



Trabajo Fin de Grado

Diseño y análisis de redes homeostáticas adaptativas

Design and analysis of homeostatic adaptive
networks

Autor

Alberto Martínez Menéndez

Directores

Manuel González Bedía

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2018



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. Alberto Martínez Menéndez,

con nº de DNI 25356895Q en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)
Grado , (Título del Trabajo)
Diseño y análisis de redes homeostáticas adaptativas

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 25 de junio de 2018

Fdo: Alberto Martínez Menéndez

Diseño y análisis de redes homeostáticas adaptativas

RESUMEN

Conseguir dotar a un agente artificial con la capacidad de adaptabilidad presente en los seres vivos le proporcionaría habilidades necesarias para realizar tareas para las que no ha sido entrenado o para las que se le ha entrenado pero en un entorno cambiante o con incertidumbre. En este trabajo nos interesaremos, desde una perspectiva ingenieril, por unos de los sistemas de autoregulación presentes en los organismos vivos conocidos como mecanismos homeostáticos, que están en la base de ciertas capacidades de adaptación.

En particular, se ha partido del diseño e implementación de un agente homeostático basado en redes neuronales recurrentes de tiempo continuo con capacidad de fototaxis (búsqueda y acercamiento a una serie de luces en un espacio). Este agente se corresponde con el mejor candidato de una cierta población, seleccionado mediante un algoritmo genético. La peculiaridad del mismo, es que despliega comportamiento homeostático a través de mecanismos de plasticidad, que le permiten modificar sus variables internas para alcanzar un estado estable y mantenerse en el mismo aunque se vea sometido a perturbaciones externas.

Una vez obtenido el agente con las propiedades de fototaxis y plasticidad homeostática, se ha buscado dotar al mismo de capacidades para que emergan comportamientos sociales con el fin de analizar cómo interacciona en situaciones donde hay más de un agente involucrado. El comportamiento social se ha añadido siguiendo dos aproximaciones diferentes. La primera, asume que las capacidades del agente son aditivas. Es decir, que el agente tras ser capaz de navegar y orientarse hacia un foco de luz (o comida), necesitaría una nueva estructura neuronal con la que codificar otra capacidad novedosa (en este caso, coordinación social). Se modelará el sistema con dos mecanismos independientes en el controlador del sistema, uno para la navegación y otro para el aspecto social. La segunda, defiende que el comportamiento social es una característica estructural del sistema. Por tanto, una nueva capacidad atraviesa (en ocasiones se dice que “percola”) las capacidades previas, reestructurando el controlador neuronal del agente en su conjunto. El objetivo es comparar estas dos aproximaciones y sacar conclusiones sobre ello mediante experimentos en los que varios agentes tienen que interactuar de manera social para la consecución de un objetivo.

Design and analysis of homeostatic adaptative networks

ABSTRACT

Being able to give adaptability features present in living beings to artificial agents, would provide them with the necessary skills to perform tasks for which they have not been trained for, or for which they have been trained but in a changing environment or with uncertainty. In this project we are interested, from an engineering perspective, in some of the self-regulating systems present in living organisms, known as homeostatic mechanisms, which are in the base of certain adaptation features.

In particular, the starting point has been the design and implementation of an homeostatic agent based on continuous time recurrent neuronal networks with fototaxis features (ability to search and get closer to a lightsource). This agent will be the best candidate within a certain population, selected by a genetic algorithm. Which is interesting from the agent is that he develops homeostatic behaviour by plasticity mechanisms that let him modify his internal variables in order to reach a stable state and to keep himself in it even if affected by external disturbances.

Once the agent with fototaxis behaviour and plasticity properties is obtained, we will give him certain features to let social behaviours appear, with the goal of analizing how does he interact in situations where more than one agent is involved. This social behaviour will be added following two different approaches. The first one assumes that agent features are additive. This means that the agent, once being able to orientate and move closer to a lightsource (or food), would need a new neuronal structure to encode a new feature (social coordination in this case). The system will be modeled with two different mechanisms in the controller, one for navigation and one for the social part. The sencond one defends that social behaviour is a structural feature of the system. Therefore, a new feature breaks throught previous features, restructuring the neuronal controller of the agent as a whole. The objective is to compare this two approaches and get conclusions about it by executing experiments in which various agents have to interact in a social way to achieve an objective.

Índice

Lista de Figuras	VII
Lista de Tablas	IX
1. Introducción	1
1.1. Contexto y motivación	2
1.2. Objetivos	2
1.3. Estructura	3
1.4. Planificación	3
2. Estado del arte	1
3. Fundamentos teóricos	1
3.1. Redes neuronales recurrentes de tiempo continuo (CTRNN)	1
3.2. Homeostasis	2
3.3. Algoritmo genético	4
4. Modelo e implementación	1
4.0.1. Plasticidad	2
4.1. Agentes	4
4.1.1. Agente 0: agente sin comportamiento social	4
4.1.2. Agente 1: agente colectivo aditivo	5
4.1.3. Agente 2: agente colectivo estructural	6
4.2. Evolución	7
4.2.1. Codificación de los agentes	7
4.2.2. Creación de la población inicial	8
4.2.3. Función de selección	8
4.2.4. Función de recombinación	8
4.2.5. Función de mutación	9
4.2.6. Función fitness	9
4.3. Simetría	12

4.4. Parámetros y rangos	12
5. Experimentos	13
5.1. Experimento individual	13
5.1.1. Resultados del Agente 0	13
5.2. Experimento colectivo	14
5.2.1. Resultados del Agente 1	15
5.2.2. Resultados del Agente 2	22
6. Conclusiones	28
7. Conclusiones personales	31
8. Bibliografía	32
Anexos	34
A. Homeostat	35
B. Algoritmo genético	37
B.0.1. Componentes, estructura y terminología	37
C. Gráficas completas de las activaciones neuronales del grupo de agentes de tipo 1	41
D. Gráficas completas de las activaciones neuronales del grupo de agentes de tipo 2	44
E. Planificación y cronograma	47

Listas de Figuras

3.1. Valor de salida de una CTRNN. Izquierda: Un nodo autoconectado. Derecha: Función sigmoide aplicada para calcular la salida de una neurona.	2
3.2. Esquema de un Homeostato individual. La unidad recibe las entradas del resto de unidades ($w_i s_i$) y de si misma ($w_1 s_1$). La salida s_1 es una función lineal por partes de la suma ponderada de las entradas.)	3
3.3. Ejemplo de función de transferencia de una unidad Homeostat. La función es lineal por partes con puntos en $(x_1, j_1), \dots, (x_p, y_p)$ donde x_1 $= 0$ y $x_p = N$, para todas las unidades en una red de N unidades. Las líneas discontinuas representan el rango homeostatico de la red.	3
4.1. Esquema del agente utilizado.	1
4.2. Esquema de la estructura del controlador del “Agente 0”	4
4.3. Esquema de la estructura del controlador del “Agente 1”	5
4.4. Esquema de la estructura del controlador del “Agente 1”	6
4.5. Representación codificado de un cromosoma que representa un agente de tipo 0.	7
4.6. Representación codificado de un cromosoma que representa un agente de tipo 1 o 2.	7
5.1. Trayectoria del “Agente 0” durante la realización del experimento individual.	14
5.2. Fragmento de trayectorias del conjunto de agentes de tipo 1 que demuestran comportamiento de fototaxis colectiva.	16
5.3. Activaciones de las 6 neuronas del agente representativo del grupo de agentes de tipo 1.	17
5.4. Diagramas de caja para las mediciones de la fitness en presencia de ruido en los sensores de los agentes de tipo 1.	18
5.5. Diagramas de caja para las mediciones de la fitness en presencia de ruido en los ritmos de plasticidad de los agentes de tipo 1.	20

5.6. Diagramas de caja para las mediciones de la fitness en presencia de ruido en los pesos sinápticos de los agentes de tipo 1.	21
5.7. Fragmento de trayectorias del conjunto de agentes de tipo 2 que demuestran comportamiento de fototaxis colectiva.	22
5.8. Activaciones de las 6 neuronas del agente representativo del grupo de agentes de tipo 2.	23
5.9. Diagramas de caja para las mediciones de la fitness en presencia de ruido en los sensores de los agentes de tipo 2.	24
5.10. Diagramas de caja para las mediciones de la fitness en presencia de ruido en los ritmos de plasticidad de los agentes de tipo 2.	26
5.11. Diagramas de caja para las mediciones de la fitness en presencia de ruido en los pesos sinápticos de los agentes de tipo 2.	27
A.1. Gráfica de salidas del Homeostato implementado. Eje X: tiempo. Eje Y: salidas de cada una de las neuronas que componen el Homeostato.	36
B.1. Ejemplo de estructura de población, cromosomas y genes.	38
B.2. Ejemplo de operación de recombinación de dos cromosomas.	39
C.1. Activaciones de las 6 neuronas del agente número 1 del tipo 1.	41
C.2. Activaciones de las 6 neuronas del agente número 2 del tipo 1.	42
C.3. Activaciones de las 6 neuronas del agente número 3 del tipo 1.	42
C.4. Activaciones de las 6 neuronas del agente número 4 del tipo 1.	43
C.5. Activaciones de las 6 neuronas del agente número 5 del tipo 1.	43
D.1. Activaciones de las 6 neuronas del agente número 1 del tipo 2.	44
D.2. Activaciones de las 6 neuronas del agente número 2 del tipo 2.	45
D.3. Activaciones de las 6 neuronas del agente número 3 del tipo 2.	45
D.4. Activaciones de las 6 neuronas del agente número 4 del tipo 2.	46
D.5. Activaciones de las 6 neuronas del agente número 5 del tipo 2.	46
E.1. Cronograma del proyecto dividido por semanas, incluyendo las horas dedicadas cada semana a cada tarea.	47

Lista de Tablas

4.1.	Rango de valores para los parámetros de la red	12
5.1.	Tabla con el área que cada neurona de cada agente del conjunto de agentes de tipo 1 ha permanecido fuera de los límites de estabilidad establecidos.	18
5.2.	P-valores obtenidos de la ejecución del test <i>t de Student</i> sobre los pares de poblaciones de mediciones obtenidas.	19
5.3.	P-valores obtenidos de la ejecución del test <i>t de Student</i> sobre los pares de poblaciones de mediciones obtenidas.	20
5.4.	P-valores obtenidos de la ejecución del test <i>t de Student</i> sobre los pares de poblaciones de mediciones obtenidas.	21
5.5.	Tabla con el área que cada neurona de cada agente del conjunto de agentes de tipo 2 ha permanecido fuera de los límites de estabilidad establecidos.	24
5.6.	P-valores obtenidos de la ejecución del test <i>t de Student</i> sobre los pares de poblaciones de mediciones obtenidas.	25
5.7.	P-valores obtenidos de la ejecución del test <i>t de Student</i> sobre los pares de poblaciones de mediciones obtenidas.	26
5.8.	P-valores obtenidos de la ejecución del test <i>t de Student</i> sobre los pares de poblaciones de mediciones obtenidas.	27

Capítulo 1

Introducción

Incorporar comportamiento adaptativo en agentes artificiales es uno de los medios esenciales para que estos puedan desenvolverse en entornos con cambios o incertidumbre de manera satisfactoria. En este proyecto estamos interesados, en particular, en unos de los sistemas de autoregulación presentes en los organismos adaptativos conocidos como mecanismos homeostáticos.

En ingeniería, uno de los pioneros trabajos en este ámbito se debe a William Ashby (autor del famoso libro “Design for a Brain” [1]), en el que se interesó por investigar cómo funcionan los mecanismos de adaptación en sistemas vivos. La tesis de Ashby fue que todos los organismos tienen ciertas variables esenciales, que deben mantenerse dentro de unos límites, y que a menudo están vinculadas entre sí, provocando que cambios en algunas de ellas puedan afectar a las demás. El estado de supervivencia del organismo ocurre cuando un comportamiento no mueve ninguna variable esencial fuera de sus límites. Al conjunto de mecanismos que permiten esta estabilidad se lo conoce como regulación homeostática. Ashby exploró estos mecanismos que permitían generar un comportamiento adaptativo autoinducido mediante el mantenimiento de la estabilidad interna de sus variables esenciales desde una perspectiva matemática y de sistemas dinámicos.

Aunque este tipo de mecanismos son de carácter fisiológico, y relacionan formas de automantenimiento del sistema con la estabilidad de ciertos niveles físico-químicos de carácter metabólico, trabajos recientes en neurociencia proponen extender estas nociones basadas en mecanismos reguladores homeostáticos de eficacia sináptica en el cerebro, como modos de estabilización de conducta. Estos mecanismos entre neuronas están impulsados por la necesidad de adaptarse a los cambios que dependen de una actividad, al tiempo que mantienen la estabilidad interna.

Dotar a un agente artificial de este tipo de propiedades de adaptación homeostática le permitiría adaptarse y resolver problemas para los que inicialmente no fue entrenado, permitiéndole alterar sus variables internas hasta obtener una estabilidad completa de

su sistema al mismo tiempo que ejecuta las tareas encomendadas.

1.1. Contexto y motivación

El proyecto se centra en el análisis del comportamiento de agentes con objetivos de fototaxis y propiedades adaptativas basadas en la plasticidad. Se probarán las hipótesis en entornos colectivos, donde más de un agente participa en una determinada tarea u objetivo.

Las capacidades sociales de los agentes se añadirán a los mismos siguiendo dos ideas distintas. La primera, defiende que el comportamiento social es una capacidad que, cuando surge, estructura por completo el “cerebro” de los agentes programados. La segunda defiende que el comportamiento social es una capacidad aditiva y que puede añadirse de manera independiente al individuo. Por tanto, se desarrollarán dos agentes, cada uno con las capacidades sociales añadidas siguiendo cada una de las anteriores ideas, para posteriormente someter a ambos a una prueba de la que se puedan obtener conclusiones que permitan comparar las dos aproximaciones.

1.2. Objetivos

En este trabajo de fin de grado se pretenden aplicar y analizar las ideas de ultrastabilidad de William Ashby en agentes artificiales. Dichos agentes serán diseñados mediante controladores basados en redes neuronales recurrentes a las que se les impondrá una regulación homeostática de su actividad sináptica (las neuronas tendrán funciones de activación adecuadas para generar estas estructuras de estabilidad). Los agentes serán evolucionados para solucionar un problema de fototaxis (búsqueda y acercamiento a una fuente de luz). Una vez programados los agentes artificiales homeostáticos se buscará analizar:

- Si los agentes pueden adaptarse a cambios en las reglas del entorno, aunque no hayan evolucionado específicamente para resolver nuevas tareas. Se buscará probar que, tal como ocurre en organismos vivos, estos mecanismos de adaptación facilitan implícitamente el mantenimiento de estabilidad y el desarrollo de nuevos comportamientos.
- Si en entornos sociales de agentes artificiales, donde los comportamientos para resolver problemas se basen en estrategias colectivas, los patrones de estabilidad son más robustos según programemos, tal y como hemos indicado anteriormente, las capacidades sociales como “aditivas” o “estructurales”. Para ello se someterán

las configuraciones obtenidas a estímulos ruidosos o perturbaciones, y mediremos el grado de estabilidad de los patrones homeostáticos que se generan.

1.3. Estructura

La memoria cuenta con un primer apartado de introducción, seguido por otro del estado del arte, en los que se explica el contexto del problema a resolver, los principales objetivos, motivaciones y situación actual del ámbito en el que se desarrolla el trabajo. A continuación se encuentra el apartado de diseño e implementación, en el cual se habla sobre las distintas herramientas y técnicas utilizadas para la realización del trabajo. Por último, en la sección de experimentación se enuncian y se analizan los dos procesos que se han ejecutado para la realización de los dos objetivos anteriormente descritos.

En los anexos se puede observar el tiempo invertido en este proyecto junto con las tareas las que se dedicó. Además, se incluye el código correspondiente en el lenguaje de programación Python desarrollado para la obtención de los resultados necesarios.

1.4. Planificación

En la planificación inicial del proyecto, se identificaron una serie de tareas que se han abordado en orden hasta alcanzar los objetivos finales. El proyecto comenzó (obviando el proceso inicial de aprendizaje y la introducción a la materia) con el diseño y la implementación de un Homeostato, con el fin de comprender de manera práctica los fundamentos de la homeostasis. La siguiente tarea fue la de diseñar e implementar un agente artificial básico, dotado con dos sensores, dos motores y un controlador neuronal (basado en una CTRNN) para comprobar el correcto funcionamiento de su comportamiento de fototaxis y sus mecanismos de plasticidad. Durante esta tarea también se implementó el algoritmo genético encargado de evolucionar a todos los agentes. Una vez con un agente individual completado se procedió al diseño e implementación de los agentes colectivos (“Agente 1” y “Agente 2”). Por último, se implementaron una serie de pruebas a las que someter a los agentes diseñados con el fin de estudiar su comportamiento, analizar los resultados y sacar las conclusiones pertinentes.

El cronograma del proyecto, dividido por semanas, junto con las horas dedicadas a cada tarea puede verse en el Anexo E.

Capítulo 2

Estado del arte

Hábitos

En nuestro comportamiento diario desplegamos numerosos patrones de conducta que se han ido fortaleciendo con la experiencia, de un modo continuado, y sostenidos por la repetición. Por ejemplo, mirar hacia la izquierda o hacia la derecha antes de cruzar la calle, atarse los cordones de los zapatos o simplemente caminar, pueden entenderse como patrones anidados de coordinación sensomotora, consolidados en nuestra conducta por una historia de refuerzos: A estos esquemas de conducta les llamamos hábitos.

Durante un tiempo, los hábitos fueron la piedra angular en psicología hasta el surgimiento del computador en la década de los sesenta del siglo pasado. Desde entonces, los modelos psicológicos que se extendieron y que funcionaron como inspiración entre los ingenieros que desarrollaban máquinas inteligentes, asumieron típicamente modelos internos de procesamiento simbólico-lingüístico, relegando el concepto de hábito al de un simple automatismo de estímulo-respuesta, lo que supone una simplificación muy alejada de la esencia y la génesis de los hábitos.

¿A qué o cómo definiríamos un hábito en sentido estricto? Podríamos decir que es un patrón automantenido de conducta. Decimos auto-mantenido porque la estabilidad de la conducta está acoplada con la estabilidad de los mecanismos que la generan. El hábito “llama” a realizar una conducta y la conducta refuerza el hábito.

En este trabajo, estamos interesados en analizar hábitos desde un punto de vista operativo para implementar agentes virtuales cuyo comportamiento se base precisamente en ellos. De modo que nuestras implementaciones deberán cumplir aspectos como que: (i) los hábitos no presupongan una prioridad causal entre percepción y acción, sino que integren a las dos; (ii) los hábitos sean plásticos y maleables (a diferencia de la rigidez que acompañan a las nociones de arco reflejo); (iii) los hábitos provean un sentido concreto de automantenimiento en el nivel de la conducta (es decir, sean formas de acción auto-reforzadas).

Agentes artificiales basados en hábitos: Operativización de la noción de homeostasis

Para modelar este tipo de hábitos en agentes artificiales que se desenvuelvan en entornos no conocidos y desarrollen tareas (es decir, para programar este tipo de patrones que se automantienen y fortalecen por la experiencia de practicarlos) nos interesaremos por las nociones que, en la década de los cuarenta, William Ross Ashby (1903-1972), desarrolló para operativizar e implementar en forma de artefacto, el concepto de homeostasis.

En general, la homeostasis es la capacidad de un organismo para mantener constante las propiedades físico-químicas de su medio interno. Es una habilidad que permite que su situación físico-química característica se conserve constante dentro de unos límites, aunque en el exterior existan fuentes que puedan ser motivo de alteración. Por ejemplo, el cuerpo humano, frente a fuentes de cambio externas, moviliza mecanismos de autorregulación (como el sistema nervioso central, sistema endocrino, sistema respiratorio, sistema circulatorio, etc.) para que se mantengan de forma constante sus condiciones de vida.

Aunque tradicionalmente es un término vinculado a la biología o la fisiología, otras ciencias y técnicas alejadas se han interesado por el fenómeno homeostático y han adoptado también este concepto a sus intereses. En su adopción técnica, la homeostasis es simplemente el rasgo de sistemas autorregulados que consiste en la capacidad para mantener ciertas variables en un estado estacionario, de equilibrio dinámico o dentro de ciertos límites, cambiando parámetros de su estructura interna. En este sentido, Ashby fue el primero que capturó operativamente estas ideas y construyó un homeostato artificial para demostrar las características de comportamiento de, lo que él llamó, un sistema ultraestable. El homeostato de Ashby respondía de manera aleatoria a las desviaciones de ciertos valores en sus parámetros esenciales y solo descansaba en su variación al azar cuando encontraba un comportamiento que mantenía los valores de esas variables críticas dentro de sus límites deseables. Las ideas de Ashby, desarrolladas en su famoso libro “Design for a Brain”, dieron lugar al campo de estudio de los sistemas biológicos como sistemas homeostáticos y adaptativos en términos puramente de matemática de sistemas dinámicos.

Estas sugerentes ideas permiten operativizar nociones de automantenimiento biológico y fisiológico y llevarlas a un entorno de implementación de agentes artificiales. Como recordamos, nuestro objetivo era tener esquemas ultraestables en el ámbito de la conducta de un agente virtual. ¿Cuáles y cómo podrían ser los componentes de una organización neuronal-conductual para poder hacer una analogía entre la vida orgánica y la conducta de un agente programado?

Si, a partir de conceptos como la homeostasis, la supervivencia de la vida orgánica puede definirse como una red auto-mantenida de reacciones químicas alejadas del equilibrio, la conducta podría definirse análogamente como una red auto-mantenida de hábitos o estructuras neurodinámicas. El objetivo es extender las nociones fisiológicas a estructuras neuronales. Y, en este segundo caso, la conservación adaptativa de organizaciones neurodinámicas (en lugar de organizaciones fisiológicas) no vendría por mecanismos de tipo biológico sino mediante regulación sensomotora asociada a la conducta del agente (en lugar de basada en esquemas metabólicos).

Controladores con plasticidad homeostática

Esta analogía vida-conducta, que podríamos denominar interesada en nociones de “homeostasis en sistemas nerviosos artificiales”, ha sido poco explorada[2, 3, 4, 5, 6]. Tomaremos la misma aproximación que otros investigadores han hecho y que se basa en el uso de redes recurrentes continuas para modelar el sustrato de un controlador neuronal sobre el que se incorporan mecanismos basados en plasticidad hebbiana para garantizar que regímenes de activación de las redes en su conjunto, tienen características homeostáticas que les permiten lograr condiciones de ultraestabilidad.

Esta “plasticidad homeostática” no es solo un mecanismo formal sino que tiene evidencia empírica [7], pues se ha observado que las neuronas buscan estabilidad en su frecuencia de disparo. El modo de conseguir estabilidad conductual, por tanto, parece exigir que los modos de estabilidad en la conducta aparezcan acoplados a los mecanismos de estabilidad de los parámetros sinápticos, de manera que cuando la estabilidad conductual se pierde induce inestabilidad sináptica hasta recuperar la estabilidad conductual.

El método para implementar plasticidad homeostática normalmente se apoya en los resultados de la teoría Hebbiana, que describe un mecanismo básico de plasticidad sináptica en el que el valor de una conexión sináptica se incrementa si las neuronas de ambos lados de dicha sinapsis se activan repetidas veces de forma simultánea. Introducida por Donald Hebb, en 1949, es también llamada “regla de Hebb” o “Teoría de la Asamblea Celular” y afirma que, en palabras del propio Hebb: «Cuando el axón de una célula A está lo suficientemente cerca como para excitar a una célula B y repetidamente toma parte en la activación, ocurren procesos de crecimiento o cambios metabólicos en una o ambas células de manera que tanto la eficiencia de la célula A, como la capacidad de excitación de la célula B son aumentadas».

De manera menos rigurosa, la teoría se resume a menudo como que “las células que se disparan juntas, permanecerán conectadas”, aunque esto es una simplificación y no debe tomarse literalmente, así como no representa con exactitud la declaración

original de Hebb sobre cambios de la fuerza de conectividad en las células. Sin embargo, ese principio es que se suele utilizar en el ámbito del Aprendizaje automático y de los modelos en Neurocomputación. La teoría es comúnmente evocada para modelar algunos tipos de aprendizajes asociativos en redes neuronales artificiales en los que la activación simultánea de las células conduce a un pronunciado aumento de la fuerza sináptica (conocido como aprendizaje de Hebb).

Controladores diseñados mediante computación evolutiva

Hasta ahora hemos visto que sería posible implementar agentes cuya conducta esté basada en hábitos a partir de mecanismos neuronales con plasticidad hebbiana que garantizarían condiciones de homoeostasis en la red como conjunto. Nos queda un último punto: seleccionar el tipo de red neuronal más adecuada para que puedan implementarse estos procesos y que suponga la base de un controlador de agentes virtuales. Para ello, nos fijamos en aspectos de la metodología denominada Computación Evolutiva, que se usa para el diseño de controladores neuronales evolutivos genéticamente determinados. Dichos controladores son aplicados a agentes (reales o virtuales) tipo Khepera para llevar a cabo tareas (por ejemplo, despliegue de trayectorias en navegación) dentro de un entorno no conocido previamente. Los controladores están basados en redes neuronales, que mediante evolución de parámetros con algoritmos genéticos, pueden ajustarse en forma on-line sin entrenamiento adicional a cambios del entorno, permitiendo un control apropiado de los este tipo de agentes robóticos. En los últimos años, en la mayoría de los experimentos de computación evolutiva son utilizados sistemas de control mediante redes neuronales con conexiones recurrentes, las cuales permiten que la red contemple aspectos de dinámica temporal[8].

La metodología empleada es la siguiente: una población de cromosomas artificiales es creada aleatoriamente y probada dentro del entorno del agente virtual. Concretamente, cada elemento de la población codifica el sistema de control de un robot. Cada robot es libre de actuar según al controlador utilizado, el cual fue genéticamente determinado, mientras es evaluado su desempeño (fitness) al realizar una tarea. Este proceso de generación de controladores robóticos y su evaluación en el entorno del robot, es llevado a cabo hasta satisfacer un criterio preestablecido (una fitness mínima que se supone aceptable por el programador) vinculado a la tarea a desarrollar.

En relación con la metodología mencionada, el desarrollo de tales controladores neuronales presupone la aceptación de algunos criterios, como los que a continuación se describen:

- Tipo de controlador neuronal: por lo general, el tipo de controlador utilizado es una red recurrente continua en el tiempo (Continuous Time Recurrent Neural Network. CTRNN, en su acrónimo en inglés). Estas son redes continuas cuyas neuronas están totalmente conectadas (se explotan los diversos bucles retroalimentados) con la peculiaridad matemática de que pueden aproximar cualquier sistema dinámico [9].
- Codificación genética de los controladores: se usan controladores genéticamente determinados (los pesos sinápticos son determinados genéticamente) y, en ocasiones, controladores con sinapsis adaptativas (los pesos sinápticos son siempre aleatorios y posteriormente modificados según reglas de adaptación de las sinapsis, las cuales son determinadas genéticamente).
- Modelo del robot: el tipo de robot a utilizar debe poseer un conjunto de sensores, los cuales permitan la navegación autónoma basada en ellos. En su mayoría, los trabajos presentados utilizan un agente virtual inspirado en el robot denominado Khepera.
- Tarea a realizar por el robot: diferentes clases de problemas pueden ser afrontados al momento de evaluar un controlador como el aquí expuesto. Principalmente, algunos de estos problemas se enmarcan en Aprendizaje por Refuerzo, análisis de las capacidades de adaptación a los cambios, estudio de la influencia de cambios en el entorno, navegación autónoma, etc.
- Hipótesis a probar: Existe un amplio espectro de áreas de investigación que pretenden ser estudiadas con el enfoque aquí presentado. Dichas áreas involucran desde aspectos ingenieriles hasta científicos[10], algunos de los cuales son: controladores bio-inspirados, modularidad y plasticidad neuronal, la perspectiva comportamental, aprendizaje, etc.

Capacidades sociales

El impacto que tiene lo social en nuestras capacidades individuales ha sido objeto de disputa y análisis a lo largo de la historia en el campo de la Ciencia Cognitiva. Por un lado podemos distinguir entre aquellos autores que creen que las capacidades individuales se ven afectadas por un nivel social de normas, reglas y convenciones culturales sin que estas capacidades individuales varíen en su naturaleza. En este sentido, el nivel social vendría a complementar las capacidades cognitivas de los individuos añadiendo un nivel cognitivo más que permitiría a los individuos organizarse, coordinarse y cooperar en el desarrollo de estrategias que satisfarían a todos a nivel

individual. En este marco, las capacidades individuales tales como percepción, acción o memoria no se verían afectadas por los cambios que puedan existir en las dinámicas sociales que se generen. Por otro lado, otros autores defienden que las capacidades individuales sí se verían afectadas por las dinámicas sociales. En esta propuesta, el aspecto no sólo supondría un nivel nuevo que permite a los individuos organizarse y obtener una recompensa individual, sino que ese mismo nivel y esas mismas actividades afectarían a las capacidades de los individuos.

La primera aproximación, por la cual lo social simplemente añade un nivel de complejidad mayor a lo que ya viene dado tras años de evolución y que permanece impenetrable a lo social, se conocería como aproximación aditiva [11]. La segunda aproximación afirmaría que las capacidades sociales transforían a las capacidades individuales, de tal modo que la socialización generaría un efecto *top-down* que cambia por completo la naturaleza de las capacidades individuales, lo que se conoce como una aproximación transformativa [11].

Existen muchas afirmaciones a favor y en contra de ambas aproximaciones pese a que los autores no utilicen explicitamente esta distinción. Por ejemplo, a favor del poder meramente aditivo de las capacidades sociales tenemos trabajos como los de Peters Carruthers [12]. Entre los defensores del poder transformativo de lo social tenemos a autores como John McDowell[13]. Es posible ver este debate entre el carácter aditivo vs. el carácter transformativo como un reflejo actualizado del debate tradicional que se mantuvo sobre el origen de capacidades mentales asociadas a naturaleza o cultura.

Conclusiones

En conclusión, recopilando lo visto hasta ahora: (i) nos interesa el estudio de comportamiento de agentes basados en hábitos y exige incorporar en nuestros modelos nociones de plasticidad homeostática; (ii) los modos de implementación se basarán en modelos de aprendizaje hebbiano y en redes recurrentes continuas como controladores de los agentes virtuales programados; (iii) metodológicamente, se diseñará una arquitectura neuronal con parámetros libres, se codificará genéticamente y se evolucionará en relación con la tarea a realizar, lo que implica definir una función de evaluación o función de fitness, la cual determinará el grado de aceptación de cada controlador neuronal en el proceso evolutivo.

Capítulo 3

Fundamentos teóricos

3.1. Redes neuronales recurrentes de tiempo continuo (CTRNN)

En contraste con las redes neuronales feed-forward, las cuales soportan únicamente comportamientos reactivos, en las Redes Neuronales Recurrentes de Tiempo Continuo (CTRNN) [14] pueden existir ciclos en su estructura y la activación de sus neuronas es asíncrona y multiescalada en el tiempo. Este tipo de redes neuronales también facilita describir el agente como un sistema dinámico acoplado al entorno en el que está ubicado, ya que está demostrado que son el modelo más simple de red neuronal dinámica continua no lineal [9] Además, la interpretación neurobiológica de las CTRNN ha sido demostrada y puede consultarse en [14].

Descripción matemática de una CTRNN

Las CTRNN están formadas por neuronas cuyo comportamiento se describe en las ecuaciones 3.1 y 3.2

$$\dot{y}_i = \frac{1}{\tau_i} * \left(-y_i + \sum_{j=1}^N w_{ji} * \sigma(y_j + \theta_j) + I_i \right) \quad i = 1, 2, \dots, N \quad (3.1)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$

donde y_i es el estado de la neurona, w_{ji} es el peso de la conexión entre las neuronas i y j, θ es el término bias, I representa una entrada externa y τ hace que cada una de las neuronas dependa del tiempo, ya que para diferentes valores la caída del nivel de activación de la neurona es más rápida o lenta. En la fórmula 3.1 la velocidad de actualización de la red neuronal debe ser notablemente mayor (el intervalo entre dos actualizaciones será menor) que el valor de τ para no obtener comportamientos no deseados.

Valores de activación y de salida de la neurona de una CTRNN

Para poder entender cómo deben interpretarse la activación y la salida de una CTRNN, se va a utilizar como ejemplo una CTRNN formada por una única neurona autoconectada como la de la figura 3.1. El valor de salida o de una neurona será un valor real entre 0 y 1 obtenido al aplicar la función sigmoide (ecuación 3.2) a la suma del estado actual y de la neurona con su valor bias θ , tal y como puede verse en la figura 3.1.

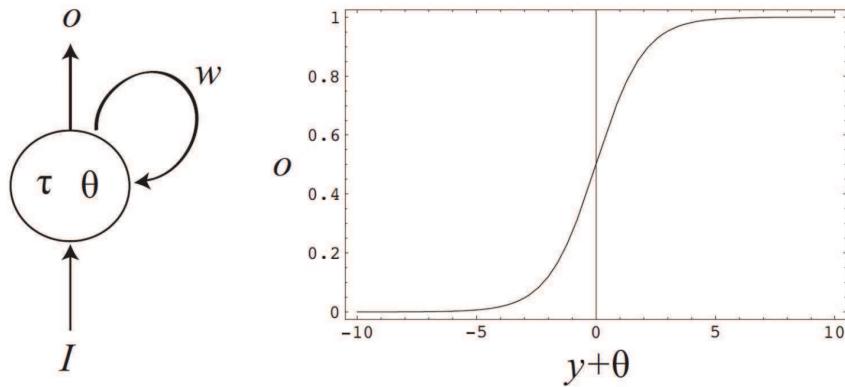


Figura 3.1: Valor de salida de una CTRNN. **Izquierda:** Un nodo autoconectado. **Derecha:** Función sigmoide aplicada para calcular la salida de una neurona.

En cuanto al valor de activación, a diferencia de una red neuronal feed-forward, la cual realiza un mapeo directo entre entrada y salida de la red, el comportamiento de una CTRNN corresponde al de un sistema dinámico [14], por lo que el valor de activación de la neurona convergerá a un punto de equilibrio. Para el análisis de las dinámicas del sistema formado por el agente controlado por la CTRNN y el entorno, se analizarán sus diagramas de bifurcación, los cuales muestran todos los puntos de equilibrio para la activación de las neuronas de la red.

3.2. Homeostasis

A pesar de que la homeostasis es una propiedad biológica, es posible representar un sistema lógico que se comporte de forma similar a como sistemas vivos con estas propiedades lo harían[15].

Un *Homeostato*, es un sistema artificial que presenta propiedades de homeostasis. El Homeostato desarrollado por Ashby era un aparato electromagnético mientras que el que va a ser utilizado es un componente implementado computacionalmente.

Un Homeostato es un sistema de N neuronas completamente interconectadas entre

si. Cada neurona recibe N entradas, de si misma y del resto de neuronas, dependientes del peso de la fuerza de conexión de las uniones entre ellas (ver Figura 3.2).

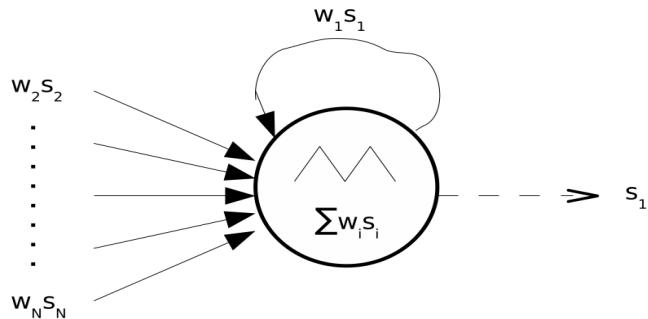


Figura 3.2: Esquema de un Homeostato individual. La unidad recibe las entradas del resto de unidades ($w_i s_i$) y de si misma ($w_1 s_1$). La salida s_1 es una función lineal por partes de la suma ponderada de las entradas.)

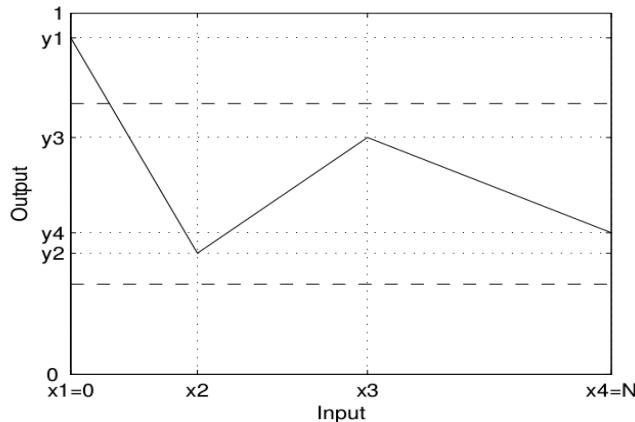


Figura 3.3: Ejemplo de función de transferencia de una unidad Homeostat. La función es lineal por partes con puntos en $(x_1, j_1), \dots, (x_p, y_p)$ donde $x_1 = 0$ y $x_p = N$, para todas las unidades en una red de N unidades. Las líneas discontinuas representan el rango homeostático de la red.

La suma ponderada I (ver ecuación 3.3) de las entradas de una unidad determina su salida s , tal y como está especificado por la función lineal de transferencia por partes F (ecuación 3.4 y figura 3.3).

$$I = \sum_i^N w_i s_i \quad (3.3)$$

$$s = F(I) = \begin{cases} y_1 + (y_2 - y_1)\left(\frac{I-x_1}{x_2-x_1}\right) & : x_1 \leq I < x_2 \\ y_2 + (y_3 - y_2)\left(\frac{I-x_2}{x_3-x_2}\right) & : x_2 \leq I < x_3 \\ y_3 + (y_4 - y_3)\left(\frac{I-x_3}{x_4-x_3}\right) & : x_3 \leq I \leq x_4 \end{cases} \quad (3.4)$$

Al inicializar la red, los pesos de las conexiones se aleatorizan a partir de una distribución uniforme de rango apropiado. El rango objetivo (o rango límite) $R = [0,5 - \delta, 0,5 + \delta]$ es especificado para la salida, donde δ determina la rigidez de la restricción homeostática. Si $s \in R$ la unidad es homeostática. Si $s \notin R$ significa que la homeostasis se ha perdido y se activan los mecanismos de cambio adaptativo.

Hay dos mecanismos de cambio adaptativo que se aplican a los parámetros de las unidades no homeostáticas. El primer mecanismo asigna valores aleatorios a los pesos de las conexiones aferentes a la unidad (ecuación 3.5). El segundo mecanismo asigna nuevos valores aleatorios a los parámetros que indican las coordenadas de la función de transferencia de la unidad (ecuación 3.6). Los rangos de estas reasignaciones son los mismos utilizados en la inicialización. Donde $rand(a, b)$ representa a la función que devuelve un número real aleatorio obtenido de una distribución uniforme en el rango $[a, b]$.

$$IF \quad (s \notin R) \quad THEN \quad [w = rand(0,00, 1,00) \quad \forall w \in \{w_1, \dots, w_N\}] \quad (3.5)$$

$$\begin{aligned} IF \quad (s \notin R) \quad THEN \quad &[x = rand(0,00, N) \quad \forall x \in \{x_2, \dots, x_{p-1}\}] \\ AND \quad &[y = rand(0,00, 1,00) \quad \forall y \in \{y_1, \dots, y_p\}] \end{aligned} \quad (3.6)$$

Durante la realización de este trabajo de diseño e implementó un sencillo Homeostato para comprender mejor los fundamentos que lo definen. La descripción de este Homeostato puede encontrarse en el Anexo A y puede servir como ejemplo sencillo para ayudar a comprender los fundamentos de la homeostasis.

3.3. Algoritmo genético

Los algoritmos genéticos son un método para resolver problemas de optimización. Estos algoritmos están basados en la selección natural, que es el proceso que dirige la evolución biológica. Un algoritmo genético modifica repetidas veces una población de posibles soluciones. En cada iteración, se seleccionan las mejores posibles soluciones encontradas hasta el momento y se genera a partir de ellas una nueva generación de candidatos. Generación tras generación la población evoluciona hacia la solución óptima.

Los algoritmos genéticos difieren de los algoritmos clásicos de optimización en tres puntos principales:

1. Los algoritmos clásicos generan un único punto con cada iteración mientras que los algoritmos genéticos generan una población de puntos con cada iteración.

2. En los algoritmos clásicos la secuencia de puntos se aproxima en cada iteración a la solución óptima mientras que en los algoritmos genéticos es el mejor punto de la población el que se aproxima a la solución óptima.
3. Los algoritmos clásicos generan el siguiente punto de la secuencia mediante un cálculo determinista mientras que los algoritmos genéticos selecciona el siguiente punto en la secuencia mediante cálculos que utiliza generadores de números aleatorios.

Una descripción más detalla de los algoritmos genéticos, tanto de su funcionamiento como de las partes que lo componen, puede encontrarse en el Anexo B.

Capítulo 4

Modelo e implementación

Para la realización de este trabajo, se han diseñado y desarrollado una serie de agentes a los que se les ha entrenado para la obtención de un comportamiento de fototaxis, es decir, de búsqueda y acercamiento a fuentes de luz.

Cada agente está modelado como si tuviera un cuerpo esférico con dos sensores en la parte frontal que permiten recibir lecturas de los diferentes estímulos presentes (ya sean luces u otros agentes) y dos motores opuestos que permiten que el agente pueda moverse libremente por el espacio, como puede verse en la figura 4.1. El agente tiene en todo momento una posición en el espacio dada por unas coordenadas (x, y) , así como una orientación respecto al eje X. Cada sensor está separado 60° del eje de orientación del agente. Los sensores tienen un arco de visión de 160° , para representar que el propio cuerpo del agente pueda encontrarse entre el sensor y la luz, evitando las lecturas.

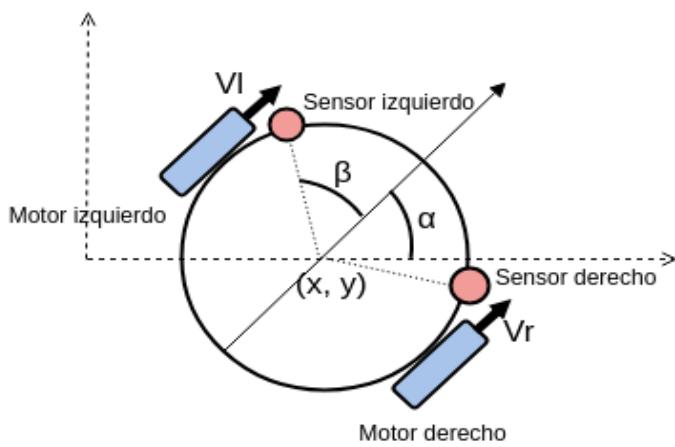


Figura 4.1: Esquema del agente utilizado.

El movimiento translacional del agente se calcula a partir de la velocidad lineal (v) de su centro de masa (ecuación 4.1). El movimiento angular del agente se calcula obteniendo la velocidad angular (ω) de su centro de masa (ecuación 4.2).

$$v = \frac{Vr + Vl}{2} \quad (4.1)$$

$$\omega = \frac{Vr - Vl}{2R} \quad (4.2)$$

La nueva posición del agente para cada ciclo de ejecución se calcula mediante las ecuaciones (4.3, 4.4 y 4.5), donde x e y representan la posición actual del agente, θ representa la orientación actual del agente y δt representa el tamaño del tiempo de integración de ciclo.

$$x(t+1) = x(t) + v.\cos(\theta).\delta t \quad (4.3)$$

$$y(t+1) = y(t) + v.\sin(\theta).\delta t \quad (4.4)$$

$$\delta(t+1) = \delta(t) + \omega.\delta t \quad (4.5)$$

Los sensores obtendrán en cada iteración, si tienen visual directo con la luz, lecturas sobre la intensidad que perciben de ella. La intensidad(I) que recibe cada sensor se calcula con la fórmula 4.6, donde la *Intensidad de la fuente* es la intensidad que emite la fuente de luz y D^2 es la distancia entre el sensor que realiza la lectura y la fuente de luz.

$$I = \frac{\text{Intensidad de la fuente}}{D^2} \quad (4.6)$$

Para los experimentos realizados en este trabajo, el radio del agente siempre se ha tomado como **4.0**.

4.0.1. Plasticidad

La ultraestabilidad de los agentes desarrollados se ha conseguido dotando a los mismos de mecanismos de plasticidad, que les permiten modificar los valores de algunos de sus atributos internos. En este caso, los agentes pueden alterar los valores de los pesos w_{ij} de las conexiones entre las neuronas de la red cuando la frecuencia de activación de una neurona es demasiado baja o demasiado alta.

La homeostasis no se encuentra programada explícitamente, pero aparece de forma implícita debido a los mecanismos de evaluación de la evolución genética que recompensan con mejor puntuación a aquellos agentes que, durante la realización del entrenamiento, han mantenido un mayor número de sus neuronas estables.

La plasticidad de cada conexión está gobernada por la actividad sináptica de la conexión junto con una regla de plasticidad codificada genéticamente. Estas reglas

de plasticidad vienen dadas por las ecuaciones (4.7, 4.8, 4.9 y 4.10), donde Δw_{ij} es el incremento por unidad de tiempo de un determinado peso sináptico (w_{ij}), y z_i y z_j son las frecuencias de activación de las neuronas presinápticas y postsinápticas respectivamente.

Las reglas de plasticidad se construyen en torno a cuatro parámetros: la frecuencia de aprendizaje (n_{ij}), un valor límite (z_{ij}^o), el grado de facilitación plástica local (p_j) y un factor de amortiguación lineal que restringe el cambio dentro de los límites establecidos para los valores de los pesos sinápticos (δ).

R0: Sin plasticidad:

$$\Delta w_{ij} = 0 \quad (4.7)$$

R1: Aprendizaje Hebbiano acotado:

$$\Delta w_{ij} = \delta n_{ij} p_j z_i z_j \quad (4.8)$$

R2: Potenciación o depresión amortiguadas de la neurona presináptica cuando la eficacia sináptica es muy alta o muy baja:

$$\Delta w_{ij} = \delta n_{ij} p_j (z_i - z_{ij}^o) z_j \quad (4.9)$$

R3: Potenciación o depresión amortiguadas de la neurona postsináptica cuando la eficacia sináptica es muy alta o muy baja:

$$\Delta w_{ij} = \delta n_{ij} p_j z_i (z_j - z_{ij}^o) \quad (4.10)$$

El grado de facilitación plástica local (p_j) se ve incrementado linealmente (hasta un valor máximo de 1) cuando la frecuencia de activación aumenta hasta salirse de sus límites, facilitando así los cambios plásticos. Cuando la frecuencia de activación entra en sus límites, p_j decrece linealmente (hasta un valor mínimo de -1), disminuyendo la facilidad de los cambios plásticos.

El cambio en la conexión sináptica depende del signo de n_{ij} , de forma que cada neurona puede actuar de manera independiente facilitando los cambios plásticos en la dirección indicada.

El factor de amortiguación lineal (δ) asegura que los valores de los pesos se mantienen dentro de los límites establecidos para ellos.

El factor de valor límite (z_{ij}^o) depende linealmente del valor actual del peso que se está actualizando plásticamente, por lo que es calculado en cada modificación.

Los pesos son actualizados cada iteración (si procede, dependiendo de su regla de plasticidad) a partir de la ecuación 4.11.

$$w_{ij}(t + 1) = w_{ij}(t) + \Delta w_{ij} \quad (4.11)$$

4.1. Agentes

Durante la realización de este trabajo se han diseñado e implementado tres tipos de agentes distintos. En esta sección se describe cada uno de ellos exponiendo la estructura de la red neuronal que actúa de controlador y las diferencias entre los controladores de cada agente.

4.1.1. Agente 0: agente sin comportamiento social

Este primer agente fue diseñado e implementado como punto de partida para probar tanto las funciones de evolución como el correcto funcionamiento de los sensores, motores, plasticidad y del comportamiento de fototaxis que sería utilizado posteriormente en los agentes con habilidades sociales.

El “Agente 0” no cuenta con habilidades sociales. Su comportamiento se centra únicamente en la fototaxis, por lo que se acercará a las luces y se mantendrá cerca de ellas.

Controlador

El controlador de este agente consiste en una CTRNN de 4 neuronas completamente interconectadas entre sí y autoconectadas (conectadas con todas las demás y con ellas mismas). Como puede verse en la figura 4.2, las neuronas 1 y 2 están conectadas a los sensores luminosos y son las encargadas de recibir las mediciones y procesarlas. Las neuronas 3 y 4 están conectadas a los motores, y sus salidas se convierten en la velocidad de este, que permite al agente moverse libremente.

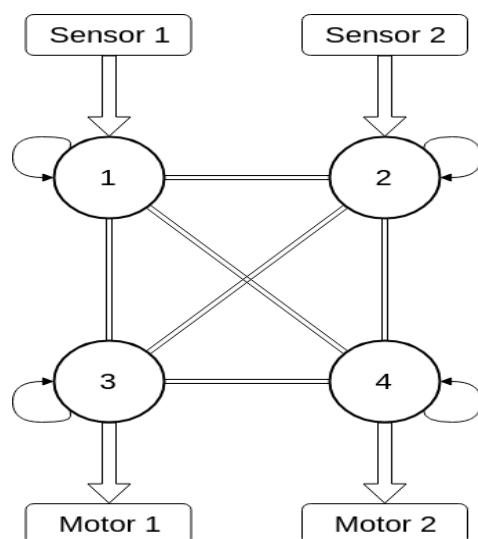


Figura 4.2: Esquema de la estructura del controlador del “Agente 0”.

4.1.2. Agente 1: agente colectivo aditivo

Este agente, al igual que el “Agente 0”, está diseñado para seguir un comportamiento de fototaxis, pero además cuenta con habilidades sociales para interactuar con el resto de agentes. Las capacidades sociales del agente se han diseñado e implementado asumiendo que las habilidades cerebrales de los seres vivos pueden interpretarse como módulos independientes que se entrenan por separado. Es decir, que el comportamiento social del agente se encuentra aislado del comportamiento individual del mismo, y sus evoluciones son por ello independientes.

El agente obtiene lecturas de los mismos sensores usados para la fototaxis, sobre las posiciones del resto de agentes que puede ver, pero estas lecturas son procesadas por una red independiente y por tanto no conectadas a la red de navegación.

Controlador

El controlador de este agente es similar al del “Agente 0”, una red CTRNN de 4 neuronas completamente interconectadas. Aunque al contar con habilidades sociales, se ha añadido una capa independiente de 2 neuronas que será la encargada de procesar las lecturas de los sensores sobre las posiciones de los agentes y de transmitir la señal adecuada a los motores del agente. Como puede verse en la figura 4.3, las neuronas 1 y 2 están conectadas a los sensores y se encargan de recibir mediciones de luz y de procesarlas. Las neuronas 5 y 6 están conectadas a los motores, igual que en el Agente 0. Las neuronas 3 y 4 se encargan de procesar las señales de los sensores sobre el posicionamiento del resto de agentes, procesarlas y transmitir su salida a los motores. Los motores juntan las salidas de las neuronas 3, 4, 5 y 6 para generar una velocidad.

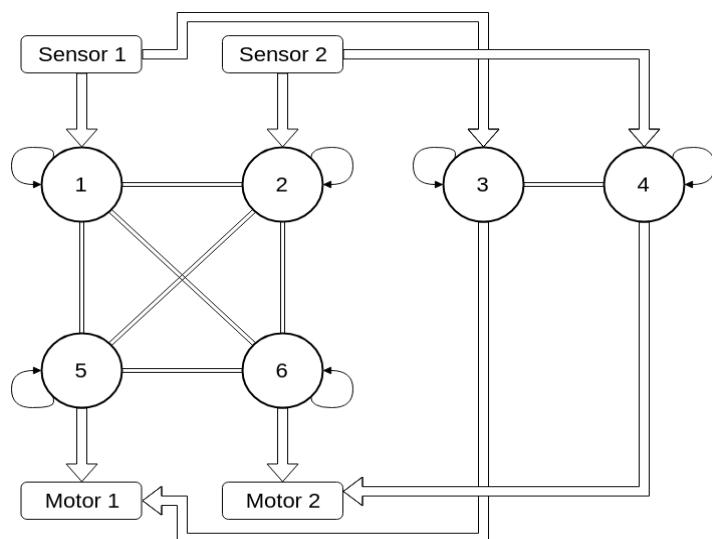


Figura 4.3: Esquema de la estructura del controlador del “Agente 1”.

4.1.3. Agente 2: agente colectivo estructural

El controlador de este agente es una extensión del controlador del “Agente 0”. A la red CTRNN de 4 neuronas se le han añadido 2 neuronas más, las cuales están conectadas a los sensores y se encargan de procesar las mediciones sobre el posicionamiento del resto de agentes. La diferencia con el “Agente 1” es que estas dos neuronas extra que están interconectadas completamente con el resto de la red, no son independientes. Esta implementación simularía que las capacidades cerebrales de los seres vivos puedes aprenderse y evolucionar en momentos diversos de su vida, pero esta evolución afecta al resto de neuronas haciéndolas cambiar también. Por tanto, en el “Agente 2”, el comportamiento social no se encuentra aislado del comportamiento de fototaxis.

Controlador

El controlador de este agente es similar al del “Agente 1”, salvo porque la cada extra de 2 neuronas no se encuentra aislada del resto, sino mezclada con ellas. Como puede verse en la figura 4.4, las neuronas 1 y 2 están conectadas a los sensores y se encargan de recibir mediciones de luz y de procesarlas. Las neuronas 3 y 4 están conectadas a los sensores y se encargan de procesar las señales de los sensores sobre el posicionamiento del resto de agentes, pero se encuentran completamente interconectadas con el resto de neuronas. Las neuronas 5 y 6 generan las salidas que los motores convierten en velocidades.

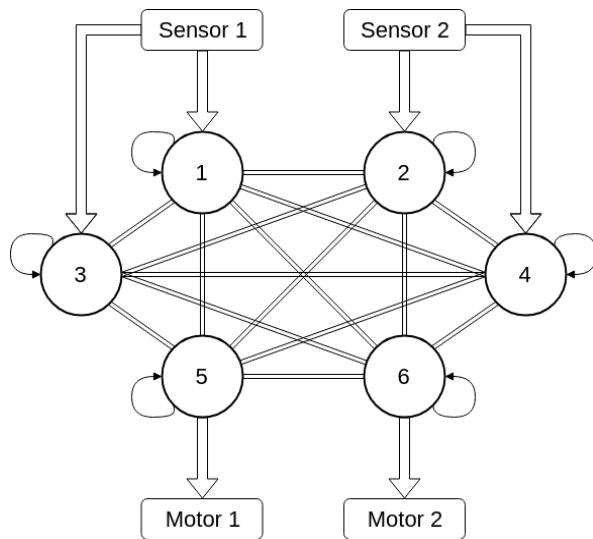


Figura 4.4: Esquema de la estructura del controlador del “Agente 1”.

4.2. Evolución

Para la evolución de los tres tipos de agentes anteriormente descritos se ha implementado un algoritmo genético. El algoritmo es común para los tres agentes, pero algunas de las funciones varían entre los tres tipos para adaptarse a sus estructuras, características y objetivos propios.

4.2.1. Codificación de los agentes

Cada agente candidato es codificado como un vector que contiene cada uno de sus atributos. El “Agente 0” se codifica como un vector de 46 componentes formado por 42 componentes reales y 4 componentes enteras, como puede verse en la figura 4.5.

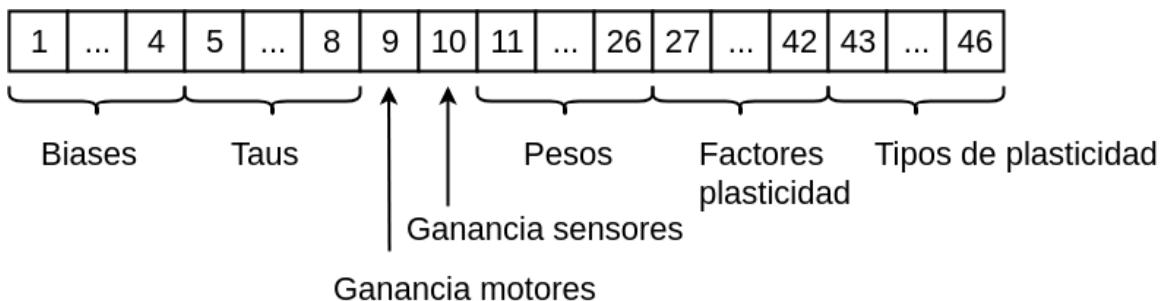


Figura 4.5: Representación codificada de un cromosoma que representa un agente de tipo 0.

Los agentes de tipo 1 y 2 están formados por 6 neuronas, a diferencia de los agentes de tipo 0 que están compuestos por 4 neuronas. Esto hace que el vector de un “Agente 1” o de un “Agente 2” este formado por 92 elementos, siendo 86 de ellos reales y 6 enteros, como puede verse en la figura 4.6

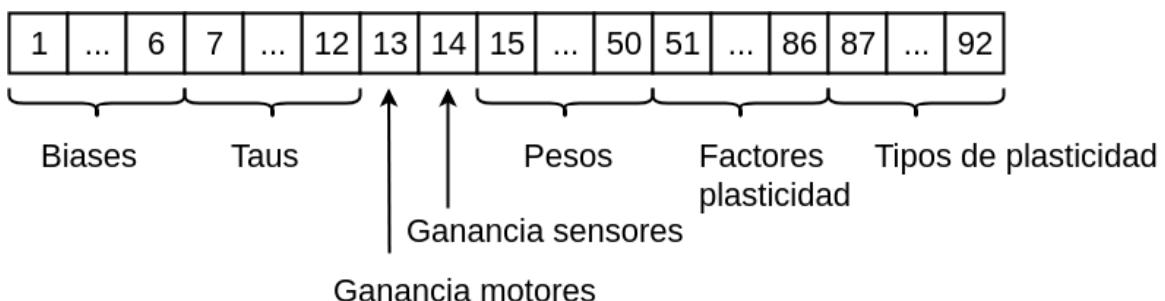


Figura 4.6: Representación codificada de un cromosoma que representa un agente de tipo 1 o 2.

Todas las componentes reales de los agentes codificados están formadas por valores entre [0, 1], los cuales serán escalados a sus rangos establecidos a la hora de evaluar el

agente. Las componentes enteras estan formadas por valores del 0 al 3, y representan los tipos de plasticidad nombrados en el apartado de plasticidad de este documento.

4.2.2. Creación de la población inicial

Para la creación de la población inicial se llama N veces a la función de creación de candidato. Esto da como resultado una población de candidatos aleatorios que constituyen la primera generación.

Se ha seleccionado un tamaño de población inicial (N) de **60** candidatos. Este tamaño se mantiene para el resto de las poblaciones de las siguientes generaciones.

4.2.3. Función de selección

La función de selección es la encargada de elegir los candidatos que serán utilizados para engendrar una nueva generación. Existen muchas formas de seleccionar estos candidatos (de manera aleatoria, los k mejores, los k peores, etc.). En este caso se ha elegido utilizar la llamada “selección por torneo”. Este tipo de selección consiste en elegir de manera aleatoria un número determinado de candidatos de entre la población inicial para “participar en el torneo”, el ganador del torneo es el candidato con mejor fitness de entre los participantes, el cual es seleccionado para engendrar la nueva generación. El torneo se repite hasta que se han elegido ganadores suficientes como para formar una nueva generación.

Se ha elegido este tipo de selección ya que, por una parte selecciona a aquellos candidatos de la generación actual que mejor fitness han conseguido (quedándose con los mejores), mientras que permite que candidatos menos buenos puedan ser también elegidos como ganadores (por ejemplo, si los participantes elegidos aleatoriamente son aquellos con peores valores de fitness). El permitir que no solamente los candidatos mejores engendren dota de cierta riqueza al algoritmo genético, permitiendo explorar un mayor abanico de posibilidades que, quedándonos solo con los mejores, no sería posible explorar.

En nuestro algoritmo genético, el número de participantes en los torneos se ha fijado en **10** candidatos.

4.2.4. Función de recombinación

La función de recombinación o *crossover* se encarga de crear nuevos individuos a partir de los candidatos seleccionados con la función de selección. En nuestro caso, esta recombinación se aplica con una probabilidad del **50 %**. En ella se eligen dos candidatos aleatorios de la nueva generación y se aplica el llamado “*crossover uniforme*”. En

este tipo de recombinación, para cada componente del vector de componentes, los dos candidatos elegidos intercambian sus valores con una cierta probabilidad. El resultado son dos nuevos candidatos formados por componentes de los elegidos.

En nuestro caso, la probabilidad de que los dos candidatos elegidos intercambien el valor de una componente es del **0.5**.

4.2.5. Función de mutación

La función de mutación se encarga de aplicar un cierto nivel de aleatoriedad en los candidatos, dando lugar a una mayor riqueza de individuos y una mayor amplitud de resultados posibles. En nuestro caso se ha optado por una mutación básica, la cual se da con una probabilidad del **50 %**. La mutación consiste en recorrer el vector de componentes del candidato elegido para mutar, generando un nuevo valor (dentro de sus rangos establecidos) aleatorio para cada componente con una cierta probabilidad.

En nuestro caso, la probabilidad de generar un nuevo valor para una componente real es del **0.5**, mientras que para una componente entera es del **0.1**.

4.2.6. Función fitness

La función fitness es la encargada de evaluar a los agentes candidatos, por lo que es de gran importancia tanto que la función represente bien el estado inicial y unos objetivos que alcanzar, como que la forma de puntuar positiva y negativamente a los candidatos según sus resultados sea la adecuada y refleje correctamente el grado de cumplimiento de los objetivos especificados. Una función fitness adecuada dará como resultado mejores candidatos y puede significar un tiempo de convergencia menor ante una función fitness menos adecuada.

En este proyecto ha sido necesario diseñar e implementar dos funciones fitness. La primera es utilizada por el “Agente 0” y se centra en desarrollar comportamientos de fototaxis individual. La segunda es utilizada por el “Agente 1” y el “Agente 2” y se centra en desarrollar comportamientos de fototaxis colectiva, es decir, teniendo en cuenta a más de un agente a la vez. Las puntuaciones fitness están acotadas entre 0 y 1.

Fitness individual

En un espacio de dos dimensiones, el agente candidato es situado inicialmente en el punto central (0, 0). Durante la ejecución, aparecen **6** luces hacia las que el agente debe aprender a acercarse. Sólo habrá una luz encendida a la vez, la cual permanecerá en el espacio durante un número de iteraciones elegidas aleatoriamente del intervalo

$[0,75T, 1,25T]$, teniendo $T = 1600$. Cuando una luz se apaga, se genera otra a una distancia $D \in [10R : 25R]$ de la posición actual del agente, siendo R el radio del agente. La intensidad (I) que emite la luz se elige en el momento de su creación aleatoriamente de manera que $I = [500, 1500]$.

Para el entrenamiento individual se busca recompensar tres comportamientos, los cuales se utilizarán para calcular la puntuación fitness de cada agente al finalizar la ejecución:

1. Que para cada luz, el agente acabe más cerca de ella de lo que empezó.
2. Que para cada luz, el agente permanezca el mayor número de iteraciones posibles a menos de una distancia a partir de la que se considera que el agente está cerca ($D_{ok} = R * 4$, siendo R el radio del agente).
3. Que para cada luz, el agente mantenga el mayor número de sus neuronas estables.

Por tanto, nuestra puntuación fitness estará formada por tres factores; Fd para el comportamiento de acercamiento, Fp para el comportamiento de permanencia cerca de la luz y Fh para el comportamiento de estabilidad. Para cada luz, en el instante que se apague se calculará la puntuación que el agente ha obtenido en ella mediante las fórmulas 4.12 (siendo D_f la distancia final y D_i la distancia inicial entre el agente y la luz), 4.13 y 4.14.

$$Fd = \begin{cases} 0,0 & \text{IF } D_f > D_i \\ 1 - (D_f/D_i) & \text{IF } D_f \leq D_i \end{cases} \quad (4.12)$$

$$Fp = \frac{\text{Nº ciclos cerca de la luz}}{\text{Nº de ciclos que la luz ha estado encendida}} \quad (4.13)$$

$$Fh = \frac{\text{Nº neuronas que no han perdido estabilidad}}{\text{Nº de neuronas del agente}} \quad (4.14)$$

La puntuación final del agente para esa luz será una suma ponderada de los factores, de forma que se da más peso a aquellos que se consideran más importantes, como puede verse en la ecuación 4.15.

$$\text{fitness} = (0,34Fd + 0,54Fp + 0,12Fh) \quad (4.15)$$

Una vez obtenida la puntuación fitness del agente para cada una de las 6 luces, la fitness total se corresponde con su media, siendo esa la puntuación final del agente.

Fitness colectiva

En la fitness colectiva, el funcionamiento del sistema de luces y los parámetros del mismo son iguales a los de la fitness individual. Sin embargo, algunos aspectos cambian. Al ser un experimento colectivo ya no tenemos un único agente, sino **5**. Al comienzo se sitúan a una determinada distancia del centro del espacio, que se selecciona de forma aleatoria del intervalo $[4R, 8R]$ siendo R el radio del agente, y con una orientación también aleatoria. A diferencia de la fitness individual, en la fitness colectiva las luces se generan a una distancia $D \in [10R, 25R]$ del centroide formado por los agentes, siendo R el radio del agente. Además, los agentes con comportamientos colectivos perciben al resto de agentes de forma similar a como perciben las luces (cada agente tiene una intensidad que en función de la distancia a la que se encuentre del sensor que lo lee producirá una señal más fuerte o más suave). La intensidad (I_{agente}) que emite cada agente es un valor fijo $I_{agente} = 750$.

En el entrenamiento colectivo se busca recompensar tres comportamientos, los cuales se utilizarán para calcular la puntuación fitness del grupo de agentes:

1. Que para cada luz, cada agente acabe más cerca de ella de lo que empezó.
2. Que para cada luz, se respete correctamente la restricción colectiva añadida.
3. Que para cada luz, cada agente mantenga el mayor número de sus neuronas estables.

Como podemos observar, el segundo comportamiento buscado es distinto al del entrenamiento individual. Este nuevo factor premiará que los agentes se mantengan cerca de la luz que se encuentre actualmente encendida el mayor número de ciclos posible, pero con una restricción colectiva añadida que consiste en que sólamente **3** agentes pueden estar cerca de una luz simultáneamente. Se utilizará un contador colectivo para llevar el control de este comportamiento, de manera que cuando un agente se encuentre cerca de la luz y el total de agentes cerca de esa luz sea menor que 3, aumentará el contador en una unidad. Sin embargo, si el número de agentes que se encuentran a la vez cerca de la luz excede del límite de 3, se considera que todo el conjunto de agentes ha fallado y se disminuirá el contador en 5 unidades (una unidad por cada agente del grupo). El máximo de puntuación que el conjunto de agentes puede obtener en el comportamiento colectivo para una luz es de $(3 * \text{Ciclos luz encendida})$, que sería el resultado de que durante todo el tiempo que la luz ha estado encendida, sólo 3 agentes hayan estado cerca.

Por tanto, nuestra puntuación fitness estará formada por tres factores; F_d para el comportamiento de acercamiento, F_p para el comportamiento colectivo y F_h para el

comportamiento de estabilidad. Al finalizar la última luz se calculará la puntuación que el conjunto de agentes ha obtenido mediante las fórmulas 4.16 (siendo D_{fk} la distancia final del agente k y D_{ik} la distancia inicial del agente k , entre el él y la luz), 4.17 y 4.18.

$$Fd = \frac{\sum_{k=1}^5 \left(\frac{\sum_{j=1}^{N^o Luces} Fd_{kj}}{N^o Luces} \right)}{5} \Rightarrow Fd_{kj} = \begin{cases} 0,0 & \text{IF } D_{fkj} > D_{ikj} \\ 1 - (D_{fkj}/D_{ikj}) & \text{IF } D_{fkj} \leq D_{ikj} \end{cases} \quad (4.16)$$

$$Fp = \frac{\sum_{i=1}^{N^o Luces} \text{Contador}_i}{(\sum_{i=1}^{N^o Luces} \text{Ciclos encendida}_i) * 3} \quad (4.17)$$

$$Fh = \frac{\sum_{i=1}^{N^o Luces} \frac{\sum_{k=1}^5 \left(\frac{N^o \text{ neuronas que no han perdido estabilidad}_{ki}}{N^o \text{ de neuronas del agente}_{ki}} \right)}{5}}{N^o Luces} \quad (4.18)$$

Una vez obtenidos estos valores, se ponderan y se obtiene la fitness total del conjunto de agentes (4.19).

$$\text{fitness} = (0,44Fd + 0,44Fp + 0,12Fh) \quad (4.19)$$

4.3. Simetría

Las redes CTRNN que actúan como controladores de los agentes cuentan con simetría en las ganancias de los motores y los sensores. Esto quiere decir que ambos motores comparten el mismo valor para la ganancia y que ambos sensores comparten el mismo valor para la ganancia.

4.4. Parámetros y rangos

Para todos los agentes anteriormente descritos, los rangos establecidos para sus variables internas pueden verse en la tabla 4.1.

Parametro	Descripción	Valor mínimo	Valor máximo
τ_i	Constante de decaimiento	0.4	4.0
θ_i	Bias	-3.0	3.0
$Gain$	Ganancia motora y sensora	0.1	10.0
w_{ij}	Peso de la conexión sináptica	-10.0	10.0
n_{ij}	Ritmo de plasticidad	-0.9	0.9

Tabla 4.1: Rango de valores para los parámetros de la red

Capítulo 5

Experimentos

En esta sección se recogen los experimentos a los que se han sometido los tres agentes introducidos en el capítulo anterior. Tras una explicación del experimento se analizarán los resultados obtenidos.

5.1. Experimento individual

La finalidad de este experimento fue la de comprobar que los mecanismos sensoriales, motores y de plasticidad homeostática del agente individual (o “Agente 0”) funcionaban correctamente. Este experimento fue un paso previo al desarrollo de los agentes con comportamientos colectivos. El agente sometido a este experimento fue el “Agente 0” que mejor puntuación obtuvo en el proceso de evolución mediante el algoritmo genético. Su puntuación fitness fue de 0.81.

El experimento consistía (de manera similar a la función fitness individual) en situar al agente en el punto $(0, 0)$ del espacio e ir mostrando de una en una 6 luces. No más de una luz puede estar encendida al mismo tiempo, la cual permanece en el espacio durante un número de iteraciones elegidas aleatoriamente del intervalo $[0,75T, 1,25T]$, teniendo $T = 1600$. Cuando una luz se apaga, se genera otra a una distancia $D \in [10R : 25R]$ de la posición actual del agente, siendo R el radio del agente. La intensidad (I) que emite la luz se elige en el momento de su creación aleatoriamente de manera que $I = [500, 1500]$.

Durante la ejecución del experimento se guardan las posiciones por las que el agente pasa durante la ejecución del mismo.

5.1.1. Resultados del Agente 0

En este experimento el resultado importante a analizar es la trayectoria efectuada por el agente durante la realización del mismo, con la finalidad de comprobar que su comportamiento de fototaxis es el adecuado. Como puede observarse en la figura 5.1,

el agente se desplaza a la luz activa en cada momento y se mantiene cerca de ella hasta que se apaga.

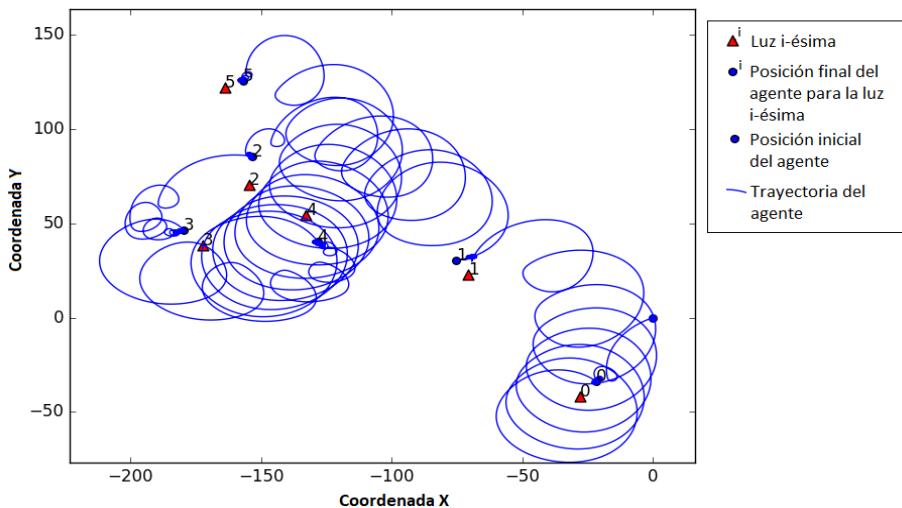


Figura 5.1: Trayectoria del “Agente 0” durante la realización del experimento individual.

Dado por comprobado el correcto funcionamiento del agente individual, se prorcedió a experimentar con los agentes colectivos.

5.2. Experimento colectivo

La finalidad de este experimento recae en comprobar, para los dos tipos de agentes colectivos (“Agentes 1” y “Agentes 2”) un correcto comportamiento de fototaxis con restricciones colectivas. El “Agente 1” que participa en el experimento es el que mayor puntuación fitness ha obtenido durante su evolución, la cual ha sido de 0.427 tras un periodo de 10 horas de evolución. En el caso del “Agente 2”, se ha alcanzado una fitness de 0.591 tras 10 horas de evolución. Para ambos tipos de agente, el experimento consta de tres partes:

En la primera parte, el experimento consiste (de manera similar a la función fitness colectiva) en situar al conjunto de 5 agentes del tipo que se quiere analizar a una determinada distancia del punto $(0, 0)$ del espacio, que se selecciona de forma aleatoria del intervalo $[4R, 8R]$ siendo R el radio del agente, y con una orientación también aleatoria. Una vez situados, de igual manera que en el experimento individual, 6 luces irían apareciendo de forma progresiva (teniendo sólamente una activada a la vez), sólo que esta vez en vez de aparecer a una determinada distancia del agente aparecen a una distancia $D \in [10R, 25R]$ del centroide formado por los agentes, siendo R el radio del agente. Además, los agentes con comportamientos colectivos perciben al resto de

agentes de forma similar a como perciben las luces (cada agente tiene una intensidad que en función de la distancia a la que se encuentre del sensor que lo lee producirá una señal más fuerte o más suave). La intensidad (I_{agente}) que emite cada agente es un valor fijo $I_{agente} = 750$.

Durante la ejecución de esta parte del experimento se guardan las trayectorias de los 5 agentes y los valores de las activaciones de las 6 neuronas de cada uno de los agentes durante cada iteración de la ejecución.

La segunda parte del experimento consiste en añadir cierto ruido a las señales de lectura de los sensores luminosos y colectivos, para analizar el impacto que este ruido tiene en la puntuación fitness del experimento. Por ello, para esta parte, se calcula la puntuación fitness de la ejecución de manera exactamente igual a la de la función fitness colectiva utilizada durante la evolución de los agentes.

El ruido R de cada señal S se calcula mediante la fórmula 5.1, donde α es el factor de ruido. Para este experimento se han analizado las variaciones en la fitness ante la presencia de ruido con unos valores α de 0, 0.2, 0.34 y 0.44.

$$R = \alpha S \quad (5.1)$$

La señal final S_f (posterior a la aplicación de ruido) de cada uno de los sensores se calcula mediante la fórmula 5.2, donde S representa la señal leída por los sensores y R representa el ruido aplicado a esa señal. El término $rand(a, b)$ representa un valor elegido aleatoriamente del rango $[a, b]$.

$$S_f = rand(S - R, S + R) \quad (5.2)$$

Para esta parte del experimento se han guardado las puntuaciones fitness obtenidas en 30 ejecuciones para cada nivel de ruido.

En la tercera parte del experimento, se añade ruido (de forma similar a la segunda parte), pero, esta vez, primero a los ritmos de plasticidad y después a los pesos de las conexiones sinápticas entre las neuronas, para comprobar la robustez de cada uno de los agentes frente a estas perturbaciones. Al igual que en el apartado dos, se tomarán 30 mediciones de la fitness resultante de la ejecución para cada nivel de ruido.

5.2.1. Resultados del Agente 1

En esta sección se presenta el análisis de los resultados del conjunto de agentes de tipo 1 en el experimento colectivo.

Trayectorias

El análisis de las trayectorias nos permite visualizar si el comportamiento de fototaxis colectivo del grupo de agentes del tipo 1 ha sido el correcto. Como puede verse en la figura 5.2, tres agentes (trayectorias de colores verde, azul y negro) han permanecido cerca de la fuente de luz número 4 mientras estaba activa, lo cual puede visualizarse observando las concentraciones de circunferencias presentes en las inmediaciones de la fuente de luz. Los otros dos agentes del conjunto (trayectorias cian y roja) no se han acercado a la luz, cumpliendo de manera satisfactoria la restricción colectiva establecida (máximo 3 agentes al mismo tiempo cerca de una luz).

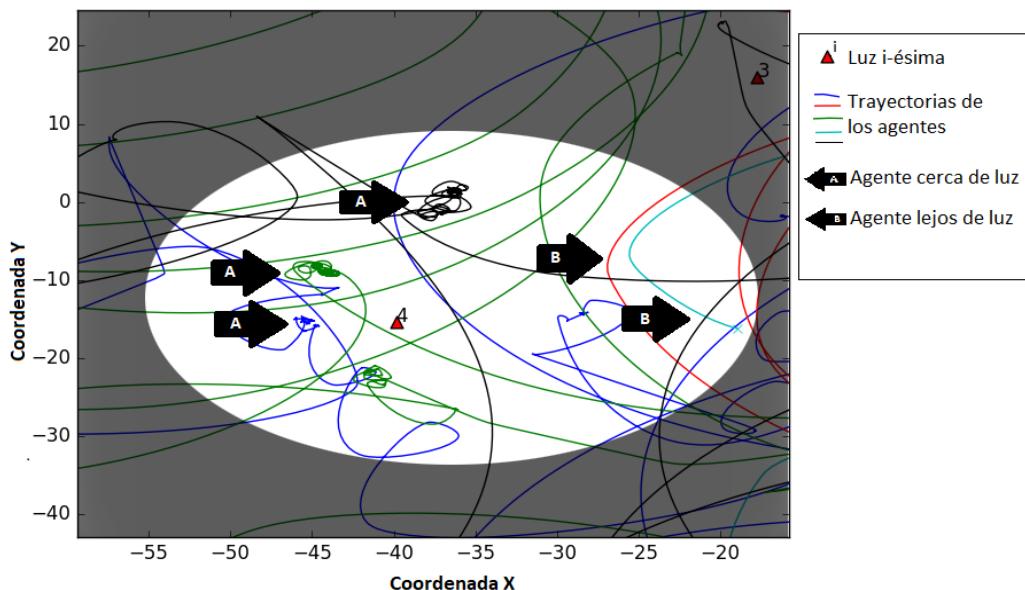


Figura 5.2: Fragmento de trayectorias del conjunto de agentes de tipo 1 que demuestran comportamiento de fototaxis colectiva.

Activaciones neuronales

Para analizar las activaciones neuronales del conjunto de agentes de tipo 1, se ha seleccionado un único agente como representante del conjunto. Las activaciones de las neuronas que componen el controlador de este agente pueden verse en la figura 5.3, cuyos valores se ven representados por la línea azul. Las dos líneas horizontales negras situadas en los valores del eje Y 4 y -4 representan el límite superior e inferior de la zona estable de la neurona.

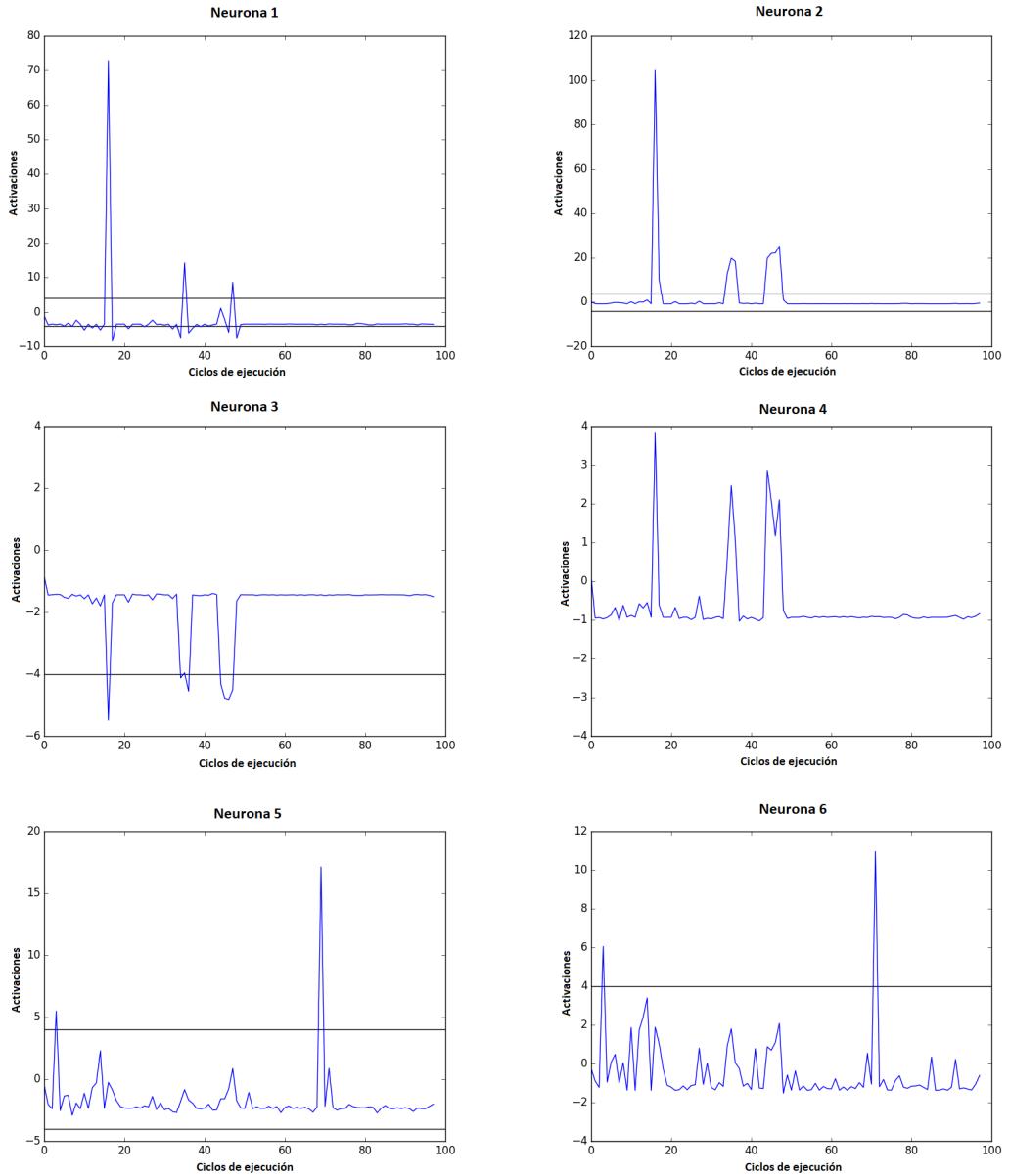


Figura 5.3: Activaciones de las 6 neuronas del agente representativo del grupo de agentes de tipo 1.

Se ha calculado, para cada neurona de cada uno de los agentes de tipo 1 que componen el conjunto de agentes utilizados para el experimento, el área (integrando) del espacio que las activaciones de la neurona han permanecido fuera de los límites, tanto superior como inferior, de la zona de estabilidad (líneas horizontales negras de las gráficas). En la tabla 5.1 se pueden ver los valores de estas áreas para cada neurona de cada agente.

Agente	Neuronas					
	1	2	3	4	5	6
1	2.7186	6.1104	61.468	480.64	1.089	0.0
2	5456.456	7588.078	34.91	33.783	36.65	0.0
3	43.742	190.657	53.7353	96.011	1.98	0.0
4	1909.514	2173.154	1.1268	98.664	34.29349	0.0
5	104.5142	218.7017	14.631	9.008	4.545	0.0

Tabla 5.1: Tabla con el área que cada neurona de cada agente del conjunto de agentes de tipo 1 ha permanecido fuera de los límites de estabilidad establecidos.

Se ha observado, que en este tipo de agentes las salidas de las activaciones neuronales de la zona estable son bastante frecuentes, pero de breve duración (el agente se recupera rápido).

La totalidad de las gráficas de las activaciones neuronales de los agentes del conjunto de agentes de tipo 1 se encuentra en el Anexo C.

Robustez ante ruido en los sensores

Como se ha explicado anteriormente, para cada nivel de ruido se han tomado 30 mediciones de los resultados fitness obtenidos en la ejecución del experimento. El objetivo es analizar la robustez (consistencia frente al ruido en los sensores) del conjunto de agentes. Las mediciones obtenidas se han representado en forma de diagramas de caja para visualizar su distribución (figura 5.4).

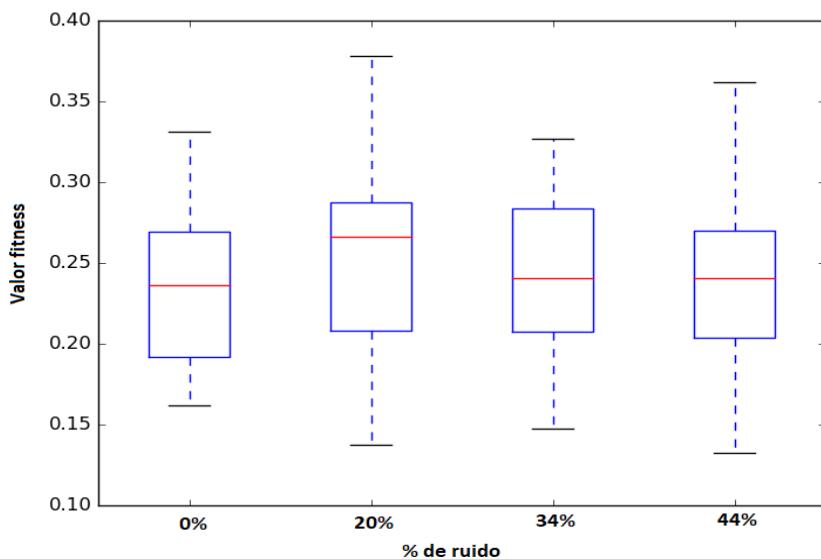


Figura 5.4: Diagramas de caja para las mediciones de la fitness en presencia de ruido en los sensores de los agentes de tipo 1.

Además, para aumentar la rigurosidad, se ha efectuado un test de *t de Student*

para cada par de población de mediciones, para comprobar si las medias de las dos poblaciones de datos examinadas (de tamaño pequeño y con distribución normal) pueden considerarse iguales. Los pares de poblaciones probados han sido: sin ruido y 0.2, sin ruido y 0.34, sin ruido y 0.44. Los resultados del test pueden verse en la tabla 5.2.

Test t-Student para Agente 1	
Población de mediciones comparadas	p-valor obtenido
Sin ruido - 0.2	0.075
Sin ruido - 0.34	0.39
Sin ruido - 0.44	0.54

Tabla 5.2: P-valores obtenidos de la ejecución del test *t de Student* sobre los pares de poblaciones de mediciones obtenidas.

Al obtener, para todos los pares de poblaciones, un *p-valor* mayor que 0,05 se confirma la hipótesis de que las medias pueden considerarse iguales y, por tanto, que el ruido en los sensores no afecta de manera significativa a la puntuación de los agentes de tipo 1.

Robustez ante ruido en parámetros de la plasticidad

Para comprobar la robustez de este tipo de agentes ante presencia de perturbaciones que afecten a la plasticidad se ha probado a añadir ruido a dos parámetros involucrados en los mecanismos de plasticidad. Primero, se ha aplicado ruido a los ritmos de plasticidad de las neuronas de los agentes, guardando los valores fitness resultantes de ejecutar el experimento con presencia de dicho ruido. Los niveles de ruido probados, al igual que en el caso de los sensores, han sido de 0, 0.2, 0.34 y 0.44. Para cada nivel de ruido se han medido 30 puntuaciones fitness. Estas mediciones se han representado en forma de diagramas de caja para visualizar su distribución (figura 5.5).

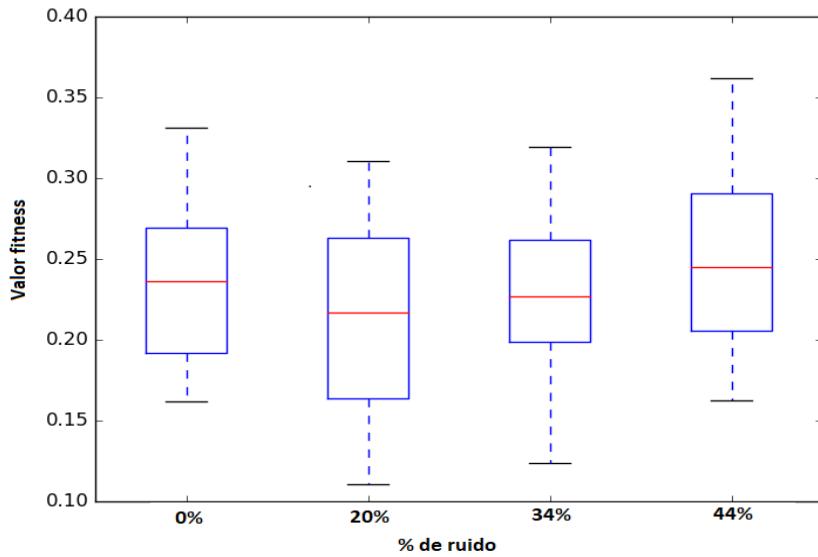


Figura 5.5: Diagramas de caja para las mediciones de la fitness en presencia de ruido en los ritmos de plasticidad de los agentes de tipo 1.

Además, para aumentar la rigurosidad, se ha efectuado un test de *t de Student* para cada par de población de mediciones, para comprobar si las medias de las dos poblaciones de datos examinadas (de tamaño pequeño y con distribución normal) pueden considerarse iguales. Los pares de poblaciones probados han sido: sin ruido y 0.2, sin ruido y 0.34, sin ruido y 0.44. Los resultados del test pueden verse en la tabla 5.3.

Test t-Student para Agente 1	
Población de mediciones comparadas	p-valor obtenido
Sin ruido - 0.2	0.15
Sin ruido - 0.34	0.52
Sin ruido - 0.44	0.25

Tabla 5.3: P-valores obtenidos de la ejecución del test *t de Student* sobre los pares de poblaciones de mediciones obtenidas.

Al obtener, para todos los pares de poblaciones, un *p-valor* mayor que 0,05 se confirma la hipótesis de que las medias pueden considerarse iguales y, por tanto, que el ruido en los ritmos de plasticidad no afecta de manera significativa a la puntuación de los agentes de tipo 1.

Por último, se ha aplicado ruido a los pesos de las conexiones sinápticas de las neuronas, durante el proceso de plasticidad, guardando los valores fitness resultantes de ejecutar el experimento con presencia de dicho ruido. Los niveles de ruido probados, de nuevo, han sido de 0, 0.2, 0.34 y 0.44. Para cada nivel de ruido se han medido 30

puntuaciones fitness. Estas mediciones se han representado en forma de diagramas de caja para visualizar su distribución (figura 5.6).

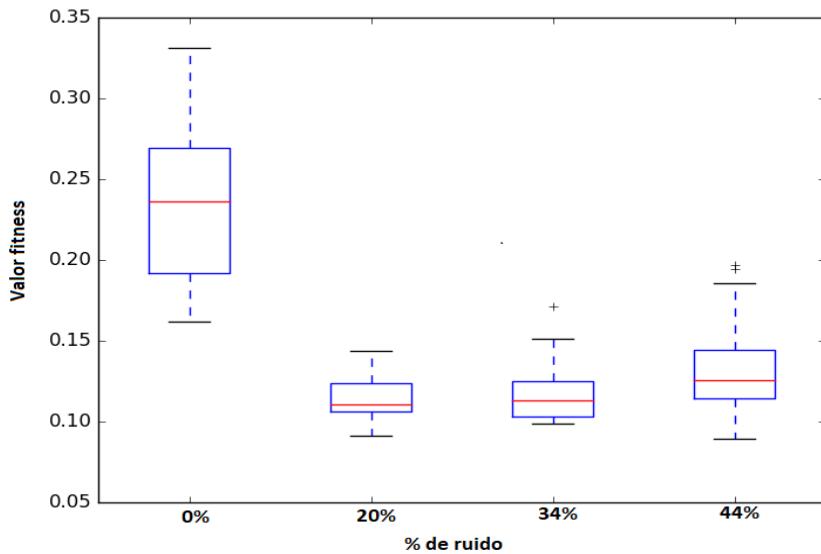


Figura 5.6: Diagramas de caja para las mediciones de la fitness en presencia de ruido en los pesos sinápticos de los agentes de tipo 1.

De nuevo, para aumentar la rigurosidad, se ha efectuado un test de *t de Student* para cada par de población de mediciones, con el fin de comprobar si las medias de las dos poblaciones de datos examinadas (de tamaño pequeño y con distribución normal) pueden considerarse iguales. Los pares de poblaciones probadas han sido: sin ruido y 0.2, sin ruido y 0.34, sin ruido y 0.44. Los resultados del test pueden verse en la tabla 5.4.

Test t-Student para Agente 1	
Población de mediciones comparadas	p-valor obtenido
Sin ruido - 0.2	0.0000000011
Sin ruido - 0.34	0.000000002
Sin ruido - 0.44	0.000001744

Tabla 5.4: P-valores obtenidos de la ejecución del test *t de Student* sobre los pares de poblaciones de mediciones obtenidas.

Al obtener, para todos los pares de poblaciones, un *p-valor* menor que 0,05 se rechaza la hipótesis de que las medias pueden considerarse iguales y por tanto, se asume que el ruido en los pesos sinápticos afecta de manera significativa a la puntuación de los agentes de tipo 1.

5.2.2. Resultados del Agente 2

En esta sección se presenta el análisis de los resultados del conjunto de agentes de tipo 2 en el experimento colectivo.

Trayectorias

El análisis de las trayectorias nos permite visualizar si el comportamiento de fototaxis colectivo del grupo de agentes de tipo 2 ha sido el correcto. Como puede verse en la figura 5.7, tres agentes (trayectorias de colores verde, azul y rojo) han permanecido cerca de la fuente de luz número 5 mientras estaba activa, lo cual puede visualizarse observando las concentraciones de circunferencias cerradas presentes en las inmediaciones de la fuente de luz. Los otros dos agentes del conjunto (trayectorias cian y negra) no se han acercado a la luz, cumpliendo de manera satisfactoria la restricción colectiva establecida (máximo 3 agentes al mismo tiempo cerca de una luz).

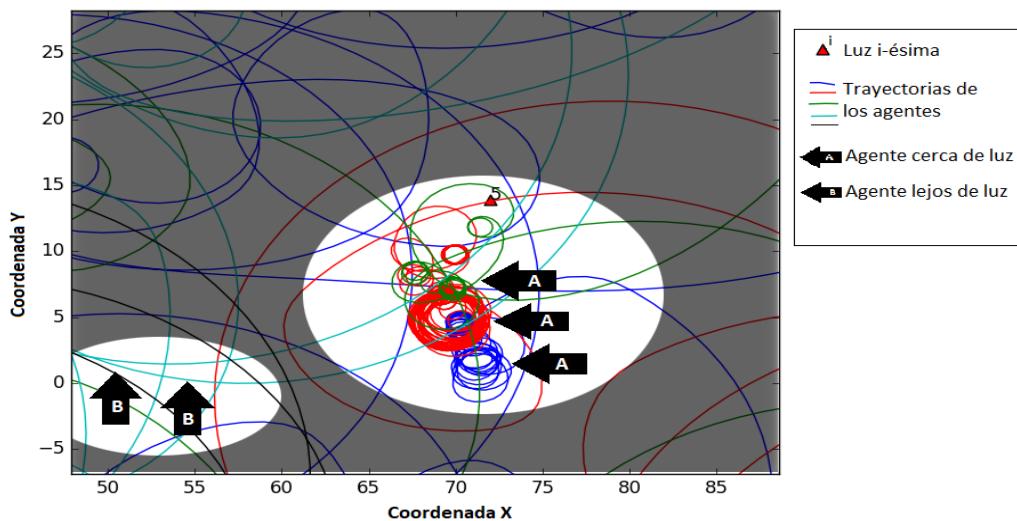


Figura 5.7: Fragmento de trayectorias del conjunto de agentes de tipo 2 que demuestran comportamiento de fototaxis colectiva.

Activaciones neuronales

Para analizar las activaciones del conjunto de agentes de tipo 2, se ha seleccionado un único agente como representante del conjunto. Las activaciones de las neuronas que componen el controlador de este agente pueden verse en la figura 5.8, cuyos valores se ven representados por la línea azul. Las dos líneas horizontales negras situadas en los valores del eje Y 4 y -4 representan el límite superior e inferior de la zona estable de la neurona.

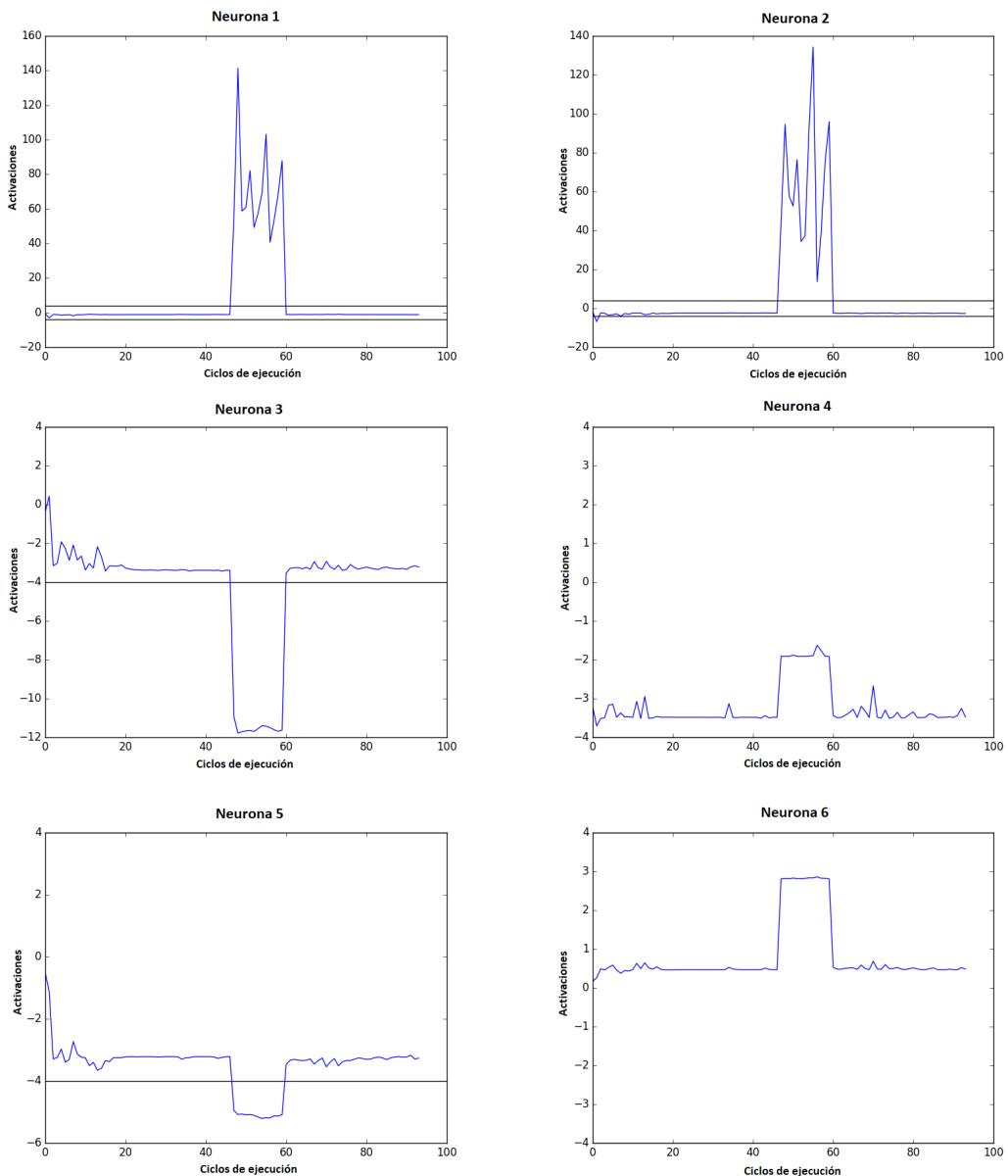


Figura 5.8: Activaciones de las 6 neuronas del agente representativo del grupo de agentes de tipo 2.

Se ha calculado, para cada neurona de cada uno de los agentes de tipo 2 que componen el conjunto de agentes utilizados para el experimento, el área (integrando) del espacio que las activaciones de la neurona han permanecido fuera de los límites, tanto superior como inferior, de la zona de estabilidad (líneas horizontales negras de las gráficas). En la tabla 5.5 se pueden ver los valores de estas áreas para cada neurona de cada agente.

Agente	Neuronas					
	1	2	3	4	5	6
1	875.9847	799.4	98.248	0.0	14.455	0.0
2	1291.039	828.222	31644.64	5919.85	109.994	0.0
3	0.0	5.1314	0.0	0.0	0.3	0.0
4	7119.155	19108.05	7203.446	7052.342	98.927	0.0
5	5770.375	4217.139	6793.95	26173.23	50.245	0.0

Tabla 5.5: Tabla con el área que cada neurona de cada agente del conjunto de agentes de tipo 2 ha permanecido fuera de los límites de estabilidad establecidos.

Se ha observado, que en este tipo de agentes las salidas de las activaciones de la zona estable son poco frecuentes, pero suelen tener una duración elevada (el agente tarda en recuperarse).

La totalidad de las gráficas de las activaciones de las neuronas del conjunto de agentes de tipo 2 se encuentra en el Anexo D.

Robustez ante ruido en los sensores

Como se ha explicado anteriormente, para cada nivel de ruido se han tomado 30 mediciones de los resultados fitness obtenidos en la ejecución del experimento. El objetivo es analizar la robustez (consistencia frente al ruido) del conjunto de agentes. Las mediciones obtenidas se han representado en forma de diagramas de caja para visualizar su distribución (figura 5.9).

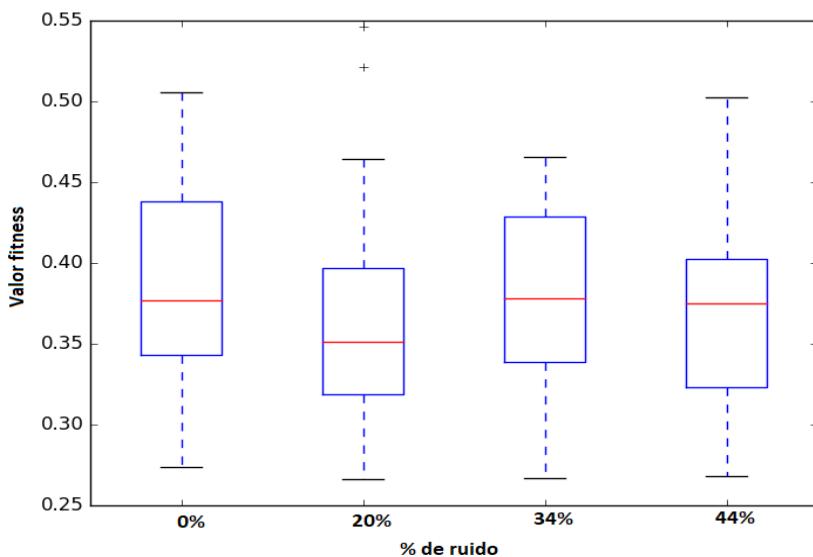


Figura 5.9: Diagramas de caja para las mediciones de la fitness en presencia de ruido en los sensores de los agentes de tipo 2.

Además, para aumentar la rigurosidad, se ha efectuado un test de *t de Student*

para cada par de población de mediciones, con el fin de comprobar si las medias de las dos poblaciones de datos examinadas (de tamaño pequeño y con distribución normal) pueden considerarse iguales. Los pares de poblaciones probados han sido: sin ruido y 0.2, sin ruido y 0.34, sin ruido y 0.44. Los resultados del test pueden verse en la tabla 5.6.

Test t-Student para Agente 2	
Población de mediciones comparadas	p-valor obtenido
Sin ruido - 0.2	0.17
Sin ruido - 0.34	0.56
Sin ruido - 0.44	0.31

Tabla 5.6: P-valores obtenidos de la ejecución del test *t de Student* sobre los pares de poblaciones de mediciones obtenidas.

Al obtener, para todos los pares de poblaciones, un *p-valor* mayor que 0,05 se confirma la hipótesis de que las medias pueden considerarse iguales y por tanto, que el ruido en los sensores no afecta de manera significativa a la puntuación de los agentes de tipo 2.

Robustez ante ruido en parámetros de la plasticidad

Para comprobar la robustez de este tipo de agentes ante presencia de perturbaciones que afecten a la plasticidad se ha probado a añadir ruido a dos parámetros involucrados en los mecanismos de plasticidad. Primero, se ha aplicado ruido a los ritmos de plasticidad de las neuronas de los agentes, guardando los valores fitness resultantes de ejecutar el experimento con presencia de dicho ruido. Los niveles de ruido probados, al igual que en el caso de los sensores, han sido de 0, 0.2, 0.34 y 0.44. Para cada nivel de ruido se han medido 30 puntuaciones fitness. Estas mediciones se han representado en forma de diagramas de caja para visualizar su distribución (figura 5.10).

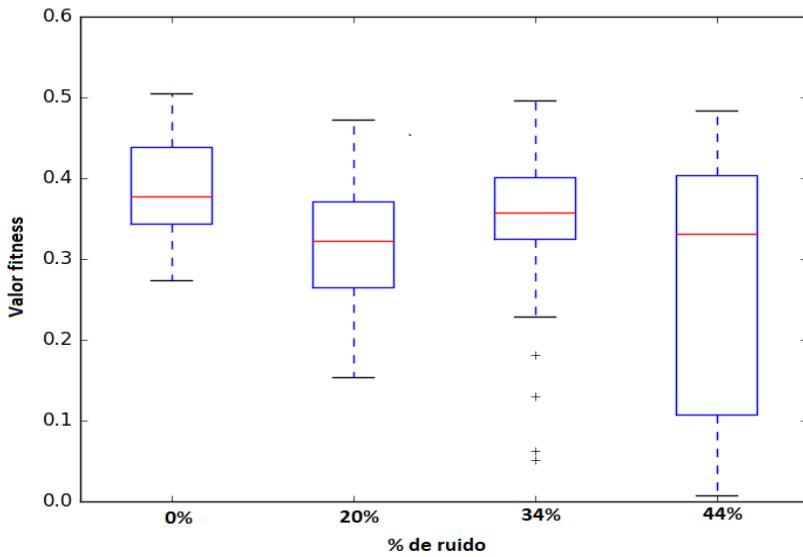


Figura 5.10: Diagramas de caja para las mediciones de la fitness en presencia de ruido en los ritmos de plasticidad de los agentes de tipo 2.

Además, para aumentar la rigurosidad, se ha efectuado un test de *t de Student* para cada par de población de mediciones, con la finalidad de comprobar si las medias de las dos poblaciones de datos examinadas (de tamaño pequeño y con distribución normal) pueden considerarse iguales. Los pares de poblaciones probados han sido: sin ruido y 0.2, sin ruido y 0.34, sin ruido y 0.44. Los resultados del test pueden verse en la tabla 5.7.

Test t-Student para Agente 2	
Población de mediciones comparadas	p-valor obtenido
Sin ruido - 0.2	0.001
Sin ruido - 0.34	0.0347
Sin ruido - 0.44	0.00041

Tabla 5.7: P-valores obtenidos de la ejecución del test *t de Student* sobre los pares de poblaciones de mediciones obtenidas.

Al obtener, para todos los pares de poblaciones, un *p-valor* menor que 0,05 se rechaza la hipótesis de que las medias pueden considerarse iguales y por tanto, se asume que el ruido en los ritmos de plasticidad afecta negativamente a la puntuación fitness para los agentes de tipo 2.

Por último, se ha aplicado ruido a los pesos de las conexiones sinápticas de las neuronas, durante el proceso de plasticidad, guardando los valores fitness resultantes de ejecutar el experimento con presencia de dicho ruido. Los niveles de ruido probados, de nuevo, han sido de 0, 0.2, 0.34 y 0.44. Para cada nivel de ruido se han medido 30

puntuaciones fitness. Estas mediciones se han representado en forma de diagramas de caja para visualizar su distribución (figura 5.11).

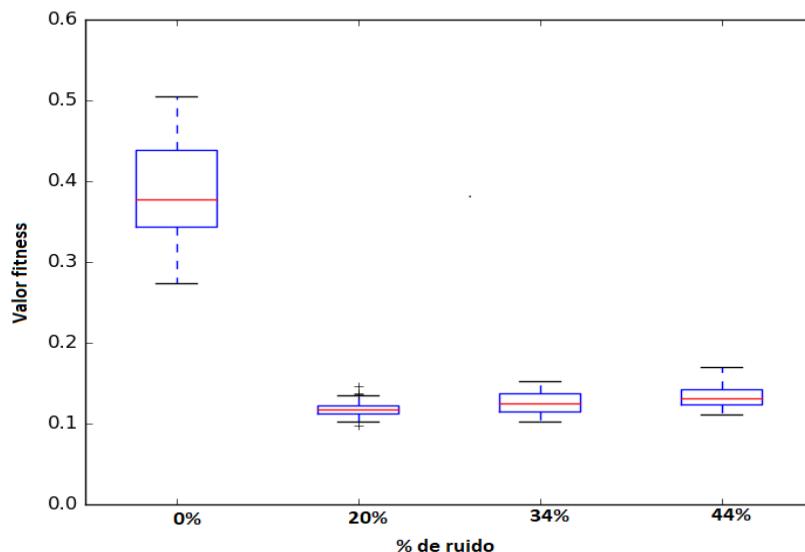


Figura 5.11: Diagramas de caja para las mediciones de la fitness en presencia de ruido en los pesos sinápticos de los agentes de tipo 2.

De nuevo, para aumentar la rigurosidad, se ha efectuado un test de *t de Student* para cada par de población de mediciones, para comprobar si las medias de las dos poblaciones de datos examinadas (de tamaño pequeño y con distribución normal) pueden considerarse iguales. Los pares de poblaciones probados han sido: sin ruido y 0.2, sin ruido y 0.34, sin ruido y 0.44. Los resultados del test pueden verse en la tabla 5.8.

Test t-Student para Agente 2	
Población de mediciones comparadas	p-valor obtenido
Sin ruido - 0.2	~ 0.0
Sin ruido - 0.34	~ 0.0
Sin ruido - 0.44	~ 0.0

Tabla 5.8: P-valores obtenidos de la ejecución del test *t de Student* sobre los pares de poblaciones de mediciones obtenidas.

Al obtener, para todos los pares de poblaciones, un *p-valor* menor que 0,05 se rechaza la hipótesis de que las medias pueden considerarse iguales y por tanto, se asume que el ruido en los pesos sinápticos afecta de manera significativa a la puntuación de los agentes de tipo 2.

Capítulo 6

Conclusiones

Este apartado se centrará en: (1) resumir cuáles han sido los resultados más relevantes tras los experimentos realizados, procurando ponerlos en relación entre sí, y (2) exponer algunas reflexiones sobre el carácter del modelado cognitivo y sobre las posibles líneas futuras que se podrían emprender para seguir avanzando en el tema del proyecto.

Como se avanzó en la primera parte de esta memoria, existe actualmente un debate sobre qué capacidades cognitivas permiten a individuos u organismos participar en acciones conjuntas. Generalmente, se ha supuesto que esta no es más que una nueva capacidad que surge en la evolución o en el desarrollo, que se agrega a capacidades ya existentes. Por ejemplo, autores como Bratman [16], proponen la necesidad de una nueva capacidad dedicada a planificar y coordinar correctamente sus acciones con otros; Tomasello [17], defiende la aparición, hacia el final del primer año de vida, de una nueva capacidad con la potencialidad de simular mentalmente las motivaciones de otros sujetos del grupo, imprescindible para poder coordinarse conjuntamente. Aunque sus propuestas son diferentes, lo que ambas sugieren es que las capacidades colectivas se agregan a otras capacidades preexistentes del individuo y se ejercen siempre que el individuo participe en actividades de cooperación.

Más recientemente, han comenzado a surgir evidencias en el sentido contrario: las capacidades colectivas son elementos transformativos de todas las capacidades que los agentes ya tenían de manera individual [11]. Es decir, es algo más que una capacidad específica que un individuo actualiza cuando coopera con otros y que incluso está presente fuera de las actividades conjuntas con otros. Por ejemplo, muchas de las habilidades individuales que aprendemos en la infancia, lo hacemos por imitación. Aunque la habilidad sea propia del sujeto, en su base no deja de ser una capacidad compartida y guiada por el ejemplo de otros.

Una vez analizados los resultados de los experimentos las conclusiones que pueden obtenerse comparando a los agentes de tipo 1 y de tipo 2 son, las siguientes.

Comportamientos o conducta externa

1. Los mecanismos de doble circuito planteados hacen que, estructuralmente, los agentes de tipo 1 tengan menor eficacia en la consecución de sus tareas (su fitness es sustancialmente inferior (0.44) en relación con los agentes de tipo 2 (0.59). Los primeros, son agentes con una nueva capacidad, pero esta debe combinarse con otras y, en esa combinación, el éxito no es muy elevado.
2. Por otro lado, si nos fijamos en sus trayectorias, pueden observarse que tienen un carácter más errático, posiblemente un rasgo relacionado con el punto anterior. Es, en función de estos resultados, que podemos decir que el Agente 1, en lo que se refiere a su dimensión de comportamiento, presenta algunos rasgos comparativamente peores en relación con el Agente 2.
3. Finalmente, como medida para cuantificar la robustez de ambas conductas, se han realizado experimentos en los que sometemos a los agentes a una señal ruidosa en sus sensores y vemos el efecto en la fitness global. En este apartado, podemos observar que ambos agentes presentan resultados parecidos.

A la vista de estos resultados, podríamos afirmar que, al menos a un nivel de modelado mínimo, las soluciones que consideran las capacidades sociales como estructurales derivan en agentes más robustos.

Estructura o robustez interna

Una vez analizados algunos de los indicadores de la conducta externa de los agentes, nos interesamos por resultados de robustez interna que llevaremos a cabo con medidas de estabilidad en relación con aleatorizar valores en parámetros internos del agente.

1. Las perturbaciones en los pesos sinápticos afectan significativamente al rendimiento de los dos tipos de agente. Es decir, las estructuras neuronales que se obtienen por evolución son sensibles a cambios. En ambos, al cambiar levemente los pesos, su comportamiento se ve afectado.
2. Si nos centramos en los patrones de activación neuronal en relación con sus límites homeostáticos, observamos que los agentes de tipo 1 pierden estabilidad con más frecuencia (sus patrones son más irregulares) al compararlos con agentes del tipo 2 (con patrones más constantes, incluso en la zona fuera de su límite homeostático). Además, los segundos, aunque tardan más en volver a la estabilidad que los del tipo 1, se mantienen posteriormente estables más tiempo (en promedio, solo salen una única ocasión de la zona homeostática).

3. Paradójicamente, las perturbaciones en los ritmos de plasticidad afectan al rendimiento del Agente 2, pero no del Agente 1. Es decir, aunque a nivel comportamental despliega características más robustas, mantiene un nivel más crítico en su estructura homeostática.

Aunque este es un debate proveniente del ámbito de la Psicología o Estudios del desarrollo cognitivo, en este trabajo nos hemos planteado abordarlo desde el punto de vista del modelado cognitivo mínimo [18]. La cognición mínima busca modelos que, con mínimos recursos, sean capaces de materializar capacidades cognitivas primitivas o los orígenes de capacidades cognitivas más elaboradas. En la última década, la relevancia en la comprensión de fenómenos cognitivos a partir de modelos mínimos ha ido en aumento. Concentrarse en una versión simplificada, mínima, de un fenómeno de estudio, sirve al investigador como herramienta para pensar sobre el fenómeno. Por ejemplo, algunos modelos deliberadamente simples pueden confirmar que existen alternativas explicativas a fenómenos que se suponía que funcionaban de una determinada manera. En palabras de autores como Rohde [19], los modelos sirven como “gimnasia mental”, es decir, son entidades equivalentes a los experimentos mentales clásicos, son artefactos que nos ayudan a pensar. En el área del estudio ingenieril de la inteligencia, podemos encontrar dos motivaciones: (i) La primera es puramente técnica. Trata de encontrar algoritmos útiles en campos tecnológicos, que tienen una amplia gama de beneficios. (ii) La segunda motivación es científica, esto es, se usan computadores, modelos de simulación, bots, etc. como plataformas experimentales para la investigación de cuestiones acerca de la inteligencia. Esta es la conocida como “metodología sintética” que se presenta como alternativa a la tradicional, y que defiende que existe espacio para un estudio ingenieril sobre las capacidades cognitivas que debería funcionar más como un campo científico que permita examinar hipótesis particulares, y en el que haya una investigación teórica para averiguar por qué ciertos modelos son mejores o no en términos de las asunciones que suponen. Los modelos mínimos consisten en agentes sintéticos que habitan entornos virtuales y con ellos se pretende capturar aspectos concretos de una tarea cognitiva. Su condición de “mínimos” permite que puedan ser estudiados de manera completa. Estos modelos no son estructuras que organizan datos experimentales. Su valor proviene del valor científico que tiene el estudio de los patrones observados y la relación de tales patrones con hipótesis teóricas. Sirven como herramientas para cuestionar concepciones sobre como un cierto comportamiento es generado. Son modelos con una complejidad manejable que permiten un análisis completo y muchas veces analítico.

Capítulo 7

Conclusiones personales

Este proyecto me ha permitido introducirme más en un área en la que estoy enormemente interesado y de la que, desgraciadamente, no he aprendido tanto como me gustaría durante la carrera, la Inteligencia Artificial. En un futuro me gustaría seguir por este camino y aprender más sobre esa temática.

Aprovecho para agradecer a Manuel Gonzalez Bedía, director de este proyecto, su involucración e interés desde el día de comienzo. He podido contar con él en cualquier momento que lo haya podido necesitar y su pasión y conocimientos me han introducido en un mundo científico del que me gustaría seguir aprendiendo.

Capítulo 8

Bibliografía

- [1] W. Ashby. Design for a brain: The origin of adaptive behaviour. *New York, Wiley.*, 1952.
- [2] E. A. Di Paolo. Homeostatic adaptation to inversion of the visual field and other sensorimotor disruptions. *From Animals to Animats 6, Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior, Sep 2000, J-A. Meyer, A. Berthoz, D. Floreano, H. Roitblat and S. Wilson., MIT Press*, pages 440 – 449, 2000.
- [3] E. A. Di Paolo. Organismically-inspired robotics: homeostatic adaptation and teleology beyond the closed sensorimotor loop. *Erasmus Seminar Espaces d'action, espaces de perception. Université de Technologie de Compiègne*, 2003.
- [4] E. A. Di Paolo. Evolving robust robots using homeostatic oscillators. 2002.
- [5] B. Mathayomchan and R. D. Beer. Centre-crossing recurrent neural networks for the evolution of rhythmic behaviour. 14:2043 – 2051, 2002.
- [6] T. Hoinville and P. Henaff. Evolving plastic neural controllers stabilized by homeostatic mechanisms for adaptation to a perturbation. *International Conference on the Simulation and Synthesis of Living Systems (ALIFE9, Sep 2004, Boston, United States.*, pages 81 – 88, 2004.
- [7] G. G. Turrigiano. Homeostatic plasticity in neuronal networks: The more things change, the more they stay the same. *Trends Neuroscience*, 21:221–227, 1990.
- [8] J. Blynel and D. Floreano. Levels of dynamics and adaptive behavior in evolutionary neural controllers. *In B. Hallam, D. Floreano, J. Hallam, G. Hayes, and J.-A. Meyer, editors. Recompilation of papers*, 2002.

- [9] K. Funahashi and Y. Nakamura. Aproximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6:801–806, 1996.
- [10] Nolfi S. and D. Floreano. Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines. *MA: MIT Press/Bradford Books*, 2000.
- [11] A. Kern and H. Moll. On the transformative character of collective intentionality and the uniqueness of the human. *Philosophical Psychology*, 30(3):315–333, 2017.
- [12] P. Carruthers. The architecture of the mind: massive modularity and the flexibility of thought. *Oxford University Press*, 2006.
- [13] J. McDowell. Mind and world. *Harvard University Press*, 1996.
- [14] R. D. Beer. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 3(2):459–509, 1995a.
- [15] Hywel Thomas Parker Williams. Homeostatic adaptive networks. *Doctoral dissertation. University of Leeds*, pages 26–27, 2016.
- [16] M. E. Bratman. Shared agency: A planning theory of acting together. *New York, NY: Oxford University Press*, 2014.
- [17] M. Tomasello. A natural history of human thinking. *Cambridge, MA: Harvard University Press.*, 2014.
- [18] M. Van Duijn, F. Keijzer, and D. Franken. Principles of minimal cognition: Casting cognition as sensorimotor coordination. *Adaptive Behavior 2006*, 14(157), 2006.
- [19] M. Rohde. Enaction, embodiment, evolutionary robotics. simulation models in the study of human cognition. *Atlantis Press, Amsterdam, Series: Thinking Machines.*, 2010.
- [20] J. Carr. An introduction to genetic algorithms. 2014.

Anexos

Anexos A

Homeostat

Para comprender mejor los fundamentos de la homeostasis se diseñó e implementó un Homeostat en el lenguaje de programación Python siguiendo los fundamentos teóricos descritos en el punto explicativo de la homeostasis en este mismo documento.

Para ello, se implementó una sencilla red de 4 neuronas totalmente interconectadas. En cada iteración, cada neurona recibe una serie de entradas del resto de neuronas y de ella misma. El peso de estas entradas está definido por la fuerza de conexión entre ellas. El sumatorio ponderado de estas entradas sobre una neurona determina su salida s_i .

Al comienzo, los pesos de la fuerza de las conexiones están distribuidos de forma aleatoria dentro de los rangos apropiados $R = [0,5 - \alpha, \alpha + 0,5]$, donde α determina la opresión de la restricción homeostática. Tras cada iteración, si $s_i \in R$, la neurona es estable. Si $s_i \notin R$ la neurona ha perdido la estabilidad y se activa el mecanismo de cambio adaptativo.

En cuanto se detecta que una neurona ha salido del rango de estabilidad sus parámetros comienzan a variar de forma aleatoria (pero dentro de los límites establecidos para cada parámetro) hasta que la neurona vuelve a la estabilidad.

La finalidad de este experimento es la comprobar como al comienzo, las neuronas del sistema se podían encontrarse fuera de los rangos de estabilidad (debido a que al principio sus parámetros tienen valores aleatorios), pero tras una serie de iteraciones todas las neuronas se estabilizan.

En la figura A.1 podemos ver una gráfica que refleja el funcionamiento del Homeostato. En ella, cada línea de color representa la salida de una de las neuronas que componen el Homeostato. Como podemos observar, al comienzo, partiendo de valores aleatorios en los parámetros, las salidas son inestables y se encuentran fuera del rango establecido. Podemos observar como la neurona representada con el color verde es la última que se estabiliza. Es interesante también observar como las neuronas se influyen entre ellas durante este proceso de estabilización. Esto puede verse en el

último pico ascendente de la neurona representada con el color verde, el cual provoca un ligero movimiento en la salida de la neurona representada con el color azul, la cual no llega a desestabilizarse (la neurona verde arrastra a la azul con algunos de sus movimientos). Al cabo de un tiempo, todas las neuronas consiguen estabilizarse dentro del rango establecido. Una vez estables, sin la existencia de ninguna perturbación que provoque la salida de alguna de ellas de la zona de estabilidad, se mantendrán inmutables indefinidamente.

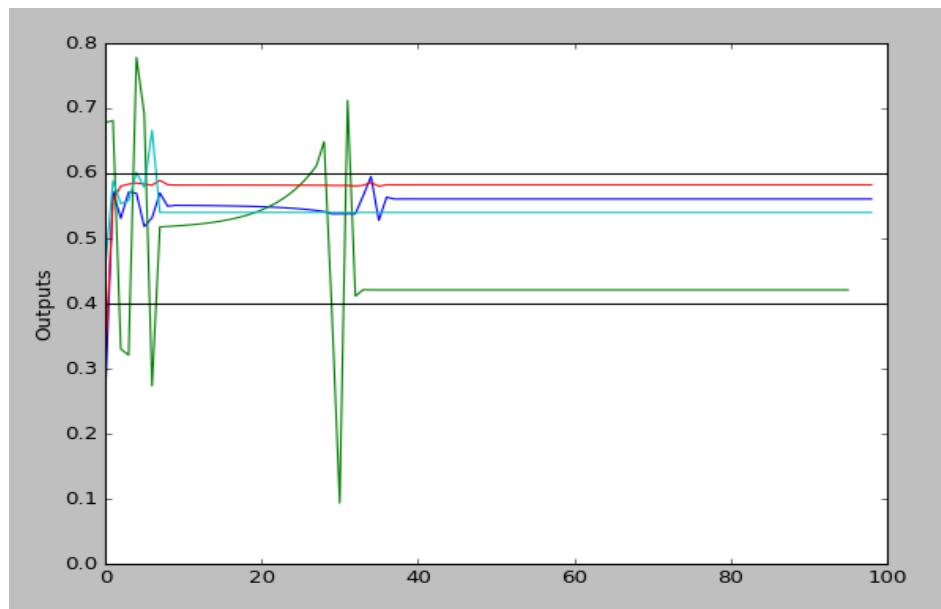


Figura A.1: Gráfica de salidas del Homeostato implementado. **Eje X:** tiempo. **Eje Y:** salidas de cada una de las neuronas que componen el Homeostato.

Anexos B

Algoritmo genético

Los algoritmos genéticos son un tipo de algoritmos de optimización, lo que quiere decir que se utilizan para obtener la solución o soluciones óptimas de un problema computacional[20].

Estos algoritmos representan una rama dentro del campo conocido como “Computación evolutiva”. En ella, se imitan los procesos biológicos de reproducción y selección natural con el fin de encontrar la mejor solución al problema para el que son creados. Al igual que en la evolución, muchos de los procesos de un algoritmo genético son aleatorios, aunque las técnicas de optimización permiten limitar el nivel de aleatoriedad y adaptar el control sobre el algoritmo que se considere necesario.

Este tipo de algoritmos permiten encontrar soluciones a problemas que otros algoritmos de optimización no pueden calcular por falta de continuidad, derivabilidad, linealidad u otras características.

B.0.1. Componentes, estructura y terminología

Dado que los algoritmos genéticos están diseñados para simular procesos biológicos, gran parte de su terminología ha sido tomada prestada de la biología. A pesar de ello, las entidades a las que esta terminología se refieren en el entorno de los algoritmos genéticos son mucho más simples que sus equivalentes biológicos. Los componentes básicos de prácticamente todos los algoritmos genéticos son:

- Una función **fitness** para la optimización
- Una **población** de **cromosomas**
- Una función de **selección** para elegir los cromosomas que van a reproducirse
- Una función de **recombinación** (o *crossover*) para la producción de la siguiente generación de cromosomas

- Una función de **mutación** para aleatorizar algunos cromosomas de la nueva población

La *función fitness* es la función que el algoritmo está tratando de optimizar. La palabra “fitness” se toma prestada de la teoría evolutiva. Se utiliza porque la función de fitness prueba y valora como de buena (como de “fit”) es cada solución potencial del problema. El término *cromosoma* hace referencia al valor o valores numéricos que representan posibles soluciones candidatas al problema que se está intentando resolver. Cada cromosoma está formado por una lista de parámetros, llamados *genes*. Al conjunto de cromosomas que van a evaluarse como posibles candidatos a solución del problema se le llama *población*. Antiguamente los cromosomas eran codificados como listas de unos y ceros (codificación binaria), sin embargo, actualmente la computación moderna permite codificar los cromosomas como números reales u otros objetos.

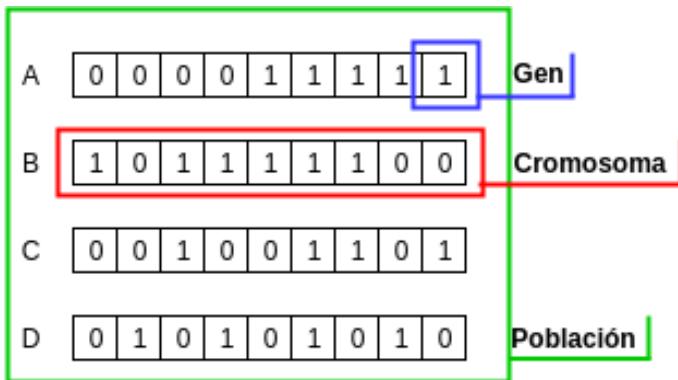


Figura B.1: Ejemplo de estructura de población, cromosomas y genes.

Un algoritmo genético comienza con una *población inicial* de cromosomas, generados de forma aleatoria, que representan la primera generación de posibles soluciones. A continuación, se evalúa cada cromosoma de dicha población mediante la función fitness para comprobar cómo de bien resuelve el problema planteado. Al finalizar, cada candidato tiene asignada una puntuación de fitness que representa lo bien (o lo mal) que lo ha hecho intentando resolver el problema. Una vez con la población inicial completamente evaluada, el *operador de selección* escoge de entre toda la población de candidatos algunos cromosomas para la reproducción (creación de una nueva generación). Existen muchos tipos de algoritmos de selección, pero generalmente en todos ellos se eligen los cromosomas que mejor puntuación fitness han obtenido, buscando que las siguientes generaciones creadas a partir de ellos den resultados como mínimo igual de buenos a los suyos. En el siguiente paso, el *operador de recombinación* simula las operaciones de recombinación de cromosomas presentes en las operaciones de división celular de tipo *meiosis*. En la figura B.2 se puede observar un ejemplo

de operación de recombinación. En este proceso se seleccionan dos cromosomas de la población y se recombinan para crear dos descendientes.

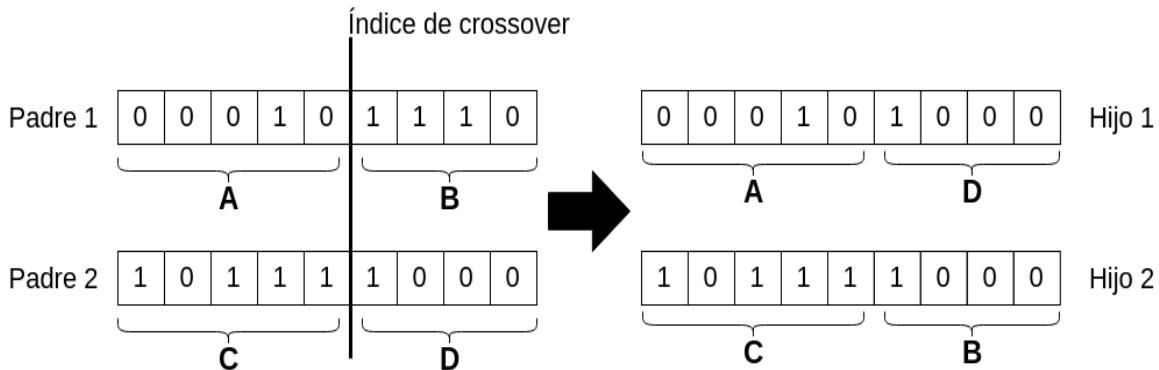


Figura B.2: Ejemplo de operación de recombinación de dos cromosomas.

El *operador de mutación* modifica de forma aleatoria genes de los cromosomas descendientes (por ejemplo cambiando un 1 por un 0 y viceversa, en cromosomas binarios). Estas mutaciones ocurren generalmente con una probabilidad bastante baja. A primera vista, el proceso de mutación puede parecer innecesario e incluso entorpecedor. La realidad es que toma un papel muy importante. Las operaciones de selección y recombinación mantienen la información genética de los cromosomas con mejor puntuación fitness, pero estos cromosomas sólo son considerados los mejores respecto a la población de la generación en la que fueron elegidos. Esto puede provocar que el algoritmo converga demasiado rápido y de una solución final alejada de la solución máxima. En otras palabras, puede provocar que el algoritmo se quede atascado en un máximo local antes de encontrar el máximo global óptimo. El operador de mutación ayuda a proteger el algoritmo de este problema manteniendo la diversidad en la población, a costa de aumentar el tiempo de convergencia.

Lo normal es que la selección, recombinación y mutación continúen hasta que el número de descendientes es igual al número de cromosomas de la población inicial, para que esta segunda generación esté compuesta en su totalidad por descendientes y reemplace a la población anterior.

Ahora la segunda generación se evalúa mediante la función fitness y todo el proceso se repite. Es una práctica común almacenar el cromosoma con la mayor puntuación de fitness junto con dicha puntuación para guardar un registro del mejor cromosoma hasta el momento.

Los algoritmos genéticos iteran hasta que la puntuación fitness del mejor cromosoma hasta el momento se estabiliza y no cambia en una serie de generaciones. Esto significa que el algoritmo ha convergido en una o solución. A todo el proceso de iteraciones se llama *carrera* (en inglés, *run*). Al final de cada carrera se tiene, normalmente, al menos

un cromosoma que es una solución lo suficientemente buena (lo suficientemente *fit*) del problema. Dependiendo del tipo de algoritmo utilizado, la mejor solución puede ser el mejor cromosoma que ha aparecido en todas las iteraciones o el mejor cromosoma de la última generación.

El rendimiento de un algoritmo genético depende enormemente de los métodos utilizados para codificar posibles soluciones en cromosomas y el criterio utilizado en la función fitness para evaluar a los candidatos. Otros detalles importantes a tener en cuenta son las probabilidades de recombinación y mutación, el tamaño de la población y el número de iteraciones. Estos valores puedes ser ajustados convenientemente después de analizar el rendimiento del algoritmo tras unas carreras de prueba.

Los algoritmos genéticos tienen una gran variedad de aplicaciones. Algunos ejemplos destacados son la *programación automática* y el “*machine learning*”. Funcionan también especialmente bien para modelar distintos fenómenos en economía, ecología, el sistema inmunológico humano, genética y sistemas sociales.

Anexos C

Gráficas completas de las activaciones neuronales del grupo de agentes de tipo 1

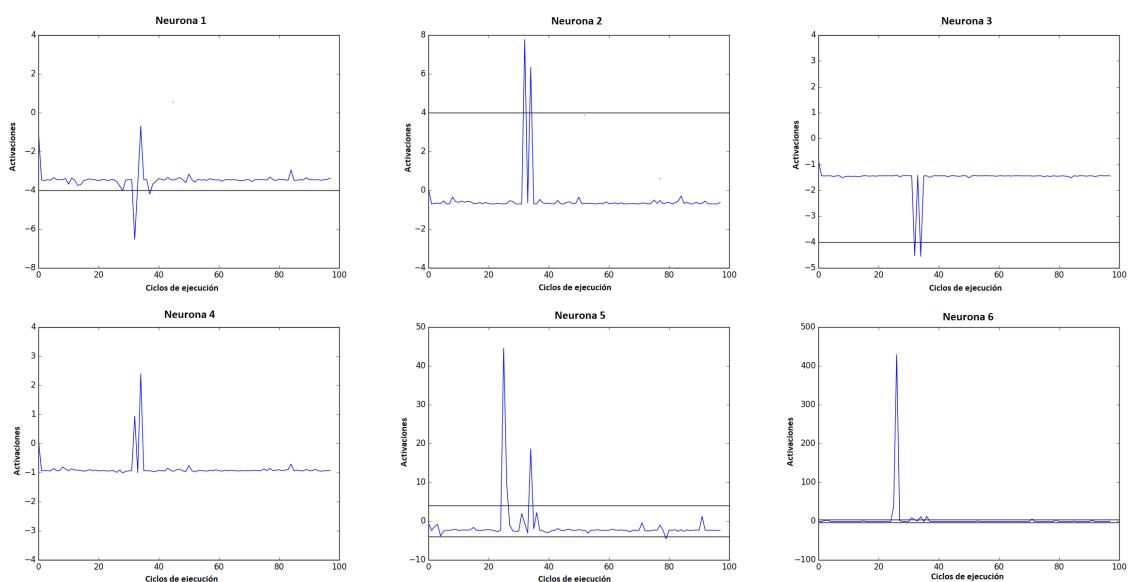


Figura C.1: Activaciones de las 6 neuronas del agente número 1 del tipo 1.

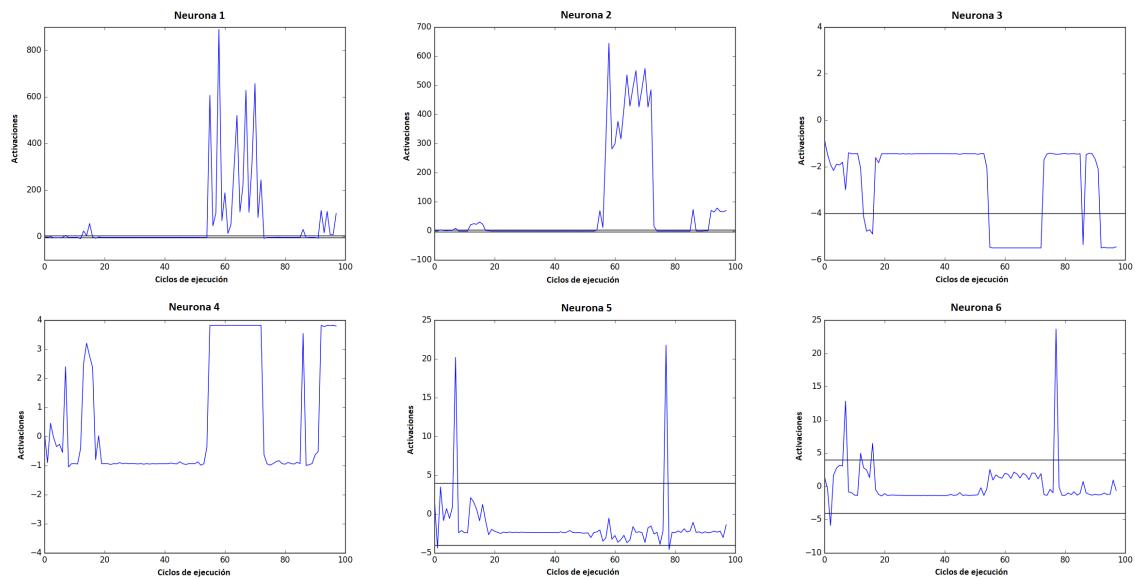


Figura C.2: Activaciones de las 6 neuronas del agente número 2 del tipo 1.

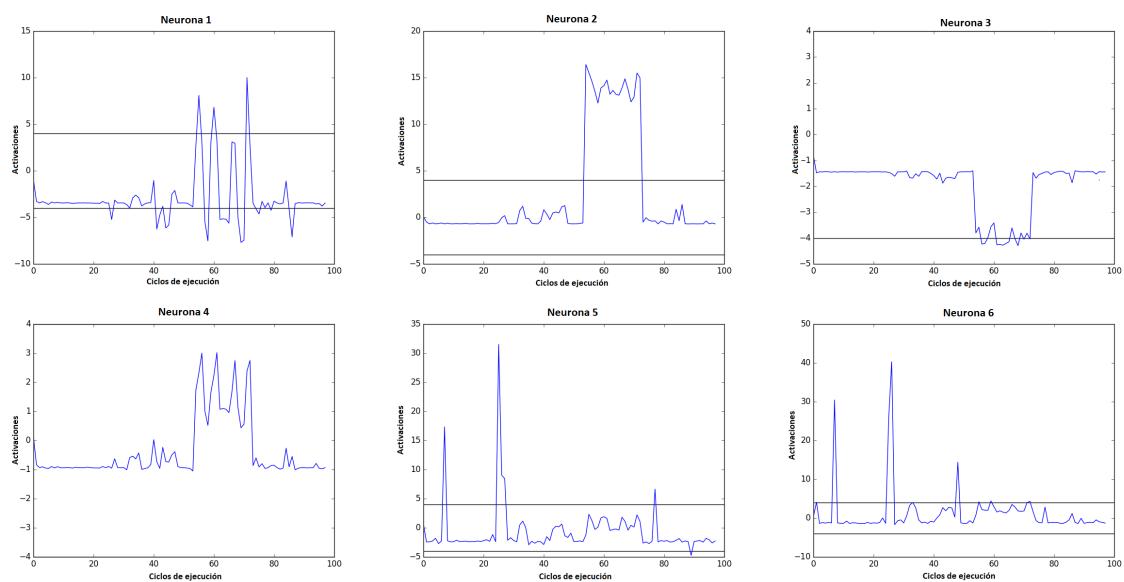


Figura C.3: Activaciones de las 6 neuronas del agente número 3 del tipo 1.

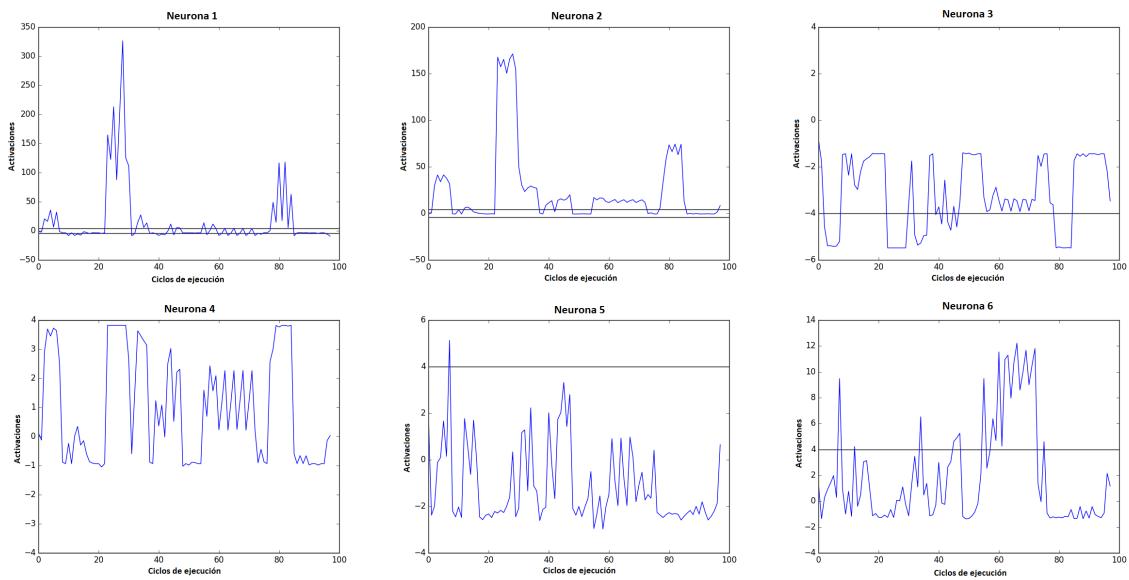


Figura C.4: Activaciones de las 6 neuronas del agente número 4 del tipo 1.

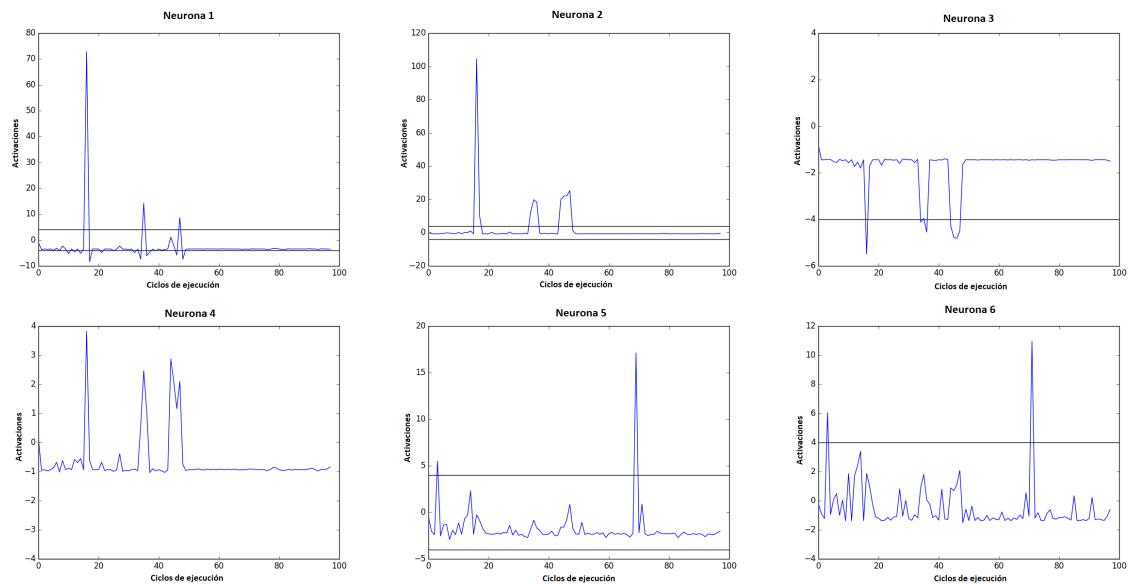


Figura C.5: Activaciones de las 6 neuronas del agente número 5 del tipo 1.

Anexos D

Gráficas completas de las activaciones neuronales del grupo de agentes de tipo 2

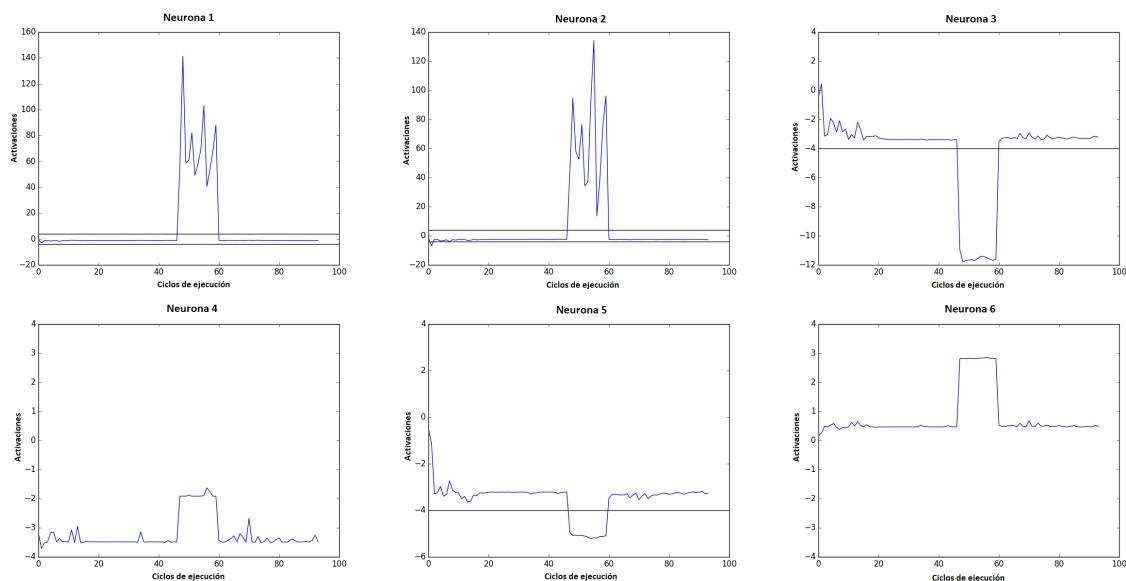


Figura D.1: Activaciones de las 6 neuronas del agente número 1 del tipo 2.

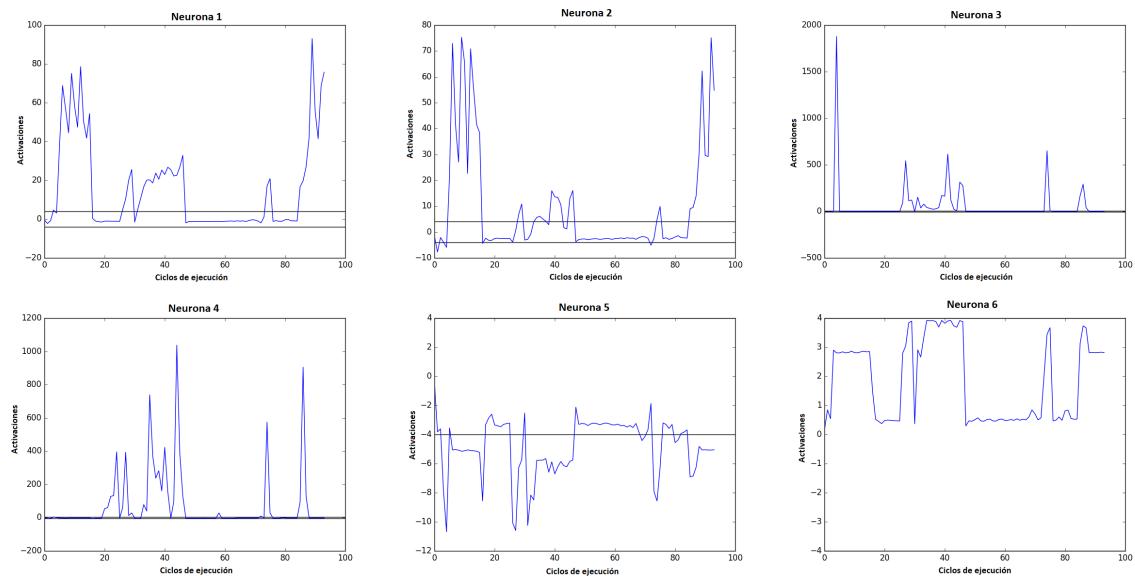


Figura D.2: Activaciones de las 6 neuronas del agente número 2 del tipo 2.

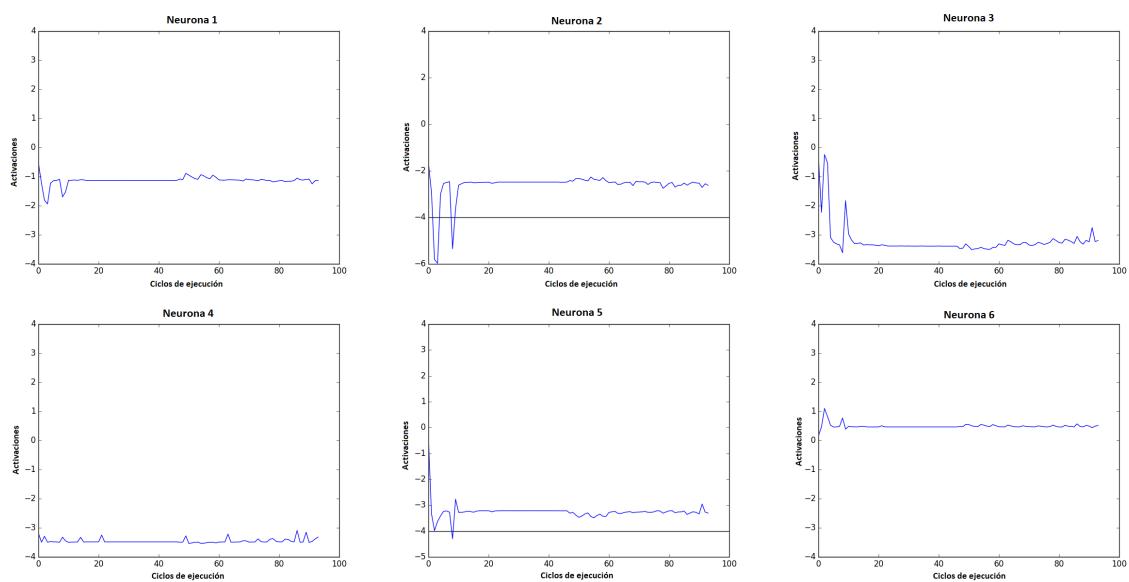


Figura D.3: Activaciones de las 6 neuronas del agente número 3 del tipo 2.

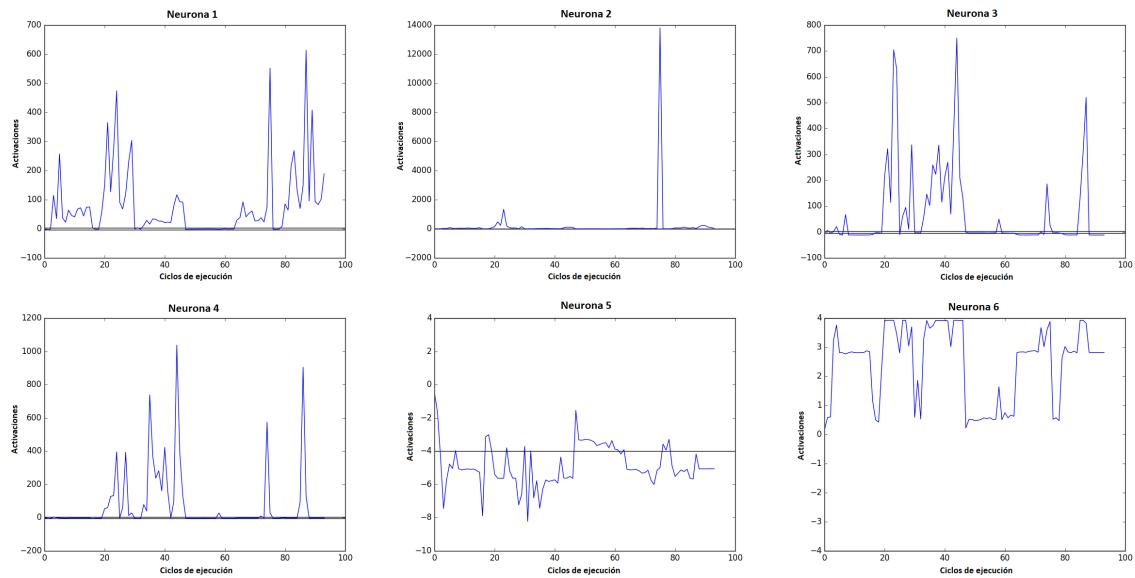


Figura D.4: Activaciones de las 6 neuronas del agente número 4 del tipo 2.

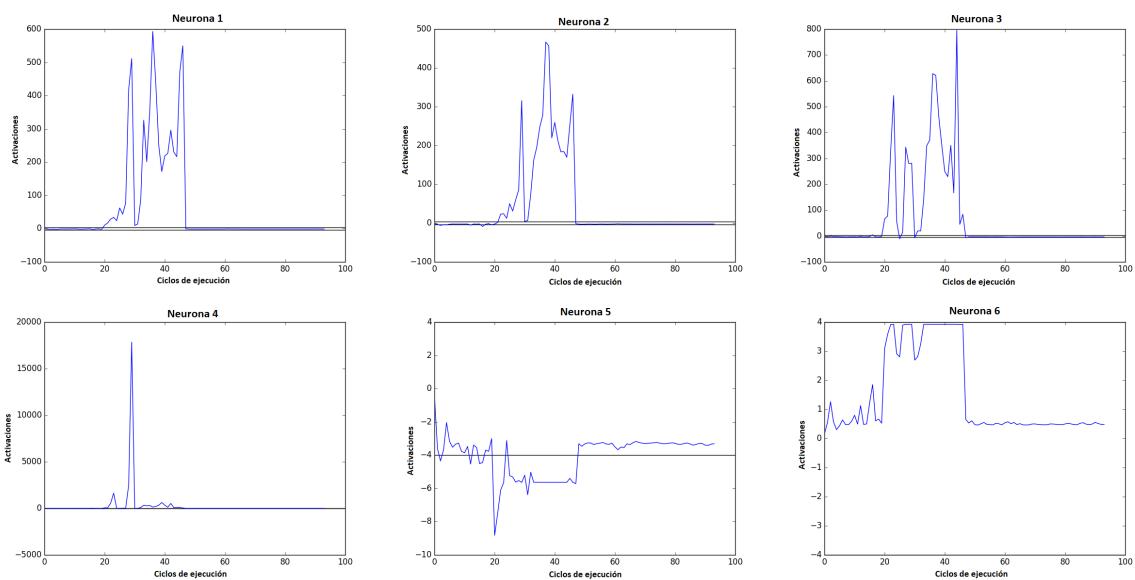


Figura D.5: Activaciones de las 6 neuronas del agente número 5 del tipo 2.

Anexos E

Planificación y cronograma

El control de horas del proyecto puede verse representado en la figura E.1, donde están indicadas las horas dedicadas a cada tarea del proyecto cada semana de la duración del mismo. Cabe aclarar que no se ha llevado control de horas de la tarea “investigación y lectura de documentación” ya que ha sido una tarea presente durante toda la duración del proyecto.

Tareas	Marzo				Abril				Mayo				Junio					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Diseño e implementación de Homeostato				9														
Diseño, implementación y evolución del Agente 0					7	4	9	20	18	11	11	6	4	5	19	13	1	
Ejecución y análisis del experimento individual							9	2	18	8		9	12	15	29	37	8	
Diseño, implementación y evolución del Agente 1														3	6	2		
Diseño, implementación y evolución del Agente 2														3		13	6	
Ejecución y análisis del experimento colectivo																6	24	
Redacción de la memoria del proyecto														2	3	9	16	26
Reuniones presenciales con el director del proyecto				1		1	1	1	1	2	1		1	2	1	2		
Investigación y lectura de documentación																		

Figura E.1: Cronograma del proyecto dividido por semanas, incluyendo las horas dedicadas cada semana a cada tarea.

En total se han invertido 417 horas a la realización de este Trabajo de Fin de Grado.