

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Programa de Tecnología en Cómputo (Proteco)

Curso de Matlab

Proyecto de Introducción a la Inteligencia Artificial con Matlab

Generación 36 de pre-becarios del PROTECO

Alumnos:

Barredo Olguín Patricio

Carballido García Alicia

Martínez Flores Emanuel

Martínez Ostóa Néstor

Índice

1 Introducción

2 Desarrollo

2.1 Qué es la inteligencia artificial?

2.1.1 Enfoques de la inteligencia artificial

- ¿ Actuar como humano: Enfoque cognitivo
- ¿ Pensar como humano: Enfoque por las leyes del pensamiento
- ¿ Actuar de manera racional: Enfoque por agentes racionales

2.2 Fundaciones de la inteligencia artificial

2.2.1 Filosofia

2.2.2 Matemticas

2.2.3 Economa

2.2.4 Neurociencia

2.2.5 Psicologa

2.2.6 Ingeniera en computacin

2.2.7 Teora de control y ciberntica

2.2.8 Lingstica

2.3 Historia de la inteligencia artificial

2.3.1 Gestin de la inteligencia artificial (1943-1955)

2.3.2 Nacimiento de la inteligencia artificial (1956)

2.3.3 Entusiasmo temprano, grandes expectativas (1952-1969)

2.3.4 Sistemas Expertos (1969-1979)

2.3.5 La IA se vuelve industria (1980-presente)

2.3.6 El regreso de las redes neuronales (1986-presente)

2.3.7 La IA adopta el mtodo cientfico (1987-presente)

2.3.8 Surgimiento de los agentes inteligentes (1995-presente)

2.3.9 Grandes bancos de datos (Big Data) (2001-presente)

2.4 Estado actual de la Inteligencia Artificial

2.4.1 Qu puede hacer la IA hoy en da?

2.4.2 Qu no puede hacer la IA hoy en da?

2.5 Agentes Inteligentes

2.5.1 Agentes Inteligentes

Agentes y entorno

Buen comportamiento: El concepto de racionalidad

Omnisciencia, aprendizaje y autonomía

Racionalidad vs omnisciencia

Racionalidad vs perfección

Dependencia de conocimiento previo. Falta de autonomía

Entorno de tareas

Especificación del entorno de tareas

Rendimiento

Entorno

Actuadores

Sensores

Propiedades del entorno de tareas

Visibilidad completa

Visibilidad parcial

Único agente vs Multi-agente

Determinístico vs estocástico

Estático vs Dinámico

Discreto vs continuo

Conocido vs desconocido

Estructura de agentes

Agentes reflexivos simples

Agentes reflexivos basados en modelos

Agentes basados en metas

Agentes basados en utilidades

Agentes que aprenden

2.6 Resolución de problemas mediante búsqueda

2.6.1 Agentes que resuelven problemas

Problemas bien definidos y soluciones

Formulación de problemas

2.6.2 Travelling Salesman Problem (TSP)

2.6.3 Búsqueda de soluciones

2.6.4 Parámetros de rendimiento de solución de problemas

Complejidad

Qué tan óptimo es

Complejidad espacial

Complejidad temporal

2.6.5 Estrategias de búsqueda ciega

Búsqueda primero por amplitud (BFS)

Búsqueda primero por profundidad (DFS)

Búsqueda de costo uniforme (UCS)

2.6.6 Estrategias de bsqueda heurstica

Bsqueda voraz (Greedy best-first search)

A Estrella (A *)

2.6.7 Funciones heursticas

Heurísticas adminisibles

Aprendizaje de heursticas por experiencia

3 Conclusiones

Inteligencia Artificial con Matlab

Introducción

El ser humano ha tratado de entender cómo piensa, por ello desarrolló el campo de la Inteligencia Artificial. Esta ciencia que es reciente, y ha canalizado sus esfuerzos en no sólo comprender el pensamiento humano, sino que también en construir entidades inteligentes.

Comenzó a desarrollarse tiempo después de la Segunda Guerra Mundial, y su nombre se definió en 1956. Desde entonces ha sido un campo genuinamente universal ya que abarca áreas de propósito general como el aprendizaje y la percepción, a otras más específicas como el ajedrez, la demostración de teoremas matemáticos, la escritura de poesía y el diagnóstico de enfermedades.

La IA sintetiza y automatiza tareas intelectuales y es, por lo tanto, potencialmente relevante para cualquier ámbito de la actividad intelectual humana. En este sentido, es un campo genuinamente universal.

El proyecto que se presenta a continuación consiste en la creación de una aplicación en MATLAB que permita encontrar la ruta más óptima de una estación a otra estación y de la red de transporte público metro de la CDMX utilizando como fundamento teórico la teoría de agentes y los problemas de búsqueda que están basados en los conceptos de Inteligencia Artificial.

Desarrollo

¿Qué es la inteligencia Artificial?

La inteligencia artificial tiene varias definiciones; algunas se refieren a procesos mentales y al razonamiento, y otras aluden a la conducta. Unas definen el éxito en términos de la fidelidad en la forma de actuar de los humanos, mientras que otras toman como referencia un concepto ideal de inteligencia, que llamaremos racionalidad.

A través del tiempo varios autores han intentado definir lo que es la inteligencia artificial. Para definir la IA han surgido diferentes enfoques.

Existe un enfrentamiento entre ellos ya que unos se centran en los humanos y otros entorno a la racionalidad. El enfoque centrado en el comportamiento humano debe ser una ciencia empírica, que incluya hipótesis y confirmaciones mediante experimentos. El enfoque racional implica una combinación de matemáticas e ingeniería. A continuación se presentarán algunos ejemplos de definiciones con respecto a su enfoque:

Sistemas que piensan como humanos

”El nuevo y excitante esfuerzo de hacer que los computadores piensen... mquinas con mentes, en el ms amplio sentido literal”.(Haugeland, 1985).

”La automatizacin de] actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolucin de problemas, aprendizaje...” (Bellman, 1978).

Sistemas que actan como humanos

”El arte de desarrollar mquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia”.(Kurzweil, 1990).

”El estudio de cmo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor”. (Rich y Knight, 1991).

Sistemas que piensan racionalmente

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

”El estudio de los clculos que hacen posible percibir, razonar y actuar”. (Winston, 1992).

Sistemas que actan racionalmente

”La Inteligencia Computacional es el estudio del diseo de agentes inteligentes”. (Poole, 1998).

”IA... est relacionada con conductas inteligentes en artefactos”. (Nilsson, 1998).

A continuacin, daremos detalles de los enfoques vistos anteriormente.

Enfoques de la Inteligencia Artificial

Comportamiento humano: el enfoque de la Prueba de Turing

En 1950, Alan Turing ide la prueba de Turing que fue diseada para proporcionar una definicin operacional y satisfactoria de inteligencia. Esta prueba se basaba en la incapacidad de diferenciar entre entidades inteligentes y seres humanos. El computador supera la prueba si un evaluador humano no es capaz de distinguir si las respuestas, a una serie de preguntas planteadas, son de una persona o no. Para que el computador pudiera superar estas pruebas deba poseer las siguientes capacidades: Procesamiento de lenguaje: comunicarse satisfactoriamente en ingls.

Capacidad de almacenar lo que conoce o siente. Razonamiento automático para utilizar la información almacenada para responder a preguntas y extraer nuevas conclusiones. Aprendizaje automático para adaptarse a nuevas circunstancias y para detectar patrones.

Su prueba dejó de lado la interacción física en ese entonces dado que para medir la inteligencia es innecesario simular físicamente a una persona. Existe otra prueba llamada Prueba Global de Turing en donde el computador sí requiera de capacidades físicas para percibir, manipular y mover objetos.

Actuar como humano: Enfoque cognitivo

Antes de poder decir que un programa puede pensar como un ser humano, es necesario conocer cómo piensa éste; para ello es necesario introducirse en las mentes humanas. Se puede lograr a través de la introspección, ya sea a través de experimentos psicológicos o intentando atrapar nuestros propios pensamientos conforme éstos van apareciendo.

Conociendo datos sobre cómo trabaja la mente, se podrá expresar esa teoría en la forma de un programa de computador. A través de datos de entrada/salida del programa se analizan tiempos de reacción en busca de similitudes.

Pensar como humano: Enfoque por las "leyes del pensamiento"

A través de la historia, Aristóteles (filósofo griego) fue uno de los primeros en intentar definir la manera correcta de pensar, a manera de razonamiento irrefutable. Su estudio fue el principio de la lógica. Sus estructuras de argumentación mediante las que siempre se llega a conclusiones correctas si se parte de premisas correctas. Estas leyes de pensamiento supuestamente gobiernan la manera de operar de la mente.

En el siglo XIX los estudiosos de la lógica definieron sentencias para representar afirmaciones acerca de la igualdad y desigualdad entre números. Para 1965 ya existían programas que resolvían problemas utilizando esta notación. La llamada tradición lógica dentro del campo de la inteligencia artificial trata de construir sistemas inteligentes a partir de estos programas, pero presenta dos obstáculos. -Expresar el conocimiento informal a un lenguaje formal. -Agotamiento de los recursos computacionales.

Actuar de manera racional: Enfoque por agentes racionales

Primero vamos a definir qu es un agente. En trminos generales, es algo que razona (del latn agere, hacer), pero de los agentes informticos se espera que tengan otros atributos: como que sean capaces de percibir su entorno, que persistan durante un periodo de tiempo, se adapten a los cambios, y que sean capaces de alcanzar objetivos diferentes. Un agente racional es aquel que acta con la intencin de alcanzar el mejor resultado o el mejor resultado esperado.

La IA se enfoca en que segn las leyes del pensamiento debe poner todo el nfasis en hacer inferencias correctas. La obtencin de estas inferencias correctas puede, a veces, formar parte de lo que se considera un agente racional. Sin embargo, el efectuar una inferencia correcta no depende siempre de la racionalidad, ya que existen situaciones para las que no hay nada correcto que hacer y en las que hay que tomar una decisin, por ejemplo un acto reflejo.

Es necesario contar con la capacidad para representar el conocimiento y razonar basndonos en l, porque ello permitir alcanzar decisiones correctas en una gran variedad de situaciones. Es necesario ser capaz de generar sentencias comprensibles ya que esto permite a los agentes desenvolverse en una sociedad compleja. Profundizar en el conocimiento de cmo funciona el mundo facilita la concepcin de cmo manejarse en l. La percepcin visual es necesaria para poder tener una idea mejor de lo que una accin puede llegar a representar.

Si no se encuentra una solucin, el programa nunca debe parar de buscarla.

El estudiar la IA desde el enfoque del diseo de un agente racional ofrece ventajas. Una de ellas es que al utilizar el enfoque que proporcionan las leyes del pensamiento, al efectuar inferencias correctas se garantiza la racionalidad. Pero hacer siempre lo correcto no es posible en entornos complejos. La demanda computacional que esto implica es demasiado grande.

Fundaciones de la inteligencia artificial

A continuacin se presenta una breve historia de las disciplinas que han contribuido con ideas, puntos de vista y tcnicas al desarrollo de la IA.

Filosofa (desde el ao 428 a.C. hasta el presente)

Aristteles (384-322 a.C.) Formul leyes sobre la racionalidad de la inteligencia. Adems desarroll un sistema de silogismos para razones adecuadamente a partir de premisas. Ramn Lull (d. 1315) Aport la idea de que a travs de medios artificiales era posible generar razonamiento. Thomas Hobbes (1588-1679) Tena la idea del razonamiento como una computacin numrica, en donde nosotros sumamos

y restamos silenciosamente en nuestros pensamientos. Leonardo da Vinci (1452-1519) Dise una calculadora mecánica con un diseño funcional. Wilhelm Schickard (1592-1635) Construy la primer calculadora conocida alrededor de 1623 Blaise Pascal (1623-1662) En construy la calculadora conocida como pascalina. Adems escribi que la mquina aritmética produce efectos que parecen ms similares a los pensamientos que a las acciones animales. Gottfried Wilhelm Leibniz (1646-1716) Construy un dispositivo mecánico con el objetivo de llevar a cabo operaciones sobre conceptos en lugar de sobre nmeros, pero su campo de accin era muy limitado. Ren Descartes (1596-1650) Aporta la idea sobre la clara distincin entre la mente y la materia y los problemas que surgen. Uno de los problemas se presenta con el libre albedro. Ren fue un defensor del dualismo. Sostenía que existe una parte de la mente (o del alma o del espritu) que est al margen de la naturaleza, exenta de la influencia de las leyes fsicas. Los animales, por el contrario, no poseen esta cualidad dual.

El materialismo se hace presente y consideraba que las operaciones del cerebro realizadas de acuerdo a las leyes de la fsica constituyen la mente. El libre albedro es simplemente la forma en la que la percepcin de las opciones disponibles aparecen en el proceso de seleccin. *Novum Organum* 3, de Francis Bacon (1561-1626), fue el precursor del empirismo que se caracteriza por el aforismo de John Locke (1632-1704): Nada existe en la mente que no haya pasado antes por los sentidos. David Hume (1711-1776), propuso en *A Treatise of Human Nature* (Hume, 1739) lo que actualmente se conoce como principio de induccin. Posteriormente se desarroll la doctrina del positivismo lgico. La teora de la confirmacin de Carnap y Carl Hempel (1905-1997) intenta explicar cmo el conocimiento se obtiene a partir de la experiencia.

La inteligencia requiere tanto accin como razonamiento. Ms an, simplemente con comprender cmo se justifican determinadas acciones se puede llegar a saber cmo construir un agente cuyas acciones sean justificables (o racionales). Cmo es que el pensamiento viene acompaado en algunos casos de acciones y en otros no?, en algunos casos por movimiento y en otros no?. Nosotros no reflexionamos sobre los fines, sino sobre los medios.

Matemáticas (aproximadamente desde el ao 800 al presente)

La filosofa delimita los conceptos ms importantes del pensamiento que rigen la IA. Para darle formalidad es necesario formular matemticamente en tres reas que son: lgica, computacin y probabilidad.

Antigua Grecia

George Boole (1815-1864) defini la lgica proposicional o Booleana .En 1879, Gotlob Frege (1848-1925) extendi la lgica de Boole para incluir objetos y relaciones, y cre la lgica de primer orden que se utiliza hoy como el sistema ms bsico de representacin de conocimiento. Alfred Tarski (1902-1983) introdujo una teora de referencia que enseña cmo relacionar objetos de una lgica con objetos del mundo real. El paso siguiente consisti en definir los lmites de lo que se poda hacer con la lgica y la informtica.

Algoritmo

Eucldeo aport el primer algoritmo no trivial de la historia, que trataba sobre el clculo del mximo comn divisor. Los escritos de al-Khowarazmi, (matemtico persa del siglo IX) y los de Boole, presentaron algoritmos para llevar a cabo deducciones lgicas y hacia el final del siglo XIX.

David Hilbert (1862-1943) Present una lista de 23 problemas que acertadamente predijo ocuparan a los matemticos durante todo ese siglo, entre ellos el famoso Entscheidungsproblem, o problema de decisin).

Kurt Gdel (1906-1978) En 1930, demostr que existe un procedimiento eficiente para demostrar cualquier aseveracin verdadera en la lgica de primer orden. En 1931, demostr que, en efecto, existen lmites reales. Aport el teorema de incompletitud.

Alan Turing (1912-1954) Existen algunas funciones de los nmeros enteros que no se pueden representar mediante un algoritmo, es decir no se pueden calcular. Lo anterior llev a Alan a tratar de caracterizar exactamente aquellas funciones que s eran susceptibles de ser caracterizadas. Turing tambin demostr que existen algunas funciones que no se pueden calcular mediante la mquina de Turing.

Si bien ser no decidible ni computable son importantes para comprender el proceso del clculo, la nocin de intratabilidad tuvo repercusiones ms importantes. En trminos generales se dice que un problema es intratable si el tiempo necesario para la resolucin de casos particulares de dicho problema crece exponencialmente con el tamao de dichos casos.

(Cobham, 1964; Edmonds, 1965). La diferencia entre crecimiento polinomial y exponencial de la complejidad se destac por primera vez a mediados de los aos 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 generacin de una conducta inteligente en subproblemas que sean tratables en vez de manejar problemas intratables.

Cmo se puede reconocer un problema intratable?

La teora de la NP-completitud fue propuesta por Steven Cook (1971) y Richard Karp (1972). Demostraron la existencia de grandes clases de problemas NP completos que son seguramente intratables (aunque no se ha demostrado que los problemas NP completos son necesariamente intratables, la mayor parte de los tericos as lo creen).

Recientemente la IA ha ayudado a explicar por qu algunos ejemplos de problemas NP completos son difciles de resolver y otros son fciles. Adems de la lgica y el clculo, la tercera gran contribucin de las matemticas a la IA es la teora de la probabilidad.

Probabilidad

Gerolamo Cardano (1501-1576) fue el primero en proponer la idea de probabilidad, presentndola en trminos de los resultados de juegos de apuesta. La probabilidad se convirti pronto en parte imprescindible de las ciencias cuantitativas, ayudando en el tratamiento de mediciones con incertidumbre y de teoras incompletas.

Pierre Fermat (1601-1665), Blaise Pascal (1623-1662), James Bernoulli (1654-1705), Pierre Laplace (1749-1827), entre otros, hicieron avanzar esta teora e introdujeron nuevos mtodos estadsticos. Thomas Bayes (1702-1761) propuso una regla para la actualizacin de probabilidades subjetivas a la luz de nuevas evidencias. La regla de Bayes y el rea resultante llamado anlisis Bayesiano conforman la base de las propuestas ms modernas que abordan el razonamiento incierto en sistemas de IA.

Economa (desde el ao 1776 hasta el presente)

Adam Smith (1723-1790) Public *An Inquiry into the Nature and Causes of the Wealth of Nations*. A pesar de que los griegos haban hecho contribuciones al pensamiento econmico, l fue el primero en tratarlo como una ciencia, utilizando la idea de que las economas pueden concebirse como un conjunto de agentes individuales que intentan maximizar su propio estado de bienestar econmico.

Lon Walras (1834-1910) Formaliz el tratamiento matemtico del "beneficio deseado" o utilidad, y fue posteriormente mejorado por Frank Ramsey (1931) y despus por John von Neumann y Oskar Morgenstern.

Toma de decisiones

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 en general realizadas bajo incertidumbre. Así se captura 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 posee un papel importante junto con los demás.

Teoría de juegos

Los desarrollos de von Neumann y Morgenstern a partir de la teoría de juegos mostraban el hecho sorprendente de que, en algunos juegos, un agente racional deba actuar de forma aleatoria o, al menos, aleatoria en apariencia con respecto a sus contrincantes. La gran mayoría de los economistas no se preocuparon de la tercera cuestión; se enfocan en los resultados de las acciones de forma secuencial.

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. La complejidad radica en 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, proporcionando una descripción del comportamiento humano real).

Neurociencia (desde el año 1861 hasta el presente)

Neurociencia ha puesto sus esfuerzos en el estudio del sistema neurológico, y en el funcionamiento del cerebro que aún es uno de los grandes misterios de la ciencia. Aproximadamente en el 335 a.C. Aristóteles escribió, de entre todos los animales el hombre tiene el cerebro más grande en proporción a su tamaño. Aunque, no fue hasta mediados del siglo XVIII cuando se aceptó mayoritariamente que el cerebro es la base de la conciencia. Antes se había considerado que el pensamiento estaba localizado en el corazón, el bazo y la glándula pineal.

El estudio de Paul Broca (1824-1880) sobre la afasia (dificultad para hablar) en pacientes con el cerebro dañado, convenció a la sociedad médica de que cada área del cerebro cumple con funciones cognitivas específicas. En esta época ya se sabía que

el cerebro estaba formado por células nerviosas o neuronas.

En 1873 cuando Camillo Golgi (1843-1926) logró observar neuronas individuales en el cerebro. Santiago Ramón y Cajal (1852-1934) utilizó esta técnica en sus estudios pioneros sobre la estructura neuronal del cerebro.

En la actualidad se dispone de información sobre la relación existente entre las áreas del cerebro y las partes del cuerpo humano que controlan o de las que reciben impulso sensoriales. Tales relaciones pueden cambiar de forma radical incluso en pocas semanas pero más aún, no se tiene totalmente claro cómo algunas áreas se pueden encargar de ciertas funciones que eran responsabilidad de áreas dadas. No hay prácticamente ninguna teoría que explique cómo se almacenan recuerdos individuales.

En 1929 los estudios sobre la actividad cerebral dieron inicio con el descubrimiento del electroencefalograma (EEG) desarrollado por Hans Berger. Es reciente el descubrimiento de las imágenes de resonancia magnética funcional (IRMf). Proporciona a los neurólogos imágenes detalladas de la actividad cerebral, permitiéndoles obtener medidas que se corresponden con procesos cognitivos.

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. 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 (desde el año 1879 hasta el presente)

Hermann von Helmholtz (1821-1894) Alemán. Fue el precursor del método científico al estudio de la vista humana junto con su discípulo Wilhelm Wundt (1832-1920). Su obra *Handbook of Physiological Optics* se ha considerado

como el tratado actual ms importante sobre la fsica y la fisiologa de la vista humana. Wundt puso mucho nfasis en la realizacin de experimentos controlados cuidadosamente en la que sus operarios realizaban tareas de percepcin asociacin al tiempo que sometan a introspeccin sus procesos mentales.

Con el paso del tiempo, la psicologa logr convertirse en una ciencia, pero la naturaleza subjetiva de los datos hizo poco probable que un investigador pudiera contradecir sus propias teoras. El movimiento conductista, liderado por John Watson (1878-1958) aplic este punto de vista a los humanos, rechazando cualquier teora en la que intervinieran procesos mentales, argumentando que la introspeccin no aportaba una evidencia fiable.

Construcciones mentales como conocimientos, creencias, objetivos y pasos en un razonamiento quedaron descartadas por ser consideradas "psicologa popular" no cientfica. El conductismo hizo muchos descubrimientos experimentando con animales pero tuvo menos xito en la comprensin de los seres humanos. An as, su influencia en la psicologa fue notable.

William James en sus obras, conceptualiz al cerebro como un dispositivo de procesamiento de informacin. Helmholtz tambin pone nfasis en que la percepcin entraa cierto tipo de inferencia lgica inconsciente.

La obra *The Nature of Explanation*, de Kenneth Craik (1943), establece tres elementos clave que hay que tener en cuenta para disear un agente basado en conocimiento:

- (1) el estmulo deber ser traducido a una representacin interna.
- (2) esta representacin se debe manipular mediante procesos cognitivos para as generar nuevas representaciones interna
- (3) stas, a su vez, se traducirn de nuevo en acciones.

Si el organismo tiene en su cabeza un modelo a pequea escala de la realidad externa y de todas sus posibles acciones, ser capaz de probar diversas opciones, decidir cul es la mejor, planificar su reaccin ante posibles situaciones futuras, emplear lo aprendido de experiencias pasadas en situaciones presentes y futuras, y en todo momento, reaccionar ante los imprevistos que acontezcan de manera satisfactoria.

Los psiclogos comparten en la actualidad el punto de vista comn de que la teora cognitiva debe ser como un programa de computador (Anderson, 1980), o dicho de otra forma, debe describir un mecanismo de procesamiento de informacin detallado, lo cual lleva consigo la implementacin de algunas

funciones cognitivas.

Ingeniera en computacin(desde el ao 1940 hasta el presente)

La inteligencia artificial requiere de dos cosas: inteligencia y un artefacto. Se ha elegido el artefacto como computador. Este electrónico digital moderno se invent de manera independiente y casi simultánea por científicos en tres países involucrados en la Segunda Guerra Mundial. Por ejemplo Heath Robinson, creado por el equipo de Alan Turing en 1940; fue el primer computador operacional de carácter electromecánico, con el propósito de descifrar mensajes alemanes.

En 1943 el mismo grupo desarrolló el Colossus, una máquina potente de propósito general basada en válvulas de vacío. El primer computador operacional programable fue el Z-3, inventado por Konrad Zuse en Alemania, en 1941. Zuse también inventó los números de coma flotante y el primer lenguaje de programación de alto nivel, Plankalkül.

El primer computador electrónico el ABC, fue creado por John Atanasoff junto a su discípulo Clifford Berry entre 1940 y 1942 en la Universidad Estatal de Iowa. El ENIAC, desarrollado en el marco de un proyecto militar secreto se considera el precursor de los computadores modernos. 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.

Antes de los computadores ya había dispositivos de cálculo. La primera máquina programable fue un telar, desarrollado en 1805 por Joseph Marie Jacquard (1752-1834) que utilizaba tarjetas perforadas para almacenar información sobre los patrones de los bordados. Adems Charles Babbage (1792-1871) diseñó dos máquinas, que no llegó a construir. La Máquina de Diferencias y La Máquina Analítica. La última fue el primer artefacto dotado de los elementos necesarios para realizar una computación universal.

Ada Lovelace, fue seguramente la primera programadora. Ella escribió programas para la inacabada Máquina Analítica e incluso especuló acerca de la posibilidad de que la máquina jugara al ajedrez y compusiera música. La IA también ha generado numerosas ideas novedosas de las que se ha beneficiado la informática en general: 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

Teoría de control y cibernética (desde el año 1948 hasta el presente)

La primera máquina autocontrolada que fue un reloj de agua, fue construido por Ktesibios de Alejandria (250 a.C.). Contaba con un regulador que mantenía el a un ritmo constante y predecible. Otros ejemplos de sistemas de control auto regulables y retroalimentados son el motor de vapor, creado por James Watt (1736-1819), y el termostato, inventado por Cornelis Drebbel (1572-1633).

La teoría matemática de los sistemas con retroalimentación estable se desarrolló en el siglo XIX. El precursor de la teoría de control fue Norbert Wiener (1894-1964). Wiener y sus colegas Arturo Rosenbluth y Julian Bigelow veían el comportamiento determinista como algo emergente de un mecanismo regulador que intenta minimizar el "error" (la diferencia entre el estado presente y el estado objetivo). El libro de Wiener, *Cybernetics* (1948), fue un bestseller y reveló al público las posibilidades de las máquinas con inteligencia artificial.

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. 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 fundió en parte para escapar de las limitaciones matemáticas de la teoría de control en los años 50.

Lingüística (desde el año 1957 hasta el presente)

En 1957, se publicó una obra que presentaba una visión extensa y detallada desde el enfoque conductista al aprendizaje del lenguaje. Estaba escrita por los expertos más destacados de este campo. A pesar de esto Chomsky mostró cómo la teoría conductista no abordaba el tema de la creatividad en el lenguaje: no explicaba cómo es posible que un niño sea capaz de entender y construir oraciones que nunca antes

ha escuchado. La lingstica moderna y la IA nacieron, al mismo tiempo y maduraron juntas, solapndose en un campo hbrido llamado lingstica computacional o procesamiento del lenguaje natural.

El problema del entendimiento del lenguaje se hizo presente. ste requiere la comprensin de la materia bajo estudio y de su contexto, y no solamente el entendimiento de la estructura de las sentencias. Gran parte de los primeros trabajos de investigacin en el rea de la representacin del conocimiento estaban vinculados al lenguaje y a la bsqueda de informacin en el campo del lenguaje, y su base eran las investigaciones realizadas durante dcadas en el anlisis filosfico del lenguaje.

Historia de la inteligencia artificial:

Gestin de la inteligencia artificial (1943-1955)

Los autores del primer trabajo de IA reconocidos son Warren McCulloch y Walter Pitts (1943). Se basaron en conocimientos sobre la fisiologa bsica y funcionamiento de las neuronas en el cerebro, el anlisis formal de la lgica proposicional de Russell y Whitehead y la teora de la computacin de Turing.

Propusieron un modelo constituido por neuronas artificiales caracterizadas por estar "activadas" o "desactivadas"; sto se refera a la existencia de estmulo. Con ello mostraron, por ejemplo, que cualquier funcin de cmputo podra calcularse mediante alguna red de neuronas interconectadas, con conectores lgicos y se podran implementar utilizando estructuras de red sencillas.

Donald Hebb (1949) propuso y demostr una sencilla regla de actualizacin para modificar las intensidades de las conexiones entre neuronas. En 1951, los estudiantes Princeton, Marvin Minsky y Dean Edmonds, construyeron el primer computador a partir de una red neu. Utilizaba 3.000 vlvulas de vaco y un mecanismo de piloto automtico obtenido de los desechos de un avin bombardero B-24 para simular una red con 40 neuronas. Minsky posteriormente prob teoremas influyentes que mostraron las limitaciones de la investigacin con redes neuronales.

Hay un nmero de trabajos iniciales que se pueden caracterizar como de IA, pero fue Alan Turing quien articul primero una visin de la IA en su artculo Computing

Machinery and Intelligence, en 1950. Ah, introdujo la prueba de Turing, el aprendizaje automático, los algoritmos genéricos y el aprendizaje por refuerzo.

Nacimiento de la inteligencia artificial (1956)

Todo inició en el Dartmouth College en donde 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, Allen Newell y Herbert Simon, acapararon la atención ya que contaban con un programa de razonamiento, el Terico Lgico (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.

Se dice que Russell se manifestó complacido cuando Simon le mostró que la demostración de un teorema que el programa había generado era más corta que la que aparecía en Principia. Los editores de la revista Journal of Symbolic Logic resultaron menos impresionados y rechazaron un artículo cuyos autores eran Newell, Simon y el Terico Lgico (TL).

El taller de Dartmouth no produjo ningún avance notable, pero puso en contacto a las figuras importantes de este campo. Durante los siguientes 20 años, el campo estuvo dominado por estos personajes, así como por sus estudiantes y colegas del MIT, CMU, Stanford e IBM. Quiz lo último que surgió del taller fue el consenso en adoptar el nuevo nombre propuesto por McCarthy para este campo: Inteligencia Artificial.

La IA comenzó a convertirse en un campo separado. ¿Por qué? La primera respuesta es que la IA desde el primer momento abarcó la idea de duplicar facultades humanas como la creatividad, la auto-mejora y el uso del lenguaje. Ninguno de los otros campos tenían en cuenta esos temas. La segunda respuesta está relacionada con la metodología. La IA es el único de estos campos que es claramente una rama

de la informática (aunque la investigación operativa comparte el nfasis en la simulación por computador), además la IA es el nico campo que persigue la construcción de máquinas que funcionen automáticamente en medios complejos y cambiantes.

Entusiasmo inicial, grandes esperanzas (1952-1969)

El hecho de que slo unos pocos años antes, a los computadores se les consideraba como artefactos que podían realizar trabajos aritméticos y nada más, result sorprendente que un computador hiciese algo remotamente inteligente. Se prefería creer que una máquina nunca podría hacer tareas, pero los investigadores de IA responderían demostrando la realización de una tarea tras otra.

Al temprano éxito de Newell y Simon siguió el del sistema de resolución general de problemas, o SRGP. Este programa se diseñó para que imitara protocolos de resolución de problemas de los seres humanos. Este fue posiblemente fue el primer programa que incorporó el enfoque de "pensar como un ser humano". Su éxito y de los programas que le siguieron, como los modelos de cognición, llevaron a Newell y Simon (1976) a formular la famosa hipótesis del sistema de símbolos físicos, que afirma que "un sistema de símbolos físicos tiene los medios suficientes y necesarios para generar una acción inteligente". Lo que ellos querían decir es que cualquier sistema (humano o máquina) que exhibiese inteligencia debería operar a base de símbolos.

En IBM, Nathaniel Rochester y sus colegas desarrollaron algunos de los primeros programas de IA. Herbert Gelernter (1959) construyó el demostrador de teoremas de geometría (DTG), el cual era capaz de probar teoremas que muchos estudiantes de matemáticas podían encontrar muy complejos de resolver. A comienzos 1952, Arthur Samuel escribió una serie de programas para el juego de las damas. De paso, echó por tierra la idea de que los computadores sólo pueden hacer lo que se les dice: su programa pronto aprendió a jugar mejor que su creador. El programa se presentó en la televisión en febrero de 1956 y causó una gran impresión.

En 1958 en el Laboratorio de IA del MIT Memo Número 1, McCarthy definió el lenguaje de alto nivel Lisp, que se convertiría en el lenguaje de programación dominante en la IA; es el segundo lenguaje de programación más antiguo que se utiliza en la actualidad, ya que apareció un año después de FORTRAN.

Esta fue una gran herramienta pero el acceso a los escasos y costosos recursos

de cómputo era un problema serio. Para solucionarlo, él, junto a otros miembros del MIT, inventaron el tiempo compartido. También, en 1958, McCarthy publicó un artículo titulado *Programs with Common Sense*, en el que describía el Generador de Consejos, un programa hipotético que podría considerarse como el primer sistema de IA completo. Diseñó su programa para buscar la solución a problemas utilizando el conocimiento. Pero, a diferencia de los otros, manejaba el conocimiento general del mundo.

El Generador de Consejos incorporaba a los principios centrales de la representación del conocimiento y el razonamiento: es útil contar con una representación formal y explícita del mundo y de la forma en que la acción de un agente afecta al mundo, así como, ser capaces de manipular estas representaciones con procesos deductivos. Es sorprendente constatar cómo mucho de lo propuesto en el artículo escrito en 1958 permanece vigente incluso en la actualidad.

1958 fue el año en el que Marvin Minsky se trasladó al MIT. Sin embargo, su colaboración inicial no duró demasiado. McCarthy se centró en la representación y el razonamiento con lógica formal, mientras que Minsky estaba más interesado en lograr que los programas funcionaran y eventualmente desarrollaran un punto de vista antilógico.

En 1963 McCarthy creó el Laboratorio de IA en Stanford. Gracias al descubrimiento de J. A. Robinson del método de resolución su Generador de consejos tuvo un gran avance. El trabajo realizado en Stanford hacía énfasis en los métodos de propósito general para el razonamiento lógico. Algunas aplicaciones de la lógica incluían los sistemas de planificación y respuesta a preguntas de Cordell Green (1969b), así como el proyecto de robótica de Shakey en el nuevo Instituto de Investigación de Stanford (Stanford Research Institute, SRI).

Minsky supervisó el trabajo de una serie de estudiantes que eligieron un número de problemas limitados cuya solución pareciera requerir inteligencia. Estos dominios limitados se conocen como micromundos. El programa SAINT de James Slagle (1963) fue capaz de resolver problemas de integración de cálculo en forma cerrada, habituales en los primeros cursos de licenciatura. El programa ANALOGY de Tom Evans (1968) resolvía problemas de analogía geométrica que se aplicaban en las pruebas de medicina de inteligencia. El programa STUDENT de Daniel Bobrow (1967) podía resolver problemas de álgebra.

El mundo de los bloques de micromundos fue el punto de partida para el proyecto de visión de David Huffman (1971), la visión y el trabajo de propagación con restricciones de David Waltz (1975), la teoría del aprendizaje de Patrick Winston (1970),

del programa para la comprensión de lenguaje natural de Terry Winograd (1972) y del planificador de Scott Fahlman (1974). El trabajo realizado por McCulloch y Pitts con redes neuronales hizo florecer esta rama. El trabajo de Winograd y Cowan (1963) mostró cómo un gran número de elementos podría representar un concepto individual de forma colectiva, lo cual llevaba consigo un aumento proporcional en robustez y paralelismo. Los métodos de aprendizaje de Hebb se reforzaron con las aportaciones de Bernie Widrow, quien llamó adalines a sus redes, y por Frank Rosenblatt (1962) con sus perceptrones.

Rosenblatt demostró el famoso teorema del perceptrón, con lo que mostró que su algoritmo de aprendizaje podría ajustar las intensidades de las conexiones de un perceptrón para que se adaptaran a los datos de entrada, siempre y cuando existiera una correspondencia.

Sistemas Expertos (1969-1979)

La IA estaba centrada en el desarrollo de mecanismos de búsqueda de propósito general para encontrar las soluciones completas. A estos procedimientos se les ha denominado métodos débiles, debido a que no tratan problemas más amplios o más complejos. Podría afirmarse que para resolver un problema en la práctica, es necesario saber de antemano la correspondiente respuesta.

El programa DENDRAL constituye uno de los primeros ejemplos de este enfoque. Los creadores de DENDRAL consultaron con químicos analíticos y se dieron cuenta de que estos trabajaban buscando patrones conocidos de picos en el espectro que sugieran estructuras comunes en la molécula. La trascendencia de DENDRAL se debió a ser el primer sistema de conocimiento intenso que tuvo éxito: ya que estaba formada por grandes cantidades de reglas de propósito particular. En sistemas diseñados posteriormente se incorporaron también los elementos fundamentales de la propuesta de McCarthy para el Generador de Consejos, la nítida separación del conocimiento de la parte correspondiente al razonamiento.

Pronto se dio 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 de infecciones sanguíneas. Con 450 reglas aproximadamente, MYCIN era capaz de hacer diagnósticos tan buenos como los de un experto.

Se distinguía de DENDRAL en dos aspectos principalmente debido a que no se contaba con un modelo teórico desde el cual se pudiesen deducir las reglas de MYCIN. En segundo lugar, las reglas debían reflejar la incertidumbre inherente

al conocimiento mdico.

MYCIN contaba con un elemento que facilitaba el clculo de incertidumbre denominado factores de certeza que al parecer (en aquella poca) corresponda muy bien a la manera como los mdicos ponderaban las evidencias al hacer un diagnstico.

Se demostr la importancia del dominio en el rea de la comprensin del lenguaje natural. El sistema SHRDLU era capaz de resolver los problemas de ambigüedad e identificar los pronombres utilizados, gracias a que se haba diseado especialmente para un rea (el mundo de los bloques). Fueron varios los investigadores que, como Eugene Charniak, estudiante de Winograd en el MIT, opinaron que para una slida comprensin del lenguaje era necesario contar con un conocimiento general sobre el mundo y un mtodo general para usar ese conocimiento.

Roger Schank reforz lo anterior al afirmar: No existe eso que llaman sintaxis, lo que irrit a muchos lingüistas, pero sirvi para iniciar un til debate. Schank y sus estudiantes disearon una serie de programas cuyo objetivo era la comprensin del lenguaje natural. El foco de atencin estaba menos en el lenguaje pero se ms en los problemas vinculados a la representacin y razonamiento del conocimiento necesario para la comprensin del lenguaje. Entre los problemas estaba el de la representacin de situaciones estereotipo, la descripcin de la organizacin de la memoria humana y la comprensin de planes y objetivos.

Creci la necesidad de resolver problemas del mundo real, lo que provoc el aumento en la demanda de esquemas de representacin del conocimiento que funcionaran. Se desarroll una considerable cantidad de lenguajes de representacin y razonamiento diferentes.

La IA se vuelve industria (1980-presente)

Se comenz a comercializar el primer sistema experto llamado R1 que inici su actividad en Digital Equipment Corporation (McDermott, 1982). Se utiliz en la elaboracin de pedidos de nuevos sistemas informticos. En 1986 representaba para la compaa un ahorro de millones de dlares. En 1988, el grupo de Inteligencia Artificial de DEC haba distribuido ya 40 sistemas expertos, y haba ms en camino. Du Pont utilizaba ya 100 y estaban en etapa de desarrollo 500 ms, lo que le generaba ahorro de diez millones de dlares anuales aproximadamente. Casi todas las compaas importantes de Estados Unidos contaban con su propio grupo de IA.

Los japoneses anunciaron el proyecto Quinta Generacin en 1981, un plan de diez aos para construir computadores inteligentes en los que pudiese ejecutarse Prolog.

Como respuesta Estados Unidos constituyó la Microelectronics and Computer Technology Corporation (MCC), para mantener la competitividad nacional. 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.

La industria de la IA creció rápidamente, pero poco tiempo después de este período llegó la poca llamada El Invierno de la IA, que afectó a muchas empresas que no fueron capaces de desarrollar los extravagantes productos prometidos.

El regreso de las redes neuronales (1986-presente)

A finales de los años 70, el trabajo sobre redes neuronales continuó en otros campos. John Hopfield (1982) utilizó técnicas de la mecánica estadística para analizar las propiedades de almacenamiento y optimización de las redes, tratando colecciones de nodos como colecciones de tomos. Psicólogos como David Rumelhart y Geoff Hinton continuaron con el estudio de modelos de memoria basados en redes neuronales. En la década de los 80, al menos cuatro grupos distintos reinventaron el algoritmo de aprendizaje de retroalimentación. El algoritmo se aplicó a diversos problemas de aprendizaje en los campos de la informática y la psicología y los resultados fueron publicados. Parallel Distributed Processing, 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). Es importante reconocer que la manipulación de los símbolos desempeña algún papel justificable en determinados modelos de cognición.

La IA adopta el método científico (1987-presente)

Actualmente es más usual el desarrollo sobre teorías que la propuesta de nuevas. La IA se fundó en parte en el marco de una rebelión contra de las limitaciones de los campos existentes como la teoría de control o la estadística, y ahora abarca estos campos. En los primeros años de la IA parecía perfectamente posible que las nuevas

formas de la computación simbólica; 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.

La IA ya forma parte del ámbito de los métodos científicos. El uso de Internet y el compartir repositorios de datos de prueba y código, ha hecho posible que ahora se puedan contrastar experimentos.

En la década de los 70 se sometió a prueba una gran variedad de arquitecturas y enfoques. Muchos de ellos fueron resultaban frágiles, y fueron probados sólo en unos pocos ejemplos elegidos especialmente. MMO, han pasado a dominar el área. Estos tienen las características de tener rigurosa teoría matemática, lo cual ha permitido a los investigadores del lenguaje basarse en los resultados de investigaciones matemáticas hechas en otros campos a lo largo de varias décadas. En segundo lugar, los modelos se han generado mediante un proceso de aprendizaje en grandes corpus de datos de lenguaje reales.

La tecnología del habla y el campo relacionado del reconocimiento de caracteres manuscritos están actualmente en transición hacia una generalizada utilización en aplicaciones industriales y de consumo.

Las redes neuronales también siguen esta tendencia. La mayor parte del trabajo realizado con redes neuronales en la década de los 80 se realizó con la idea de dejar a un lado lo que se podía hacer y de descubrir en qué se diferenciaban las redes neuronales de otras técnicas "tradicionales". Se mejoraron las metodologías y los marcos teóricos. Como resultado de estos desarrollos, la tecnología denominada minería de datos ha generado una nueva y vigorosa industria.

Quizás animados por el progreso en la resolución de subproblemas de IA, los investigadores han comenzado a trabajar de nuevo en el problema del "agente total". El trabajo de Allen Newell, John Laird, y Paul Rosenbloom es el ejemplo mejor conocido de una arquitectura de agente completa. El llamado 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. Existen otros como por ejemplo motores de búsqueda, sistemas de recomendación, y los sistemas

para la construcción de portales Web. En particular, ahora se cree mayoritariamente que los sistemas sensoriales (visión, sonar, reconocimiento del habla, etc.) no pueden generar información totalmente fidedigna del medio en el que habitan.

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

A un centenar de millones de millas de la Tierra, el programa de la NASA Agente Remoto se convirtió en el primer programa de planificación autónoma a bordo que controlaba la planificación de las operaciones de una nave espacial desde abordó.

Juegos:

Deep Blue de IBM fue el primer sistema que derrotó a un campeón mundial en una partida de ajedrez cuando superó a Garry Kasparov. Este dijo que había percibido un nuevo tipo de inteligencia al otro lado del tablero.

Control autónomo:

Un sistema fue entrenado para dirigir un coche de forma que siguiese una línea.

Diagnóstico:

Los programas de diagnóstico médico basados en el análisis probabilista han llegado a alcanzar niveles similares a los de médicos expertos en algunas áreas de la medicina. El creador del programa le sugirió que le preguntase al computador cómo había generado el diagnóstico. La máquina indicó los factores más importantes en los que había basado su decisión y explicó la ligera interacción existente entre varios de los síntomas en este caso.

Planificación logística: durante la crisis del Golfo Pérsico de 1991, las fuerzas de Estados Unidos desarrollaron la herramienta Dynamic Analysis and Replanning Tool 32 para automatizar la planificación y organización logística del transporte. Incluyó hasta 50.000 vehículos, carga y personal a la vez, teniendo en cuenta puntos de partida, destinos, rutas y la resolución de conflictos entre otros parámetros. Las técnicas de planificación de IA permitieron que se generara un plan en cuestión de horas que podría haber llevado semanas con otros métodos.

Robótica: muchos cirujanos utilizan hoy en día asistentes robot en operaciones de microcirugía.

Estos son algunos de los ejemplos de sistemas de inteligencia artificial que existen hoy en día. No se trata de magia o ciencia ficción, son más bien ciencia, ingeniería y matemáticas, para los que este libro proporciona una introducción.

Agentes inteligentes

Introducción

Uno de los conceptos fundamentales dentro de la IA son los agentes. Un agente es cualquier elemento capaz de percibir su ambiente a través de sensores y actuadores. El agente responde a cambios en el ambiente según diferentes condiciones y criterios, más adelante profundizaremos en los tipos de agentes. Un agente se diferencia de un programa de computación debido a que el agente es capaz de actuar por sí solo (autónoma), tomar decisiones, aprender y modificar su funcionamiento a lo largo del tiempo para solucionar un problema de la mejor manera posible.

¿Cómo deben actuar los agentes?

Un agente racional es un agente cuyo objetivo es hacer lo correcto para solucionar un problema dentro de un determinado ambiente, lo cual nos lleva a la pregunta casi obligada: ¿Cómo medimos el éxito de ese agente?

Necesitamos una performance measure o una medida del rendimiento; la cual nos dirá que tan exitoso es un agente. Esta medida es solo un elemento de cuatro que debemos considerar para obtener un agente racional; estos son:

Medida del rendimiento

Secuencia de percepción: todo lo que el agente ha captado a través de sus sensores -Lo que el agente sabe sobre el ambiente

-Las acciones que el agente puede realizar

Con esta información, podemos llegar a la siguiente definición: un agente racional es aquel que para cada secuencia de percepción deberá realizar una acción que tiene como objetivo maximizar la medida del rendimiento.

A partir de este punto es importante considerar lo siguiente: las acciones de un agente dependen de la secuencia de percepción, es decir, de los elementos que perciba del ambiente a través de sus sensores.

Existen dos elementos importantes dentro de la vida de un agente: función de mapeo y autonomía. Primero, la función de mapeo (mapping) es una función encargada de controlar las acciones de un agente; es decir, esta le dirá al agente cómo comportarse según lo que está percibiendo en ese momento. Segundo, la autonomía, una característica fundamental, se refiere a que un agente es autónomo siempre y cuando su

comportamiento y acciones estn determinados por su propia experiencia.

Estructura de los agentes inteligentes

En este punto veremos la utilidad de la Inteligencia Artificial cuyo trabajo ser disear un agent program o programa de agentes que es una funcin que implementa la funcin de mapeo de percepciones a acciones. Este programa correr sobre un dispositivo al cual llamaremos arquitectura. Esta arquitectura puede ser una simple computadora o hardware ms especfico.

$$\text{Agente} = \text{arquitectura} + \text{programa}$$

Tomemos el ejemplo de un agente enfocado en ser un taxista; ara lograr construir este agente tenemos que considerar cinco elementos:

1. Tipo de agente: taxista
2. Sensores: cmaras, velocmetros, GPS, etc.
3. Acciones: Controlar el volante, acelerar, frenar
4. Objetivos/Goals: Maximizar ganancias, seguridad, legal, cmodo
5. Ambiente: Calles, trfico, peatones, semforos, clientes, otros carros, etc

Agent programs:

Todos los programas tendrn el mismo esqueleto: recibirn percepciones del ambiente y generarn acciones. Hay dos particularidades sobre este esqueleto; primero, el agent program recibe nicamente una entrada y es decisin del agente construir la secuencia de percepcin en memoria. Segundo, el objetivo o performance measure no es parte del esqueleto del programa debido a que se aplica externamente para juzgar el comportamiento del agente.

Tipos de agent programs

Agentes de reflejo simple: responden a reglas de accin-condicin en donde actan con base a los eventos del ambiente.

Agentes que llevan un registro del ambiente: estos agentes llevan un registro (internal state) de los eventos que ocurrieron en el ambiente para tomar decisiones con base a los nuevos eventos que est percibiendo.

Agentes basados en objetivos: adems de necesitar una descripcin actual del estado del ambiente, estos agentes toman acciones con base en su objetivo.

Agentes basados en utilidad: agentes que utilizan una funcin de utilidad, la

cual asocia un número real al estado actual o serie de estados y este número nos indica que tan próxima es la acción.

Ambientes

Tipos de ambientes

1. Accesible vs inaccesible: un agente es accesible si sus sensores detectan todos los aspectos relevantes del ambiente para efectuar correctamente una acción.
2. Determinístico vs no determinístico: si el siguiente estado del ambiente está completamente determinado por el estado actual hablamos de un agente determinístico. Sin embargo, si el sistema es inaccesible, tenemos un no determinístico.
3. Episódico vs no episódico: en un ambiente la experiencia del agente puede dividirse en episodios. Cada episodio consiste en un agente percibiendo y actuando; esto significa que si tenemos un entorno episódico las acciones del agente no repercutirán en los episodios posteriores.
4. Estático vs dinámico: si el ambiente cambia mientras el agente actúa, hablamos de un ambiente dinámico; de lo contrario, hablamos de un ambiente estático.
5. Discreto vs continuo: un ambiente es discreto si existe un número limitado de percepciones que los sensores puedan captar.

Cabe destacar que los agentes más difíciles de concretar son los inaccesibles, no deterministas, no episódicos, dinámicos y continuos.

Resolviendo problemas mediante búsqueda

Un agente puede actuar a través del establecimiento de objetivos y secuencias específicas para lograr esos objetivos. Un objetivo, o goal, y el conjunto de medios para alcanzar ese objetivo se le denomina problema y el proceso de exploración se le denomina búsqueda.

Agentes que resuelven problemas o problem solving agent

Problem solving agent: subtipo de agentes que decide qué hacer al encontrar secuencias de acciones para llevarlos a un estado deseado.

Resolución de problemas:

1. Formulación de un objetivo: si durante una secuencia nos encontramos con acciones que lleven a un estado no favorable las podemos descartar por completo.
2. Formulación del problema: proceso de decidir las acciones a seguir.
3. Búsqueda: el objetivo es encontrar la mejor secuencia para llegar a un resultado favorable.
4. Solución: el algoritmo de búsqueda toma un problema y regresa una solución como una secuencia de acciones.
5. Ejecución: se lleva a cabo una secuencia de acciones.

Formulación de problemas

Nos enfocaremos en los diferentes tipos y cantidad de conocimiento que un agente puede tener con respecto a sus acciones y el estado actual en el que está. Recordemos que un agente es un ente que lleva a cabo una serie de acciones basadas en la información percibida por sus sensores. En resumen, las acciones de un agente estarán dadas por la forma en la que este se conecte con el ambiente o entorno.

Tipos de problemas:

1. Problemas de estado simple: el agente conoce perfectamente el estado en el que se encuentra y sabe, por lo tanto, las acciones específicas que debe tomar.
2. Problemas de estado múltiple: cuando el entorno no es completamente conocido, el agente debe ser capaz de razonar sobre las acciones que debe tomar para alcanzar un objetivo dado.
3. Problemas de contingencia: en la realidad es imposible determinar todos los estados en los que se puede encontrar un agente; por esto, el agente debe ser capaz de utilizar sus sensores, detectar el estado y por ende actuar. Es decir, como programadores, no podemos definir una lista casi infinita de acciones que el agente debe tomar si se encuentra en un estado en específico.
4. Problemas de exploración: el agente descubre las consecuencias de sus acciones y decide basado en esas consecuencias.

Problemas y soluciones bien definidos: es crucial que al agente sepa definir el problema en el que se encuentra, es decir, el estado, porque a partir de esto él decide qué acciones llevar a cabo. Para la formulación de un problema necesitamos los siguientes cuatro puntos:

1. Estado inicial: el agente debe saber en donde se encuentra.
2. Operador: describe una acción en términos del estado al que se va a llegar en caso

de implementarse dicha accin.

Nota: estos dos puntos describen el espacio de estado que es el conjunto de estados alcanzables desde el estado inicial al implementarse una lista de acciones. Una ruta es una secuencia de acciones que llevan al agente de un estado a otro. Por lo tanto, el espacio de estado lo Podemos definir c

Estrategias de bsqueda

Las estrategias de bsqueda se basan en cuatro criterios principales:

1. Completa: la estrategia garantiza una solucin donde se necesite.
2. Complejidad temporal: cunto tiempo toma encontrar una solucin?
3. Complejidad espacial: cunta memoria necesita para realizar la bsqueda?
4. Optimizacin: la estrategia encuentra la solucin de mayor calidad en un mar de diferentes soluciones?

Es importante destacar que existen dos tipos principales de bsquedas: bsquedas sin informacin o bsqueda ciega la cual se lleva a cabo cuando el agente no tiene informacin sobre los pasos a seguir desde el estado actual al estado objetivo. Por el otro lado, tenemos las bsquedas con informacin o bsquedas heursticas en donde los agentes toman estrategias considerando informacin extra, como, por ejemplo, las coordenadas geogrficas del estado actual con respecto al estado objetivo para tomar decisiones sobre la ruta. Analizaremos seis estrategias de bsqueda ciega:

1. Bsqueda Breath-first: expande en cada nodo del rbol
2. Bsqueda de costo uniforme: expande el nodo de menor costo dentro de una coleccin de nodos en espera de ser expandidos. Esta estrategia asegura encontrar el costo ms barato.
3. Bsqueda Depth first: siempre expande uno de los nodos en el nivel ms profundo del rbol y expande nodos a niveles ms altos una vez que haya llegado a un camino sin salida.
4. Bsqueda Depth limited: similar a Depth first pero esta estrategia impone un nivel mximo de bsqueda.
5. Bsqueda iterative deepening: en la estrategia del punto 4 surge un inconveniente: cmo saber el mejor nivel de profundidad? Iterative deepening se quita este problema de encima para escoger el mejor nivel de profundidad al probar todos los niveles posibles de manera secuencias: primero el nivel 0, despus el nivel 1 y posteriormente el nivel n.
6. Bsqueda bidireccional: estrategia que busca desde el estado inicial hacia adelante y desde el estado objetivo hacia atrs de tal manera que espera encontrar ambos en un punto intermedio.

Estrategias de bsqueda informada

A la aproximacin general que consideraremos se le llamar bsqueda primero el mejor. La bsqueda primero el mejor es un caso particular del algoritmo general de BSQUEDA-RBOLES o de BSQUEDA-GRAFOS en el cual se selecciona un nodo para la expansin basada en una funcin de evaluacin, $f(n)$. La bsqueda primero el mejor puede implementarse dentro de nuestro marco general de bsqueda con una cola con prioridad, una estructura de datos que mantendr la frontera en orden ascendente de f -valores.

Hay una familia entera de algoritmos de BSQUEDA-PRIMERO-MEJOR con funciones de evaluacin diferentes. Una componente clave de estos algoritmos es una funcin heurstica, denotada $h(n)$:

$h(n)$ = coste estimado del camino ms barato desde el nodo n a un nodo objetivo.

Las funciones heursticas son la forma ms comn de transmitir el conocimiento adicional del problema al algoritmo de bsqueda.

Bsqueda voraz primero el mejor

La bsqueda voraz primero el mejor trata de expandir el nodo ms cercano al objetivo, alegando que probablemente conduzca rpidamente a una solucin. As, evala los nodos utilizando solamente la funcin heurstica:

$$f(n) = h(n)$$

La bsqueda voraz primero el mejor se parece a la bsqueda primero en profundidad en el modo que prefiere seguir un camino hacia el objetivo, pero volver atrs cuando llegue a un callejn sin salida. Sufre los mismos defectos que la bsqueda primero en profundidad, no es ptima, 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 $O(bm)$, donde m es la profundidad mxima del espacio de bsqueda.

Búsqueda A*: minimizar el costo estimado total de la solución

A la forma más ampliamente conocida de la búsqueda primero el mejor se le llama búsqueda A* (pronunciada búsqueda A-estrella). Evalúa los nodos combinando $g(n)$, el costo para alcanzar el nodo, y $h(n)$, el costo de ir al nodo objetivo:

$$f(n) = g(n) + h(n)$$

Ya que la $g(n)$ nos da el costo del camino desde el nodo inicio al nodo n , y la $h(n)$ el costo estimado del camino más barato desde n al objetivo, tenemos:

$$f(n) = \text{costo más barato estimado de la solución a través de } n$$

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 BSQUEDA-RBOLES. 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 costo de alcanzar el objetivo.

Las heurísticas admisibles son por naturaleza optimistas, porque piensan que el costo de resolver el problema es menor que el que es en realidad. Ya que $g(n)$ es el costo exacto para alcanzar n , tenemos como consecuencia inmediata que la $f(n)$ nunca sobrestima el costo verdadero de una solución a través de n .

Una heurística $h(n)$ es consistente si, para cada nodo n y cada sucesor n' de n generado por cualquier acción a , el costo estimado de alcanzar el objetivo desde n' no es mayor que el costo de alcanzar n más el costo estimado de alcanzar el objetivo desde n : $h(n') \leq c(n, a, n') + h(n)$. Esto es una forma de la desigualdad triangular general, que especifica que cada lado de un triángulo no puede ser más largo que la suma de los otros dos lados.

En nuestro caso, el triángulo está formado por n , n' , y el objetivo más cercano a n . Otra consecuencia importante de la consistencia es la siguiente: si $h(n)$ es consistente, entonces los valores de $f(n)$, a lo largo de cualquier camino, no disminuyen. La demostración se sigue directamente de la definición de consistencia. Supongamos

que n es un sucesor de n ; entonces $g(n) = g(n) + c(n, a, n)$ para alguna a , y tenemos $f(n) = g(n) + h(n) = g(n) + c(n, a, n) + h(n) \geq g(n) + h(n) = f(n)$. Se sigue que la secuencia de nodos expandidos por A^* utilizando la BSQUEDA-GRAFOS estn en orden no decreciente de $f(n)$. De ah que, el primer nodo objetivo seleccionado para la expansin debe ser una solucin ptima, ya que todos los nodos posteriores sern al menos tan costosos.

Una observacin final es que entre los algoritmos ptimos de este tipo (los algoritmos que extienden los caminos de bsqueda desde la raz) A^* es ptimamente eficiente para cualquier funcin heurstica.

Bsqueda heurstica con memoria acotada

La forma ms simple de reducir la exigencia de memoria para A^* es adaptar la idea de profundizar iterativamente al contexto de bsqueda heurstica, resultando as el algoritmo A^* de profundidad iterativa (A^*PI). La diferencia principal entre A^*PI y la profundidad iterativa estndar es que el corte utilizado es el f -coste ($g + h$) ms que la profundidad; en cada iteracin, el valor del corte es el f -coste ms pequeo de cualquier nodo que excedi el corte de la iteracin anterior. A^*PI es prctico para muchos problemas con costos unidad y evita el trabajo asociado con el mantenimiento de una cola ordenada de nodos.

La bsqueda recursiva del primero mejor (BRPM) es un algoritmo sencillo recursivo que intenta imitar la operacin de la bsqueda primero el mejor estndar, pero utilizando slo un espacio lineal.

Como A^* , BRPM es un algoritmo ptimo si la funcin heurstica $h(n)$ es admisible. Su complejidad en espacio es $O(bd)$, pero su complejidad en tiempo es bastante difcil de caracterizar: depende de la exactitud de la funcin heurstica y de cmo cambia a menudo el mejor camino mientras se expanden los nodos. Tanto A^*PI como BRPM estn sujetos al aumento potencialmente exponencial de la complejidad asociada con la bsqueda en grafos. A^*PI y BRPM sufren de utilizar muy poca memoria. Entre iteraciones, A^*PI conserva slo un nmero: el lmite f -coste actual. BRPM conserva ms informacin en la memoria, pero usa slo $O(bd)$ de memoria: incluso si hubiera ms memoria disponible, BRPM no tiene ningn modo de aprovecharse de ello.

A^*MS es completo si hay alguna solucin alcanzable, es decir, si d , la profundidad

del nodo objetivo más superficial, es menor que el tamaño de memoria (expresada en nodos). Es óptimo si cualquier solución óptima es alcanzable; de otra manera devuelve la mejor solución alcanzable. En términos prácticos, A*MS bien podría ser el mejor algoritmo de uso general para encontrar soluciones óptimas, en particular cuando el espacio de estados es un grafo, los costos no son uniformes, y la generación de un nodo es costosa comparada con el gasto adicional de mantener las listas abiertas y cerradas.

Aprender a buscar mejor

El método se apoya sobre un concepto importante llamado el espacio de estados metanivel. Cada estado en un espacio de estados metanivel captura el estado interno (computacional) de un programa que busca en un espacio de estados a nivel de objeto.

Cada acción en el espacio de estados metanivel es un paso de cómputo que cambia el estado interno; por ejemplo, cada paso de cómputo en A* expande un nodo hoja y añade sus sucesores al árbol. Para problemas más difíciles, habrá muchos de estos errores, y un algoritmo de aprendizaje metanivel puede aprender de estas experiencias para evitar explorar subárboles no prometedores.

El objetivo del aprendizaje es reducir al mínimo el coste total de resolver el problema, compensar el costo computacional y el coste del camino.

Conclusiones

Barredo Olguín Patricio

Para desarrollar el algoritmo utilizamos el algoritmo disktra por la eficiencia que nos provee para encontrar rutas más cortas. Para la parte del grafo, pensamos que cada nodo fuese una estación del metro que permite conexión con otra, y el arista que fuese el tiempo en tardar de llegar de una estación a otra, sin embargo por la cantidad muy grande de datos no se logró terminar. Se desarrolló la mayor parte del algoritmo sin embargo una parte complicada fue desarrollar la matriz de asociada al grafo, pero por la cantidad de datos no se pudo implementar. Se realizó una interfaz gráfica, pero no se pudo conectar con el algoritmo.

Carballido García Alicia

Realizar el trabajo de investigación previo al algoritmo fue la parte esencial para poder trabajar en la solución del problema. Los conceptos aprendidos durante la

bsqueda de informacin, nos facilitaron el entendimiento y nos proporcion las herramientas necesarias para comprender el problema y saber cmo afrontarlo. Utilizamos el algoritmo de Disktra, el cul cumple con la funcin de buscar las rutas ms cortas en un grafo. La cantidad de datos para solventar el problema era exuberante, por lo que no se concluy. Sin embargo, la base del algoritmo est lista. Se trat de sintetizar lo mejor posible el contenido de la investigacin para tener una mejor comprensin.

Martínez Flores Emanuel

El proyecto se cumpli satisfactoriamente, debido a que logramos nuestros objetivos principales, que eran desarrollar la aplicacin para conocer la ruta ms corta en el Sistema de Transporte Colectivo (metro) de la Ciudad de México, además de adquirir más conocimientos en MATLAB así como conocer los fundamentos y las aplicaciones de la IA. Debido a la gran cantidad de teoría e investigación que involucro el desarrollo, me pude dar cuenta que en este proyecto en específico lo que más importa es reducir los costos en cuanto a tiempo, ya que muchas veces los bocetos que nos pueden llegar a mostrar no son fidedignos con la realidad, por eso nuestra aplicacin lo hace todo de una manera en la que en realidad buscara de manera eficiente las rutas que podríamos tomar. Por otro lado tocando el tema de la IA fue interesante aprender como es que la computadora puede aprender por medio de funciones, y que una de las cosas más importantes para esto son los diferentes tipos de bsqueda, ya que muchas veces la IA aprende de los errores que va cometiendo y de los aciertos, hasta que llega a aprender y mejorar por sí sola.

Martínez Ostúa Néstor

Con este trabajo pude presenciar el alcance y poder de la inteligencia artificial, que sin lugar a dudas es una herramienta muy poderosa para resolver casi cualquier problema de la vida moderna. Es cierto, vivimos en un mundo rodeado de tecnología e innovación, y es justo por eso que la IA juega un papel aún más fundamental. Como ingenieros en computación debemos abordar esta nueva revolución, la tan llamada Cuarta Revolución Industrial, con herramientas como la IA, para poder resolver los problemas más demandantes de nuestra civilización. Ahora, si bien este trabajo apenas abord los temas más básicos, es sin duda alguna, un trabajo ya de una complejidad interesante puesto que, por un lado la investigación la tuvimos que realizar por nuestra cuenta lo cual tom bastante tiempo. Un problema que encontramos fue lograr sintetizar adecuadamente los capítulos, especialmente la historia de la IA y las estrategias de bsqueda. Temas que dan para un trabajo de más de 40 cuartillas pero tuvimos que ser meticulosos a la hora de sintetizar. Por el otro lado, el desarrollo aplicativo fue un gran reto puesto que plasmar cada una de las estaciones del metro fue demandante. En cuanto al algoritmo, creo que hicimos una buena implementación y se ve demostrado en nuestros resultados.