



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

RED DISCRETA COMO MEDIO DINÁMICO PARA SU EMPLEO
EN COMPUTACIÓN DE RESERVORIO

T E S I S

QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS (COMPUTACIÓN)

P R E S E N T A:

JAVIER GARDUÑO CIMENTAL

TUTOR

DR. CARLOS GERSHENSON GARCÍA
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS
Y EN SISTEMAS - UNAM

MÉXICO D.F., 2016



Universidad Nacional
Autónoma de México



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.

Agradecimientos

Ninguna cosa surge espontáneamente de la nada y este trabajo no es la excepción. Sus raíces más profundas se remontan a todo aquello que ha desembocado en mi propia existencia, educación y formación.

Como no resulta práctico ni posible enumerar tal cantidad de influencias me limitaré, como es costumbre, al conjunto de seres humanos que de forma consciente o no han influido en la creación de este trabajo.

A mis padres, J. Janet Cimental Velasco y Javier Garduño Hernández, de los que siempre he obtenido el mayor de los apoyos y alientos, cimentando desde mi nacimiento los pilares fundamentales de mi carácter.

A Heidy Jojana Oliva Tello, que me acompaña, inspira y motiva, caminando al lado mío.

A todos aquellos amigos que forma informal o no, han contribuido con sus diálogos a esclarecer ideas y conceptos.

Agradezco también a la Universidad Nacional Autónoma de México (UNAM), al Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), al Consejo Nacional de Ciencia y Tecnología (CONACYT). A mi tutor, el Dr. Carlos Gershenson, por su apoyo y la libertad otorgada durante el desarrollo del proyecto; y a mis sinodales por su paciencia, observaciones, guía y comentarios.

Índice general

Agradecimientos	1
Resumen	7
1. Introducción	9
1.1. Estado del arte	9
1.2. Objetivos	10
1.3. Una reflexión	11
1.3.1. En la raíz de la complejidad	16
1.3.2. Donde la complejidad y el aprendizaje se unen	17
1.4. Motivación	17
1.5. Organización del documento	18
2. Computación de reservorio	21
2.1. Antecedentes	21
2.1.1. La inspiración biológica	22
2.1.2. Una pequeña revolución: la retropropagación del error	23
2.1.3. Recurrencia	24
2.2. Sistemas dinámicos	27
2.2.1. Entidades continuas	28
2.2.2. Particularidades de los sistemas discretos	29
2.3. Computación en el borde del caos	29
2.4. Necesidades en el <i>cómputo de reservorio</i>	32
2.4.1. Memoria evanescente	32
2.4.2. Separación puntual	34
3. Redes booleanas	35
3.1. Redes Booleanas Aleatorias	35
3.1.1. El modelo clásico	36

3.2.	Perturbación en redes booleanas	40
3.2.1.	Definiendo la perturbación	40
3.2.2.	Normas	42
3.2.3.	Estímulos	42
3.3.	Ideando un comportamiento deseable	42
3.3.1.	Normas ideales	44
3.3.2.	Memoria evanescente en normas idealizadas	45
3.3.3.	Inspiración biológica	46
3.3.4.	El modelo <i>integrar y disparar</i>	48
3.3.5.	Redes neuronales de pulso	49
3.4.	Redes Booleanas de Pulso	50
3.4.1.	Modelo abstracto	51
3.4.2.	Como red de nodos multivaluados	54
3.4.3.	Como red booleana	57
4.	Mediciones	63
4.1.	Teoría matemática de la comunicación	64
4.1.1.	Entropía	64
4.2.	Auto-organización	65
4.3.	Complejidad	66
4.4.	Homeostasis	66
4.4.1.	Homeostasis y auto-organización	68
4.5.	Autopoiesis	69
4.6.	Separación	70
4.6.1.	Calidad del kernel y rango de generalización	71
4.7.	Evanescencia	73
4.8.	Taza de desvanecimiento	75
5.	Configuración experimental	77
5.1.	Descripción	77
5.2.	Redes booleanas	77
5.3.	Retardos	78
5.4.	Varianza nodal	78
5.5.	Entradas	78
5.6.	Topologías	79
5.6.1.	«Erdős»	79

5.6.2. «K-variable»	80
5.6.3. «Mundo pequeño»	80
5.6.4. «Conexión por distancia»	83
5.7. Unidad de lectura	83
5.8. Tareas	84
5.8.1. Densidad temporal	84
5.8.2. Paridad temporal	84
5.8.3. Paridad de cambio	85
5.9. Entrenamiento y evaluación	86
5.9.1. Algoritmo genético ecléctico	86
5.9.2. Función de evaluación	87
5.10. Software empleado	88
6. Resultados	89
6.1. Puntos de interés	90
6.1.1. Piscinas RBA	90
6.1.2. Piscinas RBP	94
6.2. Resolución de tareas	96
6.2.1. Medidas vs. desempeño	99
6.2.2. Persistencia de la memoria	107
6.3. Defectos	109
7. Conclusiones	113
Glosario	115
Bibliografía	122
Índice de figuras	124
Índice de tablas	125

Resumen

Se propone un modelo de red discreta, capaz de ser implementada como una red booleana clásica, proclive a ser utilizada como medio excitable para un sistema de *computación de reservorio*.

La *computación de reservorio* es un marco de trabajo enfocado en el procesamiento de información con auxilio de un sistema dinámico, llamado *reservorio* o *piscina*, que gracias a su comportamiento es capaz de mapear las entradas que recibe a un espacio de estados que facilita su clasificación, haciendo las veces de una función *kernel* temporal. Se encuentra fuertemente relacionada con las redes neuronales recurrentes, pero no está limitada a las mismas.

Si bien se ha investigado de forma incipiente la viabilidad de Redes Booleanas Aleatorias como piscina, poseen profundas desventajas asociadas a las características propias de su espacio de estados y las transiciones entre los mismos, ya que favorecen un mapeado diferente para entradas idénticas, tan sólo por el momento en que son inyectadas al sistema.

Esta tesis presenta un formalismo para el concepto de «perturbación» aplicado a redes booleanas, y propone un modelo de red booleana con base en él, e inspirado en las Redes Neuronales de Picos (*spiking neural networks*), que favorece mapeados consistentes para entradas similares, dada su propensión a poseer una cuenca de atracción gigante con un atractor puntual.

Se exploran las capacidades computacionales del modelo propuesto y se comparan con las de Redes Booleanas Aleatorias típicas. Ambos modelos se someten a tareas de clasificación y cálculo, a fin de verificar si las predicciones teóricas sobre su desempeño son correctas.

Capítulo 1

Introducción

All the wonders of our universe can in effect be captured by simple rules, yet [...] there can be no way to know all the consequences of these rules, except in effect just to watch and see how they unfold.

«Todas las maravillas de nuestro universo efectivamente pueden ser capturadas por reglas simples, a pesar de que [...] no hay manera de conocer todas las consecuencias de estas normas, salvo contemplándolas y observando cómo se desarrollan».

Stephen Wolfram

1.1. Estado del arte

La *computación de reservorio* (*reservoir computing*)¹ es un marco de trabajo enfocado en el procesamiento de información con auxilio de un sistema dinámico, llamado *reservorio* o *piscina*, que gracias a su comportamiento es capaz de mapear las entradas que recibe a un espacio de estados que facilita su clasificación, haciendo las veces de una función *kernel* temporal. Se encuentra fuertemente relacionada con las redes neuronales recurrentes, pero no está limitada a las mismas.

La idea central fue desarrollada separadamente, casi de forma simultánea, por Jäger (2001) así como Maass, Natschläger y Markram (2002). El primero ideando las llamadas *Máquinas de Estado-Eco* (*Echo-State Machines*) y los segundos gracias a las *Máquinas de Estado Líquido* (*Liquid State Machines*).

La aproximación propuesta hecha en la computación de reservorio nace en la necesidad de aprovechar el poder computacional de las redes recurrentes, que se resisten a los algoritmos de entrenamiento tradicionales, o bien, exigen

¹El tema se verá con más detalle en el capítulo 2.

tiempos extremadamente largos para los mismos. A la vez, son mecanismos ideales para el procesamiento de series temporales, dado que son sistemas que funcionan «a través del tiempo».

Los enfoques iniciales se han extendido, desarrollándose en este mismo campo diversos algoritmos para el entrenamiento y preparación de la piscina, como la *regla de aprendizaje por retropropagación-decorrelación* (Steil, 2006), o el método *Evolino (EVolution of recurrent systems with LINear Output)* (Schmidhuber, Wierstra, Gagliolo & Gomez, 2007), que combina un entrenamiento de los pesos de salida y técnicas evolutivas para realizar ajustes en la piscina.

La viabilidad de las Redes Booleanas Aleatorias como piscina se ha investigado someramente en (Snyder, Goudarzi & Teuscher, 2012) y en (Snyder, Goudarzi & Teuscher, 2013). Sin embargo, a la luz de nuestros conocimientos actuales al respecto, las Redes Booleanas Aleatorias típicas poseen ciertas desventajas asociadas a las características propias de su espacio de estados y las transiciones entre los mismos, ya que favorecen un mapeado diferente para entradas idénticas, tan sólo por el momento en que son inyectadas al sistema. Es decir, no suelen ser invariantes en el tiempo.

Otras propuestas de redes discretas asociadas a la computación de reservorio se encuentran en las *Máquinas de Estado-Eco Cuantizadas (Quantized Echo-State Machines)* (Schrauwen, Büsing & Legenstein, 2009; Büsing, Schrauwen & Legenstein, 2010), que como su nombre lo indica, cuantizan las funciones de activación de los nodos que constituyen a la red.

1.2. Objetivos

El presente trabajo persigue los siguientes objetivos:

- Otorgar un significado más preciso al término «perturbación» para redes booleanas.
- Con base en dicha definición, proponer un modelo de red discreta proclive a ser utilizada como medio dinámico para un sistema de computación de reservorio. Esto implica la búsqueda de un sistema más sencillo de implementar que los utilizados actualmente por dicho paradigma de cómputo, con la pretensión de que el mismo sea realmente viable para dichas tareas. A este modelo se le ha bautizado como Red Booleana de

Pulsos (RBP), un arreglo particular de red booleana clásica sensible a la perturbación, que pretende imitar características de las Redes Neuronales de Picos (*Spiking Neural Networks*). Estos últimos, arreglos que emplean como elemento fundamental a neuronas de picos, en particular, aquellas que siguen el modelo llamado *integrar y disparar*.

- Establecer comparaciones cualitativas y cuantitativas para el desempeño de dicho modelo con respecto a Redes Booleanas Aleatorias tradicionales, en las cuales se inspira.

1.3. Una reflexión

En nuestra búsqueda e indagación por la inteligencia de las máquinas, hemos terminado casi siempre por imponer lo que pensamos es nuestro propio proceder en ellas. Como padres consentidores, incrédulos ante las capacidades de sus hijos aún tiernos, acabamos por colocar en bandeja de plata el mundo a sus pies de forma que, sin esfuerzo, sean capaces de acceder a él y sus beneficios.

Si deseamos que caminen, las tomamos de la mano, señalando oportunamente la senda que deben seguir. Si van a mantener una conversación, pacientemente les indicamos lo que deben decir. Ante un problema determinado, planificamos su actuar y pensar.

En realidad, lo que hacemos con las máquinas es aún más burdo que eso. Con grandes esfuerzos hemos conseguido que sepan discriminar entre un conjunto arbitrario de objetos. Con todo, ese logro se encuentra restringido a circunstancias y entornos controlados. Un cambio no demasiado grande, un paso más allá de tales condiciones, y nuestros ingenios tienden a fallar estrepitosamente.

Los intentos más exitosos al respecto consiguen grandes resultados a un costo que implica una ingente cantidad de datos y potencia de cómputo. Más de la que, quizá, habíamos contemplado hace algunas décadas. Un ejemplo de ello lo podemos encontrar en (Karpathy y col., 2014) para la clasificación de secuencias de video, o en (Girshick, Donahue, Darrell & Malik, 2014) para la clasificación de imágenes fijas, que involucran modelos extraordinariamente grandes, consistentes en redes neuronales compuestas por una gran cantidad de elementos, con una sofisticada y enrevesada disposición entre ellos. Sin embargo, estos tan sólo son algunos casos entre muchos otros.

¿Acaso lo que parece simple a nuestros ojos es en realidad un proceso de alta complejidad? Eso no lo duda nadie. ¿Cómo abordar dicha problemática? Una «desventaja» que concierne a las redes neuronales en particular, y a los sistemas complejos en general, es su cualidad para funcionar como «cajas negras». Entiéndase tal frase como *nuestra incapacidad* para comprender con claridad qué sucede exactamente ahí dentro. Una red neuronal, convenientemente entrenada, es capaz de predecir o clasificar valores con alta precisión. Sin embargo, nunca nos queda claro exactamente cuáles fueron los criterios que el sistema empleó para realizar su labor.

¿Es una falla del sistema?, ¿una debilidad de la metodología empleada? O por el contrario, ¿es una fortaleza? ¿Hasta dónde está justificado querer entender los criterios que utilizó el sistema para clasificar, predecir o calcular? La problemática que acarrea tal pregunta nos traslada a reflexiones filosóficas profundas que no pueden ser desatendidas.

Uno de los ejemplos más simples a este respecto es la discusión a la que puede conducirnos preguntar «¿qué es una silla?». Esta interrogante, de aparente simplicidad, nos revela las limitaciones que tenemos al momento de «atrapar» o «definir» a una entidad en términos capaces de ser expresados de forma sencilla por cualquiera de nosotros.

La RAE define «silla» como un «asiento con respaldo, por lo general con cuatro patas, y en que solo cabe una persona» (RAE, 2014), aunque parece una respuesta válida a primera vista, pronto revela sus carencias. Un sillón la cumple, y ninguno de nosotros diría que es una silla. Siendo el caso, ¿qué pasaría si dos niños pequeños fueran capaces de sentarse en lo que llamamos «silla»? La definición restringida al número de personas tampoco es completamente satisfactoria, y todavía la siguen cumpliendo multitud de objetos que no llamaríamos «silla».

Puestos a analizar las cosas, pronto descubrimos que seríamos capaces de llamar «silla» a cosas que ni siquiera sirven para que un ser humano se siente, como aquellas diminutas que habitan las casas de muñecas utilizadas por nuestros niños cuando juegan. ¿Qué es entonces una silla? Algo seguramente difícil, más no imposible, de reducir a un conjunto de sentencias verbales que la describan. Aún si necesitáramos cientos, miles o millones de ellas.

La simpleza del concepto «silla» es aparente. Se encuentra en función de miriadas de variables, muchas de ellas seguramente anti-intuitivas o absurdas vistas de manera aislada. Tal concepto, como todos los demás que usamos en

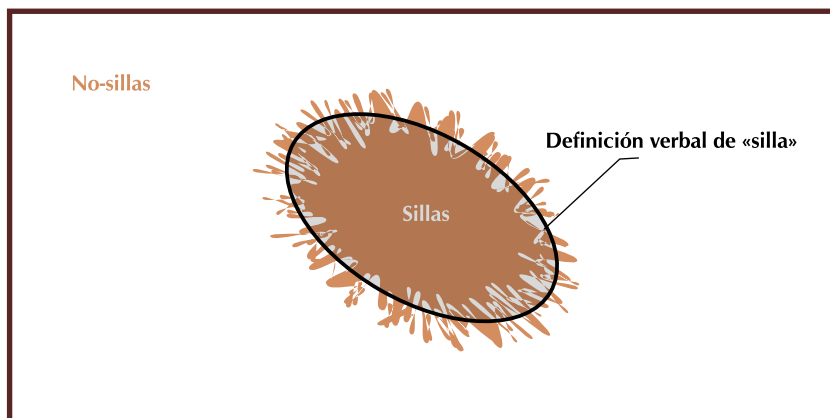


Figura 1.1: El conjunto de objetos reconocidos como sillas (representado aquí como la mancha café), siendo un producto de la compleja interacción entre reglas y conceptos mentales, se resiste a ser completamente descrito por la simpleza de una definición verbal típica, que en sus límites suele dejar fuera multitud de objetos del grupo y colocar dentro algunos que no lo son. La complejidad del concepto es representada en la irregular geometría del conjunto.

la vida diaria, es la consecuencia de una compleja interacción que existe entre otros conceptos, percepciones y pensamientos, que a su vez son la consecuencia de otros tantos. Y así, hasta llegar al nivel más bajo, el asociado a la recepción de los estímulos provenientes del mundo exterior a través de los sentidos.

¿Por qué insistimos en la validez de las definiciones simples que nos creamos? Una posible respuesta quizá se encuentre en la capacidad para racionalizar la información que recibimos o generamos nosotros mismos. Al respecto, se pueden mencionar las investigaciones realizadas en personas con cerebro dividido. Estos individuos, los cuales han sufrido la separación de sus hemisferios cerebrales por medio de la cirugía, presentan una condición que hace considerar seriamente que las dos mitades de su cerebro trabajan por separado, ignorantes cada una de los estímulos y maquinaciones de la otra (Gazzaniga, 1967). Por ejemplo, si es presentada una imagen diferente a cada hemisferio, y luego se solicita al individuo elegir entre un grupo de otras imágenes alguna relacionada con las anteriores, se harán dos elecciones, cada una en correspondencia con las dos mostradas. Al cuestionar al sujeto por sus preferencias, el hemisferio izquierdo (que es quien responde por residir ahí la habilidad del habla), crea una interpretación inventada enteramente por él sobre la razón por la cuál se eligió la figura escogida por el hemisferio derecho. Dicho paciente (o hemisferio) cree sinceramente su propia explicación, que en realidad es creada *a posteriori*, una vez expuesto a la imagen elegida (Gazzaniga, 1995).

Cuando construimos la definición de un objeto, ¿caemos en la misma trampa? ¿Es posible que las descripciones que nos inventamos sean mentiras creadas *a posteriori* por nosotros mismos, que poco o nada tengan que ver con el verdadero entramado mental que les da origen? ¿De qué otra forma podríamos insistir en creer que una silla es un «asiento con respaldo, por lo general con cuatro patas, y en que solo cabe una persona», a pesar de la abrumadora evidencia en contra de la validez de tal afirmación?

Hay un paralelismo conceptual entre las definiciones verbales que dan cuenta de un objeto y las expresiones matemáticas que establecen los límites de una clase, en tareas de clasificación y aprendizaje de máquina. Expresiones simples establecerán límites sencillos para las clases, que tal vez no capten los matices y sutilezas propios de los objetos que se trate, o de los estímulos provenientes de ellos que la máquina recibe, o de esas cosas que a cierta persona «jamás se le habrían ocurrido», que sin poder explicarlas cabalmente, las entendemos todos de una u otra manera.



Figura 1.2: Una silla miniatura.

Fuente: Citlalli EB (<https://www.flickr.com/photos/citlalli-eb/>)
Creative Commons - Attribution-NonCommercial-NoDerivs 2.0 Generic

Quizá por la dificultad que representa para nosotros pensar en varias cosas simultáneamente, tengamos problemas para concebir de manera precisa el resultado que emerge de la interacción de muchos elementos entre sí actuando al mismo tiempo. Lo que solemos hacer es encadenar nuestros pensamientos y reflexiones uno detrás de otro, y las soluciones que ofrecemos a los problemas que se posan delante nuestro conservan esa estructura. Tiene sentido que la naturaleza haya desarrollado seres que piensen de esa manera, después de todo, los individuos difícilmente podemos hacer muchas cosas a la vez. Esto, quizá, nos hace miopes a la realidad cuando las interacciones y su impacto en la dinámica de un sistema se vuelven relevantes.

Sin embargo, esa complejidad e interacción resulta omnipresente. Se encuentra en las neuronas del cerebro, aquella estructura que es el modelo generador natural por excelencia de la inteligencia (cualidad que tratamos de imitar). Esa complejidad la hemos vislumbrado vagamente con redes neuronales que nosotros mismos hemos construido. Se encuentra en la danza de los números que constituyen la evaluación de un conjunto de funciones de activación actuando al unísono.

Una queja sobre las redes neuronales artificiales en particular, es su cualidad de «cajas negras». De alguna manera queremos entender cómo logra su cometido en conceptos simples para nosotros. ¿Y si tal simpleza no existe? ¿Si esa complejidad es realmente irreducible?

Pretender conocer, aprehender, o entender los criterios que un sistema considera al momento de realizar una clasificación, puede ser por sí mismo un empeño que limite su alcance, dinamismo y versatilidad, al momento de construir las y diseñarlas. Si las interrelaciones, los conceptos y las operaciones involucradas son tan complejas, ¿es posible que aspirar a tal control o entendimiento tradicional resulte ser excesivamente ingenuo?

1.3.1. En la raíz de la complejidad

Esta visión reduccionista que empleamos al momento de abordar la naturaleza, es herencia de nuestros éxitos pasados. Los cimientos de nuestro conocimiento sobre el mundo se anclan en tal visión. La física, química, y buena parte de las llamadas «ciencias duras», tienen el implícito empeño de reducirlo todo a su mínima expresión. A encontrar o idear las reglas simples, elementales, que subyacen debajo de todo. Realmente funciona. Se han encontrado estructuras ciertamente elegantes y majestuosas, pero también es verdad que no lo hace siempre.

Hay sistemas que, sin dejar de estar dominados por reglas simples, nos revelan comportamientos complejos, irreducibles, que no parecen ser la suma de sus partes. Y esto compete, incluso, a los sistemas más simples entre los simples. Nos da cuenta de ello Stephen Wolfram con los autómatas celulares elementales (Wolfram, 2002), hecho que lo ha llevado a reflexionar y sugerirnos abordar el *universo computacional* como algo que puede ser «explorado» y «minado», cual aventureros en busca de tierras lejanas.

Aunque es muy difícil definirlo en forma sencilla, entiéndase por *sistema complejo* aquel compuesto por múltiples entidades que «interactúan de manera tal que el comportamiento de cada elemento depende [...] de otros elementos. Es por estas interacciones que uno no puede predecir el comportamiento [...] al estudiarlos por separado» (Gershenson, 2014, p. 28). Todo ello en un sentido tal, que su dinámica no pueda ser reducida a otras reglas más simples. Las interacciones producen información nueva que no se encuentra presente en los

componentes individuales, pero resulta capital para el comportamiento global del sistema.

Los ejemplos típicos a este respecto son los movimientos colectivos de las bandadas o los cardúmenes, que sin poseer un control central, son capaces de expresar una dinámica colectiva compleja, como si fueran «una sola entidad»².

La alergia por «lo complejo» y enrevesado, incapaz de ser expresado en términos simples en su totalidad, puede traslucir un amor mal entendido a la elegancia inherente a las matemáticas y estructuras formales, y podría ser un impedimento para llevar más allá los alcances de sistemas diseñados para enfrentarse, precisamente, a lo complejo. Como es el caso de los sistemas inteligentes y la inteligencia artificial en general.

1.3.2. Donde la complejidad y el aprendizaje se unen

Una aproximación a las teorías de clasificación y aprendizaje de máquina relacionada con el tema de la complejidad, se encuentra en la llamada *computación de reservorio* (*reservoir computing*). Ésta se basa en la idea de perturbar un medio dinámico, donde su respuesta a dicho estímulo se considera un proceso de cómputo aplicado a la entrada, que se mapea al espacio de estados del medio. Una unidad lectora es entrenada para interpretar dichos estados a fin de realizar tareas de clasificación sobre series temporales.

Si bien los medios dinámicos suelen ser implementados como redes neuronales recurrentes de diversos tipos (Schrauwen, Verstraeten & Campenhout, 2007), el principio teórico que sustenta este marco de trabajo sólo exige la existencia de un medio dinámico, cualquiera que este sea, con tal de que la transformación de la entrada no sea lineal, y tenga una riqueza adecuada para cumplir la llamada *memoria evanescente* (Boyd & Chua, 1985) y la *propiedad de separación* (Maass y col., 2002; Natschläger, Maass & Markram, 2002).

1.4. Motivación

Considerando las reflexiones expuestas con anterioridad, este trabajo se encuentra motivado por el deseo de explorar aproximaciones al aprendizaje de

²Como resulta obvio, nosotros mismos nos consideramos «una sola entidad» sólo en la medida que permanecemos indiferentes a la existencia de las células de nuestro cuerpo, que interactúan incesantemente, como bandadas gigantescas.

máquina que tengan relación estrecha con el análisis de sistemas complejos, por considerar a estos un camino prometedor a largo plazo para el desarrollo y creación de sistemas de clasificación y razonamiento artificial.

De igual manera, es una oportunidad para investigar sobre sistemas poco explorados como medios dinámicos para la computación de reservorio. En particular, redes booleanas enteramente discretas, que dadas sus características tienen el potencial de ser implementadas en *hardware* con más facilidad que sistemas continuos o pseudo-continuos.

Esta última afirmación es sencilla de comprender si consideramos que las redes no sólo son discretas (lo que implica la ausencia total de aritmética de punto flotante), sino que además la comunicación entre elementos tiene un ancho de banda de uno o dos bits para los modelos presentados aquí. Esto favorece una alta densidad de nodos y canales de comunicación más simples y fiables entre ellos.

Un antecedente para sistemas neuronales complejos implementados en *hardware* lo podemos encontrar en el proyecto *NeuroGrid* (Benjamin y col., 2014), y el proyecto *SyNAPSE* de IBM (P. Merolla y col., 2011; Arthur y col., 2012). Ambos centrados en la construcción de neuronas biológicamente realistas con sistemas digitales o híbridos (el modelo neuronal utilizado en estos chips, llamado de *integración y disparo*, ha servido de inspiración para la red propuesta en este trabajo. Detalles sobre este modelo se presentan en el capítulo 3).

Componentes electrónicos basados en las redes discretas propuestas aquí podrían, en principio, ser una alternativa más económica y simple a las arquitecturas antes mencionadas, al menos en tareas de computación de reservorio.

1.5. Organización del documento

Este trabajo se desarrolla de la manera descrita a continuación:

El capítulo 2 da una muy breve reseña del origen, historia, motivación y fundamentos de la computación de reservorio. El capítulo 3 introduce al lector a las redes booleanas, a fin de prepararlo para la comprensión del modelo propuesto en este trabajo, también incluido aquí. El capítulo 4 establece las medidas y criterios que serán utilizados para la caracterización de los modelos propuestos. El capítulo 5 detalla la preparación del modelo experimental a fin de

ser evaluado. El capítulo 6 muestra los resultados y la interpretación asociada a ellos. El capítulo 7 presenta las conclusiones extraídas del trabajo.

Capítulo 2

Computación de reservorio

Memory is an ocean and he bobs on its surface.

«La memoria es un océano y él se mece en sus olas».

Yann Martel

2.1. Antecedentes

La imagen que inmediatamente viene a la mente de un científico de la computación cuando concibe eso, «la computación», probablemente sea la de una Máquina de Turing. Y no faltan razones. Dicho modelo es su abstracción por excelencia, y es fácil percibirla como una entidad de la que no podemos prescindir, que está ineludiblemente unida a tal concepto.

Aunque tal máquina abarca y define todo aquello que es computable, tal cual se muestra en (Turing, 1936), no se debe perder de vista que es una de las muchas formas en que puede expresarse el objeto al que representa. Ahí tenemos el cálculo lambda (Church, 1932) y la tesis de Church-Turing (Kleene, 1943), que sugiere la equivalencia entre ambas teorías.

Aunque, como ya se sabe, la predilección por elegir el modelo de Turing como la imagen por excelencia de la computación está relacionada con su sencillez, y la fácil aprehensión que tiene el sentido común de él —una cinta, símbolos escritos en la misma, un cabezal de lectura/escritura, la lista de instrucciones que dictan las acciones a realizar con todas esas cosas—, en contraste al alto nivel de abstracción involucrado en el concepto de función recursiva al que nos conducen las reflexiones de Church.

La simplicidad implícita en la concepción de la máquina no es casual. Turing deseaba encontrar un *mecanismo* que, en principio, pudiera ser implementado en el mundo físico, y que al mismo tiempo sirviera como solución al problema

planteado por Hilbert: el *Entscheidungsproblem* (Hilbert & Ackermann, 1928). Tal cual se conoce, el modelo de Turing es uno de los prototipos de implementación más simples de una máquina universal. Sabemos que podemos complicarla cuanto queramos: más cintas, más cabezales, etc., pero su poder expresivo será siempre el mismo.

Las computadoras actuales, particulares encarnaciones de tales conceptos, siendo diferentes poseen algunas similitudes a la Máquina de Turing, tanto en estructura como en funcionamiento. Tienen su particular cinta, representada por la memoria, sus cabezales, instrucciones y demás. Su operación tampoco dista mucho de la máquina original; usualmente realiza sus actividades secuencialmente, leyendo su memoria «a saltos» y ejecutando la acción que dicta su programa paso a paso.

Si bien las computadoras tradicionales son máquinas universales en el sentido de Turing (obviando el ineludible problema de no poseer cintas indefinidamente largas), revelan sus limitaciones prácticas cuando se enfrentan a problemas que requieren el manejo de una gran cantidad de datos, o bien, a cálculos que demandan una gran cantidad de pasos de ejecución.

Una solución ha sido intentar, tanto como sea posible, paralelizar los cálculos. Es decir, que la máquina realice simultáneamente varios de ellos, para así computar más en menos tiempo. Sin embargo, este enfoque tiene sus desventajas e inconvenientes. No es posible realizar dos operaciones simultáneamente si una de ellas necesita el resultado todavía no calculado de la otra. Esto se puede generalizar para un número arbitrario de operaciones. Las circunstancias que hacen posible la paralelización no son fáciles de encontrar, y en ocasiones resultan imposibles.

2.1.1. La inspiración biológica

Un cerebro, ¿es una computadora? Tal pregunta, que formó parte del ideario de los pioneros de la inteligencia artificial, inspiró en los años cuarentas y cincuentas a una generación para concebir un nuevo paradigma en el procesamiento de datos: las redes neuronales artificiales. En particular, Frank Rosenblatt ideó el *perceptrón* (Rosenblatt, 1958). Con un arreglo de los mismos fue capaz de reconocer todas las letras del alfabeto.

La idea básica es la siguiente: un conjunto de «unidades de umbral» que llamamos de forma general «neuronas», se conectan entre sí formando una red,

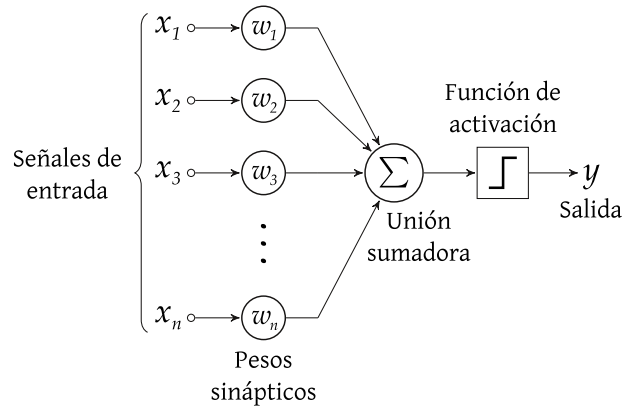


Figura 2.1: Esquema de un perceptrón, el cual recibe n entradas que son ponderadas con un peso w asignado a cada una de ellas. El valor obtenido es evaluado por una función de activación que definirá la salida de ese perceptrón para esa entrada en particular.

de tal suerte que su comportamiento se encuentre en función de los estímulos que reciban de sus demás compañeras. Todo ello tratando de simular, de una forma muy simplificada, el comportamiento neuronal verdadero. En este paradigma, las unidades colaboran entre sí para producir una salida de acuerdo al estímulo que ha sido inyectado a la red.

El desarrollo de este tipo de estructuras quedó estancado durante varios años, debido a la escasez de modelos de aprendizaje y aprovechamiento de las mismas, y en particular, al desánimo que provocó en la comunidad el libro de Minsky y Papert llamado *Perceptrons: An Introduction to Computational Geometry* (Minsky & Papert, 1969), donde se hacían predicciones pesimistas sobre la utilidad de tales entidades, sacando a la luz limitaciones no descubiertas hasta entonces. En particular, su incapacidad de «aprender» y clasificar correctamente funciones no separables de forma lineal como la XOR.

2.1.2. Una pequeña revolución: la retropropagación del error

La indagación de arquitecturas diferentes, principalmente de aquellas donde las neuronas se organizaban en «capas», y las demostraciones sucesivas que

dejaban al descubierto su capacidad para describir funciones no-lineales, y en general, su calidad de aproximadores universales (Hornik, Stinchcombe & White, 1989), hicieron superar los escollos anteriores. Pero de poco servían esas cualidades, si resultaba imposible ajustar adecuadamente los parámetros de los modelos.

El mayor impulso vino de la mano del algoritmo de *retropropagación del error* (*backpropagation*), que en la versión propuesta en (Rumelhart, Hinton & Williams, 1985), dio el empujón definitivo a la anteriormente abandonada área de las redes neuronales artificiales. Ahora resultaba posible «entrenar», a las redes con la eficacia suficiente para volverlas capaces de aproximar, clasificar y separar conjuntos de datos no vistos con anterioridad.

Hasta ese momento, la arquitectura que predominaba era la de *alimentación hacia adelante* (*feedforward*), que consiste en múltiples capas, un mínimo de tres, conectadas de tal manera que las neuronas de una se conectan única y exclusivamente con las neuronas de la capa siguiente. Así, hasta llegar a la última capa: la de salida. El análisis hasta ese momento realizado sobre las redes neuronales se centraba en su mayor parte en este tipo de disposición.

2.1.3. Recurrencia

Los «circuitos» biológicos son, sin embargo, recurrentes. Las conexiones neuronales forman ciclos de retroalimentación entre ellas, no permanecen conectadas a la manera de las redes descritas con anterioridad. Era necesario, por tanto, desarrollar una teoría que los contemplara. Walter Freeman, un neurocientífico, fue de los primeros que insistió en analizar los sistemas neurales desde el punto de vista de los sistemas dinámicos. Un libro que condensa esos primeros trabajos es (Freeman, 1975). Poco después de esto se inició la investigación sobre las propiedades computacionales de los circuitos neurales recurrentes.

Los primeros trabajos se concentraron en los atractores de los sistemas, a fin de restringir su dinámica y lograr propiedades bien definidas. Indagaciones al respecto se pueden encontrar en (Amari, 1972; Little, 1974). Esta corriente de trabajo culminó con los estudios de Hopfield sobre las redes de memoria estable que llevan su nombre (Hopfield, 1982). En estas redes, la «memoria» se encuentra asociada a estados estables, atractores puntuales de la red.

Término	Descripción
Entrada por lotes	Todos los datos se encuentran disponibles al inicio de la computación.
Computación en línea	Recibe un flujo permanente, quizá sin fin, de datos de entrada. Resulta necesario integrar la información recibida previamente del flujo con la información actual.
Computación en tiempo real	El resultado de la computación debe ser entregado en un plazo de tiempo determinado.
Algoritmo de cualquier momento	La mejor salida disponible se puede acceder en cualquier momento, sin seguir una agenda específica.
Computación fuera de línea	Si no existe un tiempo límite para la entrega de la salida.

Tabla 2.1: Clasificación de sistemas de computación, siguiendo las definiciones expuestas en (Maass & Markram, 2006).

Posteriormente apareció la idea de utilizar la rica dinámica de los sistemas neuronales, tal cual se observan en los circuitos corticales, en vez de intentar restringirlos y controlarlos (Buonomano & Merzenich, 1995).

Dentro de sus señalamientos, cabe destacar que aprovechar el poder de estos sistemas implica observar los *cómputos en línea*³ del mismo.

Todos estos trabajos llevaron al desarrollo de las *Máquinas de Estado Líquido* (*Liquid State Machines*) por Maass y col. (2002), así como las *Máquinas de Estado-Eco* (*Echo-State Machines*) por parte de Jäger (2001). Ambos trabajos desarrollados independientemente. Las primeras bajo un enfoque orientado a las ciencias biológicas, y el segundo bajo una perspectiva ingenieril.

La *computación de reservorio* (*reservoir computing*), que es la forma en que se ha denominado al marco de trabajo utilizado por estas máquinas, está enfocado en el procesamiento de información por medio de un sistema dinámico, llamado *reservorio* o *piscina*, que gracias a su comportamiento es capaz de mapear las entradas que recibe a un espacio de estados que facilita su clasificación, haciendo las veces de una función *kernel* temporal (Lukoševičius & Jäger, 2009).

³Esta y otras clasificaciones para el cómputo y su alimentación se presenta en la tabla 2.1.

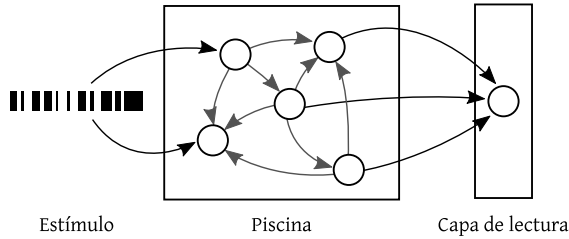


Figura 2.2: Un esquema del cómputo de reservorio. La entrada causa una perturbación en la piscina, el cual altera su dinámica, creando una representación de dicho estímulo. La capa de salida se encarga de leer el estado instantáneo del medio dinámico, el cual ha sido construido de forma aleatoria bajo ciertas condiciones. El entrenamiento sólo es aplicado a las conexiones existentes entre el depósito y la capa de lectura, que es la encargada de clasificar e interpretar aquello que recibe.

La idea subyacente es perturbar el medio, que sostendrá dicha alteración por un lapso de tiempo, de tal forma que el estado del sistema en un instante determinado contenga información relativa a los eventos sucedidos en el pasado, información que debemos ser capaces de extraer a fin de tratar con ella. Este último paso se lleva a cabo a través de una unidad de lectura que debe aprender a mapear los estados del sistema a las salidas deseadas. En teoría, esta unidad puede llegar a ser lineal. Si bien un estado particular del sistema debe poseer información útil sobre los eventos acaecidos con anterioridad, no es deseable que lo recuerde todo. A la capacidad para «olvidar» del sistema se le conoce como *memoria evanescente* (*fading memory*).

Se debe hacer hincapié en que el medio dinámico que constituye la piscina no es creado ni diseñado expresamente para una tarea específica, por el contrario, es creado al azar. La aproximación típica es el empleo de redes neuronales recurrentes como reservorio, tal cual hacen los trabajos mencionados anteriormente. Sin embargo, el principio que gobierna este enfoque no está restringido a las redes neuronales. Es posible emplear, teóricamente, cualquier medio con suficiente riqueza en su dinámica, tal que lo haga capaz de realizar un mapeo adecuado de las entradas a un estado del sistema. Un ejemplo de ello es la aplicación de estas ideas utilizando agua líquida como medio excitable (Fernando & Sojakka, 2003).

Una de las ventajas de este marco de trabajo radica en volver innecesario ajustar los pesos de las conexiones que forman parte de la red neuronal recurrente que hace las veces de medio y, en general, cualquier modificación guiada al mismo. Tan sólo debe aplicarse entrenamiento a un subconjunto muy pequeño de conexiones, aquellas asociadas a los nodos de lectura del sistema, que son ajustadas con una técnica de aprendizaje de máquina tradicional. La piscina sólo funge como integradora de la información temporal proveniente del exterior. Ésta particularidad es muy importante, dado que evita lidiar con la dificultad que representa entrenar los pesos de las conexiones de una red que hiciera las veces de reservorio, si fuera el caso.

No está de más mencionar algo muy importante, y es que resulta posible conectar una cantidad arbitraria de unidades de lectura o salida a un mismo medio dinámico, cada una con su propio entrenamiento, a fin de extraer información de muy diversa índole simultáneamente. Se destaca que dichas unidades son en extremo simples.

En el asunto que nos ocupa, se desea investigar la viabilidad de un medio discreto como piscina para un sistema de computación de reservorio, por tal motivo, el análisis se concentrará en las particularidades propias de estos sistemas.

2.2. Sistemas dinámicos

Estudiar analíticamente los sistemas mencionados no es tarea trivial. Afortunadamente existe mucha literatura que permite abordar su estudio en el marco de la teoría de los sistemas dinámicos. En particular, sobre aquellos localizados entre los comportamientos ordenados y caóticos, que resultan pertinentes para el tema tratado en este trabajo. Se introduce aquí en conceptos clave de los mismos.

Los sistemas dinámicos hacen referencia tanto a entidades discretas como a continuas, siempre que las mismas evolucionen en el tiempo. Las descripciones dadas a continuación se refieren a sistemas continuos, pero tienen una equivalencia con sistemas discretos que se verá más adelante.

2.2.1. Entidades continuas

El *espacio fásico* de un sistema N -dimensional es el espacio con coordenadas x_1, x_2, \dots, x_N . El estado de un sistema en el instante t se representa por el vector de estado $x(t) = (x_1(t), x_2(t), \dots, x_N(t))$. Dado un punto en el espacio fásico, el sistema evolucionará de acuerdo a su dinámica, describiendo una trayectoria en él. El espacio fásico también es conocido como *espacio de estados*.

Un estado x_s se llama *estado estacionario*, si residiendo en él un sistema no describe alguna trayectoria que abandone dicho punto. Los estados estacionarios pueden ser o no estables a perturbaciones externas arbitrariamente pequeñas. Entiéndase por «estable» si la trayectoria del sistema a consecuencia de la perturbación converge al estado estacionario nuevamente. En caso contrario, se dice que es inestable.

Un *atractor* A es un conjunto de puntos en el espacio de estados, tal que un volumen a su alrededor converge asintóticamente hacia él. Tal conjunto es invariante en el tiempo. Es decir, para cualquier estado $x(t) \in A$, se cumple que $x(t + \Delta) \in A$, para una Δ arbitrariamente grande. El conjunto de estados que convergen al atractor se conoce como *cuenca de atracción*.

Los atractores pueden ser de diferentes tipos. Si consiste en un único punto se conoce como *atractor puntual*. Si está compuesto por una trayectoria cerrada que se recorre una y otra vez, se le conoce como *ciclo límite*. Existen más tipos de atractores que forman estructuras geométricas cada vez más complejas. Usualmente esto depende del número de dimensiones del espacio de estados, donde una mayor diversidad topológica es posible.

Un *atractor extraño* puede considerarse como un conjunto de puntos en el espacio de estados tal que, al igual que los atractores tradicionales, posee un volumen a su alrededor tendiendo a él. Con la diferencia que, dadas dos condiciones iniciales arbitrariamente próximas, tendrán una evolución que hará divergir sus trayectorias, al grado de no existir una correspondencia con la distancia inicial de los estados iniciales y finales. Esto implica también que dos estados finales muy próximos pueden tener como origen dos iniciales lejanos entre sí.

Esta extrema sensibilidad a las condiciones iniciales caracteriza el comportamiento llamado *caótico*. La predicción de la evolución temporal del sistema se vuelve en la práctica imposible, aún si es determinista, dado que es requerido un conocimiento con precisión infinita del sistema entero.

2.2.2. Particularidades de los sistemas discretos

La dinámica de los sistemas discretos difiere en algunos aspectos con la de sistemas continuos, dadas sus características inherentes. Esencialmente son dos:

- Los estados son discretos, por esta razón las trayectorias en el espacio de estados se pueden unir, y no simplemente se aproximan asintóticamente entre sí.
- El número de estados es finito, esto significa que la evolución del sistema inexorablemente nos llevará a un estado visitado previamente.

Los conjuntos de estados que se repiten y visitan una y otra vez en la evolución temporal del sistema, son análogos a los atractores o ciclos límite de los sistemas dinámicos continuos. Si tales conjuntos están constituidos por más de un estado se conocen como *estados cíclicos* o *atractores a secas*. Si el atractor está constituido por un sólo estado se conoce como *atractor puntual*.

Se conoce como *longitud del estado cíclico* al número de estados que lo constituyen. Un estado que no forme parte de un atractor se llama *estado transiente* o *transitorio*. La cantidad de estados transitorios que son visitados antes de alcanzar un atractor de cualquier tipo, se conoce como *tiempo transiente*.

El comportamiento caótico en sistemas discretos está asociado a longitudes muy grandes para sus estados cíclicos y cuencas de atracción pequeñas. Esto convierte a los sistemas muy sensibles a condiciones iniciales, ya que fácilmente pasan de un estado cíclico a otro, o bien, se alejan con poco esfuerzo del estado estable en el cual se mantenían.

2.3. Computación en el borde del caos

¿Qué significa computación en el contexto de los sistemas dinámicos? Como se ha mencionado ya, diversos tipos de computación existen, y varios de ellos pueden ser implementados en los sistemas dinámicos. Las aproximaciones iniciales estaban dirigidas a la computación con entradas por lotes, donde la información necesaria para llevar a cabo la computación se encontraba codificada en el estado inicial del sistema. Las reglas que determinaban su evolución eran «el programa», y el estado final era la salida o resultado (Mitchell, Hraber & Crutchfield, 1993).

Para el caso de sistemas discretos, y en particular los autómatas celulares⁴, se mostró que tenían la capacidad de ser máquinas universales en el sentido de Turing (Smith III, 1971), funcionando de forma análoga: para una configuración de entrada particular se obtiene, después de un número indeterminado de pasos, el resultado.

Stephen Wolfram conjeturó que los autómatas con altas capacidades de cómputo se encontraban bajo un régimen dinámico particular (Wolfram, 1984). Él estableció cuatro clases posibles para los autómatas estudiados por él:

Clase 1. Sistemas que evolucionan a un estado homogéneo. Es decir, uno donde el estado de sus elementos es el mismo.

Clase 2. Sistemas que evolucionan a un conjunto de estados estables simples o estados cíclicos de pequeña longitud.

Clase 3. Sistemas que evolucionan a patrones caóticos.

Clase 4. Autómatas con grandes tiempos transientes, que «evolucionan a estructuras complejas localizadas».

Según sus conjeturas, los autómatas de clase 4 son capaces de computación universal (asumiendo autómatas arbitrariamente grandes).

Los sistemas de clase 4 tienen asociadas cuencas de atracción de gran tamaño, dados los largos tiempos transientes y gran estabilidad para sus atractores.

Estudios posteriores hechos por Langton (1990), mostraron que, efectivamente, una alta capacidad de cómputo estaba asociada a sistemas al borde del caos, que se correspondían cualitativamente con la clase 4 descrita por Wolfram. En particular, Langton utilizó un parámetro de orden que él llamó λ , para establecer la construcción de sus autómatas celulares.

En palabras simples, se define λ como el porcentaje de transiciones, para cada una de las células o nodos del mismo, que no llevan a un estado particular del sistema, previamente predefinido y nombrado como «inactivo».

Norman Packard, aplicando algoritmos genéticos, concluyó cosas similares (Packard, 1988). Su objetivo era encontrar autómatas tales que, si la mayoría de sus nodos se encontraban en estado 1, entonces evolucionaran a un estado homogéneo 1 (donde todos los nodos toman dicho valor), en caso contrario, que evolucionaran a un estado homogéneo 0. Los valores de la λ de Langton

⁴Se puede consultar (Niesche, 2006) para una introducción a los mismos.

que se correspondían con las mejores reglas, giraban alrededor de los valores 0.2 y 0.8. Valores asociados a los autómatas de clase 4.

Sin embargo su trabajo tuvo críticas. Un grupo de investigadores examinó nuevamente sus resultados, concluyendo que el valor de λ dependía de la tarea computacional a realizar (Mitchell y col., 1993). Ellos hablaron de tres diferentes tipos de computación:

1. El estado inicial es la entrada, las reglas del sistema son el programa, y la salida es alguna clase de estado final «útil» y «significativo».
2. Dado un arreglo especial en el estado del autómata, este es capaz de simular una máquina universal de Turing, siendo posible emular el comportamiento de una computadora programable, simular puertas lógicas, memoria, etc.
3. La computación es «intrínseca», y no se interpreta como una transformación «útil» a la manera en que lo es la computación del tipo 1 o 2.

Respecto a este último punto, en (Mitchell y col., 1993, p. 5) se puede leer:

Here computation is not interpreted as the performance of a “useful” transformation of the input to produce the output. Rather, it is measured in terms of generic, structural computational elements such as memory, information production, information transfer, logical operations, and so on. It is important to emphasize that the measurement of such intrinsic computational elements does not rely on a semantics of utility as do the preceding computation types.⁵

Es destacable que esa «computación intrínseca», descrita como una transformación «no útil» de la información, es en realidad «útil», si es utilizada una unidad de lectura que mapee dichos estados a una salida deseada. Es precisamente tal mecanismo el que se encuentra detrás de la idea de la computación de reservorio, y el mecanismo aprovechado posteriormente por las Máquinas de Estado Líquido, las Máquinas de Estado-Eco, y el resto de las encarnaciones de este paradigma computacional.

⁵«Aquí computación no es interpretada como el desempeño de una transformación “útil” de la entrada al producir la salida. Más bien, es medida en términos de elementos computacionales y estructurales genéricos como la memoria, producción de información, transferencia de información, operaciones lógicas, y así. Es importante resaltar que la medida de esos elementos computacionales intrínsecos no se basa en una semántica de utilidad como los tipos de cálculo anteriores». Traducción personal.

De igual forma, no está de más comentar el paralelismo que existe con el tema mencionado en la introducción de este trabajo: el aparente significado, o ausencia del mismo, que tiene la transformación de la información en las «cajas negras». Mitchell y colaboradores «despreciaron» aquello que llaman «computación intrínseca» por su carencia de sentido inmediato, en términos que sean significativos para nosotros.

2.4. Necesidades en el cómputo de reservorio

En los sistemas biológicos, la computación es diferente a la llevada a cabo por nuestras máquinas electrónicas. En oposición al procesamiento que utiliza entrada por lotes, típico en la computación tradicional, los datos llegan al sistema como un flujo ininterrumpido, y las salidas tienen una naturaleza similar. Una señal que varía en el tiempo es mapeada a otra señal variante en el tiempo. Estos mapeados son también llamados *filtros*.

En particular, la idea de aprovechar toda la rica dinámica que existe en redes corticales, en vez de restringirla, dio origen a la computación de reservorio y todas sus implementaciones.

La computación aprovechada por este tipo de sistemas es la «intrínseca», y es un tipo de computación en línea, es decir, es capaz de integrar información en el pasado con la actual al momento de generar su salida, en oposición a la computación por lotes.

Para que tal tipo de computación sea plenamente aprovechada los sistemas deben cumplir las características explicadas a continuación.

2.4.1. Memoria evanescente

Una cualidad del régimen ordenado en los sistemas dinámicos útiles para el cómputo en línea, propio de la computación de reservorio, puede ser descrito en términos de lo que se ha dado en llamar *memoria evanescente* (Boyd & Chua, 1985). Para sistemas continuos, los filtros invariantes en el tiempo⁶ con me-

⁶En este contexto, con *invariante en el tiempo* se pretende decir que, para un sistema F , el desplazamiento temporal de una función de entrada $u \in U$ causará el mismo desplazamiento en la función de salida $y = Fu$. Es decir, para un desplazamiento Δ tal que $u^\Delta(t) = u(t + \Delta)$, se debe cumplir que $(Fu^\Delta)(t) = (Fu)(t + \Delta)$. Se asume que U es cerrado bajo desplazamientos temporales.

moria evanescente son, precisamente, aquellos que pueden ser representados por series de Volterra, y de forma general, tal concepto es expresado siempre con auxilio de dicha construcción matemática. Sin embargo, tal definición no resulta adecuada para sistemas discretos como los propuestos aquí. En el capítulo 3, se ofrece una aproximación intuitiva al significado del término para los sistemas empleados en este trabajo.

Se dice que un sistema posee memoria evanescente cuando se cumple lo siguiente:

Definición 2.1. Sea F un sistema dinámico que proyecta las entradas provenientes del universo U en el espacio de estados R . Es decir, $F : U \rightarrow R$. Si para cualquier $u \in U$ y cualquier $\varepsilon > 0$ existe una $\delta > 0$ y una $\mathcal{T} > 0$ tal que $\|(Fu)(0) - (Fv)(0)\| < \varepsilon$ para todo $v \in U$ con $\|u(t) - v(t)\| < \delta$ para todo $t \in [-\mathcal{T}, 0]$, se dice que F tiene *memoria evanescente*.

En otras palabras: la memoria evanescente es una propiedad de los sistemas que demanda para cualquier función $u \in U$, que la salida actual de Fu pueda ser aproximada por la salida actual de Fv , para cualquier otra función de entrada $v \in U$ que se aproxime o sea igual a u durante un tiempo lo suficientemente largo $[-\mathcal{T}, 0]$.

Otra forma de expresarlo es decir que, para entradas iguales, el sistema converge al mismo estado, sin importar las condiciones iniciales en las cuales se encontrara previamente.

Intuitivamente, el sistema posee esta propiedad si dos señales que son similares en el pasado reciente, pero no necesariamente cercanas en el pasado remoto, producen salidas actuales que se encuentren también cercanas entre sí. Es decir, las salidas convergen y se acercan una a la otra conforme pasa el tiempo.

Esta característica de algunos sistemas dinámicos es deseable para el cómputo en línea, ya que está relacionada fuertemente con la capacidad de generalización del sistema, y ésta hace alusión a la habilidad que tendrá para clasificar bajo la misma clase a entradas similares entre sí.

Este rasgo es característico de los sistemas dinámicos en régimen ordenado, ya que implica no sólo poca sensibilidad a las condiciones iniciales, sino convergencia en su comportamiento a un estado particular, que favorece la invarianza temporal del sistema.

2.4.2. Separación puntual

Otra propiedad importante para la computación en línea y la de reservorio en general, es la *separación puntual*, y ésta hace referencia a la capacidad del sistema para separar entradas diferentes. Es decir, que tiene la capacidad para establecer una distancia en el espacio de estados al mapeo de entradas que también son distantes en su propio espacio.

Esta propiedad es seductoramente similar a la tendencia que tienen los sistemas caóticos de crear trayectorias divergentes entre sí para estados iniciales diferentes, por poca que sea su diferencia.

Definición 2.2. Se dice que una red perturbable F posee la propiedad de *separación puntual* respecto a las funciones de entrada de U , si para dos funciones $u, v \in U$, cuando $u(s) \neq v(s)$ para algún $s \leq 0$, se cumple que $Fu(0) \neq Fv(0)$.

Este rasgo es deseable para la computación en línea, ya que se desea favorecer una gran distancia en el espacio de estados para entradas también distantes, a fin de alentar que sean asignadas a clases diferentes.

Como es evidente, tanto la *memoria evanescente* como la *separación puntual* son propiedades antagonistas en cierto sentido. Mientras una pretende crear trayectorias convergentes en el espacio de estados, la otra intenta lo opuesto. En principio, el régimen dinámico que permite la convivencia de ambas características es crítico, y se encuentra en el borde del caos del sistema. De ahí que se hable de «cómputo al borde del caos».

Capítulo 3

Redes booleanas

*Die ganzen Zahlen hat der liebe Gott gemacht, alles andere ist
Menschenwerk.*

«Dios creó los números naturales, el hombre fabricó el resto».

Leopold Kronecker

3.1. Redes Booleanas Aleatorias

Las *Redes Booleanas Aleatorias* (RBA) tienen su origen como modelos para las redes de regulación genética⁷, y fueron desarrolladas por Stuart Kauffman en 1969 (Kauffman, 1969, 1993). Él propuso su modelo para dar soporte a la hipótesis de que los seres vivos podrían estar constituidos por una plétora de componentes creados más o menos al azar, sin la necesidad de elementos expresamente fabricados y diseñados para seguir un propósito específico.

También conocidas como redes de Kauffman o modelos $N - K$, estas redes son generales, dado que sus componentes no asumen ninguna funcionalidad particular, y la topología de su conexión es completamente arbitraria. La investigación desarrollada a su alrededor suele establecer de forma aleatoria los parámetros destinados a su construcción, de ahí el nombre.

A pesar de su sencillez, tienen el potencial de ser utilizadas como modelo para multitud de procesos sin importar su naturaleza, siempre que los mismos se vean dominados por un comportamiento cuasi-booleano o de dos estados; quizá porque rijan umbrales de acción o algún otro fenómeno equiparable.

Las redes booleanas se han empleado en muy diversos campos que van, desde la biología a las matemáticas; pasando por la computación o la sociología.

⁷Son una colección de segmentos de ADN dentro de una célula que interactúan entre sí y con otras sustancias, de tal forma que regulan la velocidad a las que los genes de la red se transcriben en ARNm.

3.1.1. El modelo clásico

El elemento fundamental de una RBA es el nodo, que posee K conexiones entrantes que pueden provenir, cada una, de cualquier otro nodo de la red, incluido él mismo. En cualquier instante del tiempo, un nodo posee uno de dos estados posibles: «0» o «1». Un nodo actualiza su estado en el tiempo t de acuerdo a un mapeo booleano K a 1 de esas K entradas, de forma tal que el estado de un nodo en el tiempo $t + 1$ se determina enteramente por el estado de las K entradas en el tiempo t y una de las 2^{2^k} funciones booleanas posibles que puede emplear. La RB es una colección de N nodos del tipo mencionado anteriormente, con las correspondientes conexiones entre ellos y las funciones que dominan su comportamiento.

En este modelo clásico, la actualización del estado de todos los nodos se realiza a la vez. Este esquema de actualización se conoce como *síncrono*.

Entiéndase por *estado de la red* al conjunto de todos los estados de sus nodos constituyentes. Dado un estado de la red existe uno y sólo un estado sucesor del mismo, el cual la red alcanzará al seguir cada uno de sus nodos las reglas de transformación determinadas por las funciones booleanas asociadas a ellos.

Considerando que existen 2^N estados para una RBA, y siendo este número finito, llegamos a la conclusión de que tarde o temprano la sucesión de estados debe repetirse formando un ciclo. Este tipo de ciclos se repetirán siempre de la misma manera, dada la naturaleza determinista de la dinámica que gobierna a la red. Cada uno de estos ciclos se conoce como *atractor*, tal cual se define en la teoría de los sistemas dinámicos.

En consonancia con la terminología usada en dicha teoría, si el atractor consiste en un solo estado, éste se conoce como *atractor puntual* o *estado estable*. Por el contrario, si consiste en dos o más es conocido como *atractor de ciclo* o *estado cíclico*. Al conjunto de estados que siguiendo la dinámica del sistema conducen a un atractor, incluyéndolo a él mismo, se le conoce como *cuenca de atracción*. Se llamará *periodo del atractor* al número de transiciones que deben realizarse antes de repetir un estado visitado previamente.

El número de pasos necesario para alcanzar un atractor se conoce como *tiempo transiente*. Si un nodo es inalcanzable desde algún otro, éste toma el nombre de *jardín del Edén*.

Una RBA puede poseer más de una cuenca de atracción, con un atractor asociado a cada una de ellas. Si el espacio de estados de una RBA es concebido como

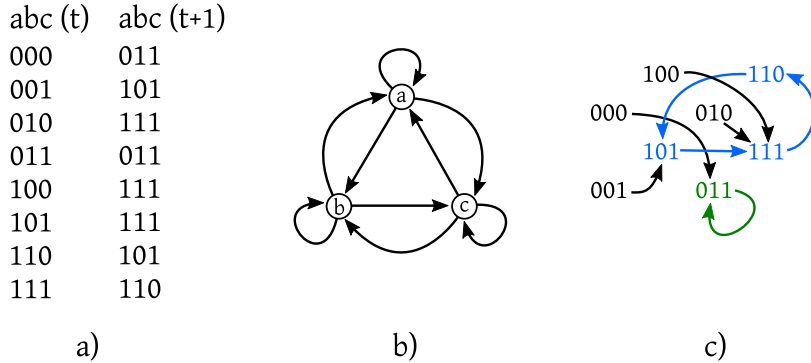


Figura 3.1: Una RBA de tres nodos. a) Tabla de transiciones entre estados. b) Diagrama de conexión entre los nodos, en este caso, todos los nodos afectan a todos. c) Diagrama del espacio de estados. Existen dos atractores: uno cíclico (azul), y uno puntual (verde). Los estados en negro son jardines del Edén. *Diagrama de (Gershenson, 2004)*

un grafo dirigido, donde cada estado corresponda a un vértice, y cada arista a la transición de un estado a otro, cada *cuenca de atracción* corresponderá a un componente de la gráfica desconectado del resto.

Orden, caos, y fase crítica

La dinámica de una RBA suele clasificarse en uno de dos diferentes regímenes: *ordenado* y *caótico*. Se distinguen por el porcentaje de nodos que cambian constantemente de estado conforme la red evoluciona a lo largo del tiempo.

Para una red en régimen ordenado, la mayoría de los nodos que lo constituyen permanecen estáticos, sin cambios de estado, y solo una minoría o ninguno alternando entre sus dos estados posibles.

Para una red en régimen caótico, la mayoría de los nodos que lo constituyen alternan sus estados, y solo una minoría o ninguno permanece estático.

La transición entre uno y otro es violenta, y constituye la llamada *fase crítica* que, en redes construidas al azar, suele presentarse cuando $2 \leq \langle K \rangle \leq 3$, siendo $\langle K \rangle$ el grado entrante promedio de los nodos que conforman la red (Derrida & Pomeau, 1986).

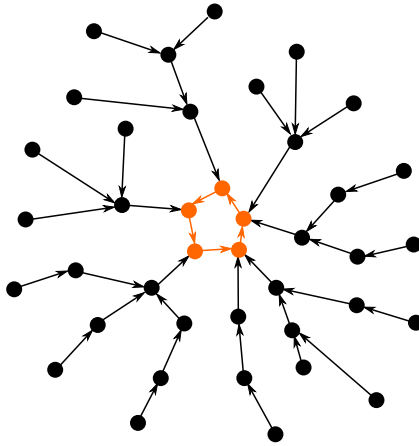


Figura 3.2: Una cuenca de atracción representada como un grafo dirigido. Cada vértice representa un estado, y las aristas las transiciones entre ellos. Los nodos y transiciones de color claro representan aquellos que forman parte del atractor. Los estados sin predecesores son llamados *jardines del Edén*.

Formas alternas de RBA

Existen sutilezas varias al momento de construir y definir el funcionamiento de las redes booleanas, entre las variantes más destacadas se pueden mencionar los diversos mecanismos de actualización del estado de cada nodo. Existen algunas donde el modo de actualización no es síncrono, lo que significa que no todos los nodos modifican su estado a la vez.

El origen de semejantes esquemas tiene su raíz en la idea misma de las RBA como modelos para las redes de regulación genética. La principal crítica proclamada hace mención al hecho de que los genes «no marchan al unísono». Por tal motivo se desarrollaron esquemas de actualización más versátiles. Entre ellos podemos encontrar los asíncronos deterministas y no deterministas. Como estos esquemas de actualización escapan del uso dado a las RBA en este trabajo, tan sólo son mencionados por completitud.

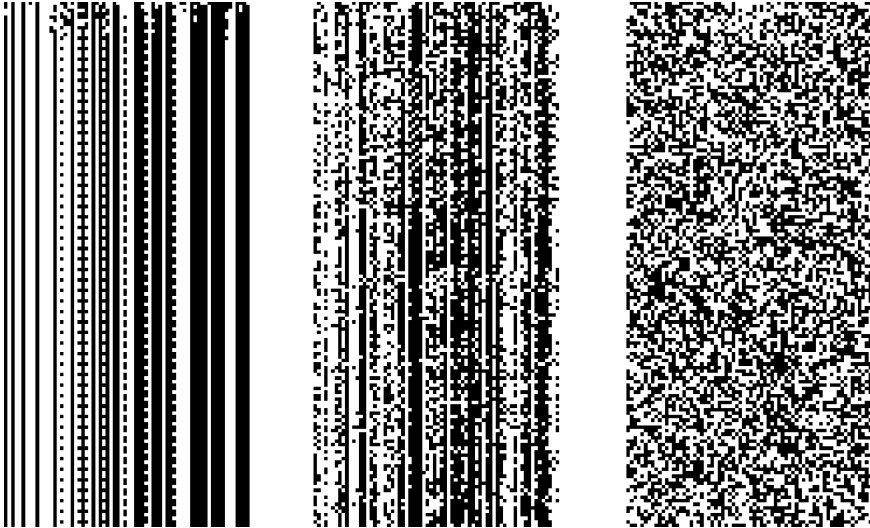


Figura 3.3: Evolución temporal de tres Redes Booleanas Aleatorias en diferentes regímenes. El tiempo fluye hacia abajo. *Izquierda*: ordenada con $\langle K \rangle = 1$. *Centro*: Crítica con $\langle K \rangle = 2.5$ (el valor se acerca a 2 cuando N tiende a ∞). *Derecha*: Caótica con $\langle K \rangle = 7$. Es fácil apreciar como en el último caso el sistema se mueve violentamente a través de su espacio de estados, resultando difícil percibir una regularidad en la escala de tiempo mostrada.

3.2. Perturbación en redes booleanas

En el tema que nos ocupa deseamos perturbar una red, a fin de que haga las veces de medio dinámico en un sistema de computación de reservorio pero, ¿qué sentido tiene hablar de perturbación en un medio que es, por definición, cerrado como las redes booleanas?

En la literatura, la perturbación de un medio discreto como las RBA suele describirse como el cambio aleatorio y arbitrario de los estados de algunos o todos los nodos que lo constituyen. Sin embargo, se ha considerado relevante redefinir aquí el significado del término, de forma que coadyuve al análisis y comprensión del fenómeno en cuestión. A continuación se explica detalladamente el significado que dicha palabra posee en el contexto que nos compete. Se muestran sus implicaciones y un conjunto de nuevas ideas fruto de esa comprensión. Ideas que son fundamentales y base de buena parte de esta tesis.

3.2.1. Definiendo la perturbación

En una RB tradicional el estado siguiente que puede adoptar la misma está predeterminado. A un estado cualquiera le corresponde uno y sólo un estado que le sucede. Esta premisa no se cumple necesariamente para una red sensible a la perturbación. El cómo esta perturbación es aplicada se explica detalladamente a continuación:

Sea M una RB cualquiera y E el subconjunto de todos los nodos en M que posean una conexión entrante proveniente de un nodo binario ω , donde $\omega \notin M$. En cada paso de tiempo el nodo ω es libre de tomar un valor arbitrario entre los dos posibles $\{0, 1\}$, con independencia del comportamiento y dinámica de M . Como el nodo ω forma parte de las tablas que representan la función lógica que define el comportamiento de todos los nodos en E , es capaz de afectar su dinámica con independencia del curso de acción que tengan cualquiera de los otros nodos de M a los que estén conectados dichos elementos de E . Este hecho por sí mismo nos muestra que, para un estado cualquiera de la RB, no necesariamente el siguiente se encuentra predeterminado.

Convengamos en llamar *red perturbable* a cualquiera que cumpla las características antes mencionadas.

	ω	a_K	...	a_2	a_1	$(t+1)$	
2^{K+1} entradas totales	{	0	0	...	0	0	→ 0
		0	0	...	0	1	→ 0
		0	0	...	1	0	→ 0
		⋮	⋮		⋮	⋮	⋮
		0	1	...	1	1	→ 0
	1	0	...	0	0	→ 1	
	1	0	...	0	1	→ 1	
	1	0	...	1	0	→ 1	
	⋮	⋮		⋮	⋮	⋮	
	1	1	...	1	1	→ 1	

$\left. \vphantom{\begin{matrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{matrix}} \right\} 2^K \text{ entradas para } p=0$

Figura 3.4: Tabla de transiciones para un nodo cualquiera con K conexiones típicas y una a un nodo ω de entrada. La tabla completa tiene 2^{K+1} entradas en total. El valor que tome ω en un momento determinado «bloquea» las reglas asociadas al valor opuesto. La mitad de la tabla cada vez. Esto equivale a alternar entre dos tablas con 2^K entradas, lo que significa alternar entre dos RB clásicas diferentes.

Cambio de RB en tiempo de ejecución

A continuación se desea mostrar que el comportamiento detallado anteriormente es equivalente a sustituir una RB clásica (cerrada) por otra con la misma topología y espacio de estados, pero diferentes funciones booleanas, según el valor que tome el nodo ω en cada paso de tiempo.

Sea $e_i \in E$ un nodo que se encuentra conectado a K_i nodos cualesquiera de la red M y a un nodo independiente ω . Como es fácil descubrir, el número de entradas en la tabla que define su comportamiento es de 2^{K_i+1} . Pero es de notar que, sin importar el estado de los otros K_i nodos de la red, tan sólo es posible hacer efectiva la mitad de dichas entradas según el valor que tome ω . Es decir, un cambio en el estado de ω alterna entre dos diferentes tablas con 2^{K_i} entradas cada una, para todo nodo e_i unido a él.

Esto equivale a alternar entre dos RB con la misma topología y espacio de estados, pero una dinámica dominada por reglas diferentes. Las reglas que cambian, por supuesto, son las asociadas a los nodos e_i . Esto puede generalizarse para los casos donde existan más nodos ω . En dichas situaciones el sistema alternará entre $2^{|\Omega|}$ redes diferentes, donde Ω es un conjunto de todos los nodos ω conectados al sistema, independientes entre sí.

3.2.2. Normas

Si el estado de los nodos ω que sirven de entrada alternan al sistema entre $2^{|\Omega|}$ redes diferentes, llamemos *norma* a cada una de estas redes, y nómbrense de acuerdo a la combinación de estados que presenten los nodos ω que las hacen efectivas. Así, por ejemplo, de existir un sólo nodo ω , existirán las normas: $\{0, 1\}$. De existir dos nodos ω las normas existentes serían: $\{00, 01, 10, 11\}$.

3.2.3. Estímulos

Elijase arbitrariamente una norma, es decir, una combinación particular de estados para todos los nodos ω , como aquel asociado a la ausencia de estímulos. Por convención, aquí se seleccionan como tales aquellos donde el estado de todos los nodos ω es igual a 0. A esta norma particular la llamaremos *de reposo*, al resto *de estímulo*.

Con esto en mente, podemos hacer la siguiente declaración:

Definición 3.1. Se dice que una red perturbable M ha recibido un *estímulo* o *pulso* en el tiempo t , si $\omega(t) = 1$, para cualquier $\omega \in \Omega$, siendo Ω el conjunto de todos los nodos independientes al sistema conectados al mismo.

3.3. Ideando un comportamiento deseable

No debemos olvidar el objetivo que persiguen las RB en este trabajo, y dicho propósito es: integrar información temporal. Mapear una secuencia temporal de estímulos a otra secuencia conformada por los estados de la red. Todo de forma que la transformación no lineal aplicada dé como resultado información sobre las entradas que permita realizar una clasificación de las mismas, bajo un criterio arbitrariamente elegido.

En la clasificación del mundo real no importa mucho el momento en que inicia un estímulo, tanto como el estímulo en sí. Por ejemplo, a un sistema reconocedor de voz no debe preocuparle mucho el instante en que una persona comience a decir la palabra «casa», tanto como reconocer que se ha dicho tal vocablo. Un sujeto puede comenzar a pronunciar su discurso al inicio del flujo de datos, o bien, pasado cierto tiempo. Como sea, el reconocedor debe ser capaz de clasificar adecuadamente la información entrante.

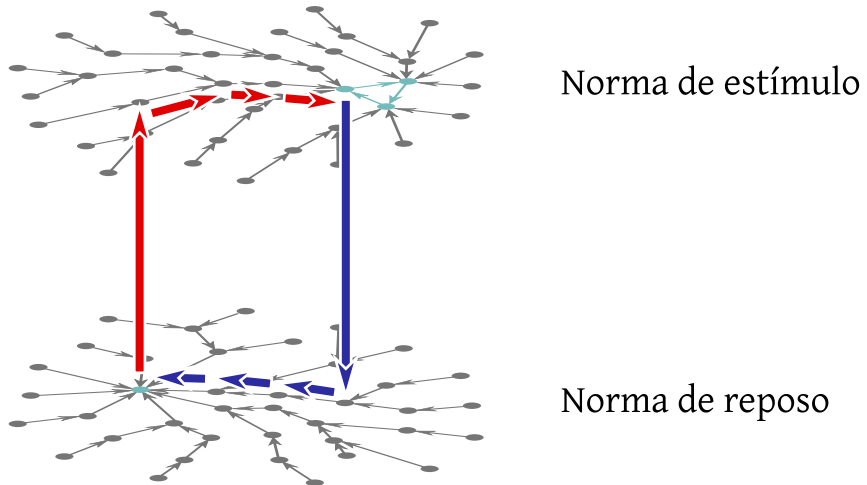


Figura 3.5: Un estímulo equivale al cambio de una RB por otra con el mismo espacio de estados, pero diferente dinámica, en tiempo de ejecución. En el ejemplo, se muestra un hipotético flujo en el espacio de estados del sistema. La piscina sale del equilibrio debido a una perturbación (en rojo), describiendo una trayectoria en el espacio de estados guiado por las reglas de la norma de estímulo. Cuando el estímulo cesa, la norma de reposo toma de nuevo el control y el sistema viaja de regreso al atractor puntual de su RB asociada (azul). El estado en que se encuentre el sistema en un momento dado contiene información sobre los estímulos acaecidos en el pasado.

Para el caso que nos ocupa este tópico es fundamental porque significa que la red, para el mismo estímulo aplicado en dos momentos diferentes, debe terminar en el mismo estado, o al menos en uno muy parecido. De lo contrario, se podría dar el extraño caso de que una misma entrada sea clasificada como algo completamente diferente tan sólo por el momento en que haya sido alimentada al sistema, y con independencia de la información que realmente contenga. Es decir, deseamos que el sistema sea invariante en el tiempo.

A primera vista, lograr la congruencia entre los estados a los cuales nos conduce un mismo estímulo, con independencia del momento en que es inyectado, no parece posible si recordamos que, por norma general, las RB se mueven en todo momento a través de su espacio de estados, o bien, desconocemos la cuenca en la cual se pueda ubicar en ese momento, y que la posición en que se encuentre al momento de iniciar el estímulo afectará dramáticamente el estado en el que se encontrará al final del mismo.

Una posible solución consiste en que la red se encuentre en el mismo estado al inicio de todos los estímulos, de esta forma entradas iguales generarán el mismo estado final. El gran problema estriba en que, bajo ciertas circunstancias, resulta imposible prever en qué momento se llevará a cabo un estímulo o cuánto durará, por lo que forzar manualmente el estado de la red no parece viable. Además que esto no implica invarianza alguna en el tiempo. Sin embargo, existe una aproximación alternativa que, además, se muestra cualitativamente más cercana al comportamiento de las piscinas empleadas en trabajos que no usan RB.

3.3.1. Normas ideales

La distinción entre normas resulta pertinente dada la pretensión por alcanzar una dinámica y características muy particulares para la llamada norma de reposo. La ideal debe poseer sólo una cuenca de atracción y un sólo atractor puntual. Y todo debe conseguirlo de formas no triviales. No vale que el atractor sea el sucesor inmediato de cualquier estado de la red. Es decir, debe poseer un *tiempo transiente* promedio mayor a 1. Se espera que los estados de la red sigan un camino que consista en diversos estados antes de llegar al atractor. Cuantos más sean, mejor. Esto último la convertiría en altamente estable a las perturbaciones. Sin importar el tipo de perturbación al que estuviera sujeta, terminaría

por regresar a su estado atractor puntual, su punto de equilibrio, cumpliendo cabalmente así el requisito de invarianza temporal.

Es decir, se pretende que el sistema posea las características asociadas a los autómatas de clase 4 propuesta por Wolfram.

¿Por qué deseamos ese comportamiento?

Un atractor de periodo 1 favorece que el estado de inicio para cualquier estímulo sea el mismo siempre. De esta forma, dos estímulos idénticos tenderán a generar el mismo estado en la red al finalizar, sin importar el momento en que hayan sido inyectados al sistema. No lo generarán siempre, dada la posibilidad de que inicien cuando la red no se encuentre aún en su atractor, aunque sí camino a él.

Mientras una red de estas características no se encuentre en su estado de reposo, su posición actual en el espacio de estados brindará información de las perturbaciones pasadas que lo condujeron a dicho lugar. Sin embargo, asegurar la creación de una red con una sola cuenca y un solo atractor de periodo 1 es difícil. A pesar de esta dificultad, más adelante se propone un modelo que puede favorecer dicho comportamiento.

3.3.2. Memoria evanescente en normas idealizadas

Esta cualidad para redes con normas idealizadas puede ser entendida en términos de trayectorias convergentes. Unos conceptos previos:

Definición 3.2. Dado un estado x^M cualquiera de la red booleana M , sea la *trayectoria que inicia en x^M* el conjunto ordenado de estados que incluye a x^M y a sus sucesores n -ésimos.

Definición 3.3. Dadas dos trayectorias f y g diferentes entre sí, se dice que son *trayectorias convergentes* si existe un estado $x^M \in f$ y un estado $y^M \in g$, tal que $\text{succ}(x^M) = \text{succ}(y^M)$.

Dos trayectorias que pertenezcan a la misma cuenca de atracción serán siempre convergentes, compartiendo los mismos estados a partir del instante en que se unan. Esto es una consecuencia inevitable de que un estado sólo puede tener uno y sólo un sucesor, no así antecesores, que puede tener cero, uno o

más. Una cuenca de atracción también puede ser definida como el conjunto de todas las trayectorias que son convergentes entre sí.

Sí un estado cualquiera de una red booleana perturbable con normas ideales es diferente del atractor puntual de su norma de reposo, podemos decir que dicho estado almacena información sobre el flujo de entrada que lo ha provocado. Esto se debe a que sólo un subconjunto particular de entradas será capaz de conducir al sistema a dicho estado. Esta característica permite separar los flujos en el espacio de estados del sistema.

Mientras las trayectorias de dos estados diferentes al reposo no converjan, las entradas que los han provocado son distinguibles entre sí. Dada la irreversibilidad de la dinámica, una vez que lo hagan, será imposible discriminar entre las entradas que son su origen. Aún así, como conjunto, serán distinguibles de otras tantas que produzcan trayectorias a las cuales no hayan convergido hasta ese momento.

Como es fácil notar, cada vez que el recorrido de una trayectoria llega a un estado que posea más de un antecesor, hay una pérdida de «especificidad» en la misma, ya que resultará imposible distinguir entre ella y el resto de trayectorias que, por otras vías, llegan al mismo estado. No importa la norma en que suceda dicha convergencia. La información se desvanece en cada paso de tiempo que trayectorias diferentes se unen, desapareciendo por completo en el estado donde todas las trayectorias convergen: el atractor puntual de la norma de reposo.

3.3.3. Inspiración biológica

En el modelo típico de computación de reservorio se utilizan redes neuronales recurrentes como medio dinámico, aunque el principio que las rige no exige que lo sean. Tan solo es necesario una piscina proclive a ser perturbada, con la dinámica adecuada para realizar su función correctamente (Natschläger y col., 2002).

Este modelo de cómputo, creado independientemente por Maass y col. (2002) y Jäger (2001), a diferencia de otros, como las máquinas Boltzman o las redes de Hopfield donde la memoria se consigue gracias a estados estables alcanzados por el sistema, necesita un estado estable del cual partir. Es en la perturbación y el estado inestable donde se encuentra la información.

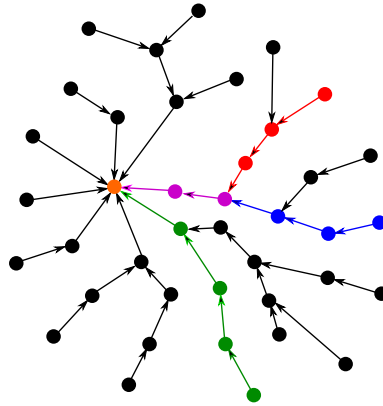


Figura 3.6: Esquema simplificado de una cuenca de atracción. Las trayectorias roja y azul avanzan hacia el atractor puntual del sistema hasta que convergen, momento a partir del cual las entradas que las generaron son indistinguibles entre sí. A pesar de ello, como conjunto son distinguibles de la entrada que generó la trayectoria verde.

En el caso particular de las *Máquinas de Estado Líquido* (MEL), una implementación particular de la computación de reservorio, se han utilizado simulaciones realistas de neuronas biológicas. Casi siempre empleando el modelo *integrar y disparar* de las mismas, propuesto en (Stein, 1967), ya que ofrece una razonable fidelidad con el comportamiento de las neuronas reales, a la vez que no resulta demasiado costoso computacionalmente respecto de otros modelos.

Piscinas creadas con este tipo de neuronas ofrecen las características deseadas, aunque se hacen aún grandes esfuerzos por dilucidar completamente las cualidades y razones que hacen a una piscina constituida de esta manera ser más adecuada que otra, según la tarea que les sea encomendada.

La inspiración para la arquitectura de las MEL proviene de la biología. En particular, sus autores originales se basaron en el trabajo de Tsodyks, Uziel y Markram (2000), para las configuraciones iniciales de las mismas. Aunque el comportamiento inicial distaba aún mucho de la dinámica neuronal real, sujeta a un número mayor de influencias aparte de los estímulos sinápticos.

No resulta trivial el estudio de tales aparatos, aunque existen esfuerzos constantes, como los realizados por conocer las particularidades que mantienen el

entramado neuronal pulsando en todo momento a un ritmo constante sin perder sus cualidades integradoras (Sussillo, Toyozumi & Maass, 2007).

3.3.4. El modelo *integrar y disparar*

Este modelo de neurona biológica, a la vez simple y razonablemente fiel a su comportamiento, ampliamente usado en la computación de reservorio, es el que ha servido de inspiración para la creación del modelo propuesto en esta tesis. Por tal motivo se reseñarán brevemente sus características y funcionamiento.

El circuito básico de un modelo *integrar y disparar* consiste en un condensador C en paralelo a una resistencia R , que conducen una corriente $I(t)$. La corriente así se divide en dos componentes, $I(t) = I_R + I_C$. El primer componente es la corriente que pasa a través de la resistencia lineal R . Tal puede ser calculada por la ley de Ohm como $I_R = \frac{u}{R}$, donde u es el voltaje a lo largo de la resistencia. El segundo componente I_C carga el condensador C . Por la definición de capacitancia $C = \frac{q}{u}$, donde q es la carga y u el voltaje. De tal suerte que $I_C = C \frac{du}{dt}$. Se tiene:

$$I(t) = \frac{u(t)}{R} + C \frac{du}{dt} \quad (3.1)$$

Multiplicamos la ecuación (3.1) por R , e introducimos la constante de tiempo $\tau_m = RC$, esto nos lleva a la forma estándar:

$$\tau \frac{du}{dt} = -u(t) + RI(t) \quad (3.2)$$

Donde u es el potencial eléctrico de la membrana neuronal, y τ_m es la constante de tiempo de la misma.

El comportamiento del potencial no está descrito explícitamente, y este se ve afectado por lo que se conoce como *pulsos* o *picos*. Estos son eventos en el tiempo, caracterizados por un «momento de pulso» t_f . El tiempo de disparo t_f se define de acuerdo a un umbral:

$$t_f : u(t_f) = \vartheta \quad (3.3)$$

Inmediatamente después de t_f el potencial de la membrana se reinicia a un nuevo valor refractario $u_r < \vartheta$:

$$\lim_{t \rightarrow t_f; t > t_f} u(t) = u_r \quad (3.4)$$

Para tiempos mayores a t_f la dinámica es dominada por la ecuación (3.2), hasta que ocurre otro cruce del umbral ϑ . La combinación de la integración simple (3.2) y el reinicio del potencial (3.4) constituyen el modelo de *integración y disparo*.

En la versión más general de este modelo, la neurona incorpora un periodo refractario absoluto, en el cual se interrumpe por completo la dinámica durante un tiempo Δ_{abs} , inmediatamente después de un evento de disparo. Pasado este periodo, la integración toma el control tal cual sucede normalmente, con el potencial de la membrana tomando el valor de u_r como es usual.

3.3.5. Redes neuronales de pulso

A las redes neuronales consistentes en neuronas de tipo *integrar y disparar*, y en general, de cualquier tipo que simule el comportamiento del potencial eléctrico neuronal y sus eventos de disparo, se les conoce como *Redes Neuronales de Picos (Spiking Neural Networks)*.

Sin embargo, las redes de este tipo no sólo consisten en el modelado neuronal. Una parte capital de su dinámica descansa en las conexiones sinápticas que unen a las neuronas. Los modelos propuestos para tal menester son acaso más variados que los creados para el comportamiento neuronal.

Los más simples de ellos envían una corriente predeterminada a la neurona post-sináptica, los más enrevesados incluyen una compleja maquinaria que determina, según sucesos previos, en qué intensidad y de qué forma responderá a un evento de disparo pre-sináptico. Esto incluye aprendizajes a corto plazo (cambios temporales en la dinámica sináptica) y a largo plazo (modificaciones permanentes en los parámetros de la conexión).

Por razones que quedan claras más adelante, profundizar en tales constructos queda fuera del alcance de este trabajo, pero no está de más enunciar su existencia.

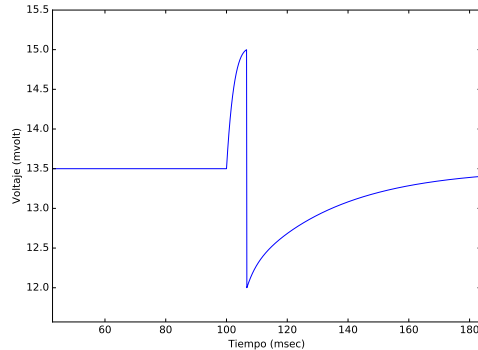


Figura 3.7: El voltaje de una neurona siguiendo el modelo *integrar y disparar*. El potencial eléctrico de la misma se mantiene en un nivel de equilibrio hasta recibir un estímulo externo. En este caso, la alcanza uno que eleva el voltaje de su membrana más allá su umbral de activación. En ese momento realiza un disparo, mientras el potencial se reduce a un valor refractario que se recupera lentamente, hasta alcanzar nuevamente su valor estable.

3.4. Redes Booleanas de Pulso

Las *Redes Booleanas de Pulso* (RBP) son la principal propuesta de este trabajo, y toman su nombre de la similitud cualitativa que existe entre ellas y las *Redes Neuronales de Picos* descritas con anterioridad.

Si bien las *Redes Neuronales de Picos* desarrolladas aquí son capaces de entrar en regímenes caóticos, tal que su actividad se vuelve frenética, también es cierto que bajo ciertas condiciones y configuraciones, son proclives a trasladarse a un estado estable mientras carezcan de estímulos, característica que las ha hecho ser propuestas en este trabajo como medios dinámicos para la computación de reservorio.

En el desarrollo presentado aquí, se pretenden simular partes de la dinámica neuronal correspondiente a dichas redes, siempre que favorezcan la existencia de dicho estado estable en carencia de estímulos, siguiendo reglas extremadamente sencillas, que sean simples de implementar y no impliquen sistemas de comunicación complejos o de alta precisión entre sus componentes, como podrían ser pesos de conexión entre los nodos y cosas similares. El modelo pre-

sentado ataño exclusivamente a cantidades y magnitudes que puedan ser expresadas como números enteros.

El propósito para tales restricciones auto-impuestas es favorecer una implementación sencilla, fiable y rápida de dichas estructuras, sea ésta en *hardware* o *software*.

Estas redes pueden ser modeladas de diversas formas: como red booleana, red de nodos multivaluados y siguiendo un modelo abstracto, que sintetiza el comportamiento global haciéndolo fácilmente comprensible, y es el detallado a continuación:

3.4.1. Modelo abstracto

El elemento fundamental de una RBP, al igual que con las RB, es el nodo, que posee K conexiones entrantes que pueden provenir, cada una, de cualquier otro nodo de la red, incluido él mismo. En cualquier instante del tiempo, un nodo a es capaz de poseer uno de dos estados posibles: «0» o «1». También posee una variable interna a_{int} y dos parámetros llamados «activación» y «refracción», representados como a_{act} y a_{ref} respectivamente. El total de nodos que componen a una RBP se divide en dos subconjuntos: *excitadores* e *inhibidores*.

Para establecer el valor de los estados y variables internas de un nodo cualquiera en el tiempo t se procede como sigue:

Sea $a_{\text{bal}}(t)$, el balance de entradas de a en t , definido de la siguiente manera:

$$a_{\text{bal}}(t) = \begin{cases} a_{\text{int}}(t-1) + \sum_i e_i(t-1) - \sum_j h_j(t-1) - 1 & : a_{\text{int}}(t-1) > 0 \\ a_{\text{int}}(t-1) + \sum_i e_i(t-1) - \sum_j h_j(t-1) + 1 & : a_{\text{int}}(t-1) < 0 \\ a_{\text{int}}(t-1) + \sum_i e_i(t-1) - \sum_j h_j(t-1) & : a_{\text{int}}(t-1) = 0 \end{cases} \quad (3.5)$$

Donde $e_i(t-1)$ representa el estado en $t-1$ del nodo excitador e_i conectado al nodo a , y $h_j(t-1)$ representa el estado en $t-1$ del nodo inhibidor h_j conectado al nodo a , y $a_{\text{int}}(t-1) \in \mathbb{Z}$.

En palabras llanas, $a_{\text{bal}}(t)$ representa el valor de la variable interna del nodo más la suma de todos los estados de los nodos calificados como «excitadores» conectados al mismo, menos la suma del estado de todos los calificados como «inhibidores» conectados también al mismo. Esto, añadiendo un pequeño valor de corrección en cada paso de tiempo, que obliga a la variable interna

a acercarse a un valor estable con ayuda de las definiciones que se presentan más adelante.

Establecer el valor de la variable interna de a en t sigue las siguientes reglas:

$$a_{\text{int}}(t) = \begin{cases} a_{\text{bal}}(t) & : a_{\text{ref}} \leq a_{\text{bal}}(t) \leq a_{\text{act}} \\ a_{\text{ref}} & : \text{en cualquier otro caso} \end{cases} \quad (3.6)$$

Donde $a_{\text{ref}} < 0$, $a_{\text{act}} > 0$, y $a_{\text{ref}}, a_{\text{act}} \in \mathbb{Z}$.

Establecer el estado del nodo a en el tiempo t se realiza de acuerdo a lo siguiente:

$$a(t) = \begin{cases} 1 & : a_{\text{bal}}(t) > a_{\text{act}} \\ 0 & : \text{en cualquier otro caso} \end{cases} \quad (3.7)$$

Con estas simples reglas se favorece la creación de redes que poseen características similares a las buscadas.

En el modelo presentado, la variable interna es el análogo del potencial eléctrico de la neurona cumpliendo la misma función: superado cierto umbral determinado por el parámetro «activación», el estado del nodo toma el valor de «1» (dispara). De otra forma no lo hace. Una vez hecho el disparo, desciende hasta tomar un valor refractario. De igual forma, la variable interna tiende a mantenerse en un mismo valor conforme pasa el tiempo, que en el caso que nos ocupa es 0.

Debe hacerse notar que el comportamiento de la variable interna es lineal al momento de acercarse a su valor estable, a diferencia del verdadero potencial eléctrico neuronal, que es exponencial.

Retardos

En el caso de las Redes Neuronales de Picos, se ha descubierto que gran parte de su poder computacional descansa en la variabilidad de la dinámica sináptica (Maass, Natschläger & Markram, 2004). Uno de los factores que enriquecen dicha dinámica son los tiempos de transmisión entre neuronas.

Para tratar de simular este fenómeno a fin de enriquecer el comportamiento de las Redes Booleanas de Pulso, pretendiendo favorecer su poder de cómputo, se ha introducido el concepto de *retardo*.

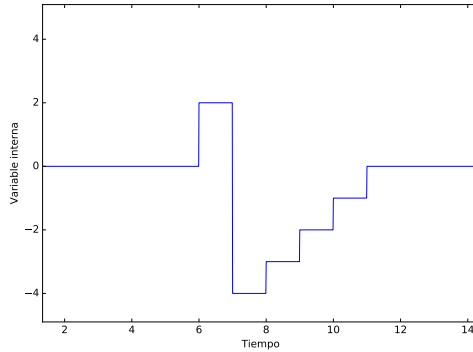


Figura 3.8: La variable interna de un nodo de pulso. De forma análoga al modelo biológicamente realista, la variable interna se mantiene en reposo hasta recibir un estímulo. En este caso, el nodo recibe uno que lo lleva más allá del umbral que lo obliga a tomar el estado «1» (no visible). A la vez, su variable interna desciende a su valor refractario, regresando paulatinamente a su punto estable. Notar su parecido con el comportamiento mostrado en la figura 3.7.

El retardo consiste en que el estado leído de un nodo por otro puede provenir de un paso de tiempo arbitrario en el pasado. En este caso, las conexiones entre nodos tienen un parámetro que indica cuantos pasos de tiempo tardará la información en llegar al nodo destino.

La implementación de los retardos implica un pequeño cambio en las reglas descritas en la ecuación (3.5). Considerando el parámetro de retardo antes mencionado, la forma definitiva sería:

$$a_{\text{bal}}(t) = \begin{cases} a_{\text{int}}(t-1) + \sum_i e_i(t+r_i) - \sum_j h_j(t+r_j) - 1 & : a_{\text{int}}(t-1) > 0 \\ a_{\text{int}}(t-1) + \sum_i e_i(t+r_i) - \sum_j h_j(t+r_j) + 1 & : a_{\text{int}}(t-1) < 0 \\ a_{\text{int}}(t-1) + \sum_i e_i(t+r_i) - \sum_j h_j(t+r_j) & : a_{\text{int}}(t-1) = 0 \end{cases} \quad (3.8)$$

Donde r_i y r_j son los parámetros de retardo asociados a la conexiones con los nodos excitadores e inhibidores e_i y h_j , respectivamente. Siendo $r_i, r_j < 0$ y $r_i, r_j \in \mathbb{Z}$.

3.4.2. Como red de nodos multivaluados

A fin de hacer válidos los razonamientos sobre la perturbación y las normas a las Redes Booleanas de Pulso, es menester mostrar que existen redes booleanas clásicas equivalentes a las mismas. Como paso intermedio para conseguir esto, se muestra que son equivalentes a redes con *nodos multivaluados*.

Un nodo multivaluado es aquel que puede tomar más de dos valores como estado, a diferencia de los nodos binarios que sólo toman dos. Las redes con este tipo de nodos tienen funciones donde las reglas de transición aceptan también valores múltiples. Por lo demás, son iguales a las redes booleanas clásicas.

A fin de explicar su funcionamiento como Redes Booleanas de Pulso, han de seguirse las siguientes convenciones:

- Para cada nodo multivaluado existirán tantos valores efectivos diferentes como la suma entre los parámetros de «activación» y «refracción» más dos. Es decir: $a_{\text{act}} + a_{\text{ref}} + 2$. Recuérdese que $a_{\text{ref}} < 0$.
- Los estados de los nodos serán etiquetados como números naturales en el rango $[a_{\text{act}}, a_{\text{ref}}]$. El estado restante se etiquetará como «D» (Disparo).
- Además de las conexiones entre nodos de la RBP original, todo nodo deberá tener una conexión a sí mismo, a fin de acceder a información sobre su estado actual.

Retardos en nodos multivaluados

La implementación de los retardos en redes con nodos multivaluados pasa por la creación de nodos especiales, que conectados a otro «copien» su estado un paso atrás en el tiempo. Es decir:

$$h_a(t) = a(t - 1) \tag{3.9}$$

Donde $a(t - 1)$ es el estado del nodo a en el tiempo $t - 1$, y h_a es el nodo que ha «copiado» su estado en el tiempo t . Este comportamiento se puede implementar fácilmente con reglas de transición tradicionales. Recordemos que el estado de un nodo en t está en función del estado de los nodos con los que está conectado en $t - 1$.

Al nodo h_a lo llamaremos *nodo historia de a*, en cuanto toma los mismos estados que a un paso de tiempo después.

Como es evidente un *nodo historia* puede tener, a su vez, su propio *nodo historia*, lo que tiene como efecto la propagación en el tiempo del estado original, tantos pasos hacia adelante como *nodos historia* se encuentren conectados en cadena.

Así, para simular el comportamiento de una conexión con retraso, basta conectarse al nodo historia en la cadena que corresponda al número de pasos en el tiempo que dicho retraso especifique.

Reglas para nodos multivaluados

Siguiendo estas convenciones y reglas, resulta sencillo traducir el comportamiento de una Red Booleana de Pulsos a una red con nodos multivaluados como las descritas. Cada fila en las tablas de transición de los nodos se construirá de esta manera:

Si la etiqueta para el estado del nodo actual es «D»:

1. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», más la etiqueta de valor más pequeño para ese nodo + 1, corresponde a cualquier etiqueta existente, asignar al nodo la etiqueta que corresponda a dicha suma.
2. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», más la etiqueta de valor más pequeño para ese nodo + 1, es mayor a cualquier etiqueta existente, asignar al nodo el estado «D».
3. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», más la etiqueta de valor más pequeño para ese nodo + 1, es menor a cualquier etiqueta existente, asignar al nodo el estado con la etiqueta de menor valor.

Si la etiqueta para el estado del nodo actual es «0»:

1. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», corresponde a cualquier etiqueta existente, asignar al nodo la etiqueta que corresponda a dicha suma.

2. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», es mayor a cualquier etiqueta existente, asignar al nodo el estado «D».
3. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», es menor a cualquier etiqueta existente, asignar al nodo el estado con la etiqueta de menor valor.

Si la etiqueta para el estado del nodo actual es mayor a 0:

1. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», más la etiqueta actual del nodo - 1, corresponde a cualquier etiqueta existente, asignar al nodo la etiqueta que corresponda a dicha suma.
2. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», más la etiqueta actual del nodo - 1, es mayor a cualquier etiqueta existente, asignar al nodo el estado «D».
3. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», más la etiqueta actual del nodo - 1, es menor a cualquier etiqueta existente, asignar al nodo el estado con la etiqueta de menor valor.

Si la etiqueta para el estado del nodo actual es menor a 0:

1. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», más la etiqueta actual del nodo + 1, corresponde a cualquier etiqueta existente, asignar al nodo la etiqueta que corresponda a dicha suma.
2. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», más la etiqueta actual del nodo + 1, es mayor a cualquier etiqueta existente, asignar al nodo el estado «D».

3. Si la cantidad de nodos excitadores entrantes con estado «D», menos la cantidad de nodos inhibidores entrantes con estado «D», más la etiqueta actual del nodo + 1, es menor a cualquier etiqueta existente, asignar al nodo el estado con la etiqueta de menor valor.

No se debe olvidar que en el caso de conexiones con retraso, ésta debe realizarse no al nodo original, sino al nodo historia correspondiente.

Para obtener el estado binario actual de la red, baste lo siguiente:

$$a_{\text{bin}}(t) = \begin{cases} 1 & : a(t) = \text{«D»} \\ 0 & : \text{en cualquier otro caso} \end{cases} \quad (3.10)$$

3.4.3. Como red booleana

De la misma forma que existen redes multivaluadas equivalentes a Redes Booleanas de Pulso, hay también redes booleanas clásicas equivalentes a redes multivaluadas. Sin embargo, suele ser necesario emplear múltiples nodos binarios para simular un solo nodo multivaluado.

Para el caso general, es necesario utilizar al menos $\lceil \log_2 v \rceil$ nodos binarios por cada nodo en la red multivaluada, donde v es el número de estados diferentes del nodo mencionado, para hacer posible capturar todos los valores del mismo. A su vez, dichos nodos binarios deben formar un grafo completo entre sí, incluyendo conexiones a sí mismos, a fin de verse alimentados con toda la información equivalente al estado del nodo multivaluado original. De esa forma podrán trabajar al unísono para crear las secuencias binarias correspondientes.

Para establecer la salida binaria que corresponda al nodo original de la Red Booleana de Pulsos, hará falta un nodo de salida s conectado a todos los del conjunto que lo representan. Este nodo de salida tomará el valor de 1 cuando la combinación binaria del conjunto corresponda al estado asignado con la etiqueta «D» de la red multivaluada equivalente. Tomará el valor de 0 en cualquier otro caso. Esta será la única información de salida de este conjunto a otros conjuntos similares.

Todas las entradas que provengan del exterior deberán alimentar a todos los nodos del conjunto que representa a un nodo multivaluado. Es decir, todos y cada uno de los nodos binarios involucrados en representar la secuencia bina-

ria asociada a cada una de las etiquetas de los nodos multivaluados, reciben la misma información.

Dicho esto es fácil ver que las reglas que gobiernan los nodos binarios tradicionales de un conjunto que representa a un nodo multivaluado, son una importación directa de las reglas de dicho nodo multivaluado, con la salvedad de que cada nodo binario tomará el valor que le corresponde en la secuencia binaria que represente a la etiqueta del nodo multivaluado asociado.

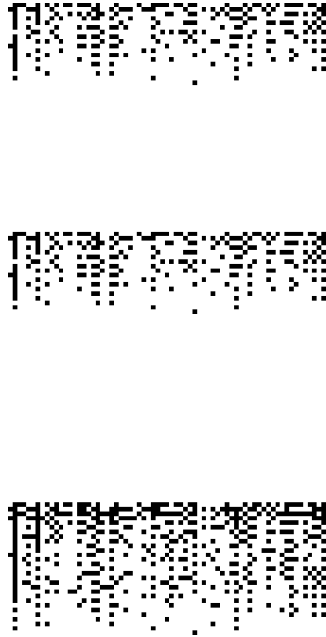


Figura 3.9: Evolución temporal de una Red Booleana de Pulsos, tal cual se describe en el texto, con $N = 70$ y $\langle K \rangle = 5$. El tiempo fluye hacia abajo. La red recibe un estímulo al inicio y uno igual un poco más tarde. Notar que los estados transitados en ambos casos son idénticos antes de alcanzar el estado de reposo y que la perturbación se desvanezca. Más tarde recibe un tercer estímulo, diferente a los dos anteriores, que lo hacen recorrer un camino alterno a través de su espacio fásico. Observar que los dos últimos estados antes del reposo son idénticos para las perturbaciones pequeñas y la mayor. En ese punto, las entradas que las han provocado son indistinguibles entre sí para el sistema.

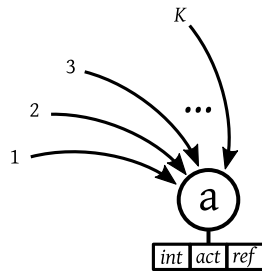


Figura 3.10: El nodo de una Red Booleana de Pulsos con K entradas. Éste consiste en el nodo mismo y tres valores asociados. *int*: que alberga el valor de la variable interna del nodo. *act*: un parámetro que señala el umbral de activación del nodo. *ref*: el valor mínimo y refractario que toma la variable interna una vez acaecido un disparo del mismo.

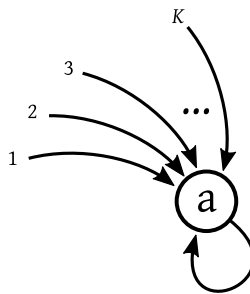


Figura 3.11: Un nodo multivaluado que equivale al nodo binario de una Red Booleana de Pulsos con K entradas. El nodo debe mantener una conexión a sí mismo, hecho que altera la topología de la red original, a fin de hacerse posible la existencia de reglas que hagan su comportamiento análogo al nodo de una Red Booleana de Pulsos equivalente.

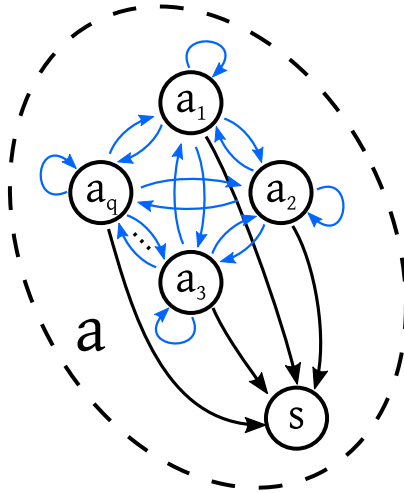


Figura 3.12: Conjunto de nodos binarios clásicos equivalente a un nodo de pulso a . Para que sea posible incorporar todos los valores de la variable interna y la condición de disparo de un nodo de pulso, es menester emplear tantos nodos binarios como sean necesarios, para que las permutaciones de sus estados permitan identificar de forma única todos los valores posibles del nodo multivaluado que representan. Deben formar un grafo completo entre ellos, incluyendo una conexión a sí mismos. Añadido a esto, un nodo de salida s que brinde la señal binaria correspondiente al exterior.

Capítulo 4

Mediciones

Information is the resolution of uncertainty.

«La información es la resolución de la incertidumbre».

Claude E. Shannon

Siendo el propósito de este trabajo evaluar el desempeño de las Redes Booleanas de Pulso como medios dinámicos para un sistema de computación de reservorio, así como la búsqueda de características que potencien dicha cualidad, se vuelve obligatorio establecer criterios que permitan caracterizarlas y estimar su capacidad computacional.

A pesar de estar construidas con entidades muy sencillas, dominadas por reglas muy simples, una aproximación formal y analítica de su comportamiento se vuelve muy difícil, quizá hasta imposible, dada la extrema complejidad que implica la interacción de dichos elementos entre sí.

Emparentadas como están a los más elementales autómatas celulares, que a pesar de su sencillez presentan *irreducibilidad computacional* de acuerdo al concepto expuesto en (Wolfram, 2002), no resultaría extraño que el mismo caso se presentara en estas redes⁸.

Esperando de ellas propiedades emergentes, difícilmente deducibles al analizar por separado a sus elementos constituyentes, se vuelve pertinente abordar el problema de forma similar a la utilizada con entidades complejas, dirigiendo la atención a las propiedades colectivas que surjan de la dinámica implícita en el sistema.

⁸La *irreducibilidad computacional* hace alusión al concepto de que resulta imposible, para algunos sistemas o programas, pronosticar y conocer su comportamiento de antemano, salvo que efectivamente se «echen a andar».

A fin de entrar en materia, resulta necesario abordar conceptos fuertemente relacionados con los criterios elegidos. Dichas nociones, y las medidas asociadas a ellas, se presentan a continuación.

4.1. Teoría matemática de la comunicación

La teoría de la información, tal y como la conocemos hoy día, fue introducida por Claude E. Shannon en 1948. Momento en que se caracterizó la información y su medida (Shannon, 1948). Tiene su origen en el contexto de las telecomunicaciones, y surgió como parte del análisis aplicado para abordar el problema de recuperar la información transmitida a través de un canal de comunicación afectado por el ruido.

Shannon estaba profundamente interesado en medir la «cantidad de información» que existía en un conjunto de datos. En su modelo la información puede ser vista como una cadena de símbolos $X = x_0x_1 \dots x_n$, donde cada x es un símbolo tomado de un conjunto finito de símbolos \mathcal{A} llamado el *alfabeto*.

Cada símbolo tiene asociada una probabilidad p_i de aparición en la cadena. Mientras mayor sea el número de ocurrencias que tenga dicho símbolo en la misma, tanto mayor será su probabilidad. Tales probabilidades son lo único que se sabe respecto a la presencia de algún símbolo en particular.

De acuerdo a Shannon es deseable que una medida de información, llamada aquí E , cumpla las siguientes propiedades:

1. E debe ser continua para toda p_i .
2. Si todas las p_i son iguales, $P(x_i) = \frac{1}{n}$, entonces E debe ser una función monótona creciente de n . Con n eventos igualmente probables hay más posibilidades de elección o incertidumbre, dado que son los más posibles.
3. Si una elección se divide en dos elecciones sucesivas, el valor de E original debe resultar de la suma ponderada de los valores individuales de E para cada evento.

4.1.1. Entropía

Con base en los principios dados anteriormente Shannon encuentra una función que los cumple, definiendo el promedio de información contenida en un

símbolo como la entropía E de una fuente de información F . Dicha entropía se expresa como:

$$E = -K \sum_{i=1}^n P(x_i) \log P(x_i) \quad (4.1)$$

Donde K es una constante positiva a la que usualmente, por conveniencia y comodidad, se le asigna el valor de 1. La base del logaritmo es de elección arbitraria. Sin embargo, por convención será utilizada la base 2.

Lo anterior representa la cantidad de información que contiene la fuente F medida en *bits* por esa última razón.

4.2. Auto-organización

De acuerdo con Gershenson y Heylighen (2003), la auto-organización de un sistema está correlacionada con un incremento en su orden, es decir, el inverso de su entropía. A mayor entropía menor orden, y viceversa.

Siguiendo esta premisa, podemos definir a la auto-organización de un sistema como:

$$S = 1 - E \quad (4.2)$$

Siempre que E se encuentre confinada al intervalo $[0, 1]$.

De acuerdo a los autores mencionados, se puede definir como una función $S : \Sigma \rightarrow \mathbb{R}$, donde $\Sigma = \mathcal{A}^{\mathbb{N}}$, con las propiedades siguientes:

1. El rango de S es el intervalo real $[0, 1]$.
2. $S(X) = 1$ si y solo si X es determinista, por ejemplo, si conocemos de antemano el valor de un proceso.
3. $S(X) = 0$ si y solo si X tiene una distribución de probabilidad uniforme. Por ejemplo, cuando cualquier estado del proceso es igualmente probable.
4. $S(X)$ es la correlación negativa de la entropía E (que los autores llaman «emergencia de información» en este contexto).

Con esto en mente, se puede concluir que $S = 1$ representa la máxima auto-organización, el máximo orden. Un estado donde no se produce nueva información. Por el contrario, cuando $S = 0$ se tiene el estado de menor orden y la información se produce en la máxima medida. Nada en el pasado nos indica qué sucederá a continuación.

4.3. Complejidad

Existen muy variadas y diversas definiciones de complejidad, cada una conveniente para el tópico del que se trate. Para este trabajo resulta relevante una definición de complejidad acorde al comportamiento de las variables a través del tiempo. De acuerdo a Lopez-Ruiz, Mancini y Calbet (2002), la complejidad se puede definir como un balance entre el cambio y la estabilidad. Basándonos en la nociones presentadas en (Fernández, Maldonado & Gershenson, 2014), el cambio está asociado a la emergencia, que se encuentra correlacionada a la entropía. Por otra parte, la estabilidad está asociada a la auto-organización, que se encuentra correlacionada con el inverso de la entropía, tal cual se ha visto en la sección precedente.

Con base en estas nociones, se da la siguiente definición de complejidad:

$$C = 4ES \quad (4.3)$$

Donde E es la entropía tal cual se ha explicado anteriormente. S es la auto-organización, que se define como $S = 1 - E$. La constante 4 tiene el propósito de normalizar la medida, de forma que exista en el rango $[0, 1]$.

La información de Shannon, para cadenas binarias, puede verse como un balance entre ceros y unos. La complejidad C , puede verse como un balance entre E y S .

4.4. Homeostasis

Mientras que las medidas previas muestran cómo una sola variable cambia a través del tiempo, la homeostasis pretende cumplir un papel similar para el sistema entero. Deseamos saber cómo cambian todas las variables del sistema conforme el tiempo pasa. Por tal motivo, es una medida enfocada en los estados

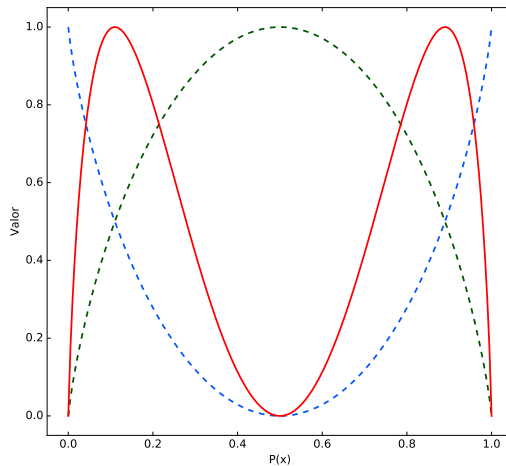


Figura 4.1: Relación de entropía, auto-organización y complejidad. Entropía en verde discontinuo. Auto-organización en azul discontinuo. Complejidad en rojo continuo.

del sistema, no en sus componentes individuales. También puede entenderse como una tasa de cambio a través del tiempo para los estados de un sistema.

En (Fernández y col., 2014) se presenta también una propuesta de dicha medida. Sea $X = x_1x_2\dots x_n$ la cadena que representa el estado de un sistema de n variables. Si tiene una homeostasis alta, es de esperar que sus estados no cambien mucho en el transcurrir del tiempo. La función homeostasis $H : \Sigma \times \Sigma \rightarrow \mathbb{R}$, que pretende medir estas propiedades debe tener las siguientes características:

1. Debe existir en el intervalo $[0, 1]$.
2. $H = 1$ si y solo si para los estados X y X' se cumple que $X = X'$. Es decir, no ha existido un cambio.
3. $H = 0$ si y solo si $\forall i, x_i \neq x'_i$. Es decir, todas las variables del sistema han cambiado.

En el caso que nos compete, constituido por variables binarias, un recurso útil al momento de comparar cadenas es la distancia de Hamming. Dicha fun-

ción mide el número de símbolos diferentes entre dos cadenas binarias cualesquiera, siempre que posean la misma longitud.

Como una homeostasis alta está asociada a la robustez y poco cambio en las variables internas de un sistema y viceversa, se toma como tal el promedio del inverso del porcentaje de símbolos diferentes entre estados consecutivos, para una cantidad de pasos de tiempo dado. Es decir:

$$H = 1 - \frac{1}{\mathcal{T} - 1} \sum_{t=1}^{\mathcal{T}-1} d(X_{t-1}X_t) \quad (4.4)$$

Donde \mathcal{T} es el número total de pasos de tiempo analizados, y $d(X_{t-1}, X_t)$ es la proporción de símbolos diferentes que existe entre los estados X_{t-1} y X_t .

4.4.1. Homeostasis y auto-organización

Ambas medidas se encuentran fuertemente relacionadas, y para muchas situaciones las medidas son similares. Para ilustrar el punto se presenta el ejemplo siguiente:

Considérese la cadena presentada a continuación como la evolución temporal del estado de un nodo binario:

010101010101010101010101010101010101

Tanto el valor de la auto-organización como de la homeostasis, siguiendo los criterios establecidos con anterioridad, será igual a 0. La entropía es máxima, y a cada paso de tiempo el estado del sistema (el nodo) cambia por completo. En casos como este $H \approx S$, siempre que los valores de cero y uno se distribuyan uniformemente a lo largo de la cadena.

El caso siguiente es diferente:

00000000000000000000111111111111111111

La entropía sigue siendo máxima, ya que la probabilidad de encontrar cualquier símbolo es la misma. Sin embargo, sólo en una ocasión se presenta un cambio de estado en el sistema. Aquí la homeostasis es mucho mayor, cercana a 1.

La homeostasis es un tipo de auto-organización, que toma en consideración la relación entre valores o estados adyacentes.

4.5. Autopoiesis

La palabra proviene de los vocablos griegos *auto* (uno mismo) y *poiesis* (creación, producción), y el concepto nace en el contexto de la biología, en la pretensión de caracterizar los procesos moleculares que toman lugar en la dinámica de los seres vivos como entidades autónomas.

En el contexto de la autopoiesis, la vida ocurre en agentes que continuamente se crean o producen a sí mismos, haciendo alusión a la interacción que existe entre un sistema y su entorno. En este sentido, está fuertemente relacionada con la autonomía. Ésta se encuentra siempre limitada en sistemas abiertos, de forma que sus estados dependen de las interacciones con aquello que los rodea.

La autopoiesis pretende ser una medida de la autonomía que un sistema posea con su entorno. De acuerdo a Fernández y col. (2014), la autopoiesis A debe cumplir las siguientes propiedades:

Si \bar{X} representa la trayectoria de los estados de un sistema, y \bar{Y} la trayectoria de estados de su entorno, entonces:

1. $A \geq 0$
2. A debería reflejar la independencia de \bar{X} sobre \bar{Y} , por lo tanto:
 - a. $A > A' \Leftrightarrow \bar{X}$ produce más información que \bar{X}' para una \bar{Y} dada.
 - b. $A > A' \Leftrightarrow \bar{X}$ produce más información en \bar{Y} que en \bar{Y}' .
 - c. $A = A' \Leftrightarrow \bar{X}$ produce tanta información como \bar{X}' para una \bar{Y} dada.
 - d. $A = A' \Leftrightarrow \bar{X}$ produce tanta información en \bar{Y} como en \bar{Y}' .
 - e. $A = 0$ si toda la información producida en \bar{X}' es producida por \bar{Y} .

Y se calcula como la razón entre la complejidad del sistema sobre la complejidad de su entorno. Es decir:

$$A = \frac{C(\bar{X})}{C(\bar{Y})} \quad (4.5)$$

La autopoiesis, para un conjunto de sistemas sujetos a un entorno de complejidad constante, se distribuirá de la misma manera que la complejidad del sistema mismo. Sistemas con alta complejidad poseerán una alta autopoiesis y

viceversa. Salvo que la información se mostrará con valores diferentes. La autopoiesis tiene la facultad de brindar información relativa a la complejidad del sistema en relación a su entorno.

Siguiendo la definición expuesta arriba, queda claro que un valor de $A > 1$ revela una complejidad mayor para el sistema que para el entorno. El caso contrario para $A < 1$, donde el entorno expresa mayor complejidad que el sistema. El primer caso se interpreta como un sistema que genera por sí mismo los patrones y comportamientos que revela por encima de los que la propia entrada induce en él. Caso contrario en el segundo caso, donde el sistema expresa menor complejidad que el estímulo que recibe.

4.6. Separación

El desempeño de una piscina viene, en parte, por su capacidad de separar entradas, de forma que se mapeen de forma distinguible en el espacio de estados del sistema. Al mismo tiempo, deseamos que tenga capacidad de generalizarlas, es decir, que los mapeados de perturbaciones ligeramente diferentes no difieran de forma radical, favoreciendo su clasificación en la misma clase.

El equilibrio de estas cualidades en conflicto (una gran distancia para entradas diferentes vs. una distancia pequeña para entradas con poca diferencia entre sí) se considera indispensable para el correcto desempeño de la piscina. Por tal motivo, se presenta a continuación un conjunto de medidas que permita estimar dichas capacidades.

De acuerdo con Maass, Legenstein y Bertschinger (2004) una forma de estimar tales cualidades en una piscina estriba en calcular la *calidad del kernel* (KQ) y su *rango de generalización* (GR).

El primero de ellos establece qué tan bien la piscina es capaz de separar diferentes entradas de datos. El segundo, qué tan bien mapea entradas similares a estados también similares.

De la combinación de ambas propiedades emerge la *propiedad de separación* de la piscina (No confundir con la *separación puntual*). Si ella es capaz de tener un alto valor para KQ y simultáneamente uno bajo para GR, su propiedad de separación se considera alta.

4.6.1. Calidad del kernel y rango de generalización

La calidad de kernel (KQ) se define formalmente como el rango de la matriz compuesta por el conjunto de estados finales a los que llega la piscina una vez inyectado con un conjunto de entradas diferentes entre sí, un estado para cada entrada.

El rango de una matriz es la dimensión de la imagen de dicha matriz, que corresponde al número de filas o columnas linealmente independientes de la misma. No resulta relevante si el cálculo se realiza por fila o columna. Como es sabido, ambos resultados son siempre el mismo (Wardlaw, 2005).

El rango de una matriz B de tamaño $n \times m$ cumple la siguiente propiedad:

$$\text{rango}(B) \leq \text{mín}(n, m) \quad (4.6)$$

Con esto en mente, digamos que sea F una piscina con n nodos y $\{u_0, u_1, \dots, u_m\}$ un conjunto de m entradas de longitud \mathcal{T} inyectadas al mismo. Sea W_{KQ} una matriz de tamaño $n \times m$, tal que sus columnas son los estados $Fu_i(\mathcal{T})$, es decir, los estados de la piscina en el tiempo \mathcal{T} una vez iniciada la inyección de cada entrada. La calidad del kernel se define como:

$$\text{KQ} = \text{rango}(W_{KQ}) \quad (4.7)$$

Cuando la matriz tiene un rango igual a m , se *garantiza* que es posible realizar una clasificación binaria perfecta a través de una unidad de lectura lineal, para cualquiera de las 2^m posibles asignaciones de clase que se pueden aplicar a esas m entradas.

Si el valor del rango es menor a m , el valor puede ser visto como una forma de medir la potencia de cálculo de la piscina, ya que el rango es el número de «grados de libertad» que una lectura lineal tiene al asignar las salidas y_i para ese conjunto particular de entradas u_i .

Esta medida es una manera empírica de estimar la calidad del kernel que comprende a la piscina, ya que nos habla indirectamente de la complejidad y diversidad de las operaciones no lineales efectuadas por él, con el fin de impulsar la clasificación lineal o cuasi-lineal aplicada al mismo.

Otro aspecto a considerar para estimar la capacidad de cómputo es su habilidad para generalizar su respuesta a nuevas entradas no conocidas previamente por él. Un criterio matemático para la misma surge de la teoría de aprendizaje estadístico (Vapnik, 1998). De acuerdo a ella es posible cuantificar la capacidad

de generalización en términos de la dimensión Vapnik-Chervonenkis (VC), que se define como la cardinalidad del mayor conjunto de puntos que un modelo de clasificación puede separar.

En otras palabras: un modelo de clasificación Θ con parámetros θ se dice que separa un conjunto de entradas $\{u_1, u_2, \dots, u_m\}$ si, para todas las posibles configuraciones de las mismas, existe un conjunto de parámetros θ que permitan al modelo Θ no obtener errores al momento de evaluar dicho conjunto de entradas.

La dimensión VC del modelo será el máximo valor de c para el cuál cualquier entrada del conjunto de tamaño c puede ser separado por Θ .

Según Maass, Legenstein y Bertschinger (2004), es posible utilizar el rango de generalización como un estimador de la dimensión VC de la piscina, considerando que es extraordinariamente difícil calcularla aún para piscinas consistentes en pocos elementos (Bartlett & Maass, 2003)⁹. Si el valor de la dimensión VC es sustancialmente menor que el tamaño del conjunto de datos, se puede argumentar que el sistema generaliza bien, ya que tendrá la tendencia a clasificar bajo la misma clase a multitud de entradas diferentes.

Ellos proponen utilizar el rango de una matriz, que aquí llamaremos W_{GR} , como una estimación de la dimensión VC de la piscina. Siendo W_{GR} una matriz constituida de forma similar a W_{KQ} , pero utilizando como columnas los estados $x_{v_i}^M(\mathcal{T})$, obtenidos gracias a un conjunto de entradas $\{v_0, v_1, \dots, v_m\}$, tal que las entradas v_i son versiones ligeramente alteradas de una sola.

Notar que «ligeramente alteradas» es un término vago, que depende del contexto y sistema de codificación elegido para las entradas. Por ejemplo, si la información contenida en la entrada está codificada en los tiempos precisos que existen, digamos, entre símbolos «1», alterar la posición relativa de los mismos consistirá en una alteración significativa de la entrada. Por el contrario, si la información se encuentra codificada en la proporción que existe de un símbolo respecto al otro, la posición de los mismos no será importante, tanto como su cantidad.

Considerando todo lo anterior, la propiedad de separación Υ es calculada como la diferencia entre ambas medidas. Es decir:

$$\Upsilon = KQ - GR \tag{4.8}$$

⁹Si bien ellos hacían alusión a redes constituidas por neuronas de pulso, la dificultad persiste para las redes booleanas.

Un valor alto para Υ implica una diferencia significativa entre el valor de KQ y GR . Siendo ambos forzosamente positivos, esto significa un alto valor para el rango de la matriz asociado a las entradas no relacionadas entre sí (un mapeo a un espacio de alta dimensionalidad), y un rango pequeño para la matriz asociado a las entradas consideradas similares (un mapeo a un espacio de baja dimensionalidad), características que se corresponden con el comportamiento deseado.

4.7. Evanescencia

Resulta capital que la piscina tenga la capacidad de «olvidar» las perturbaciones con el paso del tiempo, de lo contrario, la función para la que ha sido construido no podría llevarse a cabo con éxito.

Para entender la necesidad que existe para cumplir con tal condición considérese la situación contraria: una piscina que no tenga la capacidad de «olvidar». Entiéndase con tal vocablo que existe un momento en el tiempo para el cual un estado del sistema no posee información de los sucesos que han acontecido a cierta distancia en el pasado.

Un sistema incapaz de olvidar debería, en principio, ser capaz de asignar un estado diferente para cada una de las infinitas configuraciones de entradas y perturbaciones a las cuales ha estado sujeto en un intervalo indeterminadamente largo en el pasado. Sin embargo, esto es imposible. Considerando que el espacio de estados de una red booleana es finito, de forma obligatoria un conjunto de entradas (una cantidad infinita de ellas), tendría asignado el mismo estado de la red, volviendo imposible discriminar entre aquellas que pertenecen al mismo conjunto. El estado de la red no sería invariante en el tiempo, sino completamente dependiente de él. Así, una perturbación particular acaída en el tiempo, se proyectaría a un estado del sistema dinámico que sería dependiente del momento en que dicho estímulo fuera inyectado en el mismo. Y ese no es el comportamiento deseado para la piscina.

A fin de caracterizar la capacidad que posean las redes booleanas para manifestar rasgos de memoria evanescente (en el sentido dado en la definición 2.1), se utiliza una medida complementaria de análisis que se ha nombrado aquí *evanescencia de la piscina*, y se basa en la difusión que tienen las perturbaciones dentro de la piscina a lo largo del tiempo. Tal medida se basa en lo siguiente:

Sea M una red con n nodos, a y b con longitud \mathcal{T} dos flujos de entrada a la misma. Llamemos *perturbación de M entre las entradas a y b en \mathcal{T}* a:

$$\text{Per}(M, a, b, \mathcal{T}) = d(x_a^M(\mathcal{T}), x_b^M(\mathcal{T})) \quad (4.9)$$

Donde $d(x_a^M(\mathcal{T}), x_b^M(\mathcal{T}))$ es la proporción de símbolos diferentes que existe entre $x_a^M(\mathcal{T})$ y $x_b^M(\mathcal{T})$, que son los estados que alcanza la piscina una vez finalizada la inyección por separado de las entradas a y b respectivamente, iniciando desde el mismo estado inicial de la misma.

Es deseable que exista una gran perturbación para entradas diferentes, no así para aquellas que sean similares. La pregunta es: ¿similares desde cuándo? Para responder esta cuestión, podemos analizar la perturbación para otras dos entradas, también diferentes, pero que son idénticas en los últimos τ pasos de tiempo. Es de esperar que la diferencia se reduzca dada la presunta capacidad del medio dinámico para olvidar las perturbaciones pasadas, y al mismo tiempo, verse dominada gradualmente por las entradas más recientes.

Considerando lo anterior, midamos la evanescencia de la piscina de la siguiente manera:

Sean a y b flujos de entrada de longitud \mathcal{T} , tales que si $a_i = 0$, entonces $b_i = 1$. De igual forma, si $a_i = 1$, entonces $b_i = 0$, y sean establecidos los símbolos de una tercer entrada c de igual longitud de esta manera:

$$c_i = \begin{cases} b_i & : \text{Si } i < \mathcal{T} - \tau \\ a_i & : \text{en caso contrario} \end{cases} \quad (4.10)$$

Llamemos *evanescencia de M para τ* a lo siguiente:

$$\text{Eva}(M, \tau) = \text{Per}(M, a, b, \mathcal{T}) - \text{Per}(M, a, c, \mathcal{T}) \quad (4.11)$$

Si la piscina ha «olvidado» las perturbaciones ocasionadas por la diferencia inicial de las entradas, y su dinámica se ve gobernada exclusivamente por los símbolos más recientes, la evanescencia es máxima. Es de notar que el valor de la evanescencia, si existe, está en función del tiempo τ elegido para realizar la medida. A pesar de ello, la medida es un indicador sobre la posibilidad de que la piscina posea efectivamente una memoria evanescente.

4.8. Taza de desvanecimiento

La existencia de evanescencia en el tiempo τ , no basta para asegurar que la piscina se comporta de la manera deseada. No sólo se pretende que la misma se vea influenciada por los estímulos más recientes, es también aconsejable que la influencia de los más antiguos se pierda de forma gradual y no demasiado rápido. Una pérdida paulatina de la memoria es indicador de largos transientes y le permitiría, en principio, integrar información temporal por más tiempo.

Se propone aquí la *taza de desvanecimiento*, que pretende estimar la velocidad a la cual se pierde la memoria del sistema. Para conseguir tal propósito se pretende utilizar la razón de cambio que posee la distancia entre estados de una piscina, cuando la misma es sometida a entradas idénticas dos veces de manera independiente, pero iniciando la inyección en un estado inicial diferente del sistema cada vez.

Este caso, la inyección de la misma entrada iniciando en dos estados del sistema diferentes, se presenta en la medición de la evanescencia comentada en el punto anterior (siempre y cuando el sistema posea la capacidad de separación). Se pretende aprovechar tal circunstancia con el fin de realizar esta medida.

Sea $m_{uv}(t, \Delta)$ equivalente a la expresión siguiente:

$$m_{uv}(t, \Delta) = \frac{\text{Per}(M, u, v, t + \Delta) - \text{Per}(M, u, v, t)}{\Delta} \quad (4.12)$$

Donde u y v son entradas al sistema M . Como es evidente $m_{uv}(t, \Delta)$ denota la pendiente para la perturbación entre u y v de la red M en el intervalo temporal que va de t a $t + \Delta$.

Abreviando la expresión para incrementos de tiempo unitarios, tenemos:

$$m_{uv}(t) = \text{Per}(M, u, v, t + 1) - \text{Per}(M, u, v, t) \quad (4.13)$$

Sean las entradas a y c , tal cual se encuentran definidas en la ecuación (4.10), \mathcal{T} su longitud y τ la cantidad de símbolos que son idénticos al final de las mismas. Con auxilio de la expresión anterior, la tasa de desvanecimiento queda expresada como sigue:

$$\text{Des}(M, \tau) = \sum_{i=\mathcal{T}-\tau}^{\mathcal{T}-2} m_{ac}(i) - m_{ac}(i + 1) \quad (4.14)$$

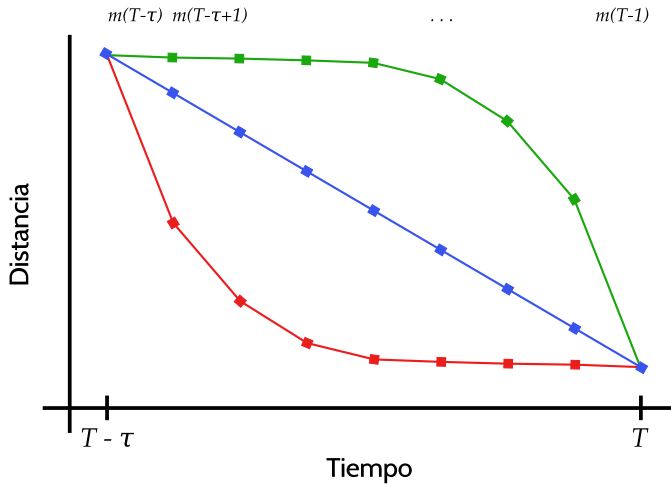


Figura 4.2: Gráfica esquemática de la *taza de desvanecimiento*. Si $m(i)$ es la pendiente del segmento que va de i a $i + 1$, para una reducción lineal en la distancia entre estados a través del tiempo, la suma de las diferencias $m(i) - m(i + 1)$ para todos los segmentos entre $\mathcal{T} - \tau$ y \mathcal{T} será próxima a cero (azul). Por el contrario, si la distancia se reduce rápidamente, al ser las pendientes cada vez mayores, la suma de las diferencias será negativa (rojo). En los casos donde la distancia se reduzca con mayor velocidad conforme pase el tiempo, la suma de las diferencias tenderá a ser positiva (verde).

Dicha medida puede verse como la suma de las diferencias entre pendientes del cambio de distancia entre vectores de estado sucesivos generados por a y c . Para valores cercanos a cero, la evanescencia es casi lineal. Lecturas negativas se interpretan como una pérdida de memoria prematura y rápida. Los extraños casos en que el valor sea positivo, deben ser interpretados como una memoria que se pierde lentamente al inicio, que en momentos posteriores lo hace rápidamente.

Esta lectura puede interpretarse también como la resistencia que el sistema ofrece para alterar su trayectoria ante cambios en la entrada. Sólo nos da información sobre la evanescencia cuando está acompañada de un alto valor para la misma.

Capítulo 5

Configuración experimental

It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong.

«No importa lo bella que sea tu teoría, no importa lo inteligente que seas. Si no está de acuerdo con el experimento, está mal».

Richard P. Feynman

5.1. Descripción

Es el fin de este trabajo evaluar el desempeño de las redes propuestas como medio dinámico, así como la comparación del mismo con el obtenido por redes booleanas tradicionales. Trabajos previos enfocados en la búsqueda de Redes Booleanas Aleatorias como medio dinámico los podemos encontrar en (Snyder y col., 2012, 2013). Se desean reproducir algunos de esos hallazgos y observar el desempeño de la propuesta de este trabajo respecto a ellos.

5.2. Redes booleanas

Para la realización de los experimentos se han construido redes de forma aleatoria, conectando sus nodos siguiendo alguna de las topologías detalladas más adelante. Las redes han sido de dos clases: 1) tipo Kauffman y 2) Redes Booleanas de Pulso. Todas construidas bajo diversas combinaciones de parámetros relativos a las reglas de transición que los dominan, así como a la conexión propia de los nodos entre sí. Ello con el objetivo de barrer el espacio de configuraciones de las mismas.

Para las redes de Kauffman se ha explorado la probabilidad de que las entradas en las reglas booleanas que rigen a los nodos contengan un «1» como resultado. Dicha probabilidad se conoce como *sesgo*, y cómo se ha visto con anterioridad, tiene un impacto en la dinámica intrínseca de la red.

Para las Redes Booleanas de Pulso se han explorado diversos valores para el umbral de activación y el valor refractario de los nodos.

5.3. Retardos

Las RBP se han implementado con retardos aplicados a las conexiones entre nodos, tal cual son detallados en el apartado 3.4.1 y el apartado 3.4.2. Se emplea el modelo abstracto y los nodos historia como tales no existen.

En caso de aplicarse un retardo, este ha sido asignado al azar a cada conexión existente. El número de pasos de tiempo aplicados como retraso se ha elegido independientemente para cada conexión, siguiendo una distribución geométrica de acuerdo a un parámetro probabilidad que también ha sido explorado en los experimentos.

5.4. Varianza nodal

En el caso particular de las Redes Booleanas de Pulso se ha aplicado una perturbación a los valores activación y refracción, siguiendo una distribución normal con σ^2 especificada por el usuario, aplicando ciertas restricciones:

- El umbral de activación debe ser positivo o cero, de forma que siempre es empleado el valor absoluto de la variable aleatoria obtenida de la distribución normal. El valor es redondeado al entero más próximo.
- El valor de refracción debe ser menor o igual a cero, de forma que siempre es empleado el negativo del valor absoluto de la variable aleatoria empleada. El valor es redondeado al entero más próximo.

5.5. Entradas

Para la inyección de estímulos a la red se ha utilizado una sola cadena de símbolos y se ha elegido de forma aleatoria un subconjunto de nodos en la red,

forzando su estado al símbolo que posea el flujo de entrada a cada momento. Estos nodos hacen las veces de un sólo nodo independiente ω . Dicho conjunto de nodos no es tomado en cuenta en la construcción del vector de estado de la red. Los nodos efectivos de las piscinas son por tanto $N - |S|$, donde S es el conjunto de nodos elegidos como entrada.

5.6. Topologías

De acuerdo con Oosawa y Savageau (2002), existe un efecto mensurable en la dinámica de las RBA en función de la topología de conexión empleada. En particular, de acuerdo a la forma en que varía el grado de conectividad de los nodos a lo largo de la red. Es razonable que no sólo el grado, sino la topología misma cause un efecto no del todo conocido. A fin de explorar un posible impacto en el desempeño de la piscina en función del mecanismo empleado para determinar las conexiones entre los nodos, se han empleado las topologías de red presentadas a continuación.

5.6.1. «Erdős»

Son redes basadas en los grafos aleatorios propuestos en (Renyi & Erdős, 1959), en la cual se eligen al azar y de forma uniforme, una cantidad particular de todas las aristas posibles entre los nodos de la red. Ya que los grafos de la publicación original no son dirigidos, se ha optado por volver las conexiones simétricas. Es decir, si existe la arista a_{ij} existe también a_{ji} . A diferencia del tipo de grafo expuesto en el documento original, se permite que los nodos se conecten a sí mismos.

Sí M es la red, K es un parámetro para señalar el grado promedio deseado para los nodos y N el número total de nodos, el pseudocódigo para realizar las conexiones es el siguiente:

```

K = K * N
total = 0
Mientras total < K:
  x = Un nodo al azar en M
  y = Un nodo al azar en M
  Si y no está conectado a x:
    Conectar y a x
    Conectar x a y
    total += 2

```

5.6.2. «K-variable»

Se construyen de la misma forma que la de tipo Erdős, pero las aristas no son necesariamente bidireccionales.

Sí M es la red, K es un parámetro para señalar el grado promedio deseado para los nodos y N el número total de nodos, el pseudocódigo para realizar las conexiones es el siguiente:

```
K = K * N
total = 0
Mientras total < K:
  x = Un nodo al azar en M
  y = Un nodo al azar en M
  Si y no está conectado a x:
    Conectar y a x
    total += 1
```

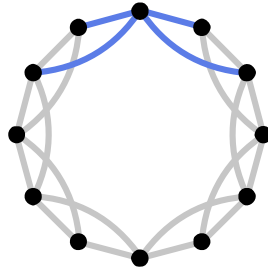
5.6.3. «Mundo pequeño»

Este tipo de red sigue un patrón de conexión definido en (Watts & Strogatz, 1998). Dichas redes disponen de dos parámetros de construcción: p y K . La red inicia conectando a cada nodo con sus vecinos laterales, tantos como diga K (se asume un arreglo circular de los mismos), para después modificar aleatoriamente con probabilidad p las conexiones resultantes. En la implementación desarrollada de este algoritmo, el grado promedio de los nodos siempre es par. El parámetro p representa el «desorden» de las conexiones. Para $p = 0$ la red es ordenada, para $p = 1$ se obtiene la máxima aleatoriedad. Con valores intermedios de p se tiende a reducir el diámetro del grafo, y en general, la distancia de todos los nodos entre sí.

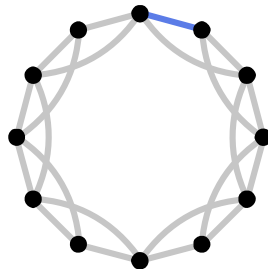
Dado que el gráfico obtenido es no-dirigido, se ha optado por volverlo dirigido haciendo las conexiones simétricas. Es decir, si existe la arista a_{ij} existe también a_{ji} .

Los pasos para la construcción de la red son los siguientes:

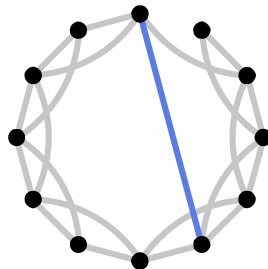
1. Se inicia con un anillo de N vértices y se conecta a todos con sus K vecinos más cercanos.



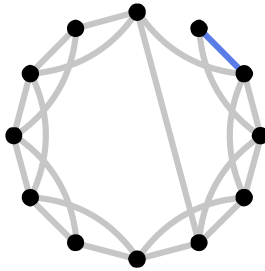
2. Se toma un nodo y su conexión más cercana en el sentido de las manecillas del reloj.



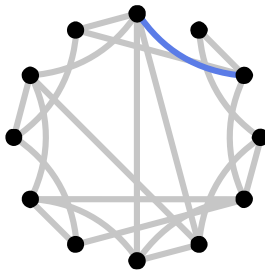
3. Con probabilidad p , se reconecta la arista a otro nodo de la red elegido al azar con probabilidad uniforme. Se evitan las conexiones de un nodo hacia sí mismo.



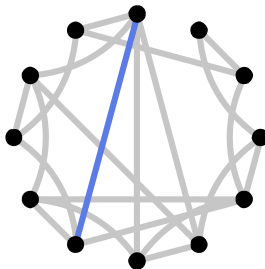
4. Se repite el proceso para cada nodo de la red, hasta dar la vuelta completa.



5. Una vez realizada dicha vuelta, se procede a elegir la siguiente arista más cercana del primer nodo.



6. Como en el caso anterior, se reasigna con una probabilidad p a otro nodo, evitando las conexiones de un nodo hacia sí mismo y las repetidas. Así hasta completar la vuelta, y terminar de revisar todas las conexiones más cercanas en el sentido de las manecillas del reloj de todos los nodos.



5.6.4. «Conexión por distancia»

Para su construcción los nodos de la red se sitúan en una rejilla tridimensional imaginaria. La probabilidad de conexión con algún nodo vecino está en función de la distancia entre ellos de acuerdo a la siguiente expresión:

$$P(c)_{ij} = a e^{-\frac{d(i,j)}{\lambda}} \quad (5.1)$$

Donde:

- $P(c)_{ij}$ es la probabilidad de conexión del nodo i hacia el nodo j .
- $d(i, j)$ es la distancia euclídea, en unidades de rejilla, entre los nodos i y j .
- a es un parámetro de escalamiento.
- λ es una constante de distancia.

5.7. Unidad de lectura

La unidad de lectura conectada a la piscina se ha construido utilizando unidades de umbral. Entidades con una función Heaviside de activación. A un arreglo de dichas unidades la llamaremos aquí *perceptrón paralelo*. La salida de un componente del arreglo es dominada por la siguiente función¹⁰:

$$r(z) = \begin{cases} 1 & : \alpha \cdot z \geq 0 \\ -1 & : \text{en otro caso} \end{cases} \quad (5.2)$$

Donde n es el número de nodos efectivos de la piscina, $\alpha \in \mathbb{R}^n$ es el vector de pesos del perceptrón paralelo, $z \in \{-1, 1\}^n$ es un vector de estado que proviene de la piscina, con un componente por cada nodo mapeado al intervalo $[-1, 1]$, $\alpha \cdot z$ denota el producto punto habitual entre vectores. Las unidades de umbral no se conectan entre sí.

La salida del perceptrón paralelo será definida como:

$$R(z) = \begin{cases} 1 & : \sum_1^d f_i(z) \geq 0 \\ -1 & : \text{En cualquier otro caso} \end{cases} \quad (5.3)$$

¹⁰Por cuestiones de simplicidad, se asume que el umbral de las unidades es 0.

Donde d es el número de unidades de umbral utilizadas. A unidades definidas de esta manera, con salidas binarias $\{-1, 1\}$, se les puede considerar como un grupo de votantes, de tal forma que el signo de la suma de los valores de salida de cada uno de ellos se considera la salida binaria del sistema completo (Auer, Burgsteiner & Maass, 2008).

En nuestro caso particular, cada unidad de umbral individual se conecta a todos los nodos de la piscina. El propósito de utilizar una capa de lectura tan elemental constituida por unidades tan simples, radica en el deseo de mostrar la capacidad de pre-procesamiento de la información que realiza la piscina, previa a su llegada a dicha capa. Si la piscina hace las veces de una función *kernel* temporal, debe contribuir al mapeo de las entradas a un nuevo espacio donde resulte más sencillo separarlas lineal o cuasi-linealmente.

5.8. Tareas

A fin de probar la capacidad de procesamiento de las piscinas, se les ha dado la tarea de realizar labores de clasificación, siguiendo los siguientes esquemas:

5.8.1. Densidad temporal

Consiste en determinar si el número de símbolos «1» en una ventana temporal, situada en el pasado de la entrada inyectada en la piscina, es mayor o igual a la cantidad de símbolos «0».

Dada una entrada u , con longitud $|u| = \mathcal{T}$, un retardo τ , y una ventana $n \geq 1$, la densidad se define como:

$$\text{DNS}_n(t) = \begin{cases} u(t - \tau) & : n = 1 \\ \bigoplus_{i=1}^{n-1} u(t - \tau - i) & : \text{en otro caso} \end{cases} \quad (5.4)$$

cuando $\tau + n \leq t \leq \mathcal{T} - \tau - n$.

5.8.2. Paridad temporal

Esta es una versión de computación en línea para la tarea de paridad presentada en (Minsky & Papert, 1969). Consiste en determinar si el número de

símbolos «1» en una ventana temporal, situada en el pasado de la entrada inyectada en la piscina, es par o impar.

Dada una entrada u , con longitud $|u| = \mathcal{T}$, un retardo τ , y una ventana $n \geq 1$, la paridad se define como:

$$\text{PAR}_n(t) = \begin{cases} 1 & : 2 \sum_{i=0}^{n-1} u(t - \tau - i) \geq n \\ 0 & : \text{en otro caso} \end{cases} \quad (5.5)$$

cuando $\tau + n \leq t \leq \mathcal{T} - \tau - n$.

Esta tarea en particular es una generalización de la prueba de clasificación para la función XOR, si es contemplada como una paridad, y es particularmente ardua y difícil. Todas las ventanas temporales cambian de clase al alterar sólo uno de sus bits.

5.8.3. Paridad de cambio

Las tareas anteriores son casos extremos. *Densidad* es fundamentalmente simple, mientras que *Paridad* es bastante ardua. A fin de evaluar el sistema con una transformación de dificultad intermedia, se propone aquí la *Paridad de cambio*, que de forma similar a *Paridad* consta de dos clases: «Par» e «Impar». Pero, a diferencia de ésta última, el conteo se realiza sobre el número de veces que el flujo de datos alterna su símbolo. Es decir, cada vez que la cadena pasa de «0» a «1» o viceversa, se agrega una unidad a un contador, y la clase estará en función de la paridad de dicho contador.

Dada una entrada u , con longitud $|u| = \mathcal{T}$, un retardo τ , y una ventana $n \geq 1$, la paridad de cambio se define como:

$$\text{PSH}_n(t) = \begin{cases} 1 & : \sum_{i=0}^{n-2} |x_{t-\tau-i} - x_{t-1-\tau-i}| \equiv 0 \pmod{2} \\ 0 & : \text{en otro caso} \end{cases} \quad (5.6)$$

Donde $|x_{t-\tau-i} - x_{t-1-\tau-i}|$ es la diferencia absoluta entre los símbolos $x_{t-\tau-i}$ y $x_{t-1-\tau-i}$ cumpliéndose que $\tau + n \leq t \leq \mathcal{T} - \tau - n$.

5.9. Entrenamiento y evaluación

Para llevar a cabo las tareas especificadas, los sistemas seleccionados fueron alimentados por un conjunto de entradas generadas aleatoriamente. A fin de agregar variabilidad a las mismas, para cada entrada generada, se elije al azar la probabilidad de aparición del símbolo «1» en el flujo de datos que representa. Se elije un rango continuo en $[0, 1]$ dentro del cual es elegida al azar dicha probabilidad.

Los vectores de entrenamiento y prueba que se utilizaran para ajustar los pesos de la unidad de salida, son tomados de todos los instantes t de dichos flujos de entrada para los cuales este definida una ventana temporal como las declaradas anteriormente. Dichos estados son etiquetados de acuerdo a las tareas a evaluar.

Antes de la inyección de cada entrada todas las redes booleanas son inicializadas al estado homogéneo 0. Esta política se emplea, tanto en la caracterización de las redes, como en las tareas de clasificación.

5.9.1. Algoritmo genético ecléctico

Como mecanismo de entrenamiento se ha elegido utilizar una variación del algoritmo genético conocido como «ecléctico», reportado en (Kuri & Villegas, 1998).

El algoritmo original utiliza selección determinista, cruzamiento anular, mutación uniforme y elitismo total, así como adaptación automática en sus parámetros de ejecución. En la variante implementada aquí, la adaptación automática de los parámetros ha sido suprimida y permanecen fijos a lo largo de toda la ejecución del mismo. También se ha agregado uno que permite establecer el porcentaje de la población que se elije como élite.

Para codificar el genoma se han concatenado cadenas binarias que representan el valor de los pesos de cada conexión para cada una de las unidades de umbral existente en el perceptrón paralelo. El valor representado en cada una de estas subcadenas es un número decimal con punto fijo, utilizando un bit extra para el signo.

Si K es el número de generaciones, N el número de individuos por generación, e elite es el porcentaje de individuos de la generación que se tomarán como élite, L es la longitud del genoma, p_c la probabilidad de cruzamiento, p_m la

probabilidad de mutación; el pseudocódigo de la implementación utilizada es la siguiente. Se asume que los índices inician en cero:

```

n_elite = techo(elite * N)
Se crea una población P con N individuos
Se evalúa la población P
Se ordena la población P de mejor a peor

Repetir K veces:
  Copiar los primeros n_elite elementos de la población P en E
  c = variable geométrica(pc) - 1 [Primer individuo a cruzar]
  Mientras c < N / 2:
    [Se cruzan los individuos c y N - c]:
      Se genera un número aleatorio  $0 \leq r < L / 2$ 
      Se intercambia el semi-anillo iniciando en la posición r de
      los individuos P(c) y P(N - c)
      c = c + variable geométrica(pc)
    [Mutación]:
      Para i de 0 hasta N:
        b = variable geométrica(pm) - 1 [Primer bit a mutar]
        Mientras b < L:
          Mutar bit P(i)[b]
          b = b + variable geométrica(pm)
      Evaluar la nueva población P
      Concatenar E a P
      Ordenar P de mejor a peor
      Conservar en P los primeros N individuos

Regresar el individuo P(0)
Terminar

```

5.9.2. Función de evaluación

Para evaluar el desempeño de un individuo de la población, que es un perceptrón paralelo, se construye un vector de salidas, donde cada elemento del mismo es la respuesta del perceptrón mismo a un vector de estado utilizado para el entrenamiento. El vector de salidas tendrá tantos componentes como vectores de estado se utilicen para el entrenamiento.

Este vector es comparado con un vector de salidas esperadas. Se suman las diferencias absolutas entre los componentes del vector de salidas esperadas y el calculado. Este valor es el que se intenta minimizar.

Sea $R(z_i)$ la evaluación del vector de estado z_i por el perceptrón R de acuerdo a 5.3. Sea $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ el vector de salidas esperadas. La evaluación de un perceptrón será:

$$\text{Eval}(R) = \sum_{i=1}^n |R(z_i) - \alpha_i| \quad (5.7)$$

5.10. Software empleado

Para la realización de las pruebas se desarrollaron varias aplicaciones destinadas a automatizar el proceso. Estas incluyen una biblioteca para la creación y ejecución de Redes Booleanas Aleatorias así como Redes Booleanas de Pulso, una biblioteca para la creación y evaluación de los perceptrones paralelos, una biblioteca para la definición de problemas y un algoritmo genético destinado a su resolución destinado al entrenamiento de la unidad de salida, interfaces gráficas para la automatización de las pruebas, las tareas, el entrenamiento y la visualización de los resultados.

El *software* fue creado utilizando el lenguaje de programación **Python** (van Rossum & Drake Jr, s.f.), auxiliándose de las siguientes bibliotecas: **NumPy** (para el manejo eficiente de arreglos, vectores y matrices) (van der Walt, Colbert & Varoquaux, 2011), **SciPy** (funciones que extienden y hacen uso de las estructuras de datos brindadas por NumPy) (Jones, Oliphant, Peterson y col., 2001–), **Matplotlib** (biblioteca destinada a la creación de gráficas y visualización de datos) (Hunter, 2007), **PyQt** (Integración para Python de las bibliotecas Qt, para la creación de interfaces gráficas multiplataforma).

Capítulo 6

Resultados

What most experimenters take for granted before they begin their experiments is infinitely more interesting than any results to which their experiments lead.

«Lo que buena parte de los experimentadores dan por sentado, antes de que inicien sus experimentos, es infinitamente más interesante que cualquier resultado al que conducen esos mismos experimentos.»

Norbert Wiener

Siendo el objetivo indagar en las características que subyacen al buen desempeño de los medios dinámicos bajo estudio, y no simplemente encontrar una buena red para el cómputo por reservorio (de existir alguna), se inició con una exploración general del espacio de configuraciones de las redes, a fin de localizar los «puntos calientes» y los parámetros de construcción dignos de una exploración más profunda.

Se crearon redes clásicas y de pulso, explorando en rangos razonables el espacio de configuraciones de las mismas bajo diferentes topologías. A fin de evitar la explosión combinatoria que ello implica, estas primeras indagaciones se hicieron a baja resolución con el objetivo de encontrar los rasgos generales y puntos de interés que serían evaluados a mayor detalle más adelante.

Las exploraciones, gráficas y mapas de calor presentados a continuación, salvo que se indique lo contrario, fueron creados siguiendo las directrices listadas a continuación:

- Para obtener las medidas de cada combinación de parámetros de construcción, se obtuvo el promedio de los valores obtenidos para cada medida de 20 redes generadas al azar.
- Cada una fue alimentada con 120 entradas con una longitud de $\mathcal{T} = 50$.

- Cada símbolo es presentado un paso de tiempo a la vez. Dichas entradas se dividen en 4 grupos de 40 integrantes cada uno.
 - Grupo A.** Entradas generadas al azar. La probabilidad de que un símbolo sea estímulo es de $p = 0.5$. Este grupo es utilizado para calcular todas las medidas, salvo que se indique lo contrario.
 - Grupo B.** Todas las entradas son una versión perturbada diferente de una misma entrada del grupo A elegida al azar. La perturbación consiste en alterar cada símbolo con una probabilidad $p = 0.04$. Este grupo es utilizado para calcular el *rango de generalización (GR)*.
 - Grupo C.** Son generadas de la misma forma que las entradas del grupo A.
 - Grupo D.** Se toman las entradas del grupo C, pero los últimos τ símbolos de la i -ésima entrada son sustituidos por los últimos τ símbolos de la i -ésima entrada del grupo A. La combinación de los grupos A, C y D es utilizada para calcular la *evanescencia* tal cual se describe en el apartado 4.7.

Dada la extrema cantidad de datos generados, buena parte de ellos sin el menor interés, se presentan aquí los resultados más relevantes obtenidos en las indagatorias.

6.1. Puntos de interés

6.1.1. Piscinas RBA

Contrario a lo esperado, para las RBA la topología de la red resultó poco relevante en lo que concierne a sus capacidades de *separación*, *evanescencia*, y cualquier otra medida elegida para caracterizarlas. Al menos en los rangos explorados. Los únicos factores con importancia manifiesta han sido el grado promedio de los nodos $\langle K \rangle$ y el sesgo de las reglas. La figura 6.1 muestra los puntos «calientes» para la *separación* y *evanescencia* comunes a todas las topologías para el plano $\langle K \rangle$ vs. *sesgo*. Por «sesgo» debe entenderse una probabilidad $p \neq 0.5$ para que una regla cambie el estado de un nodo a «1».

En ausencia de sesgo ($p = 0.5$), los puntos «calientes» corresponden a valores de K asociados con la fase crítica de las RBA acordes a su tamaño. La *evanescencia* es máxima para $p = 0.5$. Por el contrario la *separación*, si bien presenta altos

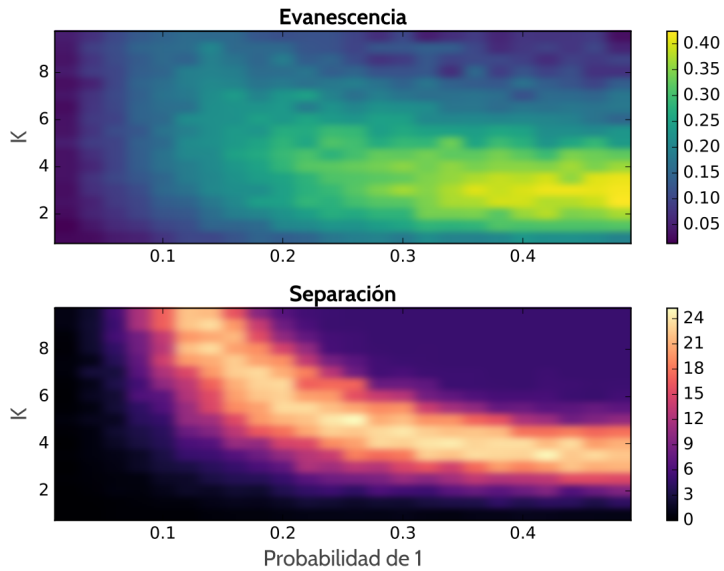


Figura 6.1: *Separación* y *evanescencia* en el plano $\langle K \rangle$ vs. *sesgo* para una RBA con $N = 160$ y 80 nodos efectivos en la piscina. La mayor *evanescencia* y *separación* se encuentran en valores de $\langle K \rangle$ asociados a su fase crítica con $p = 0.5$. Para redes de este tamaño el valor es $\langle K \rangle \approx 3.5$. El mapa es simétrico a $p = 0.5$. Sólo se presenta aquí el rango $[0, 0.5]$.

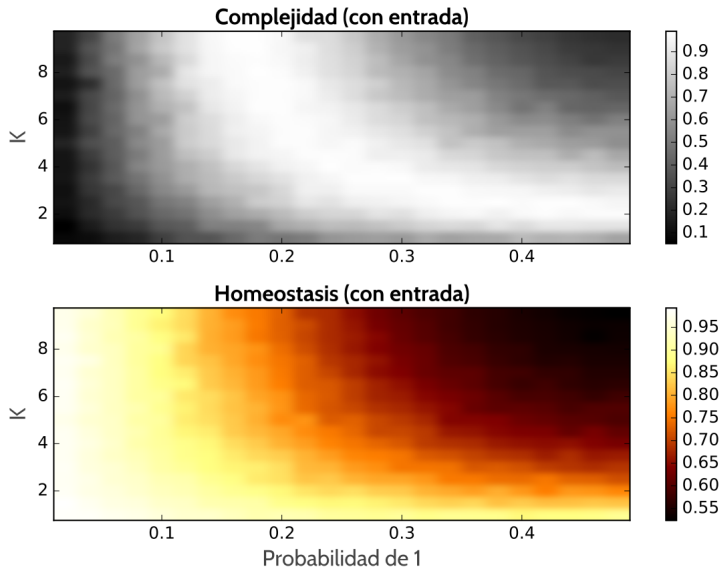


Figura 6.2: *Homeostasis y complejidad* en el plano λ vs. C para las mismas RBA presentadas en la figura 6.1.

valores también en dicha zona, se extiende a otras correspondientes a una combinación de grandes sesgos para las reglas y altos valores de $\langle K \rangle$. Dado que este tipo de redes no hace distinción alguna para los símbolos que constituyen su vector de estado, el comportamiento en el mapa es simétrico respecto al valor $p = 0.5$.

En estas redes hay claras diferencias entre las zonas que se encuentran a uno y otro lado de la frontera establecida por una alta *separación*. La zona inferior alberga redes completamente estables, que alteran poco su comportamiento ante cualquier estímulo recibido. Por el contrario, la zona superior alberga redes en estado caótico, muy sensibles a la perturbación. Esto es fácil de apreciar observando su *homeostasis*, como es posible hacerlo en la figura 6.2. Los mejores índices de separación también se encuentran asociados a altos valores para la *complejidad*, tal cual se ha definido aquí.

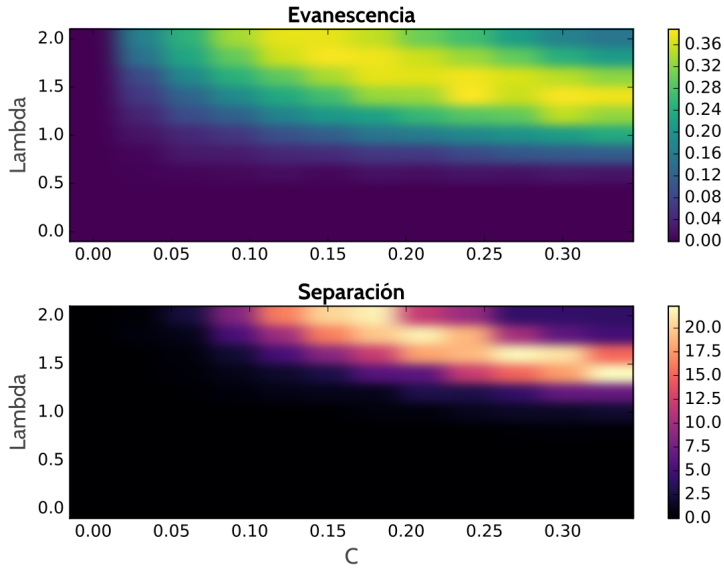


Figura 6.3: *Separación* y *evanescencia* en el plano λ vs. C para una RBA con $N = 160$, $p = 0.5$ y 80 nodos efectivos en la piscina. La rejilla tridimensional que contiene a los nodos tiene medidas $4 \times 5 \times 8$. La mayor *evanescencia* y *separación* se encuentran para combinaciones de λ y C que proporcionan un valor de $\langle K \rangle$ idéntico al que presenta las mismas características en otras topologías para este tamaño de red. Resulta irrelevante la dispersión de las conexiones a lo largo de la rejilla.

Los mismos rasgos aplican para las redes con topología por conexión a distancia. Aunque no existe como tal un parámetro de construcción que nos permita indicar un valor explícito para $\langle K \rangle$, su comportamiento se muestra congruente con el resto de las topologías, teniendo sus mejores índices de *separación* y *evanescencia* para valores calculados de $\langle K \rangle$ y p similares al resto. Un ejemplo de esto se puede encontrar en la figura 6.3.

El valor de $\langle K \rangle$ que maximiza la medida de *evanescencia* es siempre el mismo que determina la fase crítica de la red para su tamaño correspondiente.

6.1.2. Piscinas RBP

En el caso de las RBP, además de $\langle K \rangle$ también han sido importantes el porcentaje de nodos inhibidores, así como los valores de activación, refracción y retardos asociados a las mismas.

Al igual que en las RBA, la topología empleada resultó irrelevante para las medidas empleadas. La excepción son las topologías de mundo pequeño y de conexión por distancia, que tienden a «congelar» la actividad de ciertos nodos, tendiendo a fijar su estado y mantenerlos activos todo el tiempo. Dicho rasgo se presenta, para las redes de mundo pequeño, cuando el parámetro p es pequeño. Para las redes por conexión a distancia, cuando el valor de λ es grande en proporción al coeficiente. En ambos casos, parámetros con tales características están asociados a conexiones entre los nodos muy locales y poco dispersas. No se encontró evidencia de que una dispersión intermedia de las conexiones favorezca alguna medida o sus capacidades de cómputo. Pero el tema permanece abierto para redes con características y tamaños diferentes a las estudiadas aquí.

Tal cual se esperaba, las calificaciones más altas de *separación* y *evanescencia* se corresponden con fases críticas. Zonas en el espacio de configuraciones del sistema equilibradas en medio de otras donde la configuración de las mismas hace inviable cualquier uso de las redes como reservorio, bien porque son incapaces de separar adecuadamente las entradas o carecen de una capacidad de memoria evanescente adecuada.

Resultó notable que las medidas de *separación* y *evanescencia*, tal y como son cuantificadas aquí, tuvieran una relación tan estrecha sin importar el tipo de red del cual se tratase. Tanto las RBA como las RBP presentan altos índices de *evanescencia* en configuraciones que también presentan altos valores para la *separación*. Pero lo contrario no se cumple siempre: altos valores de *separación* se manifiestan en configuraciones que no expresan la capacidad para olvidar los estímulos pasados, que es lo cuantificado por la medida de *evanescencia*.

Esta peculiar situación, redes que no se encuentran en fase caótica, que a pesar de ello son incapaces de «olvidar» los estímulos del pasado, tiene una causa desconocida. En el caso particular de las RBP, una red idealizada siempre olvida, pero las redes reales pueden tener una multiplicidad de cuencas de atracción en las diferentes normas que las constituyen, que frustrarían tal pretensión en ciertas situaciones. Existe la posibilidad de que la dinámica del sistema se haya

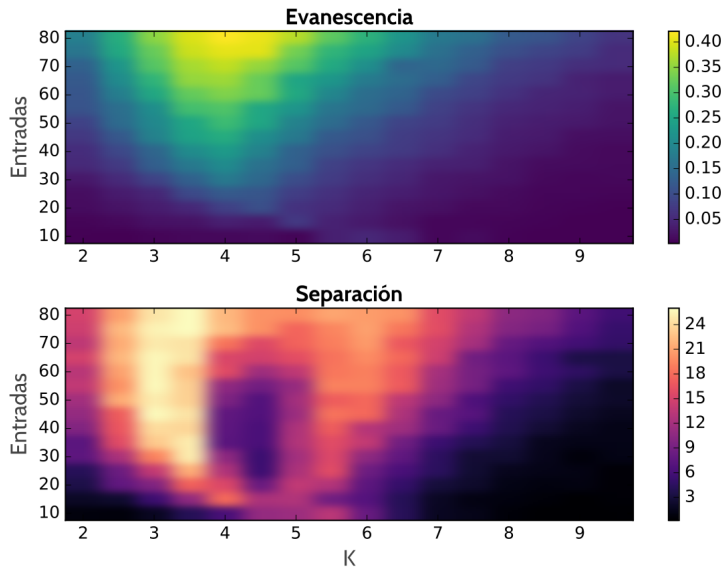


Figura 6.4: El plano $\langle K \rangle$ vs. *nodos de entrada* para piscinas RBP con las siguientes características. Topología: K-variable, N : 160, nodos inhibidores: 0.0, activación media: 1, refracción media: 1, varianza nodal: 1.0, parámetro de retardo: 0.65. El rasgo mostrado en forma de pinza es común a todas las topologías para un amplio rango de valores de configuración.

«atascado» en alguna clase de estructura no estudiada, fruto de la interacción entre la dinámica de las normas y los estímulos, que le impida «regresar» a la cuenca que posee al estado estable.

Una explicación se puede encontrar en lo siguiente: las cuencas de atracción de una norma pueden servir como «puente» entre las cuencas de atracción de otras normas. Existe la posibilidad de que no exista siempre un camino de regreso. Es decir, que una vez el sistema ha escapado de una cuenca, le sea imposible regresar a ella.

Si el sistema se ve «atascado» en su comportamiento de esta forma, siendo imposible para él alcanzar de nuevo la cuenca que contiene al estado estable en la norma 0, entonces podemos decir que la red «no olvida». El simple hecho de que el sistema se encuentre en esa situación da testimonio de que existió, en algún momento arbitrariamente lejano en el pasado, un estímulo que lo condujo a él. Esto implica la presencia de una «memoria eterna», pues resulta imposible determinar el instante en que dicho estímulo tuvo lugar. Algo completamente opuesto a la memoria evanescente necesaria para el adecuado cómputo de las entradas.

Las RBA tradicionales, si bien muestran índices de separación comparables a los que poseen las RBP, presentan niveles de evanescencia que suelen ser bastante más bajos. Es decir, tienen mayores dificultades para manifestar una adecuada memoria evanescente. Esto tiene sentido a la luz de la explicación anterior. Las RBA tradicionales por lo general poseen múltiples cuencas, que se encuentran en función de su tamaño, $\langle K \rangle$ y el sesgo de sus reglas (Wuensche y col., 1998).

A diferencia de las RBA, el valor óptimo de $\langle K \rangle$ para las RBP se encuentra en función del resto de sus parámetros. Es decir: la activación y refracción nodal, el porcentaje de nodos inhibidores en la red, etc. Redes con muy diferentes grados entrantes para sus nodos pueden presentar rendimientos y medidas similares.

6.2. Resolución de tareas

La evaluación definitiva para las redes booleanas, a fin de probar su desempeño, ha sido el someterlas a tareas de clasificación simples que revelen el verdadero potencial de su capacidad computacional.

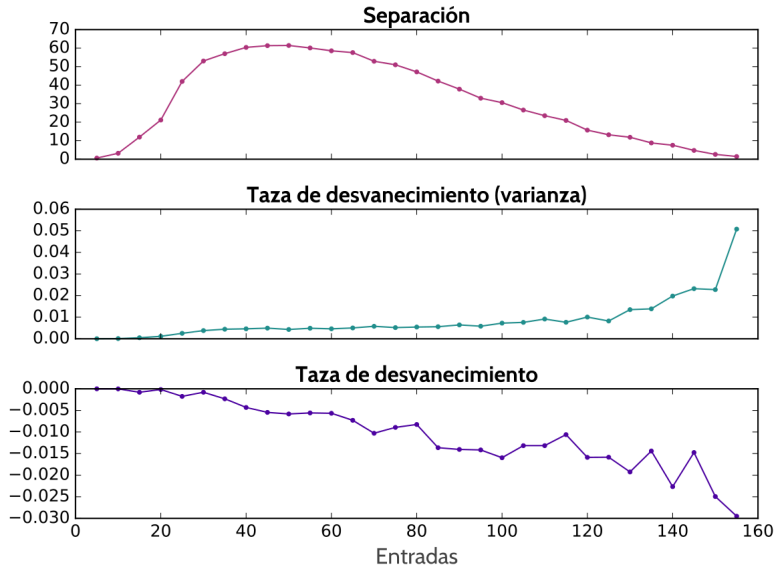


Figura 6.5: Cuando la proporción de nodos de entrada supera la mitad de nodos totales, irremediablemente los indicadores sufren una merma continua. Tal es el caso de *separación*, *calidad de kernel (KQ)* (no presentada aquí), la *taza de desvanecimiento* y su *varianza*. El caso presentado aquí es una RBP con topología K-variable con los siguientes parámetros. N : 160, refracción media: 1, activación media: 1, varianza nodal: 1.0, $\langle K \rangle$: 3.5, porcentaje de nodos inhibidores: 0.0, parámetro de retardo: .65.

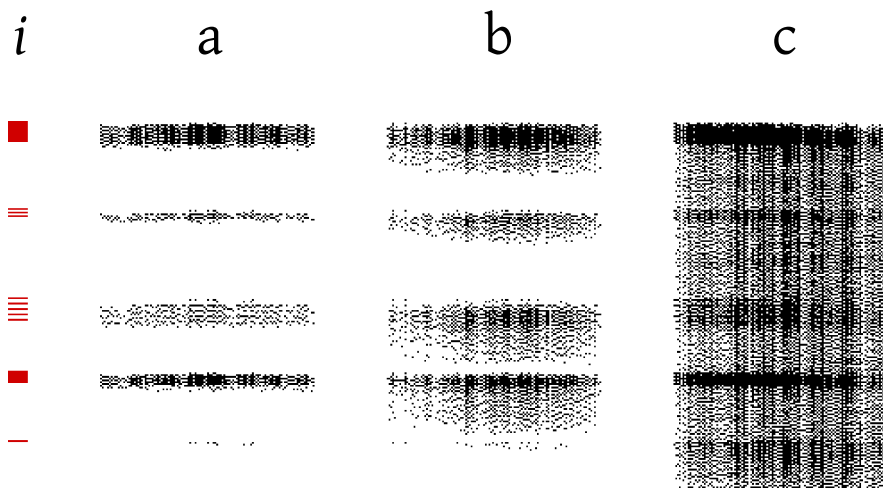


Figura 6.6: La respuesta de tres diferentes RBP a una misma entrada i . El tiempo fluye hacia abajo. La red a presenta una extrema estabilidad a la perturbación, lo que resulta en tiempos transientes extremadamente cortos y poca influencia de los estímulos en el futuro inmediato. La red b , siendo estable, presenta mayores tiempos transientes, lo que favorece su memoria. La red c es extremadamente sensible a los estímulos, entrando fácilmente a un estado caótico que vuelve imposible para ella regresar a su estado de reposo. Si no se usan retardos, el paso de las redes tipo a a las redes tipo c es una fase crítica de su espacio de configuraciones.

Para la realización de las mismas se ha utilizado exclusivamente una capa de lectura lineal para cada tarea, consistentes en un único perceptrón paralelo tal cual se describe en el apartado 5.7, siendo entrenados utilizando la técnica evolutiva descrita en el apartado 5.9. El propósito de usar unidades de lectura lineales consiste en revelar la utilidad del reservorio como una función *kernel* temporal, que permita la clasificación y cálculo de información que originalmente no es de naturaleza lineal.

6.2.1. Medidas vs. desempeño

Se puede estimar fácilmente el potencial computacional de las RBA como reservorios. La medida de *separación* es un gran indicador de dicho potencial. Aunque esto puede resultar engañoso, como veremos más adelante con las RBP. Todos los valores críticos para las RBA caen en los mismos puntos. Así que el hecho de que la medida de *separación* sea un indicador puede ser una casualidad.

Contrario a lo esperado, la *evanescencia* es prácticamente irrelevante para este tipo de redes. La *complejidad* de la piscina, por otro lado, es también un buen indicador.

La potencia de cómputo para las RBA es una combinación de la medida de *separación* y el sesgo en las reglas que conforman la red.

Sesgos muy altos para las RBA favorecen la estabilidad del sistema, creando una tendencia en la red a permanecer en un estado compuesto enteramente de ceros o unos. Algo análogo al estado estable de las RBP. Vuelve también a las redes más tolerantes a valores altos de $\langle K \rangle$ antes de entrar en actividad caótica.

Para mostrar la relación de las medidas mencionadas con el desempeño de las piscinas implementadas como RBA, se eligieron de forma más o menos arbitraria 6 zonas en el plano de configuraciones de las mismas, presentando su desempeño en las tareas de clasificación detalladas en capítulos precedentes.

Para su cálculo se utilizó el siguiente procedimiento:

- Se construyeron 4 redes al azar para cada zona, utilizando los mismos parámetros.
- Se generaron 100 entradas de tamaño $\mathcal{T} = 200$ al azar. La probabilidad de que un símbolo resultara ser un estímulo se elegía al azar para cada entrada de manera uniforme en el intervalo $[0.4, 0.6]$.

Zona	Precisión por tarea		
	Densidad	Paridad de cambio	Paridad
1	0.9791	0.7855	0.6894
2	0.9719	0.8661	0.8258
3	0.7098	0.5287	0.5092
4	0.8958	0.7751	0.7563
5	0.7185	0.5791	0.5852
6	0.9496	0.8431	0.8476

Tabla 6.1: Resultados de clasificación para piscinas RBA de acuerdo a su origen en el plano $\langle K \rangle$ vs. *sesgo*. La ubicación de cada zona puede consultarse en la figura 6.7. Como se esperaba, la precisión de la clasificación decrece de acuerdo a la dificultad de cada tarea. El valor obtenido es el promedio de cuatro validaciones cruzadas de 10 hojas cada una. La precisión de cada validación es el promedio de aciertos para cada clase.

- La red se reinició a su estado de reposo al comenzar la inyección de cada entrada.
- Se generaron 19,300 vectores de estado por red, los cuales se etiquetaron de acuerdo a lo especificado por las tareas de *Paridad*, *Densidad* y *Paridad de cambio*. Las tareas consistieron en clasificar una ventana temporal de la entrada con tamaño $n = 5$ desplazada $\tau = 2$ pasos en el pasado.
- La calificación de cada red se obtuvo promediando la precisión en la clasificación para cada clase, la cual fue obtenida a través de una validación cruzada de 10 hojas.
- La calificación de la zona es el promedio de las cuatro redes que le corresponden.

Un mapa de las zonas elegidas se puede ver en la figura 6.7. Los resultados en el cuadro 6.1.

Contemplando la gráfica y los resultados asociados a ella, es fácil descubrir que las zonas con mejor desempeño están fuertemente ligadas a altos valores para la medida de *separación*. Es el caso de las zonas 1, 2, y 6, que curiosamente son, cada una, la mejor para cada una de las tareas.

La zona 3, caracterizada por redes estables, ha resultado ser la peor evaluada. A pesar de ello, todas las zonas son relativamente buenas en la tarea más simple

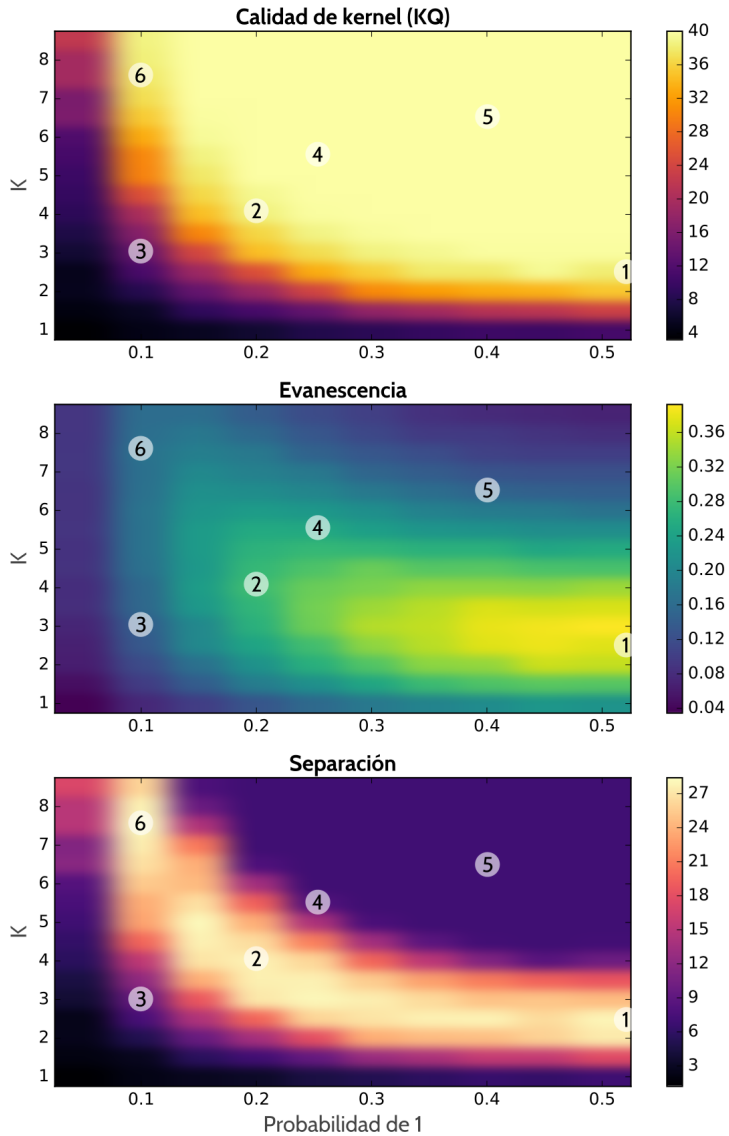


Figura 6.7: Localización de las RBA evaluadas en el plano $\langle K \rangle$ vs. sesgo (ver texto).

que es *Densidad*. Aunque su desempeño decrece estrepitosamente para el resto de tareas salvo, quizá, la zona 2, que mostró un alto desempeño para todas.

El desempeño se revela completamente desligado de la medida de *evanescencia*, al menos para este tipo de redes.

Contrario a lo esperado, en el caso de las RBP no existe una medida o combinación obvia de ellas que represente por completo el potencial de una red como reservorio adecuado para el cómputo.

Altos índices en la medida de *separación*, contrario a lo esperado y a lo que sucede en las RBA, ha resultado completamente irrelevante para estimar el poder computacional de las redes, al menos cuando realizan las tareas de clasificación evaluadas. Aunque se consideran lo suficientemente generales para resultar significativos los resultados. Por otro lado, el desempeño de la piscina parece estar íntimamente ligado a las medidas de *evanescencia* y *calidad de kernel* (KQ), circunstancia que no se presenta en las RBA. El desempeño de las RBP siempre es superior cuanto mayor es el valor combinado de dichas medidas.

Para que sea posible la existencia de un gran valor de *evanescencia* la red debe ser capaz de hacer converger su trayectoria para entradas idénticas que son inyectadas, cada una, partiendo de un estado inicial de la red muy diferente. Estado que ha sido generado previamente por la inyección de entradas no relacionadas entre sí. Esta propiedad de la red habla de su capacidad para ser dominada por los estímulos más recientes en detrimento de los más antiguos. Como dicha medida es calculada por la diferencia entre ambas distancias (las iniciales y las finales), no resulta extraño que esté fuertemente relacionada con la *calidad de kernel* (KQ), la cual mide la dimensionalidad del espacio al que son proyectados los estados de la red al terminar de ser inyectadas n entradas no relacionadas entre sí.

Si un grupo de vectores están confinados en un espacio de pocas dimensiones, tenderán a ser similares entre sí, y su distancia de hamming será menor. Si es el caso, no será posible partir de una distancia inicial muy grande al momento de calcular la *evanescencia*, circunstancia que la condena a ser pequeña.

El caso contrario no se cumple. Es posible que existan grandes valores para la *calidad de kernel* (KQ) que no se vean acompañados de altos índices de *evanescencia*. Que los vectores de estado finales para diferentes entradas se proyecten a un espacio de alta dimensionalidad no tiene, en principio, relación alguna con la capacidad que la misma red tenga por hacer converger su trayectoria si las entradas se volvieran idénticas a partir de un momento dado. La ausencia de

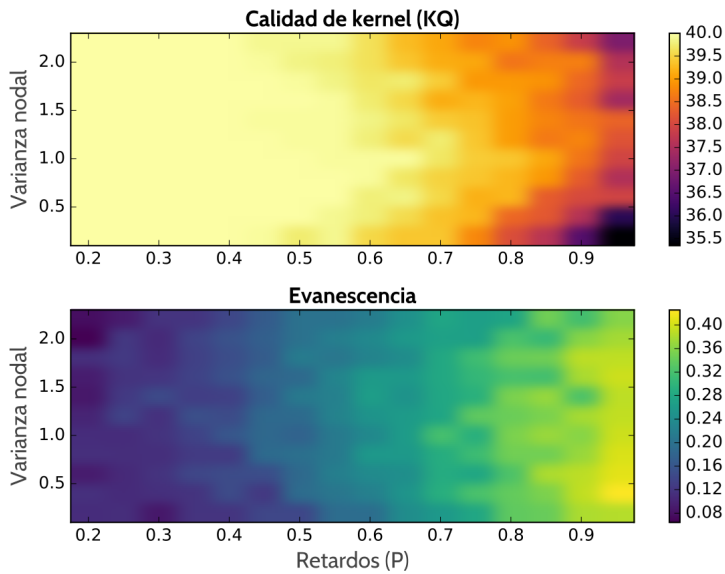


Figura 6.8: El plano *varianza nodal* vs. *retardos* de una RBP. A retrasos mayores (un parámetro de retardo cada vez más pequeño) la medida de *evanescencia* disminuye. Caso contrario de la *calidad de kernel (KQ)*. Este comportamiento es universal para todas las redes estudiadas. Las representadas aquí tienen como parámetros de construcción los que se listan enseguida. Topología: K-variable. N : 160, entradas: 80, $\langle K \rangle$: 5.5, activación media: 2, refracción media: 1, porcentaje de nodos inhibidores: 0.0.

esa habilidad para converger es propia de los sistemas caóticos. Lo opuesto, de los ordenados. En cierto sentido, la *evanescencia* y la *calidad de kernel (KQ)*, son capacidades antagonistas.

¿Por qué no usar simplemente la *evanescencia* para evaluar el desempeño potencial de la red? Porque existen casos para los cuales un descenso de la medida de *evanescencia* se corresponde con una mejora en el desempeño de la red. Y dicha mejora está acompañada de un aumento en la *calidad de kernel (KQ)*. También es verdad que altos valores para la *calidad de kernel (KQ)* no garantizan un buen desempeño si no están acompañados de valores altos para la *evanescencia*.

Es el caso cuando se varía el parámetro de retraso al momento de construir la red (Ver figura 6.8). Para valores pequeños, los nodos tienden a realizar conexiones cada vez más lejanas en el pasado. Esto mejora la memoria de la red, pero su medida de *evanescencia* descende. Tiene sentido. Traer información del pasado hace su pérdida más lenta. Pero tal descenso en la capacidad de olvidar es aparente. Si una configuración de red posee tal cualidad la mantiene si se agregan retrasos, simplemente la manifiesta a un ritmo más lento.

La ineficacia de la medida de *separación* como indicador de la capacidad para clasificar del reservorio tiene sentido si consideramos la forma en que es calculada. La *separación* es la diferencia entre la *calidad de kernel* (KQ) y el *rango de generalización* (GR). Éste último hace referencia a la dimensionalidad del espacio al que son proyectados los estados de la red al terminar de ser inyectadas n entradas similares entre sí. Pero «similares entre sí» es algo que está en función de la tarea a realizar. Se refiere a las entradas que es deseable sean clasificadas en la misma categoría. Aquí las entradas similares se han construido a partir de una sola, induciendo pequeñas perturbaciones en la misma, es decir, alterando al azar una muy pequeña cantidad de los bits que las constituyen.

Cuantificar de esta manera la *separación* atenta directamente contra la naturaleza de tareas tales como *Paridad*, donde modificar un sólo bit en la entrada la hace cambiar de clase (ver cuadro 6.3). Aquí entradas similares (usando como medida su distancia de hamming) se deben proyectar a un espacio de alta dimensionalidad. De lo contrario será imposible separarlas con un clasificador lineal.

A fin de revelar la ineficacia de la medida de *separación* para describir el poder computacional de las piscinas implementadas como RBP, y al mismo tiempo la relación que *evanescencia* y *calidad de kernel* (KQ) tienen con la misma, se presenta aquí la elección arbitraria de siete zonas en el plano $\langle K \rangle$ vs. *número de entradas*, para redes de topología K -variable con $N = 1000$ y una piscina efectiva de 563 nodos. Los parámetros fijos son: activación media: 1, refracción media: 1, varianza nodal: 1.0, nodos inhibidores 0.0, parámetro de retardos: 0.6. Se muestra su desempeño para cada una de las tareas descritas en este trabajo.

Para el cálculo de su desempeño se empleó el mismo procedimiento empleado para las piscinas implementadas como RBA descrito anteriormente. Un mapa de las zonas elegidas se puede ver en la figura 6.9. Los resultados en el cuadro 6.2.

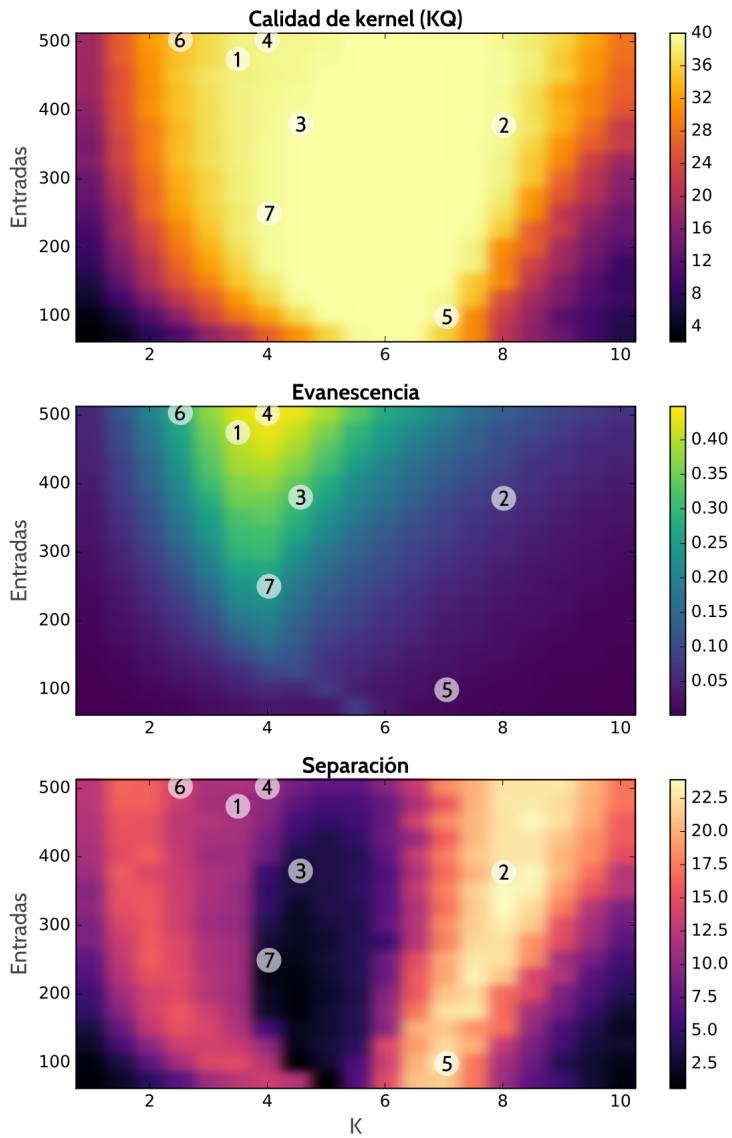


Figura 6.9: Localización de las RBP evaluadas en el plano $\langle K \rangle$ vs. número de entradas (ver texto).

Zona	Precisión por tarea		
	Densidad	Paridad de cambio	Paridad
1	0.9999	0.9378	0.7949
2	0.8777	0.7104	0.6131
3	0.9994	0.9152	0.7672
4	0.9998	0.9130	0.7943
5	0.6706	0.5613	0.5450
6	0.9905	0.9036	0.7629
7	0.9992	0.8494	0.6901

Tabla 6.2: Resultados de clasificación para piscinas RBP de acuerdo su origen en el plano $\langle K \rangle$ vs. número de entradas. La ubicación de cada zona puede consultarse en la figura 6.9. Al igual que en las RBA, clasificación decrece en precisión de acuerdo a la dificultad de la tarea. El valor obtenido es el promedio de cuatro validaciones cruzadas de 10 hojas cada una. La precisión de cada validación es el promedio de aciertos para cada clase.

Consultando el mapa y los datos salta inmediatamente a la vista que el desempeño de las redes está completamente deslindado de la medida de *separación*. La zona de mejor desempeño, la número 1, no muestra un índice particularmente alto para dicha medida.

Las zonas 2 y 5, que presentan los valores más altos para *separación*, no son simplemente malas. En realidad son las peores entre todas. Pero la relación no es inversa. Las zonas 3 y 7, sin ser demasiado buenas, no son tan malas como la 2 y 5, a pesar de contar con los índices de *separación* más bajos de todos.

Por otro lado, si ponemos nuestra atención en las medidas de *evanescencia* y *calidad de kernel* (KQ) las cosas se vuelven mucho más claras. Las mejores ubicaciones se corresponden con altos valores para la *evanescencia*. La relación es casi directa, pero no por completo. La zona 4, que presenta los índices más altos, no es la mejor. Ese lugar le corresponde a la zona 1, que está muy próxima a ella, pero no es precisamente la que maximiza tal medición.

Con todo, las redes RBP presentan mejores resultados que redes RBA de tamaño equivalente. La disparidad y aparente arbitrariedad en el criterio que determina si las redes con determinadas características son potencialmente buenas, nos habla de una profunda diferencia en el mecanismo que hace posible su eficacia, que no es plenamente comprendido.

6.2.2. Persistencia de la memoria

Aquí se ha pretendido evaluar la memoria e integración de la información a través del tiempo de las piscinas implementadas como RBP. El objetivo es averiguar con qué exactitud la piscina es capaz de recordar si un símbolo en la entrada inyectado τ pasos de tiempo atrás fue un estímulo o no, en función del parámetro de retardo. Para tal labor se ha aplicado la tarea de *Paridad* a ventanas de tamaño $n = 1$.

Objetivos similares se han realizado para la tarea de *Paridad* con ventanas de tamaño $n = 2$ y $n = 4$

Las piscinas RBP mostradas aquí se han tomado de zonas favorables al cómputo, de acuerdo a los criterios descritos en el apartado anterior. No se asegura que sean redes óptimas. En la figura 6.10 podemos ver la precisión en las labores descritas.

Como es fácil observar ahí, las tareas de identificación y clasificación mejoran sensiblemente su precisión al aumentar el número de nodos efectivos de la piscina y al reducir el parámetro de retardo. Al menos si la tarea consiste en identificar un sólo pulso en el intervalo de tiempo estudiado. Las tareas de clasificación complejas, como *Paridad* encuentran su mejor desempeño para ciertos valores del parámetro de retardo que está en función del tamaño de la red y sus características intrínsecas.

Los resultados nos dan cuenta de algo muy importante: no basta tener la capacidad de recordar de manera perfecta el momento en que existió o no un estímulo para realizar un cálculo o clasificación correctos, es necesario que la piscina posea la dinámica adecuada para integrar la información de la manera apropiada. Las redes presentadas en la gráfica superior de la figura 6.10, tienen un desempeño perfecto para $n = 1$ cuando el parámetro de retardo es inferior a 0.6. Son capaces de identificar sin error alguno si existió un estímulo τ pasos de tiempo en el pasado, siempre que se encuentre en los rangos estudiados, no importando los símbolos precedentes. Sin embargo, este conocimiento se consigue en detrimento de su capacidad para integrar y separar la información según el parámetro de retardo se vuelve cada vez más pequeño. Característica que se ve reflejada en la repentina reducción en su desempeño a partir de ese punto para la tarea de paridad con $n = 4$.

La capacidad para recordar en función de la lejanía en el tiempo de los estímulos se presenta a continuación para piscinas implementadas como RBA y

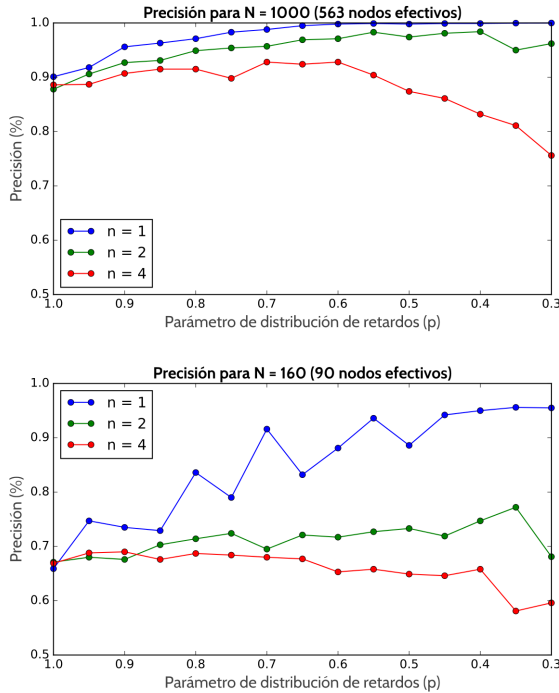


Figura 6.10: Buena parte del desempeño de las redes RBP se encuentra en función de su tamaño y la distribución de los retrasos de sus conexiones. Para la obtención de ambos gráficos las redes con $N = 160$ y $N = 1000$ se han construido bajo los mismos parámetros, conservando la proporción de nodos utilizados como entrada. Topología: K-variable, nodos inhibidores: 0.0, activación media: 1, refracción media: 1, varianza nodal: 1.0, $\langle K \rangle$: 3.5. Se ha evaluado aquí el desempeño de la tarea *Paridad*, para ventanas con $n = 1$, $\tau = 5$ (azul), $n = 2$, $\tau = 4$ (verde) y $n = 4$, $\tau = 2$ (rojo). Cada punto es el promedio en el puntaje de cuatro validaciones cruzadas de diez hojas, cada una realizada con 19,400 vectores de estado etiquetados. *Precisión* es el promedio del porcentaje de aciertos en la clasificación para cada clase.

RBP con tamaños equiparables. En este caso la tarea a desempeñar es similar a las anteriores. El objetivo es también identificar la presencia o ausencia de un estímulo en el pasado, en este caso variando el valor de τ .

Los resultados de estas tareas, para redes de características similares a las utilizadas en la prueba anterior, se pueden consultar en la figura 6.11.

Como es de esperar, la capacidad para identificar lo acontecido en un instante determinado del pasado decrece conforme τ aumenta. A mayor lejanía una menor precisión en la estimación. Es de notar que la memoria del sistema aumenta con el tamaño de la red.

6.3. Defectos

¿No deberían ser las RBP más potentes de lo que se muestran? Al analizar visualmente el comportamiento y trayectoria de las redes con mejores puntuaciones, destaca sobremanera lo «rígidas» que son. Sus tiempos transientes son en extremo pequeños. Contemplándolas resulta difícil de creer su desempeño y uno se ve tentado a considerar que deberían olvidar los estímulos pasados con gran rapidez (como de hecho pasa). ¿Qué sucede? ¿Por qué las redes mejor calificadas no son más como la red tipo «b» de la figura 6.6 en vez de ser como la red tipo «a»?

La razón, después de meditarlo, es obvia. Es verdad que largos transientes mejoran la memoria de las redes y le permiten integrar más información por más tiempo. Pero tal habilidad tiene un costo, y este es la reducción de la medida de *evanescencia* y/o la *calidad de kernel* (KQ). Las medidas obtenidas en las indagaciones lo sugieren y la razón se puede visualizar de la siguiente manera:

Imagínese una red hipotética con mil estados posibles además de su estado de reposo. Si el sucesor de todos fuera el estado estable, la red tendría a su disposición mil trayectorias sobre las cuales proyectar las entradas que recibe. El costo por tener tantas trayectorias es el olvido inmediato. Un paso de tiempo bastaría para olvidarlo todo. Sólo haría falta un paso para entrar en reposo. El opuesto extremo sería la existencia de una sola trayectoria de mil unidades de tiempo. El transiente más largo posible en dicha red. En tal caso, todas las entradas se confundirían inmediatamente entre sí, no importando cuanto tiempo sea capaz de recordarlas. Una tarea de separación se vuelve imposible.

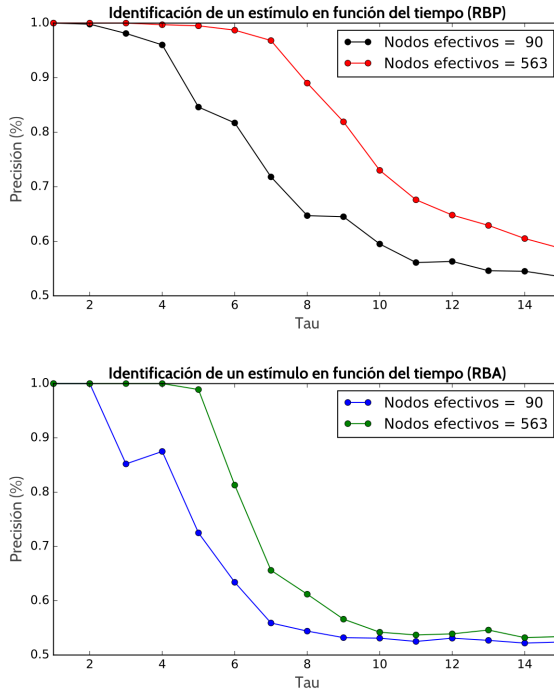


Figura 6.11: La capacidad para recordar un estímulo en función del tiempo acontecido desde el mismo. Un aumento en el tamaño de la red mejora la memoria pero ésta decae rápidamente con el tiempo. En particular las RBA lo hacen abruptamente superado cierto valor de τ que se encuentra en función de su tamaño. Por el contrario las RBP presentan una mejora y una disminución en su desempeño más suave a través del tiempo. La memoria evanescente del sistema en acción. Los valores presentados aquí son calculados de la misma forma que los mostrados en la figura 6.10.

Ventana	Densidad	Paridad de cambio	Paridad
00000	□	■	■
00001	□	□	□
00011	□	□	■
00010	□	■	□
00110	□	■	■
00111	■	□	□
00101	□	□	■
00100	□	■	□
01100	□	■	■
01101	■	□	□
01111	■	□	■
01110	■	■	□
01010	□	■	■
01011	■	□	□
01001	□	□	■
01000	□	■	□
11000	□	□	■
11001	■	■	□
11011	■	■	■
11010	■	□	□
11110	■	□	■
11111	■	■	□
11101	■	■	■
11100	■	□	□
10100	□	□	■
10101	■	■	□
10111	■	■	■
10110	■	□	□
10010	□	□	■
10011	■	■	□
10001	□	■	■
10000	□	□	□

Tabla 6.3: Asignación de clases para las tareas descritas aquí, para ventanas con $n = 5$. Se han ordenado siguiendo el código Grey, lo que hace evidente la dificultad de cada tarea. Mientras que *Densidad* presenta varios elementos contiguos para cada clase, *Paridad* los intercala al extremo, haciendo que el cambio en un sólo bit haga la diferencia entre ellas. Esto exige de la red la máxima precisión al momento de identificar los estímulos previos, así como gran capacidad para integrar dicha información. *Paridad de cambio* presenta un intercalado similar a *Paridad*, pero ligeramente más simple de resolver.

Largos transientes implican un sacrificio en el número de trayectorias efectivas utilizables por la red para mapear las entradas. Es un lujo que sólo pueden darse redes verdaderamente grandes. Las redes estudiadas aquí eran demasiado pequeñas para obtener buenos resultados con transientes largos.

Capítulo 7

Conclusiones

Se han dado evidencias de la efectividad de las RBP como medio dinámico para un sistema de cómputo de reservorio. Además se ha mostrado su superioridad sobre las RBA tradicionales, aunque no con demasiado margen por ahora. De igual forma se ha demostrado que cumplen el papel de una función kernel temporal, capaz de transformar una tarea no lineal de forma que sea posible alimentar con ella a unidades de salida enteramente lineales en tareas de clasificación.

Definir la perturbación de las redes booleanas en términos de «normas», redes que se alternan en tiempo de ejecución, ha favorecido notablemente la abstracción y comprensión de la dinámica nodal, brindando herramientas conceptuales para el análisis de la misma.

El campo de investigación de las RBP o sus posibles derivadas se muestra amplio. Aquí se revisa de manera muy somera su potencial. Sin embargo, las redes estudiadas en este trabajo son demasiado pequeñas para brindar resultados que resulten prácticos.

Hay un punto en el que no se ha hecho suficiente énfasis, pero resulta capital para entender el completo potencial de los repositorios presentados aquí como mecanismos de cómputo.

Tal vez parezca que las redes mostradas realizan funciones demasiado simples, y que además de ello son «grandes» con respecto a la trivialidad de sus operaciones. Aunque hay algo de cierto en ello, se desea hacer reconsiderar al lector si tal es su pensamiento.

Es verdad que las redes poseen muchos elementos, y resulta evidente que el número deberá ser bastante más grande para tareas con mayor utilidad práctica. Pero esto es parcialmente cierto cuando hablamos en términos de la implementación de las mismas.

Las redes neuronales tradicionales, por dar un ejemplo, consisten en elementos bastante más complejos. Buena parte de las neuronas artificiales necesitan operaciones de punto flotante de gran precisión para ser efectivas. Por contra, las redes presentadas aquí son en extremo simples. La operación de sus elementos sólo implica aritmética entera y canales de comunicación de uno o dos bits entre ellos. Las conexiones entre los nodos no están ponderadas y cada una de ellas posee el mismo peso siempre. Esto facilita enormemente su implementación en hardware, donde sería posible la construcción de redes programables masivas, capaces de brindar resultados en tiempo real.

Otra razón que las hace parecer menos efectivas de lo que en realidad pueden llegar a ser es psicológica. En su estudio nos hemos concentrado en tareas individuales. Sin embargo, no debe perderse de vista que las tareas evaluadas aquí se llevaron a cabo con los mismos vectores de estado etiquetados de manera diferente. Es decir, sólo hace falta una sola piscina para realizar muchas operaciones a la vez. Tan sólo es necesaria una unidad de lectura lineal para cada tarea. Ellas se encargarán de «traducir» la computación intrínseca de la red en información útil para nosotros. Pueden conectarse a la red tantas unidades de lectura como se desee que sigan un esquema de clasificación enteramente arbitrario. En principio la red, por el simple hecho de funcionar, calcula ya todo lo que es posible calcular sobre la entrada en los rangos de tiempo donde resulta efectiva. Lo cual es, por sí mismo, un resultado bastante notable.

Quedan aún muchos campos por explorar. ¿Cómo escalan? ¿Cómo reaccionan ante una multiplicidad de flujos de entrada? Este fue un acercamiento inicial a un nuevo modelo de red que ofrece amplias posibilidades de estudio.

Glosario

Atractor. Estado o conjunto de ellos a los que conduce la dinámica de un sistema y del cual es imposible escapar sin aplicar una perturbación al mismo.

Des(M, τ). Tasa de desvanecimiento de la red M en τ pasos de tiempo.

Espacio fásico. El conjunto de todos los estados posibles de un sistema dinámico.

Estado cíclico. Un conjunto de estados que, debido a la dinámica del sistema, se repite incesantemente. También se conoce como *atractor cíclico*.

Estímulo. Evento capaz de alterar la dinámica de un sistema dinámico abierto.

Eva(M, τ). Evanescencia de la red M en τ pasos de tiempo.

Función *kernel*. Una función que mapea un espacio a otro con diferente dimensionalidad.

GR. Rango de generalización. Hace referencia a la capacidad de una piscina para proyectar sus entradas a un espacio de dimensión mayor cuando éstas son variaciones pequeñas de la misma.

Jardín del edén. Hace referencia al estado que carece de antecesor, y por tanto, inalcanzable a través de la dinámica del sistema.

KQ. Calidad de *kernel*. Hace referencia a la capacidad de una piscina para proyectar sus entradas a un espacio de dimensión mayor cuando estas son arbitrarias y no relacionadas entre sí.

MEL. Máquina de Estado Líquido. Modelo de computación de reservorio consistente en una o más piscinas constituidas por neuronas de pulso. Su espacio fásico es continuo.

Nodo multivaluado. Nodo capaz de poseer más de dos valores para su estado interno.

Norma Red booleana idéntica en dinámica a una red perturbable sometida a una combinación particular de estímulos.

Norma de estímulo. Norma que no es de reposo.

Norma de reposo. Norma asociada a la ausencia total de estímulos aplicados a una red perturbable.

Per(M, a, b, \mathcal{T}). Perturbación de la red M entre las entradas a y b para \mathcal{T} pasos de tiempo.

Piscina. Un medio dinámico abierto capaz de ser perturbado por eventos ajenos a él mismo.

RB. Red Booleana.

RBA. Red Booleana Aleatoria.

RBP. Red Booleana de Pulsos.

Red perturbable. Una red capaz de alterar su dinámica en función de eventos ajenos a ella misma.

Succ(x). Sucesor de x .

Transiente. Número de pasos temporales que necesita un sistema dinámico discreto para ser conducido de un estado arbitrario a un atractor.

Trayectoria. Conjunto ordenado de estados, donde cada uno es el sucesor del anterior, excepto el primero de ellos.

Bibliografía

- Amari, S. I. (1972). Characteristics of random nets of analog neuron-like elements. *Systems, Man and Cybernetics, IEEE Transactions on, SMC-2*(5), 643-657.
- Arthur, J. V., Merolla, P. A., Akopyan, F., Alvarez, R., Cassidy, A., Chandra, S., ... Rubin, D. B. y col. (2012). Building block of a programmable neuromorphic substrate: A digital neurosynaptic core. En *Neural Networks (IJCNN), The 2012 International Joint Conference on* (pp. 1-8). IEEE.
- Auer, P., Burgsteiner, H. & Maass, W. (2008). A learning rule for very simple universal approximators consisting of a single layer of perceptrons. *Neural Networks, 21*(5), 786-795.
- Bartlett, P. L. & Maass, W. (2003). Vapnik-Chervonenkis dimension of neural nets. *The handbook of brain theory and neural networks*, 1188-1192.
- Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Busat, J., ... Boahen, K. (2014). Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE, 102*(5), 699-716.
- Boyd, S. & Chua, L. O. (1985). Fading memory and the problem of approximating nonlinear operators with Volterra series. *Circuits and Systems, IEEE Transactions on, 32*(11), 1150-1161.
- Buonomano, D. V. & Merzenich, M. M. (1995). Temporal information transformed into a spatial code by a neural network with realistic properties. *Science, 267*(5200), 1028.
- Büsing, L., Schrauwen, B. & Legenstein, R. (2010). Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural computation, 22*(5), 1272-1311.
- Church, A. (1932). A set of postulates for the foundation of logic. *Annals of mathematics*, 346-366.
- Derrida, B. & Pomeau, Y. (1986). Random networks of automata: a simple annealed approximation. *EPL (Europhysics Letters), 1*(2), 45.

- Fernández, N., Maldonado, C. & Gershenson, C. (2014). Information measures of complexity, emergence, self-organization, homeostasis, and autopoiesis. En *Guided self-organization: Inception* (pp. 19-51). Springer.
- Fernando, C. & Sojakka, S. (2003). Pattern recognition in a bucket. En *Advances in artificial life* (pp. 588-597). Springer.
- Freeman, W. J. (1975). *Mass Action in the Nervous System*. New York: Academic Press.
- Gazzaniga, M. S. (1967). The split brain in man. *Scientific American*, 217(2), 24-29.
- Gazzaniga, M. S. (1995). Principles of human brain organization derived from split-brain studies. *Neuron*, 14(2), 217-228.
- Gershenson, C. (2004). Introduction to random Boolean networks. *arXiv preprint nlin/0408006*.
- Gershenson, C. (2014). Enfrentando a la complejidad: Predecir vs. adaptar. *Complexica: cervell, societat i llengua des de la transdisciplinarietat*.
- Gershenson, C. & Heylighen, F. (2003). When can we call a system self-organizing? En *Advances in artificial life* (pp. 606-614). Springer.
- Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. En *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Hilbert, D. & Ackermann, W. (1928). Grundzüge der Theoretischen Logik. *Die Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen*, 27.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8), 2554-2558.
- Hornik, K., Stinchcombe, M. & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5), 359-366.
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, 9(3), 90-95.
- Jäger, H. (2001). The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148, 34.
- Jones, E., Oliphant, T., Peterson, P. y col. (2001-). SciPy: Open source scientific tools for Python. [Online; accessed 2016-02-27]. Recuperado desde <http://www.scipy.org/>
- Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. En

- Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 1725-1732).
- Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3), 437-467.
- Kauffman, S. A. (1993). *The origins of order: Self organization and selection in evolution*. Oxford University Press, USA.
- Kleene, S. C. (1943). Recursive predicates and quantifiers. *Transactions of the American Mathematical Society*, 53(1), 41-73.
- Kuri, A. F. & Villegas, C. (1998). A universal eclectic genetic algorithm for constrained optimization. En *Proceedings of the 6th European congress on intelligent techniques and soft computing* (Vol. 1, pp. 518-522).
- Langton, C. G. (1990). Computation at the edge of chaos: phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1), 12-37.
- Little, W. (1974). The existence of persistent states in the brain. *Mathematical Biosciences*, 19(1), 101-120.
- Lopez-Ruiz, R., Mancini, H. & Calbet, X. (2002). A statistical measure of complexity. *arXiv preprint nlin/0205033*.
- Lukoševičius, M. & Jäger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3), 127-149.
- Maass, W., Legenstein, R. A. & Bertschinger, N. (2004). Methods for estimating the computational power and generalization capability of neural microcircuits. En *Advances in neural information processing systems* (pp. 865-872).
- Maass, W. & Markram, H. (2006). Theory of the computational function of microcircuit dynamics. En *The interface between neurons and global brain function, Dahlem Workshop Report* (Vol. 93, pp. 371-390).
- Maass, W., Natschläger, T. & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11), 2531-2560.
- Maass, W., Natschläger, T. & Markram, H. (2004). Fading memory and kernel properties of generic cortical microcircuit models. *Journal of Physiology-Paris*, 98(4), 315-330.
- Merolla, P., Arthur, J., Akopyan, F., Imam, N., Manohar, R. & Modha, D. S. (2011). A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm. En *Custom Integrated Circuits Conference (CICC), 2011 IEEE* (pp. 1-4). IEEE.

- Minsky, M. & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, Cambridge MA.
- Mitchell, M., Hraber, P. & Crutchfield, J. P. (1993). Revisiting the edge of chaos: Evolving cellular automata to perform computations. *arXiv preprint adapt.org/9303003*.
- Natschläger, T., Maass, W. & Markram, H. (2002). The “liquid computer”: A novel strategy for real-time computing on time series. *Special issue on Foundations of Information Processing of TELEMATIK*, 8(LNMC-ARTICLE-2002-005), 39-43.
- Niesche, H. (2006). Introduction to cellular automata. En *Seminar “Organic Computing” SS2006*. Citeseer.
- Oosawa, C. & Savageau, M. A. (2002). Effects of alternative connectivity on behavior of randomly constructed Boolean networks. *Physica D: Nonlinear Phenomena*, 170(2), 143-161.
- Packard, N. H. (1988). Adaptation toward the edge of chaos. En J. A. S. Kelso, A. J. Mandell & M. F. Shlesinger (Eds.), *Dynamic patterns in complex systems* (pp. 293-301). Singapur: World Scientific.
- RAE. (2014). Silla. En *Diccionario de la Lengua Española (23.a ed.)* Real Academia Española. Recuperado desde <http://dle.rae.es/?id=XtQvCwh>
- Renyi, A. & Erdős, P. (1959). On random graphs I. *Publicationes Mathematicae*, 6, 290-297.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1985). Learning internal representations by error propagation. *Parallel Distributed Processing*, 1, 318-362.
- Schmidhuber, J., Wierstra, D., Gagliolo, M. & Gomez, F. (2007). Training recurrent networks by evolino. *Neural computation*, 19(3), 757-779.
- Schrauwen, B., Büsing, L. & Legenstein, R. (2009). On computational power and the order-chaos phase transition in reservoir computing. En *22nd Annual conference on Neural Information Processing Systems (NIPS 2008)* (Vol. 21, pp. 1425-1432). NIPS Foundation.
- Schrauwen, B., Verstraeten, D. & Campenhout, J. V. (2007). An overview of reservoir computing: theory, applications and implementations. En *Proceedings of the 15th European Symposium on Artificial Neural Networks*. p. 471-482 2007 (pp. 471-482).

- Shannon, C. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27, 379-423, 623-656.
- Smith III, A. R. (1971). Simple computation-universal cellular spaces. *Journal of the ACM (JACM)*, 18(3), 339-353.
- Snyder, D., Goudarzi, A. & Teuscher, C. (2012). Finding optimal random boolean networks for reservoir computing. En *Artificial Life* (Vol. 13, pp. 259-266).
- Snyder, D., Goudarzi, A. & Teuscher, C. (2013). Computational capabilities of random automata networks for reservoir computing. *Physical Review E*, 87(4), 042808.
- Steil, J. J. (2006). Online stability of backpropagation–decorrelation recurrent learning. *Neurocomputing*, 69(7), 642-650.
- Stein, R. B. (1967). Some models of neuronal variability. *Biophysical journal*, 7(1), 37.
- Sussillo, D., Toyozumi, T. & Maass, W. (2007). Self-tuning of neural circuits through short-term synaptic plasticity. *Journal of neurophysiology*, 97(6), 4079-4095.
- Tsodyks, M., Uziel, A. & Markram, H. (2000). Synchrony generation in recurrent networks with frequency-dependent synapses. *J Neurosci*, 20(1), 825-835.
- Turing, A. M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42), 230-265.
- van der Walt, S., Colbert, S. C. & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering*, 13(2), 22-30. doi:10.1109/MCSE.2011.37
- van Rossum, G. & Drake Jr, F. L. (nodate). *The Python Language Reference Manual (version 3.2)*.
- Vapnik, V. N. (1998). *Statistical learning theory*. Wiley New York.
- Wardlaw, W. P. (2005). Row rank equals column rank. *Mathematics Magazine*, 78(4), 316-318.
- Watts, D. J. & Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *nature*, 393(6684), 440-442.
- Wolfram, S. (1984). Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1), 1-35.
- Wolfram, S. (2002). *A New Kind of Science*. Champaign, IL: Wolfram Media. Recuperado desde <http://www.wolframscience.com>

Wuensche, A. y col. (1998). Discrete dynamical networks and their attractor basins. *Complexity International*, 6, 3-21.

Índice de figuras

1.1. Diagrama de Venn sobre el concepto «silla»	13
1.2. Silla miniatura	15
2.1. Esquema de un perceptrón	23
2.2. Esquema del cómputo de reservorio	26
3.1. Tabla de transiciones, grafo y mapa de estados de una RBA . . .	37
3.2. Cuenca de atracción	38
3.3. Evolución temporal de tres RBA (ordenada, crítica y caótica) . .	39
3.4. Tabla de transiciones con un nodo ω	41
3.5. Una perturbación como el cambio de RB en tiempo real	43
3.6. Convergencia de trayectorias en una RB	47
3.7. Voltaje de membrana neuronal a través del tiempo	50
3.8. Variable interna de un nodo a través del tiempo	53
3.9. Evolución temporal de una RBP	59
3.10. Nodo de pulso	60
3.11. Nodo multivaluado equivalente a uno de pulso	60
3.12. Conjunto de nodos binarios equivalente a un nodo de pulso . .	61
4.1. Relación de <i>entropía, auto-organización y complejidad</i>	67
4.2. Taza de desvanecimiento	76
6.1. <i>Separación y evanescencia</i> en el plano $\langle K \rangle$ vs. <i>sesgo</i> para una RBA	91
6.2. <i>Homeostasis y complejidad</i> en el plano $\langle K \rangle$ vs. <i>sesgo</i> para una RBA	92
6.3. <i>Separación y evanescencia</i> en el plano λ vs. C para una RBA . . .	93
6.4. $\langle K \rangle K$ vs. <i>nodos de entrada</i> para piscinas RBP	95
6.5. Número de entradas contra diversos parámetros de medición .	97
6.6. Respuesta de tres diferentes RBP a una misma entrada	98
6.7. Redes en el plano $\langle K \rangle$ vs. <i>sesgo</i>	101

6.8. Plano <i>varianza nodal</i> vs. <i>retardos</i> de una RBP	103
6.9. Redes en el plano $\langle K \rangle$ vs. <i>número de entradas</i>	105
6.10. Desempeño de diferentes RBP en función de los retrasos	108
6.11. Desempeño de piscinas RBA y RBP en función de tiempo	110

Índice de tablas

2.1. Clasificación de sistemas de computación	25
6.1. Resultados de clasificación para piscinas RBA	100
6.2. Resultados de clasificación para piscinas RBP	106
6.3. Asignación de clases para las tareas empleadas en ventanas con $n = 5$	111