

Programa de Doctorado en Ingeniería Matemática,
Estadística e Investigación Operativa por la
Universidad Complutense de Madrid y la
Universidad Politécnica de Madrid



Simulación Social mediante Inteligencia Artificial y Teoría de Juegos

*Aplicación al estudio del efecto de las emociones
en las comunicaciones en redes sociales*

TESIS DOCTORAL

Óscar Serrano Cuéllar

Directores:

Juan Antonio Tejada Cazorla
José Francisco Vélez Serrano

Noviembre 2023



U N I V E R S I D A D
COMPLUTENSE
M A D R I D

**DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS
PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR**

D./Dña. _____,
estudiante en el Programa de Doctorado _____,
de la Facultad de _____ de la Universidad Complutense de
Madrid, como autor/a de la tesis presentada para la obtención del título de Doctor y
titulada:

y dirigida por: _____

DECLARO QUE:

La tesis es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, de acuerdo con el ordenamiento jurídico vigente, en particular, la Ley de Propiedad Intelectual (R.D. legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, modificado por la Ley 2/2019, de 1 de marzo, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), en particular, las disposiciones referidas al derecho de cita.

Del mismo modo, asumo frente a la Universidad cualquier responsabilidad que pudiera derivarse de la autoría o falta de originalidad del contenido de la tesis presentada de conformidad con el ordenamiento jurídico vigente.

En Madrid, a ____ de _____ de 20____

Fdo.: _____

Esta DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD debe ser insertada en
la primera página de la tesis presentada para la obtención del título de Doctor.

Dedicatoria

Dedico esta Tesis a ...

Agradecimientos

I want to thank...

Resumen

Simulación Social mediante Inteligencia Artificial y Teoría de Juegos.

Resumen de la Tesis aparece a continuación. Lo siguiente es un resumen. Según el esquema de la UCM, debe tener una extensión de entre 500 y 1000 palabra (vendría a ser un folio por las dos caras las 1000), e incluir:

- El título de la tesis.
- Introducción.
- Una síntesis que incluya, al menos, objetivos y resultados.
- Conclusiones.

Debe estar redactado en español e inglés. Además debe estar incluido en el índice de la tesis.

Abstract

Simulación Social mediante Inteligencia Artificial y Teoría de Juegos.

Introducción. Aquí aparece la Introducción. Mención al problema de flujo de información en una red, y el estudio que llevaremos a cabo. Primero llevaremos a cabo un análisis de las interacciones estratégicas que se producen entre los agentes.

Objetivos y resultados. En esta parte mencionar el empleo de herramientas de Inteligencia Artificial, como es el empleo de Aprendizaje por Refuerzo (*Reinforcement Learning*). Esto nos permitirá analizar la influencia de las emociones en el proceso de comunicación.

El código de las aplicaciones que se desarrollen podría incorporarse aparte en un repositorio, del que se proporcionaría la dirección (si se da el caso de no incorporarlo directamente al final de la Tesis, posiblemente en algún Apéndice).

Conclusiones. Resumir los resultados de las simulaciones llevadas a cabo, y la forma de provocar flujo masivo de información en una red.

Índice de figuras

1.1. Grupos emocionales detectados en el estudio Framingham (Fowler, J. H. & Christakis, N. A. (2008) [32]).	2
1.2. Representación en una red regular del Juego Iterado del Dilema del prisionero (los agentes que cooperan aparecen señalados como 'C').	2
1.3. Segmentación de una imagen - Facebook IA (Deep Learning for Generic Object Detection: A Survey - Liu, L. et al (2020)) [68].	3
1.4. La segmentación es en el fondo un problema de clasificación correcta de los píxeles de los que está compuesta la imagen, en cada uno de los distintos tipos de entidades en cuyo reconocimiento ha sido entrenada la red neuronal: (class 0: grass, 1: person, 2: sheep, 3:dog).	3
1.5. Representación en una red regular de la difusión de la cooperación.	4
1.6. Partiendo de la representación de la activación emocional de la red social (codificandola con colores), buscamos un resultado: ser capaces de clasificar los nodos de la red.	4
1.7. Representación a través de un grafo de las relaciones en una Red Social.	5
1.8. Algunos tipos de estructuras de una Red Social.	6
1.9. Representación como cuadrícula de una red regular.	6
1.10. Rejilla de simulación (grid). Aparece resaltada una de sus celdas y su vecindario inmediato.	7
1.11. Simulación de la aparición de segregación racial mediante autómatas.	7
1.12. Evolución de la cultura tras 20.000, 40.000 y 80.000 ciclos (Axelrod(1997) [5]).	8
1.13. a) Dilema de los bienes públicos b) Dilema tragedia de los comunes.	8
1.14. Cada emoción (interacción local) está muy relacionada con tipos específicos de conductas. Su expansión viral genera efectos macroscópicos detectables que podemos estudiar y asociar a la presencia de cierto tipo de agentes en la red. Nos interesa descubrir su localización.	11
1.15. Esquema de los bloques que conforman la Tesis.	12
2.1. Representación del tipo de interacción entre los individuos de la Red Social.	15
2.2. Matriz de pagos del Dilema del Prisionero.	17
2.3. Modelo de dinámica de respuestas emocionales en un agente dotado de 3 emociones (ira, alegría y tristeza). Las interacciones están basadas en el Dilema del Prisionero.	18
2.4. Función de decisión del agente.	18
3.1. Captura de la aplicación para Simulaciones (Prueba con un Grid de 10x10).	21
3.2. Evolución del aprendizaje con Agentes Emocionales, y difusión viral al final.	22
4.1. Arquitectura de Red Neuronal 'CNN' (Convolutional Neural Network).	27
4.2. Arquitectura de Red Neuronal 'U-Net'.	28

4.3.	<i>Algunas funciones de activación usuales (donde: $net = W \cdot input + b$).</i>	30
4.4.	<i>Representación gráfica del procesamiento de información que realiza una Red Neuronal. Se han empleado unos inputs de dimensión 'p': $\mathbf{x} = (x_1, x_2, \dots, x_p)$ y dos capas ocultas de neuronas. La neurona final genera un valor \mathbf{y}. En todos los casos se ha empleado una función de activación tangente hiperbólica.</i>	30
4.5.	<i>Obtención del vector de parámetros óptimo θ^*.</i>	31
4.6.	<i>Algunas arquitecturas de Redes Neuronales (Esquemas adaptados de Principe, J. C. et al. (1999) [92], Goodfellow, I. et al. (2016) [38] y Brunton, S. L. & Kutz, J. N. (2022) [21]).</i>	33
4.7.	<i>Funcionamiento del bufer, y extracción posterior de una muestra para entrenar la Red.</i>	34
4.8.	<i>Funcionamiento del bufer, y extracción de una muestra ponderada por el efecto que la observación tiene en el error.</i>	37
4.9.	<i>Estructura de la Red Neuronal empleada en Dueling DQN.</i>	38
4.10.	<i>Esquema de la implementación del modelo A3C.</i>	41
5.1.	<i>Arquitectura de Red Neuronal 'CNN 3D'.</i>	45
6.1.	<i>Comparación de los resultados obtenidos por distintos agentes. El mecanismo de interacción empleado en las simulaciones es el Dilema del Prisionero.</i>	48
6.2.	<i>Evolución de la precisión durante el entrenamiento de la red U-Net. Prueba con 2 poblaciones. Curva ROC con los resultados relativos al problema de clasificación de dos poblaciones.</i>	49
6.3.	<i>Curva de la función de pérdida (entrenamiento y validación) y evolución de la precisión en el entrenamiento.</i>	49
6.4.	<i>Tratamiento del problema de localización de los nodos influyentes como un problema de segmentación de imágenes. Prueba con una pequeña red de 20x20. Los individuos mutantes aparecen que han sido correctamente identificados aparecen en color verde (1), en rojo los fallos de clasificación.</i>	49
7.1.	<i>Representación de diversos tipos de procesos.</i>	56

Índice de cuadros

Índice general

Declaración de autoría y originalidad	III
Dedicatoria	V
Agradecimientos	VII
Resumen	IX
Abstract	XI
Índice de figuras	XIV
Índice de cuadros	XV
1. Introducción	1
1.1. Conceptos	5
1.2. Antecedentes	7
1.3. Aportaciones de la Tesis	9
1.4. Hipótesis y objetivos	10
1.4.1. Las emociones actúan como un mecanismo de aprendizaje	10
1.4.2. El tipo de información genera distintos tipos de difusión	10
1.5. Estructura de la tesis. - vol I	12
2. Modelo Markoviano multiagente	15
3. Viralización de comportamientos en una Red Social	21
3.0.1. Simulación: Estudio de las interacciones entre agentes	22
4. Estudio de los agentes influyentes en una Red Social I	27
4.0.1. Estructura de Red Neuronal 1: CNN	27
4.0.2. Estructura de Red Neuronal 2: U-Net	28
4.1. Redes Neuronales	29
4.2. Entrenamiento de una red neuronal	31
4.3. Arquitecturas de Redes Neuronales	32
4.4. Utilización de Redes Neuronales en RL	34
4.5. Value based Methods.	34

4.5.1.	Deep Q-Network (DQN)	34
4.5.2.	Double Deep Q Network (DDQN)	36
4.5.3.	Deep Q-Network with Prioritized Experience Replay (PER)	37
4.5.4.	Dueling Deep Q-Network (Dueling DQN) (D2QN)	38
4.5.5.	Deep Recurrent QN (DRQN)	38
4.6.	Policy Methods.	39
4.6.1.	Proximal Policy Optimization (PPO)	39
4.6.2.	Trust Region Policy Optimization (TRPO)	39
4.6.3.	Actor-Critic using Kronecker-factored Trust Region (ACKTR)	39
4.7.	Actor-critic Methods	40
4.7.1.	Advantage Actor-Critic (A2C)	40
4.7.2.	Asynchronous Advantage Actor-Critic (A3C)	41
4.7.3.	Deep Deterministic Policy Gradient (DDPG)	41
4.7.4.	Twin Delayed Deep Deterministic Policy Gradient (TD3)	42
4.7.5.	Soft Actor-Critic (SAC)	42
5.	Estudio de los agentes influyentes en una Red Social II	45
5.0.1.	Estructura de Red Neuronal 3: CNN - 3D	45
6.	Resultados	47
7.	Conclusiones	51
7.1.	Procesos de Decisión de Markov	56
	Referencias	61
	Glosario	75

Capítulo 1

Introducción

El grado de globalización actual de nuestra sociedad facilita la influencia entre individuos. El desarrollo de las plataformas de comunicación on-line potencia aún más el proceso, debido especialmente a la inmediatez en el contacto que permiten. La información que fluye por la red tiene el efecto potencial de moldear nuestra forma de interpretar los acontecimientos a nuestro alrededor e influir en nuestra conducta, con los riesgos que esto conlleva. Es por ello que el estudio de la forma más extrema de difusión como es la difusión viral tiene un interés especial.

Algunos estudios apuntan, por ejemplo, a su papel en fenómenos como la movilización política y la generación de un clima de opinión (Bond et al. (2012)) [18]. Un peligro cada vez más real es la utilización de métodos de comunicación masiva como un medio para la manipulación. Así lo atestiguan estudios sobre la propagación de noticias falsas (Vosoughi et al. (2018)) [123], o la polarización de la población en posiciones política extremas, con el uso de *bots* para inflamar los contenidos de la red (Stella et al.) [114].

La modelización de este tipo de procesos de difusión viral es fundamental. Entender los mecanismos que subyacen, puede hacernos capaces de desarrollar medidas que puedan hacer frente a efectos dañinos como los que se acaban de mencionar.

Los estudios clásicos sobre estructuras de comunicación han recurrido fundamentalmente a su representación mediante grafos. Sin embargo el tratamiento algorítmico de este tipo de objetos matemáticos puede llegar a ser muy complejo, generando tiempos computacionales muy elevados, tal y como ponen de manifiesto trabajos clásicos en la difusión de información. Tenemos por ejemplo el planteamiento del problema de viralizar un producto de marketing en sitios de compartir información por Domingos & Richardson (2001) [27] y (2002) [95]. El Modelo de referencia lo tenemos en Kempe, Kleinberg y Tardos ([55] (2003) y [54] (2005)), donde se aborda la complejidad computacional elevada del problema.

Es por esto por lo que nos planteamos la introducción de aportaciones metodológicas que permitan estudiar el comportamiento de una red social sin acudir a este tipo de algoritmos que operan directamente sobre la estructura de la red.

El proceso de investigación que hemos llevado a cabo está centrado en dos objetivos:

1. Simulación de procesos de difusión viral en una red social.
2. Detección de los agentes influyentes de la red.

Si observamos, además, algunos de los resultados de investigaciones académicas, encontramos que en muchos de estos tipos de procesos de difusión parecen tener un papel esencial las emociones. Algunas investigaciones reflejan este fenómeno de la difusión de emociones (Miller, M. et al. (2011) [75]), y estudios clásicos como el de Framingham permiten descubrir relaciones interesantes, como que los estados emocionales parecen contagiarse a las personas cercanas (Fowler, J. & Christakis, N. (2008) [32]) (ver fig. 1.1).

La modelización que llevaremos a cabo de procesos de comunicación introducirá este efecto de las emociones de los agentes y se llevará a cabo sobre redes sencillas en forma de cuadrícula. Para el desarrollo de la investigación y los objetivos planteados distinguiremos dos fases: una de simulación en la que proponemos un modelo propio de difusión, y una fase posterior de detección de los puntos claves de la red social.

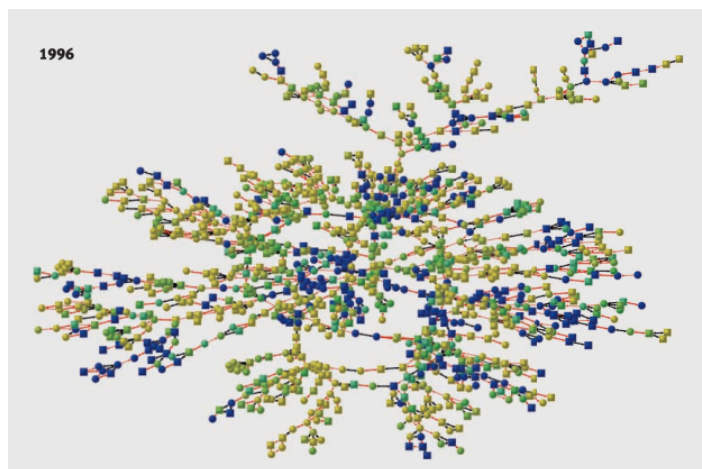


Figura 1.1: *Grupos emocionales detectados en el estudio Framingham (Fowler, J. H. & Christakis, N. A. (2008) [32]).*

FASE de SIMULACIÓN: Para integrar las emociones en la red, emplearemos como desencadenante de todo el proceso de actividad en la Red Social un juego: el Dilema del Prisionero, que los agentes jugarán de manera iterada. Pese a su aparente sencillez, permite esquematizar el tipo de relaciones que podríamos encontrar: la alegría por ser correspondido, el dolor de la traición, el gozo por vengar una afrenta previa o el pesar por haber tratado injustamente a un oponente. La difusión de la cooperación vendría a representar un proceso de difusión viral (fig. 1.2).

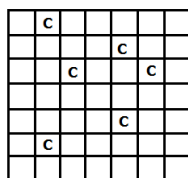


Figura 1.2: *Representación en una red regular del Juego Iterado del Dilema del prisionero (los agentes que cooperan aparecen señalados como 'C').*

El análisis estratégico del juego lo llevaremos a cabo introduciendo agentes mutantes que alteren la actividad de la red social, de modo similar a como lo harían en una red de comunicaciones quienes perturban las interacciones introduciendo conflictividad (*'trolls'*), o intentan imponer un clima de opinión.

FASE de DETECCIÓN DE AGENTES: utilizaremos técnicas de Inteligencia Artificial. El procedimiento que emplearemos consistirá en codificar la actividad emocional de la red social mediante colores, transformando la red regular de la que partiremos en una imagen. Posteriormente utilizaremos técnicas de procesamiento de imágenes para detectar patrones. Es decir, estos procesos de interacción simulados nos servirán para entrenar distintos tipos de Redes Neuronales, con la intención de detectar los patrones de actividad que generan los diferentes tipos de agentes.



Figura 1.3: Segmentación de una imagen - Facebook IA (Deep Learning for Generic Object Detection: A Survey - Liu, L. et al (2020)) [68].

La técnica elegida en concreto será la de Segmentación de imágenes (ver fig. 1.3). Este tipo de procedimientos nos permiten entrenar una Red Neuronal para detectar cierto tipo de entidades, localizarlas y trazar su silueta en la imagen.

Si observamos más detenidamente, vemos en la figura 1.4 que la detección de la forma de las distintas entidades que existen en la imagen, es en el fondo un problema de clasificación de todos y cada uno de los píxeles que forman parte de ella.

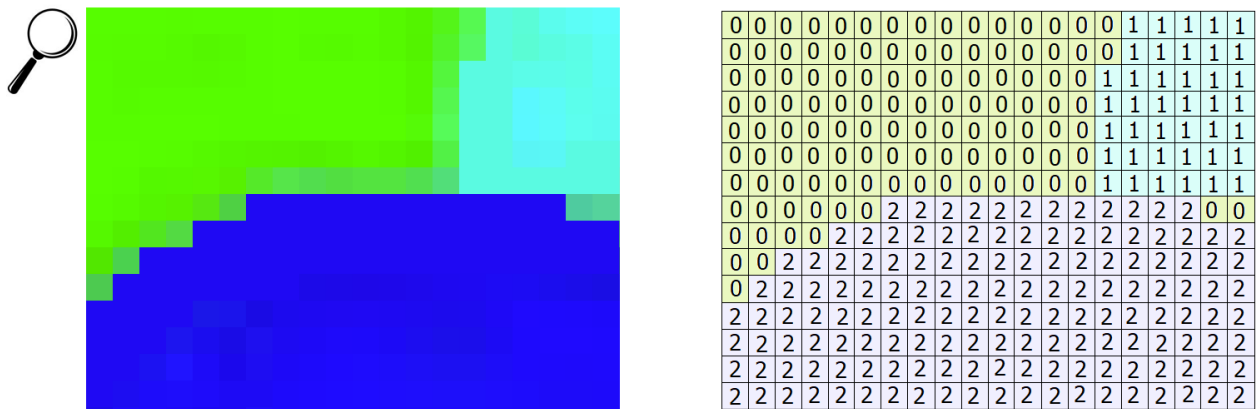


Figura 1.4: La segmentación es en el fondo un problema de clasificación correcta de los píxeles de los que está compuesta la imagen, en cada uno de los distintos tipos de entidades en cuyo reconocimiento ha sido entrenada la red neuronal: (class 0: grass, 1: person, 2: sheep, 3: dog).

Nosotros emplearemos dicho procedimiento para tratar de clasificar todos los nodos de la red y desvelar con ello la identidad de los nodos conflictivos. Todo lo anterior se condensa y concreta en las siguientes líneas.

Problema a tratar. El problema que pretendemos abordar es el estudio de la actividad de difusión viral en una Red Social. La situación que proponemos es la existencia de varios tipos de agentes. Uno de ellos intenta imponer un clima conflictivo al resto de agentes, de naturaleza emocional, e influenciables, y que pueden verse arrastrados a la situación que se pretende imponer. Nos interesa especialmente detectar la presencia de estos mutantes y su localización en la Red.

En primer lugar utilizaremos el Dilema del Prisionero Iterado sobre una cuadrícula, en línea con estudios clásicos sobre difusión de la cooperación, para representar un fenómeno de difusión viral (fig. 1.5). La naturaleza del juego nos permite introducir un agente emocional que reaccionará a los distintos resultados del juego, llevando a cabo un aprendizaje a lo largo del tiempo. Estos agentes representan una masa fácilmente influenciable que facilita en este caso la propagación de la conducta cooperativa, pero que pueden verse arrastrados en sentido opuesto. La introducción de agentes mutantes introduce el factor de conflictividad.

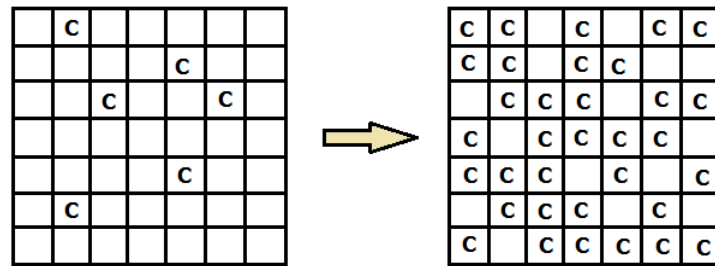


Figura 1.5: *Representación en una red regular de la difusión de la cooperación.*

Posteriormente el objetivo consistirá en detectar estos focos de conflictividad en una red, donde los agentes están intentado perturbar la actividad, y en generar una polarización de la red, y en último extremo evitar la actividad cooperativa.

La actividad emocional de la red será representada mediante imágenes, las cuales servirán para entrenar Redes Neuronales en la detección de los patrones de actividad que generan los agentes.

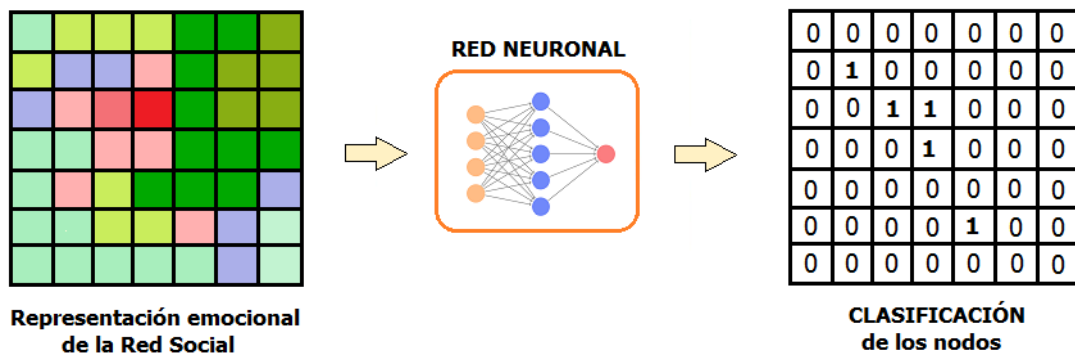


Figura 1.6: *Partiendo de la representación de la activación emocional de la red social (codificándola con colores), buscamos un resultado: ser capaces de clasificar los nodos de la red.*

Empleando técnicas de segmentación de imágenes trataremos de clasificar todos los nodos de la red y desvelar con ello la identidad de los nodos conflictivos (ver fig. 1.6). Al centrar el esfuerzo en la fase de entrenamiento, la detección de agentes se realiza automáticamente, evitando los largos tiempos aparejados a los algoritmos sobre grafos.

1.1. Conceptos

Un aspecto clave en cualquier situación social que podamos imaginar es la interacción y comunicación entre individuos: la transmisión de ideas o la adopción de prácticas comunes. En lo que sigue definimos algunos conceptos que aparecerán a lo largo del presente trabajo.

Representación de una Red Social.

Una sociedad puede interpretarse como un conjunto de múltiples elementos que interactúan entre sí. Bajo esta consideración aparece de manera natural un objeto matemático: el grafo. Definido de manera rigurosa consiste en un conjunto G que especifica el conjunto de aristas (E) y vértices (V) que componen la red [8].

Cada uno de los vértices vendría a representar a un individuo o **agente**, y las aristas que los interconectan representan las relaciones existentes entre ellos.

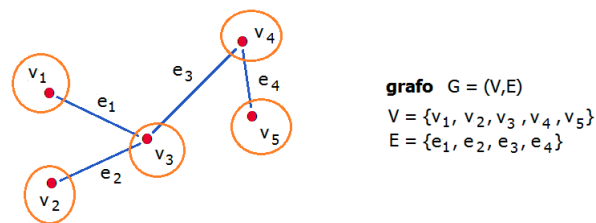


Figura 1.7: Representación a través de un grafo de las relaciones en una Red Social.

Vemos en la Figura 1.7, que una simple inspección visual puede proporcionar información del nivel de complejidad de las relaciones existentes en la Red Social. Si atendemos al grado de los vértices vemos que el grado de v_3 es igual a 3. El resto de los nodos tiene grado 1 menos v_4 que tiene grado 2.

Interacción.

En sentido general puede tratarse de la transmisión de hábitos, habilidades, creencias, ideas, valores, por ejemplo. La adopción de una determinada conducta (hábito o idea) puede modelizarse como un proceso de contagio, y resulta importante el estudio de cómo se extienden dichos comportamientos en una determinada población.

Individuos influyentes.

Entendemos que existe influencia cuando un individuo (agente) altera su comportamiento como consecuencia de las interacciones que lleva a cabo con otros individuos con los que está en contacto.

En sentido general se trata de la adopción de un determinado hábito, idea o comportamiento, o también la adquisición de un determinado producto, como consecuencia de las relaciones que entabla cada individuo con los que le rodea.

La primera dificultad consiste en distinguir si los individuos se comportan como lo hacen fruto de la influencia entre ellos, o existe una similitud previa, tal como han recalado algunos autores (Anagnostopoulos, A. et al. (2008) [4] y Crandall, D. et al (2008) [24]).

Partiremos de la hipótesis de que efectivamente existe influencia entre agentes. Algunos de los estudios que han detectado este efecto son los llevados a cabo en sistemas de recomendación (Leskovec et al. (2006) [64]), así como los estudios llevados a cabo sobre la influencia de comunicaciones en redes sociales y el grado de movilización política (Bond, R.M. et al (2012) [18]).

Estructura de una Red Social.

Si atendemos al tipo de conexiones que unen a los agentes de la red social podemos distinguir varios tipos de redes (ver fig. 1.8). Las redes 'mundo pequeño' (*small world*) son aquellas en las que todos los nodos están interconectados a través de un número reducido de conexiones. En una red aleatoria los enlaces entre cada par de nodos se crean con una cierta probabilidad 'p'. Otro tipo de red interesante es la red libre de escala. En este tipo de redes, pese a que el grado de conexión es bastante bajo, existen algunos nodos diferenciales que están conectados a un gran número de nodos, lo que les convierte en claves para la difusión de información por ejemplo.

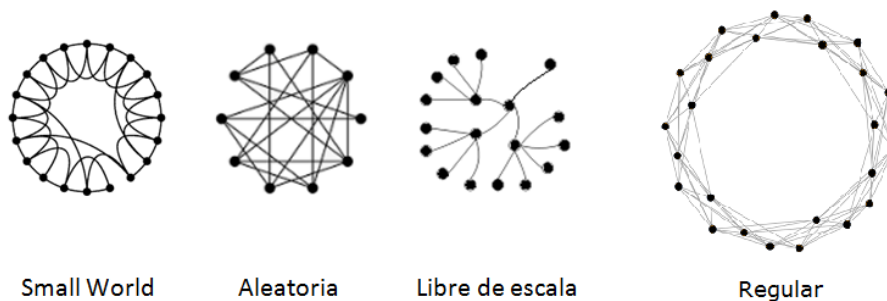


Figura 1.8: *Algunos tipos de estructuras de una Red Social.*

El tipo de red para el que desarrollaremos el presente trabajo es un tipo de red regular, que tiene la característica de que puede ser representada como una rejilla o '*grid*'. Esto nos permitirá tratar la red directamente como una imagen. A su vez sobre este tipo de estructuras es sobre la que se han desarrollado los trabajos clásicos sobre, por ejemplo, la evolución de la cooperación en Teoría de Juegos (tema que trataremos extensamente más adelante).

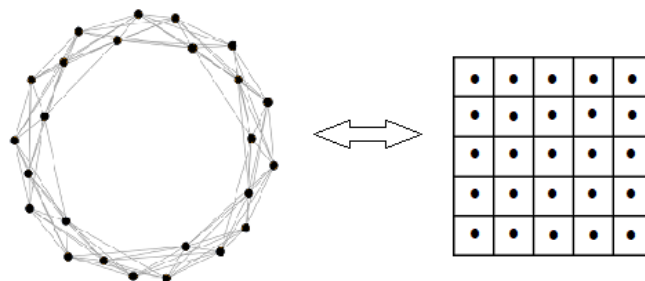


Figura 1.9: *Representación como cuadrícula de una red regular.*

Por tanto, la tesis se desarrollará sobre una estructura de red regular que admite una representación en cuadrícula, tal y como aparece en la figura 1.9, planteándonos posteriormente la extensión a otros tipos de estructuras más complejas.

1.2. Antecedentes

Los orígenes de la simulación computacional de fenómenos sociales habría que situarlos en la idea de von Neumann de los autómatas celulares [80]. Básicamente se trata de una rejilla regular formada por celdas, cada una de las cuales puede tomar un número finito de estados, y donde en cada uno de una serie de pasos sucesivos, el estado de cada celda varía de acuerdo a ciertas reglas y la influencia del estado de las celdas vecinas. La figura 1.10 permite hacerse una idea.

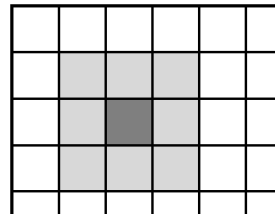


Figura 1.10: *Rejilla de simulación (grid). Aparece resaltada una de sus celdas y su vecindario inmediato.*

Pese a su aparente sencillez, esta representación puede modelizar procesos bastante complejos. Buscando la representación de la evolución de un ser vivo, el procedimiento sería puesto en práctica en el trabajo clásico de John Conway: 'El Juego de la vida' (Conway, J. (1970) [37]), poniendo de manifiesto que reglas de aplicación local pueden generar la emergencia de patrones globales en el tablero. Thomas Schelling emplearía la misma técnica posteriormente para el estudio del surgimiento de la segregación en una sociedad (Schelling, T. (1981) [104]), tal como se puede ver en la figura 1.11. Basta con inducir en los individuos una preferencia por otros vecinos cercanos similares para observar este efecto de separación. Es decir: agentes que sólo procesan información local, generan efectos macroscópicos en toda la red (Schelling, T. - "Micromotives and Macrobehavior" (2006) [105]).

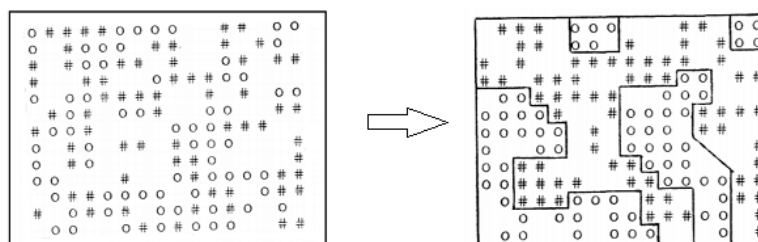


Figura 1.11: *Simulación de la aparición de segregación racial mediante autómatas.*

Simulación de fenómenos sociales y difusión de la cooperación. El concepto de autómatas sería extendido gradualmente de manera satisfactoria a otras áreas como la Teoría de Juegos (juegos iterados basados en el Dilema del Prisionero), dando lugar a análisis sobre el surgimiento de la cooperación por Axelrod en su clásico estudio sobre el tema (Axelrod, R. (1981) [6]). Análogo enfoque sería posteriormente utilizado en el campo de la Biología por Martin Nowak (Nowak, M. (2006) [82]), y John Maynard Smith (Smith, J. M. (1979) [113]), cuyos resultados arrojarían luz sobre la manera en que evolucionan las especies.

Axelrod, años después de su estudio sobre la cooperación, llevaría a cabo análisis sobre la diseminación de la cultura empleando nuevamente simulaciones computacionales (Axelrod, R. (1997) [5]). En su artículo, el autor trata el mecanismo de influencia social convergente empleando un modelo basado en agentes. Con él pretende explicar el fenómeno consistente en que, pese a la tendencia a la convergencia en creencias, actitudes y comportamiento (fruto del contacto continuado entre personas), las diferencias entre individuos y grupos persisten en cierto grado (fig. 1.12).

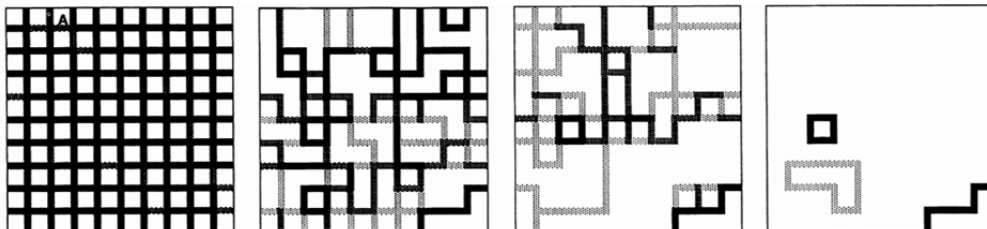


Figura 1.12: *Evolución de la cultura tras 20.000, 40.000 y 80.000 ciclos (Axelrod(1997) [5]).*

Planteado explícitamente como un dilema social secuencial tenemos el trabajo de referencia de Leibo et al. (2017) [63], que utiliza métodos de aprendizaje por refuerzo. Siguiendo este trabajo inicial tenemos una serie de artículos que explotan algunos de los dilemas que plantea, para los que desarrolla un entorno para generar las simulaciones (ver figura 1.13). Tenemos por ejemplo el estudio de Eccles, T. (2019) [28] sobre el aprendizaje de la reciprocidad; o el de Hughes, E. (2018) [50], en el que introduce la aversión a la desigualdad como factor que induce la aparición de comportamientos prosociales.

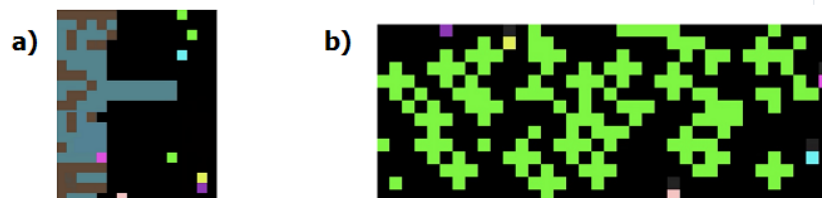


Figura 1.13: *a) Dilema de los bienes públicos b) Dilema tragedia de los comunes.*

Otros trabajos destacan por introducir el grado de influencia social como motivación de los agentes a la hora de tomar decisiones (Jaques, N. et al. (2019)) [52], o la formación de grupos de agentes como favorecedora de la emergencia de la reciprocidad (Baker, B. (2020) [7]).

Estudio de las emociones y la difusión de la cooperación. Entre los estudios que se centran en el papel de las emociones y su papel a la hora de inducir la cooperación en el Dilema del Prisionero tenemos por ejemplo los trabajos sobre sentimientos morales y su efecto en el dilema del prisionero iterado en un sistema multiagente (Bazzan, A. et al. (1998) [11]. En esta línea los autores proponen un marco de trabajo para el desarrollo de agentes dotados de emociones en el Dilema del Prisionero iterado (Bazzan, A. & Bordini, R. (2001)) [9]. Siguiendo estas líneas tenemos aportaciones como la evolución del comportamiento en agentes dotados de sentimientos morales profundizado en Bazzan, A. et al. (2002) [10], o sobre el papel de los lazos sociales en el desarrollo de la cooperación en el dilema del prisionero iterado tenemos el estudio llevado a cabo por Bazzan, A et al. (2011) [12].

El impulso de imitar estrategias favorables presente en algunos trabajos clásicos sobre evolución de la cooperación, ha sido sustituido por la imitación de emociones, como vía de elevar el bienestar social en juegos espaciales e inducir la propagación de la cooperación (Szolnoki, A. (2011) [118]).

Basándose en procedimientos de aprendizaje por refuerzo tenemos otros trabajos: empleando las emociones como una motivación intrínseca tenemos Sequeira, P. et al. (2011) [108]), y sobre el diseño de este tipo de señales intrínsecas capaces de guiar el aprendizaje en Sequeira, P. et al. (2014) [109]). Podemos encontrar también un modelo con algunas emociones en Broekens, J. et al. (2015) [20]. Y sobre la extensión en el tiempo de las estrategias cooperadoras Peysakhovich, A. & Lerer, A. (2017) [89] y (2018) [90]. En idéntico sentido tenemos también el trabajo ligando de emociones y aprendizaje por refuerzo de Yu et al. (2015) [131].

Considerando la importancia del contagio de emociones y la competición entre estados emocionales, y el papel que podría jugar en la formación de la opinión pública tenemos el enfoque de Fan et al. (2018) [30].

1.3. Aportaciones de la Tesis

Quizás hacer algún comentario sobre la importancia de la investigación. Problemas computacionales en el tratamiento de redes grandes. Transformar en imágenes y utilizar redes neuronales entrenadas en visión por computadora para detectar de manera automática los puntos críticos de la red.

Hay redes para las cuales este tipo de metodología puede ser apropiada. En la actualidad disponemos de métodos de aprendizaje automático que permiten un tratamiento del lenguaje y etiquetar nodos de red con sentimientos. Además algunas redes específicas (X - Twitter, Shina Weibo, etc.) disponen de procedimientos automáticos para etiquetado de estado emocional. Esto permitiría aplicar a este tipo de redes de comunicación los procedimientos que se han desarrollado.

Nos planteamos la introducción de aportaciones metodológicas que permitan estudiar el comportamiento de una red social sin acudir a algoritmos sobre el grafo.

1) Planteamiento de un modelo propio de difusión en el que las emociones juegan un papel esencial. Construimos agentes mediante Aprendizaje por Refuerzo, que interactúan entre sí mediante un juego (Teoría de Juegos), lo que permite expresar su sistema de emociones.

2) Simulación de la actividad emocional de la red. Codificamos en colores las interacciones entre agentes que se relacionan mediante el juego (dilema del prisionero), cuyas reacciones se van moldeando mediante Aprendizaje por refuerzo, donde las señales de aprendizaje provienen de las emociones de los agentes

3) Entrenamiento de Redes Neuronales en la detección de patrones de actividad propios de agentes conflictivos que introducimos en el juego. La aportación metodológica para la detección y clasificación de los agentes de la red será la utilización de Técnicas de Segmentación de imágenes.

1.4. Hipótesis y objetivos

Las hipótesis en que nos basaremos para desarrollar este trabajo se centran en que las interacciones locales en las que intervienen emociones generan un efecto macroscópico que podemos estudiar. Esto nos permite asociar ciertos patrones de comunicaciones a distintos tipos de agentes, los cuales pueden ser localizados en la red por el efecto que provocan (ver fig. 1.14). Los trabajos en que nos basamos se describen a continuación, para posteriormente desglosar los objetivos sobre los que vamos a trabajar.

1.4.1. Las emociones actúan como un mecanismo de aprendizaje

Sobre la naturaleza de las emociones, su funcionamiento como un refuerzo sobre la conducta, y su efecto sobre la toma de decisiones tenemos estudios como los de Rolls, Edmund T. (2014) [96]. Basándonos en ellos encontramos justificación para el diseño del agente mediante Aprendizaje por refuerzo.

Algunos autores como Ledoux (2020 [62]), resaltan el papel de cada emoción para resolver situaciones con una naturaleza muy específica. La relación de emociones como la ira/miedo con escenarios de amenaza y conflicto (Plutchick, R. (2001) [91]), así como el papel de la alegría en la conducta prosocial, justifica la modelización de relaciones emocionales como uno de los mecanismos de interacción que plantea la Teoría de Juegos.

En concreto el Dilema del Prisionero puede adaptarse bien. Podemos ligar cada emoción con el análisis estratégico que realiza el jugador antes de cada iteración.

1.4.2. El tipo de información genera distintos tipos de difusión

Artículos como el llevado a cabo en Twitter (Romero, D. M. et al. (2011) [97]) apuntan al fuerte efecto diferencial del tipo de información, sobre el grado de difusión. Otros estudios también apuntan en este mismo sentido (Berger, J. & Milkman, K. (2012) [16]) (también recogido en su libro: [15]), y en concreto, el papel fundamental que ejercen las emociones sobre muchas de nuestras decisiones (Berger, J. (2011) [14]). En este sentido tenemos estudios que buscan cuantificar este efecto como en Miller, M. et al. (2011) [75], sobre los hiperenlaces y cómo se van difundiendo las emociones.

Así mismo tenemos que otros resultados revelan que las noticias tristes se difunden más rápido que las buenas (Wu et al. (2011) [128]). Estudios clásicos llevados a cabo en Facebook y Twitter como el de Ferrara, E. & Yang, Z. (2015) muestran también que las emociones se contagian [31] y que resulta posible detectar la creación de grupos homogéneos en la red y de afinidad según las emociones.

Estudios clásicos como el de Framingham también apuntan a un fuerte efecto contagio (Fowler, J. H. & Christakis, N. A. (2008) [32]). Este efecto no sólo se circunscribe a nuestras relaciones sociales físicas. Algunos estudios revelan que las emociones tienen un fuerte efecto contagioso en las redes sociales *on-line* y plataformas de contacto virtual (Miller et al. (2011) [75]). Estudios llevados a cabo igualmente en Facebook (Kramer et al. (2012) [57] y Kramer et al (2014) [58]) apuntan en el mismo sentido. Otras plataformas como LiveJournal (Zafarani et al. (2010) [133]) también corroboran los análisis anteriores.

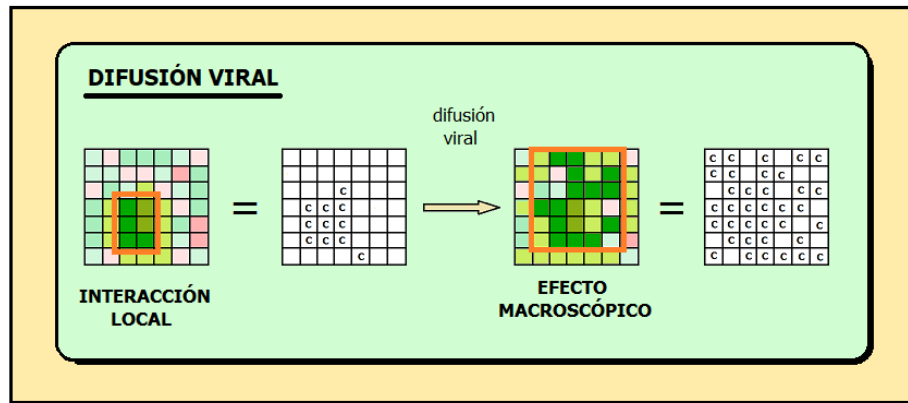


Figura 1.14: Cada emoción (interacción local) está muy relacionada con tipos específicos de conductas. Su expansión viral genera efectos macroscópicos detectables que podemos estudiar y asociar a la presencia de cierto tipo de agentes en la red. Nos interesa descubrir su localización.

■ HIPÓTESIS

H1 : Las emociones son capaces de generar procesos de difusión en una red social

H2 : Este tipo de interacciones locales genera efectos macroscópicos en la red social

■ OBJETIVO GENERAL 1

Estudio de las interacciones estratégicas entre agentes dotados de emociones, y la expansión viral de su influencia. Esto se traduce en unos objetivos específicos que se detallan a continuación.

OBJETIVOS ESPECÍFICOS

-OE1.1 : Plantear un modelo de difusión viral en el que las emociones jueguen un papel determinante. Emplear para ello juegos de Teoría de Juegos como mecanismo de interacción, y Aprendizaje por Refuerzo en los agentes para la toma de decisiones.

-OE1.2 : Llevar a cabo un estudio de la difusión de la cooperación y el papel que juegan la introducción de mutantes. Relacionar los resultados con el desempeño de agentes clásicos.

■ OBJETIVO GENERAL 2

Estudio de los fenómenos de difusión de comportamientos en una Red Social, donde los agentes guían sus decisiones por el impulso de sus emociones.

OBJETIVOS ESPECÍFICOS:

-OE2.1 : Traducir la simulación de los procesos de difusión en imágenes y entrenar con ellas redes neuronales especializadas en visión por computadora.

-OE2.2 : Identificación de los individuos influyentes, planteándolo como un problema de aprendizaje supervisado, empleando los datos generados en las simulaciones.

-OE2.3 : Determinar la viabilidad de generalizar el procedimiento de análisis a redes sociales más complejas.

1.5. Estructura de la tesis. - vol I

La metodología que hemos empleado no llevará a centrarnos en estos primeros capítulos en la parte de **Simulación**. Teniendo en cuenta esta estructura, la tesis comienza con la presentación del problema que pretendemos resolver.

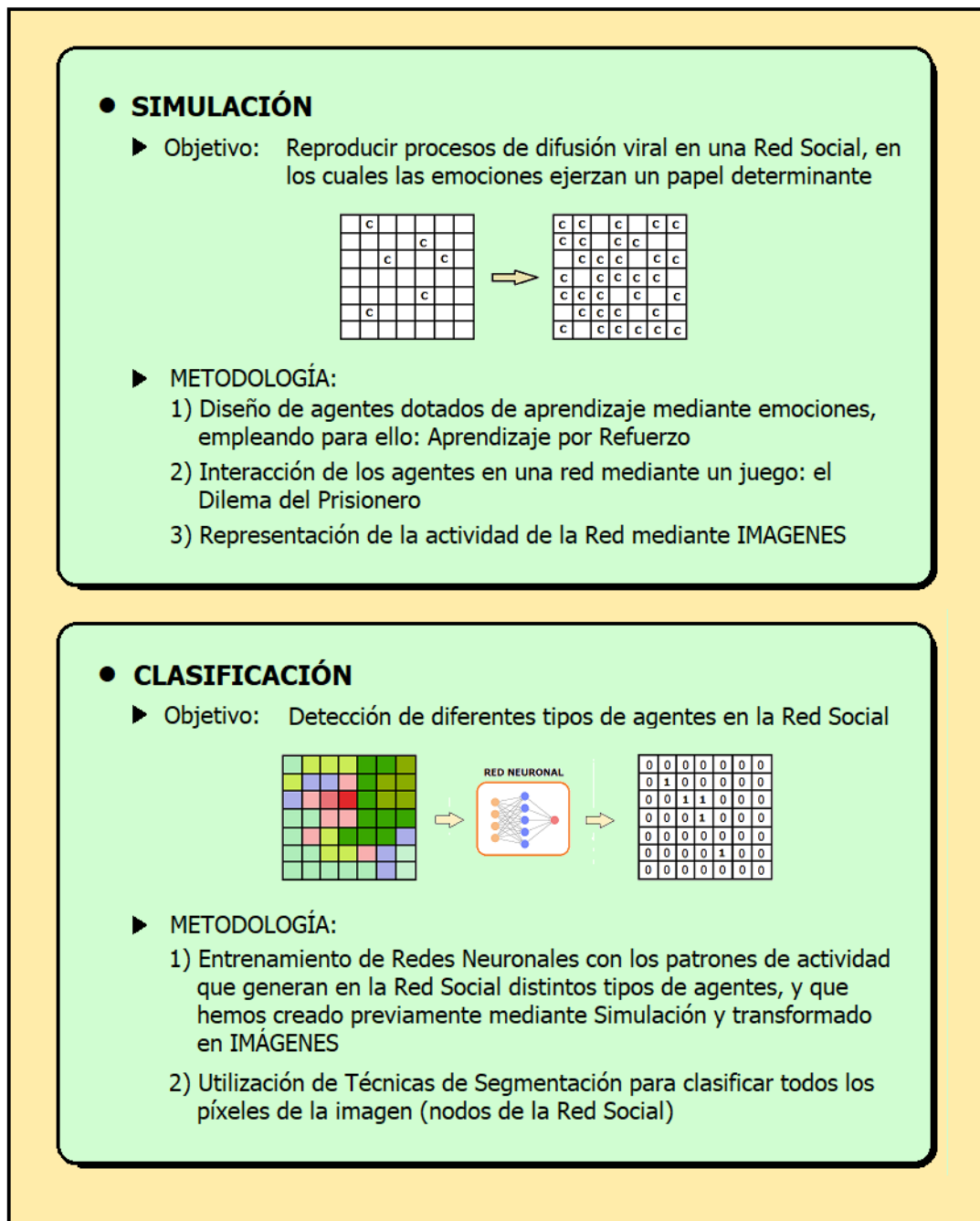


Figura 1.15: Esquema de los bloques que conforman la Tesis.

Tras abordar la implementación de agentes inteligentes para la generación de simulaciones, pasaremos al a centrarnos en el empleo de métodos de **Inteligencia Artificial** para el estudio del flujo de información en una red. Esta parte hará un uso intensivo de técnicas de Aprendizaje Automático (especialmente deep learning).

La organización por capítulos se detalla a continuación.

- Capítulo 1
Quizás planteamiento del modelo teórico. Modelo markoviano de la difusión emociones/-cooperación. Nos permite plantear nuestro modelo de difusión. Utilización de Aprendizaje por Refuerzo.
- Capítulo 2
Empezamos con el Diseño experimental. Utilizamos agentes mutantes que perturban la difusión normal en la Red Social. Difusión viral y resultados con agentes mutantes. Análisis comparado con agentes Tit-for-Tat.
- Capítulo 3
Redes estáticas. Utilizamos redes neuronales que procesan imágenes aisladas. Fundamentalmente Redes Neuronales Convolucionales.
- Capítulo 4
Redes dinámicas. Utilizamos redes neuronales que procesan secuencias de imágenes. Emplearemos estructuras complejas, empezando por Redes Neuronales Convolucionales diseñadas para secuencias de imágenes, y posteriormente emplearemos redes específicamente diseñadas para la segmentación de imágenes.
- Capítulo 5 En este capítulo presentamos los **resultados** del estudio y todos los aspectos relativos a la programación que hemos llevado a cabo en Simulación Social.
- Capítulo 6 En este capítulo esquematizaremos las **conclusiones** a las que hemos llegado, así como las pondremos en relación a los objetivos e hipótesis que habíamos establecido al inicio del trabajo.

El texto finaliza con algunos apéndices en los que se detallan algunos aspectos técnicos y de formalización de las herramientas que hemos empleado. Posteriormente aparecen los libros y artículos de investigación referenciados en el trabajo. Finalmente aparecen los símbolos y acrónimos, así como un glosario con los términos técnicos más utilizados.

Capítulo 2

Modelo Markoviano multiagente

Representaremos una red social mediante un grafo. Es decir: $G = (V, E)$, donde $V = \{v_1, \dots, v_N\}$ es el conjunto de vértices que representarán a los agentes, y $E \subset V \times V$ es el conjunto de aristas que representa los enlaces entre ellos. Analizaremos la expansión viral por la red del comportamiento que adopten los agentes al enfrentarse a algún tipo de dilema social iterado que plantearemos. En concreto, cada agente i ($i = 1, \dots, N$) interactuará con su vecindario $N(i)$ más cercano, donde: $N(i) = \{v_j \in V; \{v_i, v_j\} \in E\}$.

Para modelizar la situación acudiremos a los Procesos de Decisiones de Markov (MDP) finitos. Un proceso de este tipo queda definido cuando se especifican cada uno de los elementos que se mencionan a continuación: Un espacio finito de estados S , un espacio finito de acciones A , una función de probabilidad que determina la secuencia de transiciones entre estados y una función de recompensas R . Notado: $M = (\mathcal{S}, \mathcal{A}, P, R)$. Para adaptarlo a caso multiagente supondremos que los agentes aprenden independientemente unos de otros y modelizan su vecindario como una parte del entorno (ver representación: fig. 2.1).

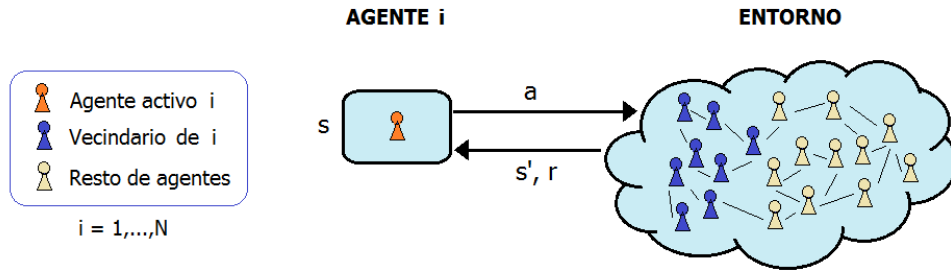


Figura 2.1: Representación del tipo de interacción entre los individuos de la Red Social.

La dinámica de la red consistirá en que, en cada etapa $k = 0, 1, 2, \dots$, cada agente i ($i = 1, \dots, N$) se encuentra en un cierto estado ($s_k^i = s \in \mathcal{S}$) y lleva a cabo una misma acción con todos sus vecinos ($a_k^i = a \in \mathcal{A}$) en el dilema que se le plantea. En función de la reacción de su entorno transicionará a un nuevo estado o permanecerá en el que estaba ($s_{k+1}^i = s' \in \mathcal{S}$). Y fruto de ello recibirá una cierta recompensa $r_{k+1}^i = r$, lo que induce un proceso de aprendizaje, que irá modificando su regla de decisiones.

La propiedad Markoviana de proceso sin memoria se concreta bajo el esquema de interacción que planteamos en la siguiente relación:

$$p(s_{k+1}^i \mid s_k^i, a_k^i, a_k^{-i}) = p(s_{k+1}^i \mid s_k^i, a_k^i, \bar{a}_k^i)$$

donde: a_k^{-i} representa las acciones del resto de agentes, y \bar{a}_k^i representa la acción más frecuente del resto de agentes (empleando una *tie-breaking rule* en cada caso concreto).

Formalmente, tenemos que los estados que consideraremos del agente son un elemento de un conjunto \mathcal{S} . Las transiciones entre un estado u otro vienen regidas por una función $P : \mathcal{S} \times \mathcal{A} \times \mathcal{A}^{N(i)} \times \mathcal{S} \rightarrow [0, 1]$ de tal manera que a cada (s, a, \bar{a}, s') le asignará la probabilidad $p(s, a, \bar{a}, s')$. De manera similar la función de recompensa se define como $R : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$ de tal manera que a cada par estado-acción (s^i, a^i) del agente i , junto a la acción más frecuente de su vecindario (\bar{a}^i) , se le asigna un determinado valor $R(s^i, a^i, \bar{a}^i)$.

Cada agente elegirá una *política* que asocie a cada uno de sus estados una acción $\pi : \mathcal{S} \rightarrow \mathcal{A}$ (o en el caso de que asocie una distribución de probabilidad $\pi : \mathcal{S} \rightarrow \Sigma(\mathcal{A})$, donde $\Sigma(\mathcal{A})$ es el conjunto de distribuciones de probabilidad en \mathcal{A}).

Se dice que una política π es óptima si maximiza el valor descontado esperado acumulado a lo largo del tiempo de un agente. Si tomamos un valor $\gamma \in (0, 1)$ como factor de descuento, tenemos que maximizar para cada par (s, a) :

$$q_\pi^i(s, a) = \mathbb{E}_\pi \left[\sum_{k=1}^{\infty} \gamma^k \cdot R(s_k^i, a_k^i, \bar{a}_k^i) \mid s_0 = s, a_0 = a \right] = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a) \cdot [r + \gamma v_\pi(s')]$$

que refleja el valor de un determinado estado s , cuando se toma una determinada acción a siguiendo la regla π , y que viene a ser la ecuación de Bellman. Cuando no conocemos las probabilidades manejaremos su estimación: $Q_\pi^i(s, a)$. Para este fin se utilizan algoritmos iterativos empleados con simulaciones del sistema. En concreto: el algoritmo Q-Learning (Watkins, C. J. & Dayan, P. (1992)) [127]. Si la tasa de aprendizaje es $\alpha \in (0, 1)$, la expresión queda:

$$Q_{k+1}^i(s, a) = Q_k^i(s, a) \cdot (1 - \alpha) + \alpha \cdot [R(s, a, \bar{a}_i) + \gamma \cdot \max_{a' \in \mathcal{A}} Q_k^i(s', a')]$$

La estrategia ϵ -greedy permite implementar las decisiones basadas en la función Q , y a la vez mantener cierta exploración del entorno. Para ello se emplea una elección aleatoria uniforme entre las acciones $U(\mathcal{A})$ con una pequeña probabilidad $\epsilon \in (0, 1)$. Es decir:

$$\pi^i(s) = \begin{cases} \arg \max_{a \in \mathcal{A}} Q^i(s, a) & \text{con probabilidad } 1 - \epsilon \\ U(\mathcal{A}) & \text{con probabilidad } \epsilon \end{cases}$$

Tipo de interacción.

Como mecanismo de interacción emplearemos el conocido como **Dilema del Prisionero**. El Dilema del Prisionero consiste en la elección a ciegas (desconociendo la estrategia del otro jugador) en

cada ronda de una acción Cooperate/Defect. Pese a que en multiagente pueden aparecer dificultades de aprendizaje (Sandholm, T. & Crites, R. (1996) [101]), en entornos sencillos Q-Learning ofrece buenos resultados tal y como hemos comprobado.

Esto se traduce en que los conjuntos que vamos a manejar son:

- Conjunto de Estados: $\mathcal{S} = \{s_1 : \text{alegría}, s_2 : \text{ira}, s_3 : \text{miedo}, s_4 : \text{tristeza}\}$
- Conjunto de Acciones: $\mathcal{A} = \{a_1 : \text{Cooperar}, a_2 : \text{No Cooperar}\}$

Como regla de aprendizaje emplearemos técnicas de Aprendizaje por Refuerzo (Sutton, R. & Barto, A. (2018) [115]). En concreto emplearemos el algoritmo Q-Learning (Watkins, C. J. & Dayan, P. (1992) [127] y Tsitsiklis, J. (1994) [121] sobre la convergencia de este algoritmo). En concreto, la función que emplearemos para gobernar la conducta de los agentes será la llamada función $Q^i(s, a)$ $i = 1, \dots, N$, empleada en esta disciplina. Los valores de esta función dependen del estado en el que se encuentra el agente (s) y de la acción (a) que lleva a cabo posteriormente.

La estructura de los pagos de este juego, se deriva de cada combinación posible de acciones (fig. 2.2).

		J2	
		Defect	Cooperate
J1	Defect	P, P	T, S
	Cooperate	S, T	R, R

Figura 2.2: Matriz de pagos del Dilema del Prisionero.

Las siglas obedecen a T (*Temptation*), R (*Reward*), P (*Punishment*), S (*Sucker*), y la relación entre los términos y algunos valores usuales son:

$$\underbrace{T(\text{Temptation})}_5 > \underbrace{R(\text{Reward})}_3 > \underbrace{P(\text{Punishment})}_1 > \underbrace{S(\text{Sucker})}_0$$

Las **transiciones entre estados** se representan por el grafo que aparece en la figura 2.3. El paso de un estado a otro se lleva a cabo teniendo en cuenta la estrategia de la mayoría siguiendo las reglas de la figura.

Cada tupla del tipo (*estado inicial, acción agente 1, acción agente 2, estado final*) tiene una probabilidad asociada. Como ejemplo tenemos las transiciones desde el estado alegre.

El sistema de transiciones emocionales permite que el agente siente alegría por ser correspondido, tristeza por dañar a otros o sentir la ira por la traición. Este conjunto de transiciones las adoptamos al conjunto de situaciones que pueden darse en el Dilema del Prisionero.

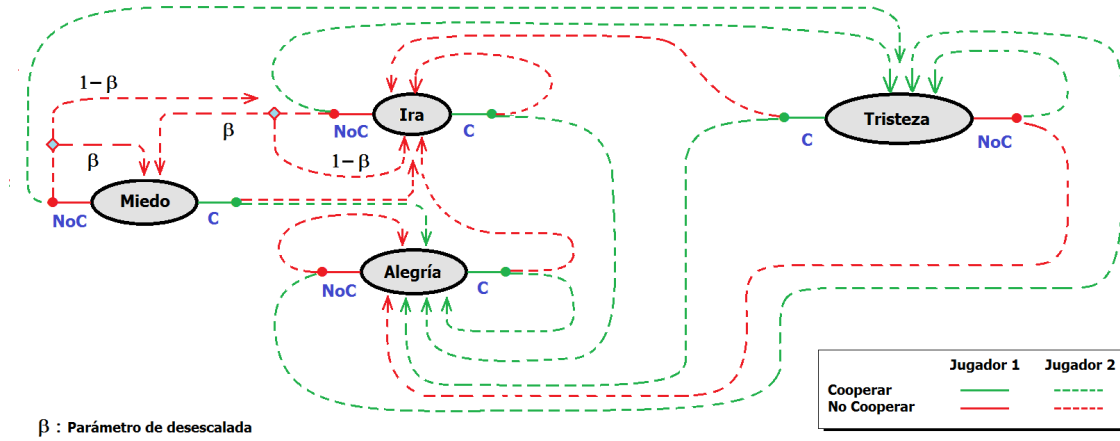


Figura 2.3: Modelo de dinámica de respuestas emocionales en un agente dotado de 3 emociones (ira, alegría y tristeza). Las interacciones están basadas en el Dilema del Prisionero.

Algunas de las transiciones reciben un refuerzo, que depende del estado en el que se encuentra el agente, y su acción y la de su vecindario. Esto permite implementar una reacción sensible al contexto. La función de recompensa quedaría tal como aparece a continuación (para más detalles ver apéndice).

$$R(s_k, a_k, \bar{a}_k) = \begin{cases} r & \text{si : } s_k = 'alegría' & a_k = C & \bar{a}_k = C \\ r & \text{si : } s_k = 'ira' & a_k = NoC & \bar{a}_k = NoC \\ r & \text{si : } s_k = 'miedo' & a_k = C & \bar{a}_k = C \\ r & \text{si : } s_k = 'tristeza' & a_k = C & \bar{a}_k = C \\ 0 & \text{en otro caso} \end{cases}$$

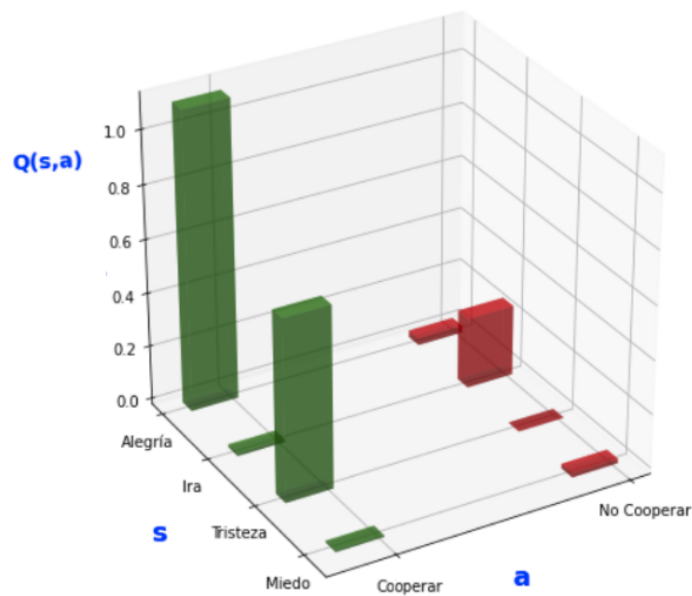


Figura 2.4: Función de decisión del agente.

El objetivo final del aprendizaje en la situación particular que estamos analizando es la de que el agente sea capaz de reaccionar de manera adecuada en la situación emocional en la que se encuentre. Emplearemos en concreto métodos de aprendizaje para que el agente sea capaz de interiorizar una función como la de la imagen [2.4](#).

valor esperado ($\sum_{j=1}^n p_{ij} \cdot g_{ij}(u_k)$)

Sobre los POMDP. Vemos algunos de los papers sobre métodos de resolución. Es decir vamos a tratar el caso en el que el ENTORNO es Markoviano, pero el acceso que tiene el agente a dicha información es limitado, y por tanto no puede establecer unívocamente el estado del sistema a partir de las observaciones que es capaz de recoger.

Uno de los artículos en abordar esta cuestión (Jaakkola, T. et al. (1994) [51]). La situación es como la planteada por los Modelos de Markov Ocultos (HMM - *Hidden Markov Model*), en los que el entorno aparece como no markoviano para el observador.

Tenemos por otra parte (Littman, M. et al. (1995) [67]). Cuando el número de estados es grande, aparecen problemas de escalabilidad. Los autores tratan formas de abordar esta situación.

Una recopilación más exhaustiva de métodos de solución la encontramos en la publicación de Cassandra, A. R. (1998) [22].

Respecto del tratamiento cuando no se recurre al estudio del valor de los estados tenemos el trabajo de Singh, S. P. et al. (1994) [112]. Entre otros temas trata el rendimiento que ofrecen algoritmos como TD(0) y Q-Learning a la hora de enfrentarse a POMDP.

Entre los mecanismos alternativos que se pueden encontrar en la literatura para desambiguar la clase de estado en la que se encuentra el sistema, tenemos la utilización de alguna forma de memoria (McCallum, L. A. (1994) [72] y McCallum, L. A. (1995) [73]).

Desde el lado de la ingeniería se ha dado un tratamiento distinto al problema de la difusión de opiniones en las redes sociales Acemoglu, D. & Ozdaglar, A. (2011) [1] y Acemoglu, D. et al. (2013) [2].

Un tutorial sobre la resolución de este tipo de problemas (Littman, M. L. (2009) [66] - Enlace non-comercial use).

Capítulo 3

Viralización de comportamientos en una Red Social

Para el desarrollo del proyecto hemos procedido a crear una sociedad artificial, para estudiar los patrones de comunicaciones que emergen entre los agentes que forman parte de ella. Esto implica centrarnos inicialmente en una parte experimental, mediante un programa creado para tal efecto.

La disposición de los agentes es en una cuadrícula: una rejilla regular que admite una representación como una Red Social. Como mecanismo de interacción entre ellos emplearemos juegos iterados, y en concreto el conocido como 'Dilema del Prisionero'. En la medida en que existen intereses enfrentados, es posible encontrar situaciones de conflicto, pero también deja lugar para la colaboración.

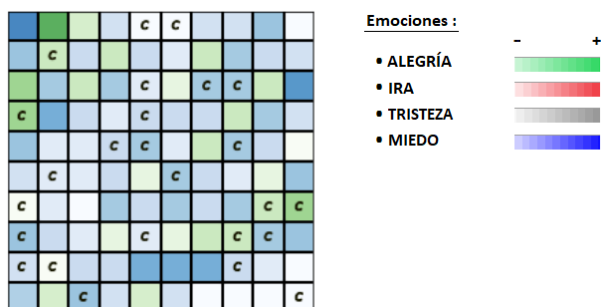


Figura 3.1: Captura de la aplicación para Simulaciones (Prueba con un Grid de 10x10).

En cuanto a la naturaleza de los agentes, inicialmente manejamos agentes dotados de emociones, que emplean aprendizaje por refuerzo para adaptarse a los estímulos que recibe de los agentes de su entorno. Aunque se pueden incluir otro tipo de agentes para el estudio de los efectos sobre la red, siendo en este caso autómatas que implementan algunas estrategias clásicas (como Tit-for-Tat).

Bajo estas premisas, procedemos a estudiar los objetivos principales que nos hemos marcado, empleando para ello las técnicas que se mencionan a continuación.

3.0.1. Simulación: Estudio de las interacciones entre agentes

- Mediante la aplicación que se ha desarrollado, es posible generar abundantes datos mediante simulación. Uno de los objetivos consistiría en reproducir el tipo de interacciones que se observan de manera cotidiana en una Red Social.

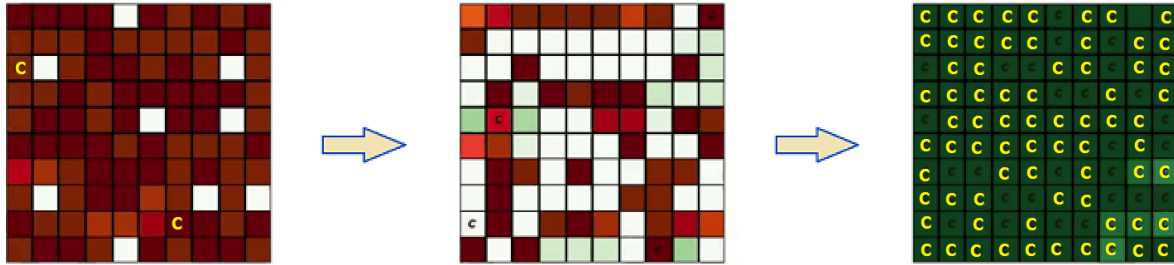


Figura 3.2: *Evolución del aprendizaje con Agentes Emocionales, y difusión viral al final.*

El programa nos permite desarrollar una sociedad artificial en la que los agentes se relacionan siguiendo ciertos mecanismos de interacción. En nuestro caso, como señalamos previamente, nos servimos de la Teoría de Juegos, y en concreto de juegos iterados para representar la forma de relacionarse. La dinámica de las relaciones que se establecen entre ellos se puede visualizar tal y como aparece en la fig. 3.2.

Construcción de una función de recompensa.

Algoritmos tradicionales de Aprendizaje por Refuerzo (*Reinforcement Learning (RL)*) muestran un rendimiento deficiente en ciertas situaciones. Es por ello por lo que algunos autores han propuesto diversas metodologías para la construcción de una función de recompensa, lo más adecuada posible para la tarea que se pretende llevar a cabo y que permita la aceleración del aprendizaje (Mataric, M. J. (1994) [70]). En este mismo sentido se han llevado a cabo estudios sobre el tipo de modificaciones en la recompensa que conservan la optimalidad de la mejor estrategia en el problema de partida (Ng, Andrew et al. (1999) [81]).

Respecto a la naturaleza de la recompensa podemos distinguir entre recompensa intrínseca y la recompensa externa que se recibe del entorno. Respecto a la primera el trabajo pionero establece el marco de trabajo para profundizar en un agente guiado por motivaciones internas (Singh, S. et al. (2009) [110]). Los mismos autores ahondan (Singh, S. et al. (2010) [111]) en la naturaleza evolutiva de las recompensas intrínsecas.

Sobre el aprendizaje en casos no markovianos (POMDP) tenemos el trabajo de Singh, S. et al. (1994) [112]. (Para un tratamiento más moderno ver: Krishnamurthy, V. (2016) [59]).

Convergencia a la Cooperación en Dilemas Iterados

Los estudios de la difusión de la cooperación en una sociedad son abundantes. Se han empleado diversos mecanismos para promover la aparición de este tipo de estrategias. Podemos resaltar los estudios clásicos sobre reciprocidad directa (Trivers, R. (1971) [120], Axelrod, R. (1984) [6], Nowak, M. & Sigmund, K. (1992) [84], Nowak, M. & May, R. M. (1992) [83]). El esquema que se suele mantener es el de emplear como espacio de interacción una rejilla regular (*grid*) donde las interacciones se dan entre celdas adyacentes. Estudios posteriores incorporan otro tipo de espacio, como puede ser distintos tipos de redes que interconectan a los individuos (Ohtsuki, H. et al. (2006) [85], Santos, F. C. & Pacheco, J. M. (2006) [102]).

En cuanto al empleo de redes neuronales entrenadas con retropropagación destaca el trabajo de Leibo, J. L. et al. (2017) [63]. En concreto trabajan con varios tipos de dilemas diseñados específicamente para el estudio de equilibrios en Dilemas Sociales Secuenciales. Aquí el agente se enfrenta a un entorno que sólo puede observar parcialmente. Emplean Deep Q-Networks. El entrenamiento es por lotes, por lo que de cara a diseñar una red social no sea el más apropiado.

Sobre este mismo diseño tenemos la adhesión como forma de promover la cooperación (Yuan, Y. et al. (2022) [132]).

El algoritmo Q-Learning ha sido estudiado en multi-agente. Pese a que las condiciones que garantizan la convergencia del algoritmo no se cumplen, se ha aplicado con éxito en entornos sencillos. Si tratamos de su aplicación en juegos matriciales algunos estudios revelan la dificultad para generar la cooperación (Wunder, M. et al. (2010) [130]). Estudios más generales tenemos el de Bloembergen, D. et al. (2015) [17].

Empleo de recompensas intrínsecas en Dilemas Iterados

El empleo de Teoría de Juegos y, en concreto, el Dilema del Prisionero como mecanismo de in-

teracción entre agentes ha impulsado el estudio de los mecanismos mediante los cuales se promueve la estrategia cooperativa. El empleo de recompensas intrínsecas es uno de dichos mecanismos.

Uno de los trabajos es el de Eccles, T. et al. (2019) [28] que emplea la reciprocidad como recompensa intrínseca. En este estudio coexisten dos tipos de agentes: innovadores e imitadores. Ambos son entrenados mediante A3C (algoritmo *Asynchronous Advantage Actor-Critic*) con una red neuronal profunda. Los primeros aprenden del entorno y los imitadores se ven influidos por el nivel de sociabilidad de los innovadores.

Entre los mecanismos empleados para impulsar la cooperación destacan: la reciprocidad, el cumplimiento de normas, las emociones o el efecto de la red.

Empleo de emociones como recompensa intrínseca

Estudios que emplean emociones en los agentes para jugar al IPD tenemos el de Bazzan, A. et al. (2001) [9], basado en el modelo de emociones de OCC. Es básicamente un modelo evolutivo. Toma como referencia el trabajo de Nowak, M. A. & May, R. M. (1992) [83] y cada celda se sustituye por la que mejores resultados obtuvo entre las vecinas implementando un sistema emocional basado en reglas. Dicho trabajo es expandido posteriormente (Bazzan, A. et al. (2002) [10]) para dotar a algunos de los agente de sentimientos morales hacia agentes del mismo grupo social. Los agentes altruistas, que inicialmente no tienen un buen desempeño, acaban mejorando sustancialmente.

También tenemos (Bazzan, A. et al. (1997) [3]), donde introduce la generosidad hacia otros o la culpabilidad por haber sido injusto con otros y analiza la evolución de la cooperación en el Dilema del Prisionero Iterado (parece más en el sentido de autómatas sin aprendizaje). Posteriormente lo tenemos desarrollado en un esquema evolutivo (Bazzan, A. et al. (2002) [10]). Se diseñan algunos agentes con sentimientos morales hacia agentes del mismo grupo (generosidad y culpabilidad por actuar injustamente).

Utilizando el Aprendizaje por Refuerzo en lugar del enfoque evolutivo tenemos el siguiente artículo, en el que se emplea el IPD para comprender el papel de las interacciones locales a la hora de inducir la expansión de la cooperación en una sociedad (Bazzan, A. et al. (2011) [12]). Aquí se introduce otro mecanismo: los lazos sociales. En concreto se analiza el papel de pertenecer a una jerarquía (una estructura organizativa preexistente) o una coalición (cuando ésta emerge de las interacciones de los agentes durante el juego).

Empleando idéntico modelo como referencia tenemos el trabajo de Sequeira, P. et al (2011) [108], en el que utilizan 4 dimensiones de evaluación de elementos del entorno (reward features) que combinan linealmente optimizando los pesos para estudiar la adaptación óptima al entorno. Posteriormente llevan a cabo un estudio más profundo sobre el tipo de diseño de la recompensa intrínseca (Sequeira, P. et al. (2014) [109]).

Efecto de las emociones en la evolución de la cooperación en dilemas sociales, mediante unos agentes con orientación social y otros más egoístas, empleo de emociones surgidas de la reciprocidad (Chen, W. et al. (2021) [23]).

Efecto de las emociones en dilemas espaciales con participación voluntaria (Wang, L. et al.

(2018) [124]).

Sobre la imitación de emociones, del perfil emocional, en lugar de estrategias como vía de elevar el bienestar social en juegos espaciales (Szolnoki, A. et al. (2011) [118]).

Imitación de las emociones en IPD con extorsión (Quan, J. et al. (2021) [94]).

Se estudia el aprendizaje del surgimiento de la cooperación utilizando 2 capas de aprendizaje que dotan al agente de capacidades cognitivas y emocionales (Yu, C. et al. (2015) [131]). Se utilizan diversos algoritmos, y los agentes interactúan en diversos espacios: un grid cuadrícula y diferentes redes sencillas.

Algunas observaciones últimas a incorporar:

Distinguimos primero entre si es una teoría de Juegos evolutivos (Ver por ejemplo Perc, M & Szolnoki, A. (2010) [87] o también: Szabo, G. & Fath, G. (2007) [117] sobre los juegos evolutivos en grafos).

Sobre los problemas de coordinación en el aprendizaje, por ejemplo en los métodos que emplean Q-Learning (Fulda, N. & Ventura, D. (2007) [36]).

Sobre el equilibrio correlado (Hines, G. & Larson, K (2012) [46]).

Sobre la detección del tipo de oponentes a los que uno se enfrenta en juegos repetidos (Hernandez-Leal, P. & Kaisers, M (2017) [44]).

Sobre un aprendizaje más rápido de los agentes (Elidrisi, M. et al. (2014) [29]).

Algoritmo Pepper para juegos matriciales (Crandall, J. W. (2012) [25]).

Sobre la diversidad social y la promoción de la cooperación en juegos en el dilema del Prisionero espacial (Perc, M. & Szolnoki, A. (2008) [88]).

Hacia la cooperación en PD secuenciales con el enfoque Deep RL multiagente (Wang, W. et al. (2018) [125]). Utiliza como mecanismo de interacción, algunos juegos con una filosofía similar a la del Dilema del Prisionero, desarrollados previamente por Leibo, J. et al. (2017) [63]. El procedimiento que siguen los autores es el de generar diversas estrategias (*policies*) con distintos niveles de cooperación en una primera fase *off-line*, y entrenar una Red Neuronal para ser capaz de detectarlas. Posteriormente, en una fase *on-line*, el agente se enfrenta al juego seleccionando su estrategia basándose en la habilidad previamente enseñada de detectar la estrategia de su oponente.

Capítulo 4

Estudio de los agentes influyentes en una Red Social I

Ahora detallamos el plan para desarrollar un clasificador que nos permita encontrar los individuos que ejercen un efecto sobre la red. Los inputs que procesaríamos serían el contenido emocional de los agentes (lo que podría obtenerse de una red a partir de análisis de sentimientos). En nuestro caso al obtener los datos por simulación, podemos identificar cada elemento en todo momento, lo que nos permite transformar el difícil problema en uno de aprendizaje supervisado.

4.0.1. Estructura de Red Neuronal 1: CNN

En primer lugar probaremos la estructura CNN (*Convolutional Neural Network*) aplicada a la segmentación de imágenes. Su origen se encuentra en el Neocognitron (Fukushima, K. & Miyake, S. (1982) [35] y aplicación a detección de patrones Fukushima, K. (1988) [34]), que se basaba en hallazgos sobre el procesamiento de la información visual (Hubel, D. H. & Wiesel, T. N. (1962) [49]). Posteriormente se desarrolló un modo eficiente de aprendizaje por retropropagación del error (no con pesos asignados manualmente) (Le Cun, Y. et al. (1989) [61])

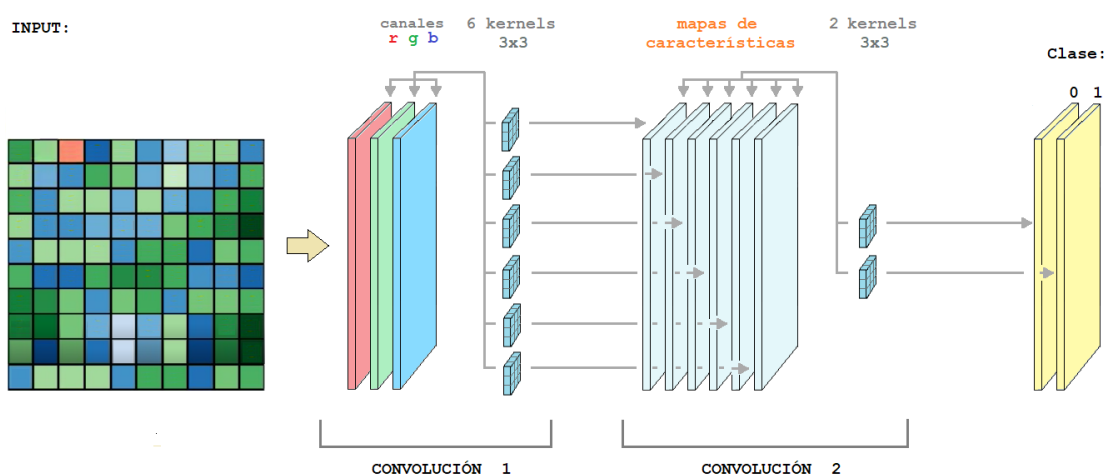


Figura 4.1: *Arquitectura de Red Neuronal 'CNN' (Convolutional Neural Network).*

Partimos de secuencias de la dinámica que se genera en la red con los agentes llevando a cabo jugadas del Dilema del Prisionero, y pretendemos que la red aprenda a clasificar cada uno

de los píxeles de la imagen, que representan a cada uno de los jugadores de la red. En la figura 4.1 podemos observar el esquema con el procesamiento que se lleva a cabo con la representación que hacemos de la actividad en la Red Social.

En primer lugar convertimos la imagen en 3 canales (RGB) que contienen codificada la información visual, posteriormente se generan 6 canales (tras aplicar los 6 kernels) y en la última capa de procesamiento generamos las dos capas últimas. En ellas aparece la prueba de clasificación en 2 clases.

4.0.2. Estructura de Red Neuronal 2: U-Net

La siguiente estructura que hemos estudiado es la llamada U-Net (Ronneberger, O. et al. (2015) [98]). La combinación de los inputs de capas anteriores con el resultado de aplicar reiteradamente capas de Convolución, permite que la red aprenda patrones locales de interacción y, a su vez, sea capaz de detectar macroestructuras en las imágenes.

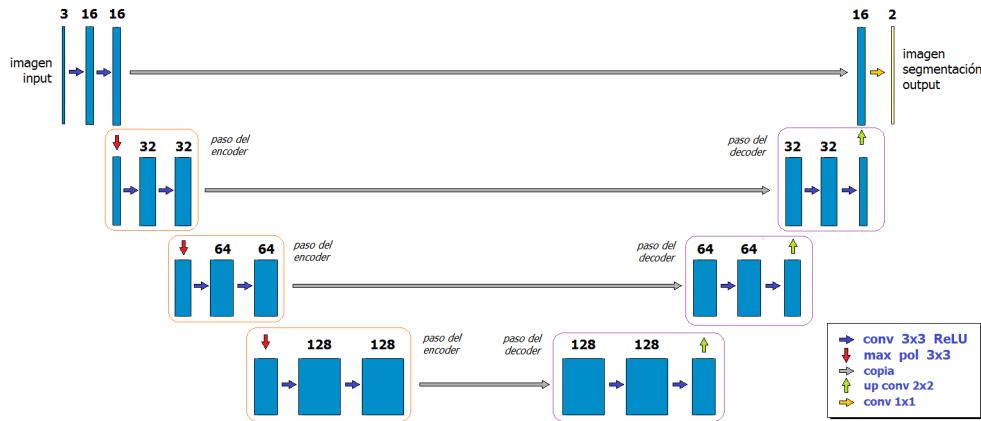


Figura 4.2: *Arquitectura de Red Neuronal 'U-Net'.*

La representación que aparece en la figura 4.2 refleja el procesamiento de imágenes. Se aplican reiteradamente Convoluciones y se integra la información previa de capas anteriores junto al procesamiento posterior.

Cuando el número de estados y de acciones es reducido, la **representación tabular** de la función de valor ($J(x)$ o $Q(x, u)$) surge de manera natural, tal y como hemos estado utilizando hasta ahora. Sin embargo, cuando el sistema bajo estudio tiene un gran número de estados o acciones, la utilización de matrices es una vía computacionalmente muy costosa. En estas condiciones lo adecuado consiste en optar por una **representación compacta** (ver capítulo 2), empleando aproximadores funcionales.

De manera simplificada nos enfrentamos al problema de construir una función $y = f(x, \theta)$ a partir de unos datos de entrenamiento (x_i, y_i) , de tal manera que dicha función se ajuste a ellos lo mejor posible. En nuestro caso, dada una estrategia μ , los valores podrían ser $(i, J^\mu(i))$ $i \in X$, obtenidos mediante experimentación o simulación.

Este esquema se traduce en encontrar el vector de parámetros θ^* que minimice una cierta función de pérdida (\mathcal{L}) que cuantifica de alguna manera las discrepancias entre lo observado (y) y lo predicho ($f(x, \theta)$). Es, decir:

$$\theta^* = \arg \min_{\theta \in \Theta} \mathcal{L}(\theta)$$

La elección del modelo a utilizar dependerá en gran medida del tipo de problema a tratar, pero el uso de Redes Neuronales es lo más común en la actualidad. Una de las razones reside en que es un aproximador universal (Hornik, et. al (1989) [48]), y la utilización de múltiples capas de neuronas puede ajustar señales muy complejas (Cybenko, G. (1989) [26]). Es por lo que procedemos a abordar este tema en las secciones que siguen.



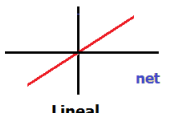
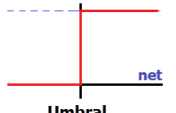
4.1. Redes Neuronales

La idea principal detrás de las redes neuronales es la de componer recursivamente una función, para generar progresivamente una señal de respuesta cada vez más compleja.

Utilizando la siguiente notación (Murphy, K. P. (2022) [79]), sea $f_l(\mathbf{x}) = f(\mathbf{x}, \theta_l)$, donde $l = 1, \dots, L$ la función paramétrica de ajuste sería:

$$f(\mathbf{x}, \boldsymbol{\theta}) = f_L(f_{L-1}(\dots(f_1(\mathbf{x}))\dots))$$

La función $f_l(x)$ se denomina **función de activación** y actúa como unidad de procesamiento. Si θ_l son los parámetros de la capa l , llamemos W_l a los pesos que afectan a una neurona en particular. Entonces la función de activación actúa transformando el producto de las entradas (*inputs*) \mathbf{x} por los pesos W_l más una constante arbitraria b . En la figura 4.3 podemos observar algunas de las más habituales.

FUNCIÓN de ACTIVACIÓN				
	Tangente hiperbólica	Logística	Lineal	Umbral
Expresión de la función	$f(\mathbf{x}) = \tanh(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$	$f(\mathbf{x}) = \frac{1}{1 + e^{-x}}$	$f(\mathbf{x}) = x$	$f(\mathbf{x}) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$

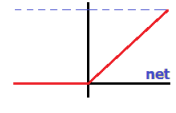

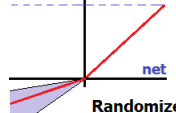
FUNCIÓN de ACTIVACIÓN			
	ReLU	Leaky ReLU	Randomized ReLU
Expresión de la función	$f(\mathbf{x}) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$	$f(\mathbf{x}) = \begin{cases} \alpha x & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$ donde $\alpha \leq 1$	

Figura 4.3: Algunas funciones de activación usuales (donde: $net = W \cdot input + b$).

Las interconexiones entre estas unidades de procesamiento quedan apiladas en forma de capas, tal y como refleja la figura 4.4.

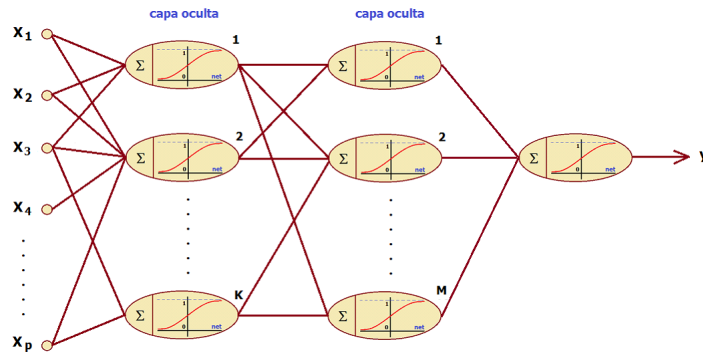


Figura 4.4: Representación gráfica del procesamiento de información que realiza una Red Neuronal. Se han empleado unos inputs de dimensión 'p': $\mathbf{x} = (x_1, x_2, \dots, x_p)$ y dos capas ocultas de neuronas. La neurona final genera un valor y . En todos los casos se ha empleado una función de activación tangente hiperbólica.

Tras disponer de una modelo, el paso siguiente consiste ajustar los parámetros para conseguir que la señal que genera la red neuronal produzca los menores errores de predicción, dados los valores respuesta observados. Este proceso recibe el nombre de **entrenamiento**.

4.2. Entrenamiento de una red neuronal

Dados un conjunto de datos muestrales (x_i, y_i) , buscamos ajustar los parámetros de la red neuronal dada por $f(\mathbf{x}, \theta)$. Los parámetros óptimos θ^* son aquellos que hacen que las diferencias $(y_i - f(x_i))$ sean mínimas en algún sentido. La forma de tratar estas diferencias vendrá dada por una función g , que suele consistir en promediar los valores cuadráticos de los errores. Esta función es la llamada **función de pérdida** \mathcal{L} .

Empleando métodos tradicionales de optimización, tenemos que el óptimo se encontrará actualizando los valores de los parámetros en el sentido dado por el mayor gradiente. El proceso esquemáticamente consistiría en:

- 1) Especificación de la función de pérdida :
 $\mathcal{L}(\theta) = g(y - (f(\mathbf{x}, \theta)))$
- 2) Cálculo del gradiente :
 $\nabla \mathcal{L}(\theta)$
- 3) Actualización de parámetros de la red:
 $\theta = \theta - \gamma \cdot \nabla \mathcal{L}(\theta)$

El procedimiento se traduce en un desplazamiento por la curva de error guiados por el valor del gradiente. En la figura 4.5 podemos observar una representación del proceso de optimización.

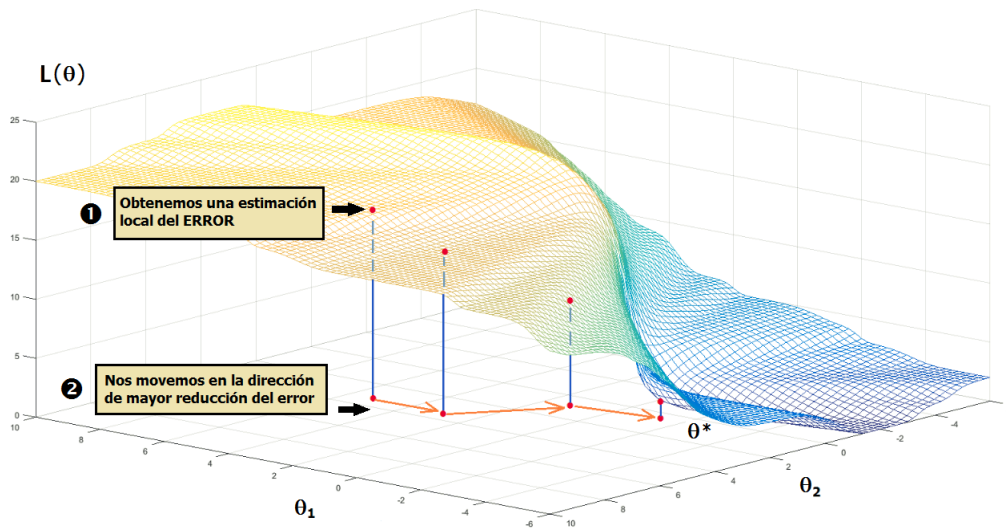


Figura 4.5: Obtención del vector de parámetros óptimo θ^* .

En la medida en que utilizamos el gradiente para avanzar en la actualización de los parámetros de la red, se pueden presentar algunos problemas como la desaparición del gradiente o su crecimiento explosivo que convierte el aprendizaje en muy inestable. Existen métodos que permiten corregir algunas de estas deficiencias y ayudar a que el algoritmo de optimización encuentre el óptimo (ver: Goodfellow, I. et al (2016) [38]).

4.3. Arquitecturas de Redes Neuronales

El origen de las redes neuronales se retrotrae a un artículo de McCulloch, W. & Pitts, W. (1943) [74], en el que los autores plantean un modelo de computación basado en el funcionamiento de las neuronas biológicas. Básicamente se trata de una función umbral que se activa (valor 1) cuando la suma de los inputs por unos ciertos pesos más una constante superan un cierto valor (ver fig. 4.6). Sería algo más tarde cuando se propondría el Perceptrón (Rosenblatt, F. (1958) [99]), como una unidad computacional dotada de aprendizaje, que varía sus parámetros para adaptarse a la información que se le proporciona.

Sus limitaciones en el procesamiento de información, tal como demostraron Minsky, M. & Seumour, P. (1969) [69], hicieron que la comunidad científica perdiera el interés en este tipo de modelos. Sin embargo el descubrimiento, años más tarde, de construcciones más complejas y la forma de entrenarlas óptimamente provocó el resurgimiento de este área de investigación.

En el siguiente esquema se recogen algunas de las principales arquitecturas de Redes Neuronales, y los tipos de datos cada vez más complejos que son capaces de procesar.

- **MLP** (*Multi-Layer Perceptron*). Este tipo de red se construye conectando neuronas y apilando capas, todas ellas con conexiones exclusivamente hacia adelante (*Feed Forward Network*), y que genera al final una señal de respuesta. Se emplea como un aproximador funcional y se entrena mediante retropropagación de error, tal y como vimos.
Los **autoencoders** son un tipo especial de red multicapa alimentada hacia delante que es entrenada para generar una salida lo más aproximada posible a la entrada que se le proporciona (Bourlard, H. & Kamp, Y. (1988) [19]). Esta red está diseñada limitando la capacidad de las capas ocultas internas, de manera que impida a la red caer en el aprendizaje obvio de reconstruir de manera perfecta la señal de entrada. Esto permite a la red funcionar como una mecanismo para la reducción de las dimensiones de los datos.
- **CNN** (*Convolutional Neural Network*). Su origen se encuentra en el Neocognitron (Fukushima, K. & Miyake, S. (1982) [35] y su aplicación a detección de patrones Fukushima, K. (1988) [34]), que se basaba en hallazgos sobre el procesamiento de la información visual (Hubel, D. H. & Wiesel, T. N. (1962) [49]). Posteriormente se desarrolló un modo eficiente de aprendizaje por retropropagación del error (no con pesos asignados manualmente) (Le Cun, Y. et al. (1989) [61]). Se utiliza fundamentalmente en el procesamiento de imágenes.
- **RNN** (*Recurrent Neural Networks*). Este tipo de arquitectura es capaz de procesar datos secuenciales y se debe a Rumelhart, D. E. et al (1985) [100]. Su estructura particular dota a este tipo de red de memoria, ya que los *outputs* en pasos previos vuelven a entrar posteriormente como *inputs*. Esto permite su utilización en áreas que se habían resistido al empleo de redes neuronales tradicionales, como la del procesamiento del lenguaje natural (*NLP - Natural Language Processing*).
Entre las dificultades que se tienen que abordar se encuentra el problema que surge con la desaparición del gradiente o su crecimiento explosivo, que fue detectado por Bengio, Y. et al (1993) [13]. Como consecuencia de ello han surgido variaciones que permiten mantener un flujo de gradiente estable como la red recurrente LSTM (Hochreiter, S. & Schmidhuber, J. (1997) [47]).

Respecto del uso en Aprendizaje por Refuerzo, tenemos que las redes neuronales se han adoptado extensamente como estimadores funcionales (MLP) (ver por ejemplo: Tsitsiklis, J. (1996) [121]), y para la extracción de 'features' (CNN).

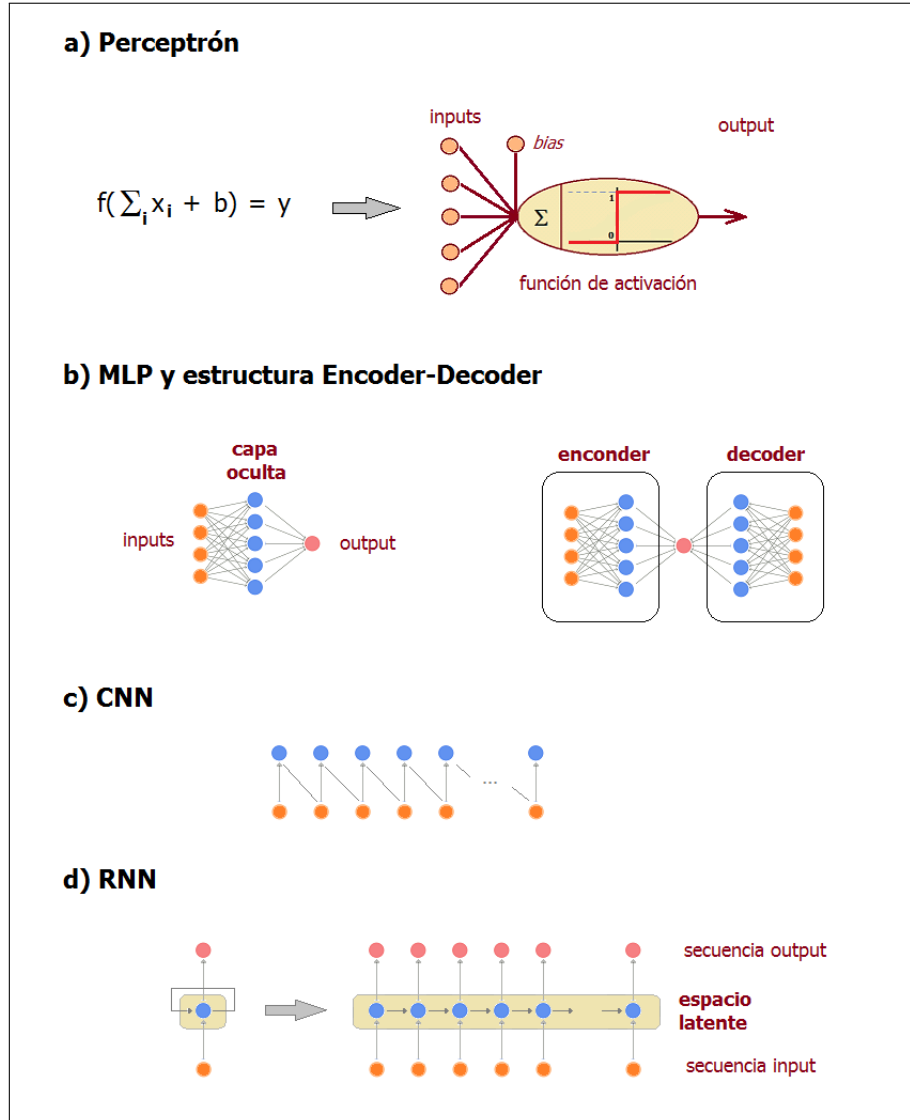


Figura 4.6: Algunas arquitecturas de Redes Neuronales (Esquemas adaptados de Principe, J. C. et al. (1999) [92], Goodfellow, I. et al. (2016) [38] y Brunton, S. L. & Kutz, J. N. (2022) [21]).

A la hora de abordar el estudio del empleo de redes neuronales, conviene clasificar los distintos métodos. Posteriormente distinguiremos entre:

- 1) **Value based methods.** Estimamos la función de valor mediante RN. Generalmente $Q(s,a)$.
- 2) **Policy based methods.** Consiste en parametrizar la función de decisión y en utilizar la red neuronal para estimar dicha función: $\mu(\cdot)$.
- 3) **Actor-Critic methdos.** Utilizan una mezcla de los métodos anteriores.

En los siguientes apartados procederemos a presentar algunos de los resultados más notables de aplicación de Redes Neuronales, comenzando primero con la construcción del tipo de función de pérdida que se emplea para el entrenamiento de las redes.

4.4. Utilización de Redes Neuronales en RL

Para construir la función de pérdida $\mathcal{L}(\theta)$ acudimos a la ecuación de Bellman. Para ello se toma en cuenta la diferencia entre el valor estimado del siguiente estado (s') - denominado *target* -, y el valor estimado para el estado actual, que sería el *valor observado*. La función de pérdida se construiría ponderando de alguna forma estos errores, mediante una función g :

$$\mathcal{L}(\theta) = g\left(\underbrace{r + \gamma \cdot \max_{a' \in \mathcal{A}} Q_{\theta}(s', a')}_{\text{target}} - \underbrace{Q_{\theta}(s, a)}_{\text{observado}} \right)$$

Como función g , podemos utilizar $g = \mathbb{E}[(\cdot)^2]$, lo que daría lugar al Error Cuadrático Medio (MSE), que es una de las medidas más usuales a utilizar.

4.5. Value based Methods.

Con los algoritmos que analizamos a continuación podemos aproximar, por medio del aprendizaje, una de las funciones de valor que hemos visto. A partir de las valoraciones que proporciona de los pares (s,a) - en el caso de la función Q -, es inmediato el obtener la estrategia óptima.

4.5.1. Deep Q-Network (DQN)

La introducción de Redes Neuronales profundas (Deep Neural Networks) permitió avances significativos en el entrenamiento por refuerzo. El algoritmo DQN es debido a Mnih et al. (2013) [77].

A partir de ahora notaremos $Q_{\theta}^{\pi}(s, a)$ a la función de valor estado-acción. Con ello queremos indicar que la red neuronal depende de un vector de parámetros θ .

En la medida en esta red neuronal va a servir para obtener las estimaciones, surge la pregunta acerca de cómo obtener los valores de los parámetros. La solución pasa por emplear un bufer ('*replay bufer*' o '*experience replay*' - notado: \mathcal{D}) que almacene las experiencias que va acumulando el agente. El bufer acumula las transiciones que se han ido generando a lo largo de sucesivos episodios o trayectorias $(\tau_1, \tau_2, \dots, \tau_h)$ (Ver figura 4.7).

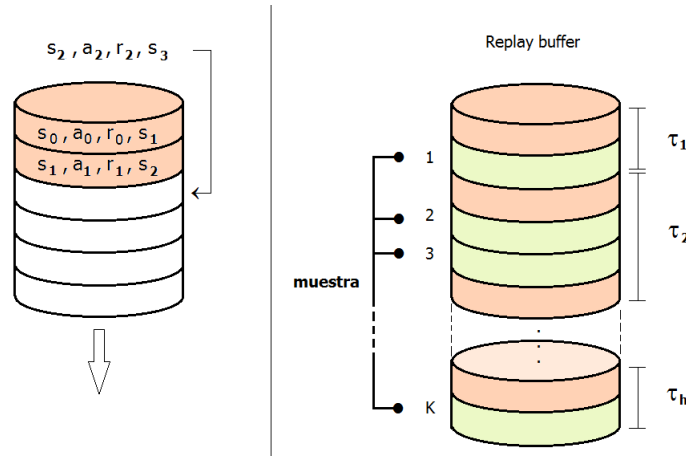


Figura 4.7: Funcionamiento del bufer, y extracción posterior de una muestra para entrenar la Red.

Como las experiencias dentro de un mismo episodio suelen estar altamente correladas, el procedimiento que se emplea es el de seleccionar una muestra aleatoria (*batch* o lote), que será el que se emplee para entrenar la red neuronal. El procedimiento es el habitual: emplear una función de pérdida $L(\theta)$, como puede ser el Error Cuadrático Medio (*Mean Square Error*). El gradiente de dicha función ($\nabla_{\theta} L(\theta)$) nos proporcionará la dirección en la que corregir los parámetros para acercar la salida al objetivo.

$$L(\theta) = MSE = \frac{1}{K} \sum_{i=1}^K \left[r_i + \gamma \cdot \max_{a' \in \mathcal{A}} Q_{\theta_{target}}^{\pi}(s'_i, a') - Q_{\theta}^{\pi}(s_i, a_i) \right]^2 \quad (4.1)$$

$$\theta = \theta - \alpha \cdot \nabla_{\theta} L(\theta) \quad (4.2)$$

Empleando el esquema que hemos venido empleando reiteradamente, los valores estado-acción quedarían, por tanto, actualizados según la siguiente expresión:

$$Q_{\theta}^{\pi}(s, a) \leftarrow Q_{\theta}^{\pi}(s, a) + \alpha \cdot \underbrace{\left[r + \gamma \cdot \max_{a' \in \mathcal{A}} Q_{\theta_{target}}^{\pi}(s', a') \right]}_{\text{target}} - \underbrace{Q_{\theta}^{\pi}(s, a)}_{\text{output}} \quad (4.3)$$

$TD(1) \text{ error}$

La secuencia de pasos completa quedaría recogida en el algoritmo 2 (incluir referencia).

Algoritmo 1 Deep Q-Network (DQN)

begin

 Inicializar la tasa de aprendizaje α

 Inicializar el factor de descuento γ

/*Llevamos a cabo una serie de episodios o trayectorias (M).

for *EPISODIO* = 1 **to** *M* **do**

/*Muestreamos K valores de bufer

 Recoger K experiencias $(s_i, a_i, r_i, s'_i, a'_i)$ generadas por la estrategia en vigor π

 for $i = 1$ **to** K **do**

 if (s'_i es un nodo terminal) **then**

 $y_i = r_i$

 else

 $y_i = r_i + \gamma \cdot \max_{a' \in \mathcal{A}} Q_{\theta_{target}}^{\pi}(s'_i, a'_i)$

/*Calcular el Error Cuadrático Medio (MSE)

$$L(\theta) = \frac{1}{K} \sum_{i=1}^K [y_i - Q_{\theta}^{\pi}(s_i, a_i)]^2$$

/*Actualizar los parámetros de la red

$$\theta = \theta - \alpha \cdot \nabla_{\theta} L(\theta)$$

4.5.2. Double Deep Q Network (DDQN)

Algunos estudios han demostrado una sistemática sobreestimación en el valor de los estados en el caso del proceso de aprendizaje con Q-Learning cuando es combinado con un aproximador funcional como son las Redes Neuronales (Thrun, S. & Schwartz, A. (1993) [119]). La razón reside en la utilización del operador 'max' a la hora de aproximar el valor del siguiente estado en una transición, ya que introduce un sesgo positivo. Además al utilizar los mismos valores tanto para seleccionar una acción como para evaluarla, hace que sea más probable el seleccionar valores sobreestimados, lo que conduce a estimaciones de valor de los estados demasiado optimistas.

Una posible solución consistiría en desacoplar la parte de selección de acciones de la de evaluación, introduciendo un modo alternativo de aproximar el valor esperado máximo (Hasselt, H. (2010) [41]). En este artículo, el autor aproxima el valor esperado máximo para cualquier conjunto de variables aleatorias, y posteriormente lo aplica al algoritmo Q-Learning introduciendo un doble estimador (ver también: Hasselt, H. (2011) [42] y Hasselt, H. et al. (2015) [122]).

Esta extensión del modelo DQN recibe el nombre de Double Deep Q-Network (DDQN). La actualización de los pesos obedece a la siguiente expresión (Ecuación 4.4):

$$Q_{\theta}^{\pi}(s, a) \leftarrow Q_{\theta}^{\pi}(s, a) + \alpha \cdot [r + \gamma \cdot \underbrace{\max_{a' \in \mathcal{A}} Q_{\theta'}^{\pi}(s', \arg \max_{a' \in \mathcal{A}} Q(s', a'))}_{\text{RN1}} - \underbrace{Q_{\theta}^{\pi}(s, a)}_{\text{RN1b}}] \quad (4.4)$$

↑
↑
↑

RN1
RN2
RN1b

Los diferentes términos que intervienen son:

$$Q_{\theta}^{\pi}(s, a) \leftarrow Q_{\theta}^{\pi}(s, a) + \alpha \cdot \underbrace{[r + \gamma \cdot \max_{a' \in \mathcal{A}} Q_{\theta'}^{\pi}(s', \arg \max_{a' \in \mathcal{A}} Q(s', a'))]}_{\text{target}} - \underbrace{Q_{\theta}^{\pi}(s, a)}_{\text{output}} \quad (4.5)$$

}
}

TD(1) error

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

4.5.3. Deep Q-Network with Prioritized Experience Replay (PER)

Parece razonable centrarse en el conjunto de transiciones para el cual la red genera un error en término de la Diferencia Temporal más grande.

$$\underbrace{r + \gamma \cdot \max_{a' \in \mathcal{A}} Q_{\theta'}^{\pi}(s', a')}_{\text{target}} - \underbrace{Q_{\theta}^{\pi}(s, a)}_{\text{output}} = \text{TD}(1) \text{ error}$$

El algoritmo que procedemos a analizar fue desarrollado por Schaul et al. (2016) [103], y se centra en la cuestión que acabamos de mencionar: utiliza un buffer con una prioridad asignada a cada transición en función del error.

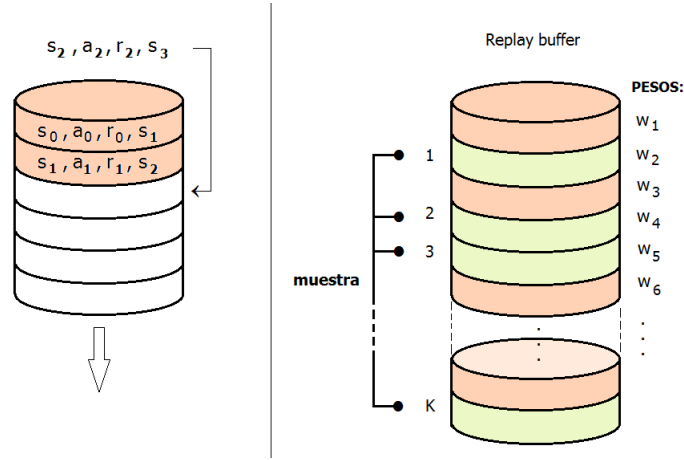


Figura 4.8: *Funcionamiento del bufer, y extracción de una muestra ponderada por el efecto que la observación tiene en el error.*

Existen dos métodos de priorización de las observaciones de la base de datos. Un primer paso consiste en asignar una valor proporcional al error, y otro emplea la ordenación previa en base al error y asigna posteriormente el rango que ocupa cada observación. Esto genera unos valores p_i

$$\begin{cases} \bullet \text{ Proporcional :} & p_i = |TD \text{ error}| + \epsilon \\ \bullet \text{ Con rango :} & p_i = \frac{1}{\text{rango}(i)} \end{cases}$$

con estos p_i procedemos a calcular $P(i)$, donde $\alpha \in [0, 1]$:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}$$

Y para evitar el sobreajuste ('*overfitting*'), se reduce la importancia de transiciones que están siendo muestreadas muchas veces. Esto genera los pesos finales con los que se establece el muestreo.

$$w_i = \left[\frac{1}{N} \cdot \frac{1}{P(i)} \right]^\beta$$

El valor de N representa el tamaño del buffer, $P(i)$ es la probabilidad de la transición a la hora de ser muestreada, y β es un valor que se ajusta mediante *annealing* (empezando por un valor alrededor de 0.4 y haciendolo tender a 1).

4.5.4. Dueling Deep Q-Network (Dueling DQN) (D2QN)

Si recordamos la definición de Ventaja ('*advantage*'), vemos que nos proporciona una medida de la bondad de una determinada acción a en un estado s en comparación con lo obtenido en promedio con el resto de acciones.

$$A(s, a) = Q(s, a) - V(s)$$

El algoritmo siguiente fue desarrollado por Wang et al. (2016) [126], y hace uso de este valor para calcular la función $Q(s, a)$. Para ello calcula $Q(s, a)$ mediante $V(s) + A(s, a)$.

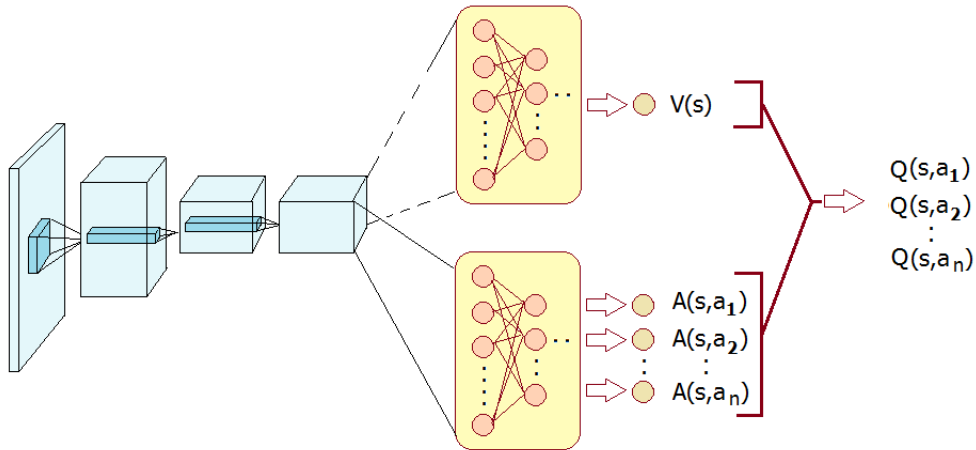


Figura 4.9: Estructura de la Red Neuronal empleada en Dueling DQN.

Se diseña una arquitectura neuronal (que podemos ver en la figura 7) cuya salida se bifurca para obtener la función de valor ($V(s)$) por un lado, y por otro lado la ventaja de todas las posibles acciones en ese estado ($A(s, a_i)$). El valor final de la función $Q(s, a)$ se obtiene agregando estos valores generados por la red.

4.5.5. Deep Recurrent QN (DRQN)

El algoritmo fue desarrollado por Hauskchnet et. al (2017) [43].

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Esto que aparece aquí es una pequeña muestra de texto para ir haciéndose una idea de la apariencia de la Tesis.

4.6. Policy Methods.

A continuación abordamos el estudio de aquellos algoritmos que permiten el aprendizaje directo de la función de estrategia π . Entre las ventajas se encuentra la convergencia local asegurada a un óptimo local (tal y como demostraron Sutton et al. (2000)) [116].

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Esto que aparece aquí es una pequeña muestra de texto para ir haciéndose una idea de la apariencia de la Tesis.

4.6.1. Proximal Policy Optimization (PPO)

El algoritmo fue desarrollado por Schulman et al. (2017) [107].

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

4.6.2. Trust Region Policy Optimization (TRPO)

El algoritmo fue desarrollado por Schulman et al. (2015) [106].

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

4.6.3. Actor-Critic using Kronecker-factored Trust Region (ACKTR)

El algoritmo fue desarrollado por Wu et al. (2017) [129].

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Esto que aparece aquí es una pequeña muestra de texto para ir haciéndose una idea de la apariencia de la Tesis.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

4.7. Actor-critic Methods

Se han detectado problemas análogos de sobreestimación es este tipo de métodos a los ya mencionados que se producen en Deep Q-Learning, lo que puede conducir al aprendizaje de estrategias subóptimas (Fujimoto, S. et al. (2018) [33]).

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

4.7.1. Advantage Actor-Critic (A2C)

Sobre este tipo de métodos destaca el artículo de Konda y John Tsitsiklis (2000) [56]

En lo que sigue se apuntan algunas de las líneas de investigación que han surgido recientemente y que abordan diferentes cuestiones que permiten mejorar los resultados obtenidos con los métodos tradicionales.

4.7.2. Asynchronous Advantage Actor-Critic (A3C)

El algoritmo fue desarrollado por Mnih et al. (2016) [76].

En lo que sigue se apuntan algunas de las líneas de investigación que han surgido recientemente y que abordan diferentes cuestiones que permiten mejorar los resultados obtenidos con los métodos tradicionales.

En lo que sigue se apuntan algunas de las líneas de investigación que han surgido recientemente y que abordan diferentes cuestiones que permiten mejorar los resultados obtenidos con los métodos tradicionales.

En lo que sigue se apuntan algunas de las líneas de investigación que han surgido recientemente y que abordan diferentes cuestiones que permiten mejorar los resultados obtenidos con los métodos tradicionales.

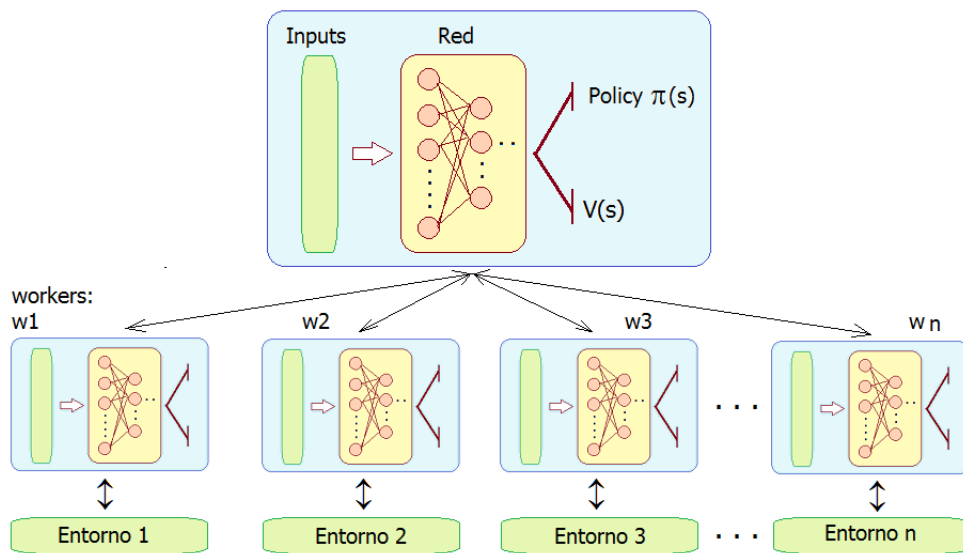


Figura 4.10: Esquema de la implementación del modelo A3C.

En lo que sigue se apuntan algunas de las líneas de investigación que han surgido recientemente y que abordan diferentes cuestiones que permiten mejorar los resultados obtenidos con los métodos tradicionales.

En lo que sigue se apuntan algunas de las líneas de investigación que han surgido recientemente y que abordan diferentes cuestiones que permiten mejorar los resultados obtenidos con los métodos tradicionales.

4.7.3. Deep Deterministic Policy Gradient (DDPG)

El algoritmo fue desarrollado por Lillicrap et al. (2016) [65].

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal,

como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

4.7.4. Twin Delayed Deep Deterministic Policy Gradient (TD3)

El algoritmo fue desarrollado por Fujimoto et al. (2018) [33].

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

4.7.5. Soft Actor-Critic (SAC)

El algoritmo fue desarrollado por Haarnoja et al. (2018) [39]. Y más en profundidades sobre este tipo de algoritmos ver también Haarnoja (2018) [40].

En lo que sigue se apuntan algunas de las líneas de investigación que han surgido recientemente y que abordan diferentes cuestiones que permiten mejorar los resultados obtenidos con los métodos tradicionales.

Artículo de McCulloch & Pitts (1943) [74].

Artículo de Rosenblatt, F. (1958) [99].

BENCHMARKS para Multi-agent Deep Reinforcement Learning:
Tenemos algunos entornos que nos permiten testar los algoritmos (Hernández-Leal, P. et al (2019) [45]).

También algunos autores han tratado de llevar a cabo una comparativa en el caso del Deep Reinforcement Learning Multiagente (Papoudakis, G. et al. (2020) [86]).

Se repasan los orígenes de las redes neuronales en las estructuras planteadas en los siguientes artículos: Artículo de McCulloch & Pitts (1943) [74] y Artículo de .

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Al ir pulsando ENTER va apareciendo un numerado en los párrafos, por lo que parece que no se necesitan comandos y que tomará de aquí directamente el texto. Posteriormente voy a ir introduciendo órdenes para las diferentes secciones para comprobar si funciona de manera normal, como cuando se incluye todo el texto en el archivo principal.

Capítulo 5

Estudio de los agentes influyentes en una Red Social II

5.0.1. Estructura de Red Neuronal 3: CNN - 3D

La arquitectura CNN (*Convolutional Neural Network*) que hemos analizado previamente, ha sido extendida por otros autores para poder manejar 'profundidad'. Es decir, secuencias de imágenes (Ji, S. et al. (2012)) [53]. Entre las aplicaciones podemos encontrar la detección de objetos (Maturana, D. & Scherer, S. (2015)) [71]. Nosotros la emplearemos para segmentar las secuencias de jugadas generadas por simulación.

Podemos observar la estructura de este tipo de arquitectura neuronal empleada en la figura 5.1.

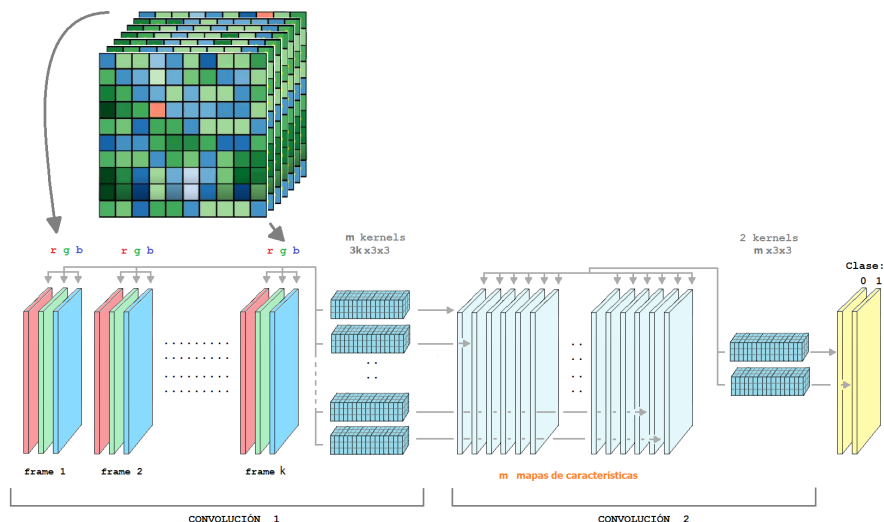


Figura 5.1: *Arquitectura de Red Neuronal 'CNN 3D'.*

Su estructura está compuesta por 2 partes:

Convolución 1.

Esta primera sección se encarga de extraer de la secuencia de jugadas, un conjunto de mapas de características a partir de la aplicación de kernels con una profundidad acorde con el número de imágenes que se le pasan.

Convolución 2.

Esta sección se encarga de combinar los mapas de características extraídas, para generar varias capas finales. Éstas contienen las probabilidades de que cada agente (uno por pixel) pertenezca a cada clase o tipo de agente que se desea detectar (y con los que ha sido entrenada la red).

Capítulo 6

Resultados

En las pruebas de simulación que hemos llevado a cabo, podemos comprobar que el agente que toma decisiones en base a su estado emocional, y a través de un proceso de aprendizaje, se desenvuelve bien en el entorno.

En las pruebas, hemos llevado a cabo análisis con todos los jugadores tomando decisiones a través del aprendizaje, aunque también se pueden introducir distintos tipos de jugadores.

En concreto hemos manejado diversos tipos de **autómatas celulares**. Tenemos casos sencillos como el agente traidor (que juega sistemáticamente a aprovecharse de sus vecinos) o el agente ingenuo (que se empeña en cooperar reiteradamente con independencia de las estrategias que los demás empleen con él).

También hemos introducido la estrategia clásica Tif-for-Tat (toma y daca), en la que el jugador comienza cooperando, pero a partir de ser traicionado responderá con la estrategia que el contrincante haya tenido con él en la ronda anterior.

El caso que sobresale especialmente es el del agente ingenuo, ya que obtiene muy malos resultados. Tenemos que la estrategia cooperativa puede ser una muy mala elección si los demás no responden de la misma manera.

En la figura 6.1 aparece el efecto de introducir una población de mutantes que sistemáticamente no cooperan con las celdas vecinas.

De entre los tipos de agentes que se han probado, el agente no cooperador es el que ejerce unos efectos más claros. Por ello comenzaremos el análisis de la detección de agentes mutantes centrándonos en el caso de una mezcla poblacional con los agentes emocionales.

Las simulaciones en este segundo caso se han llevado hasta más de 4000 ciclos, para hacer más claros los resultados.

Si observamos comportamientos distintivos entre los jugadores tendría especial interés clasificar y localizar la posición en la red de aquellos jugadores que mayor perturbación provocan.

En la parte que hemos realizado de Simulación, hemos observado el proceso de aprendizaje de los agentes que toman decisiones en base a su estado emocional. Hemos incluido alguna de las estrategias clásicas, lo que permite observar su adecuación (ver fig. 6.1) y compararlas entre sí.

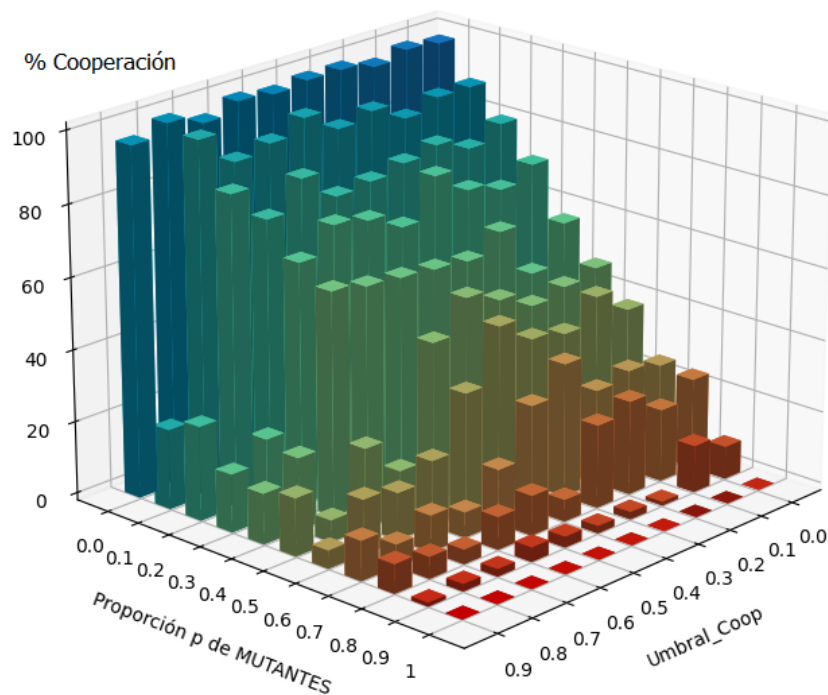


Figura 6.1: Comparación de los resultados obtenidos por distintos agentes. El mecanismo de interacción empleado en las simulaciones es el Dilema del Prisionero.

Estrategias clásicas como Tit-for-Tat obtienen buenos resultados.

Las simulaciones también permiten analizar las interacciones que se dan entre ciertos tipos de agentes. Un efecto que podemos ver repetido en algunas de las simulaciones es la propagación por todo el tablero de los efectos de las estrategias de algunos autómatas. Esto es especialmente cierto cuando los agentes se encuentran concentrados en una región del grid.

Este efecto nos presenta la oportunidad de analizar si es posible entrenar a una red neuronal con este tipo de patrones visuales que aparecen en las simulaciones, ligándolos a la presencia en la red de cierto tipo de agentes (y a ser posible encontrar su localización). Al ser generados por simulación los datos, podemos tener identificados todos los elementos que intervienen y aplicar técnicas de Aprendizaje Supervisado, como ya se comentó previamente.

El procedimiento nos permitiría encontrar los grupos de individuos influyentes de la red mediante técnicas de Aprendizaje Automático, empleando arquitecturas de redes neuronales empleadas en visión por computadora.

En las figuras 6.2 y 6.3 podemos observar los resultados de entrenamiento. Se han elaborado simulaciones con dos poblaciones una población de agentes emocionales y otra de agentes mutantes (siempre traicionan) para probar el pipeline completo de procesamiento de los datos. En la figura 6.4 aparece reflejada una salida tras el proceso de clasificación. Se encuentran resaltados en color rojo los fallos de clasificación.

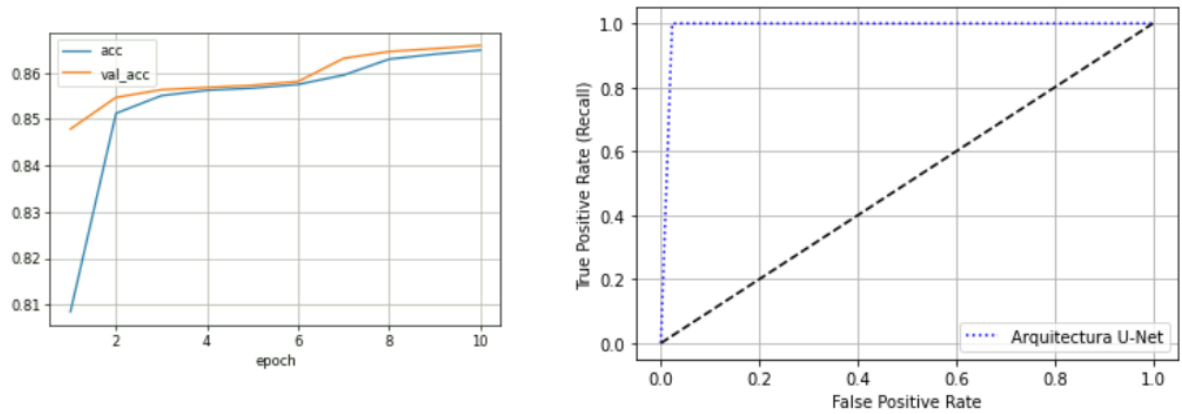


Figura 6.2: Evolución de la precisión durante el entrenamiento de la red U-Net. Prueba con 2 poblaciones. Curva ROC con los resultados relativos al problema de clasificación de dos poblaciones.

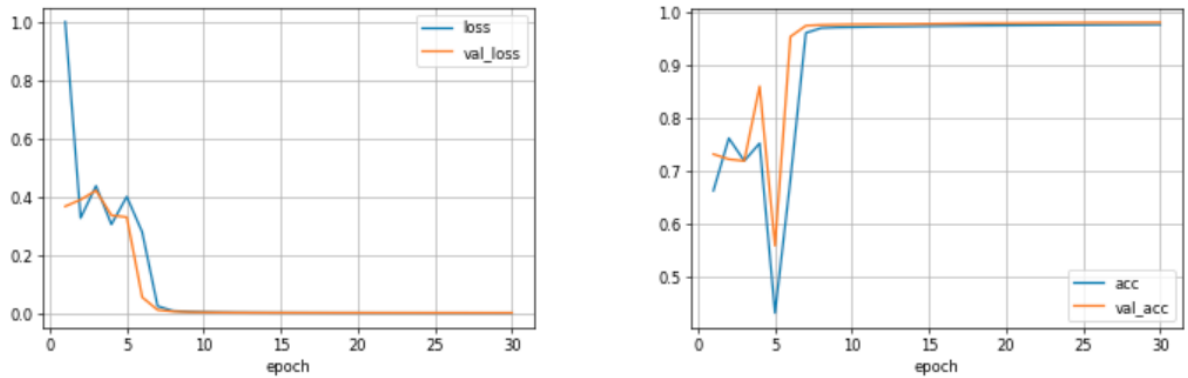


Figura 6.3: Curva de la función de pérdida (entrenamiento y validación) y evolución de la precisión en el entrenamiento.



Figura 6.4: Tratamiento del problema de localización de los nodos influyentes como un problema de segmentación de imágenes. Prueba con una pequeña red de 20x20. Los individuos mutantes aparecen que han sido correctamente identificados aparecen en color verde (1), en rojo los fallos de clasificación.

Capítulo 7

Conclusiones

Conclusiones

$$P(u_1) = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$
$$P(u_2) = \begin{bmatrix} 1 & 0 \\ \theta & (1 - \theta) \end{bmatrix}$$

Ahora las matrices. Para extraer B_{iy} con $y = 1, 2$

$$\mathbf{B} = \begin{bmatrix} p & (1 - p) \\ (1 - q) & q \end{bmatrix}$$

\Rightarrow

$$B_1 = \begin{bmatrix} p & 0 \\ 0 & (1 - q) \end{bmatrix}$$
$$B_2 = \begin{bmatrix} (1 - p) & 0 \\ 0 & q \end{bmatrix}$$

El siguiente ejemplo lo modelizaremos como un problema POMDP, resoluble recursivamente mediante Programación Dinámica.

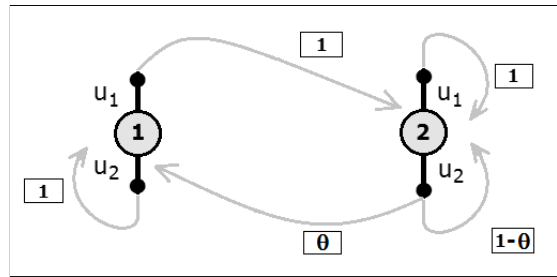
Problema de sustitución de maquinaria. Supongamos que un determinado proceso de producción puede pasar por dos estados $\mathcal{S} = \{s_1, s_2\}$ correspondiéndose con:

$$\begin{cases} s_1 : \text{ " Funcionamiento defectuoso " } \\ s_2 : \text{ " Sustitución de maquinaria " } \end{cases}$$

El estado del sistema evoluciona secuencialmente por etapas $(x_k, k = 0, 1, \dots)$, y el decisor puede actuar llevando a cabo una de las dos acciones siguientes en cada paso $\mathcal{U} = \{u_1, u_2\}$, donde:

$$\begin{cases} u_1 : \text{ " Reemplazar la máquina " } \\ u_2 : \text{ " Seguir usando la máquina " } \end{cases}$$

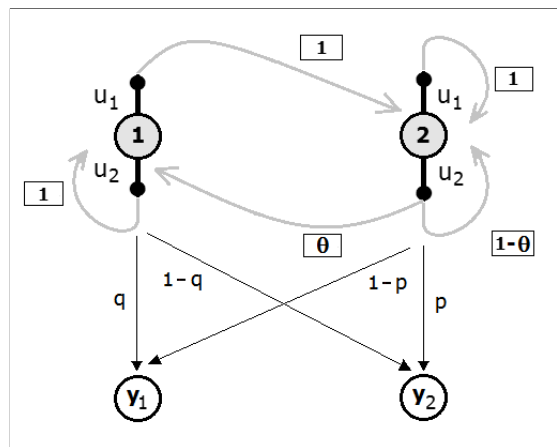
Si notamos s_i como i para simplificar notación, el proceso de funcionamiento del sistema puede representarse como refleja el siguiente gráfico.



La dificultad que surge reside en que el estado x_k no es directamente observable. Únicamente podemos juzgarlo indirectamente en base a la calidad del producto que fabrica: $\mathcal{Y} = \{y_1, y_2\}$.

$$\begin{cases} y_1 : \text{ " Producto de mala calidad " } \\ y_2 : \text{ " Producto de buena calidad " } \end{cases}$$

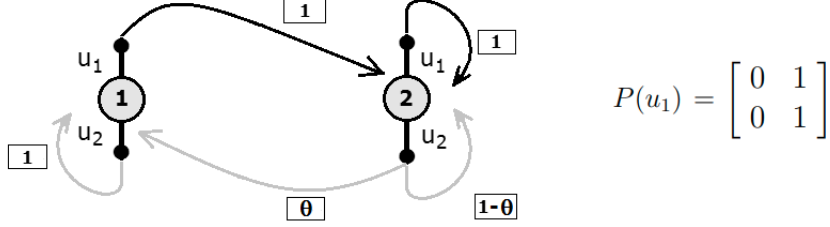
Si la probabilidad de que una máquina defectuosa genere un producto de mala calidad es p ($p(y_1|x = 1) = p$) y la probabilidad de que la calidad sea buena $1 - p$ ($p(y_2|x = 1) = 1 - p$); y la probabilidad de que una máquina en estado óptimo genere un producto de mala calidad es $1 - q$ ($p(y_1|x = 1) = 1 - q$) y la probabilidad de que la calidad sea buena q ($p(y_2|x = 1) = q$), entonces el sistema quedaría representado como:



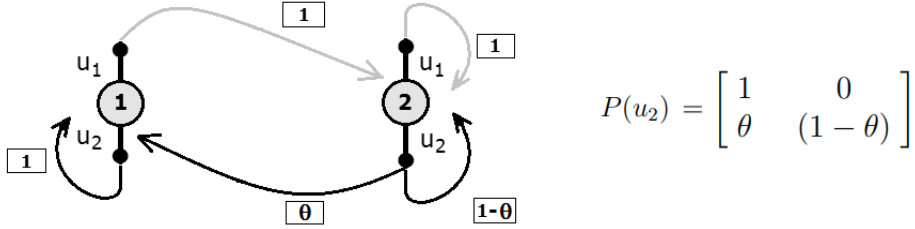
Y ahora tenemos en cuenta las transiciones relacionadas con cada acción que puede tomar el agente.

Para cada una de las decisiones ($u \in \mathcal{U}$), la matriz $P(u)$ recogerá las probabilidades: $P_{ij}(u) = p(x_{k+1} = j \mid x_k = i, u_k = u) \quad \forall i, j \in \mathcal{X}$.

En concreto, la matriz de probabilidades de transición asociadas a la decisión u_1 :



Y si la decisión tomada es u_2 tenemos que la matriz de probabilidades asociada es la siguiente:



La dificultad que surge reside en que el estado x_k no es directamente observable. Únicamente podemos juzgarlo indirectamente en base a la calidad del producto que fabrica.

Ahora para cada acción $u \in \mathcal{U}$, la matriz $B(u)$ recoge la distribución de probabilidad de las observaciones, asociada a la decisión u . En concreto tenemos que, los elementos que la constituyen son de la forma:

$$B_{iy}(u) = p(y_{k+1} = y \mid x_{k+1} = i, u_k = u) \quad \forall i \in \mathcal{X}, y \in \mathcal{Y}$$

de esta matriz extraemos las matrices $B_y(u)$ $y \in \mathcal{Y}$, colocando en la diagonal de una matriz $(0_{X \times X})$ la probabilidad de que cada uno de los estados del sistema genere la observación en cuestión 'y':

$$B_y(u) = \begin{pmatrix} B_{1y}(u) & 0 & \cdots & 0 \\ 0 & B_{2y}(u) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{Xy}(u) \end{pmatrix} \quad (7.1)$$

Lo que en nuestro caso se traduce en (*creo que hay error en el libro, lo corrijo*):

$$\mathbf{B} = \begin{bmatrix} q & (1-q) \\ (1-p) & p \end{bmatrix} \implies B_1 = \begin{bmatrix} q & 0 \\ 0 & (1-p) \end{bmatrix} \quad B_2 = \begin{bmatrix} (1-q) & 0 \\ 0 & p \end{bmatrix}$$

Para obtener la acción óptima necesitamos la **distribución a posteriori**. Llamémosla:

$$\pi_k = [\pi_k(1), \dots, \pi_k(X)]'$$

y que recoge la estimación del agente de la probabilidad de cada estado oculto, dada la información disponible hasta el periodo k ($\mathcal{I}_k = \{\pi_0, u_0, y_1, \dots, u_{k-1}, y_k\}$):

$$\pi_k(i) = p(x_k = i | \mathcal{I}_k) \quad i \in X$$

Tenemos que π_k es un estadístico suficiente, que no crece con la dimensión k . Para su cálculo nos serviremos de T , un filtro HMM (*Hidden Markov Model filter* - filtro de una modelo oculto de Markov). Es decir, el estado de creencia (*belief state o information state*) π_k se obtiene como:

$$\pi_k = T(\pi_{k-1}, y_k, u_{k-1})$$

donde:

$$T(\pi, y, u) = \frac{B_y \cdot P^t(u) \cdot \pi}{\sigma(\pi, y, u)} \quad \text{siendo:} \quad \sigma(\pi, y, u) = \mathbf{1}_X^t \cdot B_y(u) \cdot P^t(u) \cdot \pi$$

Finalmente para obtener las acciones óptimas nos basamos en el estado de creencia. Si estamos en el periodo k , tendremos que $u_k = \mu_k^*(\pi_k)$.

Para la obtención de la estrategia óptima, vendrá en un problema finito de la aplicación recursiva hacia atrás de las siguiente expresiones.

$$J_k(\pi) = \min_{u \in \mathcal{U}} \{c'_u \pi + \sum_{y \in \mathcal{Y}} J_{k+1}(T(\pi, y, u)) \cdot \sigma(\pi, y, u)\}$$

$$\mu_k^*(\pi) = \arg \min_{u \in \mathcal{U}} \{c'_u \pi + \sum_{y \in \mathcal{Y}} J_{k+1}(T(\pi, y, u)) \cdot \sigma(\pi, y, u)\}$$

lo que se traduce en:

$$J_k(\pi) = \min_{u \in \{u_1, u_2\}} \{c'_u \pi + J_{k+1}(T(\pi, y_1, u)) \cdot \sigma(\pi, y_1, u) + J_{k+1}(T(\pi, y_2, u)) \cdot \sigma(\pi, y_2, u)\}$$

realizando los cálculos recursivamente hacia atrás tendremos:

$$\begin{aligned} J_N(\pi) &= 0 \\ J_{N-1}(\pi) &= \min_{u \in \{u_1, u_2\}} \{c'_u \pi + J_N(T(\pi, y_1, u)) \cdot \sigma(\pi, y_1, u) + J_N(T(\pi, y_2, u)) \cdot \sigma(\pi, y_2, u)\} \\ J_{N-2}(\pi) &= \min_{u \in \{u_1, u_2\}} \{c'_u \pi + J_{N-1}(T(\pi, y_1, u)) \cdot \sigma(\pi, y_1, u) + J_{N-1}(T(\pi, y_2, u)) \cdot \sigma(\pi, y_2, u)\} \\ &\dots \dots \dots \end{aligned}$$

En resumen, encadenando valores recursivamente hacia atrás tenemos:

$$\begin{aligned} J_N(\pi) &= 0 \\ J_{N-1}(\pi) &= \min \{c'_{u_1} \pi + \overbrace{J_N(T(\pi, y_1, u_1))} \cdot \sigma(\pi, y_1, u_1) + \overbrace{J_N(T(\pi, y_2, u_1))} \cdot \sigma(\pi, y_2, u_1), \\ &\quad c'_{u_2} \pi + \overbrace{J_N(T(\pi, y_1, u_2))} \cdot \sigma(\pi, y_1, u_2) + \overbrace{J_N(T(\pi, y_2, u_2))} \cdot \sigma(\pi, y_2, u_2)\} \\ &= \min \{c_{u_1}^t \cdot \pi, c_{u_2}^t \cdot \pi\} \\ J_{N-2}(\pi) &= \min \{c'_{u_1} \pi + J_{N-1}(T(\pi, y_1, u_1)) \cdot \sigma(\pi, y_1, u_1) + J_{N-1}(T(\pi, y_2, u_1)) \cdot \sigma(\pi, y_2, u_1), \\ &\quad c'_{u_2} \pi + J_{N-1}(T(\pi, y_1, u_2)) \cdot \sigma(\pi, y_1, u_2) + J_{N-1}(T(\pi, y_2, u_2)) \cdot \sigma(\pi, y_2, u_2)\} \\ &\dots \dots \dots \end{aligned}$$

En el libro POMDP aparece erroneamente e_1 siendo e_2 (corregido en pdf del autor con erratas del libro), teniendo esto en cuenta el desarrollo que plantea el autor consiste en:

$$J_{N-2}(\pi) = \min \{ c'_{u_1} \pi + J_{N-1}(e_2), \\ c'_{u_2} \pi + J_{N-1}(T(\pi, y_1, u_2)) \cdot \sigma(\pi, y_1, u_2) + J_{N-1}(T(\pi, y_2, u_2)) \cdot \sigma(\pi, y_2, u_2) \} \\ \dots\dots\dots$$

Lo siguiente.

Es desarrollar las fórmulas

En resumen, encadenando valores recursivamente hacia atrás tenemos:

7.1. Procesos de Decisión de Markov

En el presente apartado ensamblamos todas las piezas que hemos manejado previamente para así, fijar las condiciones que nos permiten resolver el problema de toma de decisiones óptimas secuenciales, cuando el entorno está sujeto a un comportamiento aleatorio. Para ello damos un conjunto de definiciones introductorias que nos permiten formalizar el tipo de proceso que buscamos.

Proceso estocástico. Si observamos un sistema que evoluciona aleatoriamente, siguiendo una determinada ley de probabilidad, y dicho sistema es observado en tiempos discretos: $n = 0, 1, 2, \dots$, podemos definir una variable aleatoria X_n asociada a cada instante temporal de observación. Esta v.a. vendría a representar el estado del sistema en el instante de observación 'n'. En términos formales se ha definido un **proceso estocástico en tiempo discreto** ($\{X_n, n \geq 0\}$).

El proceso estocástico serían este conjunto de variables aleatorias indexadas, que toman valor en un determinado conjunto \mathcal{S} , llamado el espacio de estados del proceso estocástico.

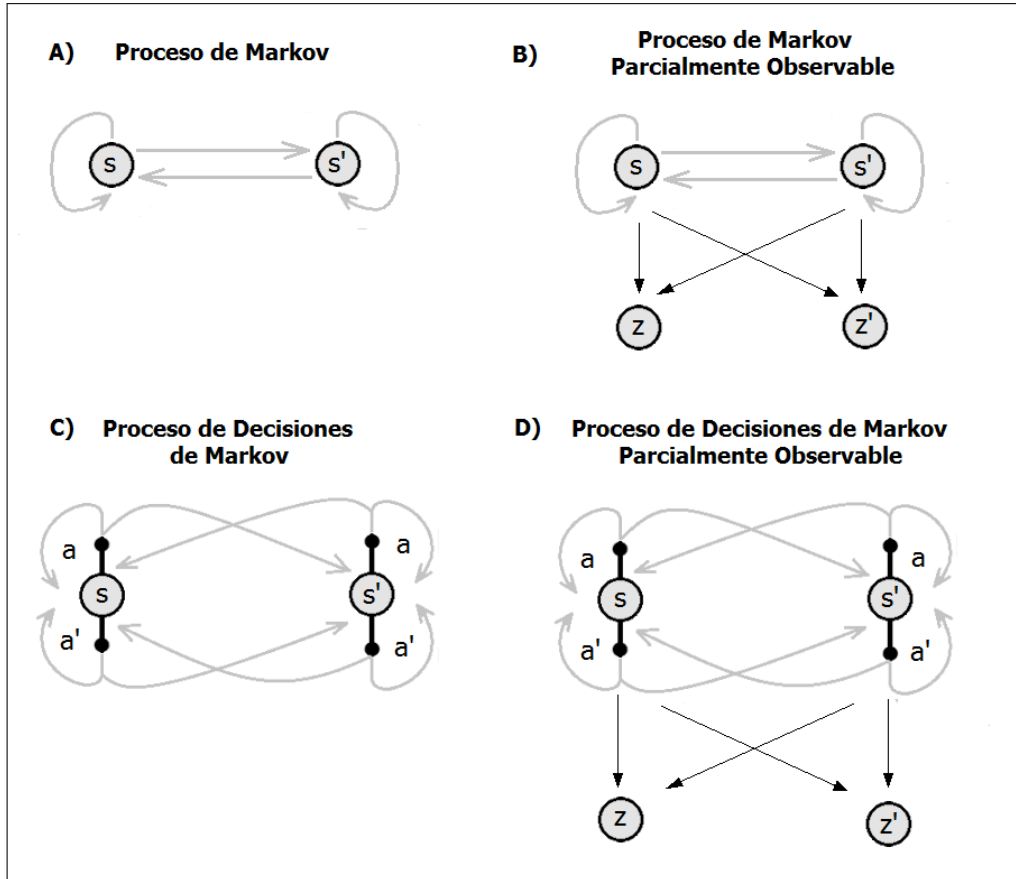


Figura 7.1: Representación de diversos tipos de procesos.

Un tipo especial de proceso es aquel que cumple la propiedad de Markov, que consiste básicamente en que, el pasado es irrelevante a la hora de predecir el futuro y únicamente aporta información el estado actual del sistema. En términos probabilísticos consistiría en que:

$p(X_{n+1} = j \mid X_n = i, X_{n-1}, \dots, X_1, X_0) = p(X_{n+1} = j \mid X_n = i)$. En este caso el proceso $\{X_n, n \geq 0\}$ recibe el nombre de *Cadena de Markov* (Kulkarni, V.G. (2016) [60]).

Proceso de Markov oculto (parcialmente observable) (HMM - *Hidden Markov Model*). Este tipo de modelo es apropiado cuando los estados del sistema bajo estudio no son directamente observables ($\{X_n, n \geq 0\}$) por lo que no podemos determinar con precisión el estado del sistema; pero, podemos observar ($\{Y_n, n \geq 0\}$) que depende del proceso de Markov X_n .

En la figura 7.1 podemos observar la representación de varios tipos de procesos, entre ellos un proceso HMM en B).

Proceso de decisiones de Markov. La situación que nos interesa analizar es la de un sistema probabilístico que evoluciona a lo largo del tiempo, y sobre el que podemos influir con la toma de decisiones, para conseguir que el sistema logre un desempeño óptimo (Puterman, M. L. (2014) [93]). La definición formal sería la que aparece a continuación.

ESPACIO

PARA

COLOCAR

EL

CUADRO

DE

COLOR

CON

LA

DEFINICIÓN

FORMAL

Si no se conocen las probabilidades de transición podemos acudir a métodos experimentales. Una posibilidad pasaría por obtener estimaciones explícitas de los parámetros del MDP (Moore, A. & Atkeson, C. (1993) [78]); es decir, de las probabilidades de transición y de las recompensas, y entonces aplicar los métodos de Programación Matemática.

Sin embargo, la opción más habitual consiste en el empleo de métodos de aprendizaje que no necesitan manejar expresamente las probabilidades. Con estos métodos podemos obtener la estimación del valor de cada estado, y determinar las acciones óptimas que conducen a ellos.

Entre los algoritmos de aprendizaje que se utilizan, podemos distinguir algoritmos *off-line*, que llevan a cabo la computación antes del proceso de control/decisión (Un ejemplo lo tenemos en el uso de Redes Neuronales, que necesitan un procesamiento por lotes y por tanto la información ha de ser acumulada y almacenada previamente). Por el contrario los algoritmos *on-line* llevan a cabo la computación cuando el proceso comienza, y los datos van estando disponibles.

Problema de sustitución de maquinaria. Supongamos que una determinada maquinaria puede pasar por tres estados $\mathcal{S} = \{s_1, s_2, s_3\}$ correspondiéndose con la calidad de la máquina en orden creciente.

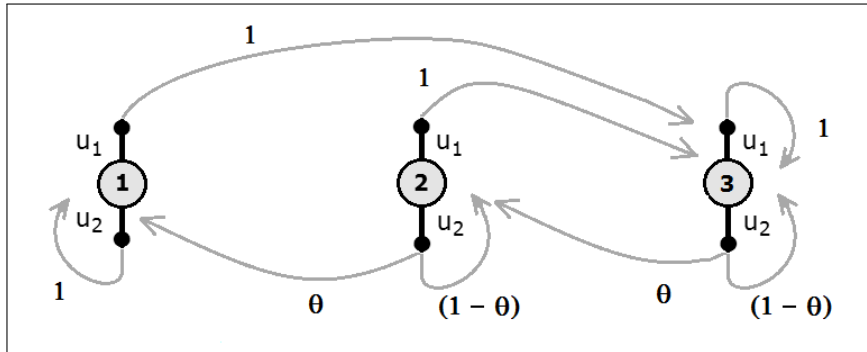
El estado del sistema evoluciona secuencialmente por etapas $(x_k, k = 0, 1, \dots)$, y el decisor puede actuar llevando a cabo una de las dos acciones siguientes en cada paso $\mathcal{U} = \{u_1, u_2\}$, donde:

$$\begin{cases} u_1 : \text{ " Reemplazar la máquina " } \\ u_2 : \text{ " Seguir usando la máquina " } \end{cases}$$

Supongamos que θ representa la probabilidad de que la maquinaria se deteriore hasta el nivel inferior de calidad. Entonces las probabilidades del proceso de decisiones de Markov son las que aparecen a continuación.

$$P(u_1) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad P(u_2) = \begin{bmatrix} 1 & 0 & 0 \\ \theta & (1 - \theta) & 0 \\ 0 & \theta & (1 - \theta) \end{bmatrix}$$

Si notamos s_i como i para simplificar notación, el proceso de funcionamiento del sistema puede representarse como refleja el siguiente gráfico, donde las probabilidades asociadas a cada transición aparecen sobre cada arco.



Cada decisión tiene un coste asociado. Si llevamos a cabo la acción u_1 (*reemplazar la máquina*) desde cualquier estado tenemos un coste de $c(x, u_1) = R$. Si por el contrario optamos por la acción u_2 y se deja funcionar sin recambiar, incurrimos en el mismo coste sea cual sea el estado: $c(x, u_2) = c, x = 1, 2, 3$.

El objetivo consiste en encontrar la secuencia de decisiones que permitan minimizar el coste asociado a la operación de la máquina (revisar la notación).

$$\mu^* = \min_{\mu} \mathbb{E}_{\mu} \left[\sum_{k=0}^{N-1} c(x_k, u_k) \mid x_0 \right]$$

Empleando la ecuación de Bellman, utilizamos programación dinámica, sustituyendo recursivamente hacia atrás, empezando por $J_N(i) = 0, i \in X$, y prosiguiendo con $k = N - 1, \dots, 0$:

$$J_k(i) = \min \{ R + J_{k+1}(1), c(i, 2) + \sum_{j=1}^3 P_{ij}(2) \cdot J_{k+1}(j) \} \quad i \in \{1, 2, 3\}$$

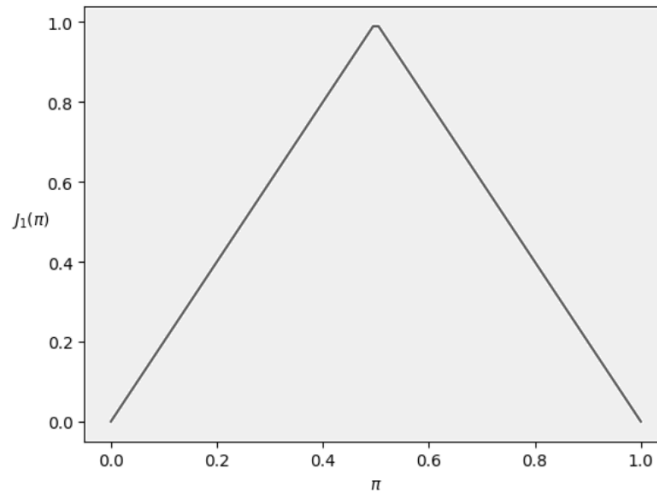
Resolver el POMDP (*con horizonte finito*) consistirá en encontrar la secuencia de decisiones óptima, es decir la policy óptima $\mu^* = (\mu_0, \mu_1, \dots, \mu_{N-1})$. Esto se traduce en encontrar la solución mediante Programación Dinámica de la ecuación de Bellman recursiva hacia atrás.

$$J_k(\pi) = \min_{u \in \mathcal{U}} \{ c_u \cdot \mu + \sum_{y \in \mathcal{Y}} J_{k+1}(T(\pi, y, u)) \cdot \sigma(\pi, y, \mu) \}$$

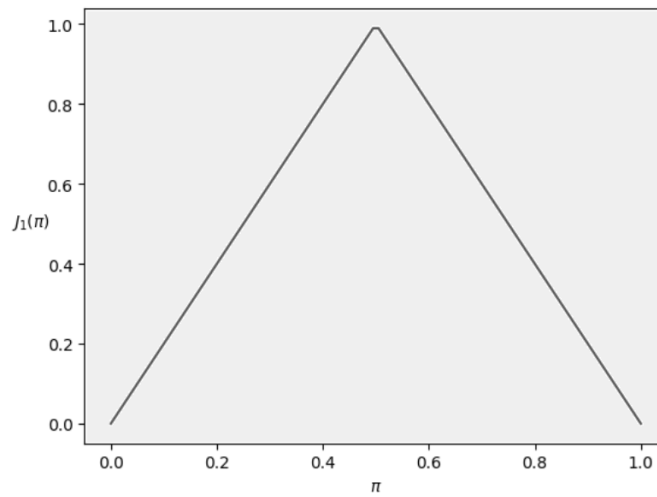
y a partir de la obtención de estos valores:

$$\mu_k^*(\pi) = \arg \min_{u \in \mathcal{U}} J_\mu(\pi_0)$$

la gráfica del POMDP quedaría para el ejemplo en cuestión:



y la siguiente gráfica quedaría:



Referencias

- [1] Acemoglu, D. y Ozdaglar, A. “Opinion dynamics and learning in social networks”. En: *Dynamic Games and Applications* 1 (2011), págs. 3-49. URL: <https://dspace.mit.edu/bitstream/handle/1721.1/71547/Acemoglu10-15.pdf?sequence=1> (vid. pág. 20).
- [2] Acemoğlu, D., Como, G., Fagnani, F. y Ozdaglar, A. “Opinion fluctuations and disagreement in social networks”. En: *Mathematics of Operations Research* 38.1 (2013), págs. 1-27. URL: <https://arxiv.org/pdf/1009.2653.pdf> (vid. pág. 20).
- [3] Ana, L., Bordini, R. H. y Campbell, J. A. “Agents with moral sentiments in an iterated prisoner’s dilemma exercise”. En: (1997). URL: <https://www.aaai.org/Papers/Symposia/Fall/1997/FS-97-02/FS97-02-002.pdf> (vid. pág. 24).
- [4] Anagnostopoulos, A., Kumar, R. y Mahdian, M. “Influence and correlation in social networks”. En: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008, págs. 7-15. URL: https://www.ccs.neu.edu/home/yzsun/classes/2014Spring_CS7280/Papers/Diffusion/influence.pdf (vid. pág. 5).
- [5] Axelrod, R. “The dissemination of culture: A model with local convergence and global polarization”. En: *Journal of conflict resolution* 41.2 (1997), págs. 203-226. URL: https://deepblue.lib.umich.edu/bitstream/handle/2027.42/67489/10.1177_0022002797041002001.pdf?sequence=2 (vid. pág. 8).
- [6] Axelrod, R. y Hamilton, W. D. “The evolution of cooperation”. En: *science* 211.4489 (1981), págs. 1390-1396. URL: <http://www-personal.umich.edu/~axe/research/Axelrod%20and%20Hamilton%20EC%201981.pdf> (vid. págs. 7, 23).
- [7] Baker, B. “Emergent reciprocity and team formation from randomized uncertain social preferences”. En: *Advances in Neural Information Processing Systems* 33 (2020), págs. 15786-15799. URL: <https://arxiv.org/pdf/2011.05373.pdf> (vid. pág. 8).
- [8] Barabási, A.-L. y col. *Network science*. Cambridge university press, 2016 (vid. pág. 5).
- [9] Bazzan, A. L. y Bordini, R. H. “A framework for the simulation of agents with emotions”. En: *Proceedings of the fifth international conference on Autonomous agents*. 2001, págs. 292-299. URL: https://web.archive.org/web/20070901220903id_/http://www.vhml.org/theses/wijayat/sources/writings/papers/p292-bazzan.pdf (vid. págs. 8, 24).
- [10] Bazzan, A. L., Bordini, R. H. y Campbell, J. A. “Evolution of agents with moral sentiments in an iterated prisoner’s Dilemma exercise”. En: *Game theory and decision theory in agent-based systems*. Springer, 2002, págs. 43-64. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.5305&rep=rep1&type=pdf> (vid. págs. 8, 24).
- [11] Bazzan, A. L., Bordini, R. H. y Campbell, J. A. “Moral sentiments in multi-agent systems”. En: *International Workshop on Agent Theories, Architectures, and Languages*. Springer. 1998, págs. 113-131 (vid. pág. 8).

- [12] Bazzan, A. L., Peleteiro, A. y Burguillo, J. C. “Learning to cooperate in the Iterated Prisoner’s Dilemma by means of social attachments”. En: *Journal of the Brazilian computer society* 17.3 (2011), págs. 163-174. URL: <https://journal-bcs.springeropen.com/track/pdf/10.1007/s13173-011-0038-2.pdf> (vid. págs. 8, 24).
- [13] Bengio, Y., Frasconi, P. y Simard, P. “The problem of learning long-term dependencies in recurrent networks”. En: *IEEE international conference on neural networks*. IEEE. 1993, págs. 1183-1188 (vid. pág. 32).
- [14] Berger, J. “Arousal increases social transmission of information”. En: *Psychological science* 22.7 (2011), págs. 891-893. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.989.4895&rep=rep1&type=pdf> (vid. pág. 10).
- [15] Berger, J. *Contagious: Why things catch on*. Simon y Schuster, 2016 (vid. pág. 10).
- [16] Berger, J. y Milkman, K. L. “What makes online content viral?” En: *Journal of marketing research* 49.2 (2012), págs. 192-205. URL: <https://jonahberger.com/wp-content/uploads/2013/02/ViralityB.pdf> (vid. pág. 10).
- [17] Bloembergen, D., Tuyls, K., Hennes, D. y Kaisers, M. “Evolutionary dynamics of multi-agent learning: A survey”. En: *Journal of Artificial Intelligence Research* 53 (2015), págs. 659-697. URL: <https://asset-pdf.scinapse.io/prod/1192553058/1192553058.pdf> (vid. pág. 23).
- [18] Bond, R. M., Fariss, C. J., Jones, J. J., Kramer, A. D., Marlow, C., Settle, J. E. y Fowler, J. H. “A 61-million-person experiment in social influence and political mobilization”. En: *Nature* 489.7415 (2012), págs. 295-298. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3834737/pdf/nihms524815.pdf> (vid. págs. 1, 6).
- [19] Bourlard, H. y Kamp, Y. “Auto-association by multilayer perceptrons and singular value decomposition”. En: *Biological cybernetics* 59.4 (1988), págs. 291-294. URL: https://www.researchgate.net/profile/Herve-Bourlard/publication/19959069_Auto-Association_by_Multilayer_Perceptrons_and_Singular_Value_Decomposition/links/57600aaa08aeeada5bc2b4cc/Auto-Association-by-Multilayer-Perceptrons-and-Singular-Value-Decomposition.pdf (vid. pág. 32).
- [20] Broekens, J., Jacobs, E. y Jonker, C. M. “A reinforcement learning model of joy, distress, hope and fear”. En: *Connection Science* 27.3 (2015), págs. 215-233. URL: <https://www.tandfonline.com/doi/pdf/10.1080/09540091.2015.1031081> (vid. pág. 9).
- [21] Brunton, S. L. y Kutz, J. N. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022 (vid. pág. 33).
- [22] Cassandra, A. R. *Exact and approximate algorithms for partially observable Markov decision processes*. Brown University, 1998. URL: <https://cs.brown.edu/research/pubs/theses/phd/1998/cassandra.pdf> (vid. pág. 20).
- [23] Chen, W., Wang, J., Yu, F., He, J., Xu, W. y Wang, R. “Effects of emotion on the evolution of cooperation in a spatial prisoner’s dilemma game”. En: *Applied Mathematics and Computation* 411 (2021), pág. 126497 (vid. pág. 24).
- [24] Crandall, D., Cosley, D., Huttenlocher, D., Kleinberg, J. y Suri, S. “Feedback effects between similarity and social influence in online communities”. En: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008, págs. 160-168. URL: <https://www.cs.bgu.ac.il/~snean151/wiki.files/6-FeedbackEffectsbetweenSimilarityandSocialInfluence.pdf> (vid. pág. 5).

- [25] Crandall, J. W. “Just add Pepper: extending learning algorithms for repeated matrix games to repeated markov games”. En: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. Citeseer. 2012, págs. 399-406. URL: https://faculty.cs.byu.edu/~crandall/papers/Crandall_AAMAS2012.pdf (vid. pág. 25).
- [26] Cybenko, G. “Approximation by superpositions of a sigmoidal function”. En: *Mathematics of control, signals and systems* 2.4 (1989), págs. 303-314. URL: <http://www.vision.jhu.edu/teaching/learning/deeplearning18/assets/Cybenko-89.pdf> (vid. pág. 29).
- [27] Domingos, P. y Richardson, M. “Mining the network value of customers”. En: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. 2001, págs. 57-66. URL: <http://snap.stanford.edu/class/cs224w-readings/domingos01networkvalue.pdf> (vid. pág. 1).
- [28] Eccles, T., Hughes, E., Kramár, J., Wheelwright, S. y Leibo, J. Z. “Learning reciprocity in complex sequential social dilemmas”. En: *arXiv preprint arXiv:1903.08082* (2019). URL: <https://arxiv.org/pdf/1903.08082.pdf> (vid. págs. 8, 24).
- [29] Elidrisi, M., Johnson, N., Gini, M. L. y Crandall, J. W. “Fast adaptive learning in repeated stochastic games by game abstraction.” En: *AAMAS*. 2014, págs. 1141-1148. URL: <https://www.ifaamas.org/Proceedings/aamas2014/aamas/p1141.pdf> (vid. pág. 25).
- [30] Fan, R., Xu, K. y Zhao, J. “An agent-based model for emotion contagion and competition in online social media”. En: *Physica a: statistical mechanics and its applications* 495 (2018), págs. 245-259. URL: <https://arxiv.org/pdf/1706.02676.pdf> (vid. pág. 9).
- [31] Ferrara, E. y Yang, Z. “Quantifying the effect of sentiment on information diffusion in social media”. En: *PeerJ Computer Science* 1 (2015), e26. URL: <https://peerj.com/articles/cs-26/> (vid. pág. 10).
- [32] Fowler, J. H. y Christakis, N. A. “Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study”. En: *Bmj* 337 (2008). URL: <https://www.bmj.com/content/bmj/337/bmj.a2338.full.pdf> (vid. págs. 2, 10).
- [33] Fujimoto, S., Van Hoof, H. y Meger, D. “Addressing function approximation error in actor-critic methods”. En: *arXiv preprint arXiv:1802.09477* (2018). URL: <https://arxiv.org/pdf/1802.09477.pdf> (vid. págs. 40, 42).
- [34] Fukushima, K. “Neocognitron: A hierarchical neural network capable of visual pattern recognition”. En: *Neural networks* 1.2 (1988), págs. 119-130. URL: http://vision.stanford.edu/teaching/cs131_fall1415/lectures/Fukushima1988.pdf (vid. págs. 27, 32).
- [35] Fukushima, K. y Miyake, S. “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition”. En: *Competition and cooperation in neural nets*. Springer, 1982, págs. 267-285. URL: <https://www.cs.princeton.edu/courses/archive/spr08/cos598B/Readings/Fukushima1980.pdf> (vid. págs. 27, 32).
- [36] Fulda, N. y Ventura, D. “Predicting and Preventing Coordination Problems in Cooperative Q-learning Systems.” En: *IJCAI*. Vol. 2007. 2007, págs. 780-785. URL: <https://www.aaai.org/Papers/IJCAI/2007/IJCAI07-125.pdf> (vid. pág. 25).
- [37] Games, M. “The fantastic combinations of John Conway’s new solitaire game “life” by Martin Gardner”. En: *Scientific American* 223 (1970), págs. 120-123. URL: <https://web.stanford.edu/class/sts145/Library/life.pdf> (vid. pág. 7).

- [38] Goodfellow, I., Bengio, Y. y Courville, A. *Deep learning*. MIT press, 2016. URL: <https://www.deeplearningbook.org/> (vid. págs. 31, 33).
- [39] Haarnoja, T., Zhou, A., Abbeel, P. y Levine, S. “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor”. En: *arXiv preprint arXiv:1801.01290* (2018). URL: <https://arxiv.org/pdf/1801.01290.pdf> (vid. pág. 42).
- [40] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P. y col. “Soft actor-critic algorithms and applications”. En: *arXiv preprint arXiv:1812.05905* (2018). URL: <https://arxiv.org/pdf/1812.05905.pdf> (vid. pág. 42).
- [41] Hasselt, H. “Double Q-learning”. En: *Advances in neural information processing systems* 23 (2010). URL: <https://proceedings.neurips.cc/paper/2010/file/091d584fced301b442654dd8c/Paper.pdf> (vid. pág. 36).
- [42] Hasselt, H. P. van. *Insights in reinforcement learning*. Hado van Hasselt, 2011 (vid. pág. 36).
- [43] Hausknecht, M. y Stone, P. “Deep recurrent q-learning for partially observable mdps”. En: *arXiv preprint arXiv:1507.06527* (2015). URL: <https://arxiv.org/pdf/1507.06527.pdf> (vid. pág. 38).
- [44] Hernandez-Leal, P. y Kaisers, M. “Towards a fast detection of opponents in repeated stochastic games”. En: *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers 16*. Springer. 2017, págs. 239-257. URL: https://ir.cwi.nl/pub/27237/2017_TIRL_Hernandez.pdf (vid. pág. 25).
- [45] Hernandez-Leal, P., Kartal, B. y Taylor, M. E. “A survey and critique of multiagent deep reinforcement learning”. En: *Autonomous Agents and Multi-Agent Systems* 33.6 (2019), págs. 750-797. URL: <https://arxiv.org/pdf/1810.05587.pdf> (vid. pág. 42).
- [46] Hines, G. y Larson, K. “Learning when to take advice: A statistical test for achieving a correlated equilibrium”. En: *arXiv preprint arXiv:1206.3261* (2012). URL: <https://arxiv.org/ftp/arxiv/papers/1206/1206.3261.pdf> (vid. pág. 25).
- [47] Hochreiter, S. y Schmidhuber, J. “Long short-term memory”. En: *Neural computation* 9.8 (1997), págs. 1735-1780. URL: <https://blog.xpgreat.com/file/lstm.pdf> (vid. pág. 32).
- [48] Hornik, K., Stinchcombe, M. y White, H. “Multilayer feedforward networks are universal approximators”. En: *Neural networks* 2.5 (1989), págs. 359-366. URL: https://www.cs.cmu.edu/~epxing/Class/10715/reading/Kornick_et_al.pdf (vid. pág. 29).
- [49] Hubel, D. H. y Wiesel, T. N. “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex”. En: *The Journal of physiology* 160.1 (1962), pág. 106. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1359523/pdf/jphysiol01247-0121.pdf> (vid. págs. 27, 32).
- [50] Hughes, E., Leibo, J. Z., Philips, M. G., Tuyls, K., Duéñez-Guzmán, E. A., Castañeda, A. G., Dunning, I., Zhu, T., McKee, K. R., Koster, R. y col. “Inequity aversion resolves intertemporal social dilemmas. CoRR abs/1803.08884 (2018)”. En: *arXiv preprint arXiv:1803.08884* (2018). URL: <https://arxiv.org/pdf/1803.08884.pdf> (vid. pág. 8).
- [51] Jaakkola, T., Singh, S. y Jordan, M. “Reinforcement learning algorithm for partially observable Markov decision problems”. En: *Advances in neural information processing systems* 7 (1994). URL: <https://proceedings.neurips.cc/paper/1994/file/1c1d4df596d01da60385f0bb17a4a9e0-Paper.pdf> (vid. pág. 20).

- [52] Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P., Strouse, D., Leibo, J. Z. y De Freitas, N. “Social influence as intrinsic motivation for multi-agent deep reinforcement learning”. En: *International Conference on Machine Learning*. PMLR. 2019, págs. 3040-3049. URL: <https://arxiv.org/pdf/1810.08647.pdf> (vid. pág. 8).
- [53] Ji, S., Xu, W., Yang, M. y Yu, K. “3D convolutional neural networks for human action recognition”. En: *IEEE transactions on pattern analysis and machine intelligence* 35.1 (2012), págs. 221-231. URL: https://www.dbs.ifi.lmu.de/~yu_k/icml2010_3dcnn.pdf (vid. pág. 45).
- [54] Kempe, D., Kleinberg, J. y Tardos, É. “Influential nodes in a diffusion model for social networks”. En: *International Colloquium on Automata, Languages, and Programming*. Springer. 2005, págs. 1127-1138. URL: <https://www.cs.cornell.edu/home/kleinber/icalp05-inf.pdf> (vid. pág. 1).
- [55] Kempe, D., Kleinberg, J. y Tardos, É. “Maximizing the spread of influence through a social network”. En: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003, págs. 137-146. URL: <https://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf> (vid. pág. 1).
- [56] Konda, V. R. y Tsitsiklis, J. N. “Actor-critic algorithms”. En: *Advances in neural information processing systems*. 2000, págs. 1008-1014. URL: <https://papers.nips.cc/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf> (vid. pág. 40).
- [57] Kramer, A. D. “The spread of emotion via Facebook”. En: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2012, págs. 767-770. URL: <https://research.fb.com/wp-content/uploads/2012/05/the-spread-of-emotion-via-facebook.pdf> (vid. pág. 10).
- [58] Kramer, A. D., Guillory, J. E. y Hancock, J. T. “Experimental evidence of massive-scale emotional contagion through social networks”. En: *Proceedings of the National Academy of Sciences* 111.24 (2014), págs. 8788-8790. URL: <https://www.pnas.org/content/pnas/111/24/8788.full.pdf> (vid. pág. 10).
- [59] Krishnamurthy, V. *Partially observed Markov decision processes*. Cambridge university press, 2016 (vid. pág. 23).
- [60] Kulkarni, V. G. *Modeling and analysis of stochastic systems*. Crc Press, 2016 (vid. pág. 56).
- [61] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W. y Jackel, L. “Handwritten digit recognition with a back-propagation network”. En: *Advances in neural information processing systems* 2 (1989). URL: <https://proceedings.neurips.cc/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf> (vid. págs. 27, 32).
- [62] LeDoux, J. *The deep history of ourselves: The four-billion-year story of how we got conscious brains*. Penguin, 2020 (vid. pág. 10).
- [63] Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J. y Graepel, T. “Multi-agent reinforcement learning in sequential social dilemmas”. En: *arXiv preprint arXiv:1702.03037* (2017). URL: <https://arxiv.org/pdf/1702.03037.pdf> (vid. págs. 8, 23, 25).
- [64] Leskovec, J., Singh, A. y Kleinberg, J. “Patterns of influence in a recommendation network”. En: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2006, págs. 380-389. URL: <https://www.cs.cornell.edu/info/people/kleinber/pakdd06-cascade.pdf> (vid. pág. 6).

- [65] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. y Wierstra, D. “Continuous control with deep reinforcement learning”. En: *arXiv preprint arXiv:1509.02971* (2015). URL: <https://arxiv.org/pdf/1509.02971.pdf> (vid. pág. 41).
- [66] Littman, M. L. “A tutorial on partially observable Markov decision processes”. En: *Journal of Mathematical Psychology* 53.3 (2009), págs. 119-125. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ff7e0db75b31ab700515641a1c3982e19b9f4881> (vid. pág. 20).
- [67] Littman, M. L., Cassandra, A. R. y Kaelbling, L. P. “Learning policies for partially observable environments: Scaling up”. En: *Machine Learning Proceedings 1995*. Elsevier, 1995, págs. 362-370. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0c02d473ae301bab2d61cd196183fa0853c5bcda> (vid. pág. 20).
- [68] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. y Pietikäinen, M. “Deep learning for generic object detection: A survey”. En: *International journal of computer vision* 128 (2020), págs. 261-318. URL: <https://arxiv.org/pdf/1809.02165.pdf> (vid. pág. 3).
- [69] Marvin, M. y Seymour, A. P. “Perceptrons”. En: *Cambridge, MA: MIT Press* 6 (1969), págs. 318-362 (vid. pág. 32).
- [70] Mataric, M. J. “Reward functions for accelerated learning”. En: *Machine learning proceedings 1994*. Elsevier, 1994, págs. 181-189. URL: <https://www.sci.brooklyn.cuny.edu/~sklar/teaching/boston-college/s01/mc375/ml94.pdf> (vid. pág. 23).
- [71] Maturana, D. y Scherer, S. “Voxnet: A 3d convolutional neural network for real-time object recognition”. En: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2015, págs. 922-928. URL: http://graphics.stanford.edu/courses/cs233-21-spring/ReferencedPapers/voxnet_07353481.pdf (vid. pág. 45).
- [72] McCallum, R. A. “Instance-based state identification for reinforcement learning”. En: *Advances in Neural Information Processing Systems* 7 (1994). URL: <https://proceedings.neurips.cc/paper/1994/file/d2ed45a52bc0edfa11c2064e9edee8bf-Paper.pdf> (vid. pág. 20).
- [73] McCallum, R. A. “Instance-based utile distinctions for reinforcement learning with hidden state”. En: *Machine Learning Proceedings 1995*. Elsevier, 1995, págs. 387-395. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=f54c33a485b9fcdd3472cbe122> (vid. pág. 20).
- [74] McCulloch, W. S. y Pitts, W. “A logical calculus of the ideas immanent in nervous activity”. En: *The bulletin of mathematical biophysics* 5.4 (1943), págs. 115-133. URL: <https://waldirbertazzijr.com/wp-content/uploads/2018/10/mcp.pdf> (vid. págs. 32, 42).
- [75] Miller, M., Sathi, C., Wiesenthal, D., Leskovec, J. y Potts, C. “Sentiment flow through hyperlink networks”. En: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 5. 1. 2011. URL: <https://cs.stanford.edu/~jure/pubs/sentiflow-icwsm11.pdf> (vid. págs. 2, 10).
- [76] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. y Kavukcuoglu, K. “Asynchronous methods for deep reinforcement learning”. En: *International conference on machine learning*. 2016, págs. 1928-1937. URL: <http://proceedings.mlr.press/v48/mniha16.pdf> (vid. pág. 41).

- [77] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. y Riedmiller, M. “Playing atari with deep reinforcement learning”. En: *arXiv preprint arXiv:1312.5602* (2013). URL: <https://arxiv.org/pdf/1312.5602.pdf> (vid. pág. 34).
- [78] Moore, A. W. y Atkeson, C. G. “Prioritized sweeping: Reinforcement learning with less data and less time”. En: *Machine learning* 13.1 (1993), págs. 103-130. URL: <https://link.springer.com/content/pdf/10.1007/BF00993104.pdf> (vid. pág. 57).
- [79] Murphy, K. P. *Probabilistic machine learning: an introduction*. MIT press, 2022 (vid. pág. 29).
- [80] Neumann, J., Burks, A. W. y col. *Theory of self-reproducing automata*. Vol. 1102024. University of Illinois Press Urbana, 1966. URL: <https://cdn.patentlyo.com/media/docs/2012/04/VonNeumann.pdf> (vid. pág. 7).
- [81] Ng, A. Y., Harada, D. y Russell, S. “Policy invariance under reward transformations: Theory and application to reward shaping”. En: *Icml*. Vol. 99. 1999, págs. 278-287. URL: <http://luthuli.cs.uiuc.edu/~daf/courses/games/AIpapers/ml99-shaping.pdf> (vid. pág. 23).
- [82] Nowak, M. A. “Five rules for the evolution of cooperation”. En: *science* 314.5805 (2006), págs. 1560-1563. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3279745/pdf/nihms49939.pdf> (vid. pág. 7).
- [83] Nowak, M. A. y May, R. M. “Evolutionary games and spatial chaos”. En: *Nature* 359.6398 (1992), págs. 826-829. URL: https://www.researchgate.net/profile/Martin-Nowak/publication/216634494_Evolutionary_Games_and_Spatial_Chaos/links/54217b730cf274a67f/Evolutionary-Games-and-Spatial-Chaos.pdf (vid. págs. 23, 24).
- [84] Nowak, M. A. y Sigmund, K. “Tit for tat in heterogeneous populations”. En: *Nature* 355.6357 (1992), págs. 250-253. URL: https://abel.math.harvard.edu/archive/153_fall_04/Additional_reading_material/tit_for_tat_in_heterogenous_populations.pdf (vid. pág. 23).
- [85] Ohtsuki, H., Hauert, C., Lieberman, E. y Nowak, M. A. “A simple rule for the evolution of cooperation on graphs and social networks”. En: *Nature* 441.7092 (2006), págs. 502-505. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2430087/pdf/nihms51831.pdf> (vid. pág. 23).
- [86] Papoudakis, G., Christianos, F., Schäfer, L. y Albrecht, S. V. “Comparative evaluation of cooperative multi-agent deep reinforcement learning algorithms”. En: *arXiv: 2006.07869* (2020). URL: <https://arxiv.org/pdf/2006.07869.pdf> (vid. pág. 42).
- [87] Perc, M. y Szolnoki, A. “Coevolutionary games—a mini review”. En: *BioSystems* 99.2 (2010), págs. 109-125. URL: <https://arxiv.org/ftp/arxiv/papers/1206/1206.3261.pdf> (vid. pág. 25).
- [88] Perc, M. y Szolnoki, A. “Social diversity and promotion of cooperation in the spatial prisoner’s dilemma game”. En: *Physical Review E* 77.1 (2008), pág. 011904. URL: <https://arxiv.org/pdf/0708.1746.pdf> (vid. pág. 25).
- [89] Peysakhovich, A. y Lerer, A. “Consequentialist conditional cooperation in social dilemmas with imperfect information”. En: *arXiv preprint arXiv:1710.06975* (2017). URL: <https://arxiv.org/pdf/1710.06975.pdf> (vid. pág. 9).
- [90] Peysakhovich, A. y Lerer, A. “Maintaining cooperation in complex social dilemmas using deep reinforcement learning”. En: (2018). URL: <https://arxiv.org/pdf/1707.01068.pdf> (vid. pág. 9).

- [91] Plutchik, R. "The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice". En: *American scientist* 89.4 (2001), págs. 344-350. URL: https://www.academia.edu/43620307/The_Nature_of_Emotions_Plutchik_2001_ (vid. pág. 10).
- [92] Principe, J. C., Euliano, N. R. y Lefebvre, W. C. *Neural and adaptive systems: fundamentals through simulations with CD-ROM*. John Wiley & Sons, Inc., 1999 (vid. pág. 33).
- [93] Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014 (vid. pág. 57).
- [94] Quan, J., Zhou, Y., Ma, X., Wang, X. y Yang, J.-B. "Integrating emotion-imitating into strategy learning improves cooperation in social dilemmas with extortion". En: *Knowledge-Based Systems* 233 (2021), pág. 107550. URL: https://personalpages.manchester.ac.uk/staff/jian-bo.yang/JB%20Yang%20Journal_Papers/QuanZhouMaWangYang-KBS-Emotion-Imitating%20.pdf (vid. pág. 25).
- [95] Richardson, M. y Domingos, P. "Mining knowledge-sharing sites for viral marketing". En: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, págs. 61-70. URL: <https://homes.cs.washington.edu/~pedrod/papers/kdd02b.pdf> (vid. pág. 1).
- [96] Rolls, E. T. *Emotion and decision-making explained*. OUP Oxford, 2014 (vid. pág. 10).
- [97] Romero, D. M., Meeder, B. y Kleinberg, J. "Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter". En: *Proceedings of the 20th international conference on World wide web*. 2011, págs. 695-704. URL: <https://www.cs.cornell.edu/home/kleinber/www11-hashtags.pdf> (vid. pág. 10).
- [98] Ronneberger, O., Fischer, P. y Brox, T. "U-net: Convolutional networks for biomedical image segmentation". En: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, págs. 234-241. URL: <https://arxiv.org/pdf/1505.04597.pdf> (vid. pág. 28).
- [99] Rosenblatt, F. "The perceptron: a probabilistic model for information storage and organization in the brain." En: *Psych. r.* 65.6 (1958), pág. 386. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=65a83117cbcc4e6eb7c6ac5be8e61195dc84b9fc> (vid. págs. 32, 42).
- [100] Rumelhart, D. E., Hinton, G. E. y Williams, R. J. *Learning internal representations by error propagation*. Inf. téc. California Univ San Diego La Jolla Inst for Cognitive Science, 1985. URL: <https://apps.dtic.mil/sti/pdfs/ADA164453.pdf> (vid. pág. 32).
- [101] Sandholm, T. W. y Crites, R. H. "Multiagent reinforcement learning in the iterated prisoner's dilemma". En: *Biosystems* 37.1-2 (1996), págs. 147-166. URL: <http://ww2.odu.edu/~jsokolow/projects/files/Multiagent%20Reinforcement%20Learning%20in%20the%20Iterated%20Prisoners%20Dilema.pdf> (vid. pág. 17).
- [102] Santos, F. C. y Pacheco, J. M. "A new route to the evolution of cooperation". En: *Journal of evolutionary biology* 19.3 (2006), págs. 726-733. URL: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1420-9101.2005.01063.x> (vid. pág. 23).
- [103] Schaul, T., Quan, J., Antonoglou, I. y Silver, D. "Prioritized experience replay". En: *arXiv preprint arXiv:1511.05952* (2015). URL: <https://arxiv.org/pdf/1511.05952.pdf> (vid. pág. 37).

- [104] Schelling, T. C. “Dynamic models of segregation”. En: *Journal of mathematical sociology* 1.2 (1971), págs. 143-186. URL: https://www.stat.berkeley.edu/~aldous/157/Papers/Schelling_Seg_Models.pdf (vid. pág. 7).
- [105] Schelling, T. C. *Micromotives and macrobehavior*. WW Norton & Company, 2006 (vid. pág. 7).
- [106] Schulman, J., Levine, S., Abbeel, P., Jordan, M. y Moritz, P. “Trust region policy optimization”. En: *International conference on machine learning*. 2015, págs. 1889-1897. URL: <http://proceedings.mlr.press/v37/schulman15.pdf> (vid. pág. 39).
- [107] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. y Klimov, O. “Proximal policy optimization algorithms”. En: *arXiv preprint arXiv:1707.06347* (2017). URL: <https://arxiv.org/pdf/1707.06347.pdf> (vid. pág. 39).
- [108] Sequeira, P., Melo, F. S. y Paiva, A. “Emotion-based intrinsic motivation for reinforcement learning agents”. En: *International conference on affective computing and intelligent interaction*. Springer. 2011, págs. 326-336. URL: <https://facsmelo.github.io/publications/sequeirallacii.pdf> (vid. págs. 9, 24).
- [109] Sequeira, P., Melo, F. S. y Paiva, A. “Learning by appraising: an emotion-based approach to intrinsic reward design”. En: *Adaptive Behavior* 22.5 (2014), págs. 330-349. URL: <https://journals.sagepub.com/doi/pdf/10.1177/1059712314543837> (vid. págs. 9, 24).
- [110] Singh, S., Lewis, R. L. y Barto, A. G. “Where do rewards come from”. En: *Proceedings of the annual conference of the cognitive science society*. Cognitive Science Society. 2009, págs. 2601-2606. URL: https://all.cs.umass.edu/pubs/2009/singh_l_b_09.pdf (vid. pág. 23).
- [111] Singh, S., Lewis, R. L., Barto, A. G. y Sorg, J. “Intrinsically motivated reinforcement learning: An evolutionary perspective”. En: *IEEE Transactions on Autonomous Mental Development* 2.2 (2010), págs. 70-82. URL: <http://www-personal.umich.edu/~rickl/pubs/singh-lewis-barto-2010-ieee.pdf> (vid. pág. 23).
- [112] Singh, S. P., Jaakkola, T. y Jordan, M. I. “Learning without state-estimation in partially observable Markovian decision processes”. En: *Machine Learning Proceedings 1994*. Elsevier, 1994, págs. 284-292. URL: <https://web.eecs.umich.edu/~baveja/Papers/ML94.pdf> (vid. págs. 20, 23).
- [113] Smith, J. M. “Game theory and the evolution of behaviour”. En: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 205.1161 (1979), págs. 475-488 (vid. pág. 7).
- [114] Stella, M., Ferrara, E. y De Domenico, M. “Bots increase exposure to negative and inflammatory content in online social systems”. En: *Proceedings of the National Academy of Sciences* 115.49 (2018), págs. 12435-12440. URL: <https://www.pnas.org/content/pnas/115/49/12435.full.pdf> (vid. pág. 1).
- [115] Sutton, R. S. y Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018. URL: <http://incompleteideas.net/book/RLbook2020.pdf> (vid. pág. 17).
- [116] Sutton, R. S., McAllester, D. A., Singh, S. P. y Mansour, Y. “Policy gradient methods for reinforcement learning with function approximation”. En: *Advances in neural inf. proc. systems*. 2000, págs. 1057-1063. URL: <https://proceedings.neurips.cc/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf> (vid. pág. 39).

- [117] Szabó, G. y Fath, G. “Evolutionary games on graphs”. En: *Physics reports* 446.4-6 (2007), págs. 97-216. URL: <https://arxiv.org/pdf/cond-mat/0607344.pdf> (vid. pág. 25).
- [118] Szolnoki, A., Xie, N.-G., Wang, C. y Perc, M. “Imitating emotions instead of strategies in spatial games elevates social welfare”. En: *EPL (Europhysics Letters)* 96.3 (2011), pág. 38002. URL: <https://arxiv.org/pdf/1109.1712.pdf> (vid. págs. 9, 25).
- [119] Thrun, S. y Schwartz, A. “Issues in using function approximation for reinforcement learning”. En: *Proceedings of the 1993 Connectionist Models Summer School Hillsdale, NJ. Lawrence Erlbaum*. Vol. 6. 1993, págs. 1-9. URL: https://www.ri.cmu.edu/pub_files/pub1/thrun_sebastian_1993_1/thrun_sebastian_1993_1.pdf (vid. pág. 36).
- [120] Trivers, R. L. “The evolution of reciprocal altruism”. En: *The Quarterly review of biology* 46.1 (1971), págs. 35-57. URL: <https://pdodds.w3.uvm.edu/files/papers/others/1971/trivers1971a.pdf> (vid. pág. 23).
- [121] Tsitsiklis, J. N. “Asynchronous stochastic approximation and Q-learning”. En: *Machine learning* 16.3 (1994), págs. 185-202. URL: <https://www.mit.edu/~jnt/Papers/J052-94-jnt-q.pdf> (vid. págs. 17, 33).
- [122] Van Hasselt, H., Guez, A. y Silver, D. “Deep reinforcement learning with double q-learning”. En: *arXiv preprint arXiv:1509.06461* (2015). URL: <https://arxiv.org/pdf/1509.06461.pdf> (vid. pág. 36).
- [123] Vosoughi, S., Roy, D. y Aral, S. “The spread of true and false news online”. En: *Science* 359.6380 (2018), págs. 1146-1151. URL: <https://science.sciencemag.org/content/359/6380/1146/tab-pdf> (vid. pág. 1).
- [124] Wang, L., Ye, S.-Q., Cheong, K. H., Bao, W. y Xie, N.-g. “The role of emotions in spatial prisoner’s dilemma game with voluntary participation”. En: *Physica A: Statistical Mechanics and its Applications* 490 (2018), págs. 1396-1407. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0378437117307665> (vid. pág. 25).
- [125] Wang, W., Hao, J., Wang, Y. y Taylor, M. “Towards cooperation in sequential prisoner’s dilemmas: a deep multiagent reinforcement learning approach”. En: *arXiv preprint arXiv:1803.00162* (2018). URL: <https://arxiv.org/pdf/1803.00162.pdf> (vid. pág. 25).
- [126] Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M. y Freitas, N. “Dueling network architectures for deep reinforcement learning”. En: *International conference on machine learning*. 2016, págs. 1995-2003. URL: <http://proceedings.mlr.press/v48/wangf16.pdf> (vid. pág. 38).
- [127] Watkins, C. J. y Dayan, P. “Q-learning”. En: *Machine learning* 8.3 (1992), págs. 279-292. URL: <https://link.springer.com/content/pdf/10.1007/BF00992698.pdf> (vid. págs. 16, 17).
- [128] Wu, S., Tan, C., Kleinberg, J. y Macy, M. W. “Does bad news go away faster?” En: *Fifth International AAAI Conference on Weblogs and Social Media*. 2011. URL: <https://www.cs.cornell.edu/home/kleinber/icwsm11-longevity.pdf> (vid. pág. 10).
- [129] Wu, Y., Mansimov, E., Grosse, R. B., Liao, S. y Ba, J. “Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation”. En: *Advances in neural information processing systems*. 2017, págs. 5279-5288. URL: <https://arxiv.org/pdf/1708.05144.pdf> (vid. pág. 39).
- [130] Wunder, M., Littman, M. L. y Babes, M. “Classes of multiagent q-learning dynamics with epsilon-greedy exploration”. En: *ICML*. 2010. URL: <https://icml.cc/Conferences/2010/papers/191.pdf> (vid. pág. 23).

-
- [131] Yu, C., Zhang, M., Ren, F. y Tan, G. “Emotional multiagent reinforcement learning in spatial social dilemmas”. En: *IEEE transactions on neural networks and learning systems* 26.12 (2015), págs. 3083-3096. URL: https://www.researchgate.net/profile/Minjie-Zhang-4/publication/273463080_Emotional_Multiagent_Reinforcement_Learning_in_Spatial_Social_Dilemmas/links/552309bc0cf29dcabb0ee0c7/Emotional-Multiagent-Reinforcement-Learning-in-Spatial-Social-Dilemmas.pdf (vid. págs. 9, 25).
- [132] Yuan, Y., Guo, T., Zhao, P. y Jiang, H. “Adherence Improves Cooperation in Sequential Social Dilemmas”. En: *Applied Sciences* 12.16 (2022), pág. 8004. URL: <https://www.mdpi.com/2076-3417/12/16/8004> (vid. pág. 23).
- [133] Zafarani, R., Cole, W. D. y Liu, H. “Sentiment propagation in social networks: a case study in livejournal”. En: *International Conference on Social Computing, Behavioral Modeling, and Prediction*. Springer. 2010, págs. 413-420. URL: https://www.researchgate.net/profile/Huan-Liu-51/publication/221118290_Sentiment_Propagation_in_Social_Networks_A_Case_Study_in_LiveJournal/links/00b495350968dcc61000000/Sentiment-Propagation-in-Social-Networks-A-Case-Study-in-LiveJournal.pdf (vid. pág. 10).

Glosario

Montecarlo (MC) En el ámbito del Aprendizaje por Refuerzo (*Reinforcement Learning*) alude al método de aprendizaje, para la estimación de las funciones de valor, que no requiere conocimiento del entorno, y sólo se basa en secuencias muestrales generadas por simulación del tipo: estado, acción y recompensa.