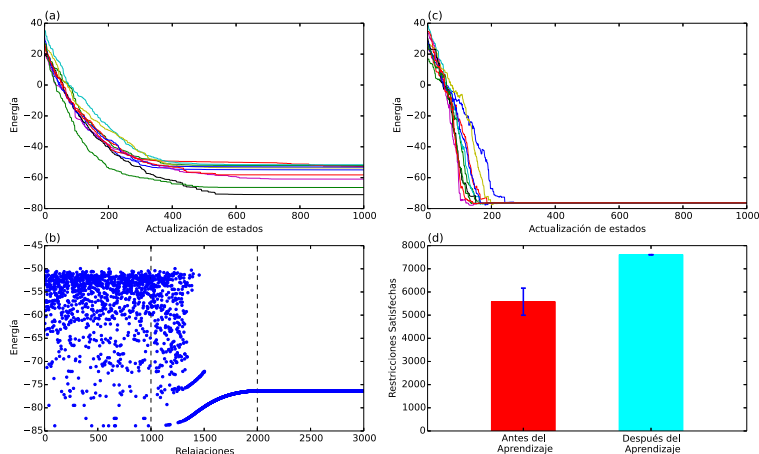


Figura 6.4: Auto-modelado en redes con restricciones modulares asimétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relajaciones 1-1000), durante el aprendizaje (relajaciones 1000-2000), y después del aprendizaje (relajaciones 2000-3000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relajaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul). Las barras de error representan un error estándar.



### 6.1.2. Resultados de 10000 procesos de auto-modelado Restricciones Aleatorias

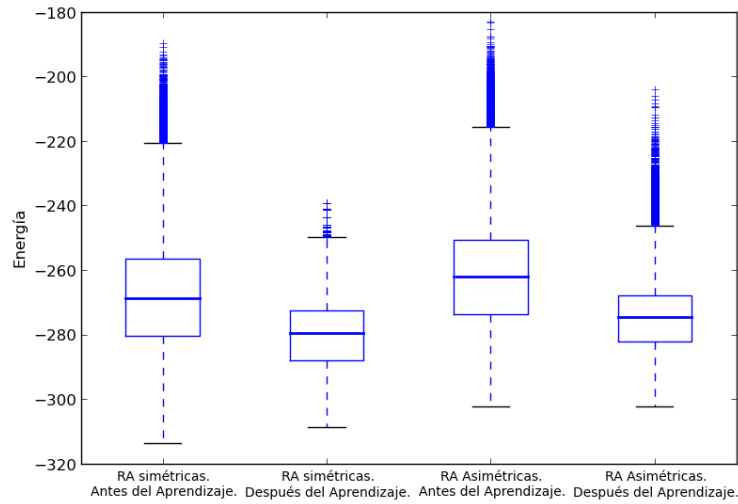


Figura 6.5: Distribución de la energía antes y después del proceso de auto-modelado en redes con restricciones aleatorias simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relajaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relajaciones.

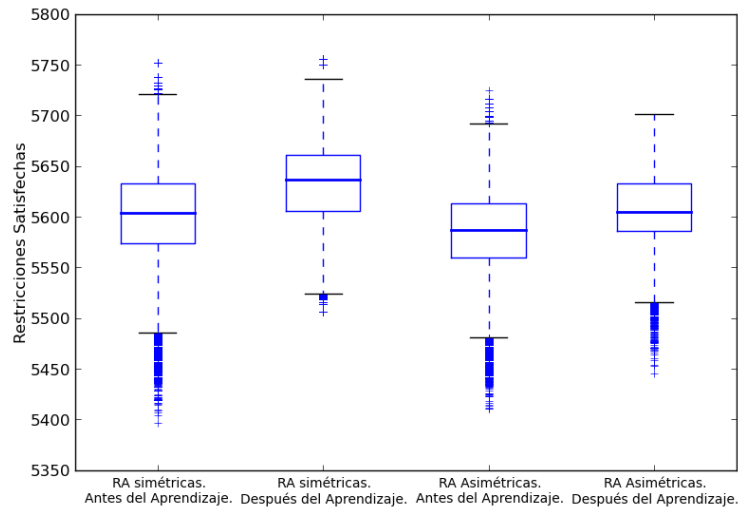


Figura 6.6: Restricciones satisfechas antes y después del proceso de auto-modelado en redes con restricciones aleatorias simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relajaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relajaciones.

## Restricciones Modulares

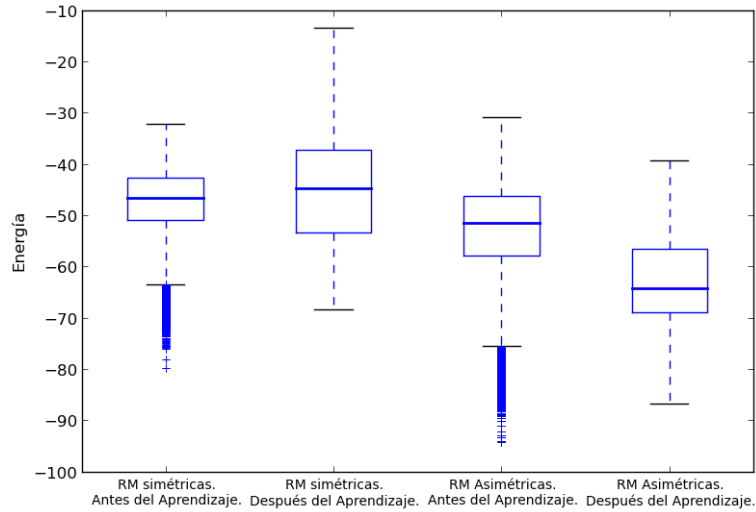


Figura 6.7: Distribución de la energía antes y después del proceso de auto-modelado en redes con restricciones modulares simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relaciones.

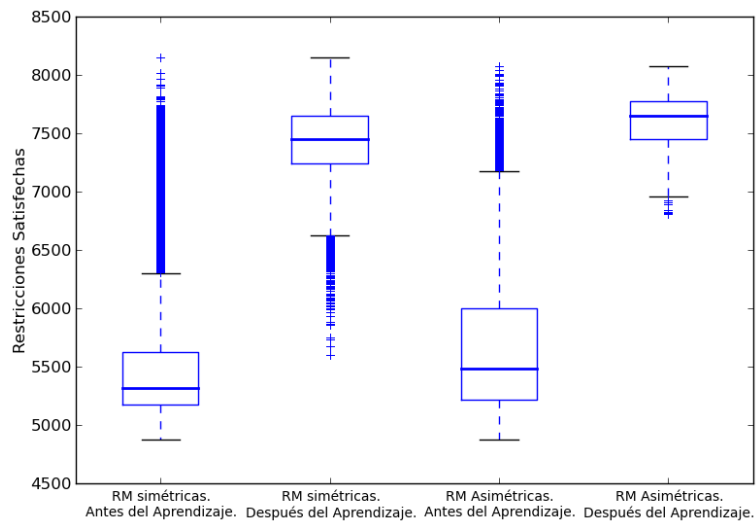


Figura 6.8: Restricciones satisfechas antes y después del proceso de auto-modelado en redes con restricciones modulares simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relaciones.

### 6.1.3. Prueba $t$ de Student

A continuación se muestran los resultados obtenidos de esta prueba estadística para saber si la diferencia entre los resultados antes y después del aprendizaje es estadísticamente significativa. La pregunta que se trata de resolver es: ¿el proceso de auto-modelado permite encontrar mejores soluciones al problema de satisfacción de restricciones en una red de Hopfield de tiempo discreto y estado discreto? Sea la media de las 100 relajaciones desde cada una de las 100 configuraciones de estados iniciales para una configuración de pesos iniciales. El conjunto de datos  $AA$  está definido por las medias de las medias calculadas de la forma anterior para cada una de las 100 configuraciones de pesos iniciales. Ahora, sea la media de las 100 relajaciones para cada una de las 100 configuraciones de pesos modificados obtenida a partir de una de las 100 configuraciones de pesos iniciales. El conjunto de datos  $DA$  está definido por las medias de las medias calculadas de la forma inmediata anterior para cada una de las 100 configuraciones de pesos iniciales. Por lo tanto, si se define la diferencia  $d = DA - AA$  para calcular los estadísticos, la hipótesis nula es  $H_0 : \mu_d \leq 0$ , y la hipótesis alternativa es  $H_a : \mu_d > 0$ , donde  $\mu_{DA}$  indica la media de  $DA$  y  $\mu_{AA}$  indica la media de  $AA$ .

#### Restricciones Aleatorias Simétricas

- Media: 29.72268
- Desviación estándar: 20.9921446121
- p:  $1.086845593525 \times 10^{-25}$

#### Restricciones Aleatorias Asimétricas

- Media: 22.30903
- Desviación estándar: 23.2203915038
- p:  $5.0046422987 \times 10^{-16}$

#### Restricciones Modulares Simétricas

- Media: 1944.065534
- Desviación estándar: 255.098901396
- p:  $8.3251973293 \times 10^{-90}$

#### Restricciones Modulares Asimétricas

- Media: 1911.851268
- Desviación estándar: 200.709942432
- p:  $2.877787338865 \times 10^{-99}$

## 6.2. Análisis

Es natural tratar de extender el proceso de auto-modelado actualizando los pesos de las conexiones utilizando la regla incremental (4.2) y el valor continuo de los estados. Sin embargo, el proceso de auto-modelado no funciona como se esperaba en términos energéticos. Esto se aprecia en la gráfica (b) de las figuras 6.1, 6.2, 6.3, y 6.4. No obstante, en la gráfica (d) se aprecia que la cantidad de restricciones satisfechas (una restricción es satisfecha si  $\alpha_{ij}s_i s_j > 0$ ) es mayor después del aprendizaje. Esta diferencia también se puede observar en las Figuras 6.5 y 6.6, en el caso de las restricciones aleatorias, y en las Figuras 6.7 y 6.8, en el caso de las restricciones modulares. En el caso de las restricciones aleatorias no se aprecia claramente de manera individual pero sí de forma estadística. Esta discrepancia en la relación de la energía con la satisfacción de restricciones se debe al efecto del aprendizaje en la dinámica de la red. La dinámica de la red provoca que ésta converja en mínimo de la función de energía dentro del hipercubo de espacio de estados debido a la convexidad del perfil energético. Los estados de la red no pueden converger en los valores extremos  $\pm 1$  debido a que los atractores en los vértices son asintóticamente estables debido al crecimiento sin límite del término integral de la función de energía en la ecuación (4.5). Es decir, el término integral aumenta sin límite conforme  $s_i$  se aproxima  $\pm 1$ . Además, este término es siempre positivo (véase la Figura 6.9) y, dado que los óptimos se encuentran en una vecindad de la esquina del hipercubo y los pesos están limitados en el intervalo  $[-1, 1]$ , el valor del término integral es en general más grande que el valor del primer término de la ecuación (4.5), cuya minimización refleja la satisfacción de restricciones. Por lo tanto, la cantidad de restricciones satisfechas no es correctamente representada por la energía.

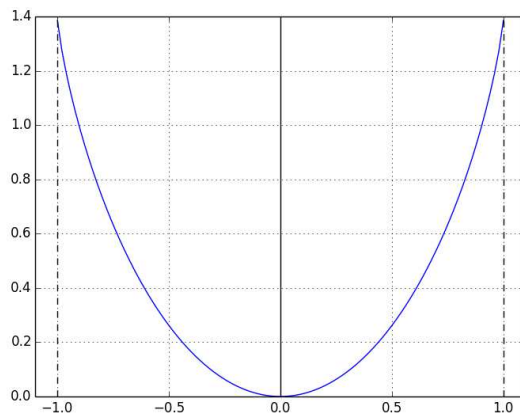


Figura 6.9: Gráfica de  $\int_0^{s_i} \sigma^{-1}(\xi) d\xi$

Por su parte, el aprendizaje Hebbiano provoca que los pesos tiendan a crecer sin límite (o hasta un límite en la plasticidad sináptica) a través de un bucle de retroalimentación positiva (Fyfe, 2005). De acuerdo con Vico y Jerez (2003), aprender nuevos patrones (en este modelo de red de Hopfield) introduce una

interferencia por la respuesta superpuesta de las neuronas mal sintonizadas con el nuevo patrón. Entonces, el continuo reforzamiento de patrones tiene un efecto acumulativo de pequeños cambios en las conexiones que termina con representaciones (atractores) fuertemente degradadas: pequeños cambios en los pesos debido al aprendizaje, hace más probable la saturación de los estados, lo que a su vez hace más probable la saturación de los pesos, y así sucesivamente. Por lo tanto, reforzar los atractores de la red provoca que éstos se ‘desplacen’ hacia las esquinas del hipercubo de espacio de estados, lo que incrementa el valor del término integral de la función de energía.

Con base en lo anterior, se puede comprender el comportamiento en la gráfica (b) (relajaciones 1000-2000). Durante el aprendizaje los atractores se desplazan hacia las esquinas del hipercubo y la energía del término integral aumenta. Entonces, dado que hay mayor probabilidad de que el valor de este término sea más grande que el valor del término de suma discreta, la energía de los atractores tiende a incrementarse. Por su parte, la competencia entre los atractores debido al proceso de auto-modelado provoca que sólo uno permanezca en el espacio de estados (al menos en el caso ideal), el cual en general estaría casi en las esquinas del hipercubo. En este trabajo, los estados de la red alcanzan los valores extremos  $\pm 1$  debido a limitaciones computacionales. De igual forma, el término integral no crece sin límite debido a que el método usado para calcular la integral le impone un valor límite cuando  $s_i = \pm 1$ . La topología original de la red (antes del aprendizaje) define un conjunto de atractores de los cuales la red es sólo capaz de converger a un subconjunto, los cuales son mínimos locales. En general, los atractores a los que la red converge durante la auto-modelación no son parte del conjunto original, no obstante en promedio resuelven más restricciones con las conexiones originales que aquellos en los que la red es capaz de converger.

En este trabajo, la actualización de los pesos se realiza al final del periodo de relajación de la red cuando la red se encuentra en un estado estable. De acuerdo con la ecuación 4.2, es posible actualizar los pesos en cada paso de tiempo del periodo de relajación, al menos para el modelo de Hopfield originalmente usada en el proceso de auto-modelado. No obstante, dado que la cantidad de aprendizaje Hebbiano no es constante y que la energía no decrece siempre al actualizar la dinámica de la red, no se puede asegurar que el efecto global sea el mismo, ya que puede existir un transitorio que necesite muchos pasos de actualización, por ejemplo véase la gráfica (a) de las figuras. Por su parte, la gráfica (c) muestra que el transitorio a un estado estable es más rápido después del proceso de auto-modelado, y puede presentar aumentos en la energía original,  $E^O$ , debido a la existencia de conexiones auto-recurrentes y a la presencia del término integral.

De acuerdo con Watson et al. (2011a), los cambios en las conexiones están determinados por la correlación entre los estados no por el valor del peso entre ellos. Por lo tanto, la cantidad de aprendizaje será constante y simétrica, es decir,  $\Delta\omega_{ij} = \delta s_i s_j = \Delta\omega_{ji}$ , incluso si las conexiones originales son asimétricas o de signos diferentes. Entonces, los pesos modificados,  $\omega_{ij}$  y  $\omega_{ji}$ , tomarán el mismo signo conforme los pesos se actualicen aunque sus valores iniciales,  $\alpha_{ij}$  y  $\alpha_{ji}$ , no tengan el mismo signo. Esto asegura que la dinámica de la red tienda a simplificarse con el tiempo, es decir, hace más probable la existencia de atractores de punto fijo que de otro tipo de conjuntos límite. En el caso para redes de Hopfield de tiempo discreto y estado continuo la cantidad de aprendizaje es simétrico, esto es  $\Delta\omega_{ij} = \Delta\omega_{ji}$ . Entonces, los pesos modificados también ten-

derán a tomar el mismo signo con el tiempo, aún cuando las conexiones iniciales sean asimétricas o de signos diferentes. En este trabajo, la débil asimetría de la matriz de conexiones hace más probable la presencia de atractores de punto fijo. Por lo tanto, en este modelo de Hopfield, hay una reducción de la complejidad de los atractores durante el aprendizaje. Siri, Berry, Cessac, Delord y Quoy analizaron el aprendizaje Hebbiano en un modelo de Hopfield con función de activación continua y conexiones aleatorias. En estos trabajos concluyeron que la reducción en la dinámica se debe a la función de la depresión sináptica en conjunto con el incremento en el nivel de saturación promedio de las neuronas (Siri, Berry, Cessac, Delord, & Quoy, 2007; Siri, Quoy, Delord, Cessac, & Berry, 2007), y/o a que el aprendizaje Hebbiano tiende a organizar la red en una red de mundo pequeño (Siri, Berry, Cessac, Delord, & Quoy, 2006). Por ejemplo, existen resultados que demuestran que, para un modelo de Hopfield de tiempo discreto y estado discreto, la depresión sináptica agranda el dominio de atracción de patrones almacenados usando una regla de aprendizaje tipo Hebbiano (Matsumoto, Ide, Watanabe, & Okada, 2005). Por su parte, los sistemas modulares tienden naturalmente a ser redes de mundo pequeño, aunque lo contrario no es siempre verdad, es decir, algunas redes de mundo pequeño no son modulares (Meunier, Lambiotte, & Bullmore, 2010). Por ejemplo, Hua et al. (1999) mostraron que la modularidad emerge en una red de Hopfield continua con conexiones aleatorias cuando se usa aprendizaje Hebbiano. De acuerdo con ese trabajo, la modularidad ayuda a eliminar la interferencia espacial y temporal entre patrones almacenados (*crossstalk*), realiza generalizaciones locales, permite una mejor representación de la tarea a ser realizada, y simplifica y acelera el aprendizaje. Esto ayuda a comprender como el proceso de auto-modelado explota la estructura emergente en la red.

## Capítulo 7

# Auto-modelado en redes de Hopfield de tiempo continuo y estado continuo: Resultados y Análisis

En este capítulo se presentan los resultados del proceso de auto-modelado para el modelo de Hopfield de tiempo continuo y estado continuo. El código fue desarrollado en C++ con ayuda de la biblioteca *Evolutionary Agents* desarrollada por Randall Beer. La integral de la función de energía se calculó usando la regla de punto medio con 1000 subintervalos. Las gráficas fueron trazadas con ayuda de las bibliotecas *matplotlib* y *numpy* de Python. La prueba estadística *t* de Student para datos pareados se realizó con ayuda del software Mathematica.

En la Sección 7.1 se describen los tipos de restricciones que se utilizan en la matriz de conexiones para evaluar el proceso de auto-modelado, así como los parámetros de la red. Después se muestran ejemplos ilustrativos, los cuales muestran el resultado del proceso de auto-modelado. Éstos son presentados en una figura con cuatro gráficas: (1) una gráfica con ejemplos de las trayectorias de la relajación de la red al estado estable antes del aprendizaje, (2) una gráfica con ejemplos de las trayectorias de la relajación de la red al estado estable después del aprendizaje, (3) una gráfica que muestra los atractores originales de la red, los atractores durante el proceso de auto-modelado y los atractores posteriores a éste, y (4) una gráfica que muestra el número de restricciones satisfechas antes y después del aprendizaje. Posteriormente se presentan los resultados de 10000 procesos de auto-modelado para cada tipo de restricción usando diagramas de caja que muestran la distribución de energía y la satisfacción de restricciones antes y después del aprendizaje. Para obtener estos datos el proceso de auto-modelado se aplicó usando 100 diferentes configuraciones de pesos originales y 100 diferentes configuraciones de estados iniciales por cada configuración de pesos, por lo tanto obteniendo 100 diferentes configuraciones de pesos modificados por cada configuración inicial. Así, desde cada una de las 100 configuraciones de estados iniciales para cada configuración de pesos iniciales la red fue relajada 100 veces (antes del aprendizaje); y para cada configuración de pesos modifica-

dos la red fue relajada 100 veces (después del aprendizaje). Entonces, la red se relajó 1000000 veces antes del aprendizaje y 1000000 después del aprendizaje. Por último se presentan los resultados de la prueba  $t$  de Student hecha para cada tipo de restricción con respecto al número de restricciones satisfechas para saber si el proceso es estadísticamente significativo. En la Sección 7.2 se presenta el análisis de los resultados, y finalmente una discusión sobre las limitaciones del proceso de auto-optimización relacionadas con la forma de la función de energía.

## 7.1. Resultados

El proceso de auto-modelado fue probado en este modelo de red de Hopfield para cada una de las siguientes restricciones en las conexiones:

- **Restricciones Aleatorias Simétricas:** cada conexión inicial  $\alpha_{ii}$  toma un valor aleatorio positivo o negativo con igual probabilidad utilizando una distribución aleatoria uniforme entre  $[-1, -0.5]$  o entre  $[0.5, 1]$ , respectivamente. Por su parte, cada conexión inicial  $\alpha_{ij}$ , con  $i \neq j$ , toma un valor aleatorio utilizando una distribución aleatoria uniforme entre  $[-0.1, 0.1]$
- **Restricciones Aleatorias Asimétricas:** cada conexión inicial  $\alpha_{ii}$  toma un valor aleatorio positivo o negativo con igual probabilidad utilizando una distribución aleatoria uniforme entre  $[-1, -0.5]$  o entre  $[0.5, 1]$ , respectivamente. Por su parte, cada conexión inicial  $\alpha_{ij}$ , con  $i \neq j$ , toma un valor de  $-0.1$  o  $0.1$  con igual probabilidad, y se le suma un valor entre  $[-0.01, 0.01]$  utilizando una distribución aleatoria uniforme.
- **Restricciones Modulares Simétricas:** cada conexión inicial  $\alpha_{ij}$  toma el valor  $|\alpha_{ij}| = 1$  si  $\lfloor \frac{i}{k} \rfloor = \lfloor \frac{j}{k} \rfloor$ ; de otro modo  $|\alpha_{ij}| = 0.01$ . En este trabajo,  $k = 3$  y  $\alpha_{ij} > 0$  con probabilidad 0.8.
- **Restricciones Modulares Asimétricas:** a cada elemento de una matriz con restricciones modulares simétrica se le suma un valor aleatorio utilizando una distribución aleatoria uniforme entre  $[-0.001, 0.001]$ .

Las matrices de pesos tienen conexiones auto-recurrentes sin más condiciones en los signos que los especificados en cada tipo de restricción, es decir durante el aprendizaje puede cambiar el signo de la conexión. En este trabajo se utilizaron los siguientes valores:

- Número de neuronas:  $N = 30$
- Tiempo de actualización:  $= 500$
- Paso de integración:  $\tau = 0.01$
- Número de relajaciones:  $r = 1000$
- La ganancia es definida utilizando una distribución aleatoria uniforme tal que  $g_i \in [10, 20]$ .
- Tasa de aprendizaje:
  - $\delta_{RA}$  Simétricas  $= 0.001$
  - $\delta_{RM}$  Simétricas  $= 0.001$
  - $\delta_{RA}$  Asimétricas  $= 0.0005$
  - $\delta_{RM}$  Asimétricas  $= 0.0005$

Las tasas de aprendizaje  $\delta$  fueron definidas experimentalmente después de asegurar que el proceso de auto-modelado funcionaba correctamente en varias ejecuciones individuales usando las restricciones definidas. La actualización de los pesos se realiza al final de cada periodo de relajación utilizando la ecuación (4.6). Para contabilizar el número de restricciones satisfechas se consideró que una restricción se satisface si  $\alpha_{ij} s_i s_j > 0$ .



### 7.1.1. Ejemplo ilustrativo de un proceso de auto-modelado

#### Restricciones Aleatorias Simétricas

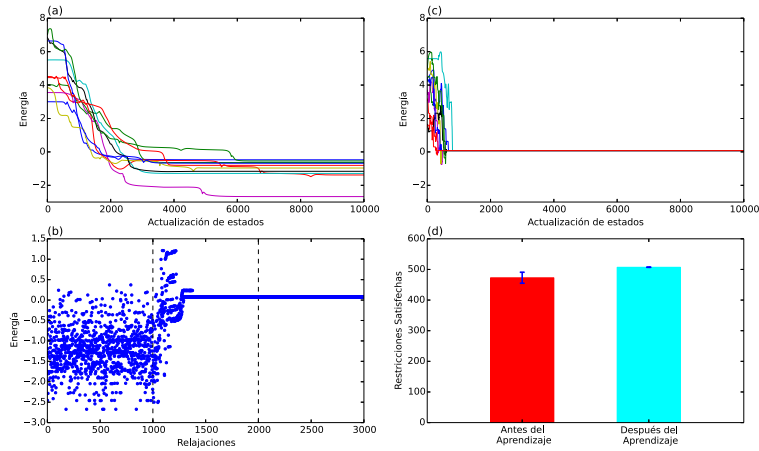


Figura 7.1: Auto-modelado en redes con restricciones aleatorias simétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relaciones 1-1000), durante el aprendizaje (relaciones 1000-2000), y después del aprendizaje (relaciones 2000-3000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul). Las barras de error representan un error estándar.

#### Restricciones Aleatorias Asimétricas

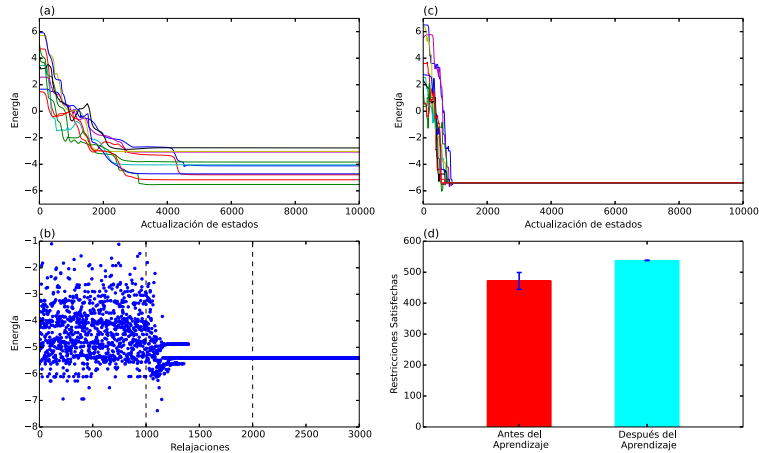


Figura 7.2: Auto-modelado en redes con restricciones aleatorias asimétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relaciones 1-1000), durante el aprendizaje (relaciones 1000-2000), y después del aprendizaje (relaciones 2000-3000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul). Las barras de error representan un error estándar.

## Restricciones Modulares Simétricas

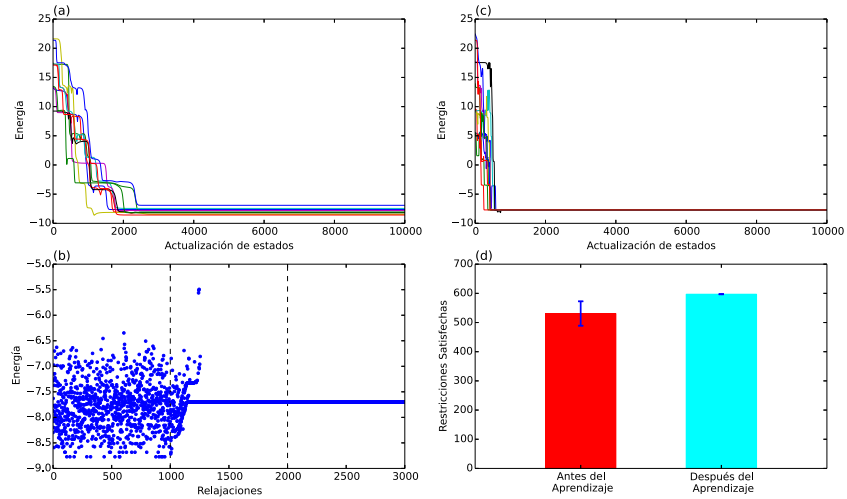


Figura 7.3: Auto-modelado en redes con restricciones modulares simétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relajaciones 1000-2000), durante el aprendizaje (relajaciones 2000-3000), y después del aprendizaje (relajaciones 3000-10000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relajaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul).

## Restricciones Modulares Asimétricas

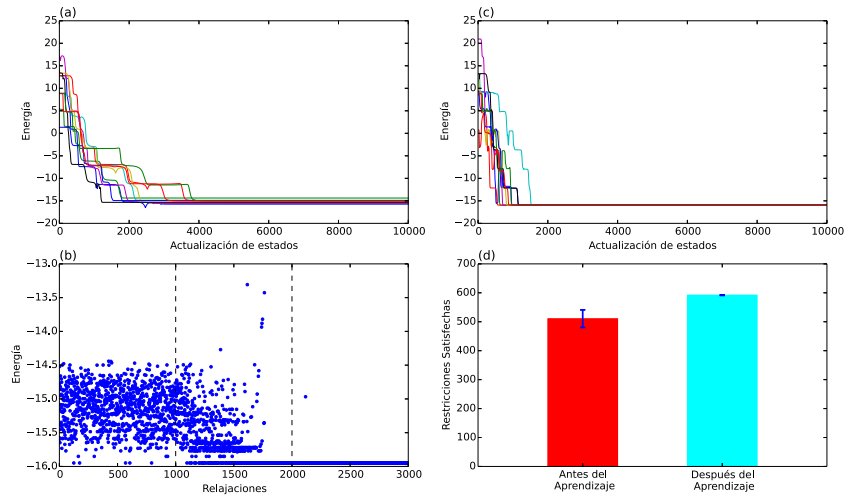


Figura 7.4: Auto-modelado en redes con restricciones modulares asimétricas. (a) Ejemplos de 10 trayectorias antes del aprendizaje; (b) atractores visitados antes del aprendizaje (relajaciones 1-1000), durante el aprendizaje (relajaciones 1000-2000), y después del aprendizaje (relajaciones 2000-3000); (c) ejemplos de 10 trayectorias después del aprendizaje; y (d) promedio de relajaciones satisfechas antes del aprendizaje (barra roja) y después del aprendizaje (barra azul).

### 7.1.2. Resultados de 10000 procesos de auto-modelado Restricciones Aleatorias

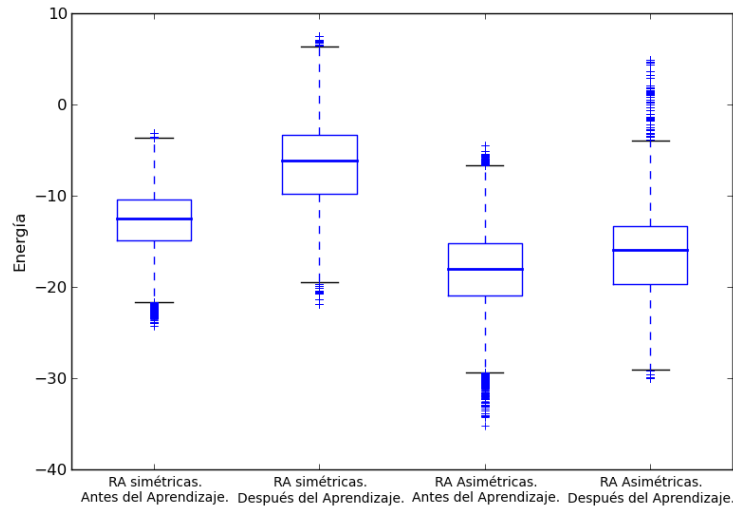


Figura 7.5: Distribución de la energía antes y después del proceso de auto-modelado en redes con restricciones aleatorias simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relajaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relajaciones.

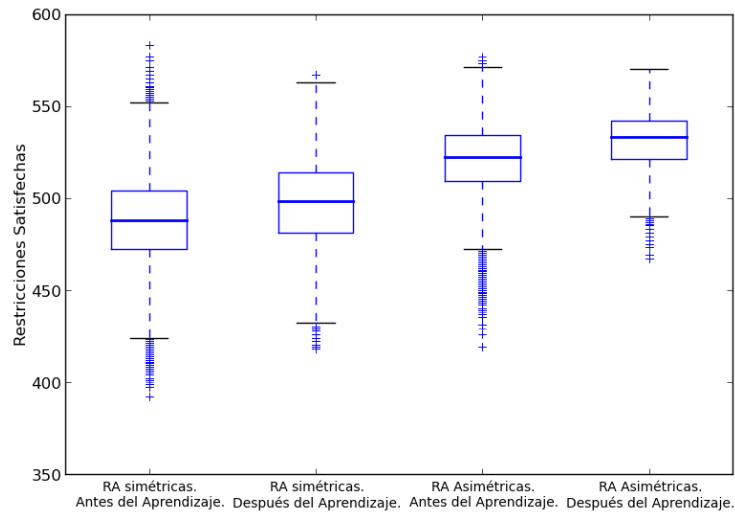


Figura 7.6: Restricciones satisfechas antes y después del proceso de auto-modelado en redes con restricciones aleatorias simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relajaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relajaciones.

## Restricciones Modulares

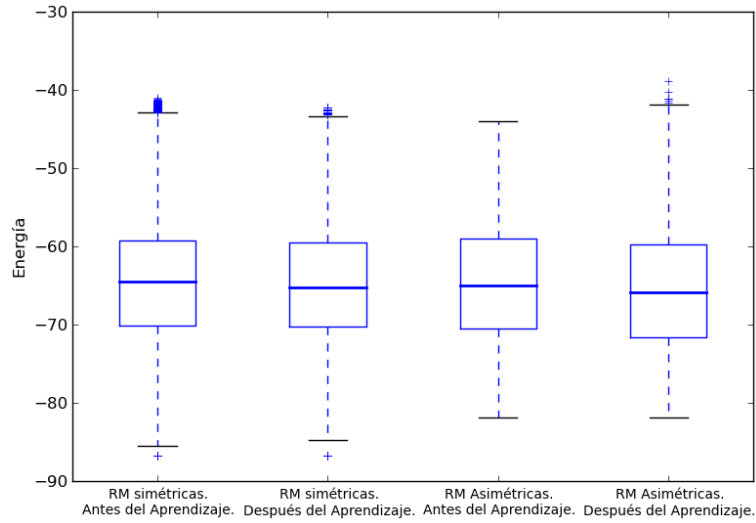


Figura 7.7: Distribución de la energía antes y después del proceso de auto-modelado en redes con restricciones modulares simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relajaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relajaciones.

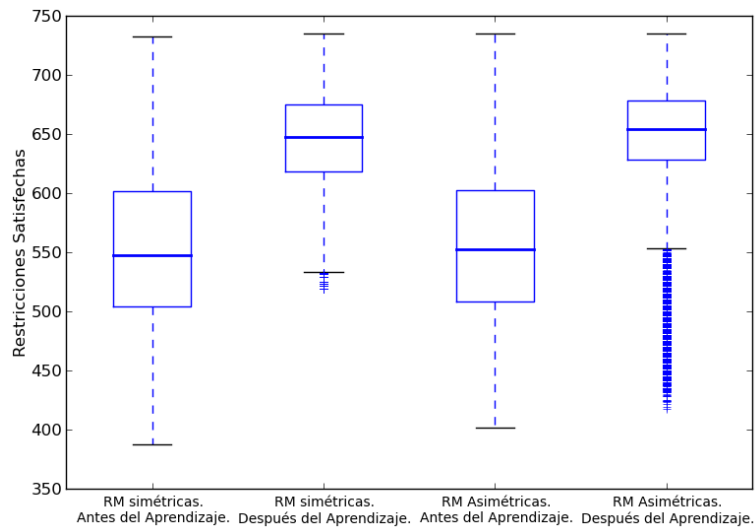


Figura 7.8: Restricciones satisfechas antes y después del proceso de auto-modelado en redes con restricciones modulares simétricas (izquierda) y asimétricas (derecha). La gráfica muestra los datos de 100 relajaciones hechas antes y después de cada proceso de auto-modelado, es decir 1000000 relajaciones.

### 7.1.3. Prueba $t$ de Student

A continuación se muestran los resultados obtenidos de esta prueba estadística para saber si la diferencia entre los resultados antes y después del aprendizaje es estadísticamente significativa. La pregunta que se trata de resolver es: ¿el proceso de auto-modelado permite encontrar mejores soluciones al problema de satisfacción de restricciones en una red de Hopfield de tiempo discreto y estado discreto? Sea la media de las 100 relajaciones desde cada una de las 100 configuraciones de estados iniciales para una configuración de pesos iniciales. El conjunto de datos  $AA$  está definido por las medias de las medias calculadas de la forma anterior para cada una de las 100 configuraciones de pesos iniciales. Ahora, sea la media de las 100 relajaciones para cada una de las 100 configuraciones de pesos modificados obtenida a partir de una de las 100 configuraciones de pesos iniciales. El conjunto de datos  $DA$  está definido por las medias de las medias calculadas de la forma inmediata anterior para cada una de las 100 configuraciones de pesos iniciales. Por lo tanto, si se define la diferencia  $d = DA - AA$  para calcular los estadísticos, la hipótesis nula es  $H_0 : \mu_d \leq 0$ , y la hipótesis alternativa es  $H_a : \mu_d > 0$ , donde  $\mu_{DA}$  indica la media de  $DA$  y  $\mu_{AA}$  indica la media de  $AA$ .

#### Restricciones Aleatorias Simétricas

- Media: 9.712518
- Desviación estándar: 14.0252225269
- p:  $2.609184978575 \times 10^{-10}$

#### Restricciones Aleatorias Asimétricas

- Media: 10.232578
- Desviación estándar: 11.1228444279
- p:  $3.838455607395 \times 10^{-15}$

#### Restricciones Modulares Simétricas

- Media: 90.18129
- Desviación estándar: 17.6035296867
- p:  $3.663404452745 \times 10^{-73}$

#### Restricciones Modulares Asimétricas

- Media: 90.045456
- Desviación estándar: 20.7892358641
- p:  $2.957640902045 \times 10^{-66}$

## 7.2. Análisis

El proceso de auto-modelado en redes de Hopfield continuas no funciona como se esperaba con respecto a la energía. Esto se aprecia en la gráfica (b) de las Figuras 7.1, 7.2, 7.3, y 7.4. No obstante, en la gráfica (d) se observa que la cantidad de restricciones satisfechas ( $\alpha_{ij}V_iV_j > 0$ ) es mayor después del aprendizaje. La discrepancia entre la minimización de la energía y el número de restricciones resueltas se observa de manera estadística en las Figuras 7.5 y 7.6, para el caso de restricciones aleatorias, y en las Figuras 7.7 y 7.8, para el caso de restricciones modulares. Al igual que en la sección anterior, esto se debe a la dualidad de la dinámica del proceso. En lo relativo a la dinámica de la red, el término integral de la función de energía en la ecuación (4.9) causa la existencia de mínimos dentro del hipercubo del espacio de estados (Joya et al., 1997). De acuerdo con Hopfield (1984), una red neuronal de Hopfield continua nunca podrá alcanzar estados estables en los vértices del hipercubo debido a que el término integral aumenta sin límites mientras  $V_i$  se aproxima a  $\pm 1$ , es decir, los atractores en los vértices son asintóticamente estables. El valor del término integral podría estar acotado si la función sigmoide está acotada (Falong & Yan-Da, 1994) o su límite puede depender de la forma de la función (Joya et al., 1997). Adicionalmente, se pueden utilizar diferentes valores para la ganancia  $g_i$  y el sesgo  $\theta_i$  en la función sigmoide de la ecuación (2.7), los cuales inciden en la dinámica de la red (por ejemplo, la ecuación (4.8)). El valor de la integral  $\int_0^{V_i} \sigma^{-1}(\xi)d\xi$  puede variar dependiendo de  $g$  (véase la Figura 7.9) y, a diferencia del modelo de Hopfield anterior, el resultado de la integral puede ser negativo gracias a  $\theta_i$  (véase la Figura 7.10). Entonces, aunado al hecho de que los pesos están limitados en el intervalo  $[-1, 1]$ , se puede deducir que el término integral es en general más grande que el primer término de la ecuación (4.9), cuya minimización refleja la satisfacción de restricciones. Por lo tanto, la cantidad de restricciones satisfechas no es representada correctamente con la disminución de la energía.

Por su parte, el aprendizaje Hebbiano usado para reforzar los atractores de la red provoca que éstos sean ‘desplazados’ hacia las esquinas del hipercubo del espacio de fase (Zhu, 2008). De acuerdo con Bibitchkov et al. (2002), un ajuste global de pesos en el modelo continuo de Hopfield es imposible, ya que conduce a cambios en la estabilidad de los estado de equilibrio. Entonces, el aprendizaje continuo de los atractores provoca que los pequeños cambios en los pesos haga más probable la saturación de los estados, lo que a su vez incrementa la cantidad de aprendizaje Hebbiano haciendo más probable que se saturen los pesos, y así sucesivamente. Por lo tanto, reforzar los atractores provoca que éstos se muevan hacia las esquinas del hipercubo, lo que incrementa el valor del término integral. El término integral de la ecuación de energía no es siempre positivo pero, dado que  $g_i$  y  $\theta_i$  son constantes, es más probable que aumente de valor con el tiempo debido a que  $s_i$  crecerá conforme los pesos se actualicen y en general  $|s_i| > |\theta_i|$ , lo que tenderá a saturar a  $V_i$ .

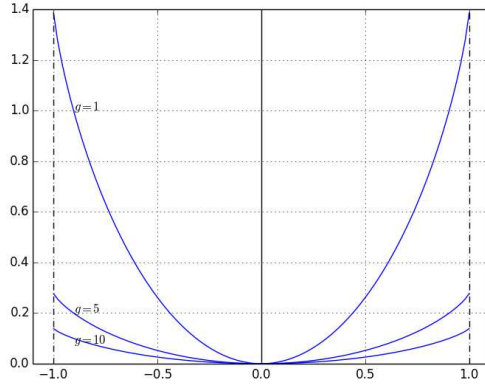


Figura 7.9: Gráfica de  $\int_0^{V_i} \sigma^{-1}(\xi) d\xi$  para diferentes valores de  $g_i$

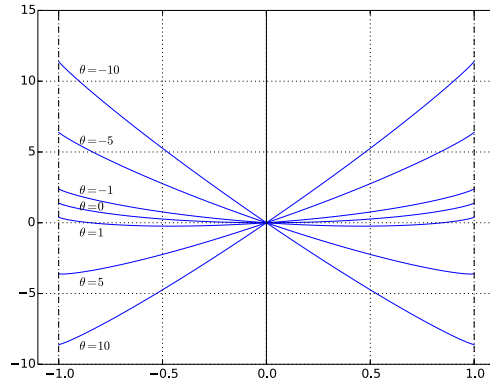


Figura 7.10: Gráfica de  $\int_0^{V_i} \sigma^{-1}(\xi) d\xi$  para diferentes valores de  $\theta_i$

Con base en lo anterior, se puede comprender el comportamiento en la gráfica (b) (relajaciones 1000-2000). Durante el aprendizaje los atractores se mueven hacia las esquinas del hipercubo y la energía del término integral aumenta. Entonces, dado que hay mayor probabilidad de que el valor de este término sea más grande que el valor del término de suma discreta, la energía de los atractores tiende incrementarse. Aunque puede haber casos en los cuales  $\theta_i$  y  $g_i$  compensen correctamente el incremento de  $s_i$  debido al incremento de los pesos de tal forma que el término integral no tenga un impacto tan fuerte en la energía (por ejemplo, la gráfica (b) en la Figura 7.4). Por su parte, la competencia entre los atractores debido al proceso de auto-modelado provoca que sólo uno permanezca en el espacio de estados (al menos en el caso ideal), el cual en general estaría casi en las esquinas del hipercubo. En este trabajo, las salidas de las neuronas  $V_i$  alcanzan los valores extremos  $\pm 1$  debido a limitaciones computacionales. De igual

forma, el término integral no crece sin límite debido a que el método usado para calcular la integral le impone un valor límite cuando  $V_i = \pm 1$ . En general, los atractores a los que la red converge durante la auto-modelación no son parte del conjunto original, no porque la probabilidad de converger en ellos sea mínima, sino porque el cambio en los pesos redefine completamente la topología de la red. No obstante, en promedio los nuevos atractores resuelven más restricciones con las conexiones originales que aquellos en los que la red es capaz de converger.

En este trabajo, la actualización de los pesos se realiza al final del periodo de relajación de la red cuando la red se encuentra en un estado estable. En las pruebas realizadas se utilizaron 500 unidades de tiempo, pero sólo se presentan 100 unidades de tiempo en la gráfica (a) y (c) de las Figuras para una mejor visualización del transitorio. Entonces, se puede pensar que la red pasa suficiente tiempo en el atractor como para poder dividir la cantidad de aprendizaje aplicada al final del proceso en cada paso de tiempo. No obstante, dado que la cantidad de aprendizaje Hebbiano no es constante y que la energía no decrece siempre al actualizar la dinámica de la red, no es posible saber que tipo de resultados pueden ocurrir Dong y Hopfield (1992). Por lo tanto, no se puede asegurar que el efecto global sea el mismo en este modelo. La gráfica (c) muestra que el transitorio a un estado estable es más rápido después del proceso de auto-modelado, y puede presentar aumentos en la energía original,  $E^O$ , debido a la existencia de conexiones auto-recurrentes y a la presencia del término integral.

En este modelo la cantidad de aprendizaje es simétrico, esto es  $\Delta\omega_{ij} = \Delta\omega_{ji}$ . Entonces, los pesos modificados tenderán a tomar el mismo signo con el tiempo, aún cuando las conexiones iniciales sean asimétricas o de signos diferentes. La débil asimetría de la matriz de conexiones hace más probable la presencia de atractores de punto fijo, pero no se puede descartar la presencia de otros tipos de conjuntos límite ya que pequeñas variaciones en los parámetros de la red producen grandes cambios en su dinámica (Beer, 1995). Dado que el proceso converge a un atractor de punto fijo, se puede pensar que en general el aprendizaje Hebbiano reduce la complejidad de la dinámica de la red, pero este tema merece un análisis topológico, dinámico y funcional más amplio, como el realizado por Siri, Berry, Cessac, Delord, y Quoy (2008) para el modelo discreto.

### 7.2.1. Auto-modelado como auto-optimización en redes de Hopfield de tiempo continuo y estado continuo

Es importante aclarar el rol de la función de energía en problemas de optimización. De acuerdo con lo explicado en la Sección 3.3, dado un problema de optimización, se tiene que encontrar una función objetivo o de costo que lo describa, después se diseña una red neuronal de Hopfield cuya función de energía tenga los mismos óptimos que la función objetivo, de tal forma que las configuraciones de estabilidad de la red correspondan a soluciones del problema (Joya et al., 1997). La función de energía de la red de Hopfield mostrada en la ecuación 2.2 es diferente de la función de energía utilizada en el proceso de auto-modelado mostrada en la ecuación 4.3. Esta última es considerada como la función objetivo del problema, cuya diferencia con la función de energía solo radica en la constante  $\frac{1}{2}$ . Entonces, la función de energía de la red se puede asociar con la función objetivo del problema. Esto es claro cuando Watson et al. (2011) comentan que los pesos iniciales representan una problema *weighted - Max - 2 - SAT* donde  $\alpha_{ij}$  denota la importancia de satisfacer la restricción. Una red de Hopfield con-



tinua también puede ser usada para resolver este tipo de problemas (Ettaouil, Loqman, Haddouch, & Hami, 2013).

Es común el argumento de que en general el modelo continuo de Hopfield es superior al discreto al ser menor probable que converja en óptimos locales, debido a su superficie más suave (Du & Swamy, 2014). No obstante, esto podría depender del problema en específico que se requiera resolver, no hay forma de asegurarlo de manera general. De acuerdo con Vidyasagar (1998), hay que diferenciar entre la propiedad de equilibrio y la propiedad de atractividad. La propiedad de equilibrio requiere que si el estado inicial de la red es un óptimo local de la función de energía, entonces la red permanece en dicho estado. La propiedad de atractividad requiere que si el estado inicial de la red está suficientemente cerca (en términos de la distancia de Hamming) de un óptimo local, entonces la trayectoria resultante converge a dicho óptimo. Ha sido demostrado que para muchos problemas de optimización en la literatura, no todos los óptimos locales de una red de Hopfield discreta tienen la propiedad de atractividad. No obstante, si se utiliza una red de Hopfield continua para el mismo problema, cada óptimo local tienen la propiedad de atractividad, es decir un radio de atracción diferente de cero. Además, en una cantidad finita de tiempo, casi todas las trayectorias de soluciones entran en un ortante (análogo del cuadrante) de  $[-1, 1]^N$  que contiene un máximo local de la función de energía y permanece en ese ortante (Vidyasagar, 1998). Presumiblemente, la propiedad de atracción del modelo continuo aumenta la probabilidad de que la red converja a atractores de baja energía es mayor que en el caso del modelo discreto. Por lo tanto, el proceso de auto-modelo tendría que ser capaz de encontrar mejores soluciones.

Con base en los resultados y la discusión aquí presentada se puede concluir que el proceso de auto-modelado no es útil para encontrar el mínimo de la función de energía de la ecuación (4.5) o (4.9), ya que el proceso no minimiza el término integral (término de “costo” en la ecuación (3.6)). Hablando en término de energía, el proceso de auto-modelado es mejor para resolver problemas cuya función objetivo tenga una forma cuadrática, cuya región factible tenga una geometría de hipercubo y cuyos óptimos estén en las esquinas de dicha región. Por lo tanto, la red de Hopfield de tiempo discreto y estado discreto, así como la red de Hopfield de tiempo continuo y estado continuo, sería mejor usarlas en el régimen de alta ganancia, en la cual los atractores de la red se encuentran en las esquinas del hipercubo del espacio de fase y el valor del término integral es mínimo. No obstante, el proceso de auto-modelado se puede considerar exitoso en términos de la cantidad de restricciones satisfechas, incluso si la función de activación tiene un término de sesgo.

## Capítulo 8

# Conclusiones

El proceso de auto-modelado permite a la red converger en óptimos tales que resuelvan un mayor número de restricciones en comparación con aquellos a los cuales la red es capaz de converger usualmente sólo a través de su relajación. Esto se debe a que la memoria asociativa permite a la red generalizar idealizando patrones almacenados y/o creando nuevas combinaciones de sub-patrones aprendidos (Watson et al., 2011a). Por lo que reforzar óptimos locales a través del aprendizaje también refuerza mejores óptimos, permitiendo que el dominio de atracción de estos últimos supere al de los primeros, ya que los atractores de baja energía (o aquellos que resuelven un mayor número de restricciones) tienden a tener un amplio dominio de atracción, y transformando el espacio de estados hasta que permanezca un sólo atractor. La combinación de una tasa pequeña de aprendizaje y la aleatorización de la configuración de estados permite reforzar múltiples óptimos locales, lo que hace más probable que el atractor final del proceso sea un óptimo global en lugar de converger prematuramente en un sub-óptimo.

En este trabajo se encontró que el proceso de auto-modelado no es útil para resolver el problema de Mínima Energía (también conocido como *Ground State Problem*) en redes de Hopfield con función de activación continua, ya que el atractor al final del proceso es solamente óptimo con respecto a las restricciones satisfechas. Esto se debe a que la dinámica de las neuronas depende de las conexiones en la red, que a su vez varían en el tiempo como una función de la dinámica de los nodos, lo que lleva a un incremento en el nivel de saturación promedio de las neuronas, y por lo tanto a un aumento del valor del término integral en las ecuaciones (4.5) y (4.9). Por lo tanto, este proceso de auto-optimización sería solamente capaz de encontrar óptimos de dichas funciones de energía cuando la red se encuentra en un régimen de alta ganancia, es decir cuando el término integral es despreciable. Esto también se puede entender cuando se observa que el primer término en ambas funciones es simplemente la suma de restricciones satisfechas. El aprendizaje Hebbiano tiende a desarrollar conexiones simétricas (o quasi-simétricas) y modulares (Hua et al., 1999), así como llevar a un rápido decaimiento de la complejidad dinámica debido a la saturación de las neuronas (Siri et al., 2008), lo cual es importante para el desarrollo de atractores de punto fijo en redes neuronales recurrentes. Entonces, el proceso de auto-modelado es capaz de explotar la estructura predefinida o emergente de la red, por lo que en general se encuentran mejores resultados en

matrices de pesos con restricciones modulares.

En los resultados obtenidos no se encontró una correlación positiva entre el valor de la energía y la cantidad de restricciones satisfechas. No obstante, se demostró que el proceso de auto-modelado puede ser aplicado en redes de Hopfield de tiempo continuo y estado continuo con conexiones asimétricas y auto-recurrentes. Esto se logró relajando las restricciones del proceso original, es decir permitiendo las conexiones asimétricas y auto-recurrentes, utilizando una función de activación continua, y por último una dinámica temporal continua. En los tres modelos de la red de Hopfield, con base en los resultados obtenidos con la prueba  $t$  de Student, se puede concluir que existe suficiente evidencia en contra de la hipótesis nula  $H_0$  dado que los valores de  $p$  son menores que 0.05 para todas las diferentes restricciones en las conexiones. Esto es, asumiendo que el proceso de auto-modelado no aumenta la capacidad de la red de encontrar atractores globalmente óptimos con respecto a la satisfacción de restricciones, la probabilidad de obtener la cantidad de restricciones satisfechas que se han obtenido después del aprendizaje es muy baja. Entonces, existe suficiente evidencia para argumentar en favor de la hipótesis alternativa  $H_a$ . Por lo tanto, la diferencia entre las medias es estadísticamente significativas, y una consecuencia del proceso de auto-modelado, no una casualidad. Por lo tanto, el proceso de auto-modelado permite encontrar en general mejores soluciones al problema de satisfacción de restricciones.

## 8.1. Trabajo a futuro

Una forma de extender los resultados presentados en el presente trabajo es analizando las siguientes características:

- Grado de asimetría. Es posible medir el éxito del proceso de auto-modelado cambiando el grado de asimetría de las matrices de conexión. La asimetría se mide con el coseno del ángulo entre la matriz de pesos vectorizada y la matriz de pesos vectorizada transpuesta; la simetría existe cuando el coseno es igual a 1 (Hua et al., 1999).
- Modularidad emergente en redes con restricciones aleatorias. De acuerdo con Hua et al. (1999), la modularidad es una consecuencia de la correlación de los estados de activación, por lo que es importante saber que tipo de modularidad, relativo al dominio específico del fenómeno modelado, surge a partir del aprendizaje Hebbiano.
- Energía de las conexiones intramodulares e intermodulares. Sería interesante saber el valor de la energía intermodular en matrices con restricciones modulares, asimétricas y con auto-recurrencia, ya que Watson et al. (2011b) encontraron que dicha energía aumenta en el caso simétrico. Esto se debe a que las restricciones intermodulares son violadas en favor de satisfacer restricciones intramodulares relativamente fuertes.

Una importante contribución sería estudiar el proceso de auto-modelado en redes neuronales de Hopfield de tiempo continuo y estado continua con entradas externas, ya que este modelo presenta una gran variedad de conjuntos límite. Además, cuando la dinámica de la red y la del aprendizaje evolucionan al mismo tiempo en este modelo con pesos iniciales asimétricos, diferentes dinámicas

pueden ocurrir (Dong & Hopfield, 1992). Incluso un modelo continuo de Hopfield sin entradas pero con una fuerte asimetría en sus conexiones presenta una amplia variedad de dinámicas. Watson et al. (2011a) opinan que un modelo de actualizaciones continua en el aprendizaje, en lugar de al finalizar el periodo de relajación, podría ser más natural cuando los atractores no son de punto fijo. Esto también cambiaría la forma de calcular el tiempo de relajación de la red. En este trabajo se definió un tiempo límite para asegurar que la red converja en un atractor. Este tiempo se estimó a través de pruebas donde el criterio de convergencia fue la ausencia de cambio en la energía durante el mayor tiempo posible una vez que la red alcanza el estado estable. Si se usa una regla Hebbiana como la ecuación (4.2), y dado que se especula que la complejidad dinámica se tiende a reducir durante el aprendizaje Hebbiano, se podría definir otro criterio de convergencia como el presentado por Golos et al. (2016), ya que no existe un consenso general sobre la validez de la función de energía en matrices con fuerte asimetría (Molnár & Ercsey-Ravasz, 2013). En aquel trabajo, la actualización de la dinámica se detiene con base en un umbral, el cual indica que la diferencia del potencial promedio de los nodos no ha cambiado en un cierto tiempo.

Por último, si el proceso es exitoso en modelos dinámicos no autónomos como un modelo continuo de Hopfield con entradas externa, se puede investigar el efecto del proceso de auto-modelado en un modelo de robótica evolutiva. Esto también nos podría dar una idea del significado físico de la función ya que aún se desconoce (Z. Wang, Liu, & Zheng, 2016), pese al famoso isomorfismo de la red Hopfield con el modelo de Ising.

# Referencias

- Amit, D. J., Gutfreund, H., & Sompolinsky, H. (1985). Spin-glass models of neural networks. *Phys. Rev. A*, *32*, 1007–1018. doi: 10.1103/PhysRevA.32.1007
- Ansari, N., & Hou, E. (1997). *Computational intelligence for optimization* (1st ed.). US:Springer.
- Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, *3*(4), 469-509.
- Beer, R. D. (2000). Dynamical approaches to cognitive science. *Trends in Cognitive Sciences*, *4*(3), 91–99. doi: 10.1016/S1364-6613(99)01440-0
- Berg, J. v. d. (1996). The most general framework of continuous hopfield neural networks. En *Proceedings of international workshop on neural networks for identification, control, robotics and signal/image processing* (p. 92-100). doi: 10.1109/NICRSP.1996.542749
- Bibitchkov, D., Herrmann, J. M., & Geisel, T. (2002). Pattern storage and processing in attractor networks with short-time synaptic dynamics. *Network: Computation in Neural Systems*, *13*(1), 115 – 129.
- Blynel, J., & Floreano, D. (2002). Levels of Dynamics and Adaptive Behavior in Evolutionary Neural Controllers. En B. Hallam, D. Floreano, J. Hallam, G. Hayes, & J.-A. Meyer (Eds.), *From Animals to Animats 7*.
- Cliff, D. (1991). Computational neuroethology: A provisional manifesto. En J.-A. Meyer & S. Wilson (Eds.), *From animals to animats 3: Proceedings of the first international conference on the simulation of adaptive behavior* (pp. 29–39). MIT Press.
- Dayan, P., & Abbott, L. F. (2000). Plasticity and learning. En *Theoretical neuroscience. Computational and mathematical modeling of neural systems* (pp. 51 – 99). MIT Press.
- Dong, D. W., & Hopfield, J. J. (1992). Dynamic properties of neural networks with adapting synapses. *Network: Computation in Neural Systems*, *3*(3), 267-283. doi: 10.1088/0954-898X\3\3\002
- Du, K.-L., & Swamy, M. N. S. (2014). *Neural networks and statistical learning* (1.<sup>a</sup> ed.). Springer-Verlag London.
- Ettaouil, M., Loqman, C., Haddouch, K., & Hami, Y. (2013). Maximal constraint satisfaction problems solved by continuous hopfield networks. *WSEAS Transactions on Computer*, *12*(2), 29–40.
- Fa-Long, L., & Yan-Da, L. (1994). Brief communication a theorem concerning the energy function of hopfield continuous-variable neural networks. *International Journal of Electronics*, *76*(3), 443-446. doi: 10.1080/00207219408925941
- Feng, G., & Douligeris, C. (2001). The convergence and parameter relations-

- hip for discrete-time continuous-state hopfield networks. En *International joint conference on neural networks, 2001. proceedings*. (Vol. 1, pp. 376–381). doi: 10.1109/IJCNN.2001.939049
- Froese, T. (2013). Can altered states of consciousness be adaptive? two proofs of concept. En *Segundo coloquio internacional de ciencias cognitivas*. Cuernavaca, Mexico: Universidad Autónoma del Estado de Morelos.
- Froese, T., Gershenson, C., & Manzanilla, L. R. (2014). Can government be self-organized? a mathematical model of the collective social organization of ancient teotihuacan, central mexico. *PLOS ONE*, *9*(10), 1–14. doi: 10.1371/journal.pone.0109966
- Fyfe, C. (2005). Background. En *Hebbian learning and negative feedback networks* (pp. 11 – 29). Springer.
- Gerstner, W. (2016). Hebbian learning and plasticity. En M. Arbib & J. Bohniuto (Eds.), *From neuron to cognition via computational neuroscience* (p. 199-218). MIT Press Cambridge.
- Golos, M., Jirsa, V., & Daucé, E. (2016). Multistability in large scale models of brain activity. *PLOS Computational Biology*, *11*(12), 1-32.
- Gurney, K. (1997). *An introduction to neural networks*. CRC press.
- Haykin, S. (2009). *Neural networks and learning machines* (3rd ed.). Pearson.
- Hebb, D. (1949). *The organization of behavior: A neuropsychological theory*. New York: Wiley.
- Hérault, L. (2005). Neural networks without training for optimization. En *Neural networks: Methodology and applications* (pp. 443–490). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/3-540-28847-3\\_8
- Hoinville, T., Siles, C. T., & Hénaff, P. (2011). Flexible and multistable pattern generation by evolving constrained plastic neurocontrollers. *Adaptive Behavior*, *19*(3), 187-207.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, *79*(8), 2554-2558.
- Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*.
- Hopfield, J. J., & Tank, D. W. (1985). “neural” computation of decisions in optimization problems. *Biological Cybernetics*, *52*(3), 141–152.
- Hua, S. E., Houk, J. C., & Mussa-Ivaldi, F. A. (1999). Emergence of symmetric, modular, and reciprocal connections in recurrent networks with hebbian learning. *Biological Cybernetics*, *81*(3), 211–225. doi: 10.1007/s004220050557
- Joya, G., Atencia, M. A., & Sandoval, F. (1997). Hopfield neural network applied to optimization problems: Some theoretical and simulation results. En J. Mira, R. Moreno-Díaz, & J. Cabestany (Eds.), *Biological and artificial computation: From neuroscience to technology: International work-conference on artificial and natural neural networks* (pp. 556–565). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/BFb0032515
- Koiran, P. (1994). Dynamics of discrete time, continuous state hopfield networks. *Neural Computation*, *6*(3), 459-468. doi: 10.1162/neco.1994.6.3.459
- Kryzhanovsky, B., & Kryzhanovsky, V. (2008). Binary optimization: On the probability of a local minimum detection in random search. En L. Rut-

- kowski, R. Tadeusiewicz, L. A. Zadeh, & J. M. Zurada (Eds.), (pp. 89–100). Springer Berlin Heidelberg.
- Kuriscak, E., Marsalek, P., Stroffek, J., & Toth, P. G. (2015). Biological context of hebb learning in artificial neural networks, a review. *Neurocomputing*, *152*, 27 - 35. doi: 10.1016/j.neucom.2014.11.022
- Liao, X., Wang, L., & Yu, P. (2007). *Stability of dynamical systems* (1st ed., Vol. 5; A. C. Luo & G. Zaslavsky, Eds.). Elsevier Science.
- Little, W. (1974). The existence of persistent states in the brain. *Mathematical Biosciences*, *19*(1), 101 – 120. doi: 10.1016/0025-5564(74)90031-5
- Looi, C.-K. (1992). Neural network methods in combinatorial optimization. *Computers & Operations Research*, *19*(3), 191 - 208. doi: 10.1016/0305-0548(92)90044-6
- Matsumoto, N., Ide, D., Watanabe, M., & Okada, M. (2005). Synaptic depression enlarges basin of attraction. *Neurocomputing*, *65*, 571 - 577. doi: 10.1016/j.neucom.2004.10.053
- Meunier, D., Lambiotte, R., & Bullmore, E. (2010). Modular and hierarchically modular organization of brain networks. *Frontiers in Neuroscience*, *4*. doi: 10.3389/fnins.2010.00200
- Milner, P. (2003). A brief history of the hebbian learning rule. *Canadian Psychology*, *5* – 9.
- Molnár, B., & Ercsey-Ravasz, M. (2013). Asymmetric continuous-time neural networks without local traps for solving constraint satisfaction problems. *PLOS ONE*, *8*(9), 1-13. doi: 10.1371/journal.pone.0073400
- Molnár, B., Toroczkai, Z., & Ercsey-Ravasz, M. (2012). Continuous-time neural networks without local traps for solving boolean satisfiability. En *13th international workshop on cellular nanoscale networks and their applications* (pp. 1–6). doi: 10.1109/CNNA.2012.6331411
- Mouret, J.-B., & Tonelli, P. (2014). Artificial evolution of plastic neural networks: A few key concepts. En T. Kowaliw, N. Bredeche, & R. Doursat (Eds.), *Growing adaptive machines: Combining development and learning in artificial neural networks* (pp. 251–261). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-55337-0\\_9
- Ogata, K. (2001). *Modern control engineering* (4th ed.). Upper Saddle River, NJ, USA: Prentice-Hall.
- Schürmann, B., Hollatz, J., & Ramacher, U. (1990). Adaptive recurrent neural networks and dynamic stability. En L. Garrido (Ed.), *Statistical mechanics of neural networks: Proceedings of the xth sitges conference sitges* (pp. 49–63). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/3540532676\_40
- Šíma, J., Orponen, P., & AnttiPoika, T. (1999). Some afterthoughts on hopfield networks. En J. Pavelka, G. Tel, & M. Bartošek (Eds.), *Sofsem 99: Theory and practice of informatics: 26th conference on current trends in theory and practice of informatics* (pp. 459–469). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/3-540-47849-3\\_34
- Siri, B., Berry, H., Cessac, B., Delord, B., & Quoy, M. (2006). Topological and dynamical structures induced by hebbian learning in random neural networks. En *International conference on complex systems*. MA, USA.
- Siri, B., Berry, H., Cessac, B., Delord, B., & Quoy, M. (2007). Local learning rules and bifurcations in the global dynamics of random recurrent neural networks. En *European conference on complex systems*. Dresden,

Germany.

- Siri, B., Berry, H., Cessac, B., Delord, B., & Quoy, M. (2008). A mathematical analysis of the effects of hebbian learning rules on the dynamics and structure of discrete-time random recurrent neural networks. *Neural Computation*, *20*(12), 2937-2966. doi: 10.1162/neco.2008.05-07-530
- Siri, B., Quoy, M., Delord, B., Cessac, B., & Berry, H. (2007). Effects of hebbian learning on the dynamics and structure of random networks with inhibitory and excitatory neurons. *Journal of Physiology-Paris*, *101*(1), 136 - 148. doi: 10.1016/j.jphysparis.2007.10.003
- Srivastava, V., Sampath, S., & Parker, D. J. (2014). Overcoming catastrophic interference in connectionist networks using gram-schmidt orthogonalization. *PLOS ONE*, *9*(9), 1-7. doi: 10.1371/journal.pone.0105619
- Tsang, E. (1996). Introduction. En *Foundations of constraint satisfaction* (pp. 1-29). Academic Press Limited.
- Vico, F. J., & Jerez, J. M. (2003). Stable neural attractors formation: Learning rules and network dynamics. *Neural Processing Letters*, *18*(1), 1-16. doi: 10.1023/A:1026258500432
- Vidyasagar, M. (1998). Are analog neural networks better than binary neural networks? *Circuits, Systems and Signal Processing*, *17*(2), 243-270. doi: 10.1007/BF01202855
- Wang, H., Wu, Y., Zhang, B., & Du, K. L. (2011). Recurrent neural networks: Associative memory and optimization. *Information Technology & Software Engineering*, *1*(2). doi: 10.4172/2165-7866.1000104
- Wang, Z., & Fan, H. (2007). Dynamics of a continuous-valued discrete-time hopfield neural network with synaptic depression. *Neurocomputing*, *71*(1), 181 - 190. doi: 10.1016/j.neucom.2007.01.004
- Wang, Z., Liu, Z., & Zheng, C. (2016). Introduction to neural networks. En *Qualitative analysis and control of complex neural networks with delays* (pp. 1-36). Springer-Verlag Berlin Heidelberg.
- Watson, R. A., Buckley, C. L., & Mills, R. (2009). *The effect of hebbian learning on optimization in hopfield networks* (Inf. Téc.). University of Southampton.
- Watson, R. A., Buckley, C. L., & Mills, R. (2011). Optimization in “self-modeling” complex adaptive systems. *Complexity*, *16*(5), 17 - 26. doi: 10.1002/cplx.20346
- Watson, R. A., Buckley, C. L., Mills, R., & Davies, A. (2010). Associative memory in gene regulation networks. En H. Fellerman et al. (Eds.), *Twelfth international conference on the synthesis and simulation of living systems (artificial life xii)* (p. 194-202). MIT Press.
- Watson, R. A., Mills, R., & Buckley, C. L. (2011a). Global adaptation in networks of selfish components: Emergent associative memory at the system scale. *Artificial Life*, *17*(3), 147-166.
- Watson, R. A., Mills, R., & Buckley, C. L. (2011b). Transformation in the scale of behavior and the global optimization of constraints in adaptive networks. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*.
- Watson, R. A., & Szathmáry, E. (2016). How can evolution learn? *Trends in Ecology & Evolution*, *31*(2), 147-157. doi: doi:10.1016/j.tree.2015.11.009
- Woodward, A., Froese, T., & Ikegami, T. (2015). Neural coordination can be enhanced by occasional interruption of normal firing patterns: A self-



- optimizing spiking neural network model. *Neural Networks*, 62, 39 - 46.
- Zarco, M., & Froese, T. (en prensa). Self-modeling in Hopfield neural networks with continuous activation function. En *Procedia Computer Science. 8th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2017*.
- Zhu, J. (2008). The effect of hebbian plasticity on the attractors of a dynamical system. *BMC Neuroscience*, 9(1), P99. doi: 10.1186/1471-2202-9-S1-P99