

Universidad Nacional Autónoma de México  
Facultad de Ingeniería  
Programa de Tecnología en Cómputo  
Proyecto de Inteligencia Artificial en MatLab

Elaborado por:

Bárcenas Martínez Erick Iván  
Bernal Sánchez Diego Arturo  
Rivera Negrete Manuel Armando

Titular: Rodrigo Castillo

14 de Junio del 2018

# Parte I

## Introducción

# Parte II

## Desarrollo

# Capítulo 1

## ¿Qué es la inteligencia artificial?

La Inteligencia Artificial tiene que ver con el diseño de programas inteligentes, algunas definiciones se refiere a procesos mentales y al razonamiento, otras a la conducta y por último a la función deseable en función de la eficiencia humana. A lo largo de la historia se han seguido los cuatro enfoques mencionados. Como es de esperar, existe un enfrentamiento entre los enfoques centrados en los humanos y los centrados en torno a la racionalidad. De acuerdo con autores recientes la Inteligencia Artificial es:

“El arte de crear Máquinas con capacidad de realizar funciones que realizadas por personas requieren de inteligencia” (Kurzweil, 1990.)

“La interesante tarea de lograr que las computadoras piensen ...*máquinas con mente*. en su amplio sentido literal.” (Haugeland, 1985.)

“El estudio de las facultades mentales mediante el uso de modelos computacionales.” (Charniak y McDermott, 1985.)

“La rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente.” (Luger y Stubblefield, 1993.)

### 1.1. Enfoques de la inteligencia Artificial

#### Actuar como humano: Enfoque cognitivo

Para poder decir que un programa determinado utiliza algún tipo de razonamiento humano es necesario previamente saber como piensan los seres humanos. Hay dos formas de hacerlo: mediante introspección (intentando atrapar nuestros propios pensamientos conforme éstos van apareciendo) y mediante experimentos psicológicos. Una vez se cuente con una teoría lo suficientemente precisa sobre cómo trabaja la mente, se podrá expresar esa teoría en la forma de un programa de computador. Si los datos de entrada/salida del programa y los tiempos de reacción son similares a los de un humano, existe la evidencia de que algunos de los mecanismos del programa se pueden comparar con los que utilizan los seres humanos.

## **Pensar como humano: Enfoque por las “leyes del pensamiento”**

Estudiosos de la lógica desarrollaron, en el siglo XIX, una notación precisa para definir sentencias sobre todo tipo de elementos del mundo y especificar relaciones entre ellos (compárese esto con la notación aritmética común, que prácticamente sólo sirve para representar afirmaciones acerca de la igualdad y desigualdad entre números). Ya en 1965 existían programas que, en principio, resolvían cualquier problema resoluble descrito en notación lógica.

La llamada tradición logista dentro del campo de la inteligencia artificial trata de construir sistemas inteligentes a partir de estos programas. Este enfoque presenta dos obstáculos. No es fácil transformar conocimiento informal y expresarlo en los términos formales que requieren de notación lógica, particularmente cuando el conocimiento que se tiene es inferior al 100 por 100. En segundo lugar, hay una gran diferencia entre poder resolver un problema «en principio» y hacerlo en la práctica. Incluso problemas con apenas una docena de datos pueden agotar los recursos computacionales de cualquier computador a menos que cuente con alguna directiva sobre los pasos de razonamiento que hay que llevar a cabo primero.

Aunque los dos obstáculos anteriores están presentes en todo intento de construir sistemas de razonamiento computacional, surgieron por primera vez en la tradición lógica.

## **Actuar de manera racional: Enfoque por agentes racionales**

Actuar racionalmente implica actuar de manera tal que se logren los objetivos deseados, con base en ciertos supuestos. Un agente es algo capaz de percibir y actuar. En el caso del enfoque de la IA según “las leyes del pensamiento”, todo el énfasis se ponía en hacer inferencias correctas. La obtención de estas inferencias a veces *forma parte* de lo que se considera un agente racional, puesto que una manera de actuar racionalmente es el razonamiento lógico que nos asegure la obtención de un resultado determinado.

Todas las “habilidades cognoscitivas” que se necesitan en la prueba de Turing permiten emprender conocimiento y razonar con base en él, pues de esta manera se podrán tomar decisiones correctas en una amplia gama de situaciones.

La percepción visual no es sólo algo divertido, sino que es algo necesario para darse una mejor idea de lo que una acción determinada puede producir.

Estudiar la IA adoptando el enfoque del diseño de un agente racional ofrece dos ventajas. Primera, es más general que el enfoque de las “Leyes del pensamiento”, dado que el efectuar inferencias correctas es sólo uno de los mecanismos existentes para garantizar la racionalidad, pero no es un mecanismo necesario. La segunda es más afín a la forma en la que se ha producido el avance científico que los enfoques basados en la conducta o pensamiento humano, porque la norma de la racionalidad está claramente definida y es de aplicación general. Por el contrario, la conducta humana se adapta bien a un entorno específico, y en parte, es producto de un proceso evolutivo complejo, en gran medida desconocido, que aún está lejos de llevarnos a la perfección.

## 1.2. Fundamentos de la inteligencia artificial

### Filosofía

Aristóteles (384-322 a.C.) fue el primero en formular un conjunto preciso de leyes que gobernaban la parte racional de la inteligencia. Él desarrolló un sistema informal para razonar adecuadamente con silogismos, que en principio permitía extraer conclusiones mecánicamente, a partir de premisas iniciales. Mucho después, Ramón Lull (d. 1315) tuvo la idea de que el razonamiento útil se podría obtener por medios artificiales. Sus «ideas» aparecen representadas en la portada de este manuscrito. Thomas Hobbes (1588-1679) propuso que el razonamiento era como la computación numérica, de forma que «nosotros sumamos y restamos silenciosamente en nuestros pensamientos». La automatización de la computación en sí misma estaba en marcha; alrededor de 1500, Leonardo da Vinci (1452-1519) diseñó, aunque no construyó, una calculadora mecánica; construcciones recientes han mostrado que su diseño era funcional. La primera máquina calculadora conocida se construyó alrededor de 1623 por el científico alemán Wilhelm Schickard (1592-1635), aunque la Pascalina, construida en 1642 por Blaise Pascal (1623-1662), sea más famosa. Pascal escribió que «la máquina aritmética produce efectos que parecen más similares a los pensamientos que a las acciones animales». Gottfried Wilhelm Leibniz (1646-1716) construyó un dispositivo mecánico con el objetivo de llevar a cabo operaciones sobre conceptos en lugar de sobre números, pero su campo de acción era muy limitado. René Descartes (1596-1650) proporciona la primera discusión clara sobre la distinción entre la mente y la materia y los problemas que surgen. Descartes sostenía que existe una parte de la mente (o del alma o del espíritu) que está al margen de la naturaleza, exenta de la influencia de las leyes físicas. Los animales, por el contrario, no poseen esta cualidad dual; a ellos se le podría concebir como si se tratasen de máquinas. Una alternativa al dualismo es el materialismo, que considera que las operaciones del cerebro realizadas de acuerdo a las leyes de la física constituyen la mente. El libre albedrío es simplemente la forma en la que la percepción de las opciones disponibles aparecen en el proceso de selección.

### Matemáticas

Los filósofos delimitaron las ideas más importantes de la IA, pero para pasar de ahí a una ciencia formal es necesario contar con una formulación matemática en tres áreas fundamentales: lógica, computación y probabilidad. Se piensa que el primer algoritmo no trivial es el algoritmo Euclídeo para el cálculo del máximo común divisor.

Se dice que un problema es intratable si el tiempo necesario para la resolución de casos particulares de dicho problema crece exponencialmente con el tamaño de dichos casos. La diferencia entre crecimiento polinomial y exponencial de la complejidad se destacó por primera vez a mediados de los años 60.

Es importante porque un crecimiento exponencial implica la imposibilidad de resolver casos moderadamente grandes en un tiempo razonable. Por tanto, se debe optar por dividir el problema de la generación de una conducta inteligente en subproblemas que sean tratables en vez de manejar problemas intratables.

La teoría de la NP-completitud, propuesta por primera vez por Steven Cook (1971) y Richard Karp (1972) propone un método. Cook y Karp demostraron la existencia de grandes clases de problemas de razonamiento y búsqueda combinatoria canónica que son NP completos. Toda clase de problema a la que la clase de problemas NP completos se pueda reducir será seguramente intratable (aunque no se ha demostrado que los problemas NP completos son necesariamente intratables, la mayor parte de los teóricos así lo creen). Además de la lógica y el cálculo, la tercera gran contribución de las matemáticas a la IA es la teoría de la probabilidad. La probabilidad se convirtió pronto en parte imprescindible de las ciencias cuantitativas, ayudando en el tratamiento de mediciones con incertidumbre y de teorías incompletas.

## **Economía**

La teoría de la decisión, que combina la teoría de la probabilidad con la teoría de la utilidad, proporciona un marco completo y formal para la toma de decisiones (económicas o de otra índole) realizadas bajo incertidumbre, esto es, en casos en los que las descripciones probabilísticas capturan adecuadamente la forma en la que se toman las decisiones en el entorno; lo cual es adecuado para «grandes» economías en las que cada agente no necesita prestar atención a las acciones que lleven a cabo el resto de los agentes individualmente.

El trabajo en la economía y la investigación operativa ha contribuido en gran medida a la noción de agente racional que aquí se presenta, aunque durante muchos años la investigación en el campo de la IA se ha desarrollado por sendas separadas. Una razón fue la complejidad aparente que trae consigo el tomar decisiones racionales. Herbert Simon (1916-2001), uno de los primeros en investigar en el campo de la IA, ganó el premio Nobel en Economía en 1978 por su temprano trabajo, en el que mostró que los modelos basados en satisfacción (que toman decisiones que son «suficientemente buenas», en vez de realizar cálculos laboriosos para alcanzar decisiones óptimas) proporcionaban una descripción mejor del comportamiento humano real (Simon, 1947). En los años 90, hubo un resurgimiento del interés en las técnicas de decisión teórica para sistemas basados en agentes (Wellman, 1995).

## **Neurociencia**

La Neurociencia es el estudio del sistema neurológico, y en especial del cerebro. La forma exacta en la que en un cerebro se genera el pensamiento es uno de los grandes misterios de la ciencia. Se ha observado durante miles de años que el cerebro está de alguna manera involucrado en los procesos de pensamiento, ya que fuertes golpes en la cabeza pueden ocasionar minusvalía mental.

No fue hasta mediados del siglo XVIII cuando se aceptó mayoritariamente que el cerebro es la base de la conciencia. Hasta este momento, se pensaba que estaba localizado en el corazón, el bazo y la glándula pineal.

La conclusión verdaderamente increíble es que una colección de simples células puede llegar a generar razonamiento, acción, y conciencia o, dicho en otras palabras, los cerebros generan las inteligencias (Searle, 1992). La única teoría alternativa es el misticismo: que nos dice que existe alguna esfera mística en la que las mentes operan

fuera del control de la ciencia física.

Cerebros y computadores digitales realizan tareas bastante diferentes y tienen propiedades distintas. Hay 1.000 veces más neuronas en un cerebro humano medio que puertas lógicas en la UCP de un computador estándar. La ley de Moore predice que el número de puertas lógicas de la UCP se igualará con el de neuronas del cerebro alrededor del año 2020. Por supuesto, poco se puede inferir de esta predicción; más aún, la diferencia en la capacidad de almacenamiento es insignificante comparada con las diferencias en la velocidad de intercambio y en paralelismo. Los circuitos de los computadores pueden ejecutar una instrucción en un nanosegundo, mientras que las neuronas son millones de veces más lentas. Las neuronas y las sinapsis del cerebro están activas simultáneamente, mientras que los computadores actuales tienen una o como mucho varias UCP. Por tanto, incluso sabiendo que un computador es un millón de veces más rápido en cuanto a su velocidad de intercambio, el cerebro acaba siendo 100,000 veces más rápido en lo que hace.

## Psicología

La conceptualización del cerebro como un dispositivo de procesamiento de información, característica principal de la psicología cognitiva, se remonta por lo menos a las obras de William James (1842-1910). Helmholtz también pone énfasis en que la percepción entraña cierto tipo de inferencia lógica inconsciente. Este punto de vista cognitivo se vio eclipsado por el conductismo en Estados Unidos, pero en la Unidad de Psicología Aplicada de Cambridge, dirigida por Frederic Bartlett (1886-1969), los modelos cognitivos emergieron con fuerza. La obra *The Nature of Explanation*, de Kenneth Craik (1943), discípulo y sucesor de Bartlett, reestablece enérgicamente la legitimidad de términos «mentales» como creencias y objetivos, argumentando que son tan científicos como lo pueden ser la presión y la temperatura cuando se habla acerca de los gases, a pesar de que éstos estén formados por moléculas que no tienen ni presión ni temperatura. Craik establece tres elementos clave que hay que tener en cuenta para diseñar un agente basado en conocimiento: (1) el estímulo deberá ser traducido a una representación interna, (2) esta representación se debe manipular mediante procesos cognitivos para así generar nuevas representaciones internas, y (3) éstas, a su vez, se traducirán de nuevo en acciones. Dejó muy claro por qué consideraba que estos eran los requisitos idóneos para diseñar un agente: Si el organismo tiene en su cabeza «un modelo a pequeña escala» de la realidad externa y de todas sus posibles acciones, será capaz de probar diversas opciones, decidir cuál es la mejor, planificar su reacción ante posibles situaciones futuras antes de que éstas surjan, emplear lo aprendido de experiencias pasadas en situaciones presentes y futuras, y en todo momento, reaccionar ante los imprevistos que acontezcan de manera satisfactoria, segura y más competente (Craik, 1943).

Los psicólogos comparten en la actualidad el punto de vista común de que «la teoría cognitiva debe ser como un programa de computador» (Anderson, 1980), o dicho de otra forma, debe describir un mecanismo de procesamiento de información detallado, lo cual lleva consigo la implementación de algunas funciones cognitivas.



## Ingeniería en computación

Para que la inteligencia artificial pueda llegar a ser una realidad se necesitan dos cosas: inteligencia y un artefacto. El computador ha sido el artefacto elegido. El computador electrónico digital moderno se inventó de manera independiente y casi simultánea por científicos en tres países involucrados en la Segunda Guerra Mundial. El equipo de Alan Turing construyó, en 1940, el primer computador operacional de carácter electromecánico, llamado Heath Robinson, con un único propósito: descifrar mensajes alemanes.

Desde mediados del siglo pasado, cada generación de dispositivos hardware ha conllevado un aumento en la velocidad de proceso y en la capacidad de almacenamiento, así como una reducción de precios. La potencia de los computadores se dobla cada 18 meses aproximadamente y seguirá a este ritmo durante una o dos décadas más. Después, se necesitará ingeniería molecular y otras tecnologías novedosas.

La IA también tiene una deuda con la parte software de la informática que ha proporcionado los sistemas operativos, los lenguajes de programación, y las herramientas necesarias para escribir programas modernos (y artículos sobre ellos). Sin embargo, en este área la deuda se ha saldado: la investigación en IA ha generado numerosas ideas novedosas de las que se ha beneficiado la informática en general, como por ejemplo el tiempo compartido, los intérpretes imperativos, los computadores personales con interfaces gráficas y ratones, entornos de desarrollo rápido, listas enlazadas, administración automática de memoria, y conceptos claves de la programación simbólica, funcional, dinámica y orientada a objetos.

## Teoría de control y cibernética

Ktesibios de Alejandría (250 a.C.) construyó la primera máquina auto controlada: un reloj de agua con un regulador que mantenía el flujo de agua circulando por él, con un ritmo constante y predecible. Esta invención cambió la definición de lo que un artefacto podía hacer.

La teoría de control moderna, especialmente la rama conocida como control óptimo estocástico, tiene como objetivo el diseño de sistemas que maximizan una función objetivo en el tiempo. Lo cual se asemeja ligeramente a nuestra visión de lo que es la IA: diseño de sistemas que se comportan de forma óptima. ¿Por qué, entonces, IA y teoría de control son dos campos diferentes, especialmente teniendo en cuenta la cercana relación entre sus creadores? La respuesta está en el gran acoplamiento existente entre las técnicas matemáticas con las que estaban familiarizados los investigadores y entre los conjuntos de problemas que se abordaban desde cada uno de los puntos de vista. El cálculo y el álgebra matricial, herramientas de la teoría de control, se utilizaron en la definición de sistemas que se podían describir mediante conjuntos fijos de variables continuas; más aún, el análisis exacto es sólo posible en sistemas lineales. La IA se fundó en parte para escapar de las limitaciones matemáticas de la teoría de control en los años 50. Las herramientas de inferencia lógica y computación permitieron a los investigadores de IA afrontar problemas relacionados con el lenguaje, visión y planificación, que estaban completamente fuera del punto de mira de la teoría de control.

## Lingüística

La lingüística moderna y la IA «nacieron», al mismo tiempo y maduraron juntas, solapándose en un campo híbrido llamado lingüística computacional o procesamiento del lenguaje natural. El problema del entendimiento del lenguaje se mostró pronto mucho más complejo de lo que se había pensado en 1957. El entendimiento del lenguaje requiere la comprensión de la materia bajo estudio y de su contexto, y no solamente el entendimiento de la estructura de las sentencias. Lo cual puede parecer obvio, pero no lo fue para la mayoría de la comunidad investigadora hasta los años 60. Gran parte de los primeros trabajos de investigación en el área de la representación del conocimiento (el estudio de cómo representar el conocimiento de forma que el computador pueda razonar a partir de dicha representación) estaban vinculados al lenguaje y a la búsqueda de información en el campo del lenguaje, y su base eran las investigaciones realizadas durante décadas en el análisis filosófico del lenguaje.

### 1.3. Historia de la inteligencia artificial

#### Gestión de la inteligencia artificial (1943-1955)

Warren McCulloch y Walter Pitts (1943) han sido reconocidos como los autores del primer trabajo de IA. Partieron de tres fuentes: conocimientos sobre la fisiología básica y funcionamiento de las neuronas en el cerebro, el análisis formal de la lógica proposicional de Russell y Whitehead y la teoría de la computación de Turing. Propusieron un modelo constituido por neuronas artificiales, en el que cada una de ellas se caracterizaba por estar «activada» o «desactivada»; la «activación» se daba como respuesta a la estimulación producida por una cantidad suficiente de neuronas vecinas. El estado de una neurona se veía como «equivalente, de hecho, a una proposición con unos estímulos adecuados». Mostraron, por ejemplo, que cualquier función de cómputo podría calcularse mediante alguna red de neuronas interconectadas, y que todos los conectores lógicos (and, or, not, etc.) se podrían implementar utilizando estructuras de red sencillas.

Hay un número de trabajos iniciales que se pueden caracterizar como de IA, pero fue Alan Turing quien articuló primero una visión de la IA en su artículo *Computing Machinery and Intelligence*, en 1950. Ahí, introdujo la prueba de Turing, el aprendizaje automático, los algoritmos genéricos y el aprendizaje por refuerzo.

#### Nacimiento de la inteligencia artificial (1956)

Princeton acogió a otras de las figuras señeras de la IA, John McCarthy. Posteriormente a su graduación, McCarthy se trasladó al Dartmouth College, que se erigiría en el lugar del nacimiento oficial de este campo. McCarthy convenció a Minsky, Claude Shannon y Nathaniel Rochester para que le ayudaran a aumentar el interés de los investigadores americanos en la teoría de autómatas, las redes neuronales y el estudio de la inteligencia. Organizaron un taller con una duración de dos meses en Dartmouth en el verano de 1956. Hubo diez asistentes en total, entre los que se

incluían Trenchard More de Princeton, Arthur Samuel de IBM, y Ray Solomonoff y Oliver Selfridge del MIT.

Dos investigadores del Carnegie Tech 13, Allen Newell y Herbert Simon, acapararon la atención. Si bien los demás también tenían algunas ideas y, en algunos casos, programas para aplicaciones determinadas como el juego de damas, Newell y Simon contaban ya con un programa de razonamiento, el Teórico Lógico (TL), del que Simon afirmaba: «Hemos inventado un programa de computación capaz de pensar de manera no numérica, con lo que ha quedado resuelto el venerable problema de la dualidad mente-cuerpo»

Poco después del término del taller, el programa ya era capaz de demostrar teoremas.

## **Entusiasmo temprano, grandes expectativas (1952-1969)**

Los primeros años de la IA estuvieron llenos de éxitos (aunque con ciertas limitaciones). Teniendo en cuenta lo primitivo de los computadores y las herramientas de programación de aquella época, y el hecho de que sólo unos pocos años antes, a los computadores se les consideraba como artefactos que podían realizar trabajos aritméticos y nadamás, resultó sorprendente que un computador hiciese algo remotamente inteligente. La comunidad científica, en su mayoría, prefirió creer que «una máquina nunca podría hacer tareas»

Al temprano éxito de Newell y Simon siguió el del sistema de resolución general de problemas, o SRGP. A diferencia del Teórico Lógico, desde un principio este programa se diseñó para que imitara protocolos de resolución de problemas de los seres humanos. Dentro del limitado número de puzzles que podía manejar, resultó que la secuencia en la que el programa consideraba que los subobjetivos y las posibles acciones eran semejantes a la manera en que los seres humanos abordaban los mismos problemas. Es decir, el SRGP posiblemente fue el primer programa que incorporó el enfoque de «pensar como un ser humano»

## **Sistemas Expertos (1969-1979)**

Feigenbaum junto con otros investigadores de Stanford dieron comienzo al Proyecto de Programación Heurística, PPH, dedicado a determinar el grado con el que la nueva metodología de los sistemas expertos podía aplicarse a otras áreas de la actividad humana. El siguiente gran esfuerzo se realizó en el área del diagnóstico médico. Feigenbaum, Buchanan y el doctor Edward Shortliffe diseñaron el programa MYCIN, para el diagnóstico de infecciones sanguíneas. Con 450 reglas aproximadamente, MYCIN era capaz de hacer diagnósticos tan buenos como los de un experto y, desde luego, mejores que los de un médico recién graduado. Se distinguía de DENDRAL en dos aspectos principalmente. En primer lugar, a diferencia de las reglas de DENDRAL, no se contaba con un modelo teórico desde el cual se pudiesen deducir las reglas de MYCIN. Fue necesario obtenerlas a partir de extensas entrevistas con los expertos, quienes las habían obtenido de libros de texto, de otros expertos o de su experiencia directa en casos prácticos. En segundo lugar, las reglas deberían reflejar la incertidumbre inherente al conocimiento médico. MYCIN contaba con un elemento que facilitaba el cálculo de incertidumbre denominado factores de certeza que al

parecer (en aquella época) correspondía muy bien a la manera como los médicos ponderaban las evidencias al hacer un diagnóstico

La importancia del conocimiento del dominio se demostró también en el área de la comprensión del lenguaje natural. Aunque el sistema SHRDLU de Winograd para la comprensión del lenguaje natural había suscitado mucho entusiasmo, su dependencia del análisis sintáctico provocó algunos de los mismos problemas que habían aparecido en los trabajos realizados en la traducción automática. Era capaz de resolver los problemas de ambigüedad e identificar los pronombres utilizados, gracias a que se había diseñado especialmente para un área (el mundo de los bloques). Fueron varios los investigadores que, como Eugene Charniak, estudiante de Winograd en el MIT, opinaron que para una sólida comprensión del lenguaje era necesario contar con un conocimiento general sobre el mundo y un método general para usar ese conocimiento.

## **La IA se vuelve industria (1980-presente)**

El primer sistema experto comercial que tuvo éxito, R1, inició su actividad en Digital Equipment Corporation (McDermott, 1982). El programa se utilizaba en la elaboración de pedidos de nuevos sistemas informáticos. En 1986 representaba para la compañía un ahorro estimado de 40 millones de dólares al año. En 1988, el grupo de Inteligencia Artificial de DEC había distribuido ya 40 sistemas expertos, y había más en camino. Du Pont utilizaba ya 100 y estaban en etapa de desarrollo 500 más, lo que le generaba ahorro de diez millones de dólares anuales aproximadamente. Casi todas las compañías importantes de Estados Unidos contaban con su propio grupo de IA, en el que se utilizaban o investigaban sistemas expertos. En 1981 los japoneses anunciaron el proyecto «Quinta Generación», un plan de diez años para construir computadores inteligentes en los que pudiese ejecutarse Prolog. Como respuesta Estados Unidos constituyó la Microelectronics and Computer Technology Corporation (MCC), consorcio encargado de mantener la competitividad nacional en estas áreas. En ambos casos, la IA formaba parte de un gran proyecto que incluía el diseño de chips y la investigación de la relación hombre máquina. Sin embargo, los componentes de IA generados en el marco de MCC y del proyecto Quinta Generación nunca alcanzaron sus objetivos. En el Reino Unido, el informe Alvey restauró el patrocinio suspendido por el informe Lighthill

## **El regreso de las redes neuronales (1986-presente)**

Aunque la informática había abandonado de manera general el campo de las redes neuronales a finales de los años 70, el trabajo continuó en otros campos. Físicos como John Hopfield (1982) utilizaron técnicas de la mecánica estadística para analizar las propiedades de almacenamiento y optimización de las redes, tratando colecciones de nodos como colecciones de átomos. Psicólogos como David Rumelhart y Geoff Hinton continuaron con el estudio de modelos de memoria basados en redes neuronales.

El impulso más fuerte se produjo a mediados de la década de los 80, cuando por lo menos cuatro grupos distintos reinventaron el algoritmo de aprendizaje de retro-

alimentación, mencionado por vez primera en 1969 por Bryson y Ho. El algoritmo se aplicó a diversos problemas de aprendizaje en los campos de la informática y la psicología, y la gran difusión que conocieron los resultados obtenidos, publicados en la colección *Parallel Distributed Processing* (Rumelhart y McClelland, 1986), suscitó gran entusiasmo. Aquellos modelos de inteligencia artificial llamados conexionistas fueron vistos por algunos como competidores tanto de los modelos simbólicos propuestos por Newell y Simon como de la aproximación lógica de McCarthy entre otros (Smolensky, 1988).

## **La IA adopta el método científico (1987-presente)**

En los primeros años de la IA parecía perfectamente posible que las nuevas formas de la computación simbólica, por ejemplo, los marcos y las redes semánticas, hicieran que la mayor parte de la teoría clásica pasara a ser obsoleta. Esto llevó a la IA a una especie de aislamiento, que la separó del resto de las ciencias de la computación. En la actualidad se está abandonando este aislamiento. Existe la creencia de que el aprendizaje automático no se debe separar de la teoría de la información, que el razonamiento incierto no se debe separar de los modelos estocásticos, de que la búsqueda no se debe aislar de la optimización clásica y el control, y de que el razonamiento automático no se debe separar de los métodos formales y del análisis estático.

En términos metodológicos, se puede decir, con rotundidad, que la IA ya forma parte del ámbito de los métodos científicos. Para que se acepten, las hipótesis se deben someter a rigurosos experimentos empíricos, y los resultados deben analizarse estadísticamente para identificar su relevancia (Cohen, 1995). El uso de Internet y el compartir repositorios de datos de prueba y código, ha hecho posible que ahora se puedan contrastar experimentos.

## **Surgimiento de los agentes inteligentes (1995-presente)**

El «movimiento situado» intenta entender la forma de actuar de los agentes inmersos en entornos reales, que disponen de sensores de entradas continuas. Uno de los medios más importantes para los agentes inteligentes es Internet. Los sistemas de IA han llegado a ser tan comunes en aplicaciones desarrolladas para la Web que el sufijo «-bot» se ha introducido en el lenguaje común. Más aún, tecnologías de IA son la base de muchas herramientas para Internet, como por ejemplo motores de búsqueda, sistemas de recomendación, y los sistemas para la construcción de portales Web.

## **Grandes bancos de datos (Big Data) (2001-presente)**

La Inteligencia Artificial en toda su magnitud y más concretamente con el Machine Learning, permite que el software que manejan las empresas puedan aprender, los patrones y comportamientos detectados en los clientes, tomando decisiones por sí mismos, pero para ello requiere de la orientación humana. En la actualidad, Deep

Learning, una disciplina dentro del Machine Learning, está trabajando para conseguir que el aprendizaje del software sea totalmente autónomo, sin intervención humana y así conseguir el gran reto de simular cómo aprende el cerebro humano. La Analítica de datos y las técnicas de Inteligencia Artificial han existido desde hace muchos años, sin embargo, el auge actual se debe al Big Data, que está permitiendo gestionar volúmenes ingentes de información y su procesamiento de forma ágil; cuanto mayor sea el volumen de información, más acertados serán los patrones y comportamientos detectados, de ahí la importancia de disponer de cantidades ingentes de datos y capacidad para procesarla rápidamente e incluso en tiempo real. La sinergia entre estas cuatro tecnologías, Analítica de datos, Machine Learning (o Deep Learning) y Big Data permiten a las empresas innovar en todas sus estructuras y ofrecer al cliente un servicio totalmente personalizado y a medida. Los datos son el petróleo del siglo XXI y estas tecnologías los explotan cuidadosamente para ofrecer servicios a medida y una nueva perspectiva, que ya demanda el cliente .

## 1.4. Estado actual de la Inteligencia Artificial

### ¿Qué puede hacer la IA hoy en día?

**Jugar una partida decente de Tenis** Los procesos de aprendizaje de robots son la base de proyectos como el robot Macgyver o de los brazos robóticos del MIT o un jugador de tenis de mesa.

Cuando el robot recibe la bola y debe responder, realiza una combinación de los 25 patrones mediante un sistema de ponderación, es decir, a cada uno de estos 25 movimientos base se les asigna un peso y el resultado final es el golpe que debe efectuar. ¿Y de dónde provienen los pesos? Aquí es donde entra en juego, de nuevo, el proceso de aprendizaje y entrenamiento, un proceso en el que se somete a prueba el sistema (se le hace jugar) y se le realimenta con los éxitos y fracasos en sus golpes, recalibrando los pesos y, mediante la práctica, obteniendo una mejor tasa de resultados.

#### **Conducir un coche**

El nuevo Mercedes-Benz Clase A 2018 tiene un asistente de voz que promete aprender del usuario gracias a la "inteligencia artificial", que tiene un panel de instrumentos cuyos gráficos ha diseñado NVIDIA y que se puede abrir con el smartphone. Es más, cualquiera con tu permiso puede abrir el coche con un smartphone.

#### **Comprar comestibles para una semana en la Web**

La inteligencia artificial ya es parte de los procesos de compras, entender tempranamente sus virtudes y limitantes, permitirá a las organizaciones obtener el mayor provecho de ella y adaptar las competencias de su personal a la nueva forma de trabajar.

#### **Jugar una partida de damas chinas**

Se dice que el juego de Damas chinas es un juego de mesa resuelto, es decir, que ya no hay forma de ganarle a una Inteligencia Artificial en este juego, lo mas a lo que podemos llegar es a un empate.

#### **descubrir y demostrar nuevos teoremas matematicos**

## ¿Qué no puede hacer la IA hoy en día?

### **Comprar comestibles para una semana en el mercado**

no existen sensores para determinar la calidad y tipo de producto se estimula que en 25 años se pueda realizar.

### **Escribir intencionalmente una historia divertida**

Esta tarea es casi imposible pues una maquina no tiene la capacidad de crear nuevas ideas ni tiene una proyeccion del tiempo.

### **Ofrecer asesoria legal competente en un area determinada**

Un computador no tiene la capacidad de razonar

### **Traducir ingles hablado al sueco hablado en tiempo real**

El grado de dificultad es bastante alto pues previamente se deberia tener un conocimiento general del tema a tratar y ademas diferencie la pronunciacion y el significado de las palabras.

### **Realizar una operacion de cirugia completa**

Hoy en dia son solamente una ayuda pues se debe manejar una perfeccion en un area tan sensible como la vida humana se considera realizable en unos 40 años

# Capítulo 2

## Agentes inteligentes

### 2.1. Agentes y entorno

Un agente es cualquier cosa capaz de percibir su medioambiente con la ayuda de sensores y actuar en ese medio utilizando actuadores. Un agente robot recibe pulsaciones del teclado, archivos de información y paquetes vía red a modo de entradas sensoriales y actúa sobre el medio con mensajes en el monitor, escribiendo ficheros y enviando paquetes por la red. Se trabajará con la hipótesis general de que cada agente puede percibir sus propias acciones. La percepción indica las entradas que puede tener un agente en cualquier instante.

### 2.2. Buen comportamiento: El concepto de racionalidad

Un agente racional es aquel que hace lo correcto, se puede decir que lo correcto es aquello que permite al agente obtener un resultado mejor. Por tanto, se necesita determinar una forma de medir el éxito. Las medidas de rendimiento incluyen los criterios que determinan el éxito en el comportamiento del agente. Cuando se sitúa un agente en un medio, éste genera una secuencia de acciones de acuerdo con las percepciones que recibe. Esta secuencia de acciones hace que su hábitat pase por una secuencia de estados. Si la secuencia es la deseada, entonces el agente habrá actuado correctamente.

La racionalidad en un momento determinado depende de cuatro factores:

- La medida de rendimiento que define el criterio de éxito.
- El conocimiento del medio en el que habita acumulado por el agente.
- Las acciones que el agente puede llevar a cabo.
- La secuencia de percepciones del agente hasta este momento.

En cada posible secuencia de percepciones, un agente racional deberá emprender aquella acción que supuestamente maximice su medida de rendimiento, basándose en las evidencias aportadas por la secuencia de percepciones y en el conocimiento que el agente mantiene almacenado.



## 2.3. Omnisciencia, aprendizaje y autonomía

Los agentes con éxito dividen las tareas de calcular la función del agente en tres períodos diferentes: cuando se está diseñando el agente, y están los diseñadores encargados de realizar algunos de estos cálculos; cuando está pensando en la siguiente operación, el agente realiza más cálculos; y cuando está aprendiendo de la experiencia, el agente lleva a cabo más cálculos para decidir cómo modificar su forma de comportarse.

### Racionalidad vs omnisciencia

Un agente omnisciente conoce el resultado de su acción y actúa de acuerdo con él. El agente racional no sólo recopila información, sino que aprende lo máximo posible de lo que está percibiendo. La configuración inicial del agente puede reflejar un conocimiento preliminar del entorno, pero a medida que el agente adquiere experiencia éste puede modificarse y aumentar.

### Racionalidad vs perfección

La racionalidad maximiza el rendimiento esperado, mientras la perfección maximiza el resultado real. Es imposible diseñar un agente que siempre lleve a cabo, de forma sucesiva, las mejores acciones después de un acontecimiento.

### Dependencia de conocimiento previo: Falta de autonomía

Se dice que un agente carece de autonomía cuando se apoya más en el conocimiento inicial que le proporciona su diseñador que en sus propias percepciones. Un agente racional debe ser autónomo, debe saber aprender a determinar cómo tiene que compensar el conocimiento incompleto o parcial inicial.

### Entorno de tareas

Un entorno de tareas es esencialmente los «problemas» para los que los agentes racionales son las «soluciones». Para ello se comienza mostrando cómo especificar un entorno de tareas, ilustrando el proceso con varios ejemplos. Posteriormente se mostrará que el entorno de tareas ofrece diferentes posibilidades, de forma que cada una de las posibilidades influyen directamente en el diseño del programa del agente.

## 2.4. Especificación del entorno de tareas

Para definir el entorno de tareas se usa el acrónimo **REAS** (**R**endimiento, **E**ntorno, **A**ctuadores, **S**ensores). En el diseño de un agente, el primer paso debe ser siempre especificar el entorno de trabajo de la forma más completa posible.

## **Rendimiento**

### **¿Cuál es el objetivo a cumplir del agente**

Se plantea de acuerdo al entorno cuál es su rendimiento. Ejemplo: En el caso de un agente taxista debe ser rápido, seguro y que llegue a su destino.

## **Entorno**

### **¿Cuál es el entorno en el que se encuentra el agente**

Siguiendo con el ejemplo del taxi, se plantea cuál es el entorno en el que se desea conducir y cuál es el entorno en el que se encuentra el taxi.

## **Actuadores**

### **¿Qué necesita el agente para ejecutar su función?**

Los actuadores disponibles en un taxi automático serán más o menos los mismos que los que tiene a su alcance un conductor humano: el control del motor a través del acelerador y control sobre la dirección y los frenos. Además, necesitará tener una pantalla de visualización o un sintetizador de voz para responder a los pasajeros, y quizás algún mecanismo para comunicarse, educadamente o de otra forma, con otros vehículos.

## **Sensores**

### **¿Dónde se encuentra el agente y que percepciones tiene?**

Sus sensores básicos deben, por tanto, incluir una o más cámaras de televisión dirigidas, un velocímetro y un tacómetro. Para controlar el vehículo adecuadamente, especialmente en las curvas, debe tener un acelerador; debe conocer el estado mecánico del vehículo, de forma que necesitará sensores que controlen el motor y el sistema eléctrico. Debe tener instrumentos que no están disponibles para un conductor medio: un sistema de posicionamiento global vía satélite (GPS) para proporcionarle información exacta sobre su posición con respecto a un mapa electrónico, y sensores infrarrojos o sonares para detectar las distancias con respecto a otros coches y obstáculos. Finalmente, necesitará un teclado o micrófono para que el pasajero le indique su destino.

## **2.5. Propiedades del entorno de tareas**

Se puede identificar un pequeño número de dimensiones en las que categorizar los entornos. Estas dimensiones determinan, hasta cierto punto, el diseño más adecuado para el agente y la utilización de cada una de las familias principales de técnicas en la implementación del agente.

### **Visibilidad completa**

Si los sensores del agente le proporcionan acceso al estado completo del medio en cada momento, entonces se dice que el entorno de trabajo es totalmente observa-

ble. Un entorno de trabajo es, efectivamente, totalmente observable si los sensores detectan todos los aspectos que son relevantes en la toma de decisiones; la relevancia, en cada momento, depende de las medidas de rendimiento. Entornos totalmente observables son convenientes ya que el agente no necesita mantener ningún estado interno para saber qué sucede en el mundo.

## Visibilidad parcial

Un entorno puede ser parcialmente observable debido al ruido y a la existencia de sensores poco exactos o porque los sensores no reciben información de parte del sistema, por ejemplo, un agente aspiradora con sólo un sensor de suciedad local no puede saber si hay suciedad en la otra cuadrícula, y un taxi automatizado no puede saber qué están pensando otros conductores.

## Único agente vs Multi-agente

La distinción entre el entorno de un agente individual y el de un sistema multiagente puede parecer suficientemente simple. Por ejemplo, un agente resolviendo un crucigrama por sí mismo está claramente en un entorno de agente individual, mientras que un agente que juega al ajedrez está en un entorno con dos agentes. Sin embargo hay algunas diferencias sutiles. Primero, se ha descrito que una entidad puede percibirse como un agente, pero no se ha explicado qué entidades se deben considerar agentes. ¿Tiene el agente A (por ejemplo el agente taxista) que tratar un objeto B (otro vehículo) como un agente, o puede tratarse meramente como un objeto con un comportamiento estocástico, como las olas de la playa o las hojas que mueve el viento? La distinción clave está en identificar si el comportamiento de B está mejor descrito por la maximización de una medida de rendimiento cuyo valor depende del comportamiento de A. Por ejemplo, en el ajedrez, la entidad oponente B intenta maximizar su medida de rendimiento, la cual, según las reglas, minimiza la medida de rendimiento del agente A. Por tanto, el ajedrez es un entorno multiagente competitivo. Por otro lado, en el medio definido por el taxista circulando, el evitar colisiones maximiza la medida de rendimiento de todos los agentes, así pues es un entorno multiagente parcialmente cooperativo. Es también parcialmente competitivo ya que, por ejemplo, sólo un coche puede ocupar una plaza de aparcamiento. Los problemas en el diseño de agentes que aparecen en los entornos multiagente son a menudo bastante diferentes de los que aparecen en entornos con un único agente; por ejemplo, la comunicación a menudo emerge como un comportamiento racional en entornos multiagente; en algunos entornos competitivos parcialmente observables el comportamiento estocástico es racional ya que evita las dificultades de la predicción.

## Determinístico vs estocástico

Si el siguiente estado del medio está totalmente determinado por el estado actual y la acción ejecutada por el agente, entonces se dice que el entorno es determinista; de otra forma es estocástico. En principio, un agente no se tiene que preocupar de la incertidumbre en un medio totalmente observable y determinista. Sin embargo,

si el medio es parcialmente observable entonces puede parecer estocástico. Esto es particularmente cierto si se trata de un medio complejo, haciendo difícil el mantener constancia de todos los aspectos observados. Así, a menudo es mejor pensar en entornos deterministas o estocásticos desde el punto de vista del agente.

## **Episódico vs secuencial**

En un entorno de trabajo episódico, la experiencia del agente se divide en episodios atómicos. Cada episodio consiste en la percepción del agente y la realización de una única acción posterior. Es muy importante tener en cuenta que el siguiente episodio no depende de las acciones que se realizaron en episodios previos. En los medios episódicos la elección de la acción en cada episodio depende sólo del episodio en sí mismo. Muchas tareas de clasificación son episódicas. Por ejemplo, un agente que tenga que seleccionar partes defectuosas en una cadena de montaje basa sus decisiones en la parte que está evaluando en cada momento, sin tener en cuenta decisiones previas; más aún, a la decisión presente no le afecta el que la próxima fase sea defectuosa. En entornos secuenciales, por otro lado, la decisión presente puede afectar a decisiones futuras. El ajedrez y el taxista son secuenciales: en ambos casos, las acciones que se realizan a corto plazo pueden tener consecuencias a largo plazo. Los medios episódicos son más simples que los secuenciales porque la gente no necesita pensar con tiempo.

## **Estático vs dinámico**

Si el entorno puede cambiar cuando el agente está deliberando, entonces se dice que el entorno es dinámico para el agente; de otra forma se dice que es estático. Los medios estáticos son fáciles de tratar ya que el agente no necesita estar pendiente del mundo mientras está tomando una decisión sobre una acción, ni necesita preocuparse sobre el paso del tiempo. Los medios dinámicos, por el contrario, están preguntando continuamente al agente qué quiere hacer; si no se ha decidido aún, entonces se entiende que ha tomado la decisión de no hacer nada. Si el entorno no cambia con el paso del tiempo, pero el rendimiento del agente cambia, entonces se dice que el medio es semidinámico. El taxista es claramente dinámico: tanto los otros coches como el taxi se están moviendo mientras el algoritmo que guía la conducción indica qué es lo próximo a hacer. El ajedrez, cuando se juega con un reloj, es semideterminista. Los crucigramas son estáticos.

## **Discreto vs continuo**

La distinción entre discreto y continuo se puede aplicar al estado del medio, a la forma en la que se maneja el tiempo y a las percepciones y acciones del agente. Por ejemplo, un medio con estados discretos como el del juego del ajedrez tiene un número finito de estados distintos. El ajedrez tiene un conjunto discreto de percepciones y acciones. El taxista conduciendo define un estado continuo y un problema de tiempo continuo: la velocidad y la ubicación del taxi y de los otros vehículos pasan por un rango de valores continuos de forma suave a lo largo del tiempo. La conducción del

taxista es también continua (ángulo de dirección, etc.). Las imágenes captadas por cámaras digitales son discretas, en sentido estricto, pero se tratan típicamente como representaciones continuas de localizaciones e intensidades variables.

## Conocido vs desconocido

### 2.6. Estructura de agentes

El trabajo de la IA es diseñar el programa del agente que implemente la función del agente que proyecta las percepciones en las acciones. Se asume que este programa se ejecutará en algún tipo de computador con sensores físicos y actuadores, lo cual se conoce como arquitectura: *Agente + arquitectura + programa*

#### Agentes reflexivos simples

El tipo de agente más sencillo es el agente reactivo simple. Estos agentes seleccionan las acciones sobre la base de las percepciones actuales, ignorando el resto de las percepciones históricas. Por ejemplo, el agente aspiradora es un agente reactivo simple porque toma sus decisiones sólo con base en la localización actual y si ésta está sucia

#### Agentes reflexivos basados en modelos

La forma más efectiva que tienen los agentes de manejar la visibilidad parcial es almacenar información de las partes del mundo que no pueden ver. O lo que es lo mismo, el agente debe mantener algún tipo de estado interno que dependa de la historia percibida y que de ese modo refleje por lo menos alguno de los aspectos no observables del estado actual.

La actualización de la información de estado interno según pasa el tiempo requiere codificar dos tipos de conocimiento en el programa del agente. Primero, se necesita alguna información acerca de cómo evoluciona el mundo independientemente del agente. Segundo, se necesita más información sobre cómo afectan al mundo las acciones del agente, este conocimiento acerca de «cómo funciona el mundo», tanto si está implementado con un circuito booleano simple o con teorías científicas completas, se denomina modelo del mundo. Un agente que utilice este modelo es un agente basado en modelos.

#### Agentes basados en metas

El conocimiento sobre el estado actual del mundo no es siempre suficiente para decidir qué hacer, además de la descripción del estado actual, el agente necesita algún tipo de información sobre su meta que describa las situaciones que son deseables. El programa del agente se puede combinar con información sobre los resultados de las acciones posibles (la misma información que se utilizó para actualizar el estado interno en el caso del agente reflexivo) para elegir las acciones que permitan alcanzar el objetivo. En algunas ocasiones, la selección de acciones basadas en objetivos es

directa, cuando alcanzar los objetivos es el resultado inmediato de una acción individual. En otras ocasiones, puede ser más complicado, cuando el agente tiene que considerar secuencias complejas para encontrar el camino que le permita alcanzar el objetivo. Aunque el agente basado en objetivos pueda parecer menos eficiente, es más flexible ya que el conocimiento que soporta su decisión está representado explícitamente y puede modificarse

## **Agentes basados en utilidades**

Una función de utilidad proyecta un estado (o una secuencia de estados) en un número real, que representa un nivel de felicidad. La definición completa de una función de utilidad permite tomar decisiones racionales en dos tipos de casos en los que las metas son inadecuadas. Primero, cuando haya objetivos conflictivos, y sólo se puedan alcanzar algunos de ellos, la función de utilidad determina el equilibrio adecuado. Segundo, cuando haya varios objetivos por los que se pueda guiar el agente, y ninguno de ellos se pueda alcanzar con certeza, la utilidad proporciona un mecanismo para ponderar la probabilidad de éxito en función de la importancia de los objetivos.

## **Agentes que aprenden**

Un agente que aprende se puede dividir en cuatro componentes conceptuales. La distinción más importante entre el elemento de aprendizaje y el elemento de actuación es que el primero está responsabilizado de hacer mejoras y el segundo se responsabiliza de la selección de acciones externas. El elemento de actuación: recibe estímulos y determina las acciones a realizar. El elemento de aprendizaje se realimenta con las críticas sobre la actuación del agente y determina cómo se debe modificar el elemento de actuación para proporcionar mejores resultados en el futuro. El diseño del elemento de aprendizaje depende mucho del diseño del elemento de actuación. Cuando se intenta diseñar un agente que tenga capacidad de aprender, la primera cuestión a solucionar no es ¿cómo se puede enseñar a aprender?, sino ¿qué tipo de elemento de actuación necesita el agente para llevar a cabo su objetivo, cuando haya aprendido cómo hacerlo? Dado un diseño para un agente, se pueden construir los mecanismos de aprendizaje necesarios para mejorar cada una de las partes del agente. La crítica indica al elemento de aprendizaje qué tal lo está haciendo el agente con respecto a un nivel de actuación fijo.

La crítica es necesaria porque las percepciones por sí mismas no prevén una indicación del éxito del agente.

El último componente del agente con capacidad de aprendizaje es el generador de problemas. Es responsable de sugerir acciones que lo guiarán hacia experiencias nuevas e informativas. Lo interesante es que si el elemento de actuación sigue su camino, puede continuar llevando a cabo las acciones que sean mejores, dado su conocimiento. Pero si el agente está dispuesto a explorar un poco, y llevar a cabo algunas acciones que no sean totalmente óptimas a corto plazo, puede descubrir acciones mejores a largo plazo. El trabajo del generador de problemas es sugerir estas acciones explora-

torias. Esto es lo que los científicos hacen cuando llevan a cabo experimentos. Galileo no pensaba que tirar piedras desde lo alto de una torre en Pisa tenía un valor por sí mismo. Él no trataba de romper piedras ni de cambiar la forma de pensar de transeúntes desafortunados que paseaban por el lugar. Su intención era adaptar su propia mente, para identificar una teoría que definiese mejor el movimiento de los objetos. En resumen, los agentes tienen una gran variedad de componentes, y estos componentes se pueden representar de muchas formas en los programas de agentes, por lo que, parece haber una gran variedad de métodos de aprendizaje. Existe, sin embargo, una visión unificada sobre un tema fundamental. El aprendizaje en el campo de los agentes inteligentes puede definirse como el proceso de modificación de cada componente del agente, lo cual permite a cada componente comportarse más en consonancia con la información que se recibe, lo que por tanto permite mejorar el nivel medio de actuación del agente.

## Capítulo 3

# Resolución de problemas mediante búsqueda

### 3.1. Agentes que resuelven problemas

Los agentes que resuelven problemas deciden qué hacer para encontrar secuencias de acciones que conduzcan a los estados deseables. Comenzamos definiendo con precisión los elementos que constituyen el «problema» y su «solución», y daremos diferentes ejemplos para ilustrar estas definiciones. Entonces, describimos diferentes algoritmos de propósito general que podamos utilizar para resolver estos problemas y así comparar las ventajas de cada algoritmo. Los algoritmos son no informados, en el sentido que no dan información sobre el problema salvo su definición.

Los objetivos ayudan a organizar su comportamiento limitando las metas que intenta alcanzar el agente. El primer paso para solucionar un problema es la formulación del objetivo, basado en la situación actual y la medida de rendimiento del agente. Consideraremos un objetivo como un conjunto de estados del mundo (exactamente aquellos estados que satisfacen el objetivo). La tarea del agente es encontrar qué secuencia de acciones permite obtener un estado objetivo. Para esto, necesitamos decidir qué acciones y estados considerar.

Dado un objetivo, la formulación del problema es el proceso de decidir qué acciones y estados tenemos que considerar. Un algoritmo de búsqueda toma como entrada un problema y devuelve una solución de la forma secuencia de acciones. Una vez que encontramos una solución, se procede a ejecutar las acciones que ésta recomienda. Esta es la llamada fase de ejecución.

### Problemas bien definidos y soluciones

Un problema puede definirse, formalmente, por cuatro componentes:

- El **estado inicial** en el que comienza el agente.
- Una descripción de las posibles **acciones** disponibles por el agente. La formulación más común utiliza una **función sucesor**. Dado un estado particular  $x$ ,  $SU-CESOR-FN(x)$  devuelve un conjunto de pares ordenados « *acción, sucesor* », donde cada acción es una de las acciones legales en el estado  $x$  y cada sucesor es un estado que puede alcanzarse desde  $x$ , aplicando la acción.



- El **test objetivo**, el cual determina si un estado es un estado objetivo. Algunas veces existe un conjunto explícito de posibles estados objetivo, y el test simplemente comprueba si el estado es uno de ellos. Algunas veces el objetivo se especifica como una propiedad abstracta más que como un conjunto de estados enumerados explícitamente. Por ejemplo, en el ajedrez, el objetivo es alcanzar un estado llamado «jaque mate», donde el rey del oponente es atacado y no tiene escapatoria.
- Una función **costo del camino** que asigna un costo numérico a cada camino. El agente resolvente de problemas elige una función costo que refleje nuestra medida de rendimiento. El costo individual de una acción  $a$  que va desde un estado  $x$  al estado  $y$  se denota por  $c(x, a, y)$ .

## Formulación de problemas

Al proceso de eliminar detalles de una representación se le llama abstracción. Además de abstraer la descripción del estado, debemos abstraer sus acciones. La abstracción es útil si al realizar cada una de las acciones en la solución es más fácil que en el problema original. La elección de una buena abstracción implica quitar tantos detalles como sean posibles mientras que se conserve la validez y se asegure que las acciones abstractas son fáciles de realizar. Si no fuera por la capacidad de construir abstracciones útiles, los agentes inteligentes quedarían totalmente absorbidos por el mundo real.

### 3.2. Búsqueda de soluciones

Una de las técnicas de búsqueda es utilizar un árbol de búsqueda explícito generado por el estado inicial y la función sucesor, definiendo así el espacio de estados. En general, podemos tener un grafo de búsqueda más que un árbol, cuando el mismo estado puede alcanzarse desde varios caminos.

La esencia de la búsqueda es llevar a cabo una opción y dejar de lado las demás para más tarde, en caso de que la primera opción no conduzca a una solución. Es importante recordar la distinción entre nodos y estados. Un **nodo** es una estructura de datos usada para representar el árbol de búsqueda. Un **estado** corresponde a una configuración del mundo. Así, los nodos están en caminos particulares, según lo definido por los punteros del nodo padre, mientras que los estados no lo están. También necesitamos representar la colección de nodos que se han generado pero todavía no se han expandido, a esta colección se le llama **frontera**.

Cada elemento de la frontera es un nodo hoja, es decir, un nodo sin sucesores en el árbol. la frontera de cada árbol consiste en los nodos dibujados con líneas discontinuas. La representación más simple de la frontera sería como un conjunto de nodos. La estrategia de búsqueda será una función que seleccione de este conjunto el siguiente nodo a expandir. Aunque esto sea conceptualmente sencillo, podría ser computacionalmente costoso, porque la función estrategia quizá tenga que mirar cada elemento del conjunto para escoger el mejor. Hay muchas formas de representar los nodos, pero vamos a suponer que un nodo es una estructura de datos con cinco componentes:

- ESTADO : el estado, del espacio de estados, que corresponde con el nodo;

- NODO PADRE : el nodo en el árbol de búsqueda que ha generado este nodo;
- ACCIÓN : la acción que se aplicará al padre para generar el nodo;
- COSTO DEL CAMINO : el costo, tradicionalmente denotado por  $g(n)$ , de un camino desde el estado inicial al nodo, indicado por los punteros a los padres.
- PROFUNDIDAD : el número de pasos a lo largo del camino desde el estado inicial.

### 3.3. Parámetros de rendimiento de solución de problemas

La salida del algoritmo de resolución de problemas es fallo o una solución. (Algunos algoritmos podrían caer en un bucle infinito y nunca devolver una salida.) El rendimiento del algoritmo se evalúa respondiendo a cuatro preguntas.

#### Complejidad

¿Está garantizado que el algoritmo encuentre una solución cuando esta exista?

#### ¿Qué tan óptimo es?

¿Encuentra la estrategia la solución óptima?

Para valorar la eficacia de un algoritmo de búsqueda, podemos considerar el costo de la búsqueda (que depende típicamente de la complejidad en tiempo pero puede incluir también un término para el uso de la memoria) o podemos utilizar el coste total, que combina el costo de la búsqueda y el costo del camino solución encontrado.

#### Complejidad espacial

¿Cuánta memoria se necesita para el funcionamiento de la búsqueda?

La complejidad en tiempo y espacio siempre se considera con respecto a alguna medida de la dificultad del problema. En informática teórica, la medida es el tamaño del grafo del espacio de estados, porque el grafo se ve como una estructura de datos explícita que se introduce al programa de búsqueda.

#### Complejidad temporal

¿Cuánto tarda en encontrar una solución?

El tiempo a menudo se mide en términos de número de nodos generados durante la búsqueda, y el espacio en términos de máximo número de nodos que se almacena en memoria.

### 3.4. Estrategias de búsqueda ciega

El término búsqueda ciega significa que los tipos de búsqueda no tienen información adicional acerca de los estados más allá de la que proporciona la definición

del problema. Todo lo que estas búsquedas pueden hacer es generar los sucesores y distinguir entre un estado objetivo de uno que no lo es.

### **Búsqueda primero por amplitud (BFS)**

La búsqueda primero primero por amplitud es una estrategia sencilla en la que se expande primero el nodo raíz, a continuación se expanden todos los sucesores del nodo raíz, después sus sucesores, etc. En general, se expanden todos los nodos a una profundidad en el árbol de búsqueda antes de expandir cualquier nodo del próximo nivel. La búsqueda primero en anchura se puede implementar llamando a la BÚSQUEDA ÁRBOLES con una frontera vacía que sea una cola primero en entrar primero en salir (FIFO), asegurando que los nodos primeros visitados serán los primeros expandidos.

### **Búsqueda primero por profundidad (DFS)**

La búsqueda primero por profundidad siempre expande el nodo más profundo en la frontera actual del árbol de búsqueda. La búsqueda procede inmediatamente al nivel más profundo del árbol de búsqueda, donde los nodos no tienen ningún sucesor. Cuando esos nodos se expanden, son quitados de la frontera, así entonces la búsqueda «retrocede» al siguiente nodo más superficial que todavía tenga sucesores inexplorados. Esta estrategia puede implementarse por la BÚSQUEDA-ÁRBOLES con una cola último en entrar primero en salir (LIFO), también conocida como una pila. Como una alternativa a la implementación de la BÚSQUEDA-ÁRBOLES, es común aplicar la búsqueda primero en profundidad con una función recursiva que se llama en cada uno de sus hijos. La búsqueda primero en profundidad tiene unos requisitos muy modestos de memoria. Necesita almacenar sólo un camino desde la raíz a un nodo hoja, junto con los nodos hermanos restantes no expandidos para cada nodo del camino. Una vez que un nodo se ha expandido, se puede quitar de la memoria tan pronto como todos su descendientes han sido explorados.

### **Búsqueda de costo uniforme (UCS)**

La búsqueda de costo uniforme no se preocupa por el número de pasos que tiene un camino, pero sí sobre su coste total. Por lo tanto, éste se meterá en un bucle infinito si expande un nodo que tiene una acción de coste cero que conduzca de nuevo al mismo estado. Podemos garantizar completitud si el costo de cada paso es mayor o igual a alguna constante positiva pequeña. Esta condición es también suficiente para asegurar optimización. Significa que el costo de un camino siempre aumenta cuando vamos por él. De esta propiedad, es fácil ver que el algoritmo expande nodos que incrementan el coste del camino. Por lo tanto, el primer nodo objetivo seleccionado para la expansión es la solución óptima.

### 3.5. Estrategias de búsqueda heurística

Una estrategia de búsqueda informada o búsqueda heurística que utiliza el conocimiento específico del problema más allá de la definición del problema en sí mismo, puede encontrar soluciones de una manera más eficiente que una estrategia no informada o búsqueda ciega.

#### Búsqueda voraz (Greedy best-first search)

La búsqueda voraz trata de expandir el nodo más cercano al objetivo, alegando que probablemente conduzca rápidamente a una solución. Así, evalúa los nodos utilizando solamente la función heurística:  $f(n) = h(n)$ . La búsqueda voraz se parece a la búsqueda primero por profundidad en el modo que prefiere seguir un camino hacia el objetivo, pero volverá atrás cuando llegue a un callejón sin salida. Sufre los mismos defectos que la búsqueda primero por profundidad, no es óptima, y es incompleta (porque puede ir hacia abajo en un camino infinito y nunca volver para intentar otras posibilidades). La complejidad en tiempo y espacio, del caso peor, es

$$O(b^m) \quad (3.1)$$

, donde  $m$  es la profundidad máxima del espacio de búsqueda. Con una buena función, sin embargo, pueden reducir la complejidad considerablemente. La cantidad de la reducción depende del problema particular y de la calidad de la heurística.

#### A Estrella (A\*)

Evalúa los nodos combinando  $g(n)$ , el coste para alcanzar el nodo, y  $h(n)$ , el coste de ir al nodo objetivo:

$$f(n) = g(n) + h(n) \quad (3.2)$$

Ya que la  $g(n)$  nos da el coste del camino desde el nodo inicio al nodo  $n$ , y la  $h(n)$  el coste estimado del camino más barato desde  $n$  al objetivo, tenemos:  $f(n)$  = coste más barato estimado de la solución a través de  $n$ . Así, si tratamos de encontrar la solución más barata, es razonable intentar primero el nodo con el valor más bajo de  $g(n) + h(n)$ . Resulta que esta estrategia es más que razonable: con tal de que la función heurística  $h(n)$  satisfaga ciertas condiciones, la búsqueda A\* es tanto completa como óptima.

La optimalidad de A\* es sencilla de analizar si se usa con la BÚSQUEDA-ÁRBOLES. En este caso, A\* es óptima si  $h(n)$  es una heurística admisible, es decir, con tal de que la  $h(n)$  nunca sobrestime el coste de alcanzar el objetivo. Las heurísticas admisibles son por naturaleza optimistas, porque piensan que el coste de resolver el problema es menor que el que es en realidad. Ya que  $g(n)$  es el coste exacto para alcanzar  $n$ , tenemos como consecuencia inmediata que la  $f(n)$  nunca sobrestima el coste verdadero de una solución a través de  $n$ .

## 3.6. Funciones heurísticas

Una función heurística es un algoritmo que abandona uno o ambos objetivos; por ejemplo, normalmente encuentran buenas soluciones, aunque no hay pruebas de que la solución no pueda ser arbitrariamente errónea en algunos casos; o se ejecuta razonablemente rápido, aunque no existe tampoco prueba de que siempre será así. Las heurísticas generalmente son usadas cuando no existe una solución óptima bajo las restricciones dadas (tiempo, espacio, etc.), o cuando no existe del todo.

A menudo, pueden encontrarse instancias concretas del problema donde la heurística producirá resultados muy malos o se ejecutará muy lentamente. Aun así, estas instancias concretas pueden ser ignoradas porque no deberían ocurrir nunca en la práctica por ser de origen teórico. Por tanto, el uso de heurísticas es muy común en el mundo real.

### Heurísticas admisibles

Un algoritmo relacionado con búsqueda de caminos, se dice que una heurística es admisible si nunca sobreestima el costo de alcanzar el objetivo, o sea, que en el punto actual la estimación del costo de alcanzar el objetivo nunca es mayor que el menor costo posible. Una heurística admisible es usada para estimar el costo de alcanzar el estado objetivo en un algoritmo de búsqueda informada. Una heurística será admisible para cierto problema de búsqueda cuando el costo estimado sea siempre menor o igual que el costo mínimo de alcanzar el estado objetivo. El algoritmo de búsqueda usa la heurística para encontrar una estimación del camino óptimo hasta el objetivo desde el nodo actual.

### Aprendizaje de heurísticas por experiencia

Una función heurística  $h(n)$ , como se supone, estima el costo de una solución que comienza desde el estado en el nodo  $n$ . ¿Cómo podría un agente construir tal función? Una solución es aprender de la experiencia. A partir de ejemplos, se puede utilizar un algoritmo de aprendizaje inductivo para construir una función  $h(n)$  que pueda (con suerte) predecir los costos solución para otros estados que surjan durante la búsqueda. Los métodos de aprendizaje inductivos trabajan mejor cuando se les suministran características de un estado que sean relevantes para su evaluación, más que sólo la descripción del estado.