E.M. nº 72 Septiembre-Diciembre 2022

CONFERENCIAS CÉLEBRES

Continuamos esta sección de la revista, dedicada a Conferencias célebres impartidas en la Universidad Autónoma de Madrid a lo largo de su historia, bien como Lecciones inaugurales de curso académico, o bien impartidas en su investidura por Doctores Honoris Causa nombrados por esta universidad. Se trata por tanto de conferencias con importantes contenidos relacionados con la ciencia y el progreso del conocimiento, e impartidas por personalidades ilustres del mundo académico, científico o social.

En esta ocasión publicamos la **Lección inaugural** de la Universidad Autónoma de Madrid del Curso académico 2006-2007, pronunciado por **D. José R. Dorronsoro**, Catedrático de Ingeniería Informática de la UAM.

TECNOLOGÍA, COMPUTACIÓN E INTELIGENCIA

José Ramón Dorronsoro Ibero
Catedrático de Ingeniería Informática de la UAM

Nota actual del Autor:

Lo que sigue es esencialmente el texto de la Lección inaugural impartida al comienzo del curso 2006-2007. El propio (y ligeramente presuntuoso) título de la Lección la situaba en un contexto muy dinámico y, en cierto sentido, condenaba a algunos apartados a una obsolescencia provocada por el continuo y muy rápido asociado a la tecnología y la computación (la inteligencia se toma las cosas con más calma). De esto se derivan dos consecuencias. La primera es que hay que leer la Lección teniendo muy presente el contexto concreto de 2006; la segunda es la necesidad de una puesta al día siquiera parcial. Hacerlo con todo lo anticuado pudiera dar lugar a una segunda lección, lo que estaría fuera de lugar aquí; por eso me he limitado a añadir un breve epílogo que, al menos, actualice algo de lo que podría ser candente en 2006 pero que ya se ha quedado atrás.

1. INTRODUCCIÓN

Cuando a primeros de junio el Rector me llamó para comunicarme su intención de proponer al Consejo de Gobierno que impartiera la lección inaugural del curso 2006-2007, mis primeras reacciones fueron, por supuesto, de agradecimiento y satisfacción. Naturalmente, también me planteé el tema a abordar. Como es obvio, la primera reacción es moverse en los terrenos de la investigación propia, donde uno se siente más seguro, aunque no hasta el punto de olvidar la necesidad de dirigirse a una amplia y exigente audiencia. También me dio que pensar que se tratara de la primera lección a cargo de un profesor del claustro de la Escuela Politécnica Superior y, en cierta medida, supusiera su presentación al resto de la UAM. Por ello, la lección también debería reflejar de alguna manera lo que hacemos en la Escuela, algo que, por otra parte, puede suscitar el interés o la curiosidad entre los demás miembros de nuestra universidad. Hay que tener en cuenta que los estudios en Ingeniería Informática y de Telecomunicación son unos recién llegados a la UAM: los primeros se iniciaron en el curso 1992-93 y los segundos en el 2002-2003. Además, el perfil tecnológico de nuestros estudios

contrasta con el más marcado carácter científico y humanista de los demás centros. Sin determinar aún nada concreto, lo anterior acotaba las posibilidades: un tema significativo en algún aspecto de lo que se hace en la Escuela Politécnica, de interés para una audiencia amplia y, a ser posible no demasiado alejado de los terrenos que me son más familiares.

Así las cosas, el título "Tecnología, Computación e Inteligencia" surgió de manera más o menos natural. Ciertamente la tecnología se supone en una escuela de ingeniería, y la computación es inherente a la informática. Sobre la inteligencia pronto habrá más que decir. En cualquier caso, junto con el título deben ir unas disculpas. Primero, ante mis colegas a cuyo trabajo mi presentación no haga justicia. Luego ante la propia audiencia de la lección, por el posible tono grandilocuente del título. He de confesar que al proponerlo no las tenía todas conmigo y menos aun cuando, al comentarlo con un compañero, me hizo la observación de que sólo le faltaban los famosos dos huevos duros de Groucho Marx. Por ello, y antes de empezar la lección, puede ser bueno delimitar su terreno. La inteligencia no es patrimonio exclusivo de ninguna disciplina, pero probablemente la psicología, la filosofía, la medicina o las ciencias de la educación tengan bastante más que decir sobre la misma que la informática. Por ello, me voy a limitar aquí a la inteligencia que sería alcanzable en última instancia por los ordenadores, esto es, lo que ha venido en llamarse Inteligencia Artificial, o más brevemente, IA. Esta primera acotación en seguida trae otra, pues me temo que, a la vista de nuestra experiencia diaria con los muchos ordenadores que nos rodean, puede que la mayoría de los presentes también intuirán aquí cuál va a ser el final de la novela. Ciertamente en el trabajo, en casa o en los múltiples servicios que usamos cada día, la informática nos es cada vez más indispensable. Esta versatilidad también tiene, por qué no decirlo, algo de prodigiosa: sólo tenemos que pensar en el modesto PC, con el que, además de hacer algo útil de vez en cuando, oímos música, vemos películas, nos comunicamos no sólo por escrito sino también por videoconferencia o, a través de la World Wide Web, tenemos literalmente el mundo a nuestro alcance. Todo ello es posible por la enorme capacidad de los ordenadores para realizar de manera incansable procesos de captura, organización, gestión, acceso y presentación de grandes volúmenes de datos. Pero, aunque en un PC y, en general, en la informática, hay ciertamente mucha inteligencia, la que otorguemos a ambos probablemente no nos parecerá en absoluto comparable a la nuestra.

Pues bien, ya les anticipo que al acabar la lección esta opinión no va a cambiar. De hecho, para esta conclusión tal vez no haga falta llegar a su final. Hace unos pocos años Steven Spielberg nos ofreció su película *Inteligencia Artificial*, donde en el siglo XXII unos robots, los mecas, viven entre los humanos, los orgas, y, salvo tal vez por su falta de parpadeo, son indistinguibles de nosotros. Por supuesto, los mecas son ya inteligentes en prácticamente cualquier sentido del término, aunque carecen de sentimientos. De hecho, la larga jornada que el protagonista, David, realiza en la segunda mitad de la película, viene a ser un test para determinar si es realmente capaz de sentir emociones; como veremos, en Inteligencia Artificial, los tests son algo recurrente. Spielberg es uno de los grandes cineastas de todos los tiempos, pero esta vez no me causó el impacto de otras muchas de sus obras. Toda película tiene su tiempo y la enorme familiaridad con los ordenadores del nuestro probablemente hiciera que las promesas de los robots de Spielberg me parecieran muy remotas. O tal vez muy poco artificiales: su protagonista, Haley Joel Osment, es demasiado humano para ser un meca creíble; Jude Law da mucho mejor el papel. Sin embargo, otra película anterior, *2001, una odisea del espacio*, de Stanley Kubrick, también nos mostraba un ordenador pensante, HAL, del que apenas veíamos un panel parpadeante y escuchábamos una voz metálica, pero que resultaba mucho más real y amenazante. Además del arte de Kubrick, a ello contribuía su tono hipnótico, su perversa astucia y el haber escapado al control de sus creadores. Pero también su tiempo fue muy importante: en 1968 había muchos menos ordenadores y su misterio, por tanto, era mucho mayor. Muy probablemente una lección inaugural sobre Inteligencia Artificial impartida ese año hubiera sido muy distinta de ésta. Vamos a intentar ver por qué las cosas eran así en 1968 y cómo son hoy día.

2. LA MECANIZACIÓN DEL RAZONAMIENTO

Si alguien nos pregunta por un ámbito donde la inteligencia se manifieste de manera clara, probablemente serán las Matemáticas lo primero que nos venga a la mente. Por supuesto, se trata en parte de un lugar común: hace pocos días hemos visto que, con motivo del Congreso Internacional de Matemáticas recién celebrado en Madrid, los medios han designado al ya famoso matemático ruso Grigory Perelman de manera más o menos automática como el hombre más inteligente del mundo. Pero también es cierto que cualquier persona expuesta a la claridad, elegancia, certeza y rigor de, por ejemplo, la geometría euclídea, probablemente reconozca en sus construcciones un grado muy alto de mérito intelectual. Además, si queremos asociar inteligencia y computación, las Matemáticas de nuevo aparecen de manera natural. El descubrimiento de nuevas verdades matemáticas sigue un camino muy parecido al proceso de prueba y error, ejemplos y refutaciones, avances parciales y vías muertas, seguido en otras ciencias. Pero una vez se ha llegado a un nuevo teorema, su demostración suele presentarse mediante un riguroso razonamiento que arranca de unos supuestos ya establecidos y llega a la nueva verdad mediante un encadenamiento de deducciones lógicas. Este proceso poco menos que inexorable ha atraído a todos aquellos que han pretendido mecanizar el pensamiento. Un ejemplo bien conocido es el filósofo y matemático Gottfried Leibniz, cuyas *characteristica universalis* y *calculus ratiocinator* fueron un intento de descripción simbólica del universo y de mecanización del pensamiento, en el que se ha visto un antecedente conceptual de los modernos ordenadores.

Sin embargo, el mayor impulso vino de las propias Matemáticas y, en concreto, del creciente trabajo de formalización del pensamiento matemático de la segunda mitad del siglo XIX. El mismo se inició con los esfuerzos de Karl Weierstrass para reemplazar el carácter eminentemente discursivo de la argumentación matemática de la época por unos fundamentos más sólidos, al que siguió una amplia serie de trabajos debidos, entre otros, a Georg Cantor, Richard Dedekind o Leopold Kronecker, que profundizaron en la búsqueda de un método simbólico y axiomático, y que en cierta medida culminaron con la publicación de los *Grundgesetze der Arithmetik* de Gottlob Frege. Sin embargo, en 1903 y poco antes de la aparición de su segundo volumen, Bertrand Russell observó que sus axiomas podían dar lugar a paradojas, como la posibilidad de definir un conjunto S como "el conjunto de todos los conjuntos que no se contienen a sí mismos como elementos", lo que lleva a la afirmación absurda de que S es un elemento de S si y sólo si S no es un elemento de S. Su equivalente coloquial es la muy conocida paradoja del barbero: si definimos un barbero como la persona que afeita a quienes no se afeitan a sí mismos y en una cierta ciudad donde todo el mundo se afeita hay un solo barbero, la pregunta ¿quién afeita al barbero? no tiene respuesta. En efecto, si respondemos que el propio barbero, se incurre en contradicción con la definición. Pero si respondemos lo contrario, el barbero no se afeitará entonces a sí mismo, por lo que debe ser afeitado... ¡por el barbero!

La paradoja de Russell y otras que le precedieron originaron un sentimiento de crisis en los fundamentos de las Matemáticas. Así las cosas, en el Congreso Internacional de Matemáticas de París de 1900 David Hilbert, propuso la demostración de la consistencia de los axiomas de la aritmética como uno de los problemas fundamentales de las Matemáticas. Por consistencia de un sistema axiomático se entiende que en el mismo no se puede llegar a una afirmación que sea simultáneamente verdadera y falsa, al revés de lo que sucede en la paradoja de Russell. Los trabajos subsiguientes de Hilbert, una de las cumbres matemáticas de todos los tiempos, y de su escuela ampliaron considerablemente la formulación inicial y desembocaron en los años 20 en lo que se conoce como el programa de Hilbert, cuyo objetivo último era dotar a las Matemáticas de un conjunto de axiomas y una formalización sólida que permitieran escribir cualquier teorema matemático en un lenguaje formal preciso y reducir su demostración a la aplicación exacta de unas reglas bien definidas. Para ello era imperativo centrarse en la manipulación simbólica y prescindir de cualquier connotación natural o intuitiva, tal y como había hecho el propio Hilbert con los conceptos de punto o recta en su axiomatización de la geometría. Esta abstracción extrema permite una gran libertad para concebir y analizar sistemas formales, pero a su vez plantea nuevas exigencias. Así, y además del requisito de consistencia, el programa de Hilbert exige la completitud del sistema axiomático buscado, esto es, el

poder demostrar cualquier afirmación matemática cierta que sea expresable en el mismo. La famosa afirmación de Hilbert de que en Matemáticas no hay un "ignorabimus", de que por muy inabordable que parezca un problema, debe ser posible resolverlo, muestra su confianza en el éxito final de la empresa. Sin embargo, justamente en este punto surgió un hecho imprevisto y revolucionario, el teorema de incompletitud de Gödel[1], que afirma que en cualquier sistema formal consistente que incluya los axiomas de la aritmética es posible construir una afirmación que, si bien es cierta, no puede ser demostrada dentro del sistema. En otras palabras, el resultado de Gödel afirma simplemente que la certeza de Hilbert no se puede sostener en su sentido estricto. De hecho, el teorema de incompletitud contiene una sutil ironía, pues entendida de forma adecuada, la proposición verdadera pero no demostrable dentro del sistema axiomático es precisamente su propia consistencia que, sin embargo, sí puede demostrarse en un meta-sistema adecuado.

El teorema de Gödel supone un límite a lo alcanzable mediante manipulación simbólica, por lo que volverá a aparecer cuando abordemos la posibilidad de computadores inteligentes. Pero otra cuestión planteada por Hilbert obligó a profundizar en el concepto mismo de computación. Dicha cuestión es el problema de decisión: demostrar la existencia de un algoritmo que decida si una cierta fórmula lógica escrita en un sistema axiomático es demostrable o no dentro del mismo. En 1936 e independientemente, Alonzo Church y Alan Turing dieron una contestación negativa. Sus trabajos son el origen de la moderna teoría de la computación, y, en el caso de Turing, puede decirse que también de los modernos ordenadores. Por esta y otras razones, Alan Turing tendrá una destacada presencia en lo que sigue[2].

Un requisito previo para poder abordar el problema de decisión es precisar el concepto de algoritmo, a su vez un concepto básico en computación. Hay una idea natural, la de una receta particularmente precisa en su formulación. Así, todos sus pasos han de especificarse sin ambigüedades, sus operaciones han de ser suficientemente elementales como para poder ser replicadas con papel y lápiz y su ejecución ha de terminar tras un número finito de pasos. Por ejemplo, todos reconocemos estas propiedades en el algoritmo por antonomasia, el de Euclides para el cálculo del máximo común divisor. Pero, aunque intuitiva y aunque el concepto de algoritmo se explique más o menos así a los estudiantes de primero de informática, las paradojas a las que condujo la también intuitiva teoría inicial de conjuntos dejan claro que esta aproximación no sirve. Por ello, la primera tarea de Church y Turing fue una formalización rigurosa del concepto de algoritmo. La solución de Church fue el Cálculo Lambda, donde un algoritmo se define mediante funciones recursivas y que ha inspirado la programación funcional, un paradigma que justamente enfatiza la evaluación de funciones como el fundamento de la computación y de la que el lenguaje LISP es un ejemplo. El paradigma alternativo es la programación imperativa, representada en lenguajes como FORTRAN o C, cuyos programas suponen la ejecución secuencial de órdenes que cambian su estado. El origen de la programación imperativa está en el trabajo de Turing, quién se preguntó por cuáles serían las operaciones más básicas que un algoritmo pudiera llevar a cabo. Su respuesta fue la máquina de Turing que formalmente consiste en un conjunto finito de estados, incluyendo los de arranque y parada, una tabla de consulta también finita, y una cinta infinita, que puede moverse a izquierda y derecha y sobre la que una cabeza lee y escribe ceros y unos. Esta cinta contiene los datos iniciales para la máquina, así como las salidas obtenidas tras su ejecución[3].

Es fácil ver la analogía entre una máquina de Turing dada y un programa de ordenador, pero hay un tipo particularmente importante de máquinas de Turing que son análogas al concepto mismo ordenador. Se trata de las máquinas universales, capaces de simular el funcionamiento de cualquier otra máquina. Más concretamente, una máquina universal U recibe como entrada una cierta codificación de una máquina particular M así como una entrada concreta I para ésta, y produce como salida el mismo resultado que hubiera proporcionado la máquina M sobre la entrada I. En sentido estricto el modelo de cómputo subyacente a los ordenadores, el de las máquinas de acceso aleatorio, es formalmente distinto. Sin embargo, en el fondo, un ordenador es también un programa capaz de ejecutar cualquier otro programa.

Turing redujo el problema de decisión al denominado problema de parada: para una máquina y una entrada dadas, determinar si su ejecución llegará a detenerse o, por el contrario, continuará indefinidamente, y demostró que no hay un algoritmo, entendido naturalmente como una máquina de Turing, que resuelva el problema de detención para otras máquina y entrada cualesquiera. Queda pendiente la cuestión de si realmente las máquinas de Turing capturan todas las posibilidades de lo que puede entenderse como una operación lógica "mecánica" o, dicho de otra forma, si cualquier concepto de computación concebible puede ser realizado por las mismas. La tesis de Church-Turing afirma que esto es así, aunque por su poco precisa formulación, debe entenderse más como una hipótesis que como una afirmación formal susceptible de demostración. Una forma de refutación sería introducir algún modelo alternativo de computación y demostrar que no es equivalente al de Turing. Sin embargo, los que se han ido proponiendo (cálculo lambda, funciones recursivas, sistemas de Post o máquinas de registros) se han demostrado equivalentes a las máquinas de Turing.

En particular, cualquier tarea computable por un ordenador moderno es también computable por una máquina de Turing, por lo que la tesis de Church-Turing ha de tenerse en cuenta si nos planteamos la posibilidad de computadores inteligentes. Su versión original ha sido extendida de diversas formas. Dos destacables son la tesis física y la tesis extendida o fuerte. La primera tiene que ver con la implementación material de cualquier procedimiento de cómputo que, obviamente, debe someterse a las leyes de la física; éstas, por tanto, también delimitarían qué es computable. La tesis física afirma justamente que cualquier procedimiento computable mediante un dispositivo físico puede también computarse mediante una máquina de Turing. A su vez, la tesis fuerte surgió dentro de la teoría de la complejidad computacional, la rama de la computación que estudia la eficacia de los algoritmos, y afirma que las máquinas de Turing son un modelo de computación tan eficaz como otro cualquiera. De manera más precisa, si una cierta computación sobre una entrada de tamaño N requiere en una cierta máquina un tiempo $T(N)$, la tesis fuerte afirma que la misma será también computable en una máquina de Turing en un tiempo $T(N)^k$ para una cierta potencia k dependiente del problema en cuestión. Una implicación de la tesis fuerte es que para conseguir ordenadores más potentes basta seguir mejorando la tecnología actual. Por su parte, si vemos al cerebro humano como un sistema gobernado por las leyes de la física y a la inteligencia como resultado de su operación, la tesis física nos llevaría a concluir que la inteligencia puede implementarse sobre una máquina de Turing.

Más adelante volveremos a estas cuestiones, pero hasta ahora toda nuestra discusión se ha realizado de manera abstracta, sin considerar realizaciones concretas. La razón es muy simple: ésta era la única posibilidad antes de los años 40. Si bien Leibniz y Blaise Pascal construyeron máquinas calculadoras, la primera máquina programable, esto es, suficientemente versátil como para realizar de forma autónoma cualquier cómputo, fue la máquina analítica de Charles Babbage. Aunque nunca llegó a construirse, había en su diseño elementos típicos de los ordenadores modernos, como una memoria, un mecanismo de entrada y salida de datos e incluso un primitivo lenguaje de programación. Es complicado establecer cuál fue la primera máquina realmente merecedora del calificativo de ordenador. En sentido estricto, probablemente el título deba recaer en el Z3, construido en 1941 por el ingeniero alemán Konrad Zuse; sin embargo, la máquina de Zuse, que no sobrevivió a la segunda guerra mundial, no era una máquina electrónica sino electromecánica, en la que unos 2.000 relés hacían la función de los transistores actuales. Los primeros ordenadores electrónicos fueron el americano Electronic Numerical Integrator and Computer, ENIAC, y los Colossi ingleses. Ambos usaban tubos de vacío y aunque su diseño era más primitivo que el del Z3, también permitían un cierto nivel de programación, que se hacía mediante el reajuste de clavijas y cables en un panel de control. Ambas máquinas se concibieron con unas tareas concretas como objetivo. Así, los Colossi se utilizaron para el criptoanálisis de mensajes cifrados con la máquina Lorenz SZ42, usada por el ejército alemán para sus comunicaciones de más alto nivel. Para mantener en secreto sus capacidades criptoanalíticas, la mayoría de los Colossi fueron desmantelados tras el fin de la guerra y su mera existencia mantenida en secreto hasta finales de los 70, por lo que apenas influyeron en el diseño de nuevos ordenadores. Sin embargo, el ENIAC, diseñado por John Prespert Eckert y John Mauchly, se mantuvo en operación

hasta 1955; su primer uso fue el cálculo de tablas balísticas, al que siguieron la predicción meteorológica, cálculos en energía atómica o estudios sobre números aleatorios. Contenía unos 18.000 tubos de vacío, no tuvo memoria hasta 1953, y era capaz de unas 300 multiplicaciones por segundo. Esto es varios órdenes de magnitud inferior a lo posible en los PCs actuales, cuya potencia se mide en gigaflops, esto es, miles de millones de operaciones por segundo y más aún frente a los superordenadores actuales, que superan los 200 teraflops y pronto llegarán al petaflop, esto es, 10^{15} operaciones por segundo. Sin embargo, suponían una ampliación extraordinaria de las capacidades humanas, por lo que es natural que, tras la puesta en funcionamiento del ordenador Mark I en Harvard, la revista TIME se preguntara en una portada de enero de 1950 "Can man build a superman?" o que la denominación de "cerebros electrónicos" pronto fuera de uso común

3. EL TEST DE TURING

Una vez hecho realidad el ordenador, las discusiones sobre la posibilidad de máquinas pensantes, restringidas hasta entonces al ámbito de lo hipotético, tenían ya un terreno concreto en el que dirimirse y en el que desde un principio estuvo presente Alan Turing. No sólo es su máquina universal un antecedente conceptual directo, sino que el propio Turing tuvo un papel muy activo en el diseño de los primeros ordenadores. Su origen está en su destacada participación durante la segunda guerra mundial en el criptoanálisis de las máquinas Enigma del ejército alemán, para lo que diseñó una serie de máquinas electromecánicas conocidas como bombas ("bombes") criptográficas. Si bien no podían programarse, permitieron a Turing familiarizarse con la tecnología necesaria para la construcción de ordenadores. Además, su amistad con John von Neumann le hizo conocer de primera mano los fundamentos del ENIAC. En 1946 Turing presentó un diseño totalmente novedoso de un ordenador electrónico, el ACE, Automatic Computing Engine, cuya construcción se completó en 1950, aunque finalmente sin su participación.

Todo esto hace muy natural que el propio Turing fuera uno de los primeros en relacionar explícitamente las nuevas máquinas con la inteligencia con su artículo de 1950 *Computing Machinery and Intelligence*[4]. Si bien en su primera línea aparece la pregunta ¿pueden pensar las máquinas?, Turing no la encuentra suficientemente satisfactoria y opta por plantear otra cuestión, el famoso test de Turing, un juego de imitación en el que un interrogador humano se enfrenta simultáneamente a una máquina y a una mujer y, mediante diversas cuestiones, debe determinar quién es quién. Si al final del juego el interrogador se equivoca, como suele pasar con cierta frecuencia cuando participan tres seres humanos, la máquina habrá pasado la prueba. Turing concede que no tiene argumentos para afirmar que en algún momento una máquina supere el test; su propuesta es más bien una forma de hacer camino al andar. Pero más bien cree que sí, estimando que hacía el año 2000, una máquina con una memoria de unos 120 MB debería engañar a los interrogadores un 30% de las veces en un interrogatorio de cinco minutos. Por supuesto, estas estimaciones han de tomarse como meramente especulativas: en 1950 había en el mundo unos 10 ordenadores, de diseño elemental, operación precaria y prestaciones limitadas. En cualquier caso, el hardware no preocupaba a Turing, que percibía claramente que la principal dificultad iba a estar en la programación, sugiriendo emplear para ello "learning machines", máquinas capaces de aprender.

A día de hoy ningún ordenador ha superado el test de Turing. De hecho, hay una competición, el premio Loebner, que además de ofrecer una recompensa global al sistema que supere su interpretación del test, cada año premia al que logre una conversación "más humana". Recientes ganadores como A.L.I.C.E. o Jabberwacky proporcionan páginas web donde interactuar con ellos; al hacerlo se obtiene una idea de por qué el premio global no se ha concedido aún. De hecho, no puede decirse que el test de Turing sea hoy objeto de una investigación muy activa, pero sí puede pensarse que fuera un catalizador para el inicio de la investigación en Inteligencia Artificial. El término fue propuesto por John McCarthy a la hora de organizar en 1956 la Dartmouth Summer Conference on Artificial Intelligence, la primera conferencia dedicada al tema y entre cuyos organizadores estaba Claude Shannon, el creador de la moderna teoría de la información, interesado entonces en cómo

construir programas que jugaran al ajedrez. Si bien el concepto de IA se presta a muchos calificativos y matizaciones, su variante más radical es la IA fuerte, que busca nada menos que construir máquinas con una capacidad intelectual indistinguible de la humana y que era el objetivo final de no pocos de los primeros investigadores del campo. En 1956 ya existían programas capaces de jugar a las damas a alto nivel (un programa ganó el campeonato mundial en 1994) y poco después John McCarthy concibió el lenguaje LISP, inspirado, como hemos dicho, en el cálculo lambda de Church, y Allen Newell y Herbert Simon, receptores en 1975 del premio Turing, el Nobel en computación (Simon obtuvo en 1978 el Nobel de economía), presentaron su Logic Theorist, un programa capaz de demostrar algunos teoremas lógicos de los Principia Mathematica de Russell y Alfred North Whitehead, provocando que, al parecer, Russell se lamentara irónicamente de no haberlo sabido en su momento y haberse podido evitar 10 años de trabajo. Pronto aparecieron también profecías muy optimistas, como que en 10 años un programa sería el campeón mundial de ajedrez o que, simplemente, en un futuro cercano aparecerían máquinas capaces de pensar, aprender, crear y, en general, afrontar desafíos afines a los de la mente humana.

En los 60 aparecieron los primeros robots con cierta capacidad de movimiento y percepción, se empezó a trabajar en traducción automática y proceso del lenguaje natural y se construyeron programas capaces de jugar al ajedrez a un cierto nivel. Pero los 70 fueron la época dorada de esta primera fase de la IA. Así, el programa SHRDLU de Terry Winograd entendía órdenes para apilar bloques dadas en un inglés básico y respondía a preguntas sobre el estado del apilamiento; estas capacidades, junto con la visualización de sus resultados en una de las primeras terminales gráficas, causaron un notable impacto. En 1972, Alain Colmerauer desarrolló Prolog, un lenguaje de programación basado en lógica de predicados y donde la ejecución de un programa viene a suponer la demostración de un cierto teorema. A mediados de los 70 David Marr, inspirado por los trabajos de David Hubel y Torsten Wiesel sobre el proceso de información en el córtex visual, desarrolló junto con Tomaso Poggio su modelo de visión, que describe los pasos que transforman una sensación bidimensional en la retina en una representación tridimensional en el cerebro, y cuyo arranque es el "primal sketch", un esbozo básico que captura bordes y regiones de la misma manera que un dibujante haría una primera representación de una cierta escena. En 1977 Douglas Lenat presentó su programa AM, abreviatura de Automated Mathematician, que a partir de un amplio catálogo de "conceptos" y reglas heurísticas era capaz de descubrir nuevos conceptos e hipótesis. Entre cierta controversia, se afirmó que AM había sido capaz de redescubrir la unicidad de la descomposición en factores primos, así como la conjetura de Goldbach, probablemente el problema más antiguo en teoría de números aún sin resolver. Y en 1979 Hans Moravec construyó en la Universidad de Stanford un robot autónomo que usaba una cámara de televisión para recorrer un espacio sorteando obstáculos.

Todos estos logros se debieron, por supuesto, al trabajo de un número creciente de investigadores, pero también fueron paralelos y en gran medida deudores de notables avances en tecnología informática, tanto en software como en hardware. Así, y tras la aparición de a partir de la segunda mitad de los 50 de los lenguajes FORTRAN, ALGOL y LISP, el formalismo de Backus-Naur para la descripción de lenguajes y los avances en diseño de compiladores facilitaron la aparición de nuevos lenguajes de programación, como APL, Basic o Simula, y la introducción de nuevos paradigmas, como la programación estructurada o la programación orientada a objetos. Por su parte, en los 60 se producen considerables avances en el diseño de sistemas operativos, con un énfasis creciente en su virtualización y en la incorporación de la multitarea. Este progreso se consolidó en los 70, una década dorada también en computación. En sus inicios Ted Codd introdujo el concepto de base de datos relacional y Klaus Wirth creó el lenguaje Pascal, orientado a la programación estructurada. Por su parte, Ken Thompson y Dennis Ritchie desarrollaron UNIX, el primer sistema operativo multitarea. Otra novedad de UNIX fue el estar escrito casi en su totalidad como un programa en C, un lenguaje de alto nivel creado por Ritchie, por lo que UNIX podía ser en principio usado en cualquier ordenador para el que hubiera un compilador de C. En una época de estándares propietarios y donde los sistemas operativos eran inseparables del hardware subyacente, UNIX fue un primer ejemplo de sistema abierto. Otra invención de los 70 y otro ejemplo de sistema abierto fueron los protocolos de comunicación de

internet desarrollados por Robert Kahn y Vinton Cerf, y que supusieron el primer paso hacia la actual intercomunicación masiva entre ordenadores.

Hubo dos importantes razones para estos avances. La primera fue la madurez conceptual alcanzada por la informática. En los 60 aparecieron los estudios de informática que, sobre todo en los EE.UU., experimentaron un crecimiento espectacular en su segunda mitad, por lo que la Association for Computing Machinery (ACM) publicó en 1968 el primero de sus currícula docentes en Ciencias de la Computación, al que siguió en 1978 el segundo. De alguna manera, ambos reflejan la concepción de la informática de su momento, y frente al todavía tentativo currículum del 68, en el del 78 se aprecia ya un carácter moderno y en gran medida aún reconocible en los estudios actuales de informática. Y, por supuesto, otra razón de peso es que, tras la introducción de los transistores y los circuitos integrados, la famosa ley de Moore empezó a dar sus frutos de manera sistemática. Hay que recordar que los ordenadores de la época usaban (y mayoritariamente usan) la arquitectura secuencial de von Neumann, donde en cada momento la unidad de proceso sólo puede atender a una tarea. Por otra parte, los sistemas operativos han de ofrecer servicios multitarea si se quiere que atiendan a las demandas simultáneas de aplicaciones, usuarios o comunicaciones. Hay pues una aparente contradicción que sólo puede ser superada aumentando la capacidad de proceso. Dicho de otra manera, la multitarea es una ilusión hecha posible por la velocidad de los procesadores, que sólo empezó a ser suficiente en los 60, cuando los microchips supusieron un gran salto en potencia de cómputo, fiabilidad, abaratamiento y, por tanto, difusión de las nuevas máquinas. Esta tendencia se reforzó de manera espectacular con la introducción de los microprocesadores a finales de los 70 y todo ello dio una gran vitalidad a la industria informática. El entorno predominante seguía siendo el gran ordenador central, o mainframe, dominado por IBM con la que competían otras compañías como Sperry-Rand, NCR o Honeywell en lo que se dio en llamar "Blancanieves y los siete enanitos". Pero también aparecieron nuevos actores como Hewlett-Packard, Digital Equipment Corporation o Sun, asociados a los nuevos minicomputadores y al inicio de la informática distribuida.

Tal vez el éxito más visible y sólido de la IA de los 70 fuera los sistemas expertos. En su origen el término era sinónimo al de sistemas basados en reglas o "sistemas de producción", un concepto derivado de un modelo de computación propuesto por el lógico Emil Post y equivalente al de Turing, y que Allen Newell y sus colaboradores adaptaron a sus estudios de modelización cognitiva, asimilando la asociación estado-regla a un mecanismo de estímulo-respuesta. Un sistema de producción tiene tres componentes básicas: una base de datos global, con los datos accesibles por el sistema, un conjunto de reglas de producción que definen las operaciones posibles en la forma condición-acción y, finalmente, un sistema de control, que de cierta manera navega entre las reglas y determina cuáles se aplicarán sucesivamente. De manera paralela, en un sistema experto típico también se suelen reconocer tres componentes características: una base de conocimiento formada por un amplio conjunto de hechos y reglas que recogen un conocimiento experto del problema a abordar, una memoria temporal de trabajo donde se guardan nuevos hechos intermedios a los que va llegando el sistema, y un motor de inferencia que "razona" mediante la aplicación de reglas a esos hechos, bien intentando llegar a alguna conclusión a partir de unos iniciales o, al revés, intentando contrastar la veracidad de una hipótesis objetivo mediante el conocimiento del sistema. Sin embargo, en los años 80 aparecieron otras concepciones e implementaciones que también se denominaron sistemas expertos; por ello, y al igual que sucede con otros términos informáticos, es más sencillo dar una definición general en términos funcionales. Así, se espera que un tal sistema alcance la capacidad de un experto humano al menos sobre dominios suficientemente acotados; además su conocimiento ha de representarse de manera simbólica susceptible de explicitarse ante sus usuarios y, finalmente, debe poder explicar sus decisiones, siquiera mediante una enumeración de las reglas aplicadas. El primer sistema experto de éxito fue MYCIN, desarrollado por Ted Shortliffe en la universidad de Stanford y aplicado al diagnóstico médico, al que siguieron muchos otros, como XCon, usado por la firma Digital Equipment Corporation en la configuración de sus máquinas VAX, o DENDRAL, que determinaba la estructura de compuestos orgánicos a partir de datos experimentales. Todos recibieron una gran publicidad y se presentaron como los primeros logros prácticos de la IA. Ciertamente lo eran, pero también pusieron

de manifiesto sus limitaciones. La principal era la necesidad de disponer de toda la información sobre el problema a abordar y que en la misma no hubiera contradicciones, algo tanto más difícil cuanto mayor es la base de conocimientos. Tampoco les era fácil manejarse en circunstancias de incertidumbre o hacer analogías entre el conocimiento que almacenaban y el aplicable en circunstancias similares, resultando en sistemas de gran fragilidad, incapaces de afrontar pequeños cambios en lo reflejado en su base de conocimiento. Además, este conocimiento correspondía en general al de un experto humano concreto; la inteligencia de este no se discutía ni tampoco la utilidad de un tal sistema, pero sí un carácter realmente inteligente del mismo.

En realidad, muchas de estas críticas no eran sino continuación de otras de carácter más fundamental que desde diversos ámbitos llevaban un cierto tiempo realizándose contra los objetivos y afirmaciones de la IA de la época y que, como consecuencia del debate sobre los sistemas expertos, se intensificaron enormemente en la década de los 80. Muchas atacaban directamente la IA fuerte que, recordamos, mantiene que la mente humana no es sino un ordenador convenientemente programado. Entre estas destaca el argumento de la Sala China propuesto por el filósofo John Searle[5] que critica específicamente la idea de que llevar a cabo una tarea, como hace un ordenador al ejecutar un cierto programa, pueda ser equivalente a una comprensión de la misma. Searle ilustra esta situación con una persona angloparlante que efectúa manualmente los distintos pasos de un supuesto programa de conversación en chino que, tras oír una historia, responde correctamente a distintas cuestiones sobre la misma. Searle argumenta que, aunque las manipulaciones del ejecutor humano den resultados satisfactorios, no por eso puede afirmarse que dicho ejecutor haya entendido la historia, las preguntas o las respuestas. En otras palabras, lo que Searle niega es que un programa de ordenador, que en el fondo obedece a reglas sintácticas, pueda poseer las capacidades semánticas que son inherentes a la mente.

Otra línea de ataque, iniciada por el filósofo John Lucas[6] y continuada por el físico y matemático Roger Penrose[7], se basa en el propio teorema de Gödel. Recordemos que, en el mismo, Gödel afirma que en cualquier sistema de axiomas que contenga la aritmética es posible construir una fórmula que no puede ser demostrada en dicho sistema, pese a ser cierta y demostrable por la mente humana en otro meta-sistema apropiado. Dado que cualquier concepto razonable de inteligencia debería incluir la aritmética, Lucas deduce una contradicción entre la hipótesis de que la mente humana obedezca a un cierto sistema formal concreto pero que sea a la vez capaz de llegar a verdades indemostrables dentro del mismo. Penrose[8] discrepa además de lo que antes hemos llamado la tesis física de Church-Turing de que todas las posibles leyes de la física sean computables. Una refutación de la tesis física haría posible una mente no simulable mediante un ordenador si en la misma tuvieran lugar procesos que obedecieran a leyes no computables. Naturalmente, todas estas discusiones han tenido sus contrarréplicas. Así, muchos físicos discrepan de la sugerencia de Penrose de leyes físicas no computables. Ha habido también fuego cruzado entre partidarios de una misma teoría, como muestran las críticas a los argumentos de Penrose del lógico Solomon Feferman[9], quién por otra parte también defiende la implausibilidad de la IA fuerte. El mismo Gödel tampoco admitía la IA fuerte, aunque por razones más cercanas al dualismo cartesiano. Por su parte, y centrándose en su test, el propio Turing había ya considerado algunas de estas objeciones. Así, su "objeción matemática" se anticipa a los argumentos de Lucas, respondiendo, no sin bastante ironía, que, aunque una máquina pueda dar respuestas incorrectas o incluso no dar ninguna, tampoco está garantizado que los seres humanos no estemos sujetos a las limitaciones lógicas encontradas por Gödel o, menos aún, que seamos consistentes. Otra de las objeciones consideradas por Turing, la que llama de Lady Lovelace, se anticipa en alguna medida a la de Searle. Ada Lovelace, hija de Lord Byron, ayudante de Babbage y en cierto sentido la primera programadora, afirmó en uno de sus escritos que la máquina analítica carece de iniciativa y sólo puede hacer lo que se le ordene. En respuesta Turing presenta su argumento de las capas de una cebolla. Ciertas respuestas cerebrales tienen un claro componente mecánico, por lo que son la primera capa que hemos de eliminar para llegar a la mente "real". Turing se pregunta dónde acabaríamos tras eliminar sucesivamente todas las capas: si, como ocurre con la cebolla, no quedara nada, tendríamos que concluir que la mente opera mecánicamente.

La discusión tiene ciertamente un carácter muy sutil y está muy lejos de haber acabado. En cualquier caso, hubo a partir de la segunda mitad de los 80 un claro decaimiento de la IA , tanto en atención mediática como en apoyo económico. La discusión anterior ciertamente hizo mella, pero la razón principal fue más bien una de promesas incumplidas y, también, inversiones fallidas y créditos impagados. Como hemos visto, parte de la comunidad de la IA no fue ni mucho menos modesta en sus afirmaciones. Si en un principio se les podía conceder el beneficio de la duda y esperar a los resultados de su esfuerzo, tras casi 30 años esto ya no era así, y más aún bajo el punto de vista de las distintas agencias promotoras de investigación científica y tecnológica. Ya en 1973 apareció en el Reino Unido el muy crítico informe Lighthill, preparado por encargo del British Science Research Council, que llevó al gobierno británico a una drástica reducción de su apoyo a la IA. Por su parte, la Defense Advanced Research Projects Agency (DARPA) norteamericana, un activo promotor de la investigación en IA, también redujo considerablemente su apoyo en los 80. Otras grandes iniciativas públicas relacionadas con la IA tampoco consiguieron sus objetivos. El ejemplo más destacado es el proyecto japonés del ordenador de quinta generación, entre cuyos objetivos estaba el desarrollo de ordenadores paralelos basados en el lenguaje Prolog y capaces de un proceso inferencial sobre bases de conocimiento masivas. Dados los éxitos previos de la industria japonesa en electrónica de consumo y automoción, el proyecto generó una cierta aprensión en otros países, que adoptaron iniciativas similares. Sin embargo, tras 10 años y una inversión pública y privada considerable, el proyecto fue cancelado en 1992 muy lejos aún de sus objetivos iniciales. Tampoco tuvieron mucha mejor suerte las distintas empresas que intentaron trasladar al mercado los logros y promesas de la IA de la época. Muchas de ellas surgieron del mundo académico; a su vez, las principales empresas informáticas de la época como IBM, DEC, Fujitsu o Sun también crearon unidades de negocio dirigidas a la explotación de la IA. Sin embargo, ninguna de estas iniciativas tuvo éxito. En cierta medida ello se debió a la dificultad intrínseca de toda compañía spin-off para adaptar su know-how científico o tecnológico a las realidades de cualquier mercado, mucho mayor en aquella época, donde el propio concepto de empresa spin-off era casi tan nuevo como las tecnologías que se intentaba explotar. Pero probablemente la principal razón estaba en la inmadurez subyacente a las mismas. Un editorial de The Economist en agosto de 1992, titulado *Artificial Stupidity*, resumía muy bien el estado de opinión imperante. Comentando cómo en el concurso Turing de ese año un programa había engañado a los jueces por su gran habilidad en simular errores tipográficos, se preguntaba sobre el interés de un ordenador que tiene entre sus méritos el cometer faltas de ortografía. En realidad, para mediados de los 80 el mismo nombre de IA empezaba a ser un paraguas que, además del núcleo inicial de la IA simbólica, cubría áreas tan diversas como reconocimiento de patrones, lenguaje natural, robótica, cibernética o teoría de la computación. Poco a poco los grupos de investigación de estas áreas comenzaron a solicitar sus proyectos de investigación y las compañías a comercializar sus productos con sus nombres específicos, abandonando la etiqueta IA. El resultado final fue el denominado invierno de la IA, en el que la financiación y atención mediática se desvió a otras áreas. El conexionismo, una componente de lo que a veces suele denominarse IA subsimbólica, ocupó un lugar destacado entre estas.

4. LA VUELTA DEL CONEXIONISMO

De manera informal puede decirse que el conexionismo busca describir los procesos mentales mediante el funcionamiento conjunto de neuronas simples interconectadas. Frente al enfoque jerárquico o top-down de la IA simbólica, donde los sistemas se construyen a partir de un diseño y un conocimiento prefijados, el conexionismo propone un punto de vista emergente o bottom-up, en el que un sistema "aprende" según su "experiencia", entendida como una recepción constante de datos, cambiando si es preciso sus reglas de operación para poder alcanzar un cierto nivel de generalización.

Su inicio puede situarse en la década de los 40, con los trabajos de Warren McCulloch y Walter Pitts[10] sobre la implementación de la lógica de primer orden mediante circuitos de carácter neuronal y la publicación del libro *The Organization of Behavior* del psicólogo norteamericano Donald Hebb, con su hipótesis general sobre el establecimiento de conexiones cerebrales, que se reforzarían cuando dos neuronas se activan simultáneamente y se debilitarían en caso contrario. Aunque en la época la

capacidad de experimentar sobre neuronas era muy limitada, los trabajos pioneros de Edgar Adrian en el Reino Unido dieron una idea del funcionamiento esencial de las mismas. En particular, se sabía que su comunicación se establecía mediante series de potenciales de acción, esto es, disparos producidos cuando el estímulo provocado por la recepción en sus dendritas de los potenciales provenientes de otras neuronas superaba un umbral. La semejanza entre el modelo de neurona electrónica de McCulloch y Pitts y los diagramas neuronales de Ramón y Cajal (de cuyo premio Nóbel se cumplen 100 años en 2006) es obvio. Los trabajos de McCulloch y Pitts estaban más cerca de la lógica que de la neurofisiología y en ellos no hay mención de posibles mecanismos de aprendizaje. Aunque no es muy conocido, fue el propio Turing el primero en sugerir cómo llevarlo a cabo. En su informe *Intelligent Machinery*[11] describe diversos tipos de lo que llama máquinas no-organizadas, entre las que se encuentran las máquinas de tipo B, formadas por neuronas conectadas mediante unos dispositivos capaces de alterar el peso de sus interconexiones. Tras unos valores iniciales aleatorios, una tal red podría ser entrenada, en palabras del propio Turing, mediante el refuerzo de ciertas conexiones y la eliminación de otras, de acuerdo con principios similares a los de Hebb. De hecho, Turing llegó a mencionar que su principal interés en el desarrollo del ACE era poder simular en el mismo dichos sistemas, tal y como en gran medida se hace hoy día.

En los años 50 se formaron diversos grupos de investigación sobre el tema en el Reino Unido y los EE.UU., entre ellos uno en el MIT, promovido por el matemático Norbert Wiener, el principal promotor de la cibernética, al que se unieron McCulloch y Pitts. Pronto se escribieron programas con los que simular el funcionamiento de grupos de neuronas, pero el primer obstáculo encontrado por los modelos conexionistas fue la incorporación de algún proceso de aprendizaje. El primer algoritmo eficaz fue propuesto en 1957 por Frank Rosenblat[12], quien buscaba simular el comportamiento del ojo humano usando un modelo donde una imagen activaba diversas unidades de una retina que, a su vez, venían a excitar o inhibir un perceptrón, esto es, una neurona electrónica del tipo McCulloch-Pitts. Rosenblatt se centró en la capacidad de un tal perceptrón para la detección de determinados patrones. En su forma más simple, un perceptrón sólo puede clasificar sus entradas si las mismas son separables en dos clases de manera lineal. Planteado así y prescindiendo de cualquier idea de aprendizaje, un perceptrón separador puede obtenerse mediante la resolución de un sistema de ecuaciones. Sin embargo, el interés de Rosenblatt estaba en llegar al mismo desde una configuración inicial arbitraria mediante la presentación sucesiva de distintos patrones aleatorios, tras la que se modificaran los pesos de las conexiones retinales para reflejar el efecto del patrón en cuestión. Para ello Rosenblatt propuso su famosa y muy sencilla regla delta. Para su formulación, supondremos que cada patrón X tiene asociada una etiqueta y con valores ± 1; una vez recibido por el perceptrón, este produce una salida $p(X)$ con valores ± 1, que será correcta si coincide con la etiqueta del peceptrón, esto es, si $yp(X) = 1$, mientras que será errónea si $yp(X) = -1$. Si al recibir un nuevo X el vector actual de pesos W lo clasifica correctamente, el mismo no se cambia. Por el contrario, si la clasificación es errónea, W se incrementa en el vector yX. Es sencillo comprobar que los nuevos pesos $W' = W + yX$ verifican que $yW' \cdot X > yW \cdot X$, por lo que el nuevo perceptrón definido por W' está más cerca de una correcta clasificación de X . Aunque no es a priori claro que de esta manera se llegue a una clasificación correcta de todos los patrones, el celebrado teorema de Novikov[13] demostró que ello era efectivamente así. Pese a la extrema simplicidad de los perceptrones, este primer éxito causó una cierta sensación en la comunidad conexionista.

Pero es obvio que una máquina capaz de resolver únicamente problemas de clasificación lineal no va a ser muy útil. La forma más sencilla de verlo es observar que los problemas separables linealmente van a ser muy excepcionales. Por ejemplo, si consideramos tres puntos en un plano que no estén en una recta, cualquiera de sus ocho dicotomías, esto es, sus posibles divisiones en dos clases, es separable linealmente. Si los puntos son 4, va a haber dos divisiones no separables entre las 16 posibles (las mismas tienen un cierto interés, pues corresponden a la tabla de verdad de la operación lógica conocida como "O exclusivo" o XOR) y si los puntos son 6, la mitad de sus dicotomías serán todavía separables linealmente. Pero según el número de puntos aumenta (y un problema será tanto más interesante cuántos más datos estén involucrados), las dicotomías separables disminuyen

rápidamente. Dicho de otra forma, los problemas en principio asequibles al perceptrón serían aquellos cuyos datos tengan una dimensión comparable al tamaño de la muestra. Pero la "maldición de la dimensionalidad", esto es, las altas dimensiones, ha sido siempre algo a evitar estadística y computación. La razón es clara: por ejemplo, estimar de manera adecuada la distribución probabilística de unos datos requiere cierta proximidad entre los mismos. Sin embargo, según aumenta su dimensión, los datos se "alejan" de tal manera que los tamaños de las muestras necesarias crecen tan rápidamente que se hace simplemente imposible disponer de un volumen suficiente de datos. Por tanto, y a primera vista, parece ser que la utilidad práctica de los perceptrones va a ser escasa. Rosenblatt era un buen comunicador (a él se le debe el término conexionismo) y un trabajador incansable, y al poco tiempo había varios grupos en los EE.UU. trabajando en el tema. Sin embargo, la investigación sobre perceptrones pronto decayó considerablemente. Esto se atribuye en parte a la publicación de libro *Perceptrons*[14], escrito por Marvin Minsky y Seymour Papert dos destacados representantes de la IA simbólica (y potencialmente opuestos a un enfoque competidor), en el que tras un detallado análisis se criticaba la capacidad de los perceptrones para calcular ciertos predicados simples. Ya hemos visto que un perceptrón no puede computar la operación XOR; tampoco es capaz de reconocer la conexión de ciertas figuras o, simplemente, de identificar si ha recibido un número par o impar de entradas. La manera de superar estas dificultades era extender la capacidad expresiva de los perceptrones mediante la composición sucesiva de varios de ellos en los denominados perceptrones multicapa (que abreviaremos de aquí en adelante como pmcs). Frente al perceptrón simple, estos pmcs tienen capas ocultas, esto es, intermedias entre la entrada y la salida y su capacidad clasificadora aumenta considerablemente. Por ejemplo, es sencillo ver que un perceptrón con una capa intermedia es capaz de separar los puntos de un triángulo del resto del plano, y que un perceptrón con dos capas ocultas es capaz de hacerlo con cualquier región poligonal y, asintóticamente, con cualquier región plana razonable. El camino estaba claro: encontrar mecanismos generales de aprendizaje aplicables a estos pmcs. Sin embargo, en los 60 esto no se consiguió: la construcción de un pmc que resolviera un problema de clasificación dado sólo se podía hacer, a lo sumo, de una manera ad-hoc, lo que resultaba imposible en problemas de cierta complejidad. Aunque se ha querido ver un carácter malicioso a la crítica de Minsky y Papert, la imposibilidad de entrenar un pmc fue probablemente la causa principal de la pérdida de interés a finales de los 60 en los perceptrones y, en general, en el conexionismo,

Sin embargo, de manera simultánea al inicio del invierno de la IA, el conexionismo experimentó un resurgimiento espectacular tras lograr éxitos destacados. Por ejemplo, John Hopfield[15] mostró cómo construir memorias accesibles por contenido mediante neuronas artificiales. Si bien la organización y acceso a la memoria de un ordenador se hace básicamente mediante las direcciones numéricas de sus bytes, de manera parecida a cómo se accede a las casas de una calle, la memoria humana claramente no funciona así. Una analogía mejor es la memoria accesible por contenido: cuando reconocemos a una persona no lo hacemos porque la imagen que tenemos delante de nuestros ojos sea idéntica a la que pudimos ver años atrás, sino porque esta imagen antigua es de alguna manera evocada por la imagen reciente. En forma similar, Hopfield propuso un mecanismo de almacenamiento de un conjunto de imágenes básicas y de su recuperación a partir de versiones ruidosas de las mismas. A su vez, se consiguió introducir métodos eficaces de aprendizaje para la construcción de perceptrones multicapa[16]. La idea básica, muy sencilla, tenía dos componentes. La primera era reemplazar la función de Heaviside o de escalón usada en la salida de un perceptrón de Rosenblatt y cuyo análisis matemático es muy difícil, por una aproximación diferenciable adecuada, como la función sigmoidal o la tangente hiperbólica. La segunda idea consistía en asociar a cada patrón un objetivo numérico e introducir un "maestro" que guíe el aprendizaje buscando minimizar para cada patrón la diferencia entre su objetivo y la salida del pmc. Por ejemplo, en el caso del XOR podemos asociar a los patrones $(0,0)$ y $(1,1)$ el objetivo 0 (su valor en la tabla de verdad del XOR) y el objetivo 1 a los patrones $(1,0)$ y $(0,1)$. De manera más precisa, el conjunto actual de pesos W de una red neuronal define una función de transferencia $F(X,W)$ que calcula la salida del perceptrón frente a un patrón concreto X. La diferencia entre esta salida y el objetivo y asociado a X permite definir una función de error $e(W)$ (típicamente el error cuadrático $e(W) = E\left[\left(y - F(X,W)\right)^2\right]$) que mide si los pesos actuales son adecuados. Cuando el error sea suficientemente pequeño, el pmc habrá aprendido el concepto buscado.

Dado que el error global ya es diferenciable, su minimización puede abordarse con métodos estándar de optimización. Sin embargo, este enfoque no tendría plausibilidad biológica y, además, frecuentemente no suele ser posible en la práctica. La alternativa es un aprendizaje patrón a patrón, tal y como se hace con el perceptrón de Rosenblatt. En el mismo, el valor actual W_{ij} del peso que conecta la unidad i con la unidad j de la capa siguiente se modifica tras recibir un patrón X de acuerdo con la fórmula $W'_{ij} = W_{ij} - \rho\, \partial_{ij} e(X, W)$, donde ρ es un factor de escala conocido como tasa de aprendizaje y $e(X, W) = \big(y - F(X, W)\big)^2$ es el error local asociado al patrón (X, y). La regla delta de Rosenblatt viene a ser un caso particular de la fórmula anterior, aunque para otra función de error. Visto así, el entrenamiento de un pmc es un ejemplo del uso de técnicas de aproximación estocástica. El algoritmo de cálculo de las derivadas parciales del error local se conoce como el método de retropropagación (backpropagation) de errores que, por extensión suele también aplicarse al proceso general de entrenamiento de pmcs. A la vista de su sencillez, cabe preguntarse por qué el algoritmo anterior no se popularizó antes. Entre las posibles respuestas está su relativamente alto coste computacional: tras su declive, el conexionismo no tenía acceso a grandes máquinas y, simplemente, hubo que esperar a que los ordenadores "normales" alcanzaran la potencia suficiente para realizar los experimentos pertinentes.

Los nuevos pmcs se aplicaron de manera inmediata a múltiples problemas, alcanzado éxitos notables. Dos de ellos merecen un breve comentario. El primero es la construcción por Gerald Tesauro de TD- Gammon, un pmc capaz de aprender a jugar al backgammon mediante técnicas de aprendizaje por refuerzo. Se recuerda que uno de los éxitos de la IA simbólica fue el desarrollo de programas que jugaban a las damas y al ajedrez. Aunque muy potentes, sus capacidades lógicas no eran "propias" sino que se incorporaban por sus programadores. Por el contrario, TD-Gammon sí era capaz de desarrollar estrategias propias aprendidas jugando contra sí mismo y, con la adición de ciertas reglas, alcanzó un alto nivel competitivo. Otro pmc de interés es NetTalk, desarrollado por Charles Rosenberg y Terry Sejnowski, capaz de aprender las reglas de la fonética inglesa, en la que una misma letra o incluso una misma palabra pueden tener muy distintas pronunciaciones en función de su contexto. El interés de Rosenberg y Sejnowski no estaba sólo en esta capacidad sino en analizar las activaciones formadas en las neuronas ocultas del pmc, donde descubrieron agrupaciones que, por ejemplo, reflejaban una distinción entre consonantes y vocales. En su momento esto reforzó la plausibilidad biológica de los pmcs que, sin embargo, pronto se desestimó. En cualquier caso, estos y otros muchos éxitos dieron lugar a una auténtica explosión de interés y a múltiples propuestas de redes neuronales, unas relacionadas con los pmcs, como las redes de funciones de base radial, los perceptrones recurrentes, o las redes autoasociativas, otras con las redes de Hopfield, como las máquinas de Boltzmann y otras, en fin, totalmente independientes, como por ejemplo los mapas asociativos de Teuvo Kohonen o las redes ART de Stephen Grossberg. Estas y otras técnicas de origen conexionista se han aplicado con éxito a una gran variedad de campos: control de vehículos y procesos, reconocimiento de objetos y caras, detección de objetivos, visión artificial, reconocimiento del habla y de caracteres, diagnóstico médico, aplicaciones financieras, control de intrusiones e incluso sistemas anti-spam.

Un ámbito de uso particularmente relevante es la minería de datos. La constante informatización de todo tipo de procesos ha puesto a disposición de las empresas y organizaciones grandes volúmenes de datos. Aunque inicialmente desperdigados por diferentes sistemas corporativos bajo modelos o formatos muy dispares, las nuevas tecnologías de Data Warehousing, o almacenamiento de datos, han permitido su consolidación y acceso de manera uniforme. Al ser reflejo en última instancia de su actividad, las entidades ven en esos datos una considerable riqueza potencial para mejorar sus procesos y operaciones. Sin embargo, una aproximación top-down a su explotación puede ser muy difícil: no se conocen a priori asociaciones entre variables, los volúmenes suelen ser demasiado grandes para una codificación explícita o, si se hiciera, el carácter dinámico de los datos obligaría cada poco tiempo a un rediseño más o menos amplio. Por el contrario, las redes neuronales son capaces de una extracción semiautomática de relaciones ocultas entre datos y su carácter adaptativo les permite mantener su eficacia frente cambios temporales. En sentido estricto, estas características no son exclusivas de las redes neuronales: hay un amplio conjunto de técnicas, englobadas bajo la denominación de aprendizaje automático, que también las poseen. En general, un

modelo de aprendizaje automático es capaz de procesar autónomamente datos o estímulos exteriores para "aprender", esto es, modificar su estructura o datos internos, de manera que su rendimiento posterior mejore. Incluso en este muy amplio campo, las buenas propiedades de modelización y de generalización de los pmcs les han permitido mantenerse de manera continuada entre las opciones más eficaces. Pero la minería de datos hace también evidentes sus limitaciones. Con frecuencia se juntan en un problema típico dos características extremas: el gran número de datos a tratar y la alta dimensionalidad de los mismos. A su vez, la dificultad de un tal problema suele requerir modelos de alta capacidad expresiva, lo que en el caso de los pmcs supone varias capas ocultas con un alto número de unidades. En esas circunstancias, el mero almacenamiento de la red y el cálculo del gradiente de la función de error pueden tener un coste prohibitivamente alto en muchos problemas de interés. Consideremos por ejemplo la detección de fraude en tarjetas de crédito. En España se procesan al día unos tres millones de operaciones lo que supone unos mil millones de operaciones al año, que potencialmente deberían tenerse en cuenta a la hora de construir un sistema de detección de fraude. A su vez, dada la abundancia de información disponible sobre el cliente o el comerciante, es razonable utilizar un alto número de variables y redes con varias capas con otras tantas unidades en cada una. La complejidad resultante hace que la detección del fraude esté ceca del límite de áreas abordables mediante pmcs.

Otras están, al menos hoy en día, ya fuera de su alcance. La más clara es la minería masiva de texto, entre cuyos objetivos está su clasificación o su ordenación por relevancia. El volumen de datos a manejar dependerá del ámbito de aplicación, pero sólo hay que pensar en la World Wide Web (WWW) para hacerse una primera idea: una estimación de 2005 cifra en unos 11.000 millones el número de páginas en la web "visible", esto es, la parte de la WWW accesible por los motores de búsqueda como Google. Un modelo habitual para representar la información en un documento es caracterizarlo por las frecuencias en el mismo de las palabras de un diccionario controlado. La dimensión de un documento es pues el tamaño del diccionario, que fácilmente puede tener varios centenares, si no miles, de palabras. Se hace pues necesario buscar técnicas que combinen capacidad expresiva con simplicidad computacional. En la discusión previa de los perceptrones de Rosenblatt ambas propiedades se nos presentaban en términos antitéticos, pero uno de los avances más importantes en aprendizaje automático, las máquinas de vectores soporte[17] han conseguido superar dicha dificultad. La idea base es muy sencilla: un clasificador lineal sólo será eficaz si trabaja sobre datos de muy alta dimensión, pero siempre que no tenga que trabajar de manera explícita sobre dicha dimensión. Por otra parte, el teorema de Mercer, un resultado matemático clásico, muestra que se puede definir un producto escalar en un espacio de dimensión potencialmente infinita a partir de un núcleo positivo $K(x,z)$ que trabaje con patrones de dimensión finita. Más concretamente, hay una proyección $X = F(x)$ tal que $X \cdot Z = F(x) \cdot F(z) = K(x,z)$. A su vez, es fácil ver que, usando una representación adecuada de sus pesos, el aprendizaje de tales modelos puede llevarse a cabo exclusivamente mediante el cálculo de productos escalares. La combinación de ambas ideas permite construir modelos sobre vectores X en espacios de dimensión potencialmente infinita mediante cálculos sobre los patrones x originales. Con el añadido a lo anterior de una búsqueda de márgenes separadores óptimos, que aseguren una buena generalización, las máquinas de vectores soporte son un ejemplo representativo del estado del arte en aprendizaje automático.

5. LA INTELIGENCIA ARTIFICIAL EN 2006

Estas últimas consideraciones pueden llevar a preguntarnos por la situación actual de la inteligencia artificial. Una tal pregunta siempre es complicada pero más aún en un campo tan amplio y fluido. Sí puede decirse que el conexionismo ocupa un lugar central, al menos si entendemos "central" más en su connotación geográfica o posicional que en la de fundamental. Un ejemplo es su papel intermedio entre la neurociencia y la IA. Aunque sorprendente a primera vista, y salvo excepciones como los modelos de David Marr, apenas hubo relación entre ambas hasta la llegada del conexionismo. Preguntado por ello, Allen Newell contestó que la neurociencia de los 50 o 60 estaba aún muy lejos de ofrecer modelos suficientemente detallados. Como hemos visto, los primeros modelos conexionistas

intentaban paliar esta situación, pero, aunque en un principio los pmcs se propusieron como modelos del cómputo que tiene lugar en las columnas corticales o las redes de Hopfield como una aproximación a la memoria, estas ideas pronto se abandonaron. Pero en estos 50 años también ha habido un enorme avance en neurofisiología y neurociencia, al que han contribuido sobremanera nuevas técnicas de registro tanto de la respuesta funcional conjunta de grupos de neuronas como de la actividad de neuronas individuales. Hay en esta segunda un hecho destacado, la presencia de respuestas estereotipadas de un tipo todo o nada. Como hemos visto, la respuesta de las neuronas de un pmc toma valores en un continuo matemático. En contraste, los potenciales de acción producidos por las neuronas biológicas reflejan básicamente una computación 0-1. Esto obliga a cambiar el punto de vista: ya no importa cuál es el valor de una respuesta neuronal sino en qué momento o con qué frecuencia se ha producido. El tiempo juega pues un papel fundamental en la computación neuronal, lo que no ocurre en los modelos tradicionales de cómputo, pero sí, por ejemplo, en la música: no es posible emocionarse con una sinfonía si uno se centra en cuáles fueron las notas; lo importante es saber en qué momento se tocaron. El papel del tiempo en el cómputo neuronal no está aun totalmente entendido. Desde los primeros experimentos se sabe que la frecuencia de los disparos de una neurona aumenta según se alarga en el tiempo el estímulo que los produce, lo que parece indicar que dicha frecuencia es la magnitud relevante. Esta ha sido durante mucho tiempo la hipótesis dominante en neurociencia computacional, pero más recientemente se han ido revelando como igualmente importantes los tiempos concretos en los que se producen los disparos. De hecho, hay sistemas neuronales cuyo comportamiento se explica mejor por una idea en detrimento de la otra, otros que parecen usar ambas en función de los estímulos recibidos e, incluso, otros que no parecen obedecer a ninguna de las dos. La decisión entre ambos modelos (o un tercero) es sólo el primer problema; un segundo muy importante y relacionado con el anterior, es el de la codificación neuronal, esto es, determinar cuál será la respuesta neuronal a un estímulo y saber qué indica una serie de disparos neuronales sobre los estímulos que la han producido. Naturalmente, todo lo anterior está relacionado con el funcionamiento coordinado de conjuntos de neuronas, así como con los mecanismos detallados de establecimiento de conexiones. Otro elemento importante y todavía no bien entendido en el procesamiento neuronal es el alto nivel de recurrencia en las conexiones cerebrales. La recurrencia hace posible la retroalimentación de señales, típicamente un requisito necesario para la estabilidad de sistemas. Por otra parte, y volviendo a la teoría de la computación, algunos intentos recientes de superar la tesis de Church-Turing se basan en el uso de redes neuronales recurrentes.

El córtex humano es posiblemente el lugar más interesante para estudiar las cuestiones anteriores. Por ello no es sorprendente que el mismo sea el objeto de estudio en Blue Brain, el proyecto más ambicioso en neurociencia computacional hasta la fecha, abordado conjuntamente por IBM y el Brain Mind Institute de la Ecole Polytechnique Fédérale de Lausana. El primer objetivo de Blue Brain será la simulación del funcionamiento de una columna cortical, la estructura básica del córtex. De manera simplificada, una tal columna es un elemento cilíndrico de menos de un milímetro de diámetro y unos 3 milímetros de longitud que contiene unas 10.000 neuronas; se estima que en el córtex hay más de un millón de tales columnas. En Blue Brain se construirá una versión software biológicamente realista de una columna con 104 neuronas interconectadas tridimensionalmente mediante unas 107 sinapsis, simulando tanto estímulos como respuestas. Los modelos neuronales a usar serán también realistas y la propia columna ajustará sus conexiones de acuerdo con diversos algoritmos. Más adelante se quiere añadir simulaciones de grupos de columnas y de regiones amplias del cerebro, siendo el objetivo último de Blue Brain la modelización completa del cerebro. La potencia de cómputo necesaria para tales simulaciones sólo es posible mediante el trabajo con supercomputadores. Por ello se utilizará una versión avanzada de la máquina Blue Gene, originalmente concebida por IBM para el estudio de cuestiones en genómica y proteómica y que se encuentra entre las 10 más potentes del mundo. Con más de 36.000 chips individuales, cada uno de los cuáles tiene a su vez 32 procesadores independientes, Blue Gene tendrá más de un millón de unidades de proceso. Blue Brain tiene también su propia versión del test de Turing: conseguir que los patrones eléctricos generados por la columna electrónica sean indistinguibles de los producidos por una columna cortical real. Probablemente podamos comprobar si este test se cumple mucho antes de que ello suceda con el original.

Al otro lado del conexionismo nos encontramos con el resto de la IA. El auge conexionista de los 80 no estuvo exento de controversias que, al contrario de lo sucedido con el primer debate sobre la IA, centrado sobre la posibilidad o no de un comportamiento mecánico de la mente, esta vez enfrentaron a actores que habían compartido el mismo lado en la discusión anterior. Por supuesto, hubo discrepancias de corte filosófico pero la mayor fricción se produjo dentro de la propia comunidad científica de la IA. Como era de esperar, la IA simbólica presentó una fuerte oposición: si bien concedía a los modelos conexionistas la capacidad de representar estructuras simbólicas, negaba que la información cognitiva de los mismos fuera superior a la proporcionada por los modelos simbólicos. El debate fue en muchas ocasiones agrio. Por una parte, el mundo conexionista achacaba a la IA simbólica su travesía del desierto en los 70. A su vez, su despegue en medio del invierno de la IA simbólica y la subsiguiente pérdida de apoyo oficial llevó la controversia más allá del ámbito meramente científico. Finalmente, el conexionismo se presentó frecuentemente como la alternativa a un paradigma relativamente agotado y tampoco fue capaz de escapar de la tentación de la hipérbole a la hora de proclamar sus méritos y promesas. Así, la agenda conexionista planteaba no sólo encontrar modelos de computación neuronal o explorar mecanismos del proceso cerebral de la información sino encontrar también métodos mejores de resolver los problemas de la IA.

Sin embargo, con el paso de los años se ha ido produciendo un acercamiento entre los campos simbólico y conexionista, siendo frecuente que en un mismo grupo de investigación se usen técnicas provenientes de ambos. No hay que olvidar que los dos estudian problemas en gran medida comunes y que los logros respectivos han sido grandes. Hay también áreas claramente fronterizas. Una son los árboles de decisión: planteados inicialmente dentro de la IA simbólica como representaciones gráficas de razonamiento lógico, pronto adquirieron una componente numérica que los ha situado claramente dentro del aprendizaje automático. Otro ejemplo son las redes bayesianas de Judea Perl, modelos gráficos que recogen dependencias causales entre variables aleatorias representadas por sus nodos. Su estructura de grafos dirigidos acíclicos permite efectuar un razonamiento lógico sobre las mismas. En estas redes es preciso tanto determinar el grafo concreto a usar como estimar las probabilidades condicionadas de un nodo respecto a sus padres, de los que depende causalmente. Ambos problemas suelen frecuentemente abordarse bajo la perspectiva del aprendizaje automático. La consecuencia de lo anterior es la situación actual no sólo de coexistencia sino de cooperación entre las dos escuelas de pensamiento de la IA. Incluso sus denominaciones han cambiado, Hoy en día no se habla tanto de IA sino de IA Convencional o incluso GOFAI, siglas de Good Old Fashioned Artificial Intelligence. Por su parte, los modelos conexionistas artificiales se han incluido dentro de lo que se conoce como Inteligencia Computacional y, como se ha dicho, hay áreas que solapan ambas denominaciones. Probablemente la razón más importante para esta nueva aproximación ecléctica a la IA es la complejidad de los problemas que la tecnología actual permite (y exige) abordar, que requieren el trabajo simultáneo en varios subproblemas que han de resolverse mediante la técnica más adecuada, probablemente distinta en cada caso. Hay muchos ejemplos de esta situación; la reciente Grand Challenge 2005 de la DARPA nos ofrece uno.

En 2004 la DARPA, sucesora de la agencia ARPA, responsable en su día del arranque de internet, celebró su primera Grand Challenge, una carrera sobre un circuito de unos 200 kilómetros de pistas en el desierto de Mojave entre vehículos-robot totalmente autónomos, sin permitirse ningún tipo de intervención manual. En esta primera convocatoria ningún vehículo fue capaz de recorrer más de 12 km. Sin embargo, en su edición del año 2005, cinco de los 23 vehículos que la iniciaron recorrieron sus 205 kilómetros en menos del tiempo límite de 10 horas. La carrera seguía un formato similar al de las contrarrelojes ciclistas, saliendo los vehículos con un intervalo de 5 minutos. El detalle de la ruta se mantuvo en secreto hasta dos horas antes del inicio de la carrera, cuando se distribuyó entre los participantes un CD con una lista de unas 2.900 coordenadas GPS junto con información sobre la anchura de la pista en cada punto y límites de velocidad. El vencedor fue Stanley, un Volkswagen Touareg especialmente acondicionado, que acabó la carrera en poco menos de 7 horas, con una velocidad media de 31 km/h. Stanley es el resultado de la colaboración de un amplio equipo de

científicos e ingenieros de Intel, Volkswagen y el SAIL, Stanford Artificial Intelligence Laboratory. El SAIL, líder del proyecto, desarrolló, entre otros, el software de navegación y control, que tiene tres módulos básicos. El primero es un módulo sensorial que recoge datos de 5 escáneres láser, un radar y una cámara en color, que son analizados por un segundo módulo perceptivo para, por ejemplo, detectar obstáculos o encontrar superficies transitables. Finalmente, un módulo de control determina en función de la información anterior cómo "conducir" el vehículo.

Stanley integra varias componentes típicas de la IA. Particularmente interesante es el módulo que "encuentra" la vía a distancias largas. Los cinco láseres de Stanley detectan obstáculos a corta y media distancia (hasta unos 22 metros). Para ello adquieren de manera continua una nube tridimensional de puntos que se clasifican como libres, ocupados o desconocidos en función de la diferencia de alturas entre puntos próximos. Estas medidas básicas deben ser modificadas para corregir errores causados por una incorrecta estimación de la posición y orientación de Stanley, lo que se hace mediante un modelo de Markov, y el resultado final es una "visión" del terreno cercano suficientemente detallada como para alcanzar velocidades de unos 35 km/h. Sin embargo, las bajas velocidades impuestas por las características de mucha parte del recorrido hacen imprescindible para ganar la carrera alcanzar en otros puntos velocidades de hasta 60 km/h. Esto obliga a detectar tanto la ubicación de la carretera como posibles obstáculos en ella a distancias superiores a los 70 metros. Para ello Stanley usa la información de la cámara, en cuyas imágenes se han de localizar zonas libres u obstáculos. Sin embargo, mientras que la resolución del láser es suficiente para que el software de Stanley clasifique directamente puntos a corta distancia nunca vistos antes, esto no es posible sobre las imágenes de la cámara.

La solución del equipo de Stanford es a la vez ingeniosa y natural dentro del aprendizaje automático e ilustra tanto la potencia de los métodos neuronales como su flexibilidad. La observación básica es que las imágenes lejanas de la cámara en un momento dado serán poco después imágenes cercanas, que los láseres de Stanley pueden analizar y clasificar. Pues bien, esta clasificación de imágenes cercanas se utiliza como objetivo para "aprender" de manera continua la correcta interpretación de las imágenes a larga distancia. El modelo usado para ello es una red neuronal de funciones radiales gaussianas, definidas sobre el espacio de color RGB de los píxeles de las imágenes lejanas. Stanley mantiene una biblioteca de gaussianas que actualiza de manera continua, bien mediante pequeños ajustes de las ya existentes para, por ejemplo, adaptarse a cambios de luminosidad, o bien reemplazando totalmente gaussianas anteriores por otras que reflejen mejor las cualidades de superficies nuevas, como sucede al pasar de una pista a una carretera. Se sigue pues un proceso continuo de actualización del modelo neuronal y de su aplicación a la imagen a larga distancia a procesar en cada momento. Si el análisis de la imagen lejana lo permite, Stanley acelera o mantiene su velocidad. Por el contrario, si se detecta en ella un posible obstáculo, Stanley disminuye su velocidad para que sus láseres lo puedan analizar en mayor detalle.

La variedad de las pistas del circuito, la complicación de algunos de sus tramos y el carácter de carrera dan un gran mérito a los vehículos que consiguieron acabarla. Sin embargo, si bien los vehículos de la Grand Challenge debían superar tanto obstáculos naturales como artificiales, puestos por la DARPA, no tenían en cambio que preocuparse por el tráfico. Con toda su dificultad, está claro que las pistas del desierto son un entorno más fácil que las autopistas de Los Angeles o las calles de Madrid. En 2007 podremos ver cómo se desenvuelve Stanley en ellas: la DARPA ha convocado una nueva Grand Challenge donde los vehículos competirán en una ruta urbana de unos 100 km, que habrán de cubrir en menos de 6 horas circulando entre otros vehículos y obedeciendo las normas de tráfico. Frente al carácter predominantemente sensorial de la Grand Challenge de 2005, en 2007 los vehículos tendrán también que evaluar el comportamiento de otros y tomar decisiones inteligentes en tiempo real. Será pues un buen momento de evaluar qué puede decir la IA al respecto.

6. BALANCE Y FUTURO

No puede discutirse que la percepción sensorial y capacidad de aprendizaje de Stanley se traducen en un comportamiento bastante más inteligente que el de muchos conductores. Pero aun siendo un éxito notable de la IA contemporánea, casi nadie le concedería capacidades cognitivas cercanas a las humanas. Esto nos lleva a preguntarnos sobre el futuro de la IA. Por supuesto, una tal pregunta es sobre todo retórica y su finalidad suele ser dar pie a quien la formula para efectuar algunas predicciones más o menos especulativas, lo que a su vez conlleva un riesgo. Sin embargo, y por suerte para el ponente, justamente en 2006 se han cumplido 50 años de la primera Conferencia Dartmouth sobre IA, lo que, naturalmente, se ha celebrado con una segunda conferencia en el Dartmouth College titulada AI@50, esto es, la inteligencia artificial a los 50 años. Y como cabía esperar, el futuro ha sido una cuestión dominante en las comunicaciones presentadas. Pero ya no se trata tanto del futuro de la IA en sí, sino de un abanico muy amplio de futuros, entre los que están los del pensamiento, el razonamiento, el aprendizaje, la visión, el lenguaje, las redes neuronales o, incluso, los juegos de ordenador, a cuyos actores la IA intenta dar un comportamiento menos mecánico y más realista. Como no hay lugar ni tiempo para su detalle, es mejor optar por un resumen, a lo que de nuevo ayuda la conferencia: como también era previsible, se organizó una encuesta entre los más de 400 asistentes. Algunos de sus resultados son esperables; otros, más sorprendentes. Entre los esperables se puede mencionar que el 60 % estaba muy de acuerdo en que la IA debe seguir un enfoque multidisciplinar en el que participen la estadística, el aprendizaje automático, la lingüística, la psicología cognitiva, la biología e incluso la filosofía, además de, naturalmente, las ciencias de la computación. Un 79 % estaba al menos parcialmente de acuerdo con la necesidad de combinar el aprendizaje automático con un razonamiento deductivo también automatizado, mientras que el 59% considera al conexionismo prometedor y un 79% cree imposible mayores avances sin nuevos descubrimientos sobre el funcionamiento del cerebro. Más sorprendente es que un 41% cree que harán falta más de 50 años para simular completamente la inteligencia humana, mientras que otro 41% cree que no se logrará nunca.

Lo cierto es que la IA sigue viva. El mismo The Economist que en 1992 hablaba de estupidez artificial, nos informaba 10 años más tarde que el campo había vuelto a ganar una "respetabilidad discreta", lograda sobre sistemas modestos en sus proclamaciones, pero notables en sus logros. Hay muchas razones para esta nueva vitalidad. En mi opinión, una muy relevante es la aplicación de un enfoque más concreto, más ingenieril, a los problemas a resolver, en el que, manteniendo grandes principios rectores, los obstáculos se van superando uno a uno y, una vez vencidos, se convierten en puntos de apoyo para nuevos avances. Un antecedente, en parte inesperado, de este punto de vista es Leonardo Torres Quevedo. Ingeniero excepcional y admirador de Babbage y de su máquina analítica, Torres Quevedo es muy conocido por sus obras civiles y como diseñador y constructor de ajedrecistas mecánicos y de calculadoras electromecánicas muy sofisticadas para su época. Estas últimas aportaciones, así como algunos de sus escritos han llevado a ver en él a un precursor de la IA. Probablemente ello sea exagerado, pues más allá de su mención a Babbage, la idea del ordenador como máquina de cómputo universal no suele aparecer en sus escritos. Lo que no tiene discusión es su papel destacado en la creación de la Automática y su clarividencia en el tema. La siguiente cita es de un trabajo suyo de 1915:

> Además, se necesita [...] que los autómatas tengan discernimiento, que puedan en cada momento, teniendo en cuenta las impresiones que reciben, y también, las que han recibido anteriormente, ordenar la operación deseada. Es necesario que los autómatas imiten a los seres vivos, ejecutando sus actos con arreglo a las impresiones que reciben y adaptando su conducta a las circunstancias.[18]

Noventa años más tarde, es difícil no ver a Stanley en estas palabras, que no proclaman que los autómatas piensen o logren objetivos sólo accesibles mediante el pensamiento, pero sí que progresivamente pueden hacer cada vez más cosas que popularmente se clasifican como inteligentes. Estas ideas sugieren un camino para avanzar en la búsqueda eficaz de sistemas inteligentes que, sin mayores pretensiones, tal vez sea interesante desarrollar. Voy a iniciarlo con una cita sobre la IA de Edsger Dijkstra, uno de los fundadores de la ciencia de la computación y receptor del premio Turing

en 1972. Quienes conozcan sus escritos puede que estén sorprendidos, pues muchas de sus opiniones son, como mínimo, provocativas. En la cita en cuestión, Dijkstra afirmó que:

> The question of whether a computer can think is no more interesting than the question of whether a submarine can swim.

A primera vista su significado es ambiguo y no revela la opinión de Dijkstra sobre la IA, pero si queremos buscarle un sentido positivo (lo que tal vez no fuera su intención), Torres Quevedo y Stanley nos pueden ayudar. Cambiemos submarino y nadar por avión y volar. En 2003 se cumplieron 100 años del primer vuelo de los hermanos Wright. Puede que intentaran emular el vuelo inaccesible y majestuoso del águila. En 1903 no lo consiguieron, pero hoy no se puede negar cierta majestuosidad al Concorde. A su vez, ni el Concorde es comparable a un águila ni, a pesar de su belleza, tampoco ésta es comparable a la mente humana. Además, puede que una aproximación paso a paso a la inteligencia quite a la empresa mucho de espectacularidad e, incluso, algo de grandeza: con todo su mérito, una cosa es Stanley y otra el teorema de Gödel. Por otra parte, incluso es posible que en algún momento la IA fuerte se demuestre inalcanzable. En un comentario sobre sus primeros 50 años, Maurice Wilkes[19], pionero del diseño de ordenadores y también premio Turing, señala que ello sería positivo pues evitaría muchos esfuerzos en direcciones equivocadas, al igual que el segundo principio de la Termodinámica ha ahorrado mucha energía que se hubiera gastado inútilmente en la búsqueda del móvil perpetuo. Sin embargo, con la amenaza del cambio climático (y el petróleo a 70 dólares), estamos obligados a seguir disminuyendo el impacto de dicho principio. En forma paralela, una demostración de la imposibilidad de la IA fuerte tendría, de entrada, implicaciones tanto o más trascendentes que las del segundo principio o el teorema de Gödel, pero, en cualquier caso, Stanley y todos los logros concretos ya alcanzados y los que sin duda se alcanzarán seguirán ahí. Además, a esta nueva forma de avanzar de la IA se le puede ver un matiz que tal vez Dijkstra hubiera apreciado.

Así como las Matemáticas tienen sus problemas emblemáticos, como la conjetura de Poincaré o la hipótesis de Riemann, hay también en computación un problema particularmente señalado, decidir si la clase P, que informalmente es la de los problemas fáciles de resolver, coincide con la clase NP , aquellos para los que, si bien obtener una solución puede tener un coste computacional prohibitivo, es sencillo comprobar que una cierta solución efectivamente lo es. Por supuesto, esta definición no es tal y está muy lejos de hacer justicia al problema. No es este el sitio para una precisión mayor, pero podemos hacernos una idea de su importancia por su presencia, millón de dólares incluido, en los Millennium Problems que la reciente resolución por Perelman de la conjetura de Poincaré ha hecho famosos. La opinión general es que P está contenido estrictamente en NP y que, por tanto, va a haber problemas cuya solución exacta estará en la práctica irremediablemente fuera de nuestro alcance. La relación P = NP no es en absoluto una cuestión meramente académica. Hay en NP un gran número de problemas de enorme interés aplicado, por más que sean en la práctica insolubles para entradas de gran tamaño. Un ejemplo es el problema del viajante, que busca determinar el recorrido más corto entre un grupo de puntos: cualquier empresa de reparto ha de resolver muchos problemas del viajante cada día. Pero, si el coste computacional de resolverlos es tan grande, ¿cómo lo hacen? Pues básicamente de igual forma a como se hace en ciencias de la computación: encontrando una solución aproximada que, si bien no garantice un recorrido óptimo, sí permita acabar el reparto y satisfacer a los clientes en un tiempo razonable. La bondad de un algoritmo aproximado suele medirse por el cociente λ entre el valor de su solución y el óptimo. Dicho λ es por definición mayor que 1 y un algoritmo aproximado es tanto mejor cuanto más pequeño es su λ. Pues bien, estos primeros 50 años de IA pueden verse como una búsqueda de soluciones aproximadas al gran problema del comportamiento inteligente. Por supuesto, se trata de una afirmación metafórica: la IA fuerte no va a estar en NP sino, si acaso, mucho más allá. Pero, siguiendo con la metáfora, la aproximación a la IA lograda hasta ahora presenta un λ muy lejano del 1 ideal o, si se quiere, hemos quitado muy pocas capas a la cebolla de Turing. Pues bien, incluso si la meta, aunque grandiosa, al final resulta inalcanzable, no por ello la jornada recorrida habrá sido menos elevada o poco provechosa.

Vamos a acabar en una nota más informal volviendo al test de Turing y, tangencialmente, al futuro de la tecnología. Ya hemos dicho que no es un tema que hoy concentre grandes esfuerzos de investigación, pero sí sigue atrayendo mucha atención: si buscamos en Google la frase "Turing test" encontraremos más de un millón de páginas, a veces en lugares algo insólitos, como el sitio www.longbets.org del proyecto Long Bets. Se trata de una serie de apuestas más o menos serias, generalmente sobre cuestiones científicas o tecnológicas. Entre las menos serias, está una sobre cuándo los EE.UU. ganarán la copa del mundo de fútbol; entre las más serias, el físico Freeman Dyson apuesta sobre dónde encontraremos ejemplos de vida extraterrestre. La de mayor importe es una sobre la superación del test de Turing antes del año 2029. Mitch Kapor, el creador de la hoja de cálculo Lotus 1-2-3 y cofundador de la Electronic Frontier Foundation, apuesta 10.000 dólares en contra y Ray Kurzweil otros 10.000 a favor. En la página web de Long Bets pueden verse sus argumentos; no son largos y su lectura es interesante. No nos vamos a detener en ellos, pero sí en la figura de Kurzweil. Se trata ciertamente de un personaje de mérito tecnológico y empresarial. Entre otras actividades, desarrolló el primer sistema de reconocimiento de caracteres multifuente y fundó una compañía que fabrica los sintetizadores de sonido más sofisticados (y caros). Ha recibido múltiples premios, como el Lemerson-MIT, el premio más importante para inventores, o la National Medal of Technology norteamericana. Pero probablemente sea más conocido por sus predicciones futuristas. Su capacidad predictiva no debe echarse en saco roto: en 1990 predijo la explosión de la World Wide Web y la inminencia de un campeón mundial de ajedrez informático, y Bill Gates le ha descrito como el mejor predictor sobre IA. La idea central en la predicción de Kurzweil sobre el test de Turing es su creencia en que la tecnología está gobernada por una ley de rendimientos acelerados que pronto alcanzará un desarrollo superexponencial y acabará en una singularidad tecnológica de consecuencias portentosas. Los argumentos de Kurzweil son sugestivos, pero en ocasiones no del todo convincentes. Por ejemplo, su fe en el recorrido de la ley de Moore no es claramente extrapolable a otras tecnologías e, incluso, puede ser demasiado optimista en un contexto puramente informático. Aunque frecuentemente se expresa en términos de potencia de cómputo, dicha ley describe en realidad la densidad creciente del empaquetamiento de transistores. Al ritmo seguido hasta ahora, se cree que en unos 15 o 20 años dicho empaquetamiento estará cerca de sus límites físicos y que la ley se detendrá entonces. Esto no preocupa a Kurzweil, quien prevé que la computación tridimensional tomará su lugar. Sin embargo, puede que la ley haya alcanzado ya otros límites como, por ejemplo, económicos. Se podría enunciar una especie de segunda ley de Moore, al afirmar que duplicar la densidad de transistores implica que el coste de producir el nuevo chip también se duplica. El coste actual de las nuevas fábricas de Intel o AMD se estima en unos 2.500 millones de dólares, que sólo puede compensarse si hay demanda suficiente. La vía tradicional de lograrlo es mediante la obsolescencia de la generación anterior de ordenadores. Sin embargo, y aunque cabe esperar que, como otras veces, la industria informática ayude, no es tan fácil hacer obsoleto el PC con un GB de RAM, 150 GB de disco duro y un chip de 1,5 GHz que compramos en un hipermercado por menos de 1.000 euros. De hecho, los nuevos chips que Intel y AMD están introduciendo ya no buscan sólo aumentar la capacidad de proceso mediante mayores densidades, sino también a través de la incorporación en un mismo chip de varios núcleos que trabajen de manera paralela. Si bien esto supone un cierto frenazo a la ley de Moore puede que, por otra parte, resulte a medio plazo en una nueva aceleración en la potencia de los ordenadores. En efecto, una asignatura pendiente de la informática es el proceso paralelo, donde los avances no han sido tantos como en otras áreas. Sin duda la disponibilidad de chips con múltiples núcleos estimulará nuevas arquitecturas de proceso paralelo y, por tanto, ordenadores más potentes.

Así pues, puede pensarse que el viento tecnológico seguirá soplando a favor de Kurzweil, aunque éste necesita mucho aire en sus velas. En efecto, en su concepción de las cosas, la superación del test de Turing será un hito menor, mero corolario de profecías tecnológicas mucho más trascendentales. Entre las más espectaculares esta su creencia que, en un futuro no muy lejano, millones de nanobots del tamaño de los glóbulos rojos circularán por nuestro cuerpo reparando arterias, huesos o, por qué no, el cerebro, y será posible mejorar nuestro código genético mediante descargas desde la web. En opinión de muchos, esta predicción de inmortalidad no da precisamente solidez a, por ejemplo, la relativa al test de Turing. No vamos a entrar en esta cuestión, pero en un tono ligero, no

sería descabellado pensar que, si en 2029 el test se supera, Mitch Kapor pagará gustoso los 10.000 dólares de una apuesta perdida, siempre que Ray Kurzweil tenga igual acierto en sus demás predicciones. Algo más seriamente, cabe preguntarse quién ganará. Mi opinión es que lo hará Mitch Kapor, pero también creo que esa es la pregunta fácil. La pregunta de mérito, la realmente difícil, es qué nos ofrecerá la tecnología en el año 2029. Voy a parar aquí, dejándola sin respuesta. Por supuesto, no la sé, pero tampoco falta tanto, no ya para saberlo, sino para verlo y vivirlo.

7. EPÍLOGO

Una vez llegados aquí es muy probable una sensación de que muchos de los logros en 2006 ya han sido ampliamente superados. Y es que, aunque el tango diga que veinte años no es nada, los dieciséis transcurridos entre 2006 y 2022 han sido más que suficientes para haber experimentado una crisis económica descomunal, sufrir la peor pandemia en 100 años, observar cómo el cambio climático avanza de manera cada vez más amenazadora y, de postre, sufrir en Europa una guerra entre dos naciones soberanas por primera vez desde 1945, que ya (y de momento) nos ha sumergido en una nueva crisis económica y energética.

Pero nada de eso ha sido obstáculo sino en muchos casos acicate para un constante progreso tecnológico en general y, muy particularmente, en computación. Como simples ejemplos, en 2006 no había smartphones (el iPhone debutó en 2007) y, salvo por unos pocos en Silicon Valley, tampoco se les esperaba. Naturalmente, Apple era ya un actor reconocido, pero muy lejos de convertirse en la compañía de mayor capitalización bursátil[20] que es ahora. Google (ahora Alphabet) y Facebook (ahora Meta) ya existían, pero también estaban muy lejos de sus actuales posiciones tercera y séptima en ese ranking. Un hecho clave para estas tres compañías fue el lanzamiento en 2010 de las redes 4G y seguro que las redes 5G que se asoman ahora producirán nuevos líderes. La quinta compañía del ranking, Amazon, que en 2006 mayormente vendía libros, lanzó en 2007 Amazon Web Services, abriendo al mundo la "cloud", la computación en la nube, y dando un paso esencial para llegar a ser el gigante tecnológico actual. Por su parte, los posibles líderes tecnológicos de 2006, IBM y Microsoft, sufrieron un cierto estancamiento, del que Microsoft se recuperó a partir de 2014 para ocupar ahora la tercera posición del ranking en gran medida gracias a la nube. Y Tesla, la sexta compañía ahora, era en 2007 una startup más bien pequeña que acababa de lanzar su primer vehículo eléctrico.

Podemos seguir enumerando muchos más cambios y avances, pero no es este el sitio, por lo que me voy a limitar a revisar algo de lo que ha pasado en el campo de la Inteligencia Artificial, en la IA. Algunas cosas han cambiado relativamente poco. Un ejemplo puede ser el test de Turing: aunque el millón de enlaces que una consulta en Google producía en 2006 se haya convertido en cerca de treinta millones en 2022, eso no implica que sea un área activa de investigación. De hecho, tras cambios y controversias varias, la última edición del premio Loebner tuvo lugar en 2019 y en la actualidad se halla suspendido. El ganador de las últimas cinco ediciones fue el chatbot Mitsuku-Kuki, con el que se puede interactuar: si se le pregunta por el test de Turing suele contestar que cree que su superación todavía está lejos, pero afirma que continúa trabajando en ello. Por su parte, la apuesta Long Bets sigue activa, pero, a juzgar por los cada vez menos frecuentes comentarios en su página web, el interés en la misma parece haber decaído bastante. Mi apuesta por Mitch Kapor sigue en pie, pero, más que nada, porque no es fácil detectar grandes esfuerzos para superar el test.

Sin embargo, otros avances han tenido un gran impacto. Probablemente el más grande haya sido el de las redes neuronales profundas o Deep Learning (DL)[21], una tercera generación de redes neuronales que surgió alrededor de 2010 y que a partir de 2012 logró éxitos muy importantes en áreas como la visión por ordenador o el reconocimiento del habla. En realidad, la idea básica de esta nueva generación no es muy distinta de la de las redes de los años 90. Ciertamente ha habido avances teóricos notables, pero sobre todo diría que gran parte de su éxito está justamente en los grandes avances en tecnología de cómputo. De entrada, y por supuesto, el gran aumento en capacidad de memoria y velocidad de cálculo[22], más aún con la incorporación de las GPUs, Graphical Processing

Units, que aceleran en un orden de magnitud las operaciones matriciales esenciales en el trabajo en DL. Pero sobre todo hay que señalar la enorme flexibilidad y complejidad de las arquitecturas neuronales de hoy en día. Su entrenamiento sigue basado en el descenso por gradiente y, por tanto, en el algoritmo de backpropagation. Pero ya no necesario derivarlo matemáticamente y programarlo ad hoc para cada diferente arquitectura neuronal. Al contrario, una vez definido el modelo neuronal (algo relativamente sencillo), basta con compilarlo, esto es, aplicar técnicas de cálculo diferencial simbólico para obtener automáticamente un módulo con el gradiente de la función de error, que se añade al resto del modelo para llegar a una red ya directamente ejecutable. Si a esto añadimos que el coste de entrenar un modelo DL crece de manera esencialmente lineal con el tamaño de la muestra, se llega a la situación actual donde los modelos DL no tienen rival en problemas Big Data, no tanto porque sean intrínsecamente mejores, sino porque los demás modelos de aprendizaje automático simplemente no se pueden aplicar.

De hecho, los logros del DL han llevado la IA al gran público. Así, estas redes están detrás de la muy buena traducción automática de Google Translate o de la transcripción de lenguaje hablado en texto que podemos hacer en nuestros móviles, y sus últimos triunfos han sido ampliamente recogidos por los medios de comunicación. Entre estos cabe citar los Generative Pre-trained Transformers (GPT), modelos de lenguaje introducidos por Open AI y que utilizan DL para producir textos similares a los de un ser humano, o el programa AlphaGo de la compañía DeepMind que, tras aprender jugando contra sí mismo, ganó en 2016 al mejor jugador del mundo de Go. Tras AlphaGo, DeepMind desarrolló AlphaZero, que no sólo ganó a AlphaGo sino que aprendió por sí mismo a jugar al ajedrez y venció a los principales programas informáticos (ganar a los humanos al ajedrez ya no tiene mérito). Ciertamente estos programas ya no se suelen ver como hitos para llegar a la inteligencia artificial fuerte y, de hecho, muchos pueden pensar que Gary Marcus, uno de los críticos más acérrimos del DL, tiene razón cuando titula su artículo sobre GPT3 como *GPT-3, Bloviator: OpenAI's language generator has no idea what it's talking about*[23]. Pero eso no quita que los GPT puedan generar crónicas periodísticas más que aceptables sobre pequeños eventos y, en sus últimas versiones, escribir incluso programas sencillos de ordenador.

Otra consecuencia es la reducción al aprendizaje automático, cuando no al DL, de casi todo lo que hoy en día se denomina IA, algo demasiado estrecho y hasta injusto. Pero, en cualquier caso, sí se ha renovado la conversación sobre la IA, donde se ha abandonado lo que podríamos llamar "pensamiento antiguo", de que bastaría apilar un gran número de reglas de decisión (en la IA simbólica) o de neuronas artificiales (en la IA conexionista) para que la inteligencia surgiera de manera poco menos que espontánea. Por el contrario, viene tomando fuerza una forma de pensar, que propone crear sistemas dotados de cierta inteligencia construyendo primero modelos especializados de aprendizaje automático distribuidos en módulos estratificados para desarrollar luego programas de supervisión que los hagan trabajar juntos. Un ejemplo claro son los sistemas de conducción autónoma, que aúnan sensores avanzados con modelos inteligentes de interpretación de datos, todo ello bajo el control de módulos globales de supervisión y decisión. Este enfoque, no tan lejos del aplicado en el ya muy superado Stanley, involucra al aprendizaje profundo, pero también a otras muchas otras tecnologías y es el seguido por los principales actores en conducción autónoma. Pero, aunque este problema parecía poco menos que pan comido hace unos pocos años, se han encontrado límites duros a lo que sería una conducción inteligente (y, de rebote, a la IA fuerte). De hecho, entre unos niveles que van del 0, que se corresponde a la conducción humana estándar de hoy en día, al 5 de una conducción inteligente completamente autónoma, el nivel 4 de una autoconducción supervisada por humanos se considera alcanzable, pero el nivel 5 no parece estar en absoluto cerca. Michael Jordan, profesor de la Universidad de California Berkeley y una de las grandes figuras del aprendizaje automático moderno, defiende en un reciente artículo[24] dar prioridad a la conjunción de este enfoque de "aumento de la inteligencia" con el desarrollo de una "infraestructura inteligente", derivada de la constante y omnipresente digitalización de cualquier actividad humana, en detrimento de un énfasis exclusivo en el desarrollo de la IA fuerte, o como él la llama, "human imitative intelligence". O, si se quiere, que debemos preocuparnos menos por "sueños de ciencia ficción o pesadillas super humanas" y más en comprender mejor estas tecnologías y guiar su desarrollo según se hacen cada vez más

presentes, influyentes y disruptivas. A un nivel más básico, podemos ver la aplicación práctica de estas ideas en la automatización constante de cada vez más tareas (y en el desplazamiento de las personas que las hacían hasta ahora).

Nos podríamos extender mucho más, pero, de nuevo, este no es el sitio y tenemos que acabar, para lo que voy a retomar la apuesta Long Bets sobre el test de Turing y la misma pregunta de 2006: qué nos ofrecerán tecnología, computación e inteligencia al final de 2029. Tampoco me atrevo a responderla hoy. Ciertamente las tres, junto con la automatización y digitalización de casi todo, traerán muchos avances y mejoras, pero con una nota de precaución: esos avances también van a trastocar la vida de muchas personas o, incluso, aumentar la desigualdad. Pero la urgencia de desafíos como el cambio climático va a requerir más tecnología, más computación y más inteligencia, no sólo artificial sino, también, y, sobre todo, natural. Por ello, y a la vista de, por ejemplo, cómo hemos superado la pandemia, y sin olvidar una cautela vigilante, las tres son también motor y motivo para el optimismo.

8. REFERENCIAS

1. El libro *Gödel's Proof* (New York University Press, 1958) de Ernest Nagel y James Newman, presenta de forma relativamente rigurosa pero accesible una demostración del teorema de incompletitud.
2. Una excelente biografía de Turing es *Alan Turing: The Enigma*, escrita por Andrew Hodges (Vintage, 1992)
3. Martin Davis ofrece una descripción relativamente sencilla de las máquinas de Turing en su libro *The Universal Computer* (W.W. Norton, 2000), así como del programa de Hilbert y del teorema de Gödel.
4. Alan Turing. *Computing Machinery and Intelligence*. Mind, vol. 59, 1950.
5. John R. Searle. *Minds, brains, and programs*. Behavioral and Brain Sciences vol. 3, 1980.
6. John R. Lucas. *Minds, Machines and Gödel*. Philosophy, vol. 16, 1961.
7. Roger Penrose. *The Emperor's New Mind*. Oxford University Press, 1989.
8. Roger Penrose. *Shadows of the Mind: A Search for the Missing Science of Consciousness*. Oxford University Press, 1994.
9. Solomon Feferman. *Penrose's Gödelian Argument*. Psyche, vol. 2, 1995.
10. W. S. McCulloch y W. H. Pitts. *A Logical Calculus of Ideas Imminent in Nervous Activity*. Bulletin of Mathematical Biophysics, 5, 1943.
11. Alan Turing. *Intelligent Machinery*. En *Collected Works of A. M. Turing: Mechanical Intelligence*. D. C. Ince (editor). Elsevier Science Publishers, 1992.
12. Frank Rosenblatt. *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*. Psychological Review, 65, 1958.
13. A. B. Novikov. *On convergence proofs on perceptrons*. Proceedings of the Symposium of the Mathematical Theory of Automata, volume XII, Polytechnic Institute of Brooklyn, 1962.
14. Marvin Minsky y Seymour Papert. *Perceptrons*. MIT Press, 1969.
15. J.J. Hopfield. *Neural networks and physical systems with emergent collective computational abilities.* Proceedings of the National Academy of Sciences, 79, 1982.
16. David Rumelhart, John McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Volúmenes 1 y 2. MIT Press, 1986.
17. Bernhard Schölkopf y Alex Smola. *Learning with kernels support vector machines, regularization, optimization, and beyond.* MIT Press, 2002. Las máquinas de vectores soporte producen modelos sumamente eficaces, pero son muy costosas cuando se aplican a grandes volúmenes de datos.
18. Leonardo Torres Quevedo. *Ensayos sobre automática. Su definición. Extensión teórica de sus aplicaciones*. Revista de la Real Academia de Ciencias Exactas, Físicas y Naturales. Madrid, 1914. Hay una versión en inglés en el libro de Brian Randell, *The Origins of Digital Computers*, Springer Verlag, 1975.

19. M. Wilkes. *Artificial Intelligence as the Year 2000 Approaches*. Communications of the ACM, 35. 1992.
20. Investopedia. Biggest Companies in the World by Market Cap.
21. Y. LeCun, Y. Bengio, G. Hinton. *Deep learning*. Nature 521, 436–444, 2015.
22. Como ejemplo, hemos pasado de los 200 teraflops del ordenador Blue Gene de IBM en 2006 a los 1,1 exaflops de Frontier, construido por Hewlett Packard. O, dicho de otra forma, Frontier es unas 5.000 veces más rápido.
23. G. Marcus, E. Davis. *GPT-3, Bloviator: OpenAI's language generator has no idea what it's talking about*. MIT Technology Review, 22 de agosto de 2020.
24. M. Jordan. *Artificial Intelligence—The Revolution Hasn't Happened Yet*. Harvard Data Science Review, issue 1.1, Summer 2019.