Inteligencia artificial

Miquel Barceló García

PID_00153113

UOC
Universitat Oberta de Catalunya
www.uoc.edu

Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea éste eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares del copyright.

Índice

Introducción 7

Objetivos 8

1. Definición, objetivos y otras denominaciones 9
1.1. El test de Turing 9
1.2. Objetivos centrales de la investigación en inteligencia artificial 10
1.3. Otras definiciones 10
1.4. Dos enfoques opuestos 12
1.5. Otras denominaciones 12

2. Aplicaciones de la IA 14
2.1. Sistemas expertos 14
2.2. Sistemas de visión por ordenador 14
2.3. Sistemas de lenguaje natural 15
2.4. Resolución de problemas 15
2.5. Robótica 15
2.6. Aprendizaje 16
2.7. Juegos y puzzles 16

3. Técnicas de la IA 17
3.1. Técnicas de base de la IA 17
3.2. Inferencia simbólica 17
3.3. Heurística 18

4. La representación del conocimiento 19
4.1. Componentes y elementos que se deben representar 19
4.2. Etapas en el uso del conocimiento 20
4.3. Propiedades de un buen formalismo de representación del conocimiento 20
4.4. Los espacios de estados 21
4.5. Esquemas modernos de representación del conocimiento 21
4.5.1. Los esquemas lógicos 21
4.5.2. Las redes semánticas 22
4.5.3. Los esquemas de representación de procedimientos 23
4.5.4. Los *frames* 23
4.5.5. Los *scripts* 24

5. Sistemas expertos 26
5.1. Definición y características 26

5.1.1. Atributos de un experto humano 26
5.1.2. Definición y características de un sistema experto 27
5.2. Estructura básica de un sistema experto 27
5.2.1. La base de conocimientos 28
5.2.2. La base de hechos 28
5.2.3. El motor de inferencia 29
5.2.4. Las interfaces 29
5.3. Bases de conocimientos: tipos de reglas 31
5.3.1. Reglas antecedente-consecuencia 31
5.3.2. Reglas condición-acción 31
5.3.3. Reglas inexactas 31
5.4. Motor de inferencia: funcionamiento y mecanismos de razonamiento 32
5.4.1. Razonamiento por encadenamiento hacia delante 32
5.4.2. Razonamiento por encadenamiento hacia atrás 33
5.4.3. Razonamiento inexacto 34
5.5. Ingeniería del conocimiento 34
5.5.1. Funciones del ingeniero del conocimiento 35
5.5.2. *Shells* y conjuntos de herramientas (*toolkits*) 36
5.6. Viabilidad y beneficios de un sistema experto 36
5.6.1. Requisitos de viabilidad de un sistema experto 37
5.6.2. Beneficios que se han de obtener de un sistema experto 37
5.7. Problemas en el uso de sistemas expertos 38
5.8. Algunos sistemas expertos famosos 39
5.8.1. MYCIN 39
5.8.2. MACSYMA 39
5.8.3. PROSPECTOR 39
5.8.4. DENDRAL 40
5.8.5. XCON 40
5.8.6. Otras aplicaciones 40

6. Redes neuronales..... 41
6.1. Descripción 41
6.1.1. Modelos de redes neuronales artificiales 41
6.1.2. Algunas características de las redes neuronales 42
6.1.3. Otras denominaciones 42
6.1.4. Aplicaciones y funcionamiento 43
6.1.5. Neuroordenadores y *netware*..... 44
6.1.6. Características particulares de las redes neuronales 45
6.2. Arquitectura y topología de las redes neuronales 45
6.2.1. Estructura topológica 45
6.2.2. Estructura en capas de una red neuronal 46
6.2.3. Conexiones entre las neuronas 48
6.3. Dinámica y aprendizaje de una red neuronal 48
6.3.1. Fase de ejecución 48
6.3.2. Fase de aprendizaje 50

6.4. Historia de las redes neuronales 51
6.4.1. El *perceptron* de F. Rosenblatt 52
6.4.2. El predictor meteorológico de Widrow 52
6.4.3. El nuevo interés por las redes neuronales 52
6.5. Las redes neuronales y el cerebro humano 53

7. Lenguajes específicos para la IA..... 55
7.1. Lisp 55
7.1.1. Origen 55
7.1.2. Objetivos y características principales 55
7.1.3. Dialectos 56
7.2. Prolog 56
7.2.1. Orígenes 56
7.2.2. Objetivos y características principales 57
7.2.3. Dialectos 57
7.3. Smalltalk 57

Actividades..... 59

Ejercicios de autoevaluación..... 59

Solucionario..... 60

Bibliografía..... 61

Introducción

La inteligencia artificial es, con toda seguridad, el intento más ambicioso de los que aborda la informática. Superado el estadio de construir máquinas más fuertes y rápidas que su propio constructor, al ser humano todavía le queda el reto de construir máquinas que al menos parezcan inteligentes, como se supone que lo somos nosotros.

Intentos de la inteligencia artificial

Después de intentos un poco esotéricos como los *homúnculos* de Paracels o el *golem* que obedece al rabino de Praga por el efecto de una palabra mágica, la inteligencia artificial intenta traer a la realidad tecnológica la construcción de máquinas y robots que se comporten inteligentemente.

Aunque pueda parecer un campo emparentado con la ciencia-ficción, la realidad es que hoy día disponemos de programas que superan a los humanos en algunos aspectos de lo que consideramos actividad en cierta manera inteligente.

La inteligencia es difícil de definir, y por ello nadie pretende construir programas inteligentes sino, simplemente, programas con un comportamiento que calificaríamos de inteligente en un ser humano. Es un poco parecido a conseguir moverse por las aguas marinas como hace, por ejemplo, un submarino, pero que pocos se atreverían a etiquetar de "nadar" lo que hace un submarino...

Este módulo presenta los **problemas centrales** de la inteligencia artificial y sus **técnicas particulares**, y explica con más detalle algunos casos concretos como los resultados obtenidos con los sistemas expertos y el renacimiento de la tecnología de las redes neuronales.

Objetivos

Los objetivos que el estudiante puede alcanzar son los siguientes:

1. Definir y delimitar el amplio campo de la IA, indicando sus principales aplicaciones.

2. Introducirse en el estudio de las principales técnicas que utiliza la IA.

3. Asumir la problemática de cómo se representa el conocimiento y las diferentes propuestas en curso.

4. Conocer con un cierto detalle las posibilidades de los sistemas expertos, las técnicas específicas y los ejemplos más característicos.

5. Comentar el renacimiento de la técnica de las redes neuronales y exponer las principales características.

6. Saber cuáles son los lenguajes específicos utilizados en la IA.

1. Definición, objetivos y otras denominaciones

Es bastante difícil definir exactamente el contenido y el alcance de la inteligencia artificial (IA), a pesar de los largos años transcurridos desde que John McCarthy acuñara el nombre en el año 1956 en la Darmouth Conference en Hannover (New Hampshire, Estados Unidos).

Quizá una de las caracterizaciones más breves y sencillas es aquella que, parafraseando Marvin Minsky, uno de los expertos e investigadores más famosos de la IA, asigna a la inteligencia artificial la "realización de sistemas informáticos con un comportamiento que en el ser humano calificaríamos de inteligente".

1.1. El test de Turing

A este efecto, es famoso el denominado **test de Turing**, establecido por Alan Turing en un artículo que data de 1950 ("Computing machinery and intelligence") para proponer la realización de un experimento que permita discernir el carácter inteligente o no del comportamiento de una máquina.

El test mencionado parte de un juego en el que un interrogador debe averiguar el sexo de dos interlocutores, A y B, situados en una habitación separada y que, aunque ambos dicen que son mujeres, en realidad son un hombre y una mujer. En la propuesta original de Turing se trata de sustituir a la mujer por un ordenador y esto se ha generalizado después en la forma siguiente: el interrogador ha de averiguar quién es la máquina a partir de la conversación con los dos interlocutores, una persona y un ordenador, aunque ambos dicen ser personas.

Este objetivo se ha de conseguir a pesar de saber que ambos interlocutores no están obligados a decir la verdad y que, por ejemplo, la máquina puede decidir dar un resultado erróneo a una multiplicación e, incluso, comunicarlo después de muchos segundos o minutos de haberlo obtenido para engañar al interrogador.

En la hipótesis optimista del mismo Turing, hacia el año 2000 se dispondría de ordenadores suficientemente potentes "para hacerles jugar tan bien el mencionado juego que un interrogador normal no tendrá más del 70% de posibilidades de efectuar la identificación correcta al cabo de cinco minutos de haber planteado las preguntas".

Si fuera así, estaríamos ante una máquina verdaderamente "inteligente" o, cuando menos, que se sabe hacer pasar por tal.

Es evidente que la predicción de Turing peca de un optimismo exagerado, algo que, como veremos, es un error muy frecuente en los inicios de la IA.

En realidad, el problema no afecta sólo a la potencia del ordenador, sino a las potencialidades de su programación para poder mantener una conducta inteligente.

1.2. Objetivos centrales de la investigación en inteligencia artificial

Actualmente se puede afirmar que cuando se investiga en el campo de la inteligencia artificial se pueden perseguir dos objetivos complementarios que ponen el énfasis respectivamente en el aspecto teórico o tecnológico de la IA.

Un primer objetivo es **el estudio de los procesos cognoscitivos en general**, lo cual justificaría la definición de Patrick J. Haye en *El estudio de la inteligencia como computación* (1973) que se orienta a la consideración de la inteligencia artificial como estudio de la conducta humana inteligente.

En este estudio es fundamental la ayuda obtenida de la informática como herramienta de gran potencia en la representación de cualquier sistema de símbolos como el que utiliza la actividad racional. De aquí los múltiples puntos de contacto de la inteligencia artificial con otras ciencias como la neurofisiología, la psicología, la lógica formal, la lingüística, etc.

Otro objetivo central de la IA es **el intento de obtener sistemas automáticos capaces de llevar a cabo tareas reservadas, incluso ahora, a los seres humanos**.

Con este enfoque, la IA aparece como una disciplina eminentemente tecnológica que persigue la construcción de máquinas y programas capaces de realizar tareas complejas con una habilidad y eficiencia iguales o superiores a las que consigue el ser humano.

1.3. Otras definiciones

Los dos objetivos se suceden en las distintas definiciones que los especialistas más importantes de la IA han ido dando a los contenidos de esta ciencia.

Para Marvin Minsky, profesor del MIT (Massachussetts Institute of Technology) y uno de los investigadores y expertos más famosos en inteligencia artificial, define la inteligencia artificial como:

> "la ciencia de construir máquinas que hacen cosas que realizadas por el hombre requieren el uso de la inteligencia".

Otro punto de vista lo ofrecen B. G. Buchanan y E. A. Feigenbaum, de la Universidad de Stanford, cuando dicen que:

> "la investigación sobre IA es la parte de la ciencia de los ordenadores que investiga procesos simbólicos, razonamientos no algorítmicos y representaciones simbólicas del conocimiento usados en máquinas inteligentes".

Es importante recordar que se trata de un uso de los ordenadores ("máquinas inteligentes") que no respeta la habitual caracterización de un programa como la conjunción de unos algoritmos actuando sobre unas estructuras de datos. En realidad los programas de la IA son también, y tal vez esencialmente, no algorítmicos y así lo constataremos más adelante.

También, en una posición intermedia, P. H. Winston, director del laboratorio de inteligencia artificial del MIT, señala que:

> "el objetivo de la IA se puede concretar en cómo conseguir hacer ordenadores más útiles para comprender los principios que hacen posible la inteligencia".

En otras definiciones se intenta analizar el contenido semántico de los términos "inteligencia" y "artificial" que componen la denominación, pero este enfoque también parece plantear graves problemas por las dificultades de definir los mismos conceptos de inteligencia o de artificialidad.

Al lado del contenido ya práctico de la definición de Minsky, otros autores no dudan en proponer contenidos todavía más pragmáticos y dicen que:

> "la inteligencia artificial es el estudio de las facultades mentales mediante el uso de modelos computacionales"

Esta definición pone énfasis en el interés de las técnicas de la IA en duplicar (reproducir) las facultades mentales de las personas: visión, lenguaje natural, razonamiento y comprensión, por poner ejemplos concretos ya tratados por las técnicas de la IA.

Otra definición

Otra definición complementaria y en la misma línea, también procedente de un libro de texto, es la que ofrece Elaine Rich en 1983 al decir que "la inteligencia artificial es el estudio de cómo se puede hacer que los ordenadores hagan cosas que, de momento, las personas hacen mejor".

Quizá estas definiciones ilustran más exactamente el contenido actual de la inteligencia artificial que el ambicioso objetivo de superar el test de Turing con una máquina o un programa de ordenador.

1.4. Dos enfoques opuestos

Un elemento importante en la realidad actual e histórica de la inteligencia artificial es que hay dos enfoques opuestos sobre cómo se ha de conseguir este estudio y duplicación tecnológica de la inteligencia y los procesos cognoscitivos habituales en el ser humano.

Por una parte, **John McCarthy**, inventor de la denominación "inteligencia artificial", creador del lenguaje Lisp y uno de los investigadores más famosos de la IA, recomienda una línea de investigación que persigue la obtención de programas de ordenador que razonen siguiendo básicamente las exigencias de los dictados de la **lógica matemática**.

En el extremo opuesto, **Marvin Minsky**, también famoso experto de la IA, propone imitar fundamentalmente el método de **razonamiento de la mente humana** que, según él, no tiene por qué ser necesariamente el mismo de la lógica matemática.

Quizá el enfoque de McCarthy presenta más facilidades para llegar, cuando menos, a ciertos resultados. Pero la observación de Minsky establece las bases de una duda razonable sobre si los resultados obtenidos por el enfoque propuesto por McCarthy tienen realmente algo que ver con la inteligencia humana o tan sólo son meras aproximaciones que pueden cerrar el camino a la verdadera imitación de la inteligencia humana, si éste es el objetivo principal, lo cual, como ya se ha dicho, es puesto en duda por otros investigadores más pragmáticos.

De aquí que, en conjunto, las técnicas de la IA, en su intento de imitar las características de la mente humana, traten, en cualquier caso, el estudio de los métodos basados en la representación simbólica del conocimiento o en la inferencia simbólica, y se preocupen por problemáticas como la comprensión de los procesos cognoscitivos, el razonamiento y la toma de decisiones, la comprensión y la utilización del lenguaje natural –en oposición a lenguajes más formales y más "fáciles" para los sistemas informáticos–, la percepción visual, los procesos de aprendizaje, etc.

1.5. Otras denominaciones

El nuevo apogeo de las técnicas de la inteligencia artificial en los últimos años provocó, en la década de los ochenta, que algunos intentaran utilizar las mismas iniciales (IA) como indicadores también de *informática adelantada*, en un intento de expresar que, a pesar de sus más de treinta años de antigüedad, el éxito de las técnicas de la IA la configuraba como uno de los aspectos más

prometedores de la informática y uno de los más importantes con vista al futuro inmediato de la tecnología informática. Esta denominación, aunque ha sido frecuentemente utilizada en el mundo comercial y empresarial, resulta bastante despreciada en el mundo académico, preocupado básicamente por los aspectos teóricos de la IA y que, por lo tanto, recomienda preferentemente denominaciones como **sistemas basados en el conocimiento** o **ciencia de los procesos cognoscitivos**.

También hay que mencionar la observación de **John Haugeland**, quien propone el término **inteligencia sintética** para destacar, al lado de la artificialidad de la inteligencia conseguida por la IA, el hecho de que ésta resulta originada en la actividad humana, que "sintetiza" así una nueva forma de su propia inteligencia.

2. Aplicaciones de la IA

En la actualidad, las técnicas de la inteligencia artificial se aplican en muchos campos. A continuación destacaremos algunos de ellos:

- Sistemas expertos.
- Sistemas de visión por ordenador.
- Sistemas de lenguaje natural.
- Resolución de problemas.
- Robótica.
- Aprendizaje.
- Juegos y puzzles.

2.1. Sistemas expertos

Los sistemas expertos constituyen el eje central de una nueva actividad informática conocida como **ingeniería del conocimiento**. Con ellos se persigue la construcción de programas que reproduzcan de manera correcta el comportamiento de un experto humano en su dominio concreto de competencia y pericia.

Los sistemas expertos también reciben el nombre de *sistemas basados en el conocimiento* y, con intenciones más teóricas, *sistemas de producción*.

2.2. Sistemas de visión por ordenador

Después del objetivo central del antiguo *perceptron* de Rosenblat, los sistemas de visión por ordenador persiguen el **reconocimiento de formas** con el objetivo de conseguir la **visión por ordenador**, tanto con la finalidad de aplicarla en el ámbito de la robótica como en su aplicación en medicina, por ejemplo para ayudar a las personas que sufren ceguera.

***Perceptron* de Rosenblat**

Este tema se trata más adelante en el apartado "El *perceptron* de Rosenblat"

2.3. Sistemas de lenguaje natural

El lenguaje que utilizamos los seres humanos se ha etiquetado de "lenguaje natural" en contraposición a los lenguajes "artificiales", que se han diseñado para los ordenadores. La **comprensión del lenguaje natural** es muy difícil, ya que hay términos o expresiones con más de un significado.

Temas de estudio

Entre los **temas de estudio** del lenguaje natural se incluyen la comprensión y el análisis de textos, la traducción automática y la generación de informes o textos. También se incluyen el tratamiento de la palabra hablada y otras técnicas auxiliares como el reconocimiento vocal y el tratamiento y la comprensión de la expresión oral.

El tratamiento del lenguaje natural es uno de los temas más interesantes que trata la IA, tanto por el objetivo del estudio en sí mismo como por la posibilidad de poder acceder en un futuro a los ordenadores utilizando el mismo lenguaje natural que utilizamos los seres humanos, propósito central del proyecto japonés de la llamada "quinta generación de ordenadores".

2.4. Resolución de problemas

Uno de los temas centrales de la IA es el uso de la inteligencia para la resolución de todo tipo de problemas.

Los objetivos centrales han sido encontrar los mecanismos de deducción, buscar soluciones y conseguir la explicación de la solución que se ha obtenido y su justificación.

Resolutor general de problemas

Después del intento del resolutor general de problemas (GPS, *general problem solver*), en 1972 Newell y Simon publicaron un influyente libro sobre este tema: *Human Problem Solving*.

También se estudia la planificación inteligente de conductas y procedimientos de acción. Aunque en este tema los sistemas expertos han avanzado mucho más en sus aplicaciones puntuales, el estudio de la resolución de problemas continúa siendo la base teórica de un planteamiento más general.

2.5. Robótica

Como resultado de la evolución directa de la ingeniería mecánica y la automatización, la IA también intenta conseguir el **aumento de la habilidad** y la **autonomía** de estos mecanismos que manipulan instrumentos y que denominamos *robots*.

La robótica necesita tanto la percepción (visual o de otra manera) del entorno donde trabaja el robot, como la planificación de la acción que se ha de emprender en cada caso. Aunque con gran éxito y una aplicación creciente en el mundo industrial, los robots que se han conseguido crear todavía están muy lejos de los de ficción, que son los que configuran la imagen y las expectativas de la robótica en la mayoría del público.

2.6. Aprendizaje

Es posible que **la esencia de la inteligencia sea la capacidad de aprender**, es decir, de obtener conocimientos nuevos a partir de la experiencia y los conocimientos ya adquiridos. El objetivo de hacer que los programas "aprendan" es central en las técnicas de la IA. Uno de los ejemplos más conocidos son los programas que juegan a ciertos juegos de contenido estratégico (ajedrez, damas, bridge, etc.) en los que, frecuentemente, también se persigue que, a partir de los resultados obtenidos, mejoren su comportamiento en futuras partidas.

Al lado del interés que los programas de ordenador puedan "aprender", también se da mucha importancia al hecho de que la informática se pueda utilizar para favorecer el proceso de aprendizaje de los seres humanos. Se habla así **de enseñanza asistida por ordenador** (EAO), en la cual destaca, entre otras utilizaciones, la creación de las llamadas *situaciones de descubrimiento* gracias a la experimentación directa con ciertas herramientas informáticas.

Un elemento central de esta utilización pedagógica de la IA es el lenguaje Logo, desarrollado en el MIT por **Seymour Papert** adaptando ideas nacidas en el Lisp. El amplio universo pedagógico que ha abierto el Logo y sus "micromundos" es uno de los escasos ejemplos en los que la utilización de una nueva tecnología en la educación también ha venido acompañada de un método pedagógico *ad hoc*.

2.7. Juegos y puzzles

Una de las aplicaciones más inmediatas de la investigación en IA ha sido la elaboración de **programas de juegos** que, al mismo tiempo, puedan ser ejemplos de la aplicación de las técnicas de resolución de problemas y de aprendizaje, cuando se aplican a universos o dominios restringidos como son, en definitiva, los dominios que describen las reglas de los juegos. En este aspecto destaca la aplicación de técnicas de IA en programas de ordenador que juegan a damas, ajedrez o bridge y otros juegos estratégicos de sala para cuya práctica se presume una cierta necesidad de inteligencia.

Se han estudiado los juegos y los puzzles porque permiten tratar de manera restringida las dificultades generales de la resolución de problemas. Básicamente se intenta encontrar una representación adecuada del espacio de casos propios del problema para configurar lo que se denomina "espacios de estados".

Ejemplos de juegos

Ejemplos utilizados sobradamente en la investigación son los juegos sencillos como el tres en ralla, los puzzles simples, o disponer ocho reinas en un tablero de ajedrez sin que se "coman" entre sí.

La reducción de posibilidades de los universos explorados en los juegos –respecto del mundo real– ofrece la posibilidad de tratar de modo particular los graves problemas derivados de la explosión combinatoria del número de casos posibles y de las soluciones que hay que probar.

3. Técnicas de la IA

Cuando se aborda la resolución de un problema con técnicas de IA, se trata tanto de obtener la solución como de poder exponer el proceso de razonamiento que se ha utilizado. Obviamente, se deberán obtener varias soluciones si existe más de una y justificar el interés de cada una de ellas. También es importante que el sistema de IA sepa mejorar el razonamiento utilizado a medida que aumenta la experiencia, hecho que constituye su forma concreta de aprendizaje.

3.1. Técnicas de base de la IA

De la misma manera que la informática tradicional se compone de algoritmos que operan sobre determinadas estructuras de datos, la **técnica propia de la IA es obtener una adecuada representación simbólica del conocimiento**.

También es posible aplicar tanto los mecanismos de la **inferencia simbólica**, propios del **razonamiento deductivo**, como los mecanismos de tipo empírico que forman la base de la busca **heurística** de soluciones.

3.2. Inferencia simbólica

Los mecanismos de deducción típicos de la inferencia simbólica incluyen la **deducción** –también denominada a veces *inferencia lógicamente correcta*– amparada en las distintas reglas de la inferencia lógica:

a) El ***modus ponens***, con reglas del tipo: "si *p* entonces *q*", es decir, del hecho o premisa *p* se puede afirmar el hecho o premisa *q*.

b) La **instanciación universal**, que nos indica que si alguna cosa es cierta de todos los elementos de un conjunto, también es cierta para cada caso particular, como recuerda el conocido razonamiento: "todos los hombres son mortales, Sócrates es un hombre, por lo tanto, Sócrates es mortal".

Pero la IA también utiliza para sus razonamientos otros tipos de reglas de inferencia lógica, no tan inmediatas ni siempre seguras.

c) Un ejemplo de estos es el mecanismo "razonador" de la **inducción** por el cual, predicada una propiedad de una amplia serie de individuos, se "induce" que la propiedad puede ser predicada de todos los individuos de la misma especie.

d) Otro ejemplo de esto es una versión particular del llamado ***modus tollens*** de la lógica en la llamada abducción, que a partir de la regla "si *a* entonces *b*" y el hecho o premisa *b* se atreve a postular la posibilidad de *a*. Aunque la abducción pueda parecer una manera de proceder extraña y arriesgada, tiene su utilidad en la tentativa de busca de soluciones.

La diagnosis médica

Un ejemplo concreto de esto es la diagnosis médica, que viene a ser, en realidad, un uso prudente de la abducción, ya que, a partir de varios síntomas, el médico puede hacer una abducción de la existencia de una causa posible.

3.3. Heurística

Pero los mecanismos de la inferencia simbólica no son suficientes, y por ello los sistemas de IA utilizan muy a menudo las técnicas de la heurística.

La heurística se podría definir como el conjunto de los criterios, métodos o principios que se utilizan para encontrar, de entre varios caminos posibles, cuál o cuáles son los más efectivos para obtener un objetivo determinado.

Evidentemente, la heurística también tiene mucho que ver con los **mecanismos experimentales y empíricos**, cuya síntesis acaba elaborando estas reglas de la experiencia que se utilizan para seleccionar un camino de acción ante la explosión combinatoria de los muchos casos posibles.

Ejemplo

Investigaciones de IA han demostrado que un jugador de ajedrez experto, no analiza *todas* las jugadas posibles en un momento determinado del juego, sino tan sólo aquellas que, heurísticamente, reconoce, gracias a su experiencia, como más prometedoras o interesantes.

4. La representación del conocimiento

Uno de los temas centrales en la inteligencia artificial moderna es el de los diferentes sistemas que la IA utiliza para la representación del conocimiento e, implícitamente, las posibilidades que la mencionada representación ofrece para su utilización.

Lo que se persigue no es sólo unas estructuras de datos que proporcionen un sistema de representación efectivo y eficiente de los conocimientos, sino también qué conocimiento se ha de representar en cada utilización particular.

Los problemas anexos e importantes son:

1) La **extracción del conocimiento** de quien lo posee.

2) Su **formalización** en un determinado sistema de representación.

3) La posibilidad de crear el conocimiento mencionado, si se da el caso, o **modificar el conocimiento existente** gracias a la interacción del sistema de IA con el entorno en un mundo cambiante.

Obviamente, cada sistema de representación del conocimiento puede ir asociado a una determinada forma de razonar con el conocimiento almacenado, de manera que este conocimiento se utilice adecuadamente para obtener la solución deseada. Un problema quizá posterior, pero no menos importante, es la implementación informática de la representación del conocimiento elegida de manera que sea eficiente.

4.1. Componentes y elementos que se deben representar

En cualquier sistema de representación del conocimiento son componentes esenciales tanto las estructuras de datos que se utilizan, como sus procedimientos de interpretación y manejo.

Los distintos elementos que se han de representar son:

a) El conocimiento sobre los **objetos**.

b) El conocimiento sobre los **procesos**.

c) El conocimiento sobre el **entorno** en el que existen objetos y procesos.

Además, también se ha de implementar la representación de distintos objetivos, motivaciones, elementos de causalidad, temporalidad, etc. Uno de los elementos más difíciles de incorporar en los sistemas de IA es el aparentemente sencillo **sentido común**, que siempre supone un conocimiento difícil de determinar y precisar y que, a menudo, se acaba convirtiendo en reglas de tipo heurístico.

4.2. Etapas en el uso del conocimiento

En el uso del conocimiento se reconocen varias etapas:

a) La **adquisición**. Se trata de acumular nuevo conocimiento y relacionarlo con el que ya se tenía.

b) La **recuperación**. Se trata de obtener, de una amplia base de conocimientos, precisamente aquel que sea verdaderamente relevante para resolver el problema en curso.

c) El **razonamiento**. Se trata de inferir alguna cosa, a partir del conocimiento ya seleccionado, para obtener y verificar hechos nuevos diferentes de los ya conocidos.

4.3. Propiedades de un buen formalismo de representación del conocimiento

A un sistema de representación del conocimiento se le suelen exigir varias propiedades como:

a) La **adecuación representacional** o posibilidad de representar todas las clases de conocimiento que sean necesarias en un dominio dado.

b) La **adecuación inferencial** o posibilidad de manipulación de las estructuras de datos que sirven para representar el conocimiento, de manera que sea posible generar nuevas estructuras que correspondan a nuevos conocimientos inferidos de los anteriores.

c) La **eficiencia inferencial** o capacidad de incorporar también un nuevo conocimiento adicional (*metaconocimiento*) que puede utilizarse para mejorar el uso de los mecanismos de inferencia y optimizar el funcionamiento del sistema de IA.

d) La **eficiencia en la adquisición del conocimiento** o facilidad de adquirir información nueva incluyendo el hecho de que, en el caso ideal, el propio sistema de IA debe ser capaz de controlar la adquisición de nuevo conocimiento o metaconocimiento.

4.4. Los espacios de estados

Uno de los primeros formalismos utilizados para la representación del conocimiento fue el denominado **espacio de estados**, que describe en sus **estados** todos los momentos y pasos tanto del problema, como de la solución. Esta solución se obtiene con unos operadores o **transiciones** que actúan transformando la situación del problema de un estado a otro dentro del espacio de estados.

Este sistema resulta adecuado cuando no se produce una explosión combinatoria del número de casos posibles, lo cual puede darse cuando se abordan universos reducidos como sucede con algunos juegos o puzzles. Pero en otros casos, como en el conocido juego del ajedrez, la explosión combinatoria de estados es tal que resulta imprescindible recurrir a reglas de tipo heurístico para simplificar la multitud de estados posibles.

4.5. Esquemas modernos de representación del conocimiento

El agotamiento de la capacidad de representación y manipulación de los espacios de estados hizo aparecer otros esquemas de representación del conocimiento.

Se suele distinguir entre los **esquemas de tipo declarativo**, como los esquemas lógicos y las redes semánticas, y los **esquemas de representación de procedimientos**, como los sistemas de representación procedural o los sistemas de producción, también denominados *sistemas expertos*. Destaca también la tendencia integradora representada por los ***frames*** y los ***scripts***.

4.5.1. Los esquemas lógicos

En los esquemas lógicos (un ejemplo sencillo de esquemas de tipo declarativo) los hechos y las reglas son fórmulas expresadas en algún sistema basado en la lógica, como sucede con el lenguaje Prolog, que utiliza la lógica de primer orden del cálculo de predicados. También existe el uso de lógicas más complejas como la lógica multievaluada, la temporal o la difusa, por mencionar ejemplos ya utilizados en la IA.

Una **fórmula lógica** es, en definitiva, una combinación de predicados, variables, conectores lógicos, cuantificadores y funciones.

La programación lógica

El ejemplo más clásico del uso de esquemas lógicos es la programación lógica implementada con el lenguaje Prolog y su uso de los mecanismos de resolución, unificación y resolución por refutación.

La ventaja de este sistema es que se trata de un procedimiento preciso, flexible, modular y bastante natural, aunque suele carecer de principios de organización y selección entre los hechos (deficiencias del metaconocimiento). También suele tratar objetos demasiado simples y resulta difícil de manipular si hay muchos datos.

4.5.2. Las redes semánticas

Las redes semánticas son unos sistemas de representación, inicialmente gráficos, que resultan muy adecuados para establecer taxonomías. Al grafo que constituye una red semántica, los **nodos** incorporan objetos o conceptos y los **arcos** son relaciones entre los nodos.

En los nodos se describen tanto **conceptos** –constantes o parámetros de la realidad que especifican objetos o abstracciones–, como **acontecimientos** –acciones que ocurren en la realidad que se está modelando– y **características** –que particularizan estados, modifican conceptos, acontecimientos y también otras características.

En los arcos se suelen distinguir los que sirven a efectos de clasificación, que relacionan un objeto con otros de su clase como sucede con los casos "miembro-de" o "instancia-de", de **agregación**, que relacionan un objeto con sus componentes con mecanismos como "parte-de", y de **generalización**, que relacionan un objeto con una clase más general con ejemplos del tipo "es-un" o "es-subconjunto-de".

Variantes evolucionadas

Algunas variantes evolucionadas a partir de las redes semánticas son las redes particionadas (como propone Hendrix), los esquemas de propagación de *markers* (propuesta de Fahlman), las jerarquías de tópicos (del tipo "es-un" o "parte-de"), las redes proposicionales (como sugiere Shapiro), etc.

La característica fundamental de las redes semánticas es su facilidad para implementar la **herencia de propiedades**, lo que permite deducir hechos nuevos derivados del hecho de que los nodos más altos en la jerarquía son también aserciones sobre los nodos más bajos.

En las figuras que podéis ver al margen se muestran dos ejemplos concretos de las muchas formas que puede adoptar una red semántica.

Formas que puede adoptar una red semántica

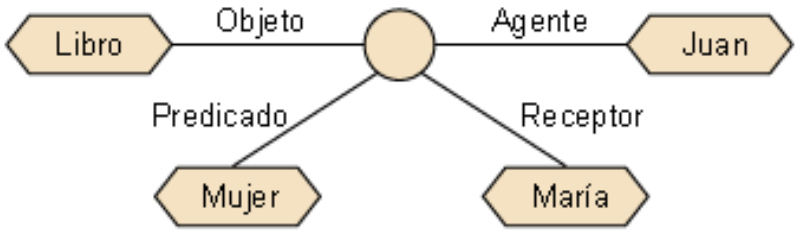

Figura 1. Juan da un libro a María

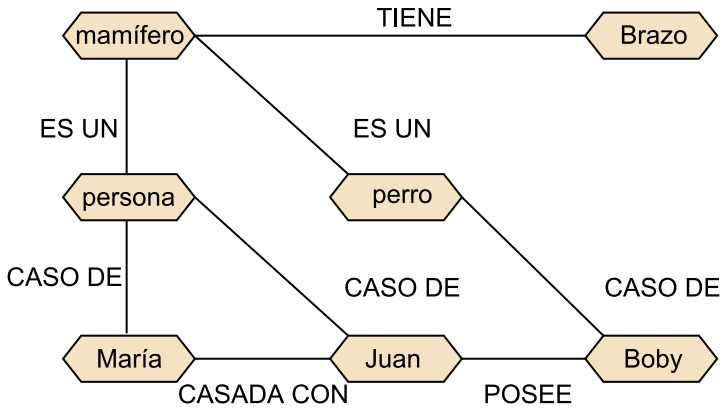

Figura 2. Una jerarquía "ES-UN"

4.5.3. Los esquemas de representación de procedimientos

Los sistemas de representación procedural se basan en la idea de almacenar el conocimiento en forma de procedimientos (*procedures*) en lugar de hacerlo en forma de proposiciones.

Pero parece que se ha obtenido mejor resultado con lo que los teóricos denominan sistemas de producción y que, más frecuentemente, reciben el nombre de sistemas expertos, que estudiaremos con más detalle en el apartado 5.

En este caso se trata de:

a) Un conjunto de reglas de producción que constituye la base de conocimientos.

b) Una base de hechos que describe el contexto de utilización del sistema experto.

c) Un intérprete o motor de inferencia.

El motor de inferencia obtiene un nuevo conocimiento por un procedimiento denominado de unificación –selección de la regla que se ha de utilizar en cada caso–, una resolución de los posibles conflictos entre reglas y una acción que decide reglas o hechos nuevos que se han de almacenar en la base de hechos.

4.5.4. Los *frames*

El ***frame***, que los expertos españoles han decidido no traducir por "marco", se puede asimilar a una red semántica compleja de estructuras que se utilizan para describir la colección de atributos que tiene un determinado objeto o una situación compleja. Compartiendo las propiedades de una red se puede esta-

Información en los frames

Propuestos por Minsky en un artículo hoy clásico, "A Framework for representing Knowledge" (1975), los *frames* incluyen en sus estructuras de datos tanto información procedural, como declarativa.

blecer una jerarquía de *frames*, también apta para establecer taxonomías tanto de pertenencia a una clase (*member link*, del tipo "Juan ES-UN hombre"), como de adscripción a otra (*subclass link*, del tipo "el hombre ES-UN mamífero").

Los *frames* disponen de **ranuras de expansión (*slots*)** como lugares físicos para insertar una parte de conocimiento o propiedad dentro de la estructura. Existen las ranuras de expansión **miembro**, que describen los atributos de cada miembro de una clase y se utilizan sólo en los *frames* que representan clases, y las ranuras de expansión propias, que describen los atributos de la clase en sí.

La ventaja de los *frames* sobre las redes semánticas es que también contienen información procedural. Esta información está en los **métodos**, que vienen a ser procedimientos escritos, por ejemplo en lenguajes como el Lisp, y que se almacenan como valores de las ranuras de expansión.

También utilizan los **valores activos (*demons*)** como procedimientos o reglas de producción ligados a una ranura de expansión y que se utilizan cuando cambia la información que contiene, tanto a la hora de añadir nueva como a la de eliminar o modificar la que ya contiene.

Además, los *frames* incorporan métodos de inferencia como la herencia de propiedades, propia de las redes semánticas y también el razonamiento mediante *facets*, que son restricciones sobre el número de valores posibles que puede tener un atributo y sobre las clases a las que puede pertenecer cada valor.

4.5.5. Los *scripts*

Un último ejemplo de sistema de representación mixta es el de los ***scripts***. Se trata de estructuras especializadas que describen secuencias de acontecimientos en un contexto particular. Los componentes principales de un *script* son:

a) Las **condiciones de entrada** que se han de satisfacer antes de que los acontecimientos descritos en el *script* se puedan iniciar.

b) El **resultado** o las condiciones que sean ciertas cuando los acontecimientos descritos se hayan cumplido.

c) Las **propiedades** (*prop*), que son ranuras de expansión que representan los objetos involucrados en los acontecimientos que describe el *script*.

d) Los **personajes** (*roles*), que son las diferentes entidades o personas que intervienen en los acontecimientos descritos.

e) La **instancia** (*track*), que es una variación específica o un caso particular de un modelo más general que se ha representado mediante este *script*.

f) Las **escenas**, que son las diferentes secuencias de acontecimientos que tienen lugar en el *script*.

5. Sistemas expertos

5.1. Definición y características

Los sistemas expertos, también denominados sistemas basados en el conocimiento, representan uno de los éxitos más importantes de la IA, al menos con respecto al éxito de sus aplicaciones en el ámbito de los sistemas de información.

La finalidad principal de los sistemas expertos es la reproducción correcta del comportamiento de un experto humano en su dominio de competencia.

Los sistemas expertos

Son uno de los ejemplos más claros de la evolución de la IA y de su éxito cuando ésta ha prescindido de los objetivos exageradamente ambiciosos de los primeros años, y ha delimitado el dominio y las características de los problemas que se han de resolver.

Algún especialista ha querido etiquetar los sistemas expertos como "la IA con éxito", algo que es, evidentemente, una exageración, pero que indica la importancia de los sistemas expertos en la superación de los "años difíciles" de la IA, a principios de los años sesenta, con varios proyectos en curso y un número excesivo de fracasos.

5.1.1. Atributos de un experto humano

Un experto humano es aceptado como tal por una serie de atributos que caracterizan su tarea, en la cual se le reconoce una verdadera competencia, habilidad y experiencia. En general, el experto humano que aquí nos interesa considerar resuelve problemas complejos, actualiza sus conocimientos y es capaz de justificar sus decisiones y respuestas, a las cuales también puede atribuir grados de credibilidad.

El buen experto humano también es capaz de destacar los aspectos más relevantes de un problema, a la vez que detecta y reconoce la desaparición de su competencia o pericia cuando el problema en cuestión se acerca a las fronteras del dominio de su experiencia, es decir, es consciente de los límites de su competencia.

Asimismo es capaz de aprender a razonar mejor, reestructurando sus conocimientos, aplicando procedimientos o reglas para los casos de excepción y elaborando nuevas reglas o procedimientos para enfrentarse a nuevos problemas. Por si no es suficiente con todo eso, además dispone de sentido común para situar su manera de proceder dentro de un ámbito más general.

5.1.2. Definición y características de un sistema experto

Un sistema experto es un sistema elaborado con técnicas de IA que, de la misma manera que el experto humano a quien intenta emular, resuelve problemas complejos y difíciles que se circunscriben a un dominio específico y delimitado.

El sistema experto utiliza procesos que imitan el razonamiento humano (deducción, inducción, estrategias de busca de soluciones, etc.) a la hora de resolver los problemas. Y todo esto lo consigue utilizando los conocimientos de base, suministrados en origen por un experto humano, a los que también incorpora los conocimientos que el sistema experto informático "aprende" durante su actividad como "experto".

Además, el sistema experto ha de ser capaz de justificar las decisiones y los resultados obtenidos y atribuirles grados de credibilidad. También, como hacen a veces a los expertos humanos, ha de ser capaz de razonar a partir de datos inciertos.

5.2. Estructura básica de un sistema experto

En la figura siguiente se muestra la estructura básica de un sistema experto en la que se percibe claramente un aspecto fundamental: la separación de los datos que forman el conocimiento, de las estrategias de resolución o procedimiento con las que se elaboran las soluciones.

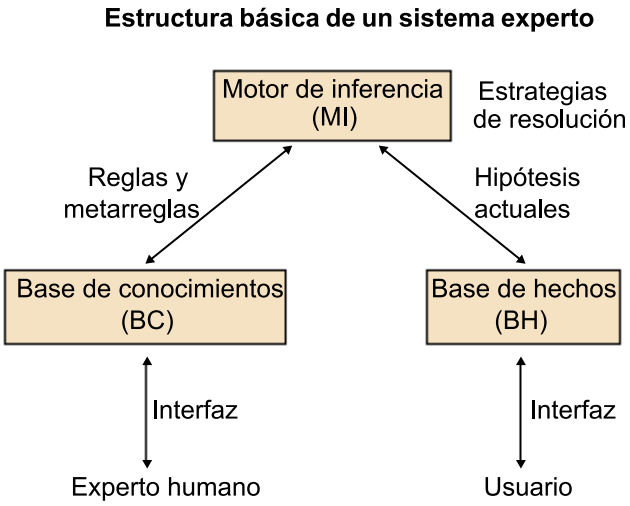

Figura 3

Los conocimientos se almacenan en la **base de conocimientos** y en la **base de hechos**, y los procedimientos capaces de razonar se implementan en el **motor de inferencia**.

El esquema también muestra la necesidad de interfaces que permitan el acceso al sistema, tanto del experto humano que alimenta la base de conocimientos como del usuario del sistema experto que proporciona los hechos que determinan una utilización concreta del sistema experto. El usuario también obtiene las respuestas que proporciona el sistema.

5.2.1. La base de conocimientos

La base de conocimientos viene a ser la **memoria a largo plazo** del sistema experto y contiene las **reglas** y las **metarreglas** que resumen el conocimiento del experto humano sobre el dominio del problema. Las metarreglas indican el orden o la preferencia con la que se deben usar las reglas que describen el conocimiento del experto en cada caso.

Una de las formas habituales (y también más simples) es la utilización de sistemas sencillos de representación del conocimiento, como son las **reglas lógicas** del tipo "si ... entonces...".

Las reglas son actualizables (y utilizables) de una en una, e independientes de su uso posterior.

Ejemplo 1

SI (la persona-X ES-UN hombre)

Y (la persona-Y ES-HIJO de la persona-X)
Y (la persona-Z ES-HIJO de la persona-Y)

ENTONCES (la persona X ES-ABUELO de la persona-Z).

Ejemplo 2

SI (el país-X y el país-Y SON-VECINOS)

ENTONCES (el color del país-X ES-DIFERENTE del color del país-Y).

5.2.2. La base de hechos

La base de hechos, también denominada a veces **memoria a corto plazo** o **memoria de trabajo**, almacena tanto los hechos proporcionados por el usuario, y que describen la situación concreta analizada, como los hechos nuevos que el propio sistema experto va obteniendo.

Ejemplo 1

Juan ES-UN hombre

Y ES-HIJO de Carlos.

Ejemplo 2

El amarillo ES-UN color.

5.2.3. El motor de inferencia

El motor de inferencia contiene el **mecanismo de razonamiento** que sigue el sistema experto: deducción, inducción, estrategias de busca de soluciones, etc.

Con ello, el motor de inferencia, al aplicar las reglas contenidas en la base de conocimientos sobre la base de hechos, obtiene nuevo conocimiento que se incorpora también a la base de hechos y es objeto, a su vez, de la aplicación de las reglas recogidas en la base de conocimientos.

Uno de los problemas centrales en el funcionamiento del motor de inferencia cuando las bases de conocimientos y de hechos son voluminosas es la selección adecuada de las reglas y los hechos que se han de considerar en cada paso del "razonamiento". En este aspecto es fundamental la existencia de **metarreglas** específicas que orienten precisamente la selección mencionada.

5.2.4. Las interfaces

Un elemento esencial en la facilidad de uso del sistema experto son las interfaces de entrada de datos que permiten la interacción, tanto del experto como del usuario, con el sistema experto. También existe la posibilidad, no recogida en el esquema de la figura 4 que repetimos al margen, de incorporar una interfaz, realizada con los sistemas de la informática tradicional, que permita la extracción de datos de la base de datos de gestión para alimentar parte de la base de hechos. En las interfaces con los usuarios humanos también es habitual buscar las formas más amigables del diálogo persona-máquina.

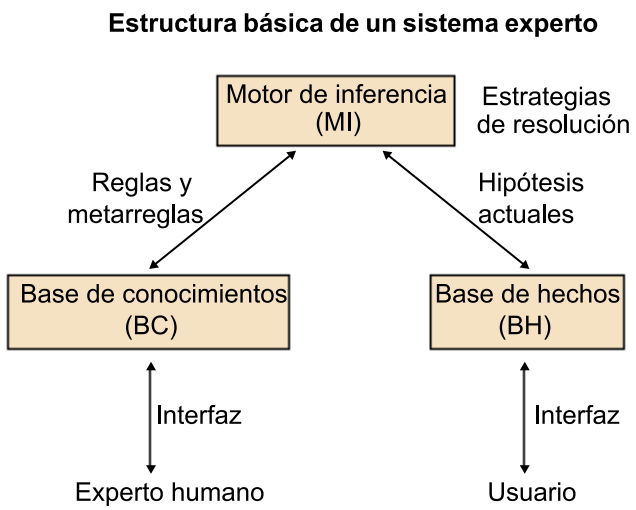

Figura 4

La figura 4, al margen, muestra el entorno de desarrollo en el que el experto humano, quizá ayudado por un ingeniero del conocimiento, dialoga con el sistema para crear la base de conocimientos.

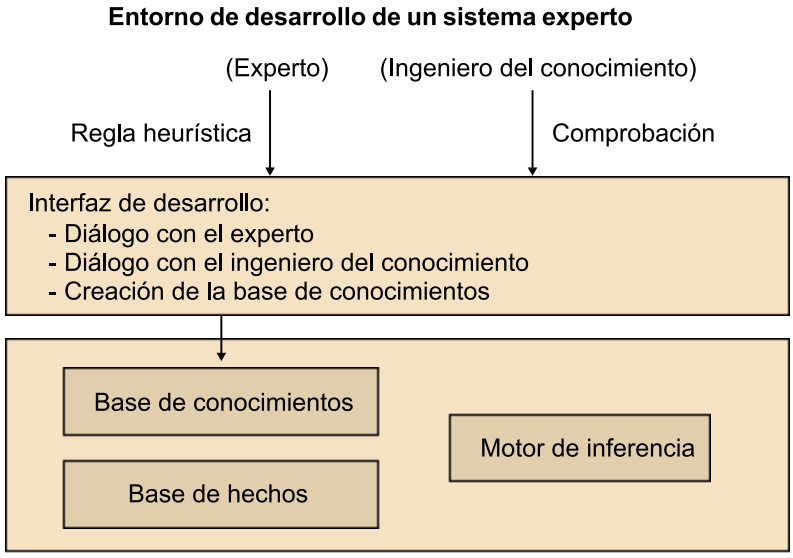

Figura 5

Asimismo, la figura 5, situada también al margen, muestra un posible entorno de utilización del sistema experto, en el cual un usuario dialoga con el sistema para crear la base de hechos e interrogar al sistema experto para obtener la justificación de las recomendaciones o decisiones de éste (proporcionadas por medio de la actividad "razonadora" y "explicadora" del motor de inferencia).

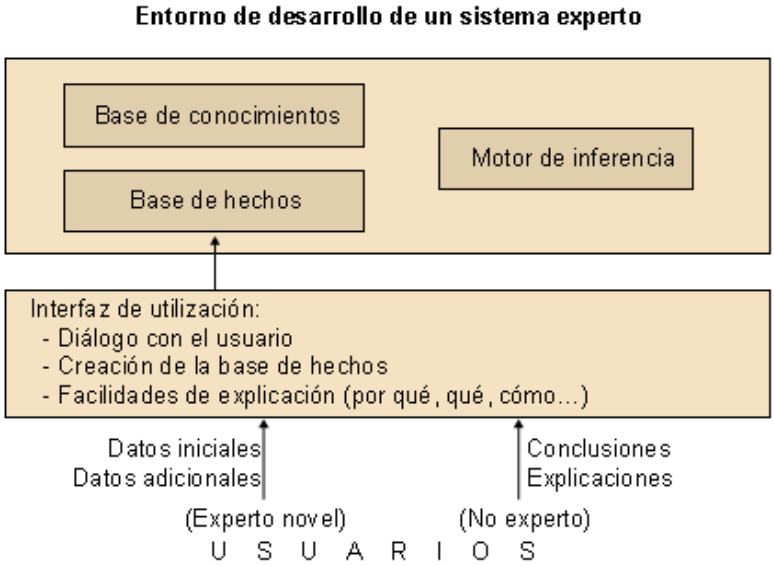

Figura 6

5.3. Bases de conocimientos: tipos de reglas

Incluso en el caso sencillo de conocimientos almacenados en forma de reglas simples de la lógica elemental, hay que considerar casos como los siguientes.

5.3.1. Reglas antecedente-consecuencia

Son del tipo *si todos los antecedentes son ciertos, entonces todas las consecuencias son ciertas* y responden al esquema general:

```
SI          <antecedente-1>
            <antecedente-2>
            ...
ENTONCES    <consecuencia-1>
            <consecuencia-2>
```

de lo que, en un sistema experto de diagnóstico médico, se obtendría el ejemplo siguiente:

```
SI          (paciente TIENE síntoma-1)
            Y (paciente TIENE síntoma-5)
            Y (paciente TIENE síntoma-7)
ENTONCES    (paciente TIENE enfermedad-A)
```

5.3.2. Reglas condición-acción

Son del tipo *si todas las condiciones son ciertas, entonces se deben ejecutar todas las acciones* y responden al esquema general:

```
SI          <condición-1>
            <condición-2>
            ...
ENTONCES    <acción-1>
            <acción-2>
            ...
```

de lo que, en un sistema experto para jugar al ajedrez, se obtendría el ejemplo siguiente:

```
SI          (torre-negra A B2)
            Y (reina-blanca A D2)
            Y (C2 ES LIBRE)
            Y (MUEVEN NEGRAS)
ENTONCES
            BORRAR:  (torre-negra A B2)
            BORRAR:  (reina-blanca A D2)
            BORRAR:  (MUEVEN-NEGRAS)
            INSERTAR:(torre-negra A D2)
            INSERTAR:(MUEVEN-BLANCAS)
```

5.3.3. Reglas inexactas

Son del tipo: *si todos los antecedentes son ciertos, entonces la consecuencia es cierta con un determinado grado de certeza* y responden al esquema general:

```
SI              <antecedente-1>
                <antecedente-2>
                ...
ENTONCES        <consecuencia-1>
CON UNA CERTIDUMBRE DEL  <factor-de-certidumbre>
```

de lo que, en un sistema experto de diagnóstico médico, se obtendría el ejemplo siguiente:

```
SI              (paciente TIENE síntoma-2)
                Y (paciente TIENE síntoma-5)
                Y (paciente TIENE síntoma-8)
ENTONCES        (paciente TIENE enfermedad-B)
CON UNA CERTIDUMBRE DEL: 0,7
```

5.4. Motor de inferencia: funcionamiento y mecanismos de razonamiento

El motor de inferencia combina los hechos y las reglas para obtener nuevos hechos.

El funcionamiento típico de un motor de inferencia se ilustra en el ejemplo siguiente:

```
HECHO:   (paciente TIENE síntoma-3)
HECHO:   (paciente TIENE síntoma-7)
REGLA:   SI (paciente TIENE síntoma-3)
         Y (paciente TIENE síntoma-7)
         ENTONCES (paciente TIENE enfermedad-C)
```

de lo que el motor de inferencia puede obtener la conclusión siguiente:

```
HECHO INFERIDO: (paciente TIENE enfermedad-C)
```

5.4.1. Razonamiento por encadenamiento hacia delante

Una de las estrategias generales a disposición del motor de inferencia es el llamado encadenamiento hacia delante (*forward chaining*).

Se trata de un procedimiento gobernado por los datos (*data driven*) en el que se parte de los hechos conocidos y, con el uso de las reglas, se van deduciendo hechos nuevos.

Razonamiento por encadenamiento hacia delante (*forward chaining*)

En el ejemplo de la figura 7, los hechos iniciales (C y F) permiten obtener el nuevo hecho E al aplicar la regla 1 (*si* C *entonces* E). Una vez se dispone del hecho E, su combinación con uno de los hechos iniciales (F) permite que la regla 3 (*si* E y F *entonces* D) obtenga un nuevo hecho D en la base de hechos.

Ahora la aplicación de la regla 2 (*si* D *entonces* B) permite obtener un nuevo hecho B. Y finalmente la aplicación de la regla 4 (*si* B y C *entonces* A) permite completar la base de hechos con el nuevo descubrimiento A.

La figura 7 muestra en su parte derecha inferior la evolución de la memoria de trabajo (base de hechos) y las conclusiones finales obtenidas.

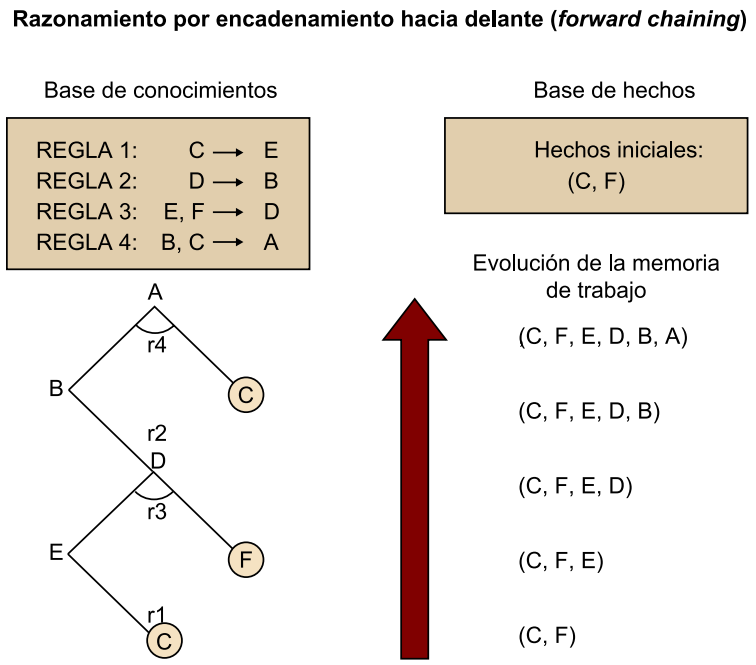

Figura 7

5.4.2. Razonamiento por encadenamiento hacia atrás

Otra estrategia habitual es el encadenamiento hacia atrás (*backward chaining*).

En el caso del encadenamiento hacia atrás (*backward chaining*), se trata de un procedimiento de busca dirigida por el intento de afirmar un objetivo concreto (*goal directed*) que se intenta probar de acuerdo con los hechos y las reglas disponibles.

Razonamiento por encadenamiento hacia atrás (*backward chaining*)

En el ejemplo de la figura 8, al intentar deducir si el objetivo A es cierto, la regla 4 (*si* B *y* C *entonces* A) permite sustituir el objetivo inicial A por el conjunto B y C.

Como C ya existe en la base de hechos, el objetivo A se cumplirá si se cumple el objetivo B. Con esto B se convierte en el nuevo objetivo que se debe verificar.

Para verificar B, la regla 2 (*si* D *entonces* B) permite sustituir el objetivo B por un nuevo objetivo D. Ahora la regla 3 (*si* E *y* F *entonces* D) dice que D se cumple si lo hacen E y F, y como F ya forma parte de los datos iniciales de la base de hechos, permite sustituir el objetivo D por un nuevo objetivo E.

Finalmente, la regla 1 (*si* C *entonces* E) hace ver que E se cumple si se cumple C. Como C forma parte de la base de hechos inicial, se cumple y, consiguientemente, también se cumplen todos los otros objetivos parciales (D y B) hasta llegar al objetivo inicial A, que queda así comprobado.

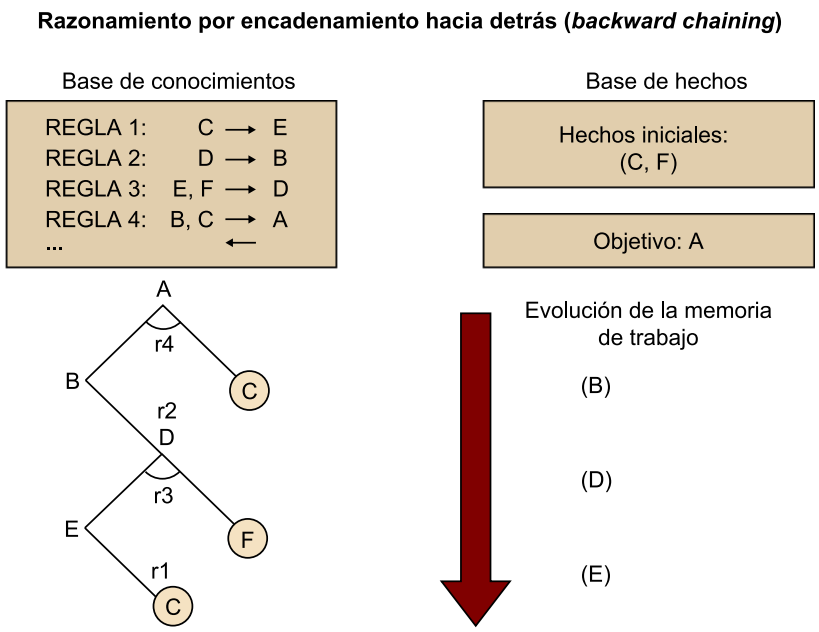

Figura 8

5.4.3. Razonamiento inexacto

Cuando las reglas pueden ser inexactas, el motor de inferencia ha de saber tratar con los factores de certeza (FC) para componer el factor de certeza de la solución propuesta. Esto se realiza utilizando reglas derivadas de la lógica difusa (*fuzzy logic*) que combinan tanto las leyes de la lógica, como las de la probabilidad.

Forma de composición de factores de certeza

En el ejemplo siguiente se muestra una posible forma de composición (producto) de factores de certeza (FC-entrada · FC-regla):

HECHO: A [FC = 0,8]
HECHO: B [FC = 0,7]
REGLA: Si A y B, entonces C [FC = 0,5]
Certidumbre de la entrada: mínimo 0,8 • 0,7 = 0,7
Certidumbre de la salida: 0,7 • 0,5 = 0,35

que permite establecer que C es cierto con un 35% de probabilidad, a partir del 80% de probabilidad asignada a la presencia del hecho A y el 70% de probabilidad asignada al hecho B. La entrada tiene, por lo tanto, una certeza del 70%, mientras que la regla aplicada sólo es cierta en el 50% de los casos. Por ello la certeza final queda reducida al 35%.

5.5. Ingeniería del conocimiento

Los sistemas expertos han llevado a la aparición de una nueva especialidad en la informática que se suele etiquetar de "ingeniería del conocimiento". De la misma manera que un administrador de bases de datos define la base de datos con un lenguaje de definición de datos (DDL), el ingeniero del conocimiento ayuda a definir la base de conocimientos con un sistema adecuado de representación del conocimiento. El paralelismo con el mundo de las bases de datos se completa con la figura del usuario del sistema experto, que actúa en cierto

modo como el usuario de un sistema de bases de datos que puede obtener, modificar y añadir nuevos datos puntuales con un lenguaje de manipulación de datos (DML). Igualmente, el usuario de un sistema experto añade nuevos hechos a la base de hechos y obtiene nuevos hechos deducidos de los que ya existen en la base.

La actividad del ingeniero del conocimiento supone la interacción con el experto humano, que es quien proporciona realmente el conocimiento que el ingeniero del conocimiento estructurará y ayude a introducir en la base de conocimientos. En algunos casos, si la interfaz de desarrollo es bastante amigable (*user friendly*): una vez decidido el formalismo de representación del conocimiento, el experto humano es quien alimenta directamente la base de conocimientos, aunque generalmente lo hace con la ayuda y supervisión del ingeniero del conocimiento.

5.5.1. Funciones del ingeniero del conocimiento

La actividad del ingeniero del conocimiento lo lleva a:

1) Descomponer un problema en subproblemas discretos.

2) Diseñar un esquema de representación del conocimiento para los hechos, las reglas y el dominio que se han de considerar.

3) Caracterizar las situaciones y las conclusiones del problema en términos del esquema de representación del conocimiento elegido.

4) Definir las reglas que permiten alcanzar soluciones cuando una situación particular llega a ser reconocida.

5) Seleccionar los estilos de razonamiento que se deben utilizar (encadenamiento hacia delante o hacia atrás, razonamiento inexacto, etc.).

6) Especificar la interfaz de utilización que se utilizará durante las actualizaciones y consultas que realice el usuario del sistema experto.

Para ello, el ingeniero del conocimiento ha de disponer de competencias propias de la **ingeniería de software**, es decir, la arquitectura de software de los sistemas expertos, los lenguajes que utiliza la IA (Lisp, Prolog, etc.) y las herramientas disponibles para la construcción del sistema experto.

El ingeniero del conocimiento también debe ser competente en lo que se denomina **ingeniería del conocimiento**, que implica básicamente conocer los sistemas de representación del conocimiento y las técnicas adecuadas, incluso de tipo psicológico, para extraer el mencionado conocimiento de los expertos humanos.

5.5.2. *Shells* y conjuntos de herramientas (*toolkits*)

Para el desarrollo efectivo de sistemas expertos es posible utilizar un **paquete adaptable**, es decir, una aplicación escrita previamente a la que realizar unos ajustes mínimos para adaptarla a unas necesidades concretas. En el extremo opuesto es posible un **desarrollo específico** utilizando los lenguajes propios de la IA (Lisp y Prolog generalmente) para construir un sistema con todos sus elementos y detalles: motor de inferencia, base de conocimientos e interfaces.

La dificultad y el coste del desarrollo específico se pueden evitar recurriendo a los paquetes generadores de sistemas expertos que se conocen habitualmente como *shells* o conjuntos de herramientas (*toolkits*).

En este caso se trata de un subsistema que incluye un motor de inferencia ya elaborado y, a menudo, un único esquema de representación del conocimiento, también prefijado. La construcción del sistema experto se reduce a obtener el conocimiento del experto humano –tarea que no es fácil ni banal– e introducirlo en el sistema gracias a interfaces de desarrollo también preexistentes y proporcionadas por la *shell* o conjunto de herramientas.

En la selección de una *shell* o conjunto de herramientas es importante conocer el esquema de representación del conocimiento que utiliza (reglas, *frames*, etc.), el mecanismo de razonamiento que incorpora su motor de inferencia (encadenamiento hacia delante o hacia atrás), el hardware en el que la *shell* está implementada (ordenadores personales, máquinas Lisp, miniordenadores, grandes sistemas, etc.) y el carácter abierto o cerrado de su arquitectura.

Otros elementos de interés son las facilidades de explicación de las decisiones obtenidas por el sistema experto, la posibilidad o no de razonamiento inexacto, el carácter de intérprete o compilador de la *shell*, principalmente en el momento de la adquisición del conocimiento, etc.

5.6. Viabilidad y beneficios de un sistema experto

Los sistemas expertos intentan emular y reemplazar a los expertos humanos y por ello son adecuados para su implantación en las organizaciones y situaciones en las que haya escasez de expertos o éstos resulten sometidos a las consultas de un amplio conjunto de usuarios muy dispersos geográficamente. También pueden suplir al experto humano en los casos en los que la formación de un experto nuevo sea muy costosa y el uso de su experiencia no sea bastante frecuente para justificarla.

Otras causas que aconsejan recurrir a la implementación de un sistema experto suelen ser el riesgo de la pérdida del experto humano y la existencia de un dominio en evolución constante.

Con respecto al problema abordado, deberá ser un problema que no tenga una solución algorítmica directa –si la tuviera, se trataría con las técnicas de la informática tradicional– y que requiera tareas deductivas e inductivas al lado de mecanismos heurísticos, en un dominio sobre el cual los conocimientos sean de tipo predominantemente cualitativo y simbólico, a menudo incompletos e imprecisos e, incluso, a veces, aparentemente contradictorios.

5.6.1. Requisitos de viabilidad de un sistema experto

Para que el sistema experto sea viable y pueda ofrecer resultados también es necesario que:

a) El dominio en el que actúa sea restringido y perfectamente definido.

b) Los casos (hechos, reglas y soluciones) sean claramente formalizables.

c) Las soluciones con un cierto grado de incertidumbre sean aceptables.

d) El problema sea de una complejidad "razonable" y su resolución por el experto no sea ni demasiado corta (caso banal) ni demasiado larga (caso demasiado complejo), lo cual se suele concretar, por ejemplo, en un número de reglas inferior al millar y una profundidad de inferencia de entre 2 y 50.

e) Existe disponibilidad de un experto humano, si es posible motivado, y con una cierta capacidad de formalización, ya que, en definitiva, es él quien proporciona el conocimiento.

> Esta disponibilidad de un experto humano motivado es un requisito primordial. Se puede decir que no puede haber un sistema experto sin un buen experto humano a quien emular.

5.6.2. Beneficios que se han de obtener de un sistema experto

Un buen sistema experto puede proporcionar muchos beneficios tanto para el experto, al cual teóricamente sustituye, como para los usuarios y las organizaciones que lo tienen a su disposición.

Para el **experto humano**, un sistema experto puede ser una herramienta de gran utilidad con respecto a formalización, estructuración y validación de sus propios conocimientos, ya que puede ver cómo estos conocimientos se aplican

a casos menos frecuentes y, quizá, con resultados insospechados. Gracias al sistema experto, el experto humano puede quedar reservado para las consultas menos rutinarias.

Para el **usuario**, un sistema experto implica la disponibilidad inmediata de la competencia del experto; una interactividad posiblemente más fácil si la interfaz de utilización está bien diseñada y se puede disponer de una justificación adecuada, tanto de las respuestas que obtiene del sistema experto como de las preguntas que el mismo sistema puede ir efectuando.

Para las **organizaciones** o empresas que utilizan un sistema experto, hay varias mejoras evidentes: la posible disponibilidad de un experto en cada punto de consulta, la perpetuación de la habilidad y competencia del experto y un aumento del nivel general de competencia de la organización.

También existe más facilidad para la formación de nuevos expertos humanos, que se pueden formar utilizando el sistema experto y comparando las recomendaciones con las que ellos mismos habrían realizado.

5.7. Problemas en el uso de sistemas expertos

Actualmente la realización de sistemas expertos todavía presenta serias limitaciones y problemas entre los cuales destacan:

a) La dificultad en la adquisición de los conocimientos obtenidos del experto humano, no siempre suficientemente colaborador y motivado.

b) Una aplicación efectiva que continúa siendo efectiva sólo en dominios todavía muy restringidos.

c) Un comportamiento muy frágil cerca de las fronteras del dominio, y una gran dificultad para que el sistema experto detecte el desbordamiento de su ámbito de competencia.

d) La obtención de explicaciones no siempre relevantes.

e) Las limitaciones serias por razones de eficiencia que obligan a restringir el número de reglas.

f) Las limitaciones implícitas a la potencialidad de las técnicas actuales de representación del conocimiento.

g) La escasa capacidad de los sistemas expertos actuales para el aprendizaje, la generalización y el razonamiento por analogía.

También es un problema la dificultad adicional de la integración de los sistemas expertos, y en general de los sistemas derivados de la IA, con la informática tradicional, hasta ahora gestora y depositaria de los "hechos".

5.8. Algunos sistemas expertos famosos

En este apartado mencionaremos algunos ejemplos concretos de sistemas expertos famosos:

- MYCIN.
- MACSYMA.
- PROSPECTOR.
- DENDRAL.
- XCON.

5.8.1. MYCIN

El MYCIN fue uno de los primeros sistemas expertos aparecido a mediados de los años setenta.

Lo desarrolló Shortliffe en la Universidad de Stanford y se ocupa del **diagnóstico de infecciones de la sangre y su terapia**. Utiliza reglas de tipo lógico, su motor de inferencia es por encadenamiento hacia atrás (*backward chaining*) y es capaz de utilizar factores de certeza.

Posteriormente se completó con **TEIRESIAS**, que es una versión "explicativa" del MYCIN que justifica el diagnóstico y la terapia recomendada.

5.8.2. MACSYMA

Derivado tardío del SAINT de Slage, desarrollado en el MIT a mediados de los años cincuenta, el MACSYMA, documentado en 1977, trata el **cálculo diferencial e integral** con un motor de inferencia compuesto de una serie de funciones implementadas directamente en Lisp.

5.8.3. PROSPECTOR

Documentado por **Hart** (1978) y **Duda** (1979), se ocupa de la **prospección y evaluación de yacimientos de minerales**, particularmente de cobre y uranio. Utiliza la inferencia probabilística.

5.8.4. DENDRAL

Desarrollado en la Universidad de Stanford bajo la dirección de Feigenbaum, es capaz de **analizar la estructura molecular de un compuesto a partir del espectograma de masas y otros datos**. Su motor de inferencia usa el mecanismo de *generate and test*, que consiste en generar casos posibles y probar si cumplen todas las condiciones.

5.8.5. XCON

El XCON es una versión operativa de un sistema anterior denominado *R1* y, posiblemente, el primer sistema experto utilizado en el ámbito comercial. Se utilizó, en la firma DEC, **para la configuración de los sistemas DEC/VAX** y lo desarrolló **J. McDermott** de la Universidad de Carnegie-Mellon. Utiliza un motor de inferencia con encadenamiento hacia delante (*forward chaining*) y es operativo desde 1981.

5.8.6. Otras aplicaciones

El campo de aplicación posible de los sistemas expertos es realmente amplio. A modo de ejemplos no exhaustivos hay que mencionar la posible utilización en banca (selección de productos para clientes, análisis de riesgos y balances, reglas de auditoría e inspección, gestión de carteras, etc.), diseño (configuración de sistemas, diseño de circuitos etc.), diagnóstico de situación (diagnóstico médico, de averías de máquinas, del funcionamiento de una organización, etc.), enseñanza, entre otras aplicaciones.

6. Redes neuronales

En la segunda mitad de la década de los ochenta se produjo la reactivación de una tecnología de inteligencia artificial ya estudiada en los años cincuenta y sesenta que había estado prácticamente abandonada a finales de los años sesenta.

Como efecto de la atención creciente al proceso en paralelo y del reciente descubrimiento de algoritmos de aprendizaje nuevos para las simulaciones de sistemas de neuronas, se vuelve a confiar en uno de los primeros campos de estudio de la inteligencia artificial, al cual se hace referencia con la expresión *redes neuronales*.

Hoy día las **redes neuronales** constituyen una tecnología "renacida", todavía en desarrollo, en la cual se depositan grandes esperanzas a la hora de tratar con éxito algunos de los problemas clásicos de la inteligencia artificial, en particular el del reconocimiento de formas y de la palabra hablada.

6.1. Descripción

A continuación describiremos algunos aspectos más destacados de las redes neuronales.

Explicaremos cuáles son los modelos de redes neuronales, cuáles son sus características, qué aplicaciones tienen, así como algunas de sus particularidades.

6.1.1. Modelos de redes neuronales artificiales

Una **red neuronal** está constituida por un número variable de procesadores interconectados entre sí y que realizan una transferencia mutua de valores denominados *activaciones*.

Cada procesador recibe una serie de activaciones (**activaciones de entrada**) y, a partir de ellas, genera un valor de salida (**activación de salida**) que, a su vez, transfiere a otro grupo de procesadores que tiene conectados.

Por analogía con las redes biológicas, se da el nombre de sinapsis a las interconexiones y **neuronas** o **unidades** a los procesadores que forman los nodos de la red de procesadores, que recibe el nombre de **red neuronal**.

Esta descripción funcional se aplica tanto a las redes neuronales biológicas, como a su modelización eléctrica, es decir, a las **redes neuronales artificiales** (ANN, *artificial neural networks*) con las que se pretende reproducir, mediante sistemas electrónicos, la estructura y el comportamiento de las redes neuronales biológicas.

La costumbre ha llevado a omitir el calificativo "artificial" y, de manera general, se habla de redes neuronales para referirse a la modelización electrónica de las redes neuronales biológicas.

6.1.2. Algunas características de las redes neuronales

En el funcionamiento de las redes neuronales es posible distinguir la fase de ejecución y la fase de aprendizaje.

En la **fase de ejecución** cada neurona realiza una actualización de la activación de salida, de acuerdo con las activaciones de entrada que recibe en un momento dado. El funcionamiento durante esta fase se sintetiza en las ecuaciones que rigen la dinámica del sistema y el comportamiento de cada neurona en un momento dado.

Antes de usar o "ejecutar" la red neuronal, en la **fase de aprendizaje**, se cambian y actualizan los pesos que afectan las distintas activaciones de entrada con el objetivo de "enseñar" a la red neuronal a comportarse según se desea, es decir, a producir una salida determinada de acuerdo con la entrada recibida.

Es posible caracterizar las redes neuronales por:

a) La arquitectura de la red, determinada por la topología y la estructura que tiene y por el grado de conexión entre las neuronas que la forman.

b) La dinámica de la red, es decir, las ecuaciones que rigen el comportamiento de cada neurona, y en conjunto de la red, durante la fase de ejecución, es decir, cuando a la red ya se la ha "enseñado".

c) La regla, procedimiento o algoritmo de aprendizaje con el que se "enseña" a la red su cometido.

6.1.3. Otras denominaciones

Aunque el nombre más extendido es el de redes neuronales, los modelos como el indicado también han recibido otras denominaciones. A causa de la interconexión de varios procesadores simples se les ha denominado **modelos conexionistas** y, de hecho, representan el soporte de este planteamiento al campo de la inteligencia artificial.

A causa del paralelismo de la actuación conjunta de varios procesadores, también se habla de **modelos de proceso en paralelo distribuido**, como se hizo en un libro famoso de 1986, *Parallel Distributed Processing: Explorations in the microstructure of Cognition*, editado por **D. E. Rumelhart** y **J. L. McClelland**, que ha servido para la reactivación del interés sobre las redes neuronales.

Por la similitud estructural con el sistema nervioso biológico se los ha denominado también **sistemas neuromórficos** (*neuromorphic systems*) y se habla también de **cálculo neuronal** (*neural computing*) para referirse al aspecto de la informática y de la inteligencia artificial que trata de las redes neuronales y sus posibilidades.

6.1.4. Aplicaciones y funcionamiento

Las redes neuronales se utilizan principalmente en los sistemas clasificadores que han de dar respuesta a un estímulo concreto para reconocer su pertenencia o no a una clase determinada.

En concreto, parece que las redes neuronales son el mecanismo que muestra más posibilidades ante problemas como **el reconocimiento de formas** y **el reconocimiento de la palabra hablada** y, en general, ante todo tipo de problemas que requieran un alta capacidad de tratamiento efectuado en paralelo. En lugar de ejecutar una secuencia de instrucciones previamente almacenadas, como hacen los ordenadores que utilizan la arquitectura clásica debida a von Neumann, los modelos con redes neuronales exploran simultáneamente varias hipótesis utilizando sobradamente el proceso en paralelo, por lo cual también han servido para impulsar este aspecto de la arquitectura de los ordenadores.

> Se suele decir que la referencia a las redes neuronales se distingue de la informática clásica porque no se "programa" una red neuronal, sino que se "enseña" a una red neuronal a comportarse de una manera determinada.

La manera habitual de hacerlo es mostrar a la red neuronal varias señales de entrada junto con las salidas esperadas, y repetir el proceso más de una vez mientras la red neuronal "aprende" ajustando los pesos y las ponderaciones que determinan la dinámica de la red y el comportamiento de las distintas neuronas y sinapsis que la forman. El problema queda resuelto cuando se ha encontrado un conjunto de pesos que permiten a la red componer la salida adecuada para cada entrada, es decir, "reconocer" la entrada.

6.1.5. Neuroordenadores y *netware*

La modelización de una red neuronal puede tener lugar en el hardware utilizando circuitos de proceso en paralelo y, a tal efecto, se utilizan arquitecturas especiales para el proceso en paralelo como, por ejemplo, los ordenadores matriciales. También se diseñan chips VLSI que incorporan la estructura de una red neuronal. Generalmente se asigna el nombre de neuroordenadores (*neurocomputers*) a estos sistemas, y se suelen utilizar como coprocesadores especializados y auxiliares de los ordenadores digitales tradicionales. En este caso se accede a la red neuronal como si fuera una subrutina del sistema.

En el caso de los *neurocomputers*, no tiene sentido hablar de su potencia haciendo referencia a los millones de instrucciones por segundo (mips) que son característicos de los ordenadores con arquitectura von Neumann. En los neuroordenadores se habla de **interconexiones por segundo** y, por ejemplo, el sistema Neuro-07 anunciado por NEC en 1988 ya incorporaba un neuromotor (*neuro-engine*) de 210.000 interconexiones por segundo. El sistema incluía un ordenador personal convencional con pantalla de color, un neuromotor con su teclado particular y software para "enseñar" a la red neuronal, todo por un precio de poco más de un millón de pesetas (unos 6.000 euros).

El *netware* es un software que simula las neuronas y su interconexión en un ordenador convencional con arquitectura von Neumann.

En el mismo año al que hacíamos referencia, 1988, se vendieron unos 10.000 paquetes de *netware* neuronal en Estados Unidos.

La década de los años ochenta

A finales de la década de los años ochenta, datos publicados en agosto de 1989 estimaban ya en unas 300 las empresas norteamericanas especializadas en la tecnología de las redes neuronales, y varias de ellas se dedican, precisamente, a elaborar *netware* para ordenadores personales y estaciones de trabajo.

Incluso con estaciones de trabajo, y también con los ordenadores personales de la microinformática, es posible la investigación y utilización de pequeñas redes neuronales (menos de cincuenta elementos procesadores o neuronas en un PC compatible, según algunos autores).

Aunque se ha de tener en cuenta las limitaciones de proceso de la arquitectura von Neumann sometida a las exigencias de simular el proceso en paralelo de las redes neuronales, y que es importante disponer de compiladores adecuados para la simulación del proceso en paralelo, los PC resultan ser adecuado para el estudio y la investigación con redes de dimensiones reducidas.

6.1.6. Características particulares de las redes neuronales

Gracias al elevado grado de paralelismo, las redes neuronales presentan una tolerancia a errores elevada (*fault tolerance*) a causa de la redundancia de elementos procesadores y a las características de su dinámica de funcionamiento. Por ello también presentan un proceso de degradación controlado y limitado (*graceful degradation*) cuando se presenta el error de uno de los procesadores.

Además, cabe recordar que las redes neuronales no son eficientes para realizar las tareas habitualmente encomendadas a los ordenadores de proceso secuencial (cálculos con gran exactitud, manipulación de cifras y datos, etc.), pero resultan mejores para el manejo de conjuntos incompletos de datos o de información difusa o contradictoria.

6.2. Arquitectura y topología de las redes neuronales

A continuación explicaremos cuál es la estructura topológica y la arquitectura en las redes neuronales.

6.2.1. Estructura topológica

Con respecto a la arquitectura y topología de la red, se puede hablar, por ejemplo, de redes con alimentación hacia delante (*feed forward*) o de redes con retroalimentación (*feedback*). En las primeras, el sentido de propagación de las activaciones en la fase de ejecución es único (hacia delante), mientras que en las segundas, las activaciones de salida de unas neuronas se utilizan también como activaciones de entrada en un circuito de retroalimentación (podéis ver la figura 9 al margen).

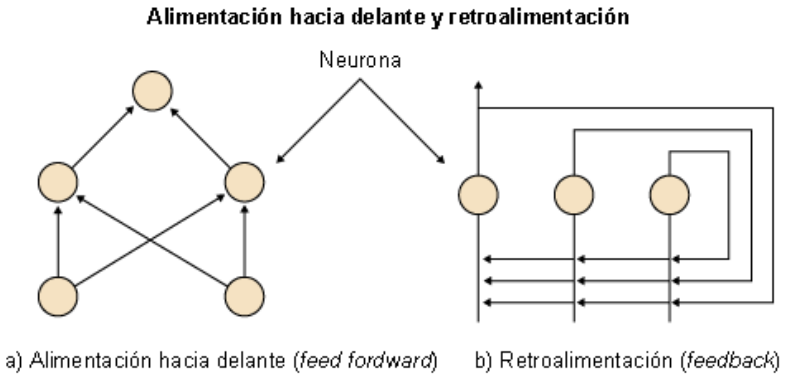

Figura 9

También se pueden diseñar redes en las cuales se forman estructuras jerárquicas entre las neuronas o utilizar cualquier otra topología.

Un caso particular de estas redes es el que se ha denominado redes de estructuras competitivas, en el cual las neuronas se agrupan en *clusters* y dentro de cada uno de ellos, tan sólo una neurona puede estar activa en un momento determinado gracias al carácter inhibidor de las sinapsis entre ellas (podéis ver la figura 10 al margen).

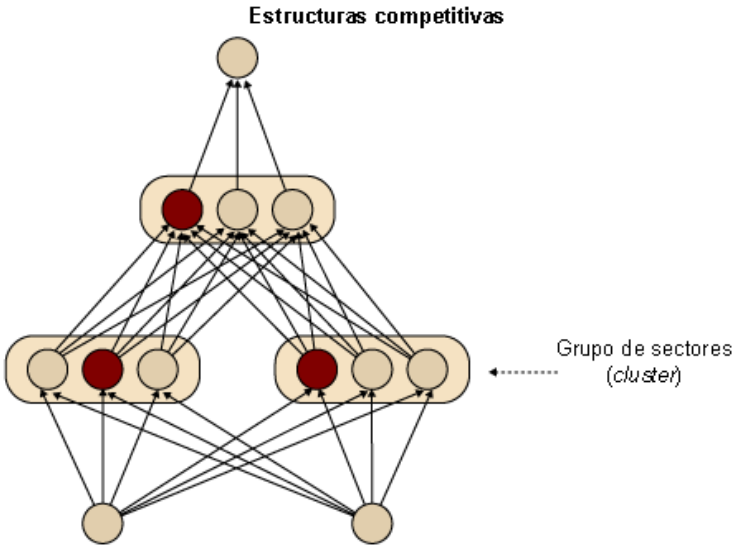

Figura 10

6.2.2. Estructura en capas de una red neuronal

En el caso más utilizado, las neuronas en una red neuronal suelen estructurarse de manera que integren una o varias "capas" o niveles (*layers*). Según los diseños, hay redes de una, dos y varias capas.

La primera capa se denomina **capa de entrada** (*input layer*) y es la que recibe los estímulos del exterior. Si hay más capas, la última de ellas es la **capa de salida** (*output layer*), de la cual se obtiene el resultado de la fase de ejecución de la red neuronal. Si hay más de dos capas, entre la primera y la última hay un conjunto de capas ocultas (*hidden layers*) formadas por neuronas que almacenan la información en una "representación interna", y por ello también reciben el nombre de unidades de representación interna (*internal representation units*) (podéis ver la figura 11 al margen).

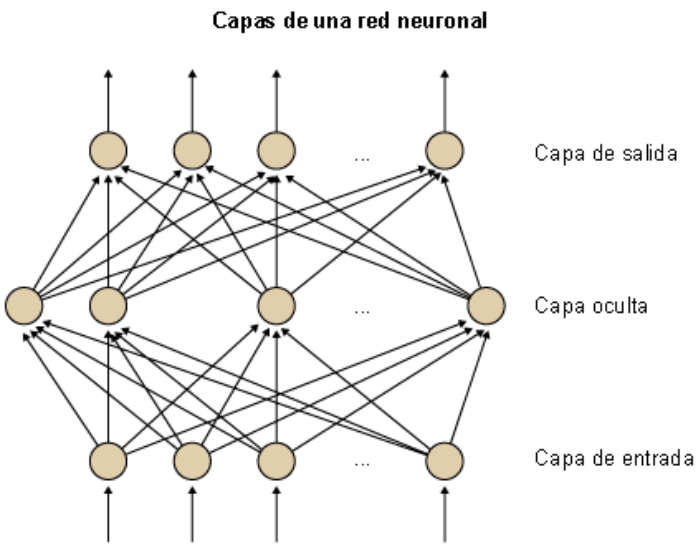

Figura 11

Las capas pueden estar constituidas por un número cualquiera de neuronas, normalmente impuesto por las características del problema que se ha de resolver o por las prestaciones que se desee obtener de la red neuronal. Así pues, en el diseño de una red neuronal se dan dos primeros grados de libertad al elegir tanto el número de capas, como el número de neuronas en cada capa.

Las dos espirales

Un ejemplo posible es el que se muestra en la figura 12, con dos neuronas de entrada, una de salida y dos capas ocultas de cinco neuronas cada una. K. Lang y M. Witbrock, de la Carnegie Mellon University, utilizaron esta red para resolver el problema conocido como "de las dos-espirales", que consiste en reconocer si un punto pertenece o no a dos espirales entrelazados. Gracias a la simplicidad del modelo, ha sido posible estudiar el comportamiento de las capas ocultas en una red neuronal y profundizar en el conocimiento de la representación interna de la información en una red con más de dos capas, uno de los problemas más interesantes y sugestivos en la investigación sobre redes neuronales.

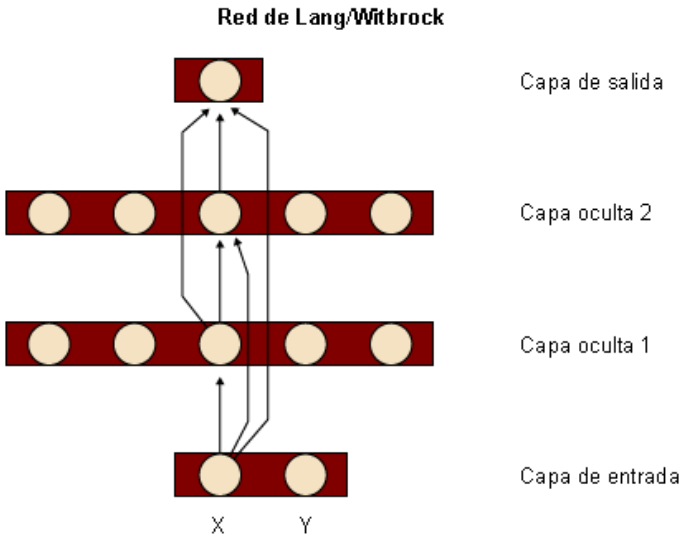

Figura 12

6.2.3. Conexiones entre las neuronas

Las sinapsis o conexiones entre las neuronas que forman las distintas capas de la red se pueden realizar de varias maneras, lo cual influye en la topología y arquitectura de la red y en su funcionamiento.

El sistema más habitual sigue el principio de la conectividad total, según el cual cada neurona perteneciente a una cierta capa está conectada con todas y cada una de las neuronas que forman la capa inmediatamente inferior y de todas ellas recibe activaciones de entrada.

Además, cada neurona de una capa está conectada también con todas y cada una de las neuronas que integran la capa inmediatamente superior, a las cuales transfiere su activación de salida. Así se mostraba al conexionado de la figura 11 que repetimos al margen, con una única capa oculta.

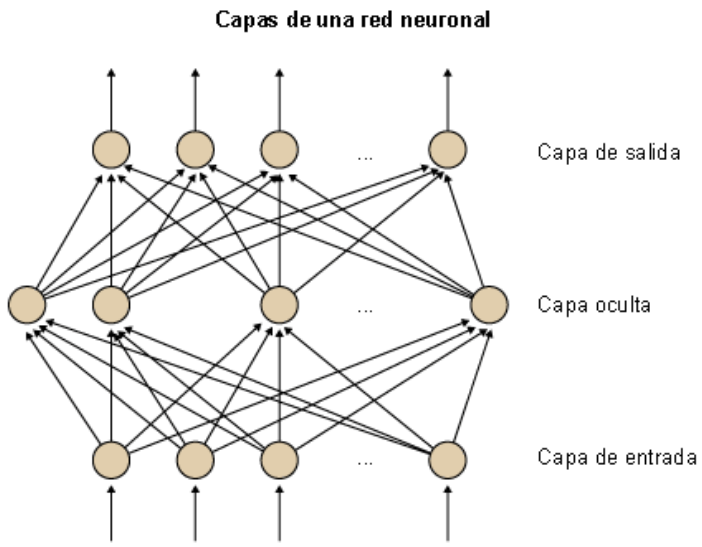

Figura 11

6.3. Dinámica y aprendizaje de una red neuronal

A continuación explicaremos la dinámica y el aprendizaje de una red neuronal en dos fases: fase de ejecución y fase de aprendizaje.

6.3.1. Fase de ejecución

En la fase de ejecución cada interconexión o sinapsis tiene un peso establecido obtenido en la fase de aprendizaje.

La dinámica de funcionamiento de la red neuronal establece que la señal de entrada en cada neurona es afectada por el peso mencionado en una combinación lineal de todas las activaciones de entrada que recibe la neurona. Una

vez efectuada esta combinación lineal, se añade al resultado una cierta función umbral característica de la neurona. Al resultado obtenido se aplica una función propia de la neurona que es la que determina finalmente la activación de salida. En el caso, hoy día frecuente, de varias capas de neuronas conectadas siguiendo el principio de la conectividad total, la fórmula que determina el comportamiento dinámico de la red neuronal es:

$$a_k(c) = f(\Sigma [W_{ki}(c) * o_k(c - 1)] + u_k(c))$$

donde:

ak(c) es la activación de salida de la neurona de la columna **k** a la capa **c**;

$\mathbf{W_{ki}(c)}$ es el peso asignado a la interconexión entre la neurona de la columna **k** de la capa **c** y la neurona de la columna **i** de la capa **c-1**;

$\mathbf{o_k(c\text{-}1)}$ es la activación de entrada para esta interconexión y es igual a $\mathbf{i_k}$ (la señal de entrada en la columna **k** de la capa de entrada) cuando la capa es la primera (c = 0), e igual a $\mathbf{a_k\ (c\text{-}1)}$ en las otras capas;

$\mathbf{u_k(c)}$ es la función umbral para la neurona de la columna **k** de la capa **c**; y finalmente,

f() es la función de activación de la neurona.

Todo esto permite esquematizar gráficamente una neurona con sus activaciones en el diagrama de la figura 13 en el margen.

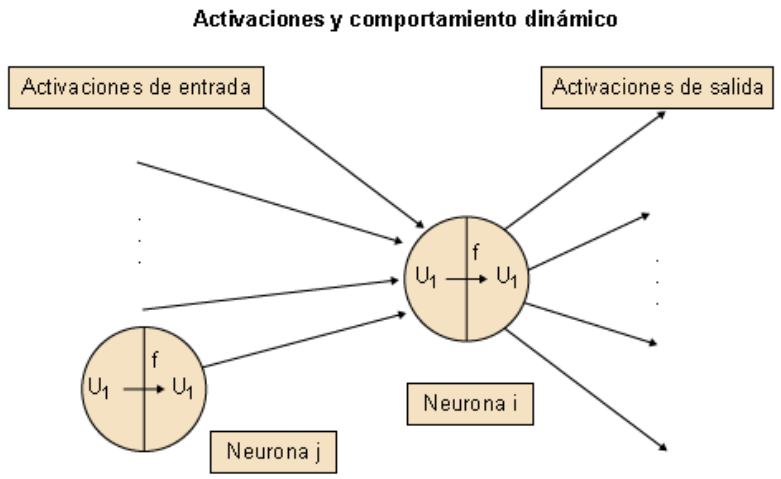

Figura 13

La función **f()**, que define el comportamiento de una neurona en una red neuronal, puede ser **determinista** o **estocástica** según si el potencial postsináptico o la activación de salida se obtenga de manera determinada de la entrada o de acuerdo con una distribución probabilística.

Las funciones deterministas pueden ser lineales como f(x) = x, pero en la mayoría de los casos son no lineales. Las funciones más utilizadas son:

a) Funciones de tipo binario si la respuesta es 0 ó 1.

b) Funciones isign o *hard delimiter* si la respuesta es +1 o –1, es decir, del tipo f(x) + sign(x).

c) Funciones sigmoides, generalmente según la tangente hiperbólica en la forma f(x) = th(x).

La fase de ejecución puede ser paralelizada fácilmente (incluso en una arquitectura von Neumann) mediante una partición del proceso de cálculo. Esto se consigue utilizando un procesador (o su simulación) para cada neurona, y el procesador mencionado se encarga de almacenar en una memoria local los pesos de ponderación asociados a la neurona que debe simular. Para ello, cada procesador toma la activación de la capa inmediatamente inferior y la multiplica por el peso correspondiente. La activación obtenida se transfiere al procesador siguiente mientras se recibe la nueva activación procedente del procesador anterior que, a su vez, se multiplica (* en la fórmula) por el peso correspondiente y se acumula (S en la fórmula) al resultado anterior. Este proceso de multiplicación y acumulación se repite hasta que se obtienen todos los productos parciales indicados en la ecuación.

6.3.2. Fase de aprendizaje

En la fase de aprendizaje se "enseña" a la red neuronal el comportamiento deseado.

Las reglas de aprendizaje pueden ser supervisadas cuando existe un "profesor" que dirige el entrenamiento de la red e indica si la respuesta al estímulo de entrada es correcta o no; o autoorganizadas cuando no hay un "profesor" externo a la red. En este último caso se habla, más frecuentemente, de una regla de aprendizaje no "supervisada".

La regla de aprendizaje

En algunos casos se prescinde del proceso de aprendizaje y se dice que la regla de aprendizaje es "instantánea". Pero, en la mayoría de los casos, la regla o el algoritmo de aprendizaje es de gran importancia.

Hay varios algoritmos o reglas para proceder a la actualización de los valores de los pesos de las interconexiones o sinapsis durante el proceso de entrenamiento de la red. Cada algoritmo da lugar a una cierta estructura y comportamiento de la red neuronal.

El problema central en la eficacia de un algoritmo de aprendizaje es determinar cuáles son las interconexiones que tienen una modificación de pesos que supone una mejora rápida de la eficiencia de la red neuronal. El problema resulta especialmente complejo cuando se trata de interconexiones en redes neuronales de varias capas y hay "capas ocultas" que mantienen una "representación interna" desconocida de la información que circula por la red neuronal. Precisamente, la difusión a finales de los años ochenta de uno de estos algoritmos de aprendizaje se considera una de las causas principales del renacimiento del interés por las redes neuronales.

El mencionado algoritmo es conocido como el de la propagación de errores hacia atrás (*back propagation of errors*), pero suele abreviarse en la literatura especializada como *back propagation* e, incluso, como *back-prop*. Se trata de un algoritmo supervisado que propuso **D. E. Rumelhart** y se puede aplicar a redes de varias capas, aunque habitualmente se utilizan redes de tres capas en la mayoría de los estudios y las investigaciones. Este algoritmo presenta, entre otras, la ventaja de permitir un amplio rango de aplicaciones, principalmente a causa del hecho de que también acepta entradas y salidas continuas.

En primer lugar se establecen al azar los valores de los pesos y se ofrece una entrada a la red neuronal que producirá actividad en cada una de las capas de la red. Un "profesor" que conoce cuál ha de ser la respuesta de cada neurona de la capa de salida indica en cada una de ellas la amplitud y el signo de su error. Las mencionadas señales de error se utilizan para ajustar los pesos de todas las interconexiones entre la capa de salida y la capa oculta inmediatamente anterior a ella, de manera que el error obtenido en la capa de salida se propaga hacia atrás, hacia las capas ocultas que, a su vez, utilizan los valores de error recibidos para ajustar el valor de sus sinapsis. Así pues, en la fase de aprendizaje el flujo de datos tiene un sentido contrario al de la fase de ejecución y se propaga "hacia atrás".

De hecho, el algoritmo consigue realizar pequeños ajustes en cada sinapsis de la red, de manera que el conjunto de las modificaciones reduce el error total en el funcionamiento de la red. Una vez se ha aplicado el mismo procedimiento varias veces, se llega a un valor mínimo del error total. Como todos los métodos de gradiente descendente, se podría tratar de un mínimo local, pero eso raramente ocurre en las redes neuronales, posiblemente por el número de neuronas involucradas y la gradualidad de su respuesta.

6.4. Historia de las redes neuronales

A continuación haremos una breve introducción a la historia de las redes neuronales, desde los primeros estudios que se hicieron sobre el tema hasta el nuevo interés que ha despertado en nuestros tiempos.

6.4.1. El *perceptron* de F. Rosenblatt

Los primeros estudios sobre redes neuronales datan de 1943, cuando **McCulloch** y **Pitts** iniciaron la interpretación mecanicista de las redes neuronales. Pero sólo a partir de los años sesenta las redes neuronales empezaron a ser consideradas como una posible solución a los problemas de realizar clasificaciones automatizadas. Para ello fue decisivo el trabajo de **Franz Rosenblatt**, un psicólogo de la Universidad Cornell que, por medio del análisis matemático, la simulación en ordenadores digitales y algunos experimentos con sistemas analógicos de proceso en paralelo, llegó a mostrar cómo se podía enseñar a actuar como clasificadores a las redes neuronales con valores variables en los pesos de las interconexiones.

El ***perceptron***, desarrollado por Rosenblatt a partir de 1958, era en realidad un clasificador de categorías aplicado a la simulación de la visión humana. Rosenblatt proponía una regla de aprendizaje que, aunque era lenta, tenía que convergir siempre que existiera una solución. En realidad, el modelo de *perceptron* con dos capas de neuronas podía computar, por ejemplo, las funciones lógicas *or* y *and*, pero no la función *xor*. La introducción de una tercera capa, una capa oculta, aportaría la potencia suficiente para implementar esta función. Pero un artículo de **Minsky** y **Papert** de 1967 demostraba que el procedimiento de convergencia de Rosenblatt sólo podía resolver problemas linealmente separables, lo que provocó que en la práctica se abandonara durante varios años esta vía de investigación.

6.4.2. El predictor meteorológico de Widrow

A pesar del descrédito que el trabajo de **Minsky** y **Papert** aportó a la teoría de los perceptrones, otro de los pioneros que trabajaba desde los años cincuenta, **Bernard Widrow**, de la Universidad de Stanford, implementó en 1963 una red neuronal que actuaba como un **sencillo sistema de predicción meteorológico** capaz de dar predicciones correctas en un 85% de los casos, frente al estimado 65% que obtenía el hombre del tiempo local. Widrow utilizaba una regla de aprendizaje con una función de "coste" que debía ser minimizada. La función de coste mencionada medía la diferencia entre la salida real de la red neuronal en su capa de salida y el valor esperado.

Pero la realidad es que, después del influyente artículo de Minsky y Papert, los perceptrones y las redes neuronales fueron prácticamente abandonadas durante los años setenta, e incluso se llegó a dudar de la posibilidad de continuar considerándolos como objetos válidos de investigación.

6.4.3. El nuevo interés por las redes neuronales

Se puede decir que el interés por las redes neuronales renació en el año 1982 a partir del éxito de los trabajos de **J. J. Hopfield** sobre redes neuronales. El modelo de Hopfield introdujo las redes neuronales que seguían el principio

de la conectividad total y utilizaba señales binarias y con una arquitectura realimentada. La red neuronal de Hopfield, en sus varias versiones, se puede utilizar como una forma de memoria asociativa y también como un sistema para resolver problemas de optimización.

Otro sistema con gran éxito e influencia fue la red **NETTalk**, presentada en 1986 por T. J. Sejnowski y C. R. Rosenberg, de la Universidad John Hopkins. Se trata de una red neuronal de tres capas que tiene como objetivo aprender a pronunciar el inglés. La entrada es texto que ha de pronunciar una máquina que puede producir sonidos. Aunque al principio la red balbucea y se equivoca, gracias al entrenamiento y la corrección de los pesos de las interconexiones, acaba pronunciando correctamente, incluso al intentarlo con textos con los que no ha sido entrenada (sólo registra un 10% de errores, posiblemente porque la pronunciación inglés a veces depende del contexto).

Con este sistema también se demostró que el número de neuronas en capas ocultas es de gran importancia: pocas neuronas impiden que la red neuronal cumpla su función, pero con un exceso de neuronas en la capa oculta la red puede funcionar un poco mejor, aunque no sabe generalizar (en el caso de la NETTalk, es incapaz, por ejemplo, de leer textos que no haya preparado previamente).

También se considera de gran influencia la difusión en 1986 del **algoritmo *back propagation*** y el trabajo recapitulativo sobre el proceso paralelo distribuido editado por Rumelhart y McCleland. Nuevos investigadores como Feldman (propiedades de los modelos conexionistas), Grossberg (el cerebro adaptativo) y muchos otros han proporcionado un gran impulso al estudio de las redes neuronales.

Los hechos que resumen el nuevo interés se centran en el desarrollo de nuevas topologías y algoritmos, la posibilidad de implementaciones en circuitos VLSI, los éxitos de sistemas como el NETtalk de Sejnowski y Rosenberg y de las mismas redes de Hopfield y, también, la creciente fascinación sobre el funcionamiento del cerebro humano que las redes neuronales ayudan a estudiar. En cualquier caso, la redes neuronales son, otra vez, una tecnología en crecimiento de la que todavía resulta prematuro hacer hipótesis sobre su futuro.

6.5. Las redes neuronales y el cerebro humano

La similitud estructural entre las redes neuronales biológicas y las redes neuronales artificiales anima a utilizar estas últimas para entender mejor el cerebro humano y la dinámica de su funcionamiento.

En cualquier caso, la analogía no puede ser completa y hay que tener en cuenta muchos factores de diferenciación como, por ejemplo, las evidentes limitaciones de tamaño de las redes artificiales. En un cerebro humano se estima que hay en torno a 10^{11} neuronas que presentan una conectividad media del

orden de 1.000; esto sitúa el número de interconexiones o sinapsis en 10^{14}, cifras por ahora inalcanzables con los modelos artificiales que se ven reducidos a casos exageradamente simples.

Por otra parte, el algoritmo de propagación hacia atrás (*back propagation*), que tanto éxito ha tenido en las redes neuronales artificiales, no parece realista en el ámbito del cerebro humano, en el que la activación de salida de una neurona (al menos en el neocórtex) es excitadora o inhibidora, pero no las dos al mismo tiempo.

Otro problema importante que surge al asimilar el comportamiento de las redes neuronales artificiales con el del cerebro humano es la exigencia de una parte diferenciada del cerebro humano, la que haría de "profesor" en el proceso de aprendizaje, ya que ha de tratarse de neuronas con propiedades nuevas y específicas que parece que no existen en el cerebro humano.

Estas razones y muchas otras que no hace falta mencionar aquí hacen pensar a algunos especialistas como **F. Cricks** que estos supuestos "modelos" neuronales no modelizan nada real en el cerebro humano, y que otra vez se trata de sistemas que "manifiestan" en algunos casos un comportamiento similar al cerebro humano, sin que de ello se deduzca que se trate de un modelo real.

De hecho, aunque las redes neuronales biológicas no utilicen, por ejemplo, un algoritmo como el de propagación hacia atrás, lo cierto es que el comportamiento final es similar en algunos casos. No hay que olvidar que las redes neuronales resultan el sistema artificial más eficiente para el reconocimiento de formas y, curiosamente, ésta es una actividad cerebral muy sencilla que incluso los animales son capaces de realizar fácilmente, aunque, por la gran potencia de cálculo que exige, ha fracasado el intento de resolverla con sistemas artificiales (a menudo basados en la abstracción matemática) que no sean las redes neuronales.

7. Lenguajes específicos para la IA

A continuación damos constancia de los lenguajes específicos para la inteligencia artificial más relevantes.

7.1. Lisp

Es frecuente la consideración de que el Lisp es el lenguaje propio de la inteligencia artificial a causa de sus características particulares, su existencia desde 1962 y por haber estado sobradamente utilizado por los investigadores en el campo de la IA.

7.1.1. Origen

El Lisp se desarrolló en el MIT a raíz de un trabajo publicado por John McCarthy en 1960: *Recursive functions of symbolic expresions and their computation by machine*. Posteriormente, en 1962, se publicó el primer manual del Lisp para el equipo IBM 7090 y sus autores fueron el mismo **McCarthy junto con Abrahams, Edwards, Hart y Levin**, todos ellos del MIT. El lenguaje utiliza también las llamadas funciones "lambda" surgidas del *lambda calculus*, un lenguaje formal sobre funciones desarrollado por **A.Church**.

7.1.2. Objetivos y características principales

El Lisp implementa el tratamiento mediante expresiones simbólicas y recursividad y dispone de una gran potencia y simplicidad en la definición de estructuras. En Lisp no hay diferencias entre las estructuras de datos y las estructuras de programa; las dos se reducen a **expresiones simbólicas** que tienen como casos particulares los **átomos** y las **listas**. De aquí el nombre de Lisp derivado de ***list p**rocessor* (procesador de listas).

En realidad, el Lisp se puede tratar como un lenguaje matemático formal que, con las extensiones adecuadas, se convierte en un lenguaje de programación. Quizá por ello es un lenguaje sumamente flexible, e incluso lo puede cambiar el mismo programador.

Se trata, en sus inicios, de un **lenguaje funcional** que no tiene por qué disponer de la instrucción de asignación, cuyos programas son funciones que se definen por composición de otras funciones más simples. Para ejecutarlo, se "aplica" a los datos de entrada, que son los parámetros de la función, para obtener un resultado, que es el valor calculado de la función.

Las variantes del Lisp más utilizadas en la práctica también incorporan muchos elementos no funcionales en nombre de la eficiencia.

7.1.3. Dialectos

Un problema habitual para los usuarios del Lisp es que no existe ninguna estandarización válida y hay varios dialectos en uso. Se reconoce un núcleo central, denominado en ocasiones Pure Lisp, y variantes como el Maclisp, utilizado en el MIT; el PSL, con pretensión de estandarización portable desarrollado en la Universidad de Utah; el Le-Lisp, versión utilizada en Francia, etc.

Los intentos de estandarización se iniciaron sin mucho éxito con el Standard Lisp de Hearn, en 1966, y llegaron a un consenso con el Common Lisp aparecido en 1984. También cabe destacar los entornos de programación Lisp, de entre los cuales el Maclisp es el que se convirtió en uno de los más conocidos al ampliarse con variantes como el Franzlisp o el Zetalisp. También es posible mencionar el Interlisp, de la Universidad de Cambridge.

El lenguaje **Logo** es, en el fondo, una derivación simplificada del Lisp para su utilización en la enseñanza, donde se utiliza para crear los **entornos de descubrimiento**.

7.2. Prolog

El Prolog es otro de los lenguajes específicos más relevantes que hay sobre inteligencia artificial.

7.2.1. Orígenes

A comienzos de los años setenta, los partidarios de la programación lógica se dedicaron a implementar en el ordenador un sistema que gestionara la lógica causal que es, en realidad, una restricción de la lógica de predicados de primer orden (cláusulas de Horn). El lenguaje que implementaba la mencionada **PROgramación LÓGica** se denominó Prolog y nació en la Universidad de Marsella, bajo la dirección de **Alan Colmerauer**.

Algunos hitos fundamentales en la historia del Prolog son un primer programa para la prueba de teoremas escrito en 1973 por Colmerauer en Fortran; la formulación completa de la programación lógica de **Robert A. Kowalski** en 1974; la implementación de un Prolog eficiente en un PDP-10 realizada en 1977 por **David H. D. Warren** y, finalmente, la estandarización de la llamada **sintaxis de Edimburgo** en el libro *Programming in Prolog*, de **Cloksin** y **Mellish** en 1981.

Un elemento adicional a favor de la difusión y la fama del Prolog fue su uso anunciado en el proyecto japonés de la "quinta generación" de ordenadores.

7.2.2. Objetivos y características principales

La implementación de la lógica causal se consigue con una representación del conocimiento a base de hechos y reglas con la forma:

HECHOS: P(X1..., Xn)
REGLAS: P(X1..., Xn):- P1(...)..., Pm(...)
(donde ":-" se lee como "**si**")

que constituyen los elementos almacenados en la base de datos (conocimientos en forma de reglas y hechos en forma de hechos) del sistema Prolog.

Por lo tanto, se trata de un sistema de producción basado en reglas y es un **lenguaje declarativo** en el cual se indican los hechos y las reglas y no el procedimiento (algoritmo) con el que se encuentran las soluciones.

El usuario proporciona los hechos y las reglas y el Prolog incorpora un motor de inferencia que implementa los mecanismos de recursividad, unificación y razonamiento por encadenamiento hacia atrás (*backtracking*). Eso se hace patente en el modo de utilizar el Prolog una vez implementada la base de hechos y reglas: preguntar si es cierto o no un objetivo determinado (*goal directed*). Todo esto provoca que el Prolog sea especialmente adecuado para la implementación de sistemas expertos basados en reglas, y en la que sea correcto el uso de este mecanismo de razonamiento.

7.2.3. Dialectos

La problemática de los muchos dialectos del Lisp no aparece en el Prolog, en cuyo uso es frecuente utilizar la sintaxis de Edimburgo, establecida en 1981 en el libro ya mencionado de Cloksin y Mellish.

7.3. Smalltalk

Aunque el lenguaje Smalltalk se creó inicialmente para otros objetivos, también se ha considerado con gran interés en el ámbito de la IA. Se trata de un lenguaje concebido como soporte de la programación orientada a objetos y que presenta la ventaja de cumplir los requisitos generales de un lenguaje para IA: incorpora tratamiento simbólico y dispone de una estructura de *array* muy similar a las listas del Lisp.

Gracias a su adaptación a la programación orientada a objetos, el Smalltalk dispone de módulos que se transmiten **mensajes** que vienen a ser órdenes de ejecución de operaciones. Cada **objeto** se define mediante el conjunto de **variables** que constituyen su estado interno y de los **métodos** asociados a las operaciones que servirán para manipular las variables mencionadas.

También es de gran importancia la facilidad del Smalltalk para utilizar el concepto de **herencia**. La definición de un objeto en el Smalltalk implica indicar su clase, ya que todo objeto es una *instancia* de una clase. Las clases se organizan jerárquicamente y toda subclase "hereda" las variables y los métodos de la clase superior.

Algunos autores apuestan por el futuro de un sistema denominado **Loops**, que integra el uso del Smalltalk con el Lisp y el Prolog, aunque cualquier decisión es prematura en este sentido.

Actividades

Una actividad complementaria al alcance de muchos es atreverse a jugar al ajedrez contra un programa informático. Sin necesidad de llegar a la sofisticación del *Deep Blue*, el programa al cual se ha enfrentado varias veces el campeón humano Gari Kasparov, la realidad es que los jugadores no profesionales suelen perder ante programas de ajedrez incluso sencillos. Y éste no es más que un primer ejemplo de las técnicas de la inteligencia artificial en funcionamiento.

Para llegar a hacerse una idea de cómo podría ser un mundo con programas y máquinas inteligentes, es recomendable la lectura de la buena narrativa de ciencia-ficción en torno a los robots.

La mejor opción, evidentemente, son los relatos y las novelas de robots del científico, divulgador y novelista norteamericano Isaac Asimov, creador de las *Tres leyes de la robótica*, verdadero código ético para los robots inteligentes del futuro. Una propuesta de lectura divertida y sugerente, que recomendamos especialmente, sería:

Relatos:

Asimov, Isaac (1950). *Yo robot*. Barcelona: Plaza y Janés.

Asimov, Isaac (1982). *Los robots*. Barcelona: Martínez Roca.

Novelas:

Asimov, Isaac (1953). *Bóvedas de acero*. Barcelona: Plaza y Janés.

Asimov, Isaac (1957). *El sol desnudo*. Barcelona: Plaza y Janés.

Otra opción, menos optimista, es la lectura de algunas críticas muy serias que han surgido en contra del proyecto de la llamada "inteligencia artificial dura". En resumen, estas críticas plantean la imposibilidad del proyecto de la inteligencia artificial.

En este sentido, resulta recomendable la lectura, no tan sencilla pero muy interesante y estimulante, del libro:

Penrose, Roger (1989). *La nueva mente del emperador*. Barcelona: Mondadori, 1991.

Ejercicios de autoevaluación

1. ¿Cuál es el objetivo principal de la inteligencia artificial?
2. Ya se han construido programas que superan el test de Turing. ¿Verdadero o falso?
3. Los programas de inteligencia artificial son capaces de aprender. ¿Verdadero o falso?
4. Mencionad un ejemplo de programas de inteligencia artificial al alcance del gran público.
5. La inducción es la "inferencia lógicamente correcta". ¿Verdadero o falso?
6. ¿Qué es la heurística?
7. ¿Qué es la base de conocimientos de un sistema experto?
8. Los ingenieros del conocimiento son los expertos cuya experiencia emulan los sistemas expertos. ¿Verdadero o falso?
9. Mencionad dos lenguajes específicos de la inteligencia artificial.
10. Una red neuronal artificial suele tener miles de neuronas. ¿Verdadero o falso?

Solucionario

Ejercicios de autoevaluación

1. El objetivo principal de la inteligencia artificial es obtener sistemas informáticos con un comportamiento que consideraríamos inteligente en un ser humano.
2. Falso. A pesar de las expectativas del mismo Turing, que creía que más o menos hacia el año 2000 se superaría el test que lleva su nombre.
3. Verdadero. El aprendizaje es uno de los objetivos importantes de la inteligencia artificial.
4. Algunos programas informáticos que juegan al ajedrez utilizan técnicas de inteligencia artificial.
5. Falso. La "inferencia lógicamente correcta" es la deducción (*modus ponens* o instanciación universal).
6. La heurística es el conjunto de criterios, métodos o principios que se utilizan para encontrar, entre varios caminos posibles, cuál o cuáles son los más efectivos para obtener un objetivo determinado. Viene a significar las "reglas de la experiencia" para seleccionar un camino de acción.
7. La base de conocimientos de un sistema experto es el conjunto de reglas que sintetizan la experiencia del experto humano que se quiere emular.
8. Falso. Son los especialistas en inteligencia artificial que ayudan a transferir la experiencia y el conocimiento de un experto humano a un sistema experto.
9. Lisp y Prolog.
10. Falso. Con muy pocas neuronas se obtienen buenos resultados. La red de Lang/Witbrock, por ejemplo, utiliza sólo trece neuronas.

Bibliografía

Bibliografía básica

Cloksin, W. F.; Mellish, C. S. (1981). *Programming in PROLOG*. Nueva York: Springer-Verlag.

Cortés, U. [*et al.*] (1995). *Inteligencia artificial*. Barcelona: Ediciones UPC (Politexto, 17).

Cortés, U.; Sierra, C. (1987). *LISP*. Barcelona: Marcombo-Boixareu Editores.

Jackson, P. (1986) *Introduction to Expert Systems*. Reading, Massachusetts: Addison-Wesley.

McCorduck, P. (1979). *Máquinas que piensan: una incursión personal en la historia y perspectivas de la inteligencia artificial*. Madrid: Tecnos, 1991.

Wasserman, P. (1989). *Neural Computing, Theory and Practice*. Nueva York: Van Nostrand Reinhold.

Bibliografía complementaria

Charniak, E.; McDermott, D. (1992). *Introduction to Artificial Intelligence* (2.ª ed.). Reading, Massachusetts: Addison-Wesley.

Feigenbaum, E.; McCorduck, P. (1983). *The Fifth Generation*. Reading, Massachusetts: Addison-Wesley.

Goldberg, A.; Robson, D.; Ingalls, D. H. (1984). *SMALLTALK 80: The Languaje and Its Implementation*. Reading, Massachusetts: Addison-Wesley.

Hancock, P. A.; Chignell M. H. (eds.) (1989). *Intelligent Interfaces: Theory, Research, and Design*. Amsterdam: North-Holland.

Hofstader, D. R.; Dennett, D. C. (eds.) (1985). *The Mind's I*. Harmondsworth, Middx: Basic Books Inc., Penguin Books Ltd.

Kowalski, R. A. (1979). *Logic for Problem Solving*. Amsterdam: North-Holland.

Newell. A; Simon, H. (1972). *Human Problem Solving*. Englewood Cliffs, N. J.: Prentice-Hall.

Rich, E. (1992). *Artificial Intelligence*. Singapore: McGraw-Hill, International Student Editon.

Siklóssy, L. (1976). *Let's Talk LISP*. Englewood Cliffs, N. J.: Prentice-Hall.

Wertz, H. (1986). *LISP: Introducción a la programación*. Barcelona: Masson.