



**TECNOLÓGICO
NACIONAL DE MÉXICO**



INSTITUTO TECNOLÓGICO DE CULIACÁN
INGENIERIA EN SISTEMAS COMPUTACIONALES

INTELIGENCIA ARTIFICIAL
HISTORIA DE LA INTELIGENCIA ARTIFICIAL

MAESTRO

ZURIEL DATHAN MORA FELIX

AYALA RODRÍGUEZ JOSÉ ERNESTO
LIZARRAGA VALENZUELA JESUS EDUARDO

HORA:

11:00-12:00

07 de febrero del 2025

Culiacán, Sinaloa

HISTORIA DE LA INTELIGENCIA ARTIFICIAL

La "Edad Oscura" o el nacimiento de la inteligencia artificial (1943-1956)

El primer trabajo reconocido en inteligencia artificial (IA) fue presentado en 1943 por Warren McCulloch y Walter Pitts, quienes propusieron un modelo de redes neuronales artificiales basado en estados binarios (encendido/apagado). Demostraron que su modelo era equivalente a una máquina de Turing y que cualquier función computable podía resolverse con una red neuronal. También evidenciaron que estructuras de red simples podían aprender.

Sin embargo, los experimentos mostraron que el modelo binario de las neuronas no era correcto, ya que estas poseen características no lineales. A pesar de ello, McCulloch sentó las bases de la computación neuronal y las redes neuronales artificiales (ANN).

John von Neumann, influenciado por este modelo, apoyó a Marvin Minsky y Dean Edmonds en la construcción de la primera computadora basada en redes neuronales en 1951. Otro investigador clave, Claude Shannon, publicó en 1950 un artículo sobre máquinas de ajedrez, donde demostró la complejidad del juego y la necesidad de heurísticas para encontrar soluciones.

Finalmente, John McCarthy, considerado otro de los padres de la IA, organizó en 1956 un taller en Dartmouth College junto a Minsky y Shannon, estableciendo formalmente el campo de la inteligencia artificial. Durante las siguientes dos décadas, la IA fue dominada por los investigadores de este evento.

El auge de la inteligencia artificial o la era de las grandes expectativas (1956-finales de la década de 1960)

Durante este período, se mejoraron los métodos de aprendizaje. Frank Rosenblatt demostró el teorema de convergencia del perceptrón, mostrando que su algoritmo podía ajustar la fuerza de conexión de un perceptrón (Rosenblatt, 1962).

Uno de los proyectos más ambiciosos fue el General Problem Solver (GPS), desarrollado por Allen Newell y Herbert Simon en la Universidad Carnegie Mellon. Este programa buscaba simular los métodos humanos de resolución de problemas, separando la técnica de solución de problemas de los datos. GPS utilizaba el análisis *means-ends*, definiendo problemas en términos de estados y aplicando operadores para reducir la diferencia entre el estado actual y el objetivo.

Sin embargo, GPS no logró resolver problemas complejos debido a su dependencia de la lógica formal, lo que generaba un número infinito de operadores posibles y lo volvía ineficiente. Además, su alto consumo de recursos computacionales llevó a su abandono.

Promesas incumplidas o el impacto de la realidad (finales de la década de 1960 - inicios de la década de 1970)

En 1965, Lotfi Zadeh, profesor de la Universidad de California en Berkeley, publicó su artículo *Fuzzy Sets*, sentando las bases de la teoría de conjuntos difusos. Dos décadas después, esta teoría se aplicaría en el desarrollo de máquinas y sistemas inteligentes basados en lógica difusa.

Para 1970, la euforia por la inteligencia artificial (IA) había desaparecido, y muchos proyectos financiados por el gobierno fueron cancelados. La IA seguía siendo un campo académico con pocas aplicaciones prácticas, salvo en juegos.

Las principales dificultades que enfrentó la IA en esta época fueron:

- Falta de conocimiento específico: Los primeros programas aplicaban estrategias de búsqueda genéricas en lugar de utilizar conocimiento del dominio, lo que los volvía ineficientes en problemas complejos. La teoría de la NP-completitud, desarrollada en los años 70, demostró que muchos problemas no podían resolverse fácilmente, incluso con computadoras más rápidas.
- Problemas demasiado amplios y difíciles: Un ejemplo fue la traducción automática. Tras el lanzamiento del Sputnik en 1957, EE. UU. financió la traducción de textos científicos rusos, pero pronto se descubrió que la mera sustitución de palabras no era suficiente, ya que la traducción requiere comprensión del contexto. En 1966, el gobierno canceló estos proyectos.
- Falta de apoyo gubernamental: En 1971, el gobierno británico también suspendió su apoyo a la IA. Un informe de Sir James Lighthill concluyó que la investigación en IA no había producido resultados significativos, cuestionando la necesidad de una ciencia independiente para este campo.

La tecnología de los sistemas expertos o la clave del éxito (inicios de la década de 1970 - mediados de la década de 1980)

Un equipo conformado por Edward Feigenbaum, Bruce Buchanan y Joshua Lederberg desarrolló DENDRAL, un sistema que marcó un cambio de paradigma en la inteligencia artificial (IA).

Importancia de DENDRAL

- Introdujo técnicas específicas y especializadas, dejando atrás los métodos débiles de propósito general.
- Su objetivo era igualar el desempeño de un químico experto, utilizando reglas heurísticas obtenidas de especialistas humanos.

- Dio origen a la metodología de los sistemas expertos, conocida como "ingeniería del conocimiento".

DENDRAL se convirtió en una herramienta útil para químicos y llegó a comercializarse en EE. UU.

El proyecto MYCIN

Desarrollado en Stanford desde 1972, MYCIN fue un sistema experto basado en reglas para el diagnóstico de enfermedades infecciosas de la sangre. También proporcionaba asesoramiento terapéutico a los médicos de manera eficiente.

PROSPECTOR: Exploración de minerales

Entre 1974 y 1983, en el Instituto de Investigación de Stanford, se creó PROSPECTOR, un sistema basado en reglas y redes semánticas que utilizaba conocimientos de expertos en geología para evaluar posibles yacimientos minerales.

Auge y limitaciones de los sistemas expertos

Para finales de los años 70, los sistemas expertos demostraron el potencial comercial de la IA, pero presentaban limitaciones:

- Dominio restringido: MYCIN solo diagnosticaba enfermedades infecciosas de la sangre.
- Falta de flexibilidad: No podían adaptarse a problemas fuera de su dominio.
- Explicaciones limitadas: Mostraban la secuencia de reglas aplicadas, pero sin un razonamiento profundo.
- Validación difícil: No existían técnicas generales para comprobar la consistencia del conocimiento.
- Falta de aprendizaje: No podían mejorar con la experiencia.
- Largo tiempo de desarrollo: Construir un sistema experto podía tomar más de 30 años-persona.

Cómo hacer que una máquina aprenda o el renacimiento de las redes neuronales (mediados de la década de 1980 en adelante)

El retraso en el desarrollo de redes neuronales artificiales se debió a la falta de computadoras suficientemente potentes para modelarlas y experimentarlas. Sin embargo, en los años 80, la necesidad de procesar información de manera similar al cerebro, junto con avances en tecnología computacional y neurociencia, llevó a un resurgimiento de las redes neuronales.

Contribuciones clave en teoría y diseño:

- Grossberg (1980) estableció la teoría de la resonancia adaptativa, base para una nueva clase de redes neuronales.
- Hopfield (1982) introdujo las redes neuronales con retroalimentación, conocidas como redes de Hopfield.
- Kohonen (1982) publicó su trabajo sobre mapas autoorganizados.
- Barto, Sutton y Anderson (1983) desarrollaron el aprendizaje por refuerzo y su aplicación en control.

El algoritmo de retropropagación, originalmente propuesto por Bryson y Ho en 1969, fue redescubierto en 1986 por Rumelhart y McClelland en su libro *Parallel Distributed Processing*. Casi al mismo tiempo, Parker (1987) y LeCun (1988) también lo redescubrieron, convirtiéndolo en la técnica más popular para entrenar perceptrones multicapa.

En 1988, Broomhead y Lowe desarrollaron un procedimiento para diseñar redes de alimentación hacia adelante con funciones de base radial, una alternativa a los perceptrones multicapa.

Computación evolutiva o aprender haciendo (inicios de la década de 1970 en adelante)

El concepto de algoritmos genéticos fue introducido por John Holland en los años 70. Su método manipulaba "cromosomas" artificiales (cadenas de dígitos binarios) mediante operaciones como selección, cruce y mutación. Estos algoritmos se basan en el Teorema del Esquema (Holland, 1975; Goldberg, 1989).

En los años 60, Ingo Rechenberg y Hans-Paul Schwefel, de la Universidad Técnica de Berlín, desarrollaron estrategias evolutivas, diseñadas para resolver problemas de optimización en ingeniería mediante cambios aleatorios en los parámetros, imitando la mutación natural (Rechenberg, 1965). Su enfoque, similar a una búsqueda Monte Carlo enfocada, representa una alternativa a la intuición del ingeniero.

Tanto los algoritmos genéticos como las estrategias evolutivas han demostrado ser soluciones robustas para problemas complejos de búsqueda y optimización no lineal (Holland, 1995; Schwefel, 1995).

La programación genética aplica los principios de los algoritmos genéticos a la programación, generando automáticamente código en lugar de simplemente optimizar representaciones codificadas de un problema.

En los años 90, John Koza impulsó este campo utilizando operaciones genéticas para manipular código en LISP (Koza, 1992, 1994). Su objetivo es lograr que las computadoras resuelvan problemas sin ser programadas explícitamente.

La nueva era de la ingeniería del conocimiento o la computación con palabras (finales de la década de 1980 en adelante)

Una tecnología clave para manejar datos vagos, imprecisos e inciertos es la lógica difusa. Mientras que los sistemas expertos clásicos utilizaban probabilidad (como los factores de certeza de MYCIN o las reglas de Bayes en PROSPECTOR), la lógica difusa permite razonar con términos cualitativos como "generalmente", "a veces" o "raramente", reflejando mejor el pensamiento humano.

El concepto central es la variable lingüística, cuyos valores son palabras en lugar de números. Los sistemas difusos emplean reglas del tipo SI-ENTONCES, como:

- SI la velocidad es alta ENTONCES la distancia de frenado es larga.
- SI la velocidad es baja ENTONCES la distancia de frenado es corta.

La teoría de conjuntos difusos fue introducida en 1965 por Lotfi Zadeh, profesor de la Universidad de Berkeley (Zadeh, 1965). A pesar de la resistencia inicial en Occidente debido a su nombre "difuso", Japón adoptó la tecnología en la década de 1980, integrándola en lavadoras, aires acondicionados, televisores, fotocopiadoras y automóviles desde 1987.

Beneficios de la lógica difusa en sistemas basados en conocimiento

1. Mayor poder computacional
 - Requiere menos reglas que los sistemas expertos tradicionales.
 - Mejora la capacidad de resolver problemas no lineales y complejos.
 - Se prevé que la mayoría de los sistemas expertos en el futuro incorporarán lógica difusa.
2. Mejor modelado cognitivo
 - Permite capturar el conocimiento experto de forma más natural.
 - No requiere fragmentar la experiencia en reglas estrictas, mejorando el rendimiento en problemas complejos.
3. Capacidad para representar múltiples expertos
 - Los sistemas expertos tradicionales dependen de un único experto.

- Los sistemas difusos pueden combinar conocimientos de múltiples expertos, incluso cuando hay desacuerdos.
- Son especialmente útiles en áreas como negocios y gestión, donde no existen respuestas únicas.