UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Centro Asociado de Cantabria

INTELIGENCIA ARTIFICIAL: MODELOS MATEMÁTICOS QUE IMITAN A LA NATURALEZA

Lección Inaugural del Curso 2000-2001

ÁNGEL COBO ORTEGA
Profesor Tutor

CONTENIDOS

PRÓLOGO

Cuando se me encomendó la labor de impartir la Lección Inaugural del curso 2000-2001 del Centro Asociado de la UNED en Cantabria, percibí una extraña mezcla de sensaciones. Por un lado, la alegría y el orgullo de haber sido elegido para tan importante acontecimiento. Por otro lado, la responsabilidad de ser el encargado de pronunciar la primera lección inaugural en el área científico-tecnológica, en la aún breve historia de este tipo de actos en el Centro Asociado. Finalmente, llegó la sensación de duda a la hora de decidir el tema de mi exposición. Estando a punto de terminar el año 2000, declarado por la Unión Matemática Internacional (UMI), y después confirmado por la UNESCO, como *Año Mundial de las Matemáticas*, y siendo yo además matemático, la decisión parecía clara: mi exposición debía de tratar de mostrar la importancia de las Matemáticas en la Ciencia moderna. Para mi propósito traté de buscar un tema que pudiera resultar ilustrativo de esa importancia, pero también atractivo para un amplio y variado auditorio. Tras barajar diferentes opciones opté finalmente por dedicar mi intervención a introducir algunos de los métodos matemáticos más conocidos dentro de la Inteligencia Artificial. Considero que se trata de un tema muy adecuado para los propósitos antes comentados. Pero mi lección no será una lección de matemáticas tradicional, ya que difícilmente contribuiría a mejorar la imagen actual de esta disciplina si enfocara mi intervención de esa manera en un acto tan particular como este, teniendo en cuenta sobre todo lo variado del auditorio. En mi exposición aparecerán términos poco esperados en una lección de matemáticas: neuronas, genética, mutaciones, evolución de especies,... Espero que el enfoque y tema elegido resulte interesante e ilustrativo de la contribución de las matemáticas al desarrollo tecnológico.

Angel Cobo Ortega
Santander, Noviembre de 2000

INTELIGENCIA ARTIFICIAL: MODELOS MATEMÁTICOS QUE IMITAN A LA NATURALEZA

1. INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL

1.1. ¿MÁQUINAS QUE PIENSAN?

"¿Pueden pensar las máquinas?". Esta es la provocadora pregunta con la que el matemático Alan M. Turing en 1950 iniciaba el que es considerado por muchos el primer artículo científico sobre "Inteligencia Artificial". En honor a este genio matemático fallecido prematuramente he considerado oportuno empezar esta lección inaugural con esa misma pregunta.

Alan M. Turing
(1912-1954)

En sus trabajos, Alan Turing sentó algunas de las bases teóricas sobre las que en nuestros días se apoya la "Inteligencia Artificial"; entre otros son célebres sus trabajos sobre diseño teórico de máquinas capaces de resolver cualquier problema con solución algorítmica (las *máquinas de Turing*), o el conocido como *test de Turing*, prueba que debía ser superada por toda máquina que se considerara "inteligente". Pero, como se verá a lo largo de las siguientes páginas, el caso de Turing no es un caso aislado, ilustres matemáticos de distintas épocas han tenido siempre grandes inquietudes por el tema de la Inteligencia Artificial y han contribuido notablemente al desarrollo de esta disciplina. En nuestros días la Inteligencia Artificial sigue siendo uno de los campos más importantes de investigación para especialistas en matemática aplicada.

Antes de tratar de dar una respuesta a la profunda y trascendental pregunta con la que se iniciaba esta lección, es conveniente intentar

definir lo que se entiende por "inteligencia", para poder posteriormente analizar si una máquina puede llegar a poseer esa virtud. No es una tarea sencilla dar esa definición; por ejemplo, Wégnez (1987) presenta más de una veintena de definiciones diferentes, para terminar resumiéndolas diciendo:

> *"La inteligencia es la facultad de percibir, comprender, interpretar, juzgar y reaccionar de una manera que responde a ciertos criterios externos".*

Se podría citar también la definición dada por el prestigioso psicopedagogo suizo Piaget:

> *"La inteligencia es un término genérico que designa las formas superiores de organización o de equilibrio de las estructuras cognoscitivas (...) La inteligencia es esencialmente un sistema de operaciones vivientes y actuantes. Es la adaptación mental más avanzada, es decir, el instrumento indispensable de los intercambios entre el sujeto y el universo."*

¿Es la condición de ser vivo esencial para poder hablar de inteligencia?. Evidentemente, no se puede considerar a un ser vivo como una máquina que calcula, o una sofisticadísima computadora, los sistemas biológicos son mucho más complejos y muy diferentes de los sistemas informáticos. Hoy en día se conoce perfectamente el funcionamiento de los más modernos ordenadores, de hecho el hombre los ha creado, pero investigar en el funcionamiento del œrebro sigue siendo una de las aventuras más apasionantes de la Ciencia. Se han hecho muchos avances en el último siglo, pero aún queda mucho camino por recorrer. Precisamente, el Premio Nobel de Medicina de este año 2000 ha sido otorgado a tres investigadores, Greengard, Kandel y Carlsson, por sus avances en el estudio del comportamiento de las células cerebrales.

A pesar de que el cerebro y los ordenadores sean difícilmente comparables en cuanto a estructura, complejidad y funcionamiento, tampoco se puede descartar la posibilidad de que la máquina pueda *simular la inteligencia del ser vivo.* La potencia y velocidad de cálculo de los actuales ordenadores es deslumbrante, es impensable que un humano sea capaz de procesar millones de operaciones por segundo, pero también son innumerables las tareas que realiza el cerebro humano y ni los más modernos supercomputadores pueden realizar. Cuesta imaginar el poder

de una máquina que sea capaz de combinar las habilidades de los humanos y de los ordenadores, éste es a grandes rasgos el objetivo de la Inteligencia Artificial.

Dos de los investigadores pioneros en el campo de la Inteligencia Artificial, A. Barr y E.A. Feigenbaum (1981), dan la siguiente definición:

> *"La Inteligencia Artificial es la parte de la Ciencia que se ocupa del diseño de sistemas de computación inteligentes, es decir, sistemas que exhiben las características que asociamos a la inteligencia en el comportamiento humano referidas a la comprensión del lenguaje, el aprendizaje, el razonamiento, la resolución de problemas, etc."*

Está claro que toda máquina que trate de simular la inteligencia, debe cumplir una condición esencial: la capacidad de *aprendizaje*; es decir, acumular y conservar información acerca de las experiencias pasadas y poder usar esa información para actuaciones futuras. Básicamente, un proceso de inteligencia artificial debería ser capaz de:

- Modificar su información como resultado de sus interacciones.
- Relacionar la información presente con las pasadas de una manera inteligente, es decir, haciendo comparaciones útiles y seleccionando las relaciones importantes e interesantes.
- Obtener conclusiones a partir de la información acumulada.
- Justificar las conclusiones obtenidas y el proceso utilizado para llegar a ellas.

La Inteligencia Artificial es un área con conexiones a muchas y muy diversas disciplinas, es por ello que en su desarrollo, tanto teórico como práctico, es muy común encontrar grupos de investigación multidisciplinares. Ya se ha comentado que el objeto principal de esta lección es mostrar el papel de las matemáticas en algunas de las técnicas más conocidas en la Inteligencia Artificial, pero las conexiones con otras ramas de la Ciencia son claras, desde la Informática a la Medicina pasando por la Ingeniería, la Psicología o la Pedagogía, por ejemplo.

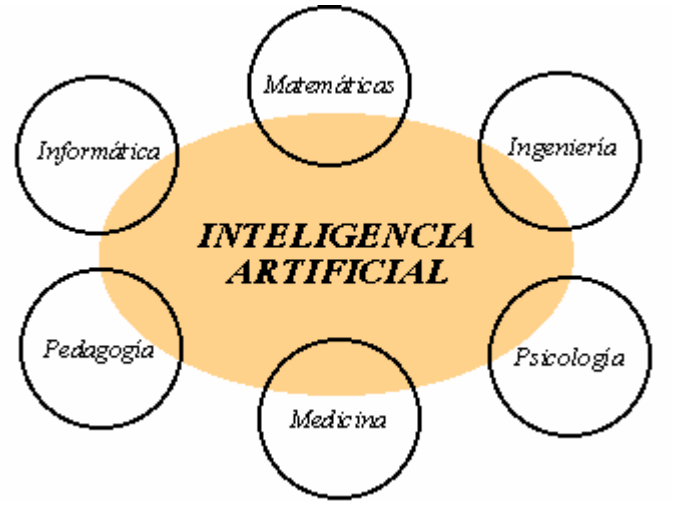


Figura 1: Conexiones de la Inteligencia Artificial con otras disciplinas.

1.2. LA ESCUELA CONEXIONISTA

Una de las escuelas de mayor peso dentro de la Inteligencia Artificial actualmente es la denominada *escuela conexionista* o *IA basada en la naturaleza*, fundada por Newell y Simon de la Universidad Carnegie-Mellon. Según esta escuela para construir máquinas inteligentes se deben desarrollar modelos que imiten el funcionamiento del cerebro humano o, en definitiva, a la naturaleza. A lo largo de las próximas secciones se tratarán de mostrar ejemplos de modelos matemáticos que siguen estas directrices y que están siendo utilizados exitosamente en la resolución de problemas de inteligencia artificial aplicados a una gran variedad de campos.

El primer ejemplo de este tipo de modelos lo constituye las conocidas como *redes neuronales artificiales.* Como se verá en la Sección 2, se trata de modelos matemáticos inspirados en las características neurofisiológicas del cerebro.

Como segundo ejemplo de modelos matemáticos que imitan a la naturaleza se presentarán, en la Sección 3, los conocidos como *algoritmos genéticos*. Se trata en este caso de algoritmos de búsqueda basados en los principios de la selección natural.

Sin embargo, antes de pasar la exposición de esos modelos, puede resultar gratificante dar un paseo por la historia. Desde los primeros intentos del hombre por crear máquinas que le imitasen, hasta la aparición de esos primeros modelos inteligentes. Puede descubrirse

también el importante papel que ilustres matemáticos de todos los tiempos han jugado en el desarrollo de esta disciplina.

1.3. HISTORIA DE LA INTELIGENCIA ARTIFICIAL

1.3.1. PRECURSORES

Desde tiempos remotos el hombre ha intentado crear autómatas que simulasen la forma y las habilidades de los seres humanos. La creación de figuras animadas, más o menos complejas, de acuerdo al desarrollo tecnológico de cada época, ha sido una constante en la historia de la humanidad.

La Grecia Clásica fue una época especialmente prolífica en este campo, los griegos fueron verdaderos maestros en la construcción de todo tipo de mecanismos automáticos que emulaban los movimientos de los seres vivos. Se podrían citar, por ejemplo, los construidos por Arquímedes para proteger a Siracusa del asedio de la flota romana o la paloma que batía sus alas construida por Architas de Tarente. La construcción del primer autómata con forma humana, sin embargo, se debe a Herón de Alejandría, matemático griego apasionado por la ingeniería y la mecánica que creó actores artificiales para representar una obra sobre la Guerra de Troya. No obstante, es en la mitología griega donde aparecen las primeras referencia a "androides"; así por ejemplo, se atribuye a Hefestos (dios del fuego y primer herrero) la fabricación de los primeros "androides" inteligentes de forma humana.

Hay que esperar la Edad Media para encontrar la continuación de estos esfuerzos. Se podrían citar, por ejemplo, a San Alberto Magno, constructor de un "mayordomo" que abría la puerta y saludaba al visitante.

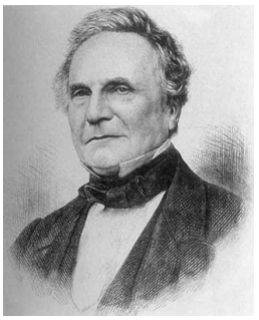
Ya en la Edad Moderna, se pueden encontrar nuevos ejemplos, entre los que es ineludible citar a los Droz, padre e hijo, quienes construyeron en la segunda mitad del siglo XVIII tres androides que maravillaron a Europa. Mediante complejos mecanismos de relojería los androides tocaban el órgano y escribían mensajes y dibujos en papel. Los dibujos, mensajes y melodías podían variarse mediante un "programa" y seis horas de trabajo de un experto relojero. Lamentablemente, cuando los Droz mostraron sus creaciones en nuestro país fueron procesados y encarcelados por la Inquisición.

Dos de los matemáticos más conocidos del siglo XVII, Pascal y Leibnitz, fueron los primeros en crear máquinas automáticas capaces de realizar cálculos matemáticos. La máquina aritmética de Pascal y la

Blaise Pascal (1623-1662)

Leibnitz (1646-1716)

calculadora universal de Leibnitz pueden ser consideradas como las precursoras de los actuales ordenadores. Durante los siglos XIX y comienzos del siglo XX se pueden encontrar nuevos ejemplos de máquinas automáticas desarrolladas principalmente por matemáticos. En 1833 el matemático inglés y profesor de la Universidad de Cambridge Charles Babbage, junto con su colaboradora Ada Lovelace, hija de Lord Byron, diseñaron su "máquina analítica", capaz de realizar operaciones matemáticas, cálculos de logaritmos y funciones trigonométricas, con posibilidad de ser programada mediante tarjetas perforadas y almacenar números en su interior. Debido a las diferencias tecnológicas de la época y la falta de ayudas económicas[1], esta máquina no llegó a construirse a pesar de que sus inventores conocían todo lo necesario para su construcción. De hecho, años después aparecieron algunas máquinas con su diseño. Por el diseño de su máquina analítica, Babbage es considerado actualmente como el padre de la Informática. Entre los inventores que utilizaron los diseños de Babbage se encuentra el ingeniero cántabro Leonardo Torres Quevedo (1852-1936), que a principios del siglo XX construyó su máquina calculadora y una máquina para jugar al ajedrez.



Charles Babbage (1791-1871)

Desde luego, la historia está llena también de anécdotas e intentos fraudulentos de presentar "máquinas inteligentes". Puede citarse, por ejemplo, el caso del "inventor" húngaro Von Kempelen, que maravilló Europa con su máquina "Malzel Chess Automaton", capaz de vencer a los mejores jugadores de ajedrez de su época. El invento llegó a entusiasmar a la mismísima emperatriz María Teresa de Austria, hasta

[1] Cuando el gobierno británico retiró los fondos a Babbage para la construcción de su máquina analítica, Disraeli, primer ministro de ese gobierno, mordazmente dijo: "para lo único que podría servir aquel aparato sería para calcular las enormes sumas de dinero público que se habían derrochado ya con él".

que se descubrió el fraude: lo que realmente encerraba la máquina era un enano, de nombre Algaller, experto jugador de ajedrez.

Todos los ejemplos señalados, y otros muchos que no han sido reflejados en esta líneas, están muy lejos de la concepción de "máquinas inteligentes" de la mitología griega. Se trata de simples mecanismos que repetían las acciones fijas y determinadas para las cuales habían sido diseñados, sin tener ninguna capacidad de aprendizaje ni improvisación. El propio Babbage reconocía tras construir su máquina analítica:

> *"La máquina analítica no tiene pretensión alguna de originar nada. Puede hacer todo aquello que sepamos ordenarle que haga. Puede realizar un análisis, pero no tiene capacidad para prever ninguna verdad o relación analítica. Su misión es ayudarnos a facilitar lo que nosotros ya conocemos."*

1.3.2. ORIGENES DE LA INTELIGENCIA ARTIFICIAL

El origen inmediato del concepto y de los criterios de desarrollo de la "Inteligencia Artificial" (IA) debe situarse en la segunda mitad del siglo XX. Para muchos autores el primer artículo científico en el que se sentaban las bases de lo que hoy conocemos como Inteligencia Artificial fue escrito por el matemático Alan M. Turing (1912-1954) en 1950, tal como ya se comentó al inicio de esta lección. Alan Turing inventó, entre otras cosas, una máquina descifradora de los mensajes de las tropas nazis y desarrolló las bases teóricas de una máquina capaz de resolver todo tipo de problemas con solución algorítmica. Otra de las aportaciones de Turing a la IA fue el diseño de un test, el conocido como "test de Turing". Una máquina, según Turing, podía considerarse inteligente cuando era capaz de pasar ese test satisfactoriamente.

La Segunda Guerra Mundial generó una necesidad urgente de nuevas técnicas destinadas a procesar datos, es decir, obtener informaciones nuevas a partir de unos datos iniciales. En esta época se sitúa también el nacimiento de una nueva disciplina: la cibernética, ciencia del control y la comunicación entre el hombre y las máquinas.

En los años cuarenta, un equipo interdisciplinar formado por antropólogos, fisiólogos, matemáticos, psicólogos y un economista, y bajo la dirección del matemático Norbert Weiner del M.I.T (Massachusett Institute of Technology), se enfrentó al problema de las trayectorias de proyectiles dirigidos hacia objetos en movimientos, como pueden ser, por ejemplo, los aviones enemigos. Para acertar, debe

predecirse la posición futura del blanco, y corregirse la trayectoria si éste cambia de dirección. Advirtieron que era un problema semejante al que resuelve el cerebro cuando conduce la mano para recoger un objeto (estático o en movimiento), de forma que su objetivo fue crear un aparato que imitaría los procesos de control existentes en el ser humano. Con la publicación en 1949 del libro "Cybernetics", basado en los trabajos del equipo dirigido por Weiner, se sentaron las bases de la cibernética, disciplina rectora de los procedimientos automáticos.

Otro equipo interdisciplinar que se ha ganado un reconocido prestigio en el campo de la Inteligencia Artificial, es el formado por McCulloch del Colegio de Medicina de la Universidad de Illinois y Pitts, matemático del M.I.T. En los años cuarenta, basándose en la idea, no del todo acertada, de que las neuronas cerebrales eran binarias, McCulloch y Pitts intentaron usar la lógica booleana y los circuitos eléctricos binarios para generar comportamiento inteligente, intentando que los ordenadores simulasen la actividad cerebral. Sus trabajos sentaron las bases de lo que hoy se conoce como redes neuronales artificiales.

El término "Inteligencia Artificial" se debe, sin embargo, a McCarthy, uno de los integrantes del "Grupo de Darmouth", un grupo de investigadores que se reunió en el verano de 1956 en el Darmouth College para discutir la posibilidad de construir máquinas que no se limitaran a hacer cálculos prefijados sino operaciones genuinamente "inteligentes". Otros integrantes de aquel famoso grupo eran Samuel, autor de un programa informático de juego de damas capaz de aprender de su propia experiencia; Minsky, matemático que trabajaba sobre razonamientos analógicos de geometría; Bernstein, que había creado un programa para jugar al ajedrez; Selfridge, que se dedicaba al reconocimiento de imágenes por ordenador, Newell, Shaw y Simon, que habían construido un programa para la demostración automática de teoremas, y algunos otros. Fueron los verdaderos iniciadores en el campo de investigación que McCarthy bautizó como "Inteligencia Artificial".

A partir de ese momento se formaron dos grandes escuelas de Inteligencia Artificial:

1. Por un lado, Newell y Simon lideraron el equipo formado en 1956 en la Universidad de Carnegie-Mellon, cuyo propósito general era el desarrollar modelos cuya estructura se pareciese lo más posible al funcionamiento del cerebro humano. Como ya se comentó anteriormente, esta es la base de la denominada *escuela conexionista* o *IA basada en la naturaleza*. El más claro ejemplo de trabajo en esta línea lo constituyen los estudios sobre redes neuronales artificiales.

2. Por otro lado, McCarthy y Minsky formaron otro equipo en el M.I.T, centrándose más en que los productos del procesamiento tengan el carácter de inteligente, sin preocuparse por que el funcionamiento o la estructura de los componentes sean parecidas a los del ser humano.

La escuela del M.I.T tuvo su mayor apogeo en sus primeros 20 años, aunque los investigadores se encontraron con que sus sistemas sucumbían ante la creciente longitud y complejidad de su programación. Quienes tenían más presente las peculiaridades del cerebro humano y optaron por el enfoque propuesto en la Universidad de Carnegie-Mellon no fueron sorprendidos, y así la escuela conexionista inició un rápido crecimiento.

1.4. ALGUNAS ÁREAS DE LA INTELIGENCIA ARTIFICIAL

Los años sesenta fueron unos años de gran desarrollo en los conceptos teóricos sobre los que hoy en día se apoya la I.A. Sin embargo, una vez más, como ya había ocurrido muchas veces antes en la historia de la Ciencia, la tecnología de la época no estaba al nivel del conocimiento científico alcanzado. El desarrollo posterior de la Informática permitió un rápido progreso de la Inteligencia Artificial y el notable aumento de su ámbito de aplicación, así como el desarrollo de nuevas técnicas.

Los primeros intentos de crear "máquinas inteligentes" fueron encaminados hacia máquinas con capacidades perceptuales y habilidades psicomotoras (robótica), comprensión del lenguaje natural, identificación de patrones, máquinas de juegos de estrategia, desarrollo de sistemas capaces de emular la pericia de un experto humano en un ámbito del conocimiento (sistemas expertos),... En las próximas secciones se trata de dar una visión muy general de estos y algunos otros campos actuales de investigación en Inteligencia Artificial.

1.4.1. SISTEMAS EXPERTOS

El avance teórico en I.A permitió que surgiera una línea de trabajo orientada a diseñar productos útiles y rentables para muy variados campos profesionales: los *sistemas expertos*, programas de consulta capaces de ayudar a resolver dudas en determinados campos.

Un sistema experto debe ser capaz de aplicar el conocimiento propio de los expertos humanos a la resolución de problemas específicos y, más aún, de formular predicciones, razonar en situaciones deterministas e inciertas, proporcionar explicaciones o aconsejar al usuario en la toma de decisiones. Se puede decir, por tanto, que un sistema experto es un sistema informático (hardware y software) que simula a los expertos humanos en un área de especialización dada. Feigenbaum, un pionero en sistemas expertos, dice:

> *Un sistema experto es un programa inteligente de ordenador que usa procedimientos cognoscitivos y de inferencia para resolver problemas suficientemente difíciles como para requerir pericia humana para su solución. Los conocimientos necesarios para trabajar a tal nivel, más los procedimientos de inferencia utilizados, pueden ser considerados como un modelo de competencia de los mejores especialistas de ese campo.*

Durante los últimos años las aplicaciones de los sistemas expertos a diversas áreas se han multiplicado. En la actualidad, se utilizan sistemas expertos en campos como la matemática, química, medicina, biología, geología, física, electrónica, economía, gestión empresarial,... Por ejemplo, en Durkin (1994) puede encontrarse una clasificación de más de 2.500 sistemas expertos por áreas de aplicación. Según se deduce de dicho estudio, la industria, la economía, el campo empresarial y la medicina son las áreas en las que se han desarrollado un mayor número de sistemas expertos.

Algunas de las ventajas de la utilización de los sistemas expertos pueden ser:

- Posibilidad de uso por parte de personal con poca experiencia.
- El conocimiento de varios expertos humanos puede combinarse.
- La velocidad de respuesta del sistema experto es mucho mayor.
- En determinados casos la complejidad del problema a abordar hace que las respuestas de un experto humano sean menos fiables que el sistema experto.

Básicamente se distinguen dos tipos de sistemas expertos, según la naturaleza de problemas para los que han sido diseñados: deterministas (basados en reglas) y estocásticos (basados en probabilidad). En los sistemas expertos deterministas el conocimiento del experto se representa mediante un conjunto de reglas y una estructura de control que permite ir

encadenándolas para llegar a obtener conclusiones (razonamiento lógico). En los sistemas expertos probabilísticos se introduce la probabilidad como medida de incertidumbre de las reglas y sus premisas. En este caso la estrategia de razonamiento se conoce como inferencia probabilística. Una descripción detallada de este tipo de sistemas expertos puede encontrarse en Castillo, Gutiérrez y Hadi (1996).

Los elementos básicos de todo sistema experto son:

1. *Base de conocimientos*: diseñada a partir del conocimiento de un experto o grupo de expertos humanos, está formada por reglas de validez general, distribuciones de probabilidad,...
2. *Motor de inferencia*: sistema de procesamiento lógico o probabilístico que permite obtener una conclusiones a partir de unos datos.
3. *Interfase de usuario*: medio de comunicación entre el usuario y la máquina. A través de la interfase el sistema solicita datos al usuario y le muestra las conclusiones obtenidas.
4. *Subsistema de adquisición de conocimiento*: controla la forma en la que nuevo conocimiento es incorporado a la base de conocimentos.
5. *Subsistema de explicación*: se encarga de justificar las conclusiones obtenidas.

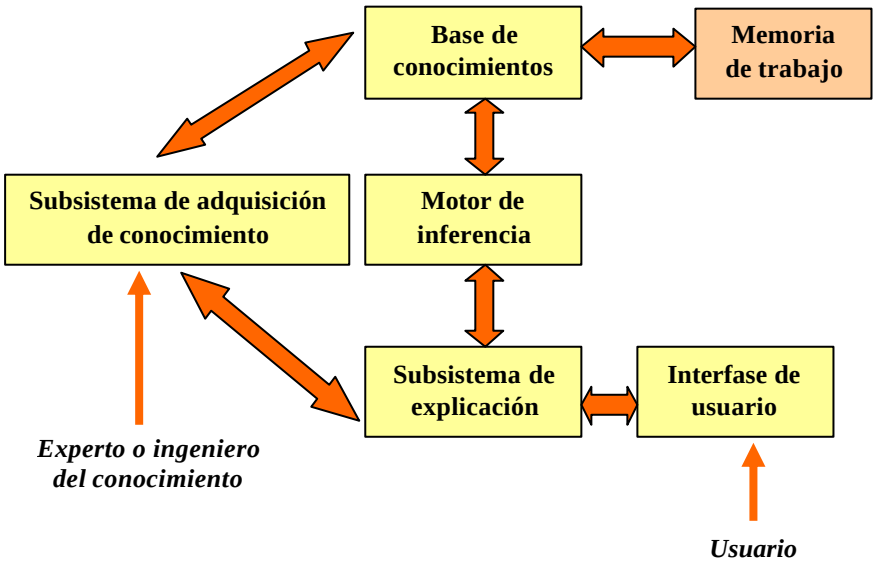


Figura 2: Esquema de funcionamiento de un Sistema Experto.

1.4.2. COMUNICACIÓN HOMBRE-MÁQUINA MEDIANTE LENGUAJE NATURAL

La comunicación entre hombres y máquinas usando un lenguaje natural, es decir, el lenguaje común diario, es otro de los campos típicos de la Inteligencia Artificial.

Uno de los primeros intentos de lograr esa comunicación con un ordenador, quizás el más famoso, lo constituyó el programa ELIZA creado por Weizenbaum[2] en 1966, en el Instituto Tecnológico de Massachusetts. Weizenbaum, se convertiría más tarde en uno de los enemigos más acérrimos de la Inteligencia Artificial. El programa ELIZA era capaz de mantener conversaciones más o menos coherentes simulando el diálogo entre un psiquiatra de la escuela "Rogeriana" y una paciente neurótica. Fue posiblemente el primer programa que logró dar la apariencia de un diálogo entre el usuario y la máquina. El programa se basaba en un sistema de palabras clave junto con un conjunto de reglas que permitían formular oraciones. Era, por lo tanto, un automatismo expresivo sin verdadero contenido significativo. Antes o después, cualquier usuario inteligente terminaba descubriendo que "algo andaba mal", como lo expresó un psiquiatra que probó sus virtudes: *"El interlocutor (la máquina) tiene una muy seria lesión cerebral"*. Este programa ha sido también el protagonista de divertidas anécdotas, como la que se recoge a continuación.

ANÉCDOTA:

Daniel Bobrow, en aquel momento investigador del centro de cálculo de BBN Inc, empresa de proyectos relacionados con la IA, tenía una versión del programa ELIZA instalada en su ordenador. Un sábado por la mañana uno de los vicepresidentes de la compañía llegó con unos clientes con la intención de realizar una demostración comercial de un sistema que se estaba desarrollando en la empresa. Al intentar utilizar el ordenador de Bobrow el vicepresidente se encontró que en el ordenador aparecía el siguiente mensaje: *"Teclee sus mensajes y acabe con un punto"*. El vicepresidente al pensar que el equipo estaba conectado con el domicilio de Bobrow inició el siguiente diálogo pensando que estaba hablando con Bobrow, cuando en realidad lo hacía con una máquina:

[2] Weizenbaum, J. *A Computer Program for the Study of Natural Language, Communication Between Man and Machine*. Communications of the ACM 9, 1966.


¿Puedo utilizar su ordenador esta mañana?.


¿Por qué me pregunta eso?


Pienso que podría hacer unas ventas adicionales de nuestro sistema.


¿Por qué no está seguro?


Mis clientes no han visto nunca una demostración del sistema.


¿Significa esto mucho para usted?


Por supuesto.


Parece usted muy seguro


Esto es absurdo.


¿Qué quiere usted decir con que esto es absurdo?


¿Existe alguna razón por la que no pueda utilizar el ordenador ahora?.


Cuénteme más cosas de sus clientes


Por favor, telefonéeme al 491-1850

Ante el enfado del vicepresidente, éste se olvido de terminar su última frase con un punto, de ahí que la conversación terminara de esa manera. Por supuesto, Bobrow tuvo que dar explicaciones al vicepresidente de la compañía.

1.4.3. DEMOSTRACIÓN AUTOMÁTICA DE TEOREMAS

Incluso antes de que el término Inteligencia Artificial empezara a hacerse popular, algunos investigadores trabajaban con el objetivo de crear programas que permitiesen demostrar teoremas matemáticos y lógicos. La capacidad de hacer deducciones lógicas fue considerada durante mucho tiempo como una posibilidad reservada a la mente humana, sin embargo, los "demostradores automáticos de teoremas" han sido ya utilizados exitosamente en diversas áreas de la matemática, como pueden ser la geometría y la lógica proposicional.

Algunas referencias básicas sobre estos temas pueden ser Bundy (1983) y Wos et al (1984).

1.4.4. RECONOCIMIENTO DE PATRONES

El reconocimiento de patrones se basa en la utilización de distintas técnicas de clasificación para identificar subgrupos con características comunes en cada grupo. Estas técnicas se utilizan, entre otras cosas, en procesos de reconocimiento de imágenes, reconocimiento de señales, diagnóstico de fallos,... Las técnicas de clasificación y reconocimiento de patrones siguen siendo objeto actualmente de estudio tanto a nivel teórico como práctico.

1.4.5. PERCEPCIÓN Y RECONOCIMIENTO DE FORMAS

Una de las aplicaciones más interesantes de la Inteligencia Artificial es la de la imitación de las capacidades sensoriales de los seres humanos, tanto audición como visión artificial. No se trata solamente que el ordenador sea capaz de percibir sonidos e imágenes, sino de identificar el sentido de lo percibido.

Una introducción general al problema de reconocimiento de voz puede verse en Rabiner y Juang (1993).

En cuanto al tema de la visión artificial, se ha avanzado mucho en el reconocimiento de formas bidimensionales, el caso de formas tridimensionales es objeto actualmente de grandes esfuerzos investigadores. Una referencia básica en el tema de visión artificial es Saphiro y Rosenfeld (1992).

1.4.6. ROBÓTICA

La robótica es una de las áreas más populares dentro de la Inteligencia Artificial. Se trata de la creación de máquinas con capacidades perceptuales y habilidades psicomotoras. Los robots combinan elementos mecánicos, sensores, dispositivos electrónicos,... con técnicas de I.A. para conseguir que interactuen con el mundo real.

1.4.7. TÉCNICAS DE PLANIFICACIÓN Y RESOLUCIÓN AUTOMATIZADA DE PROBLEMAS POR BÚS QUEDA ASISTIDA.

La resolución de determinados problemas puede ser considerada como una búsqueda entre posibilidades alternativas. El espacio de búsqueda puede ser representado como una estructura jerárquica llamada árbol. La solución del problema consiste en buscar una ruta desde el nodo raíz (estado inicial) a través de las diferentes “ramas” del árbol hasta encontrar el nodo terminal objetivo (estado final).

Cuando el problema tiene una gran complejidad, es imposible analizar directamente todas las posibles rutas del árbol, es entonces cuando intervienen algoritmos inteligentes de búsqueda.

1.4.8. ALGORITMOS DE APRENDIZAJE

El aprendizaje, desde un punto de vista pragmático, consiste en la adaptación de los parámetros de un sistema (sea artificial o natural) para obtener una respuesta deseada frente a un estímulo externo. El desarrollo de nuevos algoritmos matemáticos de aprendizaje sigue ocupando un lugar destacado en la investigación relacionada con la Inteligencia Artificial.

1.4.9. AGENTES INTELIGENTES

En los últimos años, dentro de la Inteligencia Artificial se están dedicando importantes esfuerzos a la construcción de programas conocidos como *agentes inteligentes*. Se trata de programas capaces de llegar a averiguar los gustos o preferencias del usuario y adaptarse a ellos. Aunque son muchos los posibles usos, uno de los más destacados es la búsqueda de información en Internet. En la red pueden encontrarse ya programas que son capaces de llegar a identificar los temas de interés

de un usuario particular a partir de las búsquedas que habitualmente realiza, una vez identificados esos temas pueden actuar como filtros, mostrando al usuario únicamente la información que le pueda resultar relevante, o informando automáticamente de la aparición de nuevas páginas sobre sus temas de interés.

2. REDES NEURONALES

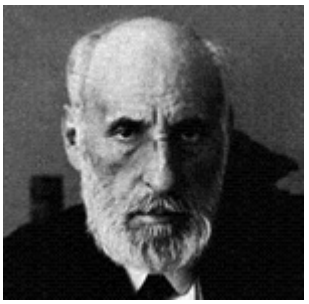
Las Redes Neuronales Artificiales (RNA) constituyen una de las áreas de mayor actividad investigadora en Inteligencia Artificial. Las RNA son modelos computacionales que tratan de imitar la estructura y funcionamiento del cerebro humano. Aunque los primeros modelos fueron construidos en los años 50, la escasez de tecnología apropiada en dichos años hizo que se alcanzaran menos logros de los esperados. El desarrollo de la microinformática producido a partir de los 70, renovó el interés por esta área y a partir de entonces se multiplicaron vertiginosamente sus aplicaciones, demostrando su valía en una gran variedad de campos. En particular, las RNA son especialmente útiles en problemas en los que se dispone de información confusa o incompleta, lo que hace que no puedan usarse técnicas de razonamiento deductivo basadas en la utilización de reglas. Las RNA automáticamente construyen asociaciones basadas en los resultados de situaciones conocidas (aprendizaje), ante una nueva situación, el sistema neuronal automáticamente se ajusta a sí mismo.

Uno de los primeros intentos de crear modelos matemáticos que imitasen el funcionamiento cerebral es debido al equipo formado por McCulloch, del Colegio de Medicina de la Universidad de Illinois, y Pitts, matemático del M.I.T. En 1943 estos dos investigadores presentaron un trabajo, McCulloch, (1943), con un modelo matemático bastante simplificado de neuronas formales, probando que toda expresión lógica finita podía ser construida mediante una combinación de esas neuronas formales. Aunque los actuales modelos de redes neuronales artificiales difieren bastante de esos simplificados modelos iniciales, los trabajos de McCulloch y Pitts siguen siendo una referencia obligada.

2.1. FUNDAMENTOS BIOLÓGICOS

2.1.1. LAS NEURONAS

El científico español Santiago Ramón y Cajal (1852-1934), premio Nobel de Medicina en 1906, descubrió hace un siglo la estructura del cerebro humano. Advirtió que el cerebro está formado por unas 100.000 millones de unidades procesadoras: *las neuronas.* Aunque



Santiago Ramón y Cajal (1852-1934)

se ha avanzado mucho, desde entonces, en la investigación sobre el funcionamiento del cerebro, aún quedan muchos aspectos por descubrir, por lo que sigue siendo un tema de plena actualidad científica.

Las neuronas, cuyo esquema se ilustra en la Figura 3, son capaces de recibir y enviar señales a sus vecinas mediante pulsos eléctricos. Una neurona tiene tres partes principales:

1. Cuerpo celular, con su citoplasma y núcleo correspondiente.
2. Dendritas, de una longitud de unas 10 micras, actúan como cables receptores de señales electroquímicas.
3. Axón o fibra nerviosa, es el conducto de salida de la señal, es mucho más largo que las dendritas. En su extremo final dispone de unas pequeñas estructuras, llamadas sinapsis, que comunican con las dendritas de otras neuronas. Generalmente una neurona tiene unas 10.000 uniones sinápticas con otras neuronas.

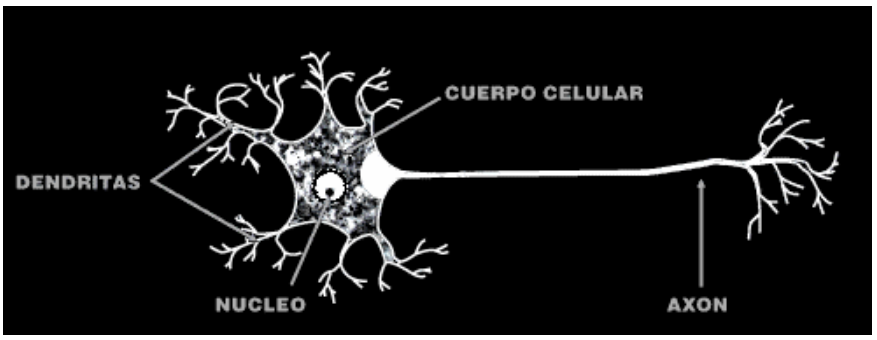


Figura 3: Esquema de una neurona

La neurona procesa las corrientes eléctricas que llegan a sus dendritas y por medio del axón transmite las corrientes eléctricas resultantes a otras neuronas conectadas a ella. El proceso de emisión es controlado por el potencial interno asociado a la neurona. Como resultado de todas las señales entrantes, se obtiene una excitación, si el potencial asociado supera un cierto umbral, se envía un impulso eléctrico al axón; en caso contrario, no se envía ninguna señal.

2.1.2. CONEXIONES ENTRE NEURONAS: GENÉTICA Y APRENDIZAJE

Los seres humanos nacen con un conjunto completo de neuronas, pero las conexiones entre ellas, sinapsis, se crean con el proceso de aprendizaje: cada vez que se aprende algo se crean nuevas conexiones.

Por tanto, la inteligencia viene determinada por el número de conexiones sinápticas, más que por el número de neuronas vivas.

Las fuerzas de los vínculos sinápticos entre las neuronas aumentan con los estímulos recibidos, de manera que la falta prolongada de estímulo entre dos neuronas conectadas hace que su conexión se debilite. Esta es la base del principio propuesto por Donald Hebb (1949).

Es importante notar que aunque el tiempo de conmutación de la neurona (*unos pocos milisegundos*) es casi un millón de veces menor que en los actuales elementos de las computadoras, ellas tienen una conectividad miles de veces superior que las actuales supercomputadoras.

2.2. ESTRUCTURA DE LAS REDES NEURONALES ARTIFICIALES

2.2.1. UNIDADES PROCESADORAS (NEURONAS)

Los modelos computacionales conocidos como Redes Neuronales Artificiales están inspirados en las características neurofisiológicas de las neuronas del cerebro humano. Una red neuronal artificial está constituida por un gran número de pequeñas unidades procesadoras (neuronas) interconectadas; cada una de las unidades procesadoras realiza cálculos simples a partir de la información recibida de las unidades vecinas. Cada entrada de una neurona lleva asociado un peso, siendo el objetivo del proceso de aprendizaje el ajuste de dichos pesos para reproducir un conjunto de patrones representativo del problema a abordar.

Una unidad procesadora o neurona está constituida por:

1. Un conjunto de nodos de entrada (inputs): $x_1, x_2,..., x_n$
2. Un nodo de salida (output): y
3. Una función neuronal, o *función de activación*, que calcula un valor de salida basado en una combinación lineal de los valores de entrada:

$$y = f\left(\sum_{i=1}^{n} w_i x_i\right)$$

Los coeficientes $w_1, w_2,..., w_n$ que intervienen en la combinación lineal de los inputs se denominan pesos. Estos pesos pueden ser tanto valores positivos como negativos, reproduciendo de esta manera el

carácter excitador o inhibitorio, respectivamente, de la correspondiente entrada.

La combinación lineal de los valores de entrada con los pesos correspondientes es lo que se conoce como *actividad lineal de la neurona*:

$$Actividad\ lineal = \sum_{i=1}^{n} w_i x_i$$

La neurona recoge todos los valores de entrada sumando todas las influencias excitadoras e inhibidoras, obteniendo de esta manera su actividad lineal, que viene a representar la *excitación* total recibida. En función del valor de esa excitación la neurona devuelve un valor de salida (*valor de activación*). Cuando la neurona esta conectada con otras neuronas, ese valor de activación se convierte en valor de entrada para las neuronas conectadas.

La Figura 4 pretende mostrar el esquema de una neurona artificial.

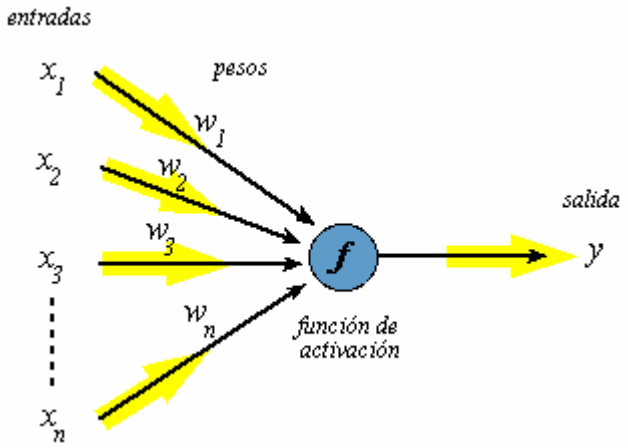


Figura 4.: Esquema de una unidad procesadora o neurona artificial.

2.2.2. FUNCIONES DE ACTIVACIÓN

De acuerdo al modelo biológico, si la excitación total de una neurona supera un cierto umbral, entonces la neurona envía un impulso eléctrico a través de su axón. La función de activación es, por tanto, la que devuelve la magnitud del impulso devuelto.

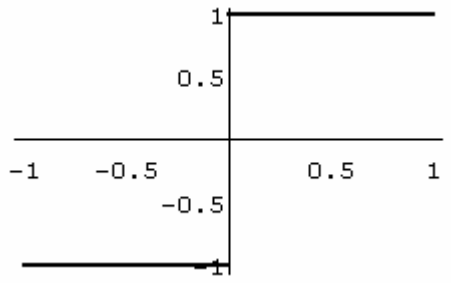
En los primeros años de las RNA, se pensaba que las neuronas cerebrales eran binarias, de forma que los primeros modelos de RNA

creados se basaban en funciones de activación binarias, que producían una salida binaria a partir de una combinación lineal de un conjunto de entradas (inputs) también binarios. El valor devuelto por estas funciones dependía de si la actividad lineal se encontraba por encima o por debajo de un cierto valor umbral.

Dos de las funciones de activación binarias más utilizadas son la función signo y la función escalón estándar.

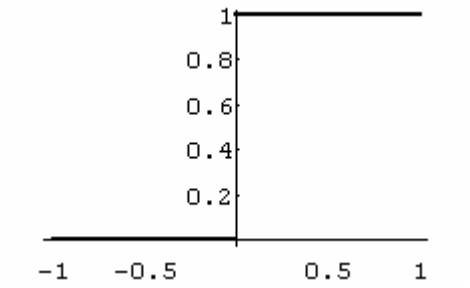
Función signo:

$$\operatorname{sgn}(x) = \begin{cases} -1 & si\ x < 0 \\ 1 & en\ otro\ caso \end{cases}$$



Función escalón estándar:

$$\Theta(x) = \begin{cases} 0 & si\ x < 0 \\ 1 & en\ otro\ caso \end{cases}$$



En los dos casos anteriores el valor umbral estaría situado en $x=0$. Cuando se desea utilizar otro valor umbral se suele recurrir a un pequeño artificio, consistente en añadir una entrada ficticia a la neurona con valor de entrada –1 y peso asociado justamente el valor umbral. De esta manera la actividad lineal de la neurona sería:

$$Actividad\ lineal\ con\ valor\ umbral = \sum_{i=1}^{n} w_i x_i - u = \sum_{i=0}^{n} w_i x_i$$

siendo $x_0 = -1$ y $w_0 = u$.

En ocasiones se suele utilizar también como función de activación la función identidad: $f(x)=x$, es decir, el valor devuelto por la neurona es directamente la actividad lineal de la misma.

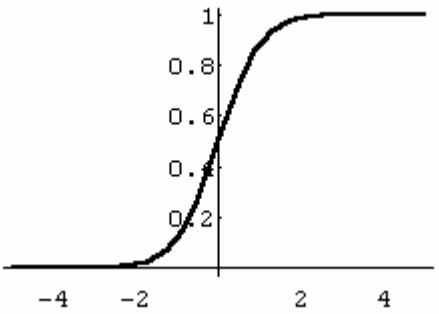
Diversos estudios sobre los procesos físicos de excitación en las neuronas cerebrales (véase por ejemplo Adrian 1946), parecen indicar que las funciones que determinan los potenciales de los impulsos generados por la neurona a partir de la excitación total recibida no son de tipo binario, sino de tipo *sigmoidal*.

Las funciones sigmoidales son funciones monótonas acotadas y no lineales. En los modelos de RNA las funciones de activación

sigmoidales más utilizadas son las funciones logísticas y las funciones tangente hiperbólica, cuyas respectivas definiciones y representaciones gráficas pueden verse a continuación.

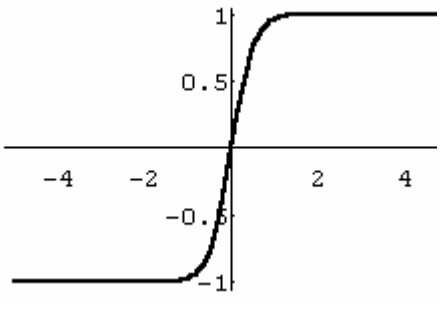
Función logística:

$$f_c(x) = \frac{1}{1+e^{-cx}}$$



Función tangente hiperbólica:

$$g_c(x) = Tanh(cx)$$



(representaciones gráficas en el caso c=2)

Como puede observarse, a diferencia de las funciones de activación con salida binaria (funciones signo o escalón), estas funciones sigmoidales tienen una salida continua.

2.2.3. TOPOLOGÍAS DE REDES

Al igual que ocurre en el cerebro humano, las neuronas se conectan entre sí, de manera que el valor de salida de una neurona se convierte en valor de entrada de otra. Establecer la topología de la red es uno de los problemas cruciales de las RNA. Muchas veces se opta por probar diferentes estructuras hasta encontrar una que produzca buenos resultados para el problema a tratar. Esta es justamente una de las características que más críticas recibe por parte de los detractores de los modelos de RNAs.

Establecer la topología de la red significa:

- Determinar el número de neuronas que la componen.
- Determinar el número y el tipo de conexiones entre ellas.

Normalmente las neuronas se suelen organizar en capas, distinguiendo tres tipos de capas:

1. *Capa de entrada*, formada por las neuronas cuyas entradas no están conectadas a otras neuronas.
2. *Capas intermedias,* formadas por neuronas que reciben entradas procedentes de otras neuronas y cuyas salidas se convierten en entradas para otras.
3. *Capa de salida*, formada por las neuronas cuyas salidas no son entradas de ninguna otra neurona.

También es importante establecer el tipo de las conexiones entre las neuronas. Existen varios tipos de uniones, las más importantes son:

- *Conexiones hacia delante*: conectan las neuronas de una capa con las de la capa siguiente. Este proceso puede entenderse como una composición funcional de las funciones de activación de ambas neuronas.

- *Conexiones laterales*: conectan neuronas de una misma capa. Este tipo de conexiones se utiliza en las denominadas *capas competitivas*, donde cada neurona se conecta a sí misma mediante un peso positivo (excitante) y a las restantes neuronas de la capa mediante pesos negativos (inhibitorios).

- *Conexiones hacia atrás o recurrentes*: las neuronas se conectan con las de las capas anteriores. Este tipo de conexiones se suelen utilizar para tratar modelos dinámicos y temporales.

Las diferentes conexiones entre las neuronas de una red pueden representarse mediante una matriz de pesos: $W=(w_{ij})$, donde w_{ij} representa el peso de la conexión entre la neurona i-ésima y la j-ésima.

2.3. APRENDIZAJE

La estructura de una RNA esta determinada tanto por su topología como por su algoritmo de aprendizaje.

Los algoritmos de aprendizaje también imitan el proceso de aprendizaje del cerebro humano relacionado con el cambio de estado de las conexiones entre las neuronas. Las RNA aprenden a partir de unos datos (experiencia), ajustando los pesos de sus conexiones para codificar esos *datos de entrenamiento*. Cuando la red detecta un error, algunos pesos deben ser modificados para compensarlo, las reglas que gobiernan

cuándo y cómo estos cambios deben producirse constituyen el denominado *algoritmo de aprendizaje.*

Una red neuronal, por tanto, no se programa en la forma tradicional (traduciendo algoritmos en secuencias de órdenes y operaciones), sino que se ajusta progresivamente en función del uso, a modo de proceso de aprendizaje. Una vez que la red ha sido entrenada, puede ser usada para responder a nuevas condiciones en el problema.

Evidentemente, existen diversos algoritmos de aprendizaje, dependiendo de la topología de la red correspondiente, pero estos algoritmos se pueden clasificar en dos grandes categorías:

- *Algoritmos de aprendizaje supervisado*: son aquellos que utilizan unos datos de entrenamiento formados por vectores de datos de entrada junto con sus respectivas salidas: $\{(\boldsymbol{x_k}, \boldsymbol{y_k}), k=1,2,...,m\}$, donde $\boldsymbol{y_k}$ está formado por los valores esperados para las neuronas de la capa de salida cuando se utilicen como valores de entrada los dados en $\boldsymbol{x_k}$. En este caso, el ajuste de los pesos de la red suele hacerse minimizando alguna función de error que mida las diferencias entre los valores esperados y los valores obtenidos por la red. Algunas medidas estándar del error son:

  Suma de los cuadrados de los errores: $\sum_{k=1}^{m} \left\| \mathbf{y}_k - \tilde{\mathbf{y}}_k \right\|^2$

  Raíz cuadrada del error cuadrático medio: $\sqrt{\left(\sum_{k=1}^{m} \left\| \mathbf{y}_k - \tilde{\mathbf{y}}_k \right\|^2 \right) \Big/ m}$

  Error máximo: $max_{\,k=1,\ldots,m} \left\| \mathbf{y}_k - \tilde{\mathbf{y}}_k \right\|$

- *Algoritmos de aprendizaje no supervisado*: en este caso los datos de entrenamiento se presentan sin ningún tipo de información adicional, por ejemplo, no hay información sobre qué valores de salida corresponden a qué valores de entrada. La red tiene que descubrir por sí misma patrones o categorías. Este tipo de aprendizaje se encuadra dentro de las técnicas autoorganizativas, o técnicas automáticas para descubrir la estructura de los datos. Algunos algoritmos conocidos de aprendizaje no supervisado son los de aprendizaje Hebbiano, o aprendizaje competitivo.

2.4. VALIDACIÓN

Terminado el proceso de aprendizaje de la red, con la correspondiente determinación de los pesos de las conexiones, es conveniente efectuar una validación cruzada para comprobar la calidad del modelo resultante. Para ello se suelen dividir los datos disponibles en dos grupos: unos utilizados para el entrenamiento de la red y otros para la comprobación del modelo. Cuando el error de comprobación es mucho mayor que el de entrenamiento es cuando se produce el conocido como sobreajuste, lo cual es un claro indicio de una falta de calidad en el modelo.

2.5. ALGUNOS MODELOS DE REDES NEURONALES

En esta sección se muestran algunos de los modelos más conocidos de RNA, comenzando por el más simple: el perceptrón.

2.5.1. EL PERCEPTRÓN, MODELO DE APRENDIZAJE SUPERVISADO

El perceptrón es una red neuronal formada por una capa de entrada a través de la cual se reciben unos valores de entrada $\{x_1,x_2,...,x_n\}$ y una capa de salida en la que se devuelven unos valores de salida $\{y_1,y_2,...,y_m\}$. La Figura 5 muestra el esquema de un perceptrón *4:3* (perceptrón con *4* entradas y *3* salidas). En el caso $m=1$ se estaría ante una red formada por una única unidad neuronal. Una de las aplicaciones más típicas de este tipo de redes son los problemas de clasificación lineal, consistentes en reconocer clases o categorías a partir de los datos de entrenamiento.

Los valores de salida son calculados mediante la expresión

$$y_i = f\left(\sum_{j=1}^{n} w_{ij} x_j\right) \qquad i = 1,2,\ldots,m$$

donde $f(x)$ es la función de activación utilizada y w_{ij} el peso asociado a la conexión de la entrada i-ésima con la neurona j-ésima.

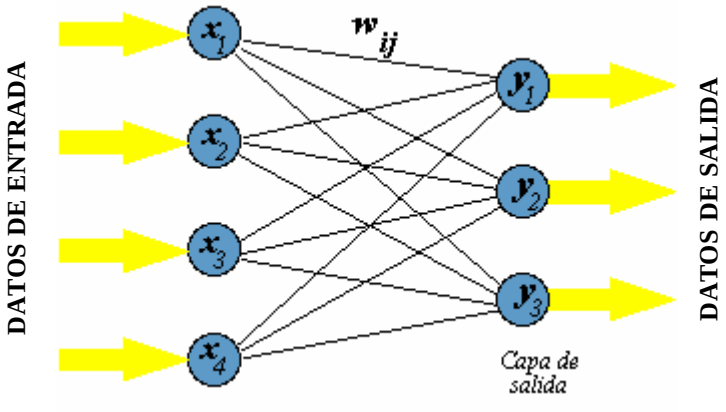


Figura 5: Estructura de un perceptrón 4:3.

El perceptrón utilizado un modelo de aprendizaje supervisado, lo que significa que para su entrenamiento se necesita una muestra de patrones de entrada con sus correspondientes salidas:

$$\textit{Muestra entrenamiento} = \left\{(a_{p1},\ldots,a_{pn};b_{p1},\ldots,b_{pm}) \,\middle|\, p=1,\ldots,q\right\}$$

donde b_{pi} es el valor de salida esperado para la unidad neural i-ésima cuando como valores de entrada se introducen $(a_{p1},...,a_{pn})$. Mientras que el valor real obtenido por la red se calcula mediante la expresión:

$$\tilde{b}_{pi} = f\left(\sum_{j=1}^{n} w_{ij}a_{pj}\right)$$

Dados unos pesos iniciales elegidos aleatoriamente $w_{ij}^{(0)}$ el aprendizaje supervisado consiste en, utilizando los datos de entrenamiento, ir sucesivamente ajustando dichos pesos de acuerdo a una expresión de la forma:

$$w_{ij}^{(k+1)} = w_{ij}^{(k)} + \Delta w_{ij}^{(k)}$$

El incremento que se aplica sobre los pesos en cada uno de los pasos puede ser calculado de diversas maneras, dependiendo del algoritmo de aprendizaje que se utilice.

Por ejemplo, en el caso del conocido como *aprendizaje Hebbiano* se toma

$$\Delta w_{ij}^{(k)} = -h\left(b_{ki} - \tilde{b}_{ki}\right)a_{kj}$$

donde η es un parámetro conocido como *índice de aprendizaje.* Como se desprende de la fórmula anterior, cuanto más grande sea la diferencia entre el valor esperado y el valor realmente obtenido por la red, mayor es el incremento que sufre el correspondiente peso. Cuando ambos valores, el esperado y el obtenido, coinciden no se produce ninguna modificación en los pesos.

Existen otras estrategias para la modificación de los pesos, por ejemplo, otra de las más utilizadas es la conocida como *regla delta*, basada en el método de minimización de descenso gradiente. En este caso se tomaría:

$$\Delta w_{ij}^{(k)} = -h \sum_k \left(b_{ki} - \tilde{b}_{ki}\right) \frac{\partial \tilde{b}_{ki}}{\partial w_{ij}} a_{kj}$$

Cuando se considera como función de activación $f(x)=x$ ambas expresiones, la dada en el aprendizaje Hebbiano y la de la regla delta, coinciden.

2.5.2. EL PERCEPTRÓN MULTICAPA

El perceptrón multicapa se diferencia del perceptrón en la presencia de capas intermedias de neuronas (capas ocultas). La adición de estas capas ocultas proporciona una mayor flexibilidad para resolver problemas en los que los perceptrones simples han demostrado no ser apropiados. Diversos estudios teóricos sobre este tipo de redes han demostrado que todo conjunto de funciones con *n* variables puede ser aproximado con un perceptrón con dos capas ocultas. Además en el caso de funciones continuas con una única capa oculta bastaría.

La estructura de perceptrón multicapa más utilizada es la formada por una capa de entrada, una capa oculta y una capa de salida, tal como se puede ver en la Figura 6.

El método de aprendizaje más popular para los perceptrones multicapa, el conocido como retro-propagación (back-propagation), está basado en la minimización del error cuadrático total usando el método de descenso gradiente.

Al intervenir en el proceso de actualización de pesos los pesos de las neuronas de las capas ocultas, el algoritmo no es tan simple como en el caso de los perceptrones simples.

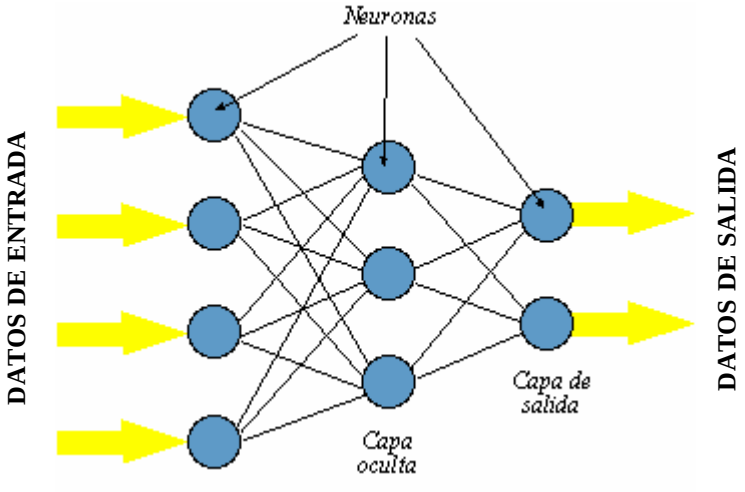


Figura 6: Estructura de un Perceptrón 4:3:2 (4 neuronas en la capa de entrada, 3 en la oculta y 2 en la de salida).

El proceso de retro-propagación de una forma muy simplificada podría resumirse así: se parte de un conjunto de entrenamiento dado, de la misma manera que en el caso anterior, por un conjunto de valores de entrada con sus correspondientes salidas. Se propagan hacia delante bs valores de entrada para obtener unos valores para las capas ocultas y unos valores de salida. Con los valores obtenidos se calculan los errores cometidos y se ajustan los pesos de la capa de salida, para más tarde, propagando hacia atrás los errores obtenidos, ajustar los pesos de las capas ocultas.

Todo perceptrón multicapa se caracteriza por las ecuaciones dinámicas siguientes:

$$u_i(l) = \sum_{j=1}^{N_l - 1} w_{ij}(l) a_j(l-1) + q_i(l)$$

$$a_i(l) = f(u_i(l)) \qquad i = 1, \ldots, N_l; \quad l = 1, \ldots, L$$

donde L representa el número de capas de la red, $a_i(0)$ los valores de las entradas y $a_i(L)$ los valores de salida. Utilizando la regla de derivación de la cadena, la expresión que permite ajustar los pesos de la red en el proceso de aprendizaje es:

$$\Delta w_{ij}^{(m)}(l) = h\, d_i^{(m)}(l) f'(u_I^{(m)}(l))\, a_j^{(m)}(l-1)$$

siendo $d_i^{(m)}(l) = -\frac{\partial\, ErrorTotal}{\partial\, a_i^{(m)}(l)}$

2.5.3. MODELOS DE REDES CON APRENDIZAJE NO SUPERVISADO

Entre los modelos de redes neuronales con aprendizaje no supervisado se podrían citar la red de Hopfield (Hopfield-1982) o la red de Kohonen. Ambos modelos son ejemplos de redes con capas competitivas.

La red de Hopfield (ver Figura 7) es una arquitectura formada por una sola capa de neuronas con conexiones laterales entre sí. Se utiliza principalmente como memoria autoasociativa, para almacenar y recuperar información. El aprendizaje de la red consiste en ajustar los pesos para que dado un patrón de entrenamiento la red sea capaz de devolver el mismo patrón. Esta red ha sido usada, por ejemplo, para la resolución de problemas de reconocimiento de caracteres.

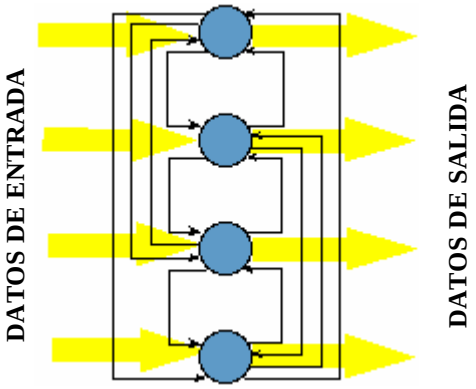


Figura 7: Red neuronal de Hopfield con cuatro neuronas.

2.6. HARDWARE BASADO EN REDES NEURONALES

Aunque la mayor parte de las experiencias y desarrollos realizados hasta ahora en el campo de las RNA han sido mediante simulaciones en ordenadores digitales tradicionales, deben destacarse también los intentos de aplicar estos modelos a la construcción de

hardware. Así por ejemplo, la compañía Fujitsu fabricó en 1988 el primer chip neuronal constituido por 32 neuronas con 1024 conexiones. Prototipos similares han sido desarrollados por la Universidad de California (San Diego), los laboratorios Bell, estos últimos desarrollaron un chip experimental con 256 neuronas y hasta 32.000 uniones sinápticas.

Las diferencias entre los ordenadores tradicionales, basados en la tecnología digital, y ordenadores basados en redes neuronales, se ilustran en la Tabla 1.

Ordenadores digitales	**Ordenadores basados en redes neuronales**
⇨ Razonamiento deductivo: se aplican reglas conocidas para producir un output a partir de unos inputs.	⇨ Razonamiento inductivo: dados unos inputs con sus outputs (datos de entrenamiento) se construyen las reglas.
⇨ Computación centralizada, síncrona y serial.	⇨ Computación colectiva, asíncrona y paralela.
⇨ Intolerancia a fallos: si un transistor falla, el equipo completo falla.	⇨ Tolerante a fallos.
⇨ Rápidos	⇨ Más lentos.
⇨ Exactos.	⇨ Inexactos.
⇨ Conectividad estática	⇨ Conectividad dinámica.
⇨ Utilizables en casos con inputs precisos y reglas a aplicar bien conocidas.	⇨ Utilizables aún con ausencia de reglas bien definidas y utilización de datos parciales.

Tabla 1: Diferencias entre ordenadores digitales y ordenadores basados en redes neuronales.

2.7. CAMPOS DE APLICACIÓN DE LAS REDES NEURONALES

Un rápido repaso a algunas de las publicaciones científicas más destacadas en el campo de la Inteligencia Artificial, puede servir para analizar los campos actuales de aplicación de las redes neuronales artificiales.

Es una pagina que te da referencias sobre la tecnología empleada en el ViaVoice, aunque este es un producto de un proyecto de investigación que desarrolla IBM.

http://www.research.ibm.com/hlt/html/overview.html

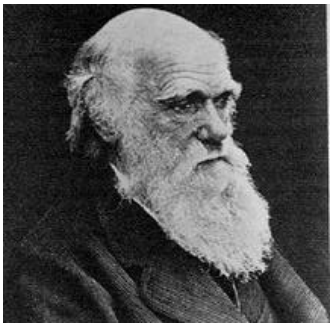
2.8. ALGUNAS EXTENSIONES DE LAS REDES NEURONALES

Como ha quedado de manifiesto en la sección anterior, las RNA han demostrado ser una herramienta muy útil en una gran variedad de campos. Sin embargo, este tipo de modelos tiene algunas limitaciones, razón por la que han ido surgiendo diversas modificaciones o extensiones, como pueden ser las redes de alto orden (Lee et al 1986), las redes neuronales probabilísticas (Specht 1990), las redes neuronales difusas (Gupta y Rao 1994), las redes basadas en “wavelets” (Zhang 1992),...

Los detractores de los modelos de redes neuronales argumentan que se trata de verdaderas “cajas negras” en las que la elección de la topología de red (neuronas, capas, conexiones,...) no viene en la mayoría de los casos justificada por las características del modelo a tratar. Otro de los intentos de generalización de las redes neuronales, lo constituyen las redes funcionales (Castillo et al 1999), cuyo objetivo es aprovechar las características funcionales y propiedades del proceso para la elección de la topología. Estos modelos se basan en una importante rama de las matemáticas: las ecuaciones funcionales.

3. ALGORITMOS GENÉTICOS

3.1. TEORÍA DE LA EVOLUCIÓN Y DE LA SELECCIÓN NATURAL



Charles Darwin (1809-1882)

En 1859 Charles Darwin publicó su obra más conocida: *"Origen de las especies*"; en ella aportó una serie de pruebas tendentes a demostrar la evolución orgánica, explicando además el mecanismo de dicha evolución: la *selección natural*.

Actualmente la teoría de la evolución se concibe como lo hiciera Darwin, como el "*conjunto de efectos de muchas pequeñas variaciones acumuladas durante un número casi infinito de generaciones*". La evolución implica dos factores:

- *Variabilidad génica*, tiene su origen en la mutación que se introduce en las poblaciones mediante recombinación sexual o parasexual.
- *Selección natural*, que modifica la variabilidad de las poblaciones.

3.2. MODELOS MATEMÁTICOS INSPIRADOS EN LA TEORÍA DE LA EVOLUCIÓN

Otra de las herramientas matemáticas que han sido exitosamente utilizadas por investigadores en I.A. son los conocidos como *algoritmos genéticos*. Detrás de este nombre tan llamativo, se esconden procedimientos de búsqueda basados en los principios de la evolución y la selección natural. Estos algoritmos pueden ser utilizados en problemas de optimización donde la búsqueda de soluciones óptimas se lleva a cabo en un espacio de soluciones codificadas mediante cadenas de longitud finita. Esta codificación es lo que se conoce como "representación genética" de cada posible solución.

Los orígenes de los algoritmos genéticos pueden situarse a comienzos de los años 50, con los trabajos sobre simulaciones por ordenador de sistemas biológicos realizados por diferentes biólogos. Sin embargo, el concepto de algoritmo genético tal como se conoce hoy en día es debido a los trabajos de John Holland de la Universidad de

Michigan a comienzos de los 70 (Holland 1975). A partir de ahí el interés por este tipo de algoritmos creció rápidamente.

Los algoritmos genéticos poseen un mecanismo de cálculo muy simple, sin necesitar ninguna condición de regularidad sobre la función objetivo. La principal característica de este tipo de algoritmos es que manejan poblaciones de soluciones potenciales, es decir que el algoritmo mantiene una población de individuos $P(t)=\{x_1(t),...,x_n(t)\}$ para cada iteración *t*, donde cada individuo representa una solución potencial al problema. En cada iteración se obtiene una nueva población mediante un proceso de selección (reproducción) basado en una medida del valor de la función objetivo de los distintos individuos y en una serie de operadores "genéticos" (cruce y de mutación). En consecuencia, los tres operadores básicos que componen todo algoritmo genético son:

1. *Reproducción*: proceso por el cual los individuos con mayor valor de la función objetivo, o función de ajuste, tienen mayor probabilidad de estar en la generación siguiente, mientras que los individuos de menor valor tienden a desaparecer. Constituye por tanto una simulación del proceso de selección natural.

2. *Cruce*: proceso por el cual a partir de dos o más individuos de la población seleccionados aleatoriamente, se obtienen nuevos descendientes que tratan de mejorar la bondad media de la población.

3. *Mutación*: proceso por el cual, con una probabilidad muy baja, un individuo de la población sufre una alteración puntual.

A pesar de su aparente simplicidad, los algoritmos genéticos han sido utilizados exitosamente para la resolución de una enorme variedad de problemas en diversos campos: aplicaciones industriales, gestión empresarial, finanzas, logística, investigación de operaciones,...

Las principales diferencias de los algoritmos genéticos con respecto a otros procedimientos de optimización son:

- Trabajan con la codificación de los parámetros.
- Buscan conjuntos de soluciones, no soluciones individuales.
- Usan reglas de transición probabilísticas, no deterministas.
- No utilizan ninguna característica especial de la función a optimizar.
- Al devolver un conjunto de posibles soluciones, facilitan al decisor seleccionar la solución más apropiada de acuerdo con criterios no estrictamente cuantitativos.

- Desde un punto de vista computacional, los algoritmos genéticos son especialmente adecuados para una implementación paralela, aumentando de esta forma su velocidad y rendimiento.

3.3. IMPLEMENTACIÓN DE UN ALGORITMO GENÉTICO

Todo algoritmo genético se caracteriza por disponer de los siguientes elementos básicos:

1. Una representación "genética" para cada una de las potenciales soluciones.
2. Una población inicial de potenciales soluciones.
3. Una función de ajuste (fitness).
4. Operadores genéticos que modifican la composición de las nuevas soluciones.
5. Varios parámetros utilizados por el algoritmo (tamaño de la población, probabilidades con las que se aplican los operadores genéticos, etc.).

3.4. REPRESENTACIÓN DE INDIVIDUOS

Cada solución potencial ("individuo") en un algoritmo genético ha de ser representada por una cadena de longitud finita, es decir una secuencia finita de caracteres en un alfabeto también finito. El alfabeto más comúnmente utilizado es el binario {0,1}, de acuerdo con el cual cada individuo es representado por una secuencia de 0's y 1's.

Por ejemplo, todo número entero entre 0 y 255 puede ser representado mediante una secuencia de 8 dígitos binarios.

$77 =$ | **0** | **1** | **0** | **0** | **1** | **1** | **0** | **1** | $= 0*2^7+1*2^6+0*2^5+0*2^4+1*2^3+1*2^2+0*2^1+1*2^0$

Esta representación mediante un alfabeto finito es lo que se conoce como "representación genética del individuo".

La representación binaria no es la única representación genética posible, como se verá en un ejemplo posteriormente en algunos casos resultan más apropiadas otro tipo de representaciones.

3.5. POBLACIÓN INICIAL

Para empezar a operar con un algoritmo genético se necesita tomar como partida una población inicial de soluciones potenciales. Muchas veces esta población inicial es elegida de forma totalmente aleatoria. En otras ocasiones, un conocimiento previo de alguna característica del problema puede sugerir la elección de determinados individuos para esa población. También el posible conocimiento de soluciones cercanas a las óptimas puede servir para incluirlas en la población inicial y mejorar de esta forma el rendimiento del algoritmo.

El tamaño de esta población inicial es uno de los aspectos más importantes a tener en cuenta y puede llegar a ser crucial en numerosas aplicaciones. Si la población inicial es muy reducida el algoritmo puede converger demasiado rápidamente sin llegar a obtener buenas soluciones, mientras que si es demasiado grande se pueden desperdiciar recursos computacionales. El tamaño de la población puede ser constante, es decir el mismo para cada generación, o bien se puede utilizar un algoritmo con tamaño de la población variable. Un estudio sobre el tamaño óptimo de la población puede encontrarse en Goldberg (1985).

3.6. FUNCIÓN DE AJUSTE

La bondad de toda solución potencial del problema debe ser medida mediante una función *F(P)* conocida como función de ajuste (*fitness*). La probabilidad de supervivencia de un individuo para las generaciones siguientes dependerá del valor de la función de ajuste en dicho individuo, cuanto mayor sea este valor mayor será la probabilidad ("supervivencia del más fuerte").

La función de ajuste debe ser siempre positiva y el objetivo del problema será encontrar una población final de individuos en la que se maximice esta función. No se exige ninguna otra condición sobre esta función de ajuste, de manera que una de las principales características de los algoritmos genéticos es su flexibilidad para abordar problemas en los que las técnicas clásicas de optimización no son aplicables.

Con objeto de mejorar la efectividad del algoritmo se podrían utilizar técnicas de escalado sobre la función de ajuste. Algunas de estas técnicas se encuentran descritas en Michalewicz (1996).

3.7. OPERADORES GENÉTICOS

El proceso de obtención de las sucesivas generaciones de individuos (soluciones potenciales), requiere la implementación de los tres operadores genéticos básicos: reproducción, cruce y mutación.

3.7.1. REPRODUCCIÓN

La idea del operador de reproducción ya se ha comentado antes, se trata de un mecanismo de "selección natural" de los "individuos más fuertes" (con mejor valor para la función de ajuste), para pasar a formar parte de la siguiente generación de individuos.

El operador de reproducción es implementado normalmente simulando el funcionamiento de una ruleta trucada, en la que los individuos con mayor valor por la función objetivo tienen mayores probabilidades de salir "elegidos".

Por ejemplo, supóngase que se tiene una población inicial de potenciales soluciones formada por 5 individuos y con los valores de ajuste que se muestran en la tabla siguiente:

Individuo	*Valor de ajuste F(P)*	*% Total ajuste*
P1	100	17%
P2	150	25%
P3	45	8%
P4	220	37%
P5	85	14%

Tabla 2: Valores de ajuste de una población de 5 individuos.

La selección de la siguiente generación de 5 individuos se haría con 5 tiradas de la ruleta trucada de la Figura 8.

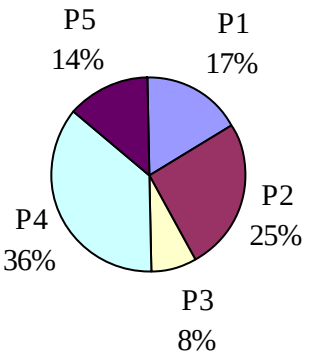


Figura 8: Ruleta de selección de individuos

Desde un punto de vista algorítmico, de cara a su implementación en un ordenador, el operador de reproducción se basa en generar un número aleatorio *x* entre 0 y 1 y elegir el índice del individuo a seleccionar como el menor índice k que verifique la condición:

$$\frac{\sum_{i=1}^{k} F(P_i)}{Total} > x$$

donde *Total* es la suma de los valores de ajuste de todos los individuos de la población.

Este proceso de selección es repetido *n* veces y todos los individuos seleccionados pasan a formar una nueva población sobre la que posteriormente se aplicarán los restantes operadores genéticos.

3.7.2. CRUCE

Una vez que se ha utilizado el operador de reproducción para seleccionar los individuos que "sobreviven", intervienen dos nuevos operadores que darán lugar a la aparición de nuevos individuos hasta ahora no existentes. El primero de tales operadores es el operador de cruce.

Un operador de cruce crea dos nuevos individuos (descendientes) combinando partes de otros dos individuos (progenitores) seleccionados de manera aleatoria en la población.

Uno de los operadores de cruce más simples consiste en establecer de forma aleatoria un punto de cruce en la representación genética de los dos individuos e intercambiar las partes obtenidas tal como se puede ver en la Figura 9.

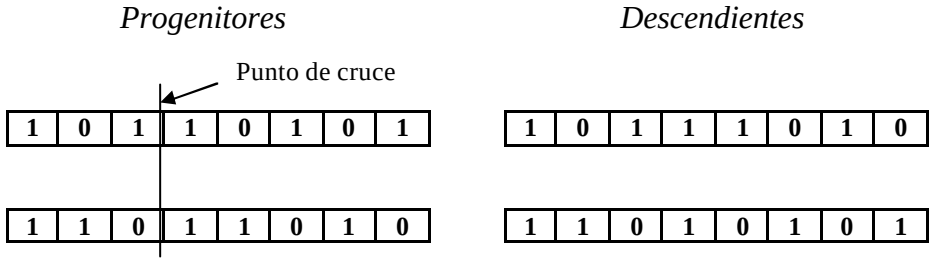


Figura 9: Operador de cruce sobre dos individuos con representación genética binaria.

3.7.3. MUTACIÓN

La mutación es una transformación en la que, con una probabilidad baja, un individuo sufre una pequeña modificación. Una mutación puede ser un simple cambio en una de las posiciones de su representación genética, por ejemplo cambiar un 1 por un 0.

3.8. PARÁMETROS DEL MODELO

Como puede observarse, los operadores que intervienen en todo algoritmo genético son bastante simples. En ocasiones puede llegar a sorprender que con esta simplicidad matemática realmente funcionen. El azar interviene en el diseño del algoritmo, pero no se puede achacar a la casualidad el que el algoritmo funcione.

También es importante una cuidadosa elección de algunos de los parámetros que intervienen en el algoritmo:

- El *tamaño de la población inicial y de las sucesivas poblaciones*: a mayor variedad de individuos en la población inicial mayores son las probabilidades de obtener buenas soluciones en las generaciones siguientes.
- El *número de generaciones*: puede ser fijado de antemano o establecido en función de la evolución del ajuste medio de los individuos de la población.

- La *probabilidad de cruce*: cuando los individuos son seleccionados con el operador de reproducción se efectúan cruces entre algunos de ellos. La probabilidad de que el cruce se efectúe debe ser establecida como parámetro en el modelo; normalmente una buena efectividad del algoritmo requiere de una probabilidad alta de cruce.
- La *probabilidad de mutación*: el operador de mutación desempeña un papel secundario en los algoritmos genéticos, pero esto no quiere decir que no sea importante. La probabilidad de que un individuo sufra una mutación es otro de los parámetros del modelo, normalmente dicha probabilidad debe ser baja, al igual que ocurre con la probabilidad de mutación en poblaciones naturales. Diversos estudios empíricos indican que para obtener buenos resultados la frecuencia de mutación debe de ser de una mutación por cada mil unidades de representación genética transferidas.

3.9. EJEMPLO DE APLICACIÓN: UN PROBLEMA DE DISTRIBUCIÓN EN PLANTA

La distribución en planta, entendiendo como tal *"el proceso de determinación de la mejor ordenación de los factores productivos disponibles, de modo que constituyan un sistema productivo capaz de alcanzar los objetivos fijados de la forma más eficiente posible"* (Domínguez Machuca et al., 1995) se configura como una de las decisiones de carácter estratégico dentro del área de operaciones de la empresa. Una buena distribución en planta puede contribuir a mejorar el grado de eficiencia del proceso productivo en cuanto permite minimizar el manejo de materiales, eliminar retrasos durante el proceso de fabricación, utilizar convenientemente el espacio disponible, disminuir la cantidad de inventarios en curso, etc.

Uno de los problemas típicos de distribución en planta es el conocido como distribución en planta orientada a procesos, consistente en decidir la disposición óptima de un conjunto de secciones o talleres en un conjunto de áreas dentro de la planta.

Por ejemplo, una empresa está plateándose la ubicación en su planta de 10 secciones diferentes: *S1,S2,...,S10*. Para ello ha dividido su planta en 10 áreas diferentes, tal como muestra la Figura 10.

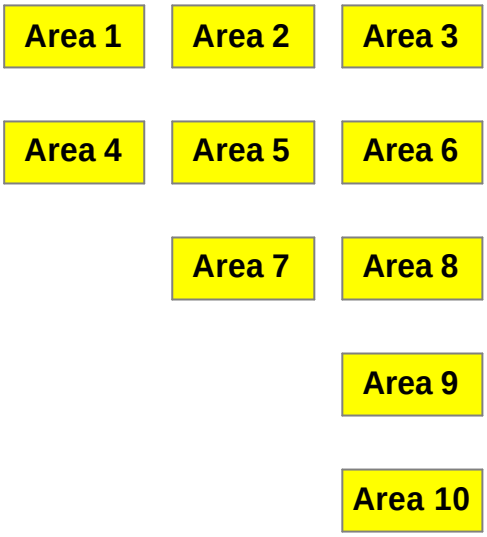


Figura 10: Disposición de las diferentes áreas en la planta de producción.

Determinadas secciones deberían estar situadas próximas debido al flujo de materiales entre ambas, suponer que estos flujos son los que aparecen en la Tabla 3.

	S1	*S2*	*S3*	*S4*	*S5*	*S6*	*S7*	*S8*	*S9*	*S10*
S1	-	10	15	12	-	-	-	21	-	5
S2	-	-	13	8	-	10	11	-	-	-
S3	20	-	-	5	10	-	-	15	8	6
S4	-	8	-	-	-	-	10	12	-	8
S5	10	12	-	-	-	8	6	4	-	-
S6	-	-	-	-	-	-	10	12	-	-
S7	-	-	-	-	-	12	-	8	-	-
S8	-	-	-	-	-	16	-	-	-	-
S9	12	6	4	-	-	-	-	-	-	-
S10	4	8	10	-	-	-	-	-	-	-

Tabla 3: Flujos entre secciones.

Además de los flujos de mercancías, han de tenerse en cuenta las distancias entre las áreas de la planta, ya que el coste del transporte de la mercancía será proporcional a la distancia que debe recorrer. Las distancias entre las áreas son las que muestra la Tabla 4.

	A1	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*	*A8*	*A9*	*A10*
A1	0	1	2	1	2	3	3	4	5	6
A2		0	1	2	1	2	2	3	4	5
A3			0	3	2	1	3	2	3	4
A4				0	1	2	2	3	4	5
A5					0	1	1	2	3	4
A6						0	2	1	2	3
A7							0	1	2	3
A8								0	1	2
A9									0	1
A10										0

Tabla 4: Distancias entre las áreas. Se asume que la matriz de distancias es simétrica.

Tal como queda planteado el problema, el número de posibles colocaciones de las secciones en la planta es:

$$10! = 3.628.800$$

una cifra considerable como para poder analizar todas esas posibilidades. El objetivo se plantea como la búsqueda de una distribución óptima en el sentido de minimizar el coste total del transporte de las mercancías entre las distintas secciones. Dicho coste total de transporte (CTT) viene dado por:

$$CTT = \sum_{i=1}^{10} \sum_{j=1}^{10} t_{ij}\, d_{ij}\, c_{ij}$$

donde t_{ij} es el número de unidades a mover entre las secciones *Si* y *Sj*, d_{ij} es la distancia existente desde el área *i* hasta el *j* y c_{ij} es el coste por unidad de distancia y carga desde *i* hacia *j*, en el ejemplo considerado se supondrá que ese coste es igual en todos los desplazamientos, es decir, c_{ij} constante.

Dado el número extraordinariamente elevado de posibles combinaciones existentes, que puede llegar a impedir obtener soluciones óptimas, para la resolución de este tipo de problemas se ha recurrido a diversas opciones, desde la utilización de algoritmos heurísticos, que al menos proporcionan soluciones factibles y satisfactorias, a técnicas informatizadas como CRAFT, SPACECRAFT, ALDEP o incluso al desarrollo de sistemas expertos, como FADES (Fisher, 1986). En este caso, se utilizará un simple algoritmo genético para mostrar su aplicabilidad a problemas concretos.

Las peculiaridades del problema a tratar hacen necesarios una serie de ajustes en la representación genética de las posibles soluciones y en los operadores genéticos a utilizar.

3.9.1 REPRESENTACIÓN DE SOLUCIONES POTENCIALES

Toda posible solución del problema de distribución en planta debe ser codificada mediante una cadena de longitud finita en un alfabeto también finito. En este caso el alfabeto binario no resulta apropiado ya que los operadores de cruce y mutación podrían producir soluciones no factibles.

La representación más adecuada es utilizar un vector de enteros, de manera que toda posible solución (individuo) puede representarse a través de una permutación $P=(S_1,S_2,...S_n)$ de $(1,2,...,n)$, donde $P_i=S_j$ indica que la sección S_j está localizada en el área A_i. Por ejemplo, el individuo $(4,5,6,1,2,8,9,10,3,7)$ corresponde a una distribución en la que las secciones cuarta, quinta y sexta están situadas en las tres primeras áreas, la secciones primera y segunda en las áreas cuarta y quita, etc.

3.9.2. FUNCIÓN OBJETIVO (FUNCIÓN DE AJUSTE)

En el problema considerado el objetivo se establece en términos de la minimización de la función de coste total de transporte, CTT. Sin embargo, todo algoritmo genético tiene por objetivo la maximización de una función de ajuste positiva.

En este caso, se podría definir la función de ajuste de la siguiente manera:

$$F(P)=C_{max} - CTT(P),$$

donde C_{max} puede definirse como el mayor valor encontrado hasta el momento en todas las evaluaciones de la función CTT, es decir,

$$C_{max}=max\{CTT(P)\}.$$

3.9.3. POBLACIÓN INICIAL

Como población inicial del problema se ha optado por generar un conjunto de $n=30$ soluciones potenciales de forma totalmente aleatoria. La Tabla 5 muestra las posibles distribuciones obtenidas junto con sus respectivos costes totales de transporte.

Distribución potencial	*Coste*	*Distribución potencial*	*Coste*
P_1=(1,10,2,6,8,4,3,7,9,5)	840	P_{16}=(8,6,10,5,4,1,3,7,9,2)	867
P_2=(5,6,10,7,1,9,4,8,3,2)	922	P_{17}=(3,10,9,1,2,5,4,8,7,6)	672
P_3=(8,2,7,5,9,1,4,10,3,6)	1002	P_{18}=(9,10,2,6,4,5,3,7,1,8)	898
P_4=(9,4,5,3,10,2,8,1,6,7)	749	P_{19}=(1,7,3,8,9,6,4,5,10,2)	903
P_5=(5,10,7,2,4,1,9,3,8,6)	768	P_{20}=(10,1,6,5,3,8,2,7,4,9)	710
P_6=(5,10,8,3,4,7,9,2,1,6)	957	P_{21}=(8,2,6,7,10,5,1,3,9,4)	854
P_7=(5,6,10,3,2,4,7,1,8,9)	812	P_{22}=(5,6,3,1,4,9,2,8,10,7)	920
P_8=(4,7,6,10,1,3,9,5,8,2)	880	P_{23}=(6,7,1,5,2,10,9,4,8,3)	902
P_9=(4,10,6,2,7,1,8,3,9,5)	784	P_{24}=(8,1,7,10,2,9,3,6,4,5)	905
P_{10}=(8,4,1,9,7,5,10,2,6,3)	978	P_{25}=(7,5,2,1,4,10,6,9,8,3)	964
P_{11}=(5,9,7,2,1,8,4,10,6,3)	919	P_{26}=(2,1,4,3,8,6,7,10,5,9)	844
P_{12}=(3,7,6,10,8,5,4,1,9,2)	875	P_{27}=(10,2,8,9,5,6,4,3,7,1)	926
P_{13}=(6,10,4,3,7,1,8,2,9,5)	852	P_{28}=(8,9,4,7,6,2,3,5,1,10)	784
P_{14}=(1,4,5,7,3,10,2,8,6,9)	853	P_{29}=(9,4,3,8,7,6,2,10,1,5)	858
P_{15}=(3,2,5,9,8,10,1,4,6,7)	802	P_{30}=(3,6,2,8,7,1,4,5,10,9)	829

Tabla 5: Población inicial generada de forma aleatoria.

Como puede observarse, la población inicial tiene un rango de costes asociados en el intervalo *[cmin,cmax]=[672,1002]*, siendo además el coste medio para la población de *860.98*, y una desviación típica de *77.67*.

3.9.4. OPERADORES DE CRUCE

El operador de cruce simple analizado anteriormente es claramente ineficaz en este problema, por cuanto conduce a individuos sin ningún sentido. Obsérvese el ejemplo ilustrado en la Figura 11. Ninguno de los dos descendientes obtenidos es admisible ya que en ambos casos existen secciones que quedarían sin ubicación física y por el contrario existen secciones con dos áreas asignadas.

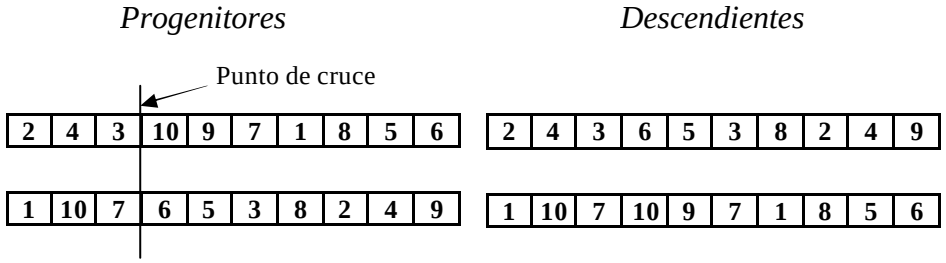


Figura 11: Ejemplo de inconsistencia del operador de cruce tradicional

Se hace necesario, por tanto, algún otro tipo de operador de cruce. Ejemplos de operadores de cruce utilizados en problemas con una estructura similar son:

- El operador de Emparejamiento Parcial (Partial Matched Crossover, PMX)
- El operador de Cruzamiento Ordenado (Order Crossover, OX)
- El operador de Cruzamiento Cíclico (Cycle Crossover, CX).

Algunos estudios comparativos sobre estos tres operadores pueden encontrarse en Oliver, Smith and Holland (1987).

El operador PMX (propuesto por Goldberg y Lingle), construye dos descendientes eligiendo al azar dos posiciones de cruce sobre el vector. Estos dos puntos definen una sección de emparejamiento que es usada para efectuar el cruce mediante operaciones de intercambio de elementos sobre los vectores progenitores. En la Figura 12 puede verse un ejemplo de aplicación del operador de cruce PMX.

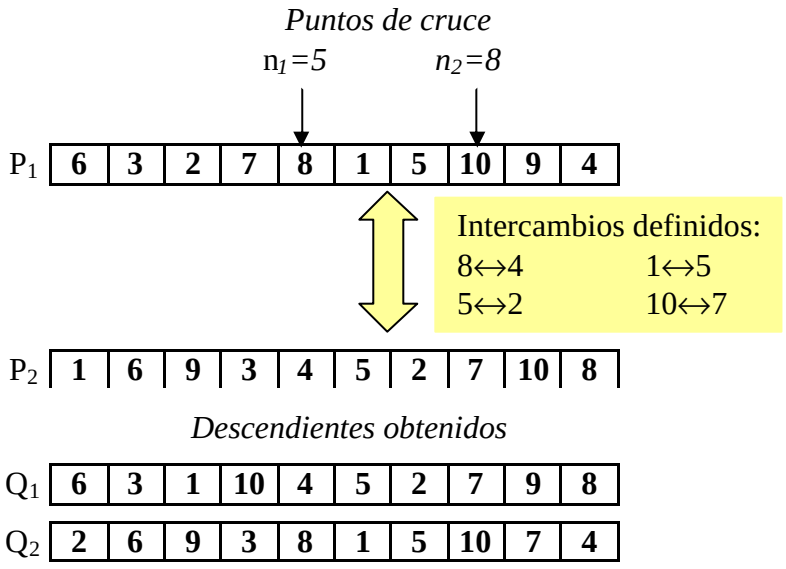


Figura 12: Ejemplo de cruce utilizando el operador PMX.

Como puede verse en la Figura 12, para obtener los dos descendientes, primero se intercambian los segmentos situados entre las posiciones de cruce obtenidas aleatoriamente. Este intercambio define además una serie de cambios puntuales (8↔4, 1↔5, 5↔2, y 10↔7) que

deben ser aplicados sobre P_1 y P_2. Como resultado de este proceso se obtienen dos nuevos vectores (descendientes): $Q_1=(6,3,5,10,4,5,2,7,9,8)$ y $Q_2=(2,6,9,3,8,1,5,10,7,4)$.

El operador OX (propuesto por Davis) comienza de una manera similar al PMX, es decir, eligiendo de forma aleatoria dos posiciones de cruce sobre los vectores y las correspondientes relaciones entre los elementos. A continuación el operador OX realiza un movimiento de desplazamiento para cubrir los huecos dejados por los elementos del vector a intercambiar; este movimiento de desplazamiento comienza a partir de la segunda posición de cruce. Posteriormente se producen los intercambios de elementos definidos por la sección de emparejamiento.

Por ejemplo, considerando las permutaciones P_1 y P_2 anteriores y los puntos de cruce $n_1=5$ y $n_2=8$, las nuevas permutaciones que se obtienen con el operador OX son: $Q_1=(3,8,1,10,4,5,2,7,9,6)$ y $Q_2=(3,4,2,7,8,1,5,10,6,9)$. El proceso de obtención de estos descendientes es ilustrado en la Figura 13.

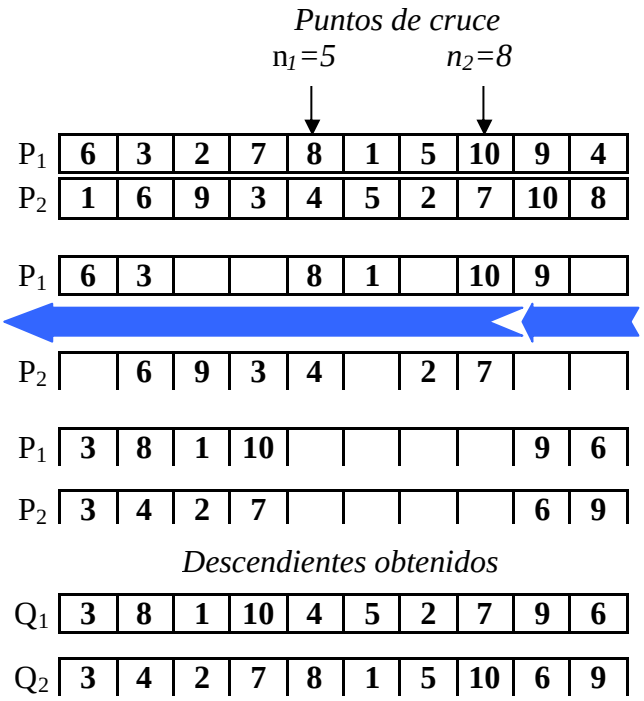


Figura 13: Ejemplo de cruce con el operador OX.

La principal diferencia entre PMX y OX es el hecho de que, mientras PMX tiende a respetar las posiciones absolutas de los elementos dentro de los vectores, OX tiende a respetar las posiciones relativas.

Finalmente, el operador CX (propuesto por Oliver) construye los descendientes de tal forma que cada elemento de los nuevos vectores (tanto su valor como su posición dentro del vector) viene de uno de sus vectores progenitores. Por ejemplo, considerando de nuevo P_1 y P_2, se toma el primer elemento del primer vector; como todo elemento debe ser tomado de uno de sus progenitores, la elección del elemento *6* de P_1 obliga a coger el elemento 1 de P_1. Esta regla se sigue aplicando hasta que se complete un ciclo, momento en el cual los restantes elementos se completan del otro vector progenitor. De esta forma se obtendrían como descendientes $Q_1=(6,3,2,7,4,1,5,10,9,8)$ y $Q_2=(1,6,9,3,8,5,2,7,10,4)$. La Figura 14 ilustra de nuevo el proceso de obtención de los descendientes.

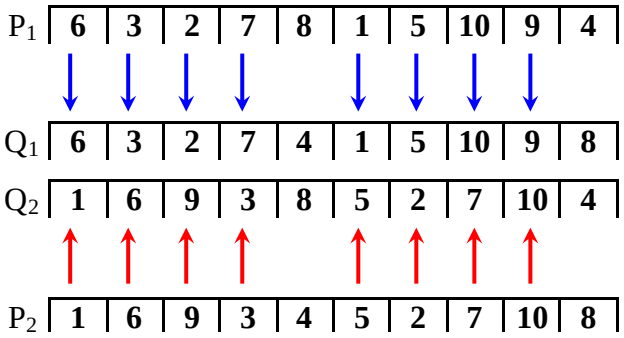


Figura 14: Ejemplo de cruce utilizando el operador CX.

Los tres operadores analizados (PMX, CX y OX) obtienen soluciones admisibles (posibles distribuciones de las secciones en las áreas) a partir de dos soluciones admisibles.

3.9.5. OPERADOR DE MUTACIÓN

Un operador de mutación para el problema considerado puede consistir en intercambiar las posiciones de dos elementos del vector elegidos de manera aleatoria.

Por ejemplo, dada la distribución $P=(1,4,7,2,8,9,5,3,10,6)$ y las posiciones de mutación $n_1=2$ y $n_2=7$, elegidas aleatoriamente, el resultado de la mutación sería una distribución ligeramente diferente: $P'=(1,5,7,2,8,9,2,3,10,6)$.

3.9.6. RESULTADOS EXPERIMENTALES

Para probar la efectividad del algoritmo, se implementó un algoritmo genético con las siguientes características:

Tamaño de población:	*30*
Número de generaciones:	*30*
Operador de cruce:	*PMX*
Probabilidad de cruce:	*0.7*
Probabilidad de mutación:	*0.1*

Tomando como base la población inicial de la Tabla 5, los resultados obtenidos se resumen en la Tabla 6 así como en las Figuras 15 y 16, en las que puede observarse la evolución decreciente del coste máximo, mínimo y medio en cada generación.

			Costes			
Generación	*Número de cruces*	*Número de mutaciones*	*Media*	*Desv. Típica*	*Min*	*Max*
1	11	0	897.1	72.9456	759.	1040.
2	11	2	852.3	63.3753	759.	982.
3	12	5	842.133	61.8506	745.	1040.
4	15	6	829.267	58.0909	711.	946.
5	10	3	818.733	82.6872	708.	1002.
6	11	2	788.733	51.0341	690.	873.
7	15	4	778.633	57.3913	675.	881.
8	9	4	765.1	63.553	633.	908.
9	12	2	771.	77.0298	609.	929.
10	9	3	762.633	83.4363	653.	1079.
11	12	6	741.8	69.7065	651.	883.
12	10	3	737.633	69.9199	633.	883.
13	13	1	710.6	62.3055	636.	859.
14	8	1	683.9	43.2812	626.	787.
15	11	1	681.033	57.6287	626.	867.
16	13	6	684.767	46.3217	626.	797.
17	13	3	672.2	42.9686	627.	794.
18	10	4	667.367	46.537	626.	859.
19	11	3	653.533	26.6959	621.	749.
20	11	3	661.467	52.5618	621.	815.
21	12	2	641.7	20.3455	609.	695.
22	8	4	644.767	39.8434	609.	814.
23	14	5	642.467	31.581	598.	748.
24	8	3	642.467	34.0838	609.	750.
25	11	8	650.5	52.3218	598.	824.
26	8	5	649.367	48.84	598.	791.

27	10	1	639.867	39.1282	598.	764.
28	8	2	624.567	21.0708	598.	693.
29	8	3	619.8	19.6704	598.	695.
30	11	3	619.967	33.1178	598.	781.

Tabla 6: Informe de evolución.

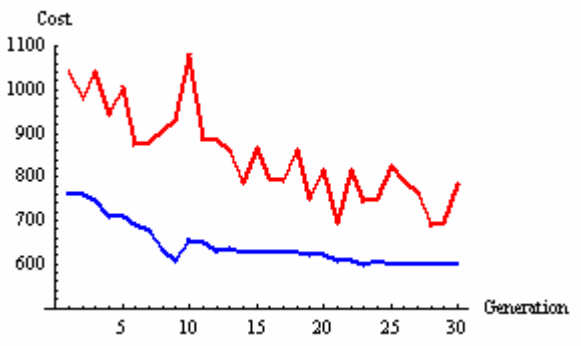


Figura 15: Evolución del coste máximo y mínimo.

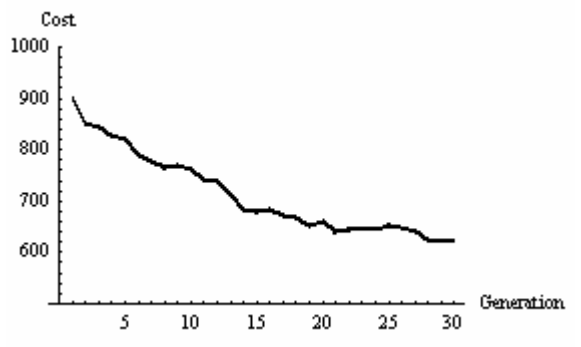


Figura 16: Evolución del coste medio.

Finalmente, en la Tabla 7 puede verse la última generación de soluciones potenciales obtenida. La mejor distribución después de las 30 generaciones es *P=(9,3,5,10,1,2,4,8,6,7*) con un coste asociado de 598. Observar también que se ha obtenido un conjunto de buenas distribuciones, además pueden descubrirse importantes similitudes (esquemas) entre los individuos de esta última generación. Por ejemplo, todas las potenciales distribuciones sitúan la sección 9 en el primer área, y las secciones 6, 7 y 8 siempre quedan colocadas en las tres últimas áreas.

Distribución potencial	*Coste*	*Distribución potencial*	*Coste*
P_1=(9,3,4,10,1,2,5,8,7,6)	624	P_{16}=(9,3,5,10,1,2,4,8,7,6)	609

P_2=(9,1,5,10,3,2,4,8,6,7)	601	P_{17}=(9,10,5,3,1,2,4,8,6,7)	618
P_3=(9,3,2,10,1,5,4,8,6,7)	616	P_{18}=(9,3,5,10,1,2,4,8,7,6)	609
P_4=(9,3,2,10,1,5,4,8,7,6)	627	P_{19}=(9,3,5,10,1,2,4,8,6,7)	598
P_5=(9,1,5,10,3,2,4,8,7,6)	612	P_{20}=(9,10,5,3,1,2,4,8,6,7)	618
P_6=(9,3,4,10,1,2,5,8,7,6)	624	P_{21}=(9,10,5,3,1,2,4,8,7,6)	629
P_7=(9,3,5,10,1,2,4,8,6,7)	598	P_{22}=(9,1,5,10,3,2,4,8,6,7)	601
P_8=(9,3,5,10,1,2,4,8,7,6)	609	P_{23}=(9,3,4,10,1,2,5,8,7,6)	624
P_9=(9,3,2,10,1,5,4,8,7,6)	627	P_{24}=(9,3,5,10,1,2,4,8,7,6)	609
P_{11}=(9,8,5,10,3,2,4,1,7,6)	781	P_{25}=(9,3,5,10,1,2,4,8,6,7)	598
P_{11}=(9,3,5,10,1,2,4,8,6,7)	598	P_{26}=(9,3,4,1,10,2,5,8,6,7)	657
P_{12}=(9,3,5,10,1,2,4,8,7,6)	609	P_{27}=(9,3,5,10,1,2,4,8,7,6)	609
P_{13}=(9,3,4,10,1,2,5,8,7,6)	624	P_{28}=(9,3,5,10,1,2,4,8,6,7)	598
P_{14}=(9,3,5,10,1,2,4,8,7,6)	609	P_{29}=(9,3,1,10,2,5,4,8,6,7)	610
P_{15}=(9,10,5,3,1,2,4,8,7,6)	629	P_{30}=(9,3,4,10,1,2,5,8,7,6)	624

Tabla 7: Población después de 30 generaciones.

Este sencillo ejemplo ha pretendido ilustrar el modo de funcionamiento de los algoritmos genéticos; evidentemente, el algoritmo puede ser refinado incluyendo aspectos como penalizaciones a determinados individuos, eliminación de duplicados, poblaciones de tamaño variable, etc.

BIBLIOGRAFÍA

ADRIAN, E. (1946): *The Physical Background of Perception*, Clarendon Press, Oxford.

ANDERSON, J. (1972): "A Simple Neural Network Generating an Interactive Memory". *Mathematical Biosciences* 14:197-220.

BARR, A. & FEIGENBAUM, E.A. (1981): *The Handbook of Artificial Intelligence (vol I)*. Ed William Kaufman.

BORRAJO, D.; JURISTO, N.; MARTÍNEZ, M. & PAZOS, J (1997): *Inteligencia Artificial. Métodos y técnicas.* Ed. Centro de Estudios Ramón Areces.

BUNDY, A. (1983): *The Computer Modelling of Mathematical Reasoning*. Academic Press.

CASTILLO, E., COBO, A., GÓMEZ, P., SOLARES, C. & CASTILLO, C. (1996): "Un Sistema Experto para el control de Semáforos en Java". *PowerScience* nº 6, pag 6.

CASTILLO, E., GUTIÉRREZ, J.M. & HADI, A.S. (1996): *Sistemas Expertos y Modelos de Redes Probabilísiticas.* Monografías de la Academia de Ingeniería, Madrid.

COBO, A. & SERRANO, A. (2000): "Application of Genetic Algorithms for Solving Job-Shop Layout Problems". *First World Conference on Production and Operations Management*, Sevilla.

DAVIS, L. (1985): "Job shop scheduling with genetic algorithms". *Proceedings of the International Conference on Genetics Algorithms and Their Applications, 136-140*.

DOMÍNGUEZ MACHUCA, J. M. (Coord. y Dtor.), ALVAREZ, M. J., GARCÍA, S., DOMÍNGUEZ, M.A. & RUIZ, A. (1995): *Dirección de Operaciones. Aspectos estratégicos en la producción y los servicios.* McGraw-Hill.

DUDA, R. & HART, P. (1973): *Pattern Classification and Scene Analysis*. Wiley, New York.

DURKIN, J. (1994): *Expert Systems: Design and Development*. Ed Maxwell McMillan.

FEIGENBAUM, E. A. (1982): *Knowledge Engineering for the 1980s*. Dpt. of Computer Science, Stanford University.

FISHER, E. L. (1986): "An AI Based Methodology for Factory Design". *AI Magazine*, 3-4 (otoño), 72-85.

GOLDBERG, D.E. (1985): "Optimal Initial Population Size for Binary-Coded Genetic Algorithms". *TCGA Report N0.85001. University of Alabama*.

GOLDBERG, D.E. (1989): *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.

GUPTA, M.M. & RAO, D.H. (1994): "On the Principles of Fuzzy Neural Networks". *Fuzzy Sets and Systems*, 61, 1-18.

HASSOUN, M.H. & CLARK, D.W. (1988): "An Adaptive Attentive Learning Algorithm for Single Layer Neural Networks". *Learning Algorithms II, Proceedings of the First Annual IEEE Conference on Neural Networks, vol. I,* pp. 431-440.

HAUGELAND, J. (1985): *Artificial Intelligence: The very Idea*. MIT, Bardford Books.

HEBB, D. (1949): *The Organization of Behavior.* Wiley, New York.

HINOJOSA, M.A., MARTÍN, J.L. & VÁZQUEZ, M.J. (1993): "Optimización de carteras de opciones mediante algoritmos genéticos". *ESIC Market, Ene-Mar 1993*, 93-113.

HOLLAND, J. (1975): *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

HOPFIELD, J. & TANK, D. (1985): *Biological Cybernetics*, Vol. 52, pp. 141-152.

HOPFIELD, J.J. (1982): "Neural Networks and Physical Systems with Emergent Collective Computational Abilities". *Proceedings of the National Academy of Sicences*, USA, 79, 2554-2558.

KUFFLER, S., NICHOLLS, J. & MARTIN, A. (1984): *From Neuron to Brain (2nd ed)*. Sinauer Publishers, Sunderland, Mass.

LEE, C., DOOLEN, G., CHAEN, H., SUN, G., MAXWELL, T., LEE, H. & GILES, C. (1986): "Machine Learning using a Higher Order Correlation Network". *Physica D*, 22, 276-306.

McCORDUCK, P. (1991): *Máquinas que piensan. Una incursión personal en la historia y las perspectivas de la inteligencia artificial*. Ed Tecnos.

McCULLOCH, W. & PITTS W. (1943): "A Logical Calculas of the Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics* 5:115-133.

MICHALEWICZ, Z. (1996): *Genetic Algorithms + Data Structure = Evolution Programs*. 3ª ed. Springer.

OLIVER, I.M., SMITH, D.J. & HOLLAND, J.R.C. (1987). "A study of permutation crossover operators on the traveling salesman problem". *Genetic Algorithms and their Applications: Proceedings of the II International Conference on Genetic Algorithms*, 132-139.

RABINER, L. & JUANG, B. H. (1993): *Fundamentals os Speech Recognition*. Prentice Hall.

SAPHIRO, L. G. & ROSENFELD, A. (1992): *Computer Vision and Image Processing*. Academic Press.

SPECHT, D.F. (1990): "Probabilistic Neural Networks". *Neural Networks*, 3, 109-118.

WEGNEZ, L. (1987): *A la recherche de l'intelligence*. O.I.L., Bruselas.

WOS, L., OVERBEEK, R, LUSK, E. & BOYLE, J. (1984): *Automated Reasoning. Introduction and Applications.* Prentice Hall.

ZHANG, Q. & BENVENISTE, A. (1992): "Wavelet Neural Networks". *IEEE Transactions on Neural Networks*, 3, 889-898.

Año 2000:
"Año Mundial de las Matemáticas"

DECLARACION de RIO de JANEIRO

El 6 de Mayo de 1992, en Río de Janeiro, durante la celebración del 40 aniversario del mundialmente conocido Instituto de Matemática Pura y Aplicada (IMPA), el profesor Jacques-Louis Lions, Presidente de la Unión Matemática Internacional (IMU), declaró en nombre de la misma, que el año 2000 sería el Año Matemático Mundial (WMY2000).

El WMY 2000 está bajo el patrocinio de la UNESCO, de la Academia de Ciencias del Tercer Mundo, del Ministerio francés de Investigación y Espacio, de la Academia Brasileña de Ciencias y del Consejo Federal Suizo.

La declaración de Río de Janeiro estableció tres objetivos:

1. Primer objetivo: Los grandes desafíos del siglo XXI

Durante su conferencia en París en 1900, David Hilbert enunció una lista con los principales problemas matemáticos que la entonces recién comenzada centuria debería abordar.

La American Mathematical Society sugirió en 1990, durante la asamblea general de la IMU en Kobe (Japón), que matemáticos de primera línea organicen sus esfuerzos para proponer cuáles deberían ser los grandes retos del año 2000. Este Comité está dirigido por el Profesor Jacob Palis Jr., IMPA (Brasil).

2. Segundo objetivo: las Matemáticas, clave par el desarrollo

Las Matemáticas puras y aplicadas son una de las claves más importantes para la comprensión del mundo y de su desarrollo. Por ello es esencial que los países miembros de la UNESCO sean capaces, gradualmente, de alcanzar un nivel que haga posible su admisión en la IMU, que actualmente cuenta con 50 países miembros.

Por tanto, el segundo objetivo de la Declaración de Río de Janeiro es que la mayor parte de los países miembros de la UNESCO alcancen tal nivel con el cambio de siglo. Esto implica un gran esfuerzo adicional en los campos de la Educación, de la Formación, y - un punto muy importante para los países que afrontan dificultades económicas-, de acceso a la información científica.

Estos esfuerzos, que ya están siendo emprendidos, se ratificarán e incrementarán por las dos principales comisiones de la IMU: ICMI (International Commission on Mathematical Instruction) y la CDE (Comisión de Desarrollo e Intercambio). Ambas comisione están en contacto con la UNESCO, que estuvo representada en Río de Janeiro por el Profesor A. Marzollo, responsable para las Matemáticas.

3. Tercer objetivo: la imagen de las Matemáticas

La declaración de Río de Janeiro se plantea como tercer objetivo, que también es de gran importancia, una presencia sistemática de las Matemáticas en la "Sociedad de la Información", gracias a ejemplos y aplicaciones que sean a la vez científicamente exactas y accesibles al mayor número de personas. Este aspecto se desarrollará en conexión con los esfuerzos que ya han sido emprendidos por varios países que son miembros de IMU.

La declaración de Río de Janeiro anunciando el Año Mundial de las Matemáticas 2000 fue calurosamente apoyado no sólo por todos los matemáticos presentes en Río, llegados de todos lo continentes y, por supuesto por muchos de los matemáticos brasileños más eminentes, sino también por especialistas en otros temas, especialmente por el Profesor Carlos Chagas, actual Presidente de la Academia Pontificia de Ciencias.

PRIMERA RESOLUCIÓN DE LA UNESCO

Asamblea General

(Aprobada el 11 de Noviembre de 1997)

- Considerando la importancia central de las Matemáticas y de sus aplicaciones en el mundo de hoy, con respecto a las ciencias, las tecnologías, las comunicaciones y la economía, entre otros campos.
- Consciente de que las Matemáticas están profundamente enraizadas en muchas culturas, y de que los más destacados pensadores contribuyeron significativamente, a lo largo de miles de años, a su desarrollo.
- Consciente de que el lenguaje y los valores de las Matemáticas son universales, por tanto favorecen y la hacen muy apropiada para la cooperación internacional.
- Recalcando que la educación matemática juega un papel decisivo, en particular en la escolarización primaria y secundaria, para la comprensión de conceptos matemáticos básicos y para el desarrollo del pensamiento racional.
- Da la bienvenida a la iniciativa de la Unión Matemática Internacional (IMU) para declarar el año 2000 Año Mundial de las Matemáticas y lleva a cabo, dentro de su sistema, actividades para promover las Matemáticas en todos los niveles a lo ancho del mundo
- Decide apoyar la iniciativa del año 2000 como Año Mundial de las Matemáticas.
- Pide al Director General la colaboración con la comunidad matemática internacional en la planificación del Año Mundial de las Matemáticas 2000 y contribuir, durante 1998-1999, con 20.000 dólares del Programa y Presupuesto Ordinario, en apoyo a las actividades preparatorias